Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2023-0184

Vol. **35**, No. 1, pp. 212-238
January 2024

Domain Decomposition Methods for Diffusion Problems with Discontinuous Coefficients Revisited

Xuyang Na and Xuejun Xu*

Key Laboratory of Intelligent Computing and Applications (Tongji University), Ministry of Education and School of Mathematical Science, Tongji University, Shanghai 200092, China.

Received 8 July 2023; Accepted (in revised version) 3 October 2023

Abstract. In this paper, we revisit some nonoverlapping domain decomposition methods for solving diffusion problems with discontinuous coefficients. We discover some interesting phenomena, that is, the Dirichlet-Neumann algorithm and Robin-Robin algorithms may make full use of the ratio of coefficients in some special cases. Detailedly, in the case of two subdomains, we find that their convergence rates are $\mathcal{O}(\nu_1/\nu_2)$ if $\nu_1 < \nu_2$, where ν_1,ν_2 are coefficients of two subdomains. Moreover, in the case of many subdomains with red-black partition, the condition number bounds of Dirichlet-Neumann algorithm and Robin-Robin algorithm are $1+\epsilon(1+\log(H/h))^2$ and $C+\epsilon(1+\log(H/h))^2$, respectively, where ϵ equals $\min\{\nu_R/\nu_B,\nu_B/\nu_R\}$ and ν_R,ν_B are the coefficients of red and black domains. By contrast, Neumann-Neumann algorithm and Dirichlet-Dirichlet algorithm could not obtain such good convergence results in these cases. Finally, numerical experiments are preformed to confirm our findings.

AMS subject classifications: 65N30, 65N55

Key words: Diffusion problem, discontinuous coefficients, finite elements, domain decomposition.

1 Introduction

Diffusion problem is a quite important model which is encountered in many physical problems and practical application fields. It is of great significance to solve diffusion equations numerically. One of the difficulties is that the diffusion coefficients are usually strongly discontinuous. A natural choice to overcome the difficulty is to use nonoverlapping domain decomposition (DD) methods to solve such kind of problems. Actually, there are lots of literature in the study of solving strong discontinuous problems

*Corresponding author. *Email addresses:* 2010166@tongji.edu.cn (X. Na), xxj@lsec.cc.ac.cn (X. Xu)

by nonoverlapping DD methods. For instance, Mandel and Brezina [6] develop a balancing domain decomposition method for steady-state diffusion problem. In [4], a FETI algorithm is proposed and it is proved that the bounds on the rate of convergence are independent of possible jumps of the coefficients. In [8, 9], Sarkis design Schwarz preconditioners for discontinuous coefficients problems by using both conforming and nonconforming elements. In [3], a Robin-Robin preconditioner is proposed for advection-diffusion problems with discontinuous coefficients. For more study of this aspect, we refer to [7, 11] and the references cited therein.

We may find that the algorithms in most of the literature achieve convergence rates or condition number bounds independent of the jumps of coefficients. Wether there is a better result? For general cases, it could not be improved. However, we find a better result in some special cases, that is, the discontinuous coefficients may accelerate the convergence of Dirichlet-Neumann (D-N) algorithm and Robin-Robin (R-R) algorithm in the case of two subdomains and the case of many subdomains with red-black partition. Detailedly, if we suppose ν_1,ν_2 are the discontinuous coefficients in the case of two subdomains, then the convergence rates of the D-N algorithm and the R-R algorithm will completely depend on the ratio of the smaller coefficient to the larger coefficient, i.e. ν_1/ν_2 if $\nu_1<\nu_2$. Here we should clear that unlike the discontinuous coefficients case, the convergence rates of D-N algorithm and R-R algorithm are bounded by a constant which is independent of mesh size h and less than 1 strictly in the case ν_1 equals ν_2. In the case of many subdomains with red-black partition, the D-N algorithm and the R-R algorithm are always regarded as preconditioned methods and the corresponding condition number bounds are $1+\epsilon(1+\log(H/h))^2$ and $C+\epsilon(1+\log(H/h))^2$, respectively, where ϵ only depends on the ratio of the coefficients of red domains and black domains. Gander and Dubois [2] also find a similar phenomenon in the case of two symmetric subdomains. But they use the Fourier analysis to analyze it, as a result, their result is hard to extend to general cases. Meanwhile, we estimate the convergence rate by analyzing the spectra radium of error reduction operators and analyzing the condition numbers of preconditioned systems. Finally, all the results are confirmed by numerical experiments.

The paper is organized as follows: In Section 2, we introduce the model problem and domain decomposition methods. In Section 3, we analyze the influence of coefficients on convergence rates in the case of two subdomains with subdomains symmetric and nonsymmetric. In Section 4, the preconditioned systems in the case of many subdomains with red-black partition are described and the bounds on the condition numbers are given. Finally, we perform several numerical experiments to verify our conclusions.

2 Model problems and domain decomposition algorithms

We consider the following elliptic problem with discontinuous coefficients:

$$\begin{cases} -\nabla\cdot(\nu(\mathbf{x})\nabla u)=f, & \text{in } \Omega, \\ u=0, & \text{on } \partial\Omega, \end{cases} \tag{2.1}$$

where Ω is a bounded, two-dimensional polygonal domain and the diffusion coefficient $\nu(\mathbf{x})$ is a piecewise constant function

$$\nu(\mathbf{x})=\begin{cases}\nu_1, & \mathbf{x}\in\Omega_1,\\ \nu_2, & \mathbf{x}\in\Omega_2.\end{cases}$$

Here Ω_1,Ω_2 are nonoverlapping subdomains which form a decomposition of Ω and Γ denotes their common interface, i.e. $\Gamma=\partial\Omega_1\cap\partial\Omega_2$.

Let $\mathcal{T}_h$ be a quasi-uniform and regular triangulation of Ω with the mesh size h and assume that interface Γ does not cut through any elements of $\mathcal{T}_h$. Let $W\subset H_0^1(\Omega)$ be a P1 conforming finite element space over $\mathcal{T}_h$. Besides, we need the following finite element spaces,

$$W_i=W\cap H^1(\Omega_i),\quad W_i^0=W\cap H_0^1(\Omega_i),\quad i=1,2,$$

and the space of the interface Γ,

$$V_\Gamma=W|_\Gamma.$$

Then, the weak form of (2.1) is as follows: Find $u\in W$, such that

$$a(u,v)=(f,v)\quad\forall v\in W,$$

where

$$a_i(u,v)=\int_{\Omega_i}\nu_i\nabla u\cdot\nabla v\quad\forall u,v\in W_i,\quad i=1,2,$$
$$(f,v)_i=\int_{\Omega_i}fv\quad\forall v\in W_i,\quad i=1,2,$$

and

$$a(u,v)=a_1(u,v)+a_2(u,v),\quad (f,v)=(f,v)_1+(f,v)_2.$$

We also use the following bilinear form on the interface,

$$\langle u,v\rangle=\int_\Gamma uv\quad\forall u,v\in V_\Gamma.$$

The model problem may be written equivalently in the following multidomain formulation:

$$\begin{cases}-\nu_1\Delta u=f, & \text{in }\Omega_1,\\ u_1=u_2, & \text{on }\Gamma,\\ \nu_1\dfrac{\partial u_1}{\partial\mathbf{n}_1}=-\nu_2\dfrac{\partial u_2}{\partial\mathbf{n}_2}, & \text{on }\Gamma,\\ -\nu_2\Delta u=f, & \text{in }\Omega_2.\end{cases}$$

The second and the third equations corresponding to Dirichlet and Neumann boundary conditions are imposed to ensure the continuity of the solution and the flux across the interface Γ. To solve the multidomain problem, we have the following three iterative methods and we would like to write them into weak forms.

Algorithm 2.1 (The Dirichlet-Neumann Algorithm [11]). Given $u_\Gamma^0(=0)\in V_\Gamma$, compute as the following steps until converge:

Step 1: solve the Dirichlet problem in Ω_1,

$$\begin{cases} a_1(u_1^{n+1},v)=(f,v)_1, & \forall v\in W_1^0,\\ u_1^{n+1}=0, & \text{on } \partial\Omega_1\backslash\Gamma,\\ u_1^{n+1}=u_\Gamma^n, & \text{on } \Gamma; \end{cases}$$

Step 2: solve the Neumann problem in Ω_2,

$$\begin{aligned} a_2(u_2^{n+1},v)&=(f,v)_2-\left\langle \nu_1\frac{\partial u_1^{n+1}}{\partial \mathbf{n}_1},v\right\rangle\\ &=(f,v)_2+(f,T_1\gamma_0 v)_1-a_1(u_1^{n+1},T_1\gamma_0 v)\quad \forall v\in W_2, \end{aligned}$$

where $T_i: V_\Gamma\to W_i$ is an arbitrary extension operator and $\gamma_0: W_1\to V_\Gamma$ is the trace operator;

Step 3: get the next iterate by a relaxation,

$$u_\Gamma^{n+1}=\theta u_2^{n+1}+(1-\theta)u_\Gamma^n \quad \text{on } \Gamma,$$

with an appropriate θ.

Algorithm 2.2 (The Neumann-Neumann Algorithm [11]). Given $u_\Gamma^0(=0)\in V_\Gamma$, compute as the following steps until converge:

Step 1: solve the Dirichlet problems in Ω_i, $i=1,2$,

$$\begin{cases} a_i(u_i^{n+1},v)=(f,v)_i, & \forall v\in W_i^0,\\ u_i^{n+1}=0, & \text{on } \partial\Omega_i\backslash\Gamma,\\ u_i^{n+1}=u_\Gamma^n, & \text{on } \Gamma, \end{cases}$$

Step 2: solve the Neumann problems in Ω_i, $i=1,2$,

$$\begin{aligned} a_i(w_i^{n+1},v)&=\delta_i^\dagger\left\langle \nu_1\frac{\partial u_1^{n+1}}{\partial \mathbf{n}_1}+\nu_2\frac{\partial u_2^{n+1}}{\partial \mathbf{n}_2},v\right\rangle\\ &=\delta_i^\dagger\left(a_1(u_1^{n+1},T_1\gamma_0^i v)-(f,T_1\gamma_0^i v)_1+a_2(u_2^{n+1},T_2\gamma_0^i v)-(f,T_2\gamma_0^i v)_2\right)\quad \forall v\in W_i, \end{aligned}$$

where $\delta_1^\dagger$ and $\delta_2^\dagger$ are positive weights with $\delta_1^\dagger+\delta_2^\dagger=1$ and $\gamma_0^i: W_i\to V_\Gamma$ is the trace operator;

Step 3: get the next iterate by a relaxation,

$$u_\Gamma^{n+1}=u_\Gamma^n-\theta(\delta_1^\dagger w_1^{n+1}+\delta_2^\dagger w_2^{n+1}),$$

with an appropriate θ.

Algorithm 2.3 (The Dirichlet-Dirichlet Algorithm [11]). Given $\lambda_\Gamma^0(=0)\in V_\Gamma$, compute as the following steps until converge:

Step 1: set $\lambda_1^n=-\lambda_2^n=\lambda_\Gamma^n$, solve the Neumann problems with in Ω_i, $i=1,2$,

$$a_i(u_i^{n+1},v)=(f,v)_i+\langle\lambda_i^n,v\rangle \quad \forall v\in H_\Gamma^1(\Omega_i);$$

Step 2: solve the Dirichlet problem in Ω_i, $i=1,2$,

$$\begin{cases} a_i(w_i^{n+1},v)=0, & \forall v\in W_i^0,\\ w_i^{n+1}=0, & \text{on } \partial\Omega_i\backslash\Gamma,\\ w_i^{n+1}=\delta_i^\dagger(u_1^{n+1}-u_2^{n+1}), & \text{on } \Gamma, \end{cases}$$

where $\delta_1^\dagger$ and $\delta_2^\dagger$ are positive weights with $\delta_1^\dagger+\delta_2^\dagger=1$;

Step 3: get the next iterate by a relaxation,

$$\lambda_\Gamma^{n+1}=\lambda_\Gamma^n-\theta\Big(\delta_1^\dagger\nu_1\frac{\partial w_1^{n+1}}{\partial\mathbf{n}_1}+\delta_2^\dagger\nu_2\frac{\partial w_2^{n+1}}{\partial\mathbf{n}_2}\Big),$$

with an appropriate θ.

The matching conditions may be changed equivalently by the combinations of the Dirichlet and Neumann interface conditions as follows:

$$\begin{cases} \gamma_1u_1+\nu_1\dfrac{\partial u_1}{\partial\mathbf{n}_1}=\gamma_1u_2+\nu_2\dfrac{\partial u_2}{\partial\mathbf{n}_1}=:g_1, & \text{on } \Gamma,\\ \gamma_2u_2+\nu_2\dfrac{\partial u_2}{\partial\mathbf{n}_2}=\gamma_2u_1+\nu_1\dfrac{\partial u_1}{\partial\mathbf{n}_2}=:g_2, & \text{on } \Gamma, \end{cases}$$

where the Robin parameters γ_1,γ_2 are positive numbers. Therefore, we have the following Robin-Robin algorithm.

Algorithm 2.4 (The Robin-Robin Algorithm [1]). Given $g_1^0(=0)\in V_\Gamma$, $\gamma_1,\gamma_2>0$, compute as the following steps until converge:

Step 1: solve the problem with Robin boundary condition in Ω_1,

$$a_1(u_1^n,v)+\gamma_1\langle u_1^n,v\rangle=(f,v)_1+\langle g_1^n,v\rangle \quad \forall v\in W_1;$$

Step 2: update the interface condition,

$$g_2^n=(\gamma_1+\gamma_2)u_1^n-g_1^n \quad \text{on } \Gamma;$$

Step 3: solve the problem with Robin boundary condition in Ω_2,

$$a_2(u_2^n,v)+\gamma_2\langle u_2^n,v\rangle=(f,v)_2+\langle g_2^n,v\rangle \quad \forall v\in W_2;$$

Step 4: update the interface condition,

$$\tilde{g}_1^n=(\gamma_1+\gamma_2)u_2^n-g_2^n \quad \text{on } \Gamma;$$

Step 5: get the next iterate by a relaxation,

$$g_1^{n+1}=\theta\tilde{g}_1^n+(1-\theta)g_1^n,$$

with an appropriate θ.

3 Influence of discontinuous coefficients on convergence rates

In this section, we will explore the influence of discontinuous coefficients on convergence rates and confirm the optimal parameters of the algorithms in the previous section.

First, we give some preliminaries. Define $\mathcal{H}_i: V_\Gamma \to W_i$ as follows:

$$\begin{cases} a_i(\mathcal{H}_i u_\Gamma, v) = 0, & \forall v \in W_i^0, \\ \mathcal{H}_i u_\Gamma = 0, & \text{on } \partial\Omega_i \backslash \Gamma, \\ \mathcal{H}_i u_\Gamma = u_\Gamma, & \text{on } \Gamma. \end{cases}$$

The operator $\mathcal{H}_i$ is known as the 'discrete harmonic extension'. We note that the coefficient ν_i in $a_i(\cdot,\cdot)$ can be omitted because of the zero source term. Define S_i to be a linear operator as follows:

$$\langle S_i u_\Gamma, v_\Gamma \rangle = a_i(\mathcal{H}_i u_\Gamma, T_i v_\Gamma) \quad \forall v_\Gamma \in V_\Gamma,$$

where $T_i: V_\Gamma \to W_i$ is an arbitrary extension operator. Obviously, S_i is symmetric and positive definite. Then, we will give the error operators of the four DD algorithms in the following lemma. Actually, the proof of the following lemma may be found in [11] and [1]. For completeness, we give a brief proof here.

Lemma 3.1. *The error operators of the D-N algorithm, N-N algorithm, D-D algorithm and R-R algorithm are R_1, R_2, R_3 and R_4, respectively, where*

$$\begin{aligned} R_1 &= I - \theta S_2^{-1}(S_1 + S_2), \\ R_2 &= I - \theta(D_1 S_1^{-1} D_1 + D_2 S_2^{-1} D_2)(S_1 + S_2), \\ R_3 &= I - \theta(D_1 S_1 D_1 + D_2 S_2 D_2)(S_1^{-1} + S_2^{-1}) \end{aligned}$$

and

$$\begin{aligned} R_4 &= I - \theta(\gamma_1 I - S_2)(\gamma_2 I + S_2)^{-1}((\gamma_2 I + S_2)(\gamma_1 I - S_2)^{-1} - (\gamma_2 I - S_1)(\gamma_1 I + S_1)^{-1}) \\ &= I - \theta(I - (\gamma_1 I - S_2)(\gamma_2 I + S_2)^{-1}(\gamma_2 I - S_1)(\gamma_1 I + S_1)^{-1}). \end{aligned}$$

Proof. To deduct the error operators, it is sufficient to consider the homogeneous case, $f \equiv 0$, by linearity. For simplicity, we use the same letters to denote the functions and corresponding errors in the proof without causing any confusion.

We first consider the D-N algorithm. From the definition of discrete harmonic extension and Step 1 of Algorithm 2.1, we know $u_1^{n+1} = \mathcal{H}_1 u_\Gamma^n$, then by the definition of S_i and Step 2 of Algorithm 2.1, we have

$$a_2(u_2^{n+1}, T_2\gamma_0 v) = -a_1(u_1^{n+1}, T_1\gamma_0 v) = -\langle S_1 u_\Gamma^n, v \rangle \quad \forall v \in V_\Gamma. \tag{3.1}$$

Here we note that if we set $v = 0$ in (3.1), we have

$$a_2(u_2^{n+1}, w) = 0 \quad \forall w \in W_2^0, \tag{3.2}$$

which reflects that $u_2^{n+1}=\mathcal{H}_2(u_2^{n+1}|_\Gamma)$. Therefore, it holds that

$$\langle S_2(u_2^{n+1}|_\Gamma),v\rangle=-\langle S_1u_\Gamma^n,v\rangle \quad \forall v\in V_\Gamma.$$

Because S_2 is a positive definite operator, we have

$$u_2^{n+1}|_\Gamma=-S_2^{-1}S_1u_\Gamma^n. \tag{3.3}$$

Combining (3.3) and Step 3 of Algorithm 2.1, we obtain the error operator of D-N algorithm, i.e.

$$\begin{aligned}u_\Gamma^{n+1}&=\theta u_2^{n+1}|_\Gamma+(1-\theta)u_\Gamma^n\\&=-\theta S_2^{-1}S_1u_\Gamma^n+(1-\theta)u_\Gamma^n\\&=(I-\theta S_2^{-1}(S_1+S_2))u_\Gamma^n.\end{aligned}$$

We next consider the N-N algorithm. From the definition of S_i and Step 1, Step 2 of Algorithm 2.2, we have

$$a_i(w_i^{n+1},T_i\gamma_0v)=\delta_i^\dagger(\langle S_1u_\Gamma^n,v\rangle+\langle S_2u_\Gamma^n,v\rangle).$$

Similar to (3.2), we have $w_i^{n+1}=\mathcal{H}_i(w_i^{n+1}|_\Gamma)$. Therefore, it holds that

$$w_i^{n+1}|_\Gamma=S_i^{-1}(\delta_i^\dagger(S_1+S_2)u_\Gamma^n). \tag{3.4}$$

By (3.4) and Step 3 of Algorithm 2.2, the error operator is obtained as follows:

$$\begin{aligned}u_\Gamma^{n+1}&=u_\Gamma^n-\theta(\delta_1^\dagger S_1^{-1}(\delta_1^\dagger(S_1+S_2)u_\Gamma^n)+\delta_2^\dagger S_2^{-1}(\delta_2^\dagger(S_1+S_2)u_\Gamma^n))\\&=\Big(I-\theta(D_1S_1^{-1}D_1+D_2S_2^{-1}D_2)(S_1+S_2)\Big)u_\Gamma^n,\end{aligned}$$

where $D_i=\delta_i^\dagger I$, $i=1,2$.

The error operator of D-D algorithm may be derived similar to the N-N algorithm. By the definitions of $\mathcal{H}_i,S_i$ and Steps 1,2 of Algorithm 2.3, we have

$$\begin{aligned}w_i^{n+1}&=\mathcal{H}_i(\delta_i^\dagger(u_1^{n+1}|_\Gamma-u_2^{n+1}|_\Gamma))\\&=\mathcal{H}_i(\delta_i^\dagger(S_1^{-1}\lambda_1^n-S_2^{-1}\lambda_2^n))\\&=\mathcal{H}_i(\delta_i^\dagger(S_1^{-1}+S_2^{-1})\lambda_\Gamma^n).\end{aligned}$$

Then by the interface update condition in Step 3, we get

$$\lambda_\Gamma^{n+1}=\Big(I-\theta(D_1S_1D_1+D_2S_2D_2)(S_1^{-1}+S_2^{-1})\Big)\lambda_\Gamma^n.$$

As to the R-R algorithm, for $i=1,2$, we have the following error equation,

$$a_i(u_i^n,T_i\gamma_0v)+\gamma_i\langle u_i^n|_\Gamma,v\rangle=\langle g_i^n,v\rangle \quad \forall v\in V_\Gamma.$$

By the definitions of $\mathcal{H}_i, S_i$, $i=1,2$, we have

$$g_i^n = (\gamma_i I + S_i) u_i^n|_\Gamma. \tag{3.5}$$

Using the interface update in Step 2 of Algorithm 2.4 and (3.5), we have

$$\begin{aligned} g_2^n &= (\gamma_1+\gamma_2)u_1^n|_\Gamma - g_1^n \\ &= (\gamma_1+\gamma_2)(\gamma_1 I+S_1)^{-1} g_1^n - (\gamma_1 I+S_1)(\gamma_1 I+S_1)^{-1} g_1^n \\ &= (\gamma_2 I - S_1)(\gamma_1 I+S_1)^{-1} g_1^n. \end{aligned} \tag{3.6}$$

Then by the second interface update in Step 4 of Algorithm 2.4, (3.6) and (3.5), it holds that

$$\begin{aligned} \tilde{g}_1^n &= (\gamma_1+\gamma_2)u_2^n - g_2^n \\ &= (\gamma_1+\gamma_2)(\gamma_2 I+S_2)^{-1} g_2^n - (\gamma_2 I+S_2)(\gamma_2 I+S_2)^{-1} g_2^n \\ &= (\gamma_1 I - S_2)(\gamma_2 I+S_2)^{-1} g_2^n \\ &= (\gamma_1 I - S_2)(\gamma_2 I+S_2)^{-1}(\gamma_2 I - S_1)(\gamma_1 I+S_1)^{-1} g_1^n. \end{aligned} \tag{3.7}$$

At last, by the relaxation step and (3.7), we obtain the error operator of R-R algorithm, i.e.

$$\begin{aligned} g_1^{n+1} &= \theta \tilde{g}_1^n + (1-\theta) g_1^n \\ &= \Big(I - \theta (I - (\gamma_1 I - S_2)(\gamma_2 I+S_2)^{-1}(\gamma_2 I - S_1)(\gamma_1 I+S_1)^{-1}) \Big) g_1^n \\ &= \Big(I - \theta (\gamma_1 I - S_2)(\gamma_2 I+S_2)^{-1}((\gamma_2 I+S_2)(\gamma_1 I - S_2)^{-1} - (\gamma_2 I - S_1)(\gamma_1 I+S_1)^{-1}) \Big) g_1^n. \end{aligned}$$

This completes the proof. □

The next task is to analyze how the spectral radius of R_i rely on the discontinuous coefficients ν_1, ν_2 and how to choose appropriate θ and $\delta_i^\dagger$ to accelerate the iterations.

3.1 Symmetric case

In this subsection, we suppose that Ω_1 and Ω_2 are symmetric with respect to Γ. Then we have

$$S_1/\nu_1 = S_2/\nu_2. \tag{3.8}$$

As a result, if (λ, v) is an eigenpair of S_1/ν_1, $\nu_1\lambda$ and $\nu_2\lambda$ will be the eigenvalues of S_1 and S_2 corresponding to eigenvector v, respectively. Then it is easy to check that

$$\begin{aligned} \lambda(R_1) &= 1 - \theta\Big(1 + \frac{\nu_1}{\nu_2}\Big), \\ \lambda(R_2) &= 1 - \theta\Big((\delta_1^\dagger)^2\Big(1+\frac{\nu_2}{\nu_1}\Big) + (\delta_2^\dagger)^2\Big(1+\frac{\nu_1}{\nu_2}\Big)\Big), \\ \lambda(R_3) &= 1 - \theta\Big((\delta_1^\dagger)^2\Big(1+\frac{\nu_1}{\nu_2}\Big) + (\delta_2^\dagger)^2\Big(1+\frac{\nu_2}{\nu_1}\Big)\Big), \\ \lambda(R_4) &= 1 - \theta\Big(1 - \frac{\gamma_1 - \nu_2\lambda}{\gamma_1+\nu_1\lambda}\cdot\frac{\gamma_2-\nu_1\lambda}{\gamma_2+\nu_2\lambda}\Big), \end{aligned}$$

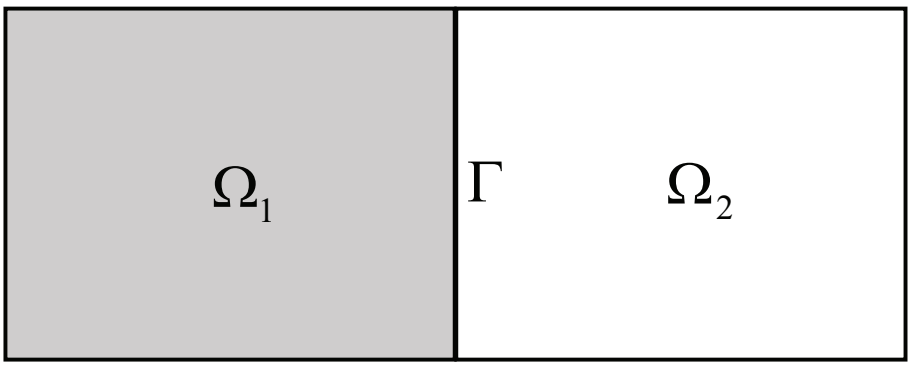

Figure 1: Ω is divided into two symmetric subdomains Ω_1,Ω_2 and their interface Γ.

where $\lambda(T)$ denotes the eigenvalue of operator T.

We may find that the eigenvalues of R_1,R_2,R_3 are independent of λ, thus we have

Theorem 3.1. *In the symmetric case, the convergence rate of Algorithms* 2.1, 2.2, 2.3 *can be reduced to* 0 *by choosing suitable* θ*, that is to say, the methods become direct solvers.*

Proof. Set

$$\theta_1=\frac{1}{1+\frac{\nu_1}{\nu_2}},$$
$$\theta_2=\frac{1}{(\delta_1^\dagger)^2\left(1+\frac{\nu_2}{\nu_1}\right)+(\delta_2^\dagger)^2\left(1+\frac{\nu_1}{\nu_2}\right)},$$
$$\theta_3=\frac{1}{(\delta_1^\dagger)^2\left(1+\frac{\nu_1}{\nu_2}\right)+(\delta_2^\dagger)^2\left(1+\frac{\nu_2}{\nu_1}\right)},$$

and we get the conclusion. □

Remark 3.1. For a general θ, the convergence behaviours will be quite different between D-N algorithm and N-N algorithm, D-D algorithm.

For a θ near 1, the convergence rate of D-N algorithm relies on ν_1/ν_2 and if $\nu_1 \ll \nu_2$, the iteration will perform quite well. In other word, the D-N algorithm benefit from the jump of discontinuous coefficients. But for N-N algorithm and D-D algorithm, we could not obtain such a good property. The range of function

$$f(\delta_1^\dagger)=(\delta_1^\dagger)^2(1+\nu_2/\nu_1)+(1-\delta_1^\dagger)^2(1+\nu_1/\nu_2)$$

is $[1,\max\{1+\nu_1/\nu_2,1+\nu_2/\nu_1\}]$. So N-N algorithm could not benefit from the discontinuous coefficients. The optimal choice of $\delta_1^\dagger,\delta_2^\dagger$ is

$$\delta_1^\dagger=\frac{\nu_1^\gamma}{\nu_1^\gamma+\nu_2^\gamma},\quad \delta_2^\dagger=\frac{\nu_2^\gamma}{\nu_1^\gamma+\nu_2^\gamma},\quad \gamma\in[1/2,\infty).$$

Here, we choose $\gamma=1/2$, then

$$f(\delta_1^\dagger)=\frac{2(\nu_1+\nu_2)}{(\sqrt{\nu_1}+\sqrt{\nu_2})^2}\in(1,2)$$

and the convergence rate of N-N algorithm will be independent of ν_1,ν_2. The case of D-D algorithm is similar.

For the Robin-Robin algorithm, we may find that the spectrum depends on the eigenvalue λ, which is different from other three algorithms. The following theorem can be obtained by analyzing the function $\lambda(R_4)$.

Theorem 3.2. *In the symmetric case, the convergence rate of Algorithm* 2.4 *is bounded by* $C\nu_1/\nu_2$ *by choosing* $\gamma_1\geq C_1\nu_2h^{-1}$, $0<\gamma_2\leq c_0\nu_1$ *and* $\theta=\frac{2}{2+\nu_1/\nu_2}$.

Proof. Let

$$\lambda(R_4)=1-\theta(1+\omega(\lambda)),$$

where

$$\omega(\lambda)=-\frac{\gamma_1-\nu_2\lambda}{\gamma_1+\nu_1\lambda}\cdot\frac{\gamma_2-\nu_1\lambda}{\gamma_2+\nu_2\lambda}.$$

The derivation of $\omega(\lambda)$ is

$$\omega'(\lambda)=\frac{(\gamma_1+\gamma_2)(\nu_1+\nu_2)(\gamma_1\gamma_2-\nu_1\nu_2\lambda^2)}{(\gamma_1+\nu_1\lambda)^2(\gamma_2+\nu_2\lambda)^2}. \tag{3.9}$$

It is known [1,12] that

$$\lambda\in\left[c_0,C_1h^{-1}\right],$$

then by choosing $0<\gamma_2\leq c_0\nu_1$, $\gamma_1\geq C_1\nu_2h^{-1}$ and (3.9), $\omega(\lambda)$ attains the maximum value at $\lambda_0=\sqrt{\frac{\gamma_1\gamma_2}{\nu_1\nu_2}}$ and we have

$$\omega(\lambda)\in\left(0,t^2\left(\frac{\eta-1/t}{\eta+t}\right)^2\right]\subset(0,t^2),$$

where $\eta=\sqrt{\frac{\gamma_1}{\gamma_2}}$, $t=\sqrt{\frac{\nu_1}{\nu_2}}$. Then the optimal choice of θ is obtained by

$$1-\theta+1-\theta\left(1+\frac{\nu_1}{\nu_2}\right)=0.$$

That is $\theta=\theta_0=\frac{2}{2+\nu_1/\nu_2}$, and the convergence rate is bounded by $1-\theta_0(1+\frac{\nu_1}{\nu_2})<\frac{\nu_1}{2\nu_2}$. □

We may find that the Robin-Robin algorithm benefits from the jump of discontinuous coefficients in the symmetric case.

3.2 Nonsymmetric case

In the nonsymmetric case, the equality (3.8) is no longer available. Instead, the next lemma is useful in the analysis.

Lemma 3.2. *There exists positive constants c_i,C_i, independent of h, such that for any $v\in W_\Gamma$,*

$$c_i|v|^2_{1,\Omega_i}\leq \|v\|^2_{H^{1/2}_{00}(\Gamma)}\leq C_i|v|^2_{1,\Omega_i}. \tag{3.10}$$

By (3.10), we may get the equivalence between S_1 and S_2 while they no longer have the same eigenvector. Therefore, none of the algorithms is a direct solver. Actually, the algorithms could be divided into two groups according to their convergence behaviours. The first group contains D-N algorithm and R-R algorithm. Both of them could benefit from the jump of discontinuous coefficients. To be detailed, we have the following two results.

Theorem 3.3. *In the discontinuous coefficients case, the convergence rate of D-N algorithm will be bounded by $\frac{\nu_1}{\nu_2}$ if $\nu_1\ll\nu_2$.*

Proof. We know

$$R_1=1-\theta S_2^{-1}S=1-\theta S_2^{-1}(S_1+S_2),$$

and we just need to find the spectrum of $S_2^{-1}S$.

By (3.10), for any $u_\Gamma\in W_\Gamma$,

$$\lambda(S_2^{-1}S)=\frac{\langle Su_\Gamma,u_\Gamma\rangle}{\langle S_2u_\Gamma,u_\Gamma\rangle}, \tag{3.11}$$

$$\langle S_2u_\Gamma,u_\Gamma\rangle\leq\langle Su_\Gamma,u_\Gamma\rangle\leq\left\langle\left(1+\frac{C_2}{c_1}\cdot\frac{\nu_1}{\nu_2}\right)S_2u_\Gamma,u_\Gamma\right\rangle. \tag{3.12}$$

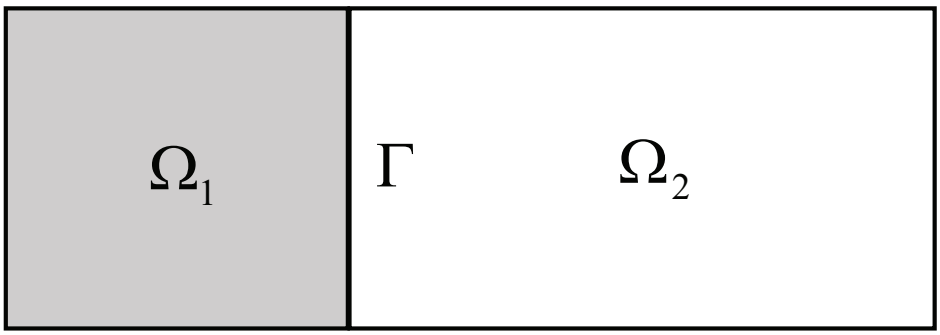

Figure 2: Ω is divided into two nonsymmetric subdomains Ω_1,Ω_2 and their interface Γ.

Combining (3.11), (3.12), we have

$$\lambda(S_2^{-1}S)\subset\Big[1,1+\frac{C_2}{c_1}\cdot\frac{\nu_1}{\nu_2}\Big],$$

and the optimal choice of θ is $\theta_0=\frac{2}{2+\frac{C_2}{c_1}\cdot\frac{\nu_1}{\nu_2}}$. Then the convergence rate is bounded by $\frac{\nu_1}{\nu_2}$. □

Theorem 3.4. *In the discontinuous coefficients case, if $\nu_1\ll\nu_2$, the convergence rate of the Robin-Robin algorithm will be bounded by $\frac{\nu_1}{\nu_2}$ with $\gamma_1\geq C_0\nu_2h^{-1}$, $0<\gamma_2\leq c_0\nu_1$.*

Proof. We have

$$R_4=I-\theta P_4^{-1}G,$$

where

$$\begin{aligned}&P_4^{-1}=(\gamma_1I-S_2)(\gamma_2I+S_2)^{-1},\\&G=(\gamma_2I+S_2)(\gamma_1I-S_2)^{-1}-(\gamma_2I-S_1)(\gamma_1I+S_1)^{-1}.\end{aligned}$$

For any $g\in W_\Gamma$, it holds that [5]

$$\frac{\gamma_2}{\gamma_1}\|g\|_{0,\Gamma}^2+\frac{\gamma_1+\gamma_2}{\gamma_1^2}|g|_{S_2}^2\leq\langle P_4g,g\rangle\leq\frac{\gamma_2}{\gamma_1}\|g\|_{0,\Gamma}^2+\frac{\gamma_1+\gamma_2}{\gamma_1^2}|g|_{S_2}^2,\tag{3.13}$$

$$\frac{\gamma_1+\gamma_2}{(2+\delta)\gamma_1^2}|g|_{S_1}^2+\frac{\gamma_1+\gamma_2}{\gamma_1^2}|g|_{S_2}^2\leq\langle Gg,g\rangle\leq\frac{\gamma_1+\gamma_2}{\gamma_1^2}|g|_{S_1}^2+\frac{2(\gamma_1+\gamma_2)}{\gamma_1^2}|g|_{S_2}^2,\tag{3.14}$$

where δ is an arbitrary positive constant independent of h, and $|\cdot|_{S_i}^2=\langle S_i\cdot,\cdot\rangle$, $i=1,2$. Additionally, we have

$$|g|_{S_i}^2\leq C\nu_ih^{-1}\|g\|_{0,\Gamma}^2.\tag{3.15}$$

By (3.13), (3.14) and (3.15), we have the upper bound estimate, i.e.

$$\begin{aligned}\langle Gg,g\rangle&\leq\frac{\gamma_1+\gamma_2}{\gamma_1^2}|g|_{S_1}^2+\frac{2(\gamma_1+\gamma_2)}{\gamma_1^2}|g|_{S_2}^2\\&\leq\frac{C'(\gamma_1+\gamma_2)}{\gamma_1^2}\Big(1+C\frac{\nu_1}{\nu_2}\Big)|g|_{S_2}^2\\&\leq(1+C\frac{\nu_1}{\nu_2})\langle P_4g,g\rangle\end{aligned}$$

with a suitable $\gamma_1\geq C\nu_2h^{-1}$.

By trace theorem and Poincaré inequality, we have

$$\|g\|_{0,\Gamma}^2\leq C|\mathcal{H}_ig|_{1,\Omega_i}^2\leq\frac{C}{\nu_i}|g|_{S_i}^2.\tag{3.16}$$

Then it follows (3.13), (3.14), (3.16) that

$$\begin{aligned}\langle P_4 g,g\rangle &\leq \frac{\gamma_2}{\gamma_1}\|g\|^2_{0,\Gamma}+\frac{\gamma_1+\gamma_2}{\gamma_1^2}|g|^2_{S_2}\\ &\leq \frac{c_0 C}{\gamma_1}|g|^2_{S_1}+\frac{\gamma_1+\gamma_2}{\gamma_1^2}|g|^2_{S_2}\\ &\leq \langle Gg,g\rangle\end{aligned}$$

with a suitable $\gamma_2 \leq c_0\nu_1$.

Therefore,

$$1\leq \lambda(P_4^{-1}G)\leq \Big(1+C\frac{\nu_1}{\nu_2}\Big). \tag{3.17}$$

Here θ is selected to be $\frac{2}{2+C\frac{\nu_1}{\nu_2}}$, and the convergence rate is bounded by $\frac{\nu_1}{\nu_2}$. □

Remark 3.2. In the two theorems above, we assume $\nu_1 \ll \nu_2$ and get convergence rates bounded by $\frac{\nu_1}{\nu_2}$. If $\nu_2 \ll \nu_1$, we could start the iterations of D-N algorithm and R-R algorithm by computing the problem with Dirichlet boundary condition and Robin boundary condition in Ω_2. Then the convergence rates will be bounded by $\frac{\nu_2}{\nu_1}$. While, for the case $\nu_1=\nu_2$, the convergence rates of D-N algorithm and R-R algorithms are bounded by a constant which is independent of h and less than 1 strictly.

Unlike the first two algorithms, in the case of two subdomains, the N-N algorithm and D-D algorithm could not take advantage of the discontinuous coefficients and their common feature is that the convergence rate may be independent of the jump of coefficients with suitable weights.

Theorem 3.5. *The convergence rate of the N-N algorithm and D-D algorithm may be independent of ν_1,ν_2 by choosing suitable $\delta_1^\dagger,\delta_2^\dagger$.*

Proof. We have

$$R_2 = I-\theta P_2^{-1}S = I-\theta(D_1S_1^{-1}D_1+D_2S_2^{-1}D_2)(S_1+S_2).$$

By (3.10), for any $u_\Gamma \in W_\Gamma$, it holds that

$$\lambda(P_2^{-1}S)=\frac{\langle Su_\Gamma,u_\Gamma\rangle}{\langle P_2u_\Gamma,u_\Gamma\rangle}, \tag{3.18}$$

and

$$\frac{c_1}{C_2}\cdot\frac{\nu_2}{\nu_1}\leq\frac{\langle S_2u_\Gamma,u_\Gamma\rangle}{\langle S_1u_\Gamma,u_\Gamma\rangle}\leq\frac{C_1}{c_2}\cdot\frac{\nu_2}{\nu_1}, \tag{3.19}$$

$$\frac{c_2}{C_1}\cdot\frac{\nu_1}{\nu_2}\leq\frac{\langle S_1u_\Gamma,u_\Gamma\rangle}{\langle S_2u_\Gamma,u_\Gamma\rangle}\leq\frac{C_2}{c_1}\cdot\frac{\nu_1}{\nu_2}. \tag{3.20}$$

Combining (3.18), (3.19), (3.20), we have

$$(\delta_1^\dagger)^2\Big(1+\frac{c_1}{C_2}\cdot\frac{\nu_2}{\nu_1}\Big)+(\delta_2^\dagger)^2\Big(1+\frac{c_2}{C_1}\cdot\frac{\nu_1}{\nu_2}\Big)\leq\lambda(P_2^{-1}S)$$
$$\leq(\delta_1^\dagger)^2\Big(1+\frac{C_1}{c_2}\cdot\frac{\nu_2}{\nu_1}\Big)+(\delta_2^\dagger)^2\Big(1+\frac{C_2}{c_1}\cdot\frac{\nu_1}{\nu_2}\Big). \quad (3.21)$$

We denote the left and right sides of inequality (3.21) by $\lambda_{\min}$ and $\lambda_{\max}$, then by choosing $\theta=\theta_0=\frac{2}{\lambda_{\min}+\lambda_{\max}}$, we have

$$\lambda(R_2)\subset\Big(-\frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}},\frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}\Big)=\Big(-\frac{\kappa-1}{\kappa+1},\frac{\kappa-1}{\kappa+1}\Big).$$

We now analyze how $\kappa=\kappa(\delta_1^\dagger)$ changes with $\delta_1^\dagger$, where

$$\kappa(\delta_1^\dagger)=\frac{\lambda_{\max}}{\lambda_{\min}}=\frac{(\delta_1^\dagger)^2(1+\frac{c_1}{C_2}\cdot\frac{\nu_2}{\nu_1})+(\delta_2^\dagger)^2(1+\frac{c_2}{C_1}\cdot\frac{\nu_1}{\nu_2})}{(\delta_1^\dagger)^2(1+\frac{C_1}{c_2}\cdot\frac{\nu_2}{\nu_1})+(\delta_2^\dagger)^2(1+\frac{C_2}{c_1}\cdot\frac{\nu_1}{\nu_2})}.$$

The derivation of $\kappa(\delta_1^\dagger)$ is

$$\kappa'(\delta_1^\dagger)=\frac{\Big(\frac{\nu_2}{\nu_1}(\frac{C_1}{c_2}-\frac{C_2}{c_1})-\frac{\nu_1}{\nu_2}(\frac{c_1}{C_2}-\frac{c_2}{C_1})\Big)\delta_1^\dagger(1-\delta_1^\dagger)}{\Big((\delta_1^\dagger)^2(1+\frac{c_1}{C_2}\cdot\frac{\nu_2}{\nu_1})+(1-\delta_1^\dagger)^2(1+\frac{c_2}{C_1}\cdot\frac{\nu_1}{\nu_2})\Big)^2},$$

then we know the minimum value of $\kappa(\delta_1^\dagger)$ is attained at $\delta_1^\dagger=0$ or $\delta_1^\dagger=1$ according to the symbol of the coefficient. However, in both the two cases, the N-N algorithm deteriorates to D-N algorithm. Although, we can choose a $\delta_1^\dagger$ near 1 or 0 and it seems that the convergence rate benefits from the discontinuous coefficients, we take it as an asymptotic convergence behaviour and we prefer D-N algorithm. In fact, we may choose

$$\delta_1^\dagger=\frac{\sqrt{\nu_1}}{\sqrt{\nu_1}+\sqrt{\nu_2}},\quad \delta_2^\dagger=\frac{\sqrt{\nu_2}}{\sqrt{\nu_1}+\sqrt{\nu_2}},$$

so that

$$\kappa\leq\frac{\max\{\frac{C_1}{c_2},\frac{C_2}{c_1}\}+1}{\min\{\frac{c_2}{C_1},\frac{c_1}{C_2}\}+1},$$

which is independent of ν_1 and ν_2.

The proof of D-D algorithm is similar to that of N-N algorithm by setting

$$\delta_1^\dagger=\frac{\sqrt{\nu_2}}{\sqrt{\nu_1}+\sqrt{\nu_2}},\quad \delta_2^\dagger=\frac{\sqrt{\nu_1}}{\sqrt{\nu_1}+\sqrt{\nu_2}}.$$

This completes the proof. □

Remark 3.3. From the analysis above, we find that the jump of discontinuous coefficients could accelerate the iteration when using the D-N algorithm and the R-R algorithm in the case of two subdomains as the ratio of the smaller coefficient to the larger one (ν_1/ν_2 when $\nu_1<\nu_2$) dominate their convergence rates. By contrast, the N-N algorithm and the D-D algorithm could not benefit from the ratio because the norms of Ω_1,Ω_2 need to be controlled by each other.

4 The case of many subdomains

In this section, we further consider how the discontinuous coefficients influence the convergence behaviours of these domain decomposition methods and wether the previous properties still hold in the case of many subdomains with red-black partition.

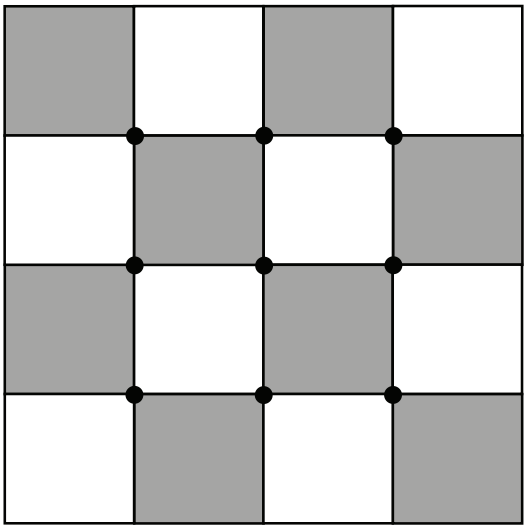

Figure 3: The red-black partition of Ω into 4×4 subdomains. The grey blocks denote the black domains and the white blocks denote the red domains. The black dots are cross points.

4.1 Preconditioned systems

The algorithms in the case of many subdomains in this paper rely on a red-black partition, so we first introduce the geometric settings. Partition the domain Ω into two classes of nonoverlapping subdomains Ω_R,Ω_B. The red domain is denoted by $\Omega_R=\bigcup\Omega_i$, $i\in\Lambda_R$ and the black domain is denoted by $\Omega_B=\bigcup\Omega_j$, $j\in\Lambda_B$, where Λ_R,Λ_B are the sets of subscripts of subdomains which belong to class Ω_R,Ω_B, respectively. The size of subdomains is H.

The intersection of subdomains in the same class is either empty or vertex. The interface is $\Gamma := \partial\Omega_R \cap \partial\Omega_B$. The vertexes of subdomains, which do not belong to $\partial\Omega$, are called the cross points. $\mathcal{T}_h$ is the triangulation same as the case of two subdomains. We assume that the subdomains boundaries do not cut through any element in $\mathcal{T}_h$.

Based on the geometric settings, some function spaces are defined. Let $W \in H_0^1(\Omega)$ be the P1 conforming finite element space. Then let $W_k, k \in \Lambda_R \cap \Lambda_B, W_R, W_B$ be the spaces which include the functions of W restricted to $\overline{\Omega}_k, \overline{\Omega}_R, \overline{\Omega}_B$. W_k^0, W_R^0, W_B^0 are the subspaces of W_k, W_R, W_B that functions of W_k^0, W_R^0, W_B^0 have vanishing traces on $\partial\Omega_k, \partial\Omega_R, \partial\Omega_B$, respectively. The space on the interface Γ is defined to be $V_\Gamma = W|_\Gamma$. We also denote $V_k := W|_{\partial\Omega_k}$. Besides, V_Δ contains functions in V_Γ who vanish at each node on cross points.

We then introduce some bilinear forms and operators in the case of many subdomains. Define the bilinear form on the subdomain Ω_k, $k \in \Lambda_R \cup \Lambda_B$ by

$$a_k(u_k, v_k) = \int_{\Omega_k} \nu_k \nabla u_k \cdot \nabla v_k \quad \forall u_k, v_k \in W_k.$$

Define local discrete harmonic extension operator $\mathcal{H}_k : V_k \to W_k$ as follows:

$$\begin{cases} a_k(\mathcal{H}_k u_\Gamma, v_k) = 0, & \forall v_k \in W_k^0, \\ \mathcal{H}_k u_\Gamma = u_\Gamma, & \text{on } \Gamma, \end{cases}$$

where $u_\Gamma \in V_k$. Here we also note that the constant coefficient ν_k does not affect the result $\mathcal{H}_k u_\Gamma$. The local Schur complement operator $S_k : V_k \to V_k$ is defined as follows:

$$\langle S_k u_k, v_k \rangle = a_k(\mathcal{H}_k u_k, T_k v_k) \quad \forall v_k \in V_k,$$

where T_k is an arbitrary extension operators. We may see that S_k is a symmetric and positive semi-definite operator, therefore it may induce a semi-norm of V_k, i.e. $|\cdot|_{S_k}^2 := \langle S_k \cdot, \cdot \rangle$ (if $\partial\Omega_k \cap \partial\Omega \neq \phi$, S_k will be positive definite and it induces a norm).

Based on the local bilinear forms and operators, we could define global operators. Define the bilinear form on Ω_R by

$$a_R(u_R, v_R) = \sum_{k \in \Lambda_R} \int_{\Omega_k} \nabla u_R \cdot \nabla v_R \quad \forall u_R, v_R \in W_R.$$

Define discrete harmonic extension operator $\mathcal{H}_R : V_\Gamma \to W_R$ as follows:

$$\begin{cases} a_R(\mathcal{H}_R u_\Gamma, v_R) = 0, & \forall v_R \in W_R^0, \\ \mathcal{H}_R u_\Gamma = u_\Gamma, & \text{on } \Gamma. \end{cases}$$

Then the Schur complement operator $S_R : V_\Gamma \to V_\Gamma$ could be defined, i.e.

$$\langle S_R u_\Gamma, v_\Gamma \rangle = a_R(\mathcal{H}_R u_\Gamma, T_R v_\Gamma) \quad \forall v_\Gamma \in V_\Gamma,$$

where $T_R : V_\Gamma \to W_R$ is an arbitrary extension operator. From the geometric settings, we may see that Ω_R consists of $\Omega_k, k \in \Lambda_R$ and they are connected by cross points. Therefore, S_R is a symmetric and positive definite operator and it induces a norm of V_Γ, i.e. $|\cdot|^2_{S_R} = \langle S_R \cdot, \cdot \rangle$. Similarly, we may define $a_B(\cdot,\cdot), \mathcal{H}_B$ and S_B. The Schur complement operator of the whole subdomains is defined as the sum of S_R and S_B, that is, $S = S_R + S_B$.

At last, we give the definitions of Schur complement operators of V_Δ. Define $\widetilde{S}: V_\Delta \to V_\Delta$ as follows:

$$\langle \widetilde{S} u_\Delta, u_\Delta \rangle = \min_{u \in V_\Gamma, u|_{\Gamma_\Delta} = u_\Delta} \langle Su, u \rangle,$$

where Γ_Δ denotes the degrees of freedom on Γ except cross points. From the minimization property and the fact that S is symmetric and positive definite, we know that $\widetilde{S}$ is also a symmetric and positive definite operator and it induces a norm of V_Δ. Similarly, $\widetilde{S}_R, \widetilde{S}_B$ could be defined and they hold the same properties as $\widetilde{S}$.

Now we are in a position to introduce algorithms in the case of many subdomains. We use the following Schur complement system in this paper:

$$\widetilde{S} u_\Gamma = \tilde{f}_\Delta.$$

The D-N algorithm and N-N algorithm could provide preconditioners for this system, which are $P^{-1}_{DN} = \widetilde{S}_B^{-1}$ and $P^{-1}_{NN} = D_R \widetilde{S}_R^{-1} D_R + D_B \widetilde{S}_B^{-1} D_B$, respectively. Here, D_R, D_B are scaling operators.

The system of flux is

$$F\lambda = d,$$

where $F = S_R^{-1} + S_B^{-1}$, $d = S_B^{-1} f_B - S_R^{-1} f_R$. For the flux system, the D-D algorithm provide a preconditioner $P^{-1}_{DD} = D_R S_R D_R + D_B S_B D_B$.

Similar to the case of two subdomains, the system of Robin boundary data g_R is

$$G g_R = ((\gamma_B I + S_B)(\gamma_R I - S_B)^{-1} - (\gamma_B I - S_R)(\gamma_R I + S_R)^{-1}) g_R = f^\star$$

and the R-R preconditioner is $P^{-1}_{RR} = (\gamma_R I - S_B)(\gamma_B I + S_B)^{-1}$.

The right hand sides $\tilde{f}_\Delta, d, f^\star$ will be illustrated in detail in the last subsection.

4.2 Condition number estimate

In this subsection, we analyze the condition numbers of these preconditioned systems. For simplicity, we consider the red-black checkerboard case, i.e.

$$\nu(\mathbf{x}) = \begin{cases} \nu_R, & \forall \mathbf{x} \in \Omega_R, \\ \nu_B, & \forall \mathbf{x} \in \Omega_B, \end{cases}$$

then the scaling operators D_R and D_B will be $\delta_R^\dagger I$ and $\delta_B^\dagger I$, respectively. We also assume that $\nu_R < \nu_B$.

As to D-N algorithms and R-R algorithm, we have following conclusions.

Theorem 4.1. *For the D-N algorithm, we have*

$$\langle \widetilde{S}_B u_\Delta, u_\Delta \rangle \leq \langle \widetilde{S} u_\Delta, u_\Delta \rangle \leq \left(1+C\frac{\nu_R}{\nu_B}\left(1+\log\frac{H}{h}\right)^2\right)\langle \widetilde{S}_B u_\Delta, u_\Delta \rangle \quad \forall u_\Delta \in V_\Delta. \tag{4.1}$$

Therefore, the condition number of D-N algorithm is bounded as follows:

$$\kappa(P_{DN}^{-1}\widetilde{S}) \leq 1+C\frac{\nu_R}{\nu_B}\left(1+\log\frac{H}{h}\right)^2. \tag{4.2}$$

Proof. Suppose

$$\langle \widetilde{S} u_\Delta, u_\Delta \rangle = \min_{u\in V_\Gamma, u|_{\Gamma_\Delta}=u_\Delta} \langle Su,u\rangle = \langle Su_\Gamma, u_\Gamma\rangle = \langle S_R u_\Gamma, u_\Gamma\rangle + \langle S_B u_\Gamma, u_\Gamma\rangle,$$

then the lower bound may be obtained by the definition of $\widetilde{S}_B$, i.e.

$$\langle \widetilde{S}_B u_\Delta, u_\Delta \rangle = \min_{u\in V_\Gamma, u|_{\Gamma_\Delta}=u_\Delta} \langle S_B u,u\rangle \leq \langle S_B u_\Gamma, u_\Gamma\rangle \leq \langle \widetilde{S} u_\Delta, u_\Delta \rangle.$$

On the other hand, suppose

$$\langle \widetilde{S}_B u_\Delta, u_\Delta \rangle = \min_{u\in V_\Gamma, u|_{\Gamma_\Delta}=u_\Delta} \langle S_B u,u\rangle = \langle S_B \tilde{u}_\Gamma, \tilde{u}_\Gamma\rangle,$$

then

$$\langle \widetilde{S} u_\Delta, u_\Delta \rangle \leq \langle S\tilde{u}_\Gamma, \tilde{u}_\Gamma\rangle = \langle S_R \tilde{u}_\Gamma, \tilde{u}_\Gamma\rangle + \langle S_B \tilde{u}_\Gamma, \tilde{u}_\Gamma\rangle. \tag{4.3}$$

Therefore, to prove the upper bound, we need to control $\langle S_R\tilde{u}_\Gamma,\tilde{u}_\Gamma\rangle$ by $\langle S_B\tilde{u}_\Gamma,\tilde{u}_\Gamma\rangle$. Let $\Pi_H: V_\Gamma \to V_\Gamma$ be the linear interpolation operator on the coarse grid, where $\Pi_H v(x)=v(x)$ for any cross point x. Then by Lemma 3.1 and Lemma 3.3 in **[?]**, we have

$$\begin{aligned}
\langle S_R\tilde{u}_\Gamma,\tilde{u}_\Gamma\rangle &= \sum_{i\in\Lambda_R} \nu_i |\mathcal{H}_i\tilde{u}_\Gamma|^2_{1,\Omega_i} \\
&\leq \sum_{i\in\Lambda_R} \nu_i \left(|\mathcal{H}_i(\tilde{u}_\Gamma-\Pi_H\tilde{u}_\Gamma)|^2_{1,\Omega_i}+|\mathcal{H}_i\Pi_H\tilde{u}_\Gamma|^2_{1,\Omega_i}\right) \\
&\leq C\nu_R \sum_{i\in\Lambda_R}\sum_{j\in\Lambda_i}\left(\|(\tilde{u}_\Gamma-\Pi_H\tilde{u}_\Gamma)\|^2_{H_{00}^{1/2}(\Gamma_{ij})}+|\tilde{u}_\Gamma(x_{ij0})-\tilde{u}_\Gamma(x_{ij1})|^2\right) \\
&\leq C\nu_R \sum_{j\in\Lambda_B}\sum_{i\in\Lambda_j}\left(\|(\tilde{u}_\Gamma-\Pi_H\tilde{u}_\Gamma)\|^2_{H_{00}^{1/2}(\Gamma_{ji})}+|\tilde{u}_\Gamma(x_{ji0})-\tilde{u}_\Gamma(x_{ji1})|^2\right) \\
&\leq C\nu_R \sum_{j\in\Lambda_B} \nu_j^{-1}\left(1+\log\frac{H}{h}\right)^2\langle S_j\tilde{u}_\Gamma,\tilde{u}_\Gamma\rangle \\
&\leq C\frac{\nu_R}{\nu_B}\left(1+\log\frac{H}{h}\right)^2\langle S_B\tilde{u}_\Gamma,\tilde{u}_\Gamma\rangle.
\end{aligned} \tag{4.4}$$

Combining (4.3) and (4.4), we get the upper bound. The estimate of condition number is as follows:

$$\kappa(P_{DN}^{-1}\widetilde{S})=\frac{\lambda_{\max}(P_{DN}^{-1}\widetilde{S})}{\lambda_{\min}(P_{DN}^{-1}\widetilde{S})}\leq 1+C\frac{\nu_R}{\nu_B}\left(1+\log\frac{H}{h}\right)^2.$$

This completes the proof. □

Theorem 4.2. *For the R-R algorithm, we assume $\gamma_R\geq C\nu_B h^{-1}$ and $0<\gamma_B\leq c\nu_R H$, then it holds that*

$$c\langle P_{RR}g,g\rangle\leq\langle Gg,g\rangle\leq C\left(1+\frac{\nu_R}{\nu_B}(1+\log\frac{H}{h})^2\right)\langle P_{RR}g,g\rangle. \tag{4.5}$$

Thus,

$$\kappa(P_{RR}^{-1}G)\leq C\left(1+\frac{\nu_R}{\nu_B}(1+\log\frac{H}{h})^2\right).$$

Proof. For the preconditioned system, we have the following estimates,

$$\frac{\gamma_B}{\gamma_R}\|g\|_{0,\Gamma}^2+\frac{\gamma_R+\gamma_B}{\gamma_R^2}|g|_{S_B}^2\leq\langle P_{RR}g,g\rangle\leq\frac{\gamma_B}{\gamma_R}\|g\|_{0,\Gamma}^2+\frac{2(\gamma_R+\gamma_B)}{\gamma_R^2}|g|_{S_B}^2, \tag{4.6}$$

and

$$\frac{\gamma_R+\gamma_B}{(2+\delta)\gamma_R^2}|g|_{S_R}^2+\frac{\gamma_R+\gamma_B}{\gamma_R^2}|g|_{S_B}^2\leq\langle Gg,g\rangle\leq\frac{\gamma_R+\gamma_B}{\gamma_R^2}|g|_{S_R}^2+\frac{2(\gamma_R+\gamma_B)}{\gamma_R^2}|g|_{S_B}^2, \tag{4.7}$$

where $\delta>0$ is an constant independent of h,H. For the details, we refer to [5].

By (4.6) and (4.7), we may get the upper bound estimate, i.e.

$$\begin{aligned}\langle Gg,g\rangle&\leq\frac{\gamma_R+\gamma_B}{\gamma_R^2}|g|_{S_R}^2+\frac{2(\gamma_R+\gamma_B)}{\gamma_R^2}|g|_{S_B}^2\\&\leq\frac{\gamma_R+\gamma_B}{\gamma_R^2}\left(2+C\frac{\nu_R}{\nu_B}(1+\log\frac{H}{h})^2\right)|g|_{S_B}^2 \qquad (4.8)\\&\leq C\left(1+\frac{\nu_R}{\nu_B}(1+\log\frac{H}{h})^2\right)\langle P_{RR}g,g\rangle. \qquad (4.9)\end{aligned}$$

The L^2 norm of g on the interface may be estimated by trace theorem, i.e.

$$\begin{aligned}\|g\|_{0,\partial\Omega_i}^2&\leq CH|\mathcal{H}_i g|_{1,\Omega_i}^2+CH^{-1}\|\mathcal{H}_i g\|_{0,\Omega_i}^2,\quad i\in\Lambda_R,\\\|g\|_{0,\partial\Omega_j}^2&\leq CH|\mathcal{H}_j g|_{1,\Omega_j}^2+CH^{-1}\|\mathcal{H}_j g\|_{0,\Omega_j}^2,\quad j\in\Lambda_B.\end{aligned}$$

Summing over all the subdomains, we get

$$\begin{aligned}\|g\|_{0,\Gamma}^2&\leq CH(|\mathcal{H}_R g|_{1,\Omega_R}^2+|\mathcal{H}_B g|_{1,\Omega_B}^2)+CH^{-1}(\|\mathcal{H}_R g\|_{0,\Omega_R}^2+\|\mathcal{H}_B g\|_{0,\Omega_B}^2)\\&\leq CH^{-1}(|\mathcal{H}_R g|_{1,\Omega_R}^2+|\mathcal{H}_B g|_{1,\Omega_B}^2),\end{aligned} \tag{4.10}$$

by using Poincaré inequality and the fact that $H \leq H^{-1}$. Then by the choice of γ_R, γ_B and assumption $\nu_R < \nu_B$, the lower bound could be obtained as follows,

$$\begin{aligned}
\langle P_{RR}g, g\rangle &\leq \frac{\gamma_B}{\gamma_R}\|g\|^2_{0,\Delta} + \frac{2(\gamma_R+\gamma_B)}{\gamma_R^2}|g|^2_{S_B} \\
&\leq C\frac{\nu_R}{\gamma_R}(|\mathcal{H}_R g|^2_{1,\Omega_R} + |\mathcal{H}_B g|^2_{1,\Omega_B}) + \frac{2(\gamma_R+\gamma_B)}{\gamma_R^2}|g|^2_{S_B} \\
&\leq C\frac{1}{\gamma_R}|g|^2_{S_R} + \Big(C\frac{\nu_R}{\nu_B\gamma_R} + \frac{2(\gamma_R+\gamma_B)}{\gamma_R^2}\Big)|g|^2_{S_B} \\
&\leq \max\left\{C\frac{(2+\delta)\gamma_R}{\gamma_R+\gamma_B}, 2+\frac{\nu_R}{\nu_B}\cdot\frac{\gamma_R}{\gamma_R+\gamma_B}\right\}\langle Gg, g\rangle \\
&\leq C\langle Gg, g\rangle. \qquad (4.11)
\end{aligned}$$

The condition number is then bounded by using (4.8) and (4.11). □

Remark 4.1. We may find that D-N algorithm and R-R algorithm could still benefit from the jumps of the discontinuous coefficients. If $\nu_R \ll \nu_B$, the condition numbers will be bounded by a nearly constant, that is,

$$\kappa(P_{DN}^{-1}\widetilde{S}) \leq 1+\mathcal{O}(\epsilon),$$

and

$$\kappa(P_{RR}^{-1}G) \leq C(1+\mathcal{O}(\epsilon)),$$

where $\epsilon = \frac{\nu_R}{\nu_B}$.

Remark 4.2. For the case that coefficients are piecewise constants in red and black domains, the condition number bounds will become

$$\begin{aligned}
\kappa(P_{DN}^{-1}\widetilde{S}) &\leq 1+C\tilde{\epsilon}\Big(1+\log\frac{H}{h}\Big)^2, \\
\kappa(P_{RR}^{-1}G) &\leq C\Big(1+\tilde{\epsilon}\Big(1+\log\frac{H}{h}\Big)^2\Big),
\end{aligned}$$

where

$$\tilde{\epsilon} = \min\Big\{\frac{\nu_R^{\max}}{\nu_B^{\min}}, \frac{\nu_B^{\max}}{\nu_R^{\min}}\Big\}.$$

Theorem 4.3. *For the N-N algorithms, we have*

$$\kappa(P_{NN}^{-1}\widetilde{S}) \leq C\Big(1+\log\frac{H}{h}\Big)^2 \qquad (4.12)$$

by choosing suitable D_R and D_B.

Proof. According to the definitions, we have

$$\begin{aligned}\langle \widetilde{S}u_\Delta,u_\Delta\rangle &= \min_{u\in V_\Gamma, u|_{\Gamma_\Delta}=u_\Delta} \langle Su,u\rangle = \langle Su_\Gamma,u_\Gamma\rangle = \langle S_R u_\Gamma,u_\Gamma\rangle + \langle S_B u_\Gamma,u_\Gamma\rangle \\ &\geq \langle \widetilde{S}_R u_\Delta,u_\Delta\rangle + \langle \widetilde{S}_B u_\Delta,u_\Delta\rangle.\end{aligned}$$

Then we estimate the lower bound of eigenvalues of $P_{NN}^{-1}(\widetilde{S}_R+\widetilde{S}_B)$ instead of $P_{NN}^{-1}\widetilde{S}$. Since $\widetilde{S}_R,\widetilde{S}_B$ are both positive definite operators, it holds that

$$\begin{aligned}&\langle P_{NN2}^{-1}(\widetilde{S}_R+\widetilde{S}_B)u_\Delta,u_\Delta\rangle \\ =&\left((\delta_R^\dagger)^2+(\delta_B^\dagger)^2\right)\langle u_\Delta,u_\Delta\rangle + (\delta_R^\dagger)^2\langle \widetilde{S}_R^{-1}\widetilde{S}_B u_\Delta,u_\Delta\rangle + (\delta_B^\dagger)^2\langle \widetilde{S}_B^{-1}\widetilde{S}_R u_\Delta,u_\Delta\rangle \\ \geq&\left((\delta_R^\dagger)^2+(\delta_B^\dagger)^2\right)\langle u_\Delta,u_\Delta\rangle,\end{aligned}$$

therefore,

$$\left((\delta_R^\dagger)^2+(\delta_B^\dagger)^2\right)\langle u_\Delta,u_\Delta\rangle \leq \langle P_{NN}^{-1}\widetilde{S}u_\Delta,u_\Delta\rangle. \tag{4.13}$$

As $\delta_R^\dagger+\delta_B^\dagger=1$, it holds that $(\delta_R^\dagger)^2+(\delta_B^\dagger)^2\geq\frac{1}{2}$. The lower bound is obtained.

We then estimate the upper bound. By using (4.4), we have

$$\langle \widetilde{S}u_\Delta,u_\Delta\rangle \leq \left(1+C\frac{\nu_R}{\nu_B}\left(1+\log\frac{H}{h}\right)^2\right)\langle \widetilde{S}_B u_\Delta,u_\Delta\rangle, \tag{4.14}$$

$$\langle \widetilde{S}u_\Delta,u_\Delta\rangle \leq \left(1+C\frac{\nu_B}{\nu_R}\left(1+\log\frac{H}{h}\right)^2\right)\langle \widetilde{S}_R u_\Delta,u_\Delta\rangle. \tag{4.15}$$

Combining (4.14), (4.15), we get

$$\lambda(P_{NN}^{-1}\widetilde{S}) \leq (\delta_R^\dagger)^2+(\delta_B^\dagger)^2+C\left((\delta_R^\dagger)^2\frac{\nu_B}{\nu_R}+(\delta_B^\dagger)^2\frac{\nu_R}{\nu_B}\right)\left(1+\log\frac{H}{h}\right)^2. \tag{4.16}$$

Similar to the case of two subdomains, if we set $\delta_R^\dagger=1$ or 0, the method becomes D-N algorithm actually. This is not a good choice. One of the optimal choices could be

$$\delta_R^\dagger=\frac{\sqrt{\nu_R}}{\sqrt{\nu_R}+\sqrt{\nu_B}},\quad \delta_B^\dagger=\frac{\sqrt{\nu_B}}{\sqrt{\nu_R}+\sqrt{\nu_B}},$$

then the upper bound, independent of ν_R,ν_B is obtained as follows,

$$\lambda(P_{NN2}^{-1}\widetilde{S}) \leq C\frac{\nu_R+\nu_B}{(\sqrt{\nu_R}+\sqrt{\nu_B})^2}\left(1+\log\frac{H}{h}\right)^2 \leq C\left(1+\log\frac{H}{h}\right)^2. \tag{4.17}$$

Combining (4.13) and (4.17), we get the conclusion (4.12). □

The conclusion and the proof of D-D algorithm is similar to that of N-N algorithms.

We have analyzed four kinds of domain decompositions in the case of many subdomains with red-black partition. Similar to the case of two subdomains, we find that they have two kinds of different behaviours. Why there is such a difference? Intuitively, the D-N algorithm and R-R algorithm use information of half the subdomains to precondition the whole system while energy norms of Ω_R and Ω_B are controlled by each other in the N-N algorithm and D-D algorithm. Therefore, in these special cases, D-N algorithm and R-R algorithm may perform very well as they fully take advantage of the ratio $\frac{\nu_R}{\nu_B}$ but N-N algorithm and D-D algorithm do not have such a good property. In fact, N-N algorithm and D-D algorithm are more applicable in general cases by choosing appropriate weights and their condition number bounds may be independent of the discontinuous coefficients. For the details, we refer to [4,6] and references therein.

4.3 Implementation of the algorithms

In the subsection, we will describe the implementation of the preconditioned systems and right hand sides.

We first illustrate the implementation of $\widetilde{S}$,$\widetilde{S}_R$ and $\widetilde{S}_B$. Reorder the vectors of unknowns into the following form:

$$u_R^T=\begin{pmatrix}u_I^{R\,T} & u_\Delta^T & u_C^T\end{pmatrix},\quad u_B^T=\begin{pmatrix}u_I^{B\,T} & u_\Delta^T & u_C^T\end{pmatrix},\quad u^T=\begin{pmatrix}u_I^T & u_\Delta^T & u_C^T\end{pmatrix},$$

then we have

$$\begin{pmatrix}A_{II} & A_{I\Delta} & A_{IC}\\ A_{\Delta I} & A_{\Delta\Delta} & A_{\Delta C}\\ A_{CI} & A_{C\Delta} & A_{CC}\end{pmatrix}\begin{pmatrix}u_I\\ u_\Delta\\ u_C\end{pmatrix}=\begin{pmatrix}f_I\\ f_\Delta\\ f_C\end{pmatrix}.$$

The systems of u_R,u_B are similar. The Schur complement system on Γ_Δ is as follows:

$$\widetilde{S}u_\Delta=\tilde{f}_\Delta,$$

where

$$\widetilde{S}=A_{\Delta\Delta}-\begin{pmatrix}A_{\Delta I} & A_{\Delta C}\end{pmatrix}\begin{pmatrix}A_{II} & A_{IC}\\ A_{CI} & A_{CC}\end{pmatrix}^{-1}\begin{pmatrix}A_{I\Delta}\\ A_{C\Delta}\end{pmatrix},$$

and

$$\tilde{f}_\Delta=f_\Delta-\begin{pmatrix}A_{\Delta I} & A_{\Delta C}\end{pmatrix}\begin{pmatrix}A_{II} & A_{IC}\\ A_{CI} & A_{CC}\end{pmatrix}^{-1}\begin{pmatrix}f_I\\ f_C\end{pmatrix}.$$

Similarly, we may get $\widetilde{S}_R$,$\widetilde{S}_B$. In the following, we should know how $\widetilde{S}$ and $\widetilde{S}_R^{-1}$,$\widetilde{S}_B^{-1}$ act on a given vector u_Δ. To determine $\widetilde{S}u_\Delta$, we first solve the following coarse problem

$$S_{CC}u_C=\hat{f}_C,$$

where

$$S_{CC}=A_{CC}-A_{CI}A_{II}^{-1}A_{IC},$$

and

$$\tilde{f}_C = A_{C\Delta}u_\Delta - A_{CI}A_{II}^{-1}A_{I\Delta}u_\Delta.$$

Then we need to solve subdomain Dirichlet problems with boundary data given by u_Δ and u_C, and finally obtain $\widetilde{S}u_\Delta$. $\widetilde{S}_R u_\Delta$,$\widetilde{S}_B u_\Delta$ could be obtained in the same way. To compute $\widetilde{S}_R^{-1}u_\Delta$, we need to solve the following problem,

$$\begin{pmatrix} A_{II}^R & A_{I\Delta}^R & A_{IC}^R \\ A_{\Delta I}^R & A_{\Delta\Delta}^R & A_{\Delta C}^R \\ A_{CI}^R & A_{C\Delta}^R & A_{CC}^R \end{pmatrix} \begin{pmatrix} w_I \\ w_\Delta \\ w_C \end{pmatrix} = \begin{pmatrix} 0 \\ u_\Delta \\ 0 \end{pmatrix}. \tag{4.18}$$

By eliminating the unknowns of type I and Δ, we get the coarse problem

$$\widetilde{S}_{CC}^R w_C = \tilde{f}_C^R,$$

where

$$\widetilde{S}_{CC}^R = A_{CC}^R - \begin{pmatrix} A_{CI}^R & A_{C\Delta}^R \end{pmatrix} \begin{pmatrix} A_{II}^R & A_{I\Delta}^R \\ A_{\Delta I}^R & A_{\Delta\Delta}^R \end{pmatrix}^{-1} \begin{pmatrix} A_{IC}^R \\ A_{\Delta C}^R \end{pmatrix},$$

and

$$\tilde{f}_C^R = -\begin{pmatrix} A_{CI}^R & A_{C\Delta}^R \end{pmatrix} \begin{pmatrix} A_{II}^R & A_{I\Delta}^R \\ A_{\Delta I}^R & A_{\Delta\Delta}^R \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ u_\Delta \end{pmatrix}.$$

After solving w_C, we may substitute it into (4.18) and solve local problems, then w_Δ is the desired vector $\widetilde{S}_R^{-1}u_\Delta$. The implementation of $\widetilde{S}_B^{-1}$ is similar.

Then we illustrate the implementation of S_R and S_B. We take S_R as an example. Reorder the vectors u_R into the following form:

$$u_R^T = \left({u_I^R}^T u_\Gamma^T \right).$$

Then we have

$$\begin{pmatrix} A_{II}^R & A_{I\Gamma}^R \\ A_{\Gamma I}^R & A_{\Gamma\Gamma}^R \end{pmatrix} \begin{pmatrix} u_I^R \\ u_\Gamma \end{pmatrix} = \begin{pmatrix} f_I^R \\ f_\Gamma^R \end{pmatrix}, \tag{4.19}$$

By eliminating the interior component, we have

$$S_R u_\Gamma = f_R,$$

where

$$S_R = A_{\Gamma\Gamma}^R - A_{\Gamma I}^R {A_{II}^R}^{-1} A_{I\Gamma}^R,$$

and

$$f_R = f_\Gamma^R - A_{\Gamma I}^R {A_{II}^R}^{-1} f_I^R.$$

S_B,f_B are obtained in the same way.

The implementation of S_R and S_B acting on a given vector is realized by Gaussian block elimination. And the implementation of S_R^{-1},S_B^{-1},$(\gamma_R M + S_R)^{-1}$,$(\gamma_B M +$

$S_B)^{-1}$,$(\gamma_R M-S_B)^{-1}$ acting on a vector may follow the same way of $\widetilde{S}_R^{-1}$ with the right hand side in the following form,

$$g^T=\begin{pmatrix}0 & g_\Delta^T & g_C^T\end{pmatrix}^T,$$

as they act on vectors of V_Γ.

Now the only thing left is the right hand side of R-R system. In fact, the R-R algorithm should be treated carefully and the system after simplification is

$$M\Big((\gamma_R M-S_B)^{-1}-(\gamma_R M+S_R)^{-1})\Big)Mg_R=M(\gamma_R M-S_B)^{-1}f_B+M(\gamma_R M+S_R)^{-1}f_R,$$

and $P_{RR}^{-1}=(\gamma_R+\gamma_B)(\gamma_B M+S_B)^{-1}-M^{-1}$, where M is the mass matrix of V_Γ.

So far, the implementation of the algorithms is completed.

5 Numerical experiments

In this section, we perform some numerical experiments to verify our conclusions. We consider the following diffusion problem with zero Dirichlet boundary condition,

$$-\nabla\cdot(\nu(\mathbf{x})\nabla u)=f \quad \text{in } \Omega, \tag{5.1}$$

where $\Omega=(0,1)^2$ and $f=-2(x^2+y^2-x-y)$.

We first test the case of two subdomains with Ω_1,Ω_2 symmetric with respect to $\Gamma=\{\frac{1}{2}\}\times(0,1)$, i.e. $\Omega_1=(0,\frac{1}{2})\times(0,1)$, $\Omega_2=(\frac{1}{2},1)\times(0,1)$. The iteration stops when the relative error is less than $tol=10^{-8}$. The weights of N-N algorithm and D-D algorithms are set to be optimal and the Robin parameters of R-R algorithm are $\gamma_1=\nu_2/h$ and $\gamma_2=\nu_1$.

Table 1 and Table 2 show the numbers of iterations of different algorithms with several groups of parameters. θ_{opt} denotes the optimal θ mentioned above. We may find that

Table 1: Numbers of iterations of D-N algorithm and N-N algorithm with different parameters.

ν_1	ν_2	h	D-N			N-N		
			θ_{opt}	1/2	1	θ_{opt}	1/3	2/3
10^{-2}	10^{2}	1/16	1	27	3	1	18	16
10^{-4}	10^{4}	1/16	1	27	1	1	17	17
10^{-6}	10^{6}	1/16	1	27	1	1	17	17
10^{-2}	10^{2}	1/32	1	27	3	1	18	16
10^{-4}	10^{4}	1/32	1	27	1	1	17	17
10^{-6}	10^{6}	1/32	1	27	1	1	17	17
10^{-2}	10^{2}	1/64	1	27	3	1	18	16
10^{-4}	10^{4}	1/64	1	27	2	1	17	17
10^{-6}	10^{6}	1/64	1	27	1	1	17	17

Table 2: The numbers of iterations of D-D algorithm and R-R algorithm with different parameters.

ν_1	ν_2	h	D-D			R-R		
			θ_{opt}	1/3	2/3	θ_{opt}	1/2	1
10^{-2}	10^{2}	1/16	1	18	16	2	27	2
10^{-4}	10^{4}	1/16	1	17	17	1	27	1
10^{-6}	10^{6}	1/16	1	17	17	1	27	1
10^{-2}	10^{2}	1/32	1	18	16	2	27	2
10^{-4}	10^{4}	1/32	1	17	17	1	27	1
10^{-6}	10^{6}	1/32	1	17	17	1	27	1
10^{-2}	10^{2}	1/64	1	18	16	2	27	2
10^{-4}	10^{4}	1/64	1	17	17	1	27	1
10^{-6}	10^{6}	1/64	1	17	17	1	27	1

Table 3: The numbers of iterations with different discontinuous coefficients.

ν_1	ν_2	D-N		N-N		D-D		R-R	
		Γ_1	Γ_2	Γ_1	Γ_2	Γ_1	Γ_2	Γ_1	Γ_2
10^{-1}	10^{1}	4	4	11	14	11	14	5	5
10^{-2}	10^{2}	2	2	11	14	11	14	2	2
10^{-3}	10^{3}	2	2	11	14	11	14	2	2
10^{-4}	10^{4}	1	1	11	14	11	14	1	1
10^{-5}	10^{5}	1	1	11	14	11	14	1	1
10^{-6}	10^{6}	1	1	11	14	11	14	1	1

if we choose the optimal θ, D-N algorithm, N-N algorithm and D-D algorithm converge in one step which corresponds to the zero convergence rate in Theorem 3.1. For D-N algorithm and R-R algorithm, if θ is set to be 1, the convergence rate will thoroughly rely on the ratio ν_1/ν_2 and the data in Table 1 and Table 2 confirm the conclusion because the iteration counts decrease with the jump in the coefficient increases. For N-N algorithm and D-D algorithm, we find that the iterations are not affected by the discontinuous coefficients ν_1,ν_2 which supports the theoretical results. Besides, we note that all of them are independent of mesh size h.

In the second experiment, we test the case of two subdomains with Ω_1,Ω_2 nonsymmetric. The first interface Γ_1 is set to be $\{\frac{1}{4}\}\times(0,1)$ and the second case is $\Gamma_2=\{\frac{3}{4}\}\times(0,1)$. The mesh size h is fixed to be 1/64. The relaxation parameter is always the optimal one of symmetric case. The other settings are the same as the previous experiment.

In Table 3, the convergence of the different methods is shown for different coefficient ratios in nonsymmetric case. The D-N algorithm and R-R algorithm with optimal θ converge very quickly. This is because their convergence rates are dominated by the coefficient ratio and the smaller the ratio is, the faster the iteration converges. For the N-N algorithm and D-D algorithm with optimal weights, the convergence rates are in-

Table 4: The numbers of iteration for 8×8 subdomains with $\nu_R=\nu_B=1$.

$\frac{H}{h}$	D-N	N-N	D-D	R-R
4	15	8	7	15
8	17	10	8	17
16	19	11	9	19
32	21	13	10	21
64	23	14	11	23

Table 5: The numbers of iteration for case of many subdomains with $\nu_R=\nu_B=1$ and fixed $\frac{H}{h}=8$.

$N\times N$	D-N	N-N	D-D	R-R
4×4	9	5	4	10
8×8	17	10	8	17
16×16	20	10	8	20
24×24	20	10	8	20
32×32	20	10	7	20

Table 6: The numbers of iteration for 8×8 subdomains with different discontinuous coefficients.

ν_B	ν_R	D-N	N-N	D-D	R-R
10^1	10^{-1}	4	17	14	4
10^2	10^{-2}	2	17	14	2
10^3	10^{-3}	2	17	14	2
10^4	10^{-4}	1	17	14	1
10^5	10^{-5}	1	17	14	1
10^6	10^{-6}	1	17	14	1

dependent of the discontinuous coefficients. In addition, by comparing the numbers of iteration with Γ_1,Γ_2, we may see that the influence of using different interfaces is little.

At last, we test the case of many subdomains with coefficients red-black checkerboard distribution. We use the PCG method and the terminal precision is chosen as 10^{-6}. The weights of N-N algorithm and D-D algorithm are optimal and the Robin parameters γ_R,γ_B are set to be $16\nu_B/h$ and $\nu_R H/2$, respectively.

Table 4 shows the corresponding iteration numbers of the four algorithms when the mesh is refined. The slight increases of the iteration numbers explain that the condition number is growing slowly as $\frac{H}{h}$ increases. Then we fix the degrees of freedom in each subdomain, that is, $\frac{H}{h}$ is a constant, we get the results in Table 5. We may see that the iteration number of each algorithm is stable which reflects that the iteration numbers rely only on $\frac{H}{h}$.

The results in Table 6 corresponds to the numerical experiment with fixed numbers of subdomains, fixed $\frac{H}{h}=8$ and increasing jumps of the coefficients. We may see that the

iteration numbers of D-N algorithms and R-R algorithm decrease rapidly as the jumps increase which confirms our theory and the iteration numbers of N-N algorithms and D-D algorithm are stable.

Acknowledgments

The authors would like to thank the referees for the helpful suggestions. This work was supported by the National Natural Science Foundation of China (grants 12071350 and 12331015).

References

[1] Wenbin Chen, Xuejun Xu, and Shangyou Zhang, On the optimal convergence rate of a Robin-Robin domain decomposition method. J. Comput. Math., 32 (2014), no. 4, 456–475.
[2] Martin J. Gander and Olivier Dubois, Optimized Schwarz methods for a diffusion problem with discontinuous coefficient. Numer. Algorithms, 69 (2015), no. 1, 109–144.
[3] Luca Gerardo-Giorda, Patrick Le Tallec, and Frédéric Nataf, A Robin-Robin preconditioner for advection-diffusion equations with discontinuous coefficients. Comput. Methods Appl. Mech. Engrg., 193 (2004), no. 9-11, 745–764.
[4] Axel Klawonn and Olof Widlund, FETI and Neumann-Neumann iterative substructuring methods: Connections and new results. Comm. Pure Appl. Math., 54 (2001), no. 1, 57–90.
[5] Yongxiang Liu and Xuejun Xu, A Robin-type domain decomposition method with red-black partition. SIAM J. Numer. Anal., 52 (2014), no. 5, 2381–2399.
[6] Jan Mandel and Marian Brezina, Balancing domain decomposition for problems with large jumps in coefficients. Math. Comp., 65 (1996), no. 216, 1387–1401.
[7] Tarek P. A. Mathew, Domain decomposition methods for the numerical solution of partial differential equations, volume 61 of Lecture Notes in Computational Science and Engineering. Springer-Verlag, Berlin, 2008.
[8] M. Sarkis, Schwarz Preconditioners for Elliptic Problems with Discontinuous Coefficients Using Conforming and Non-Conforming Elements. PhD thesis, New York University, 1994.
[9] M. Sarkis, Nonstandard coarse spaces and Schwarz methods for elliptic problems with discontinuous coefficients using non-conforming elements. Numer. Math., 77 (1997), no. 3, 383–406.
[10] Andrea Toselli and Olof Widlund, Domain Decomposition Methods-Algorithms and Theory, volume 34. Springer Science & Business Media, 2006.
[11] Olof Widlund, Iterative substructuring methods: Algorithms and theory for elliptic problems in the plane. In First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, PA, 1988.
[12] Xuejun Xu and Lizhen Qin, Spectral analysis of Dirichlet–Neumann operators and optimized Schwarz methods with Robin transmission conditions. SIAM J. Numer. Anal., 47 (2010), no. 6, 4540–4568.