Numer. Math. Theor. Meth. Appl. Vol. **19**, No. 1, pp. 155-191
doi: 10.4208/nmtma.OA-2025-0054 February 2026

Analysis of Dirichlet-Neumann and Neumann-Dirichlet Methods for Time-Periodic Parabolic Optimal Control Problems

Bo Song[1,*], Jia-Yi Lv[1] and Yao-Lin Jiang[2]

[1] *School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an 710129, P.R. China*
[2] *School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, P.R. China*

Received 28 May 2025; Accepted (in revised version) 20 November 2025

Abstract. Dirichlet-Neumann and Neumann-Neumann methods are not only the parallel strategies in the spatial domain, but also can be used as a class of parallel methods in time. In this paper, we propose the Dirichlet-Neumann and Neumann-Dirichlet algorithms for time-periodic parabolic optimal control problems. By the Lagrange multiplier approach, a coupled system is obtained with a special time-periodic condition. For this coupled system, the Dirichlet-Neumann and Neumann-Dirichlet algorithms and their three variants are derived. We present the conver-gence analysis for all proposed algorithms. The numerical performance of the con-vergence factors is shown to illustrate our theoretical analysis. Based on our anal-ysis, there is a class of algorithms with the better convergence compared with the natural Dirichlet-Neumann algorithm. Finally, numerical experiments are provided to illustrate the theoretical results.

AMS subject classifications: 65N12, 65N55, 65Y05

Key words: Time-periodic parabolic optimal control problems, Dirichlet-Neumann algorithm, Neumann-Dirichlet algorithm, time domain decomposition, convergence analysis.

1. Introduction

Optimal control problems governed by the time-periodic parabolic problem are widely used in many different fields, such as Stokes problems [2], eddy current problems [3, 25], power generating kite systems [24], and so on. Discretized parabolic

*Corresponding author. *Email addresses:* `bosong@nwpu.edu.cn` (B. Song), `jylv@mail.nwpu.edu.cn` (J.-Y. Lv), `yljiang@mail.xjtu.edu.cn` (Y.-L. Jiang)

optimal control problems lead to very large systems of equations in time and space. Domain decomposition methods have been used to solve this difficulty [8, 30]. In addition, parabolic optimal control problems can be described as a forward-backward optimality system in time, like [6, 38], and for this forward-backward optimality system, the time variable plays a vital role. Therefore, it is very interesting to derive time domain decomposition methods and parallel solvers in time for such systems. Different parallel-in-time preconditioners have been developed and analyzed for this class of optimal control problems, see, e.g., [32, 39, 40] and the references therein.

Recently, various kinds of time parallel strategies have been well developed, and the development of time parallel methods has a history of more than five decades which is completely reviewed in [14]. Parareal algorithm is a classical time parallel method proposed by Lions *et al.* [33] has been analyzed or used in many different areas, like Navier-Stokes equations [12], time-periodic problems [15], and optimal control problems governed by the initial-value parabolic problems [9, 36]. And ParaOpt algorithm is a new variant of parareal for optimal control problems, which has been derived in [17]. Besides, several new algorithms parallel in time have been proposed, such as PFASST [10], MGRiT [11, 13], ParaDiag [19, 34]. Furthermore, these new parallel-in-time methods have been applied to optimal control problems by some scholars, such as PFASST [22, 23], MGRiT [1], ParaExp [28].

Various parallel strategies have been developed both in space and time directions for optimal control problems. For the elliptic optimal control problems, the domain decomposition methods and their convergence analysis were discussed by Benamou in [4, 5]. And Dirichlet-Neumann waveform relaxation (DNWR) and Neumann-Neumann waveform relaxation (NNWR) methods were constructed to solve the parabolic optimal control problems in the context [35]. With regard to time-periodic parabolic control problems, optimized Schwarz waveform relaxation (OSWR) method was studied and analyzed in [7]. The context [29] presented time-domain decomposition method for optimal control problems governed by the wave equation, and this method was extended to optimal control problems for semilinear hyperbolic systems in [26]. Time domain decomposition methods can also be applied to solve initial-value parabolic optimal control problems. For example, Schwarz method as a time parallel algorithm was proposed by Gander and Kwok [16], and optimized Schwarz method was discussed by Kwok [27]. Furthermore, the Dirichlet-Neumann/Neumann-Neumann algorithms as the time domain decomposition algorithms were presented for initial-value parabolic optimal control problems in [20, 21]. Furthermore, space-time decomposition method applied to optimal control problems was studied in [31, 41].

In this work, we propose several time domain decomposition algorithms based on the Dirichlet-Neumann (DN) and Neumann-Dirichlet (ND) methods to solve time-periodic parabolic optimal control problems. Different from [7] which introduced OSWR for time-periodic parabolic optimal control problems, we use the DN and ND methods as time domain decomposition algorithms to solve this problem. Compared with initial-value parabolic optimal control problems [20, 21], the time-periodic condition is more difficult to deal with, because the coupled system obtained by the Lagrange multiplier approach has the time-periodic condition rather than the initial-value condition, and the initial and final values of time-periodic problem are unknown which imposes the difficulty when solving the system. The DN and ND algorithms can change the time-periodic structure of the original system, and transform it into Dirichlet condition and Neumann condition on the subdomains. For the convergence analysis of the DN and ND algorithms, we use the undetermined coefficient method to calculate the convergence factors of the proposed algorithms. Besides, based on the analysis and comparison of the convergence factors, we get a class of algorithms with better convergence behavior than the natural DN algorithm. Several numerical experiments illustrate our theoretical results.

The rest of our paper is organized as follows. In Section 2, we construct our new time domain decomposition algorithms based on the DN and ND algorithms for time-periodic parabolic optimal control problems. The convergence behavior of each algorithm is analyzed in Section 3. In Section 4, we show the numerical behavior of the proposed algorithms and the conclusions are presented finally in Section 5.

2. Dirichlet-Neumann and Neumann-Dirichlet algorithms in time for time-periodic optimal control problem

2.1. Construction of the algorithms

In this paper, we consider the following optimal control problem:

$$\min_{y,u} J(y,u) := \frac{1}{2}\|y-y_d\|^2_{L^2(\Omega\times(0,T))} + \frac{r}{2}\|u\|^2_{L^2(\Omega\times(0,T))}, \tag{2.1}$$

where $\Omega \subset \mathbb{R}^d$, $d=1,2,3$ is a bounded domain with boundary $\partial\Omega$, $r>0$, $J(y,u)$ is the cost function, y is the state variable, and y_d is the target function. This problem seeks the input control u to minimize the cost functional $J(y,u)$, which measures the deviation of y from a desired target y_d. Here, the state y is governed by the following time-periodic parabolic state equation:

$$\begin{cases} \partial_t y - \Delta y = u & \text{in } \ \Omega\times(0,T), \\ y = 0 & \text{on } \ \partial\Omega\times(0,T), \\ y(0) = y(T) & \text{in } \ \Omega. \end{cases} \tag{2.2}$$

Using the Lagrange multiplier approach to the model problem (2.1)-(2.2), one can derive the adjoint problem as follows [18, 37], which is backward in time:

$$\begin{cases} \partial_t \lambda + \Delta\lambda = y - y_d & \text{in } \ \Omega\times(0,T), \\ \lambda = 0 & \text{on } \ \partial\Omega\times(0,T), \\ \lambda(T) = \lambda(0) & \text{in } \ \Omega, \end{cases} \tag{2.3}$$

where the control u and the adjoint state λ satisfy the relation $\lambda = ru$ in $\Omega \times (0,T)$. Then, the solution of the original model problem (2.1)-(2.2) is equivalent to the problem (2.2)-(2.3). By the semi-discretization in the spatial direction to (2.2)-(2.3), we get the following forward-backward system in time:

$$\begin{cases} \dot{\boldsymbol{y}} + A\boldsymbol{y} = \boldsymbol{u} & \text{in } (0,T), \\ \boldsymbol{y}(0) = \boldsymbol{y}(T), \end{cases}$$
$$\begin{cases} \dot{\boldsymbol{\lambda}} - A^T\boldsymbol{\lambda} = \boldsymbol{y} - \boldsymbol{y_d} & \text{in } (0,T), \\ \boldsymbol{\lambda}(T) = \boldsymbol{\lambda}(0), \end{cases} \tag{2.4}$$

where $\dot{y}$, $\dot{\lambda}$ denotes the time derivative of y, λ, $A \in \mathbb{R}^{m\times m}$ is the matrix obtained by discretizing the $-\Delta$ operator, and the vectors $\boldsymbol{y}(t), \boldsymbol{\lambda}(t), \boldsymbol{u}(t) \in \mathbb{R}^{m+1}$ are obtained by the discretization of y, λ, u in space.

Eliminating the control u, the Eqs. (2.4) can be rewritten as the following forward-backward system:

$$\begin{cases} \begin{pmatrix} \dot{\boldsymbol{y}} \\ \dot{\boldsymbol{\lambda}} \end{pmatrix} + \begin{pmatrix} A & -\dfrac{1}{r}I \\ -I & -A^T \end{pmatrix} \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} 0 \\ -\boldsymbol{y_d} \end{pmatrix} & \text{in } (0,T), \\ \boldsymbol{y}(0) = \boldsymbol{y}(T), \\ \boldsymbol{\lambda}(T) = \boldsymbol{\lambda}(0). \end{cases} \tag{2.5}$$

Assuming that A is symmetric, i.e., $A = A^T$, then the matrix A can be diagonalized as $A = QDQ^T$, with $Q^TQ = I$ and $D = \operatorname{diag}(d_1, \ldots, d_m)$, d_i is the eigenvalue of A. By the transformations $\boldsymbol{z} = Q^T\boldsymbol{y}$, $\boldsymbol{\mu} = Q^T\boldsymbol{\lambda}$, and $\boldsymbol{z_d} = Q^T\boldsymbol{y_d}$, the system (2.5) can be written as follows:

$$\begin{cases} \begin{pmatrix} \dot{\boldsymbol{z}} \\ \dot{\boldsymbol{\mu}} \end{pmatrix} + \begin{pmatrix} D & -\dfrac{1}{r}I \\ -I & -D \end{pmatrix} \begin{pmatrix} \boldsymbol{z} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} 0 \\ -\boldsymbol{z_d} \end{pmatrix} & \text{in } (0,T), \\ \boldsymbol{z}(0) = \boldsymbol{z}(T), \\ \boldsymbol{\mu}(T) = \boldsymbol{\mu}(0). \end{cases} \tag{2.6}$$

Remark 2.1. For the general parabolic PDE $\dot{y} + \mathcal{L}y = u$, when it is semi-discretized in space, we get the equation $\dot{y} + Ay = u$, where the matrix A is obtained by discretizing the $\mathcal{L}$ operator. Our work is applicable to any case where the matrix A is symmetric, not only for the above heat equation (2.2).

2.2. Dirichlet-Neumann and Neumann-Dirichlet algorithms in time

We now consider the Dirichlet-Neumann and Neumann-Dirichlet algorithms in time to solve the system (2.6). Firstly, we divide the time interval $(0,T)$ into two non-overlapping subintervals $I_1 = (0,\alpha)$ and $I_2 = (\alpha,T)$. We use $\boldsymbol{z_j}$ and $\boldsymbol{\mu_j}$ to indicate the solutions $\boldsymbol{z}$ and $\boldsymbol{\mu}$ of (2.6) in I_j, $j = 1, 2$. Since the initial and final conditions of the forward-backward system (2.6) are periodic, we propose a natural Dirichlet-Neumann algorithm in time for the coupled system (2.6) as follows:

$$\begin{cases} \begin{pmatrix} \dot{\boldsymbol{z}}_1^k \\ \dot{\boldsymbol{\mu}}_1^k \end{pmatrix} + \begin{pmatrix} D & -\frac{1}{r}I \\ -I & -D \end{pmatrix} \begin{pmatrix} \boldsymbol{z}_1^k \\ \boldsymbol{\mu}_1^k \end{pmatrix} = \begin{pmatrix} 0 \\ -\boldsymbol{z_d} \end{pmatrix} \quad \text{in } I_1 = (0, \alpha), \\ \boldsymbol{z}_1^k(0) = \boldsymbol{z}_2^k(T), \\ \boldsymbol{\mu}_1^k(\alpha) = \boldsymbol{f}_{\boldsymbol{\alpha}}^{k-1}, \end{cases} \tag{2.7}$$

$$\begin{cases} \begin{pmatrix} \dot{\boldsymbol{z}}_2^k \\ \dot{\boldsymbol{\mu}}_2^k \end{pmatrix} + \begin{pmatrix} D & -\frac{1}{r}I \\ -I & -D \end{pmatrix} \begin{pmatrix} \boldsymbol{z}_2^k \\ \boldsymbol{\mu}_2^k \end{pmatrix} = \begin{pmatrix} 0 \\ -\boldsymbol{z_d} \end{pmatrix} \quad \text{in } I_2 = (\alpha, T), \\ \dot{\boldsymbol{z}}_2^k(\alpha) = \dot{\boldsymbol{z}}_1^k(\alpha), \\ \boldsymbol{\mu}_2^k(T) = \boldsymbol{\mu}_1^k(0), \end{cases} \tag{2.8}$$

and then update the value along the interface using

$$\boldsymbol{f}_{\boldsymbol{\alpha}}^k := (1-\theta)\boldsymbol{f}_{\boldsymbol{\alpha}}^{k-1} + \theta \boldsymbol{\mu}_2^k(\alpha), \tag{2.9}$$

where $\theta \in (0, 1]$ is a relaxation parameter.

Given an initial guess $\boldsymbol{f}_{\boldsymbol{\alpha}}^0$ along the interface $t = \alpha$, we solve the above DN algorithm for the iteration index $k = 1, 2, \ldots$. However, due to the initial condition value $\boldsymbol{z}_2^k(T)$ in subdomain I_1 and the final condition value $\boldsymbol{\mu}_1^k(0)$ in I_2 are unknown for the k-th iteration, we find the Eqs. (2.7) and (2.8) can not be solved which leads to the failure of decoupling the original system (2.6). Then it is easy to see that the natural algorithm (2.7)-(2.9) can not be realized in practice. Therefore, in order to solve this problem, we propose the following algorithm (2.10)-(2.12) by changing the conditions in (2.7) and (2.8) to ensure that the proposed algorithm can be realized:

$$\begin{cases} \begin{pmatrix} \dot{\boldsymbol{z}}_1^k \\ \dot{\boldsymbol{\mu}}_1^k \end{pmatrix} + \begin{pmatrix} D & -\frac{1}{r}I \\ -I & -D \end{pmatrix} \begin{pmatrix} \boldsymbol{z}_1^k \\ \boldsymbol{\mu}_1^k \end{pmatrix} = \begin{pmatrix} 0 \\ -\boldsymbol{z_d} \end{pmatrix} \quad \text{in } I_1 = (0, \alpha), \\ \boldsymbol{z}_1^k(0) = \boldsymbol{z}_2^{k-1}(T), \\ \boldsymbol{\mu}_1^k(\alpha) = \boldsymbol{f}_{\boldsymbol{\alpha}}^{k-1}, \end{cases} \tag{2.10}$$

$$\begin{cases} \begin{pmatrix} \dot{\boldsymbol{z}}_2^k \\ \dot{\boldsymbol{\mu}}_2^k \end{pmatrix} + \begin{pmatrix} D & -\frac{1}{r}I \\ -I & -D \end{pmatrix} \begin{pmatrix} \boldsymbol{z}_2^k \\ \boldsymbol{\mu}_2^k \end{pmatrix} = \begin{pmatrix} 0 \\ -\boldsymbol{z_d} \end{pmatrix} \quad \text{in } I_2 = (\alpha, T), \\ \dot{\boldsymbol{z}}_2^k(\alpha) = \dot{\boldsymbol{z}}_1^k(\alpha), \\ \boldsymbol{\mu}_2^k(T) = \boldsymbol{\mu}_1^k(0), \end{cases} \tag{2.11}$$

and then update the value along the interface using

$$\boldsymbol{f}_{\boldsymbol{\alpha}}^k := (1-\theta)\boldsymbol{f}_{\boldsymbol{\alpha}}^{k-1} + \theta \boldsymbol{\mu}_2^k(\alpha), \tag{2.12}$$

where $\theta \in (0, 1]$ is a relaxation parameter.

Remark 2.2. When we change the condition $\boldsymbol{z}_1^k(0) = \boldsymbol{z}_2^k(T)$ in (2.7) to the condition $\boldsymbol{z}_1^k(0) = \boldsymbol{z}_2^{k-1}(T)$ in (2.10), we can find that the new algorithm can be realized, and the initial and final conditions in (2.10) are Dirichlet conditions, the conditions in (2.11) are mixed Dirichlet-Neumann conditions, so the new algorithm (2.10)-(2.12) is a natural DN algorithm.

Remark 2.3. By the first equation in (2.6), we can get the following identities for $\boldsymbol{z}$ and $\boldsymbol{\mu}$:

$$\begin{aligned} \boldsymbol{z} - \boldsymbol{z_d} &= \dot{\boldsymbol{\mu}} - D\boldsymbol{\mu}, \\ \boldsymbol{\mu} &= r(\dot{\boldsymbol{z}} + D\boldsymbol{z}). \end{aligned} \tag{2.13}$$

Then we can find $\boldsymbol{z}$ and $\boldsymbol{\mu}$ can be transformed into each other. Based on the above relationships, we can propose some new algorithms using the DN technique.

For the DN and ND algorithms, there are some other approaches to implement for the model problem (2.6), by changing the conditions at the interface $t = \alpha$. We can apply the DN technique to both states $(\boldsymbol{z}, \boldsymbol{\mu})$ just like in (2.10)-(2.11). Also, based on the identities (2.13) in Remark 2.3, we know that the variables $\boldsymbol{z}$ and $\boldsymbol{\mu}$ are mutually convertible, so we can apply the DN technique just to one of these two states. Moreover, we can exchange the order of the Dirichlet and Neumann conditions. Then, we can divide the algorithms into three strategies, and each of them contains two parts, namely the DN and ND techniques. Based on this analysis, we list all six time domain decomposition algorithms in Table 1.

Table 1: Variants of the DN algorithm for time-periodic parabolic optimal control problems.

	Problem	I_1	I_2
Category I: $(\boldsymbol{z}, \boldsymbol{\mu})$	DN_1	$\boldsymbol{\mu}$	$\dot{\boldsymbol{z}}$
	ND_1	$\dot{\boldsymbol{\mu}}$	$\boldsymbol{z}$
Category II: $\boldsymbol{z}$	DN_2	$\boldsymbol{z}$	$\dot{\boldsymbol{z}}$
	ND_2	$\dot{\boldsymbol{z}}$	$\boldsymbol{z}$
Category III: $\boldsymbol{\mu}$	DN_3	$\boldsymbol{\mu}$	$\dot{\boldsymbol{\mu}}$
	ND_3	$\dot{\boldsymbol{\mu}}$	$\boldsymbol{\mu}$

3. Convergence analysis

3.1. Category I

We consider Category I firstly which applies the DN and ND techniques to the pair states (z, μ) to the model problem (2.6), and analyze their convergence.

3.1.1. DN_1

Since the model problem (2.6) is linear, we can analyze the convergence of the algorithm by studying the error equation. Thus, we can set $y_d = 0$, and study how $\boldsymbol{y}_j^{k+1}$

and $\boldsymbol{\lambda}_{\boldsymbol{j}}^{\boldsymbol{k+1}}(j=1,2)$ converge to zero as $k\to\infty$. Under the above assumption, we can obtain m independent 2×2 systems by (2.10)-(2.12) for $i=1,2,\ldots,m$ as follows:

$$\begin{cases} \begin{pmatrix} \dot{z}_{1,(i)}^k \\ \dot{\mu}_{1,(i)}^k \end{pmatrix} + \begin{pmatrix} d_i & -\dfrac{1}{r} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{1,(i)}^k \\ \mu_{1,(i)}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{in } I_1, \\ z_{1,(i)}^k(0) = z_{2,(i)}^{k-1}(T), \\ \mu_{1,(i)}^k(\alpha) = f_\alpha^{k-1}, \end{cases}$$
$$\begin{cases} \begin{pmatrix} \dot{z}_{2,(i)}^k \\ \dot{\mu}_{2,(i)}^k \end{pmatrix} + \begin{pmatrix} d_i & -\dfrac{1}{r} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{2,(i)}^k \\ \mu_{2,(i)}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{in } I_2, \\ \dot{z}_{2,(i)}^k(\alpha) = \dot{z}_{1,(i)}^k(\alpha), \\ \mu_{2,(i)}^k(T) = \mu_{1,(i)}^k(0), \end{cases} \tag{3.1}$$

and then update the value along the interface using

$$f_{\alpha,i}^k := (1-\theta) f_{\alpha,i}^{k-1} + \theta \mu_{2,(i)}^k(\alpha), \tag{3.2}$$

where $z_{j,(i)}^k$, $\mu_{j,(i)}^k$, $f_{\alpha,i}^k$ are the i-th components of $\boldsymbol{z}_{\boldsymbol{j}}^{\boldsymbol{k}}$, $\boldsymbol{\mu}_{\boldsymbol{j}}^{\boldsymbol{k}}$, $\boldsymbol{f}_{\boldsymbol{\alpha}}^{\boldsymbol{k}}$ for $j=1,2$.

Furthermore, according to the above assumption of $z_d=0$ and the identities (2.13) in Remark 2.3, we obtain the following relationships between $z_{j,(i)}$ and $\mu_{j,(i)}$, $j=1,2$:

$$z_{j,(i)}^k = \dot{\mu}_{j,(i)}^k - d_i \mu_{j,(i)}^k, \tag{3.3a}$$
$$\mu_{j,(i)}^k = r\left(\dot{z}_{j,(i)}^k + d_i z_{j,(i)}^k\right). \tag{3.3b}$$

By observing Eqs. (3.1), we find that the DN_1 algorithm maintains the forward-backward structure of the original model problem (2.6). To facilitate the convergence analysis, we substitute the above identities (3.3) into the systems (3.1)-(3.2), the following system is obtained:

$$\begin{cases} \ddot{z}_{1,(i)}^k - \sigma_i^2 z_{1,(i)}^k = 0 \quad \text{in } I_1, \\ z_{1,(i)}^k(0) = z_{2,(i)}^{k-1}(T), \\ \dot{z}_{1,(i)}^k(\alpha) + d_i z_{1,(i)}^k(\alpha) = f_{\alpha,i}^{k-1}, \end{cases} \tag{3.4a}$$

$$\begin{cases} \ddot{z}_{2,(i)}^k - \sigma_i^2 z_{2,(i)}^k = 0 \quad \text{in } I_2, \\ \dot{z}_{2,(i)}^k(\alpha) = \dot{z}_{1,(i)}^k(\alpha), \\ \dot{z}_{2,(i)}^k(T) + d_i z_{2,(i)}^k(T) = \dot{z}_{1,(i)}^k(0) + d_i z_{1,(i)}^k(0), \end{cases} \tag{3.4b}$$

and then update the value along the interface using

$$f_{\alpha,i}^k := (1-\theta) f_{\alpha,i}^{k-1} + \theta\left(\dot{z}_{2,(i)}^k(\alpha) + d_i z_{2,(i)}^k(\alpha)\right),$$

where $\sigma_i := \sqrt{d_i^2 + r^{-1}}$.

The solutions of (3.4) can be written as follows:

$$
\begin{aligned}
z_{1,(i)}^k &= A_{1,i}^k \sinh(\sigma_i t) + A_{2,i}^k \cosh(\sigma_i t),\\
z_{2,(i)}^k &= B_{1,i}^k \sinh\big(\sigma_i(T-t)\big) + B_{2,i}^k \cosh\big(\sigma_i(T-t)\big),
\end{aligned} \tag{3.5}
$$

where $A_{1,i}$, $A_{2,i}$ and $B_{1,i}$, $B_{2,i}$ are four coefficients.

To simplify the notation in the following analysis, we define $a_i := \sigma_i\alpha$, $b_i := \sigma_i(T-\alpha)$, i.e., $a_i + b_i = \sigma_i T$, and the periodic condition value is unknown, so we set

$$
m_i^k := z_{1,(i)}^{k+1}(0) = z_{2,(i)}^k(T). \tag{3.6}
$$

Substituting identities (3.5) into (3.6), we can get the following relations:

$$
z_{1,(i)}^{k+1}(0) = A_{2,i}^{k+1} = m_i^k, \quad z_{2,(i)}^k(T) = B_{2,i}^k = m_i^k.
$$

Furthermore, according to the initial and final conditions of the Eqs. (3.4a), we find

$$
\begin{aligned}
\dot{z}_{1,(i)}^k + d_i z_{1,(i)}^k \,|_{t=\alpha} = {} & \big(\sigma_i \cosh(a_i) + d_i \sinh(a_i)\big) A_{1,i}^k \\
& + \big(\sigma_i \sinh(a_i) + d_i \cosh(a_i)\big) m_i^{k-1} = f_{\alpha,i}^{k-1}.
\end{aligned}
$$

Thus, we get the expression of $A_{1,i}^k$ with $f_{\alpha,i}^{k-1}$ and m_i^{k-1} as follows:

$$
A_{1,i}^k = \frac{f_{\alpha,i}^{k-1} - (\sigma_i \sinh(a_i) + d_i \cosh(a_i)) m_i^{k-1}}{\sigma_i \cosh(a_i) + d_i \sinh(a_i)}. \tag{3.7}
$$

Similarly, we can obtain the linear equations about $B_{1,i}^k, B_{2,i}^k$ by the initial and final conditions of the Eqs. (3.4b),

$$
\begin{cases}
-\sigma_i B_{1,i}^k + d_i B_{2,i}^k = \sigma_i A_{1,i}^k + d_i m_i^{k-1},\\
\cosh(b_i) B_{1,i}^k + \sinh(b_i) B_{2,i}^k = -\cosh(a_i) A_{1,i}^k - \sinh(a_i) m_i^{k-1}.
\end{cases}
$$

Solve the above equations, which yields that

$$
B_{1,i}^k = -\frac{(d_i \cosh(a_i) + \sigma_i \sinh(b_i)) A_{1,i}^k + d_i(\sinh(a_i) + \sinh(b_i)) m_i^{k-1}}{\sigma_i \sinh(b_i) + d_i \cosh(b_i)}, \tag{3.8a}
$$

$$
B_{2,i}^k = \frac{\sigma_i(\cosh(b_i) - \cosh(a_i)) A_{1,i}^k + (d_i \cosh(b_i) - \sigma_i \sinh(a_i)) m_i^{k-1}}{\sigma_i \sinh(b_i) + d_i \cosh(b_i)}. \tag{3.8b}
$$

To shorten the identities, we denote

$$
\begin{aligned}
e_i &:= \sigma_i \cosh(a_i) + d_i \sinh(a_i), & E_i &:= \sigma_i \sinh(a_i) + d_i \cosh(a_i),\\
f_i &:= \sigma_i \cosh(b_i) + d_i \sinh(b_i), & F_i &:= \sigma_i \sinh(b_i) + d_i \cosh(b_i),\\
h_i &:= \sigma_i \cosh(\sigma_i T) + d_i \sinh(\sigma_i T), & H_i &:= \sigma_i \sinh(\sigma_i T) + d_i \cosh(\sigma_i T).
\end{aligned} \tag{3.9}
$$

Substituting (3.7) into the Eq. (3.8b), we get the expression of m_i^k as follows:

$$\begin{aligned} m_i^k = B_{2,i}^k = & \frac{\sigma_i(\cosh(b_i) - \cosh(a_i))}{e_i F_i} f_{\alpha,i}^{k-1} \\ & + \frac{\sigma_i d_i - r^{-1}\sinh(a_i)\cosh(b_i)}{e_i F_i} m_i^{k-1}. \end{aligned} \tag{3.10}$$

For $f_{\alpha,i}^k$, we obtain

$$\begin{aligned} f_{\alpha,i}^k &= (1-\theta) f_{\alpha,i}^{k-1} + \theta\left(\dot{z}_{2,(i)}^k(\alpha) + d_i z_{2,(i)}^k(\alpha)\right) \\ &= (1-\theta) f_{\alpha,i}^{k-1} + \theta\Big[\sigma_i\left(\cosh(a_i)A_{1,i}^k + \sinh(a_i)m_i^{k-1}\right) \\ &\qquad + d_i\left(\sinh(b_i)B_{1,i}^k + \cosh(b_i)B_{2,i}^k\right)\Big] \\ &= \left(1 - \theta d_i \frac{h_i - \sigma_i}{e_i F_i}\right) f_{\alpha,i}^{k-1} - r^{-1}\theta d_i \frac{\sinh(a_i) + \sinh(b_i)}{e_i F_i} m_i^{k-1}. \end{aligned} \tag{3.11}$$

Combining (3.10) and (3.11), the iterative relation is obtained as follows:

$$\begin{pmatrix} f_{\alpha,i}^k \\ m_i^k \end{pmatrix} = Q_{DN_1,i} \begin{pmatrix} f_{\alpha,i}^{k-1} \\ m_i^{k-1} \end{pmatrix},$$

where

$$\begin{aligned} Q_{DN_1,i} &:= \begin{pmatrix} Q_i^1 & Q_i^2 \\ Q_i^3 & Q_i^4 \end{pmatrix} \\ &:= \begin{pmatrix} \dfrac{e_i F_i + \theta d_i(\sigma_i - h_i)}{e_i F_i} & \dfrac{-r^{-1}\theta d_i(\sinh(a_i) + \sinh(b_i))}{e_i F_i} \\ \dfrac{\sigma_i(\cosh(b_i) - \cosh(a_i))}{e_i F_i} & \dfrac{\sigma_i d_i - r^{-1}\sinh(a_i)\cosh(b_i)}{e_i F_i} \end{pmatrix}. \end{aligned} \tag{3.12}$$

Solving the characteristic polynomial of the iterative matrix $Q_{DN_1,i}$, we get its eigenvalues as follows:

$$\lambda_{DN_1,i} := \frac{(Q_i^1 + Q_i^4) \pm \sqrt{(Q_i^1 - Q_i^4)^2 + 4Q_i^2 Q_i^3}}{2}. \tag{3.13}$$

Finally, we obtain the following convergence theorem for the DN_1 algorithm.

Theorem 3.1. *Algorithm* DN_1 (3.1)-(3.2) *converges if and only if*

$$\rho_{DN_1} := \max_{i=1,2,\ldots,m} |\rho_{Q_{DN_1,i}}| < 1, \tag{3.14}$$

where $\rho_{Q_{DN_1,i}}$ *is the spectrum of the matrix* $Q_{DN_1,i}$.

To understand the performance of the convergence factor (3.14) more deeply, we consider a few special cases. Firstly, considering the time decomposition is symmetric $\alpha = T/2$, i.e., $a_i = b_i$, we derive the following corollary.

Corollary 3.1. *If the time domain decomposition is symmetric $\alpha = T/2$, and the matrix A is not singular, the DN_1 algorithm* (3.1)-(3.2) *converges for any initial guess.*

Proof. Noting that from the definition of h_i, we find

$$h_i = \sigma_i \cosh(\sigma_i T) + d_i \sinh(\sigma_i T) > \sigma_i. \tag{3.15}$$

Then using (3.15), we obtain

$$\begin{aligned}\lambda^{(1)}_{DN_1,i} &:= \frac{(Q^1_i + Q^4_i) + (Q^1_i - Q^4_i)}{2} = Q^1_i \\ &= \frac{2\theta\sigma_i d_i + 2(1-\theta)d_i h_i + r^{-1}\sinh(\sigma_i T)}{2d_i h_i + r^{-1}\sinh(\sigma_i T)} \\ &\le 1. \quad \text{(If and only if } \theta = 0 \text{ or } d_i = 0\text{, take the equal sign.)}\end{aligned}$$

For the second kind of eigenvalues,

$$\begin{aligned}\lambda^{(2)}_{DN_1,i} &:= \frac{(Q^1_i + Q^4_i) - (Q^1_i - Q^4_i)}{2} = Q^4_i \\ &= \frac{2\sigma_i d_i - r^{-1}\sinh(\sigma_i T)}{2d_i h_i + r^{-1}\sinh(\sigma_i T)}.\end{aligned}$$

Taking the square of $\lambda^{(2)}_{DN_1,i}$, the following relation is obtained:

$$\begin{aligned}\left|\lambda^{(2)}_{DN_1,i}\right|^2 &= \left|\frac{4\sigma_i^2 d_i^2 - 4\sigma_i d_i r^{-1}\sinh(\sigma_i T) + r^{-2}\sinh^2(\sigma_i T)}{4d_i^2 h_i^2 + 4d_i h_i r^{-1}\sinh(\sigma_i T) + r^{-2}\sinh^2(\sigma_i T)}\right| \\ &\le 1. \quad \text{(If and only if } d_i = 0\text{, take the equal sign.)}\end{aligned}$$

Thus, we can conclude that $|\lambda^{(2)}_{DN_1,i}| \le 1$, if and only if $d_i = 0$, take the equal sign. Finally, if $d_i \neq 0$, $i = 1, 2, \ldots, m$, namely the matrix A is not singular, we can get

$$\rho_{DN_1} = \max_{i=1,2,\ldots,m} |\rho_{Q_{DN_1,i}}| = \max_{i=1,2,\ldots,m} |\lambda_{DN_1,i}| < 1,$$

which concludes the proof. □

Remark 3.1. If the matrix A is discretized by the Laplace operator in our model problem, there is no zero eigenvalue for A. If $d_i = 0$, according to (3.12), we obtain

$$\begin{aligned}&Q^1_i|_{d_i=0} = 1, \quad Q^4_i|_{d_i=0} = \frac{-r^{-1}\sinh(a_i)\cosh(b_i)}{e_i F_i}, \\ &Q^2_i Q^3_i|_{d_i=0} = 0, \quad (e_i F_i)|_{d_i=0} = r^{-1}\sinh(b_i)\cosh(a_i).\end{aligned}$$

Substituting the above equations into (3.13), we find

$$\lambda_{DN_1,i}^{(1)} = Q_i^1 = 1, \quad \lambda_{DN_1,i}^{(2)} = Q_i^4 = -\frac{\sinh(a_i)\cosh(b_i)}{\sinh(b_i)\cosh(a_i)} = -\frac{\tanh(a_i)}{\tanh(b_i)}.$$

If $a_i \geq b_i$, we get $|\lambda_{DN_1,i}^{(2)}| \geq 1$. Oppositely, the $|\lambda_{DN_1,i}^{(2)}| < 1$ can be obtained if $a_i < b_i$.

Finally, we conclude that $\rho_{Q_{DN_1,i}}|_{d_i=0} \geq 1$ when $a_i \geq b_i$, while if $a_i < b_i$, $\rho_{Q_{DN_1,i}}|_{d_i=0} \equiv 1$. Then the convergence behavior of the DN_1 algorithm for small eigenvalues is not good.

Remark 3.2. If d_i goes to infinity, we have $\sigma_i \sim_\infty d_i$, and then we can get $Q_i^2 Q_i^3 \to 0$, $Q_i^1 \to 1-\theta$, $Q_i^4 \to 0$. Substituting these into (3.13), we obtain

$$\lim_{d_i \to \infty} \rho_{Q_{DN_1,i}} = 1 - \theta.$$

In other words, the DN_1 algorithm is robust for high frequency convergence with relaxation, and when using $\theta = 1$, we obtain a good smoother.

3.1.2. ND_1

We change two conditions at the interface $t = \alpha$, and apply the Neumann condition to the state μ in I_1 and the Dirichlet condition to the state z in I_2. Then we obtain the ND_1 algorithm which still keeps the forward-backward structure for the iteration index $k = 1, 2, \ldots$ as follows:

$$\begin{cases} \begin{pmatrix} \dot{z}_{1,(i)}^k \\ \dot{\mu}_{1,(i)}^k \end{pmatrix} + \begin{pmatrix} d_i & -\frac{1}{r} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{1,(i)}^k \\ \mu_{1,(i)}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{in } I_1, \\ z_{1,(i)}^k(0) = z_{2,(i)}^k(T), \\ \dot{\mu}_{1,(i)}^k(\alpha) = \dot{\mu}_{2,(i)}^k(\alpha), \end{cases}$$
$$\begin{cases} \begin{pmatrix} \dot{z}_{2,(i)}^k \\ \dot{\mu}_{2,(i)}^k \end{pmatrix} + \begin{pmatrix} d_i & -\frac{1}{r} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{2,(i)}^k \\ \mu_{2,(i)}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{in } I_2, \\ z_{2,(i)}^k(\alpha) = f_{\alpha,i}^{k-1}, \\ \mu_{2,(i)}^k(T) = \mu_{1,(i)}^{k-1}(0), \end{cases} \tag{3.16}$$

and then update the value along the interface using

$$f_{\alpha,i}^k := (1-\theta) f_{\alpha,i}^{k-1} + \theta z_{1,(i)}^k(\alpha). \tag{3.17}$$

Based on (3.3), we obtain the systems as follows:

$$\begin{cases} \ddot{\mu}_{1,(i)}^k - \sigma_i^2 \mu_{1,(i)}^k = 0 \quad \text{in } I_1, \\ \dot{\mu}_{1,(i)}^k(0) - d_i \mu_{1,(i)}^k(0) = \dot{\mu}_{2,(i)}^k(T) - d_i \mu_{2,(i)}^k(T), \\ \dot{\mu}_{1,(i)}^k(\alpha) = \dot{\mu}_{2,(i)}^k(\alpha), \end{cases} \tag{3.18a}$$

$$\begin{cases} \ddot{\mu}^k_{2,(i)} - \sigma_i^2 \mu^k_{2,(i)} = 0 \quad \text{in } \ I_2, \\ \dot{\mu}^k_{2,(i)}(\alpha) - d_i \mu^k_{2,(i)}(\alpha) = f^{k-1}_{\alpha,i}, \\ \mu^k_{2,(i)}(T) = \mu^{k-1}_{1,(i)}(0), \end{cases} \tag{3.18b}$$

and then update the value along the interface using

$$f^k_{\alpha,i} := (1-\theta) f^{k-1}_{\alpha,i} + \theta \left(\dot{\mu}^k_{1,(i)}(\alpha) - d_i \mu^k_{1,(i)}(\alpha) \right).$$

And the solutions of (3.18) can be written as follows:

$$\begin{aligned} \mu^k_{1,(i)} &= A^k_{1,i} \sinh(\sigma_i t) + A^k_{2,i} \cosh(\sigma_i t), \\ \mu^k_{2,(i)} &= B^k_{1,i} \sinh\big(\sigma_i (T-t)\big) + B^k_{2,i} \cosh\big(\sigma_i (T-t)\big), \end{aligned} \tag{3.19}$$

where $A_{1,i}$, $A_{2,i}$ and $B_{1,i}$, $B_{2,i}$ are four coefficients.

To simplify the notation in the following analysis, we define

$$m^k_i := \mu^{k+1}_{2,(i)}(T) = \mu^k_{1,(i)}(0). \tag{3.20}$$

Substituting identities (3.19) into (3.20), we can get relations as follows:

$$\mu^k_{1,(i)}(0) = A^k_{2,i} = m^k_i, \quad \mu^{k+1}_{2,(i)}(T) = B^{k+1}_{2,i} = m^k_i.$$

Furthermore, according to the initial and final conditions of the right equations in (3.18), we find

$$\begin{aligned} &\dot{\mu}^k_{2,(i)}(\alpha) - d_i \mu^k_{2,(i)}(\alpha) \\ &= -\big(\sigma_i \cosh(b_i) + d_i \sinh(b_i)\big) B^k_{1,i} \\ &\quad - \big(\sigma_i \sinh(b_i) + d_i \cosh(b_i)\big) m^{k-1}_i = f^{k-1}_{\alpha,i}. \end{aligned}$$

Thus, we can get the expression of $B^k_{1,i}$ with $f^{k-1}_{\alpha,i}$ and m^{k-1}_i as follows:

$$B^k_{1,i} = -\frac{f^{k-1}_{\alpha,i} + F_i m^{k-1}_i}{f_i}, \tag{3.21}$$

where f_i, F_i are defined by (3.9).

Similarly, we get the linear equations about $A^k_{1,i}$, $A^k_{2,i}$ by the initial and final conditions of the Eqs. (3.18a),

$$\begin{cases} \sigma_i A^k_{1,i} - d_i A^k_{2,i} = \sigma_i B^k_{1,i} + d_i m^{k-1}_i, \\ \cosh(a_i) A^k_{1,(i)} + \sinh(a_i) A^k_{2,i} = -\cosh(b_i) B^k_{1,i} - \sinh(b_i) m^{k-1}_i. \end{cases}$$

Solve the above equations, which yields that

$$A^k_{1,i} = -\frac{(\sigma_i \sinh(a_i) + d_i \cosh(b_i)) B^k_{1,i} + d_i (\sinh(a_i) + \sinh(b_i)) m^{k-1}_i}{\sigma_i \sinh(a_i) + d_i \cosh(a_i)}, \tag{3.22a}$$

$$A_{2,i}^k = \frac{\sigma_i(\cosh(a_i) - \cosh(b_i))B_{1,i}^k + (d_i\cosh(a_i) - \sigma_i\sinh(b_i))m_i^{k-1}}{\sigma_i\sinh(a_i) + d_i\cosh(a_i)}. \tag{3.22b}$$

Substituting (3.21) into the Eq. (3.22b), we get

$$\begin{aligned} m_i^k = A_{2,i}^k = &\frac{\sigma_i(\cosh(b_i) - \cosh(a_i))}{f_iE_i} f_{\alpha,i}^{k-1} \\ &+ \frac{\sigma_i d_i - r^{-1}\cosh(a_i)\sinh(b_i)}{f_iE_i} m_i^{k-1}. \end{aligned} \tag{3.23}$$

For $f_{\alpha,i}^k$, the following relation is obtained:

$$\begin{aligned} f_{\alpha,i}^k &= (1-\theta)f_{\alpha,i}^{k-1} + \theta\left(\dot\mu_{1,(i)}^k(\alpha) - d_i\mu_{1,(i)}^k(\alpha)\right) \\ &= (1-\theta)f_{\alpha,i}^{k-1} + \theta\Big[-\sigma_i\left(\cosh(b_i)B_{1,i}^k + \sinh(b_i)m_i^{k-1}\right) \\ &\qquad\qquad - d_i\left(\sinh(a_i)A_{1,i}^k + \cosh(a_i)A_{2,i}^k\right)\Big] \\ &= \left(1 - \theta d_i\frac{h_i - \sigma_i}{f_iE_i}\right) f_{\alpha,i}^{k-1} + r^{-1}\theta d_i\frac{\sinh(a_i) + \sinh(b_i)}{f_iE_i} m_i^{k-1}. \end{aligned} \tag{3.24}$$

Combining (3.23) and (3.24), we obtain the iterative relation as follows:

$$\begin{pmatrix} f_{\alpha,i}^k \\ m_i^k \end{pmatrix} = Q_{ND_1,i}\begin{pmatrix} f_{\alpha,i}^{k-1} \\ m_i^{k-1} \end{pmatrix},$$

where

$$\begin{aligned} Q_{ND_1,i} &:= \begin{pmatrix} Q_i^1 & Q_i^2 \\ Q_i^3 & Q_i^4 \end{pmatrix} \\ &:= \begin{pmatrix} \dfrac{f_iE_i + \theta d_i(\sigma_i - h_i)}{f_iE_i} & \dfrac{r^{-1}\theta d_i(\sinh(a_i) + \sinh(b_i))}{f_iE_i} \\ \dfrac{\sigma_i(\cosh(b_i) - \cosh(a_i))}{f_iE_i} & \dfrac{\sigma_i d_i - r^{-1}\cosh(a_i)\sinh(b_i)}{f_iE_i} \end{pmatrix}. \end{aligned} \tag{3.25}$$

Solving the characteristic polynomial of the iterative matrix $Q_{ND_1,i}$, we get its eigenvalues as follows:

$$\lambda_{ND_1,i} := \frac{(Q_i^1 + Q_i^4) \pm \sqrt{(Q_i^1 - Q_i^4)^2 + 4Q_i^2Q_i^3}}{2}. \tag{3.26}$$

Finally, we obtain the following convergence theorem for the ND_1 algorithm.

Theorem 3.2. *Algorithm* ND_1 (3.16)-(3.17) *converges if and only if*

$$\rho_{ND_1} := \max_{i=1,2,\dots,m} |\rho_{Q_{ND_1,i}}| < 1,$$

where $\rho_{Q_{ND_1,i}}$ *is the spectrum of the matrix* $Q_{ND_1,i}$.

Firstly, considering the time decomposition is symmetric $\alpha = T/2$, i.e., $a_i = b_i$, like the DN_1 algorithm, the following corollary can be derived.

Corollary 3.2. *If the time domain decomposition is symmetric $\alpha = T/2$, and the matrix A is not singular, the ND_1 algorithm* (3.16)-(3.17) *converges for any initial guess.*

Proof. If $a_i = b_i$, we obtain

$$\lambda_{ND_1,i} = \frac{(1+\theta)\sigma_i d_i + (1-\theta)d_i h_i \pm [(1-\theta)d_i(h_i - \sigma_i) + r^{-1}\sinh(\sigma_i T)]}{2d_i h_i + r^{-1}\sinh(\sigma_i T)} = \lambda_{DN_1,i}.$$

Therefore, if $a_i = b_i$, the ND_1 algorithm and the DN_1 algorithm have the same convergence factor $\rho_{ND_1} = \rho_{DN_1}$. Thus, we can conclude that if $d_i \neq 0$, namely the matrix A is not singular, $\rho_{ND_1} = \max_{i=1,2,\dots,m} |\rho_{Q_{ND_1,i}}| < 1$, which concludes the proof. ☐

Remark 3.3. If $d_i = 0$, according to (3.25), we obtain

$$Q_i^1|_{d_i=0} = 1, \quad Q_i^4|_{d_i=0} = \frac{-r^{-1}\sinh(b_i)\cosh(a_i)}{f_i E_i},$$
$$Q_i^2 Q_i^3|_{d_i=0} = 0, \quad (f_i E_i)|_{d_i=0} = r^{-1}\sinh(a_i)\cosh(b_i).$$

Then, we find

$$\lambda^{(1)}_{ND_1,i} = Q_i^1 = 1, \quad \lambda^{(2)}_{ND_1,i} = Q_i^4 = -\frac{\sinh(b_i)\cosh(a_i)}{\sinh(a_i)\cosh(b_i)} = -\frac{\tanh(b_i)}{\tanh(a_i)}.$$

Therefore, if $a_i \le b_i$, we get $|\lambda^{(2)}_{ND_1,i}| \ge 1$. Oppositely, the $|\lambda^{(2)}_{ND_1,i}| < 1$ can be obtained if $a_i > b_i$.

Finally, we can conclude that $\rho_{Q_{ND_1,i}}|_{d_i=0} \ge 1$ when $a_i \le b_i$, and if $a_i > b_i$, $\rho_{Q_{ND_1,i}}|_{d_i=0} \equiv 1$. Then the convergence behavior of the ND_1 algorithm for small eigenvalues is not good.

Remark 3.4. If d_i goes to infinity, we have $\sigma_i \sim_\infty d_i$, and then we can get $Q_i^2 Q_i^3 \to 0$, $Q_i^1 \to 1-\theta$, $Q_i^4 \to 0$. Substituting these into (3.26), we obtain

$$\lim_{d_i \to \infty} \rho_{Q_{ND_1,i}} = 1 - \theta.$$

In other words, the ND_1 algorithm is robust for high frequency convergence with relaxation, and when using $\theta = 1$, we obtain a good smoother, which is similar to the DN_1 algorithm.

3.2. Category II

We analyze Category II which uses the DN and ND techniques only for the state z to the model problem (2.6), and analyze their convergence.

3.2.1. DN_2

We apply the Dirichlet condition in I_1 and the Neumann condition in I_2 both to the state z. For the iteration index $k = 1, 2, \ldots$, we can get the DN_2 algorithm as follows:

$$\begin{cases} \begin{pmatrix} \dot{z}^k_{1,(i)} \\ \dot{\mu}^k_{1,(i)} \end{pmatrix} + \begin{pmatrix} d_i & -\dfrac{1}{r} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z^k_{1,(i)} \\ \mu^k_{1,(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{in } I_1, \\ z^k_{1,(i)}(0) = z^{k-1}_{2,(i)}(T), \\ z^k_{1,(i)}(\alpha) = f^{k-1}_{\alpha}, \end{cases}$$
$$\begin{cases} \begin{pmatrix} \dot{z}^k_{2,(i)} \\ \dot{\mu}^k_{2,(i)} \end{pmatrix} + \begin{pmatrix} d_i & -\dfrac{1}{r} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z^k_{2,(i)} \\ \mu^k_{2,(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{in } I_2, \\ \dot{z}^k_{2,(i)}(\alpha) = \dot{z}^k_{1,(i)}(\alpha), \\ \mu^k_{2,(i)}(T) = \mu^k_{1,(i)}(0), \end{cases} \tag{3.27}$$

and then update the value along the interface using

$$f^k_{\alpha,i} := (1-\theta) f^{k-1}_{\alpha,i} + \theta z^k_{2,(i)}(\alpha). \tag{3.28}$$

Remark 3.5. Different from Category I, the DN_2 algorithm loses the forward-backward structure in time for the equations in I_1 which only imposes the initial and final condition on $z_{1,(i)}$ but without explicit constraints on $\mu_{1,(i)}$. However, it can be seen from Eqs. (3.3) that this is simply an interpretation difference. For the equations in I_1 of (3.27), we can rewrite the final condition $z^k_{1,(i)}(\alpha) = f^{k-1}_{\alpha}$ as $\dot{\mu}^k_{1,(i)}(\alpha) - d_i \mu^k_{1,(i)}(\alpha) = f^{k-1}_{\alpha}(\alpha)$ based on the Eq. (3.3a), and then the update condition (3.28) can be defined as $f^k_{\alpha,i} := (1-\theta) f^{k-1}_{\alpha,i} + \theta(\dot{\mu}^k_{2,(i)}(\alpha) - d_i \mu^k_{2,(i)}(\alpha))$. This transformation restores the forward-backward structure of the equations in I_1.

Substituting the identities (3.3) into the systems (3.27) and (3.28), we obtain the systems as follows:

$$\begin{cases} \ddot{z}^k_{1,(i)} - \sigma_i^2 z^k_{1,(i)} = 0 \quad \text{in } I_1, \\ z^k_{1,(i)}(0) = z^{k-1}_{2,(i)}(T), \\ z^k_{1,(i)}(\alpha) = f^{k-1}_{\alpha,i}, \end{cases}$$
$$\begin{cases} \ddot{z}^k_{2,(i)} - \sigma_i^2 z^k_{2,(i)} = 0 \quad \text{in } I_2, \\ \dot{z}^k_{2,(i)}(\alpha) = \dot{z}^k_{1,(i)}(\alpha), \\ \dot{z}^k_{2,(i)}(T) + d_i z^k_{2,(i)}(T) = \dot{z}^k_{1,(i)}(0) + d_i z^k_{1,(i)}(0), \end{cases} \tag{3.29}$$

and then update the value along the interface using

$$f^k_{\alpha,i} := (1-\theta) f^{k-1}_{\alpha,i} + \theta z^k_{2,(i)}(\alpha).$$

Similar to the DN_1 algorithm, the solutions (3.5) are used to determine the coefficients $A_{1,i}^k, A_{2,i}^k, B_{1,i}^k, B_{2,i}^k$. And we still use the definition of m_i^k in (3.6) to simplify the following analysis. Substituting identities (3.5) into (3.6), we can get relations as follows:

$$z_{1,(i)}^{k+1}(0) = A_{2,i}^{k+1} = m_i^k, \quad z_{2,(i)}^k(T) = B_{2,i}^k = m_i^k.$$

Furthermore, according to the initial and final conditions in (3.29), the expressions of $A_{1,i}^k, B_{1,i}^k, B_{2,i}^k$ are obtained as follows:

$$\begin{aligned}
A_{1,i}^k &= \frac{f_{\alpha,i}^{k-1} - \cosh(a_i)m_i^{k-1}}{\sinh(a_i)}, \\
B_{1,i}^k &= -\frac{(d_i\cosh(a_i) + \sigma_i\sinh(b_i))A_{1,i}^k + d_i(\sinh(a_i) + \sinh(b_i))m_i^{k-1}}{\sigma_i\sinh(b_i) + d_i\cosh(b_i)}, \\
B_{2,i}^k &= \frac{\sigma_i(\cosh(b_i) - \cosh(a_i))A_{1,i}^k + (d_i\cosh(b_i) - \sigma_i\sinh(a_i))m_i^{k-1}}{\sigma_i\sinh(b_i) + d_i\cosh(b_i)}.
\end{aligned}$$

Thus, through the above expressions, we can get the expression of m_i^k with $f_{\alpha,i}^{k-1}$ and m_i^{k-1} as follows:

$$\begin{aligned}
m_i^k = B_{2,i}^k = &\frac{\sigma_i(\cosh(b_i) - \cosh(a_i))}{\sinh(a_i)F_i} f_{\alpha,i}^{k-1} \\
&+ \frac{\sigma_i + \cosh(b_i)(d_i\sinh(a_i) - \sigma_i\cosh(a_i))}{\sinh(a_i)F_i} m_i^{k-1}.
\end{aligned} \tag{3.30}$$

For $f_{\alpha,i}^k$, we obtain

$$\begin{aligned}
f_{\alpha,i}^k &= (1-\theta)f_{\alpha,i}^{k-1} + \theta z_{2,(i)}^k(\alpha) \\
&= (1-\theta)f_{\alpha,i}^{k-1} + \theta\left(\sinh(b_i)B_{1,i}^k + \cosh(b_i)B_{2,i}^k\right) \\
&= \left(1 - \theta\frac{h_i - \sigma_i}{\sinh(a_i)F_i}\right) f_{\alpha,i}^{k-1} + \frac{\theta}{\sinh(a_i)F_i}\big(f_i + d_i\sinh(a_i) - \sigma_i\cosh(a_i)\big)m_i^{k-1}.
\end{aligned} \tag{3.31}$$

Combining (3.30) and (3.31), the iterative relation is obtained as follows:

$$\begin{pmatrix} f_{\alpha,i}^k \\ m_i^k \end{pmatrix} = Q_{DN_2,i}\begin{pmatrix} f_{\alpha,i}^{k-1} \\ m_i^{k-1} \end{pmatrix},$$

where

$$\begin{aligned}
Q_{DN_2,i} &:= \begin{pmatrix} Q_i^1 & Q_i^2 \\ Q_i^3 & Q_i^4 \end{pmatrix} \\
&:= \begin{pmatrix} \dfrac{\sinh(a_i)F_i + \theta(\sigma_i - h_i)}{\sinh(a_i)F_i} & \dfrac{\theta(f_i + d_i\sinh(a_i) - \sigma_i\cosh(a_i))}{\sinh(a_i)F_i} \\ \dfrac{\sigma_i(\cosh(b_i) - \cosh(a_i))}{\sinh(a_i)F_i} & \dfrac{\sigma_i + \cosh(b_i)(d_i\sinh(a_i) - \sigma_i\cosh(a_i))}{\sinh(a_i)F_i} \end{pmatrix}.
\end{aligned} \tag{3.32}$$

Solving the characteristic polynomial of the iterative matrix $Q_{DN_2,i}$, we get its eigenvalues as follows:

$$\lambda_{DN_2,i} := \frac{(Q_i^1 + Q_i^4) \pm \sqrt{(Q_i^1 - Q_i^4)^2 + 4Q_i^2 Q_i^3}}{2}. \tag{3.33}$$

Finally, we get the following convergence theorem for the DN_2 algorithm.

Theorem 3.3. *Algorithm DN_2 (3.27)-(3.28) converges if and only if*

$$\rho_{DN_2} := \max_{i=1,2,\dots,m} |\rho_{Q_{DN_2,i}}| < 1, \tag{3.34}$$

where $\rho_{Q_{DN_2,i}}$ is the spectrum of the matrix $Q_{DN_2,i}$.

Similar to the DN_1 algorithm, we consider a few special cases for the convergence factor (3.34). Firstly, considering the time decomposition is symmetric $\alpha = T/2$, i.e., $a_i = b_i$, we obtain the corollary as follows.

Corollary 3.3. *If the time domain decomposition is symmetric, i.e., $\alpha = T/2$, there is $\rho_{DN_2}(\theta) = \rho_{DN_2}(1-\theta)$. And if the matrix A is not singular and $\theta \neq 1$, the DN_2 algorithm* (3.27)-(3.28) *converges for any initial guess.*

Proof. If $a_i = b_i$, we can find $Q_i^2 Q_i^3 = 0$, and then

$$\lambda_{DN_2,i} = \frac{(Q_i^1 + Q_i^4) \pm (Q_i^1 - Q_i^4)}{2}.$$

Thus, we obtain

$$\lambda_{DN_2,i}^{(1)} := Q_i^1 = 1 + \theta \frac{\sigma_i - h_i}{\sinh(a_i) F_i}.$$

Derive the above formula about θ,

$$\left(\lambda_{DN_2,i}^{(1)}\right)'(\theta) = \frac{\sigma_i - h_i}{\sinh(a_i) F_i}.$$

Using (3.15), we get $(\lambda_{DN_2,i}^{(1)})' < 0$. Then, the extreme points of $\lambda_{DN_2,i}^{(1)}$ can only be obtained at $\theta = 0$ and $\theta = 1$. Furthermore, if replacing θ with $1-\theta$, we find $\lambda_{DN_2,i}^{(1)}(1-\theta) = -\lambda_{DN_2,i}^{(1)}(\theta)$. Thus, we obtain

$$\left|\lambda_{DN_2,i}^{(1)}\right|_{\max} = \left|\lambda_{DN_2,i}^{(1)}(\theta = 0)\right| = \left|\lambda_{DN_2,i}^{(1)}(\theta = 1)\right| = 1.$$

For the second kind of eigenvalues,

$$\lambda_{DN_2,i}^{(2)} := Q_i^4 = \frac{d_i \sinh(\sigma_i T) - 2\sigma_i \sinh^2(a_i)}{d_i \sinh(\sigma_i T) + 2\sigma_i \sinh^2(a_i)}.$$

According to the above formula, we get

$$\lambda^{(2)}_{DN_2,i}(1-\theta) = \lambda^{(2)}_{DN_2,i}(\theta).$$

Taking the absolute value of $\lambda^{(2)}_{DN_2,i}$, we obtain

$$\begin{aligned}\left|\lambda^{(2)}_{DN_2,i}\right| &= \left|\frac{d_i\sinh(\sigma_i T) - 2\sigma_i\sinh^2(a_i)}{d_i\sinh(\sigma_i T) + 2\sigma_i\sinh^2(a_i)}\right| \\ &\le 1. \quad \text{(If and only if } d_i = 0, \text{ take the equal sign.)}\end{aligned}$$

In conclusion, combining the analysis of the above two kinds of eigenvalues $\lambda^{(1)}_{DN_2,i}$ and $\lambda^{(2)}_{DN_2,i}$, we get $\rho_{DN_2}(\theta) = \rho_{DN_2}(1-\theta)$. Also, if $\theta \neq 0$, $\theta \neq 1$ and $d_i \neq 0$, we obtain $\rho_{DN_2} = \max_{i=1,2,\ldots,m}|\rho_{Q_{DN_2,i}}| = \max_{i=1,2,\ldots,m}|\lambda_{DN_2,i}| < 1$, which concludes the proof. ☐

Remark 3.6. If $d_i = 0$, based on the expression of (3.32), we obtain

$$\begin{aligned} Q_i^1|_{d_i=0} &= \frac{\sinh(a_i)\sinh(b_i) + \theta(1-\cosh(\sigma_i T))}{\sinh(a_i)\sinh(b_i)}, \\ Q_i^2Q_i^3|_{d_i=0} &= \theta\left(\frac{\cosh(a_i)-\cosh(b_i)}{\sinh(a_i)\sinh(b_i)}\right)^2, \\ Q_i^4|_{d_i=0} &= \frac{1-\cosh(a_i)\cosh(b_i)}{\sinh(a_i)\sinh(b_i)}. \end{aligned}$$

Substituting the above formula into the eigenvalues $\lambda_{DN_2,i}$ of $Q_{DN_2,i}$ in (3.33), it yields

$$\lambda^{(2)}_{DN_2,i} := \frac{(Q_i^1+Q_i^4) - \sqrt{(Q_i^1-Q_i^4)^2 + 4Q_i^2Q_i^3}}{2} \le -1,$$

and if and only if $a_i = b_i$, take the equal sign.

Therefore, we can conclude that $\rho_{Q_{DN_2,i}}|_{d_i=0} \ge 1$, and combing the Corollary 3.3, we know that $\rho_{Q_{DN_2,i}}|_{d_i=0} = 1$ if and only if $a_i = b_i$. Then when the eigenvalues d_i are small, the DN_2 algorithm exhibits poor convergence and may even diverge for asymmetric decomposition.

Remark 3.7. If d_i goes to infinity, we can get $Q_i^2Q_i^3 \to 0$, $Q_i^1 \to 1-2\theta$, $Q_i^4 \to 0$, thus we obtain

$$\lim_{d_i\to\infty}\rho_{Q_{DN_2,i}} = |1-2\theta|.$$

In other words, when the relaxation parameter $\theta = 0.5$, the convergence performance of the DN_2 algorithm is good for large eigenvalues d_i.

3.2.2. ND_2

For the iteration index $k = 1, 2, \ldots$, we exchange the Dirichlet and Neumann conditions at the interface $t = \alpha$ based on Table 1 as follows:

$$\begin{cases} \begin{pmatrix} \dot{z}_{1,(i)}^k \\ \dot{\mu}_{1,(i)}^k \end{pmatrix} + \begin{pmatrix} d_i & -\dfrac{1}{r} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{1,(i)}^k \\ \mu_{1,(i)}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{in } I_1, \\ z_{1,(i)}^k(0) = z_{2,(i)}^{k-1}(T), \\ \dot{z}_{1,(i)}^k(\alpha) = f_\alpha^{k-1}, \end{cases}$$
$$\begin{cases} \begin{pmatrix} \dot{z}_{2,(i)}^k \\ \dot{\mu}_{2,(i)}^k \end{pmatrix} + \begin{pmatrix} d_i & -\dfrac{1}{r} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{2,(i)}^k \\ \mu_{2,(i)}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{in } I_2, \\ z_{2,(i)}^k(\alpha) = z_{1,(i)}^k(\alpha), \\ \mu_{2,(i)}^k(T) = \mu_{1,(i)}^k(0), \end{cases} \tag{3.35}$$

and then update the value along the interface using

$$f_{\alpha,i}^k := (1-\theta) f_{\alpha,i}^{k-1} + \theta \dot{z}_{2,(i)}^k(\alpha). \tag{3.36}$$

Like the DN_2 algorithm, the equations in I_1 of the ND_2 algorithm do not have forward-backward structure. Using the method similar to Remark 3.5, we can explain this phenomenon. To analyze the convergence, substituting the identities (3.3) into the systems (3.35) and (3.36), we obtain the systems as follows:

$$\begin{cases} \ddot{z}_{1,(i)}^k - \sigma_i^2 z_{1,(i)}^k = 0 \quad \text{in } I_1, \\ z_{1,(i)}^k(0) = z_{2,(i)}^{k-1}(T), \\ \dot{z}_{1,(i)}^k(\alpha) = f_{\alpha,i}^{k-1}, \end{cases}$$
$$\begin{cases} \ddot{z}_{2,(i)}^k - \sigma_i^2 z_{2,(i)}^k = 0 \quad \text{in } I_2, \\ z_{2,(i)}^k(\alpha) = z_{1,(i)}^k(\alpha), \\ \dot{z}_{2,(i)}^k(T) + d_i z_{2,(i)}^k(T) = \dot{z}_{1,(i)}^k(0) + d_i z_{1,(i)}^k(0), \end{cases} \tag{3.37}$$

and then update the value along the interface using

$$f_{\alpha,i}^k := (1-\theta) f_{\alpha,i}^{k-1} + \theta \dot{z}_{2,(i)}^k(\alpha).$$

Similar to the DN_1 algorithm, the solutions (3.5) are used to determine the coefficients $A_{1,i}^k, A_{2,i}^k, B_{1,i}^k, B_{2,i}^k$. And we still use the definition of m_i^k (3.6) to simplify the following analysis. Substituting identities (3.5) into (3.6), we obtain the relations,

$$z_{1,(i)}^{k+1}(0) = A_{2,i}^{k+1} = m_i^k, \quad z_{2,(i)}^k(T) = B_{2,i}^k = m_i^k.$$

Furthermore, according to the initial and final conditions in (3.37), we get the expressions of $A_{1,i}^k$, $B_{1,i}^k$, $B_{2,i}^k$ as follows:

$$A_{1,i}^k = \frac{f_{\alpha,i}^{k-1} - \sigma_i \sinh(a_i) m_i^{k-1}}{\sigma_i \cosh(a_i)},$$

$$B_{1,i}^k = \frac{(d_i\sinh(a_i) - \sigma_i\cosh(b_i))A_{1,i}^k + d_i(\cosh(a_i) - \cosh(b_i))m_i^{k-1}}{\sigma_i\cosh(b_i) + d_i\sinh(b_i)},$$
$$B_{2,i}^k = \frac{\sigma_i(\sinh(a_i) + \sinh(b_i))A_{1,i}^k + (\sigma_i\cosh(a_i) + d_i\sinh(b_i))m_i^{k-1}}{\sigma_i\cosh(b_i) + d_i\sinh(b_i)}.$$

Thus, through the above expressions, we obtain the expression of m_i^k with $f_{\alpha,i}^{k-1}$ and m_i^{k-1} as follows:

$$\begin{aligned} m_i^k = B_{2,i}^k = &\frac{\sinh(a_i) + \sinh(b_i)}{\cosh(a_i)f_i} f_{\alpha,i}^{k-1} \\ &+ \frac{\sigma_i + \sinh(b_i)(d_i\cosh(a_i) - \sigma_i\sinh(a_i))}{\cosh(a_i)f_i} m_i^{k-1}. \end{aligned} \tag{3.38}$$

For $f_{\alpha,i}^k$, the following relation is obtained:

$$\begin{aligned} f_{\alpha,i}^k &= (1-\theta)f_{\alpha,i}^{k-1} + \theta\dot{z}_{2,(i)}^k(\alpha) \\ &= (1-\theta)f_{\alpha,i}^{k-1} - \theta\sigma_i\left(\cosh(b_i)B_{1,i}^k + \sinh(b_i)B_{2,i}^k\right) \\ &= \left(1 - \theta\frac{h_i - \sigma_i}{\cosh(a_i)f_i}\right) f_{\alpha,i}^{k-1} + \frac{\theta\sigma_i}{\cosh(a_i)f_i}\big(d_i\cosh(a_i) - \sigma_i\sinh(a_i) - F_i\big)m_i^{k-1}. \end{aligned} \tag{3.39}$$

Combining (3.38) and (3.39), we can get the iterative relation as follows:

$$\begin{pmatrix} f_{\alpha,i}^k \\ m_i^k \end{pmatrix} = Q_{ND_2,i}\begin{pmatrix} f_{\alpha,i}^{k-1} \\ m_i^{k-1} \end{pmatrix},$$

where

$$\begin{aligned} Q_{ND_2,i} &:= \begin{pmatrix} Q_i^1 & Q_i^2 \\ Q_i^3 & Q_i^4 \end{pmatrix} \\ &:= \begin{pmatrix} \dfrac{\cosh(a_i)f_i + \theta(\sigma_i - h_i)}{\cosh(a_i)f_i} & \dfrac{\theta\sigma_i(d_i\cosh(a_i) - \sigma_i\sinh(a_i) - F_i)}{\cosh(a_i)f_i} \\ \dfrac{\sinh(a_i) + \sinh(b_i)}{\cosh(a_i)f_i} & \dfrac{\sigma_i + \sinh(b_i)(d_i\cosh(a_i) - \sigma_i\sinh(a_i))}{\cosh(a_i)f_i} \end{pmatrix}. \end{aligned}$$

Finally, we obtain the following convergence theorem for our new algorithm.

Theorem 3.4. *Algorithm ND_2 (3.35)-(3.36) converges if and only if*

$$\rho_{ND_2} := \max_{i=1,2,\ldots,m} |\rho_{Q_{ND_2,i}}| < 1,$$

where $\rho_{Q_{ND_2,i}}$ is the spectrum of the matrix $Q_{ND_2,i}$.

Remark 3.8. If d_i goes to infinity, we can get $Q_i^2Q_i^3 \to 0$, $Q_i^1 \to 1-2\theta$, $Q_i^4 \to 0$, and then we obtain

$$\lim_{d_i\to\infty} \rho_{Q_{ND_2,i}} = |1-2\theta|.$$

In other words, when the relaxation parameter $\theta = 0.5$, the convergence performance of the ND_2 algorithm is good for large eigenvalues d_i.

3.3. Category III

We study Category III which applies the DN and ND techniques only to the state μ to the model problem (2.6), and analyze their convergence.

3.3.1. DN_3

We use the Dirichlet condition in I_1 and the Neumann condition in I_2 both on the state μ. For the iteration index $k=1,2,\dots,$ we can get the DN_3 algorithm as follows:

$$\begin{aligned}
&\begin{cases}
\begin{pmatrix}\dot{z}^k_{1,(i)}\\ \dot{\mu}^k_{1,(i)}\end{pmatrix}+\begin{pmatrix}d_i & -\dfrac{1}{r}\\ -1 & -d_i\end{pmatrix}\begin{pmatrix}z^k_{1,(i)}\\ \mu^k_{1,(i)}\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix} \quad \text{in } I_1,\\
z^k_{1,(i)}(0)=z^k_{2,(i)}(T),\\
\mu^k_{1,(i)}(\alpha)=\mu^k_{2,(i)}(\alpha),
\end{cases}\\
&\begin{cases}
\begin{pmatrix}\dot{z}^k_{2,(i)}\\ \dot{\mu}^k_{2,(i)}\end{pmatrix}+\begin{pmatrix}d_i & -\dfrac{1}{r}\\ -1 & -d_i\end{pmatrix}\begin{pmatrix}z^k_{2,(i)}\\ \mu^k_{2,(i)}\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix} \quad \text{in } I_2,\\
\dot{\mu}^k_{2,(i)}(\alpha)=f^{k-1}_{\alpha},\\
\mu^k_{2,(i)}(T)=\mu^{k-1}_{1,(i)}(0),
\end{cases}
\end{aligned} \tag{3.40}$$

and then update the value along the interface using

$$f^k_{\alpha,i}:=(1-\theta)f^{k-1}_{\alpha,i}+\theta\dot{\mu}^k_{1,(i)}(\alpha). \tag{3.41}$$

Like the DN_2 algorithm, the equations in I_2 of the DN_3 algorithm appear to lack the forward-backward structure, using the technique similar to Remark 3.5, this problem can be explained. Then to analyze the convergence, according to (3.3), we can get the systems as follows:

$$\begin{aligned}
&\begin{cases}
\ddot{\mu}^k_{1,(i)}-\sigma_i^2\mu^k_{1,(i)}=0 \quad \text{in } I_1,\\
\dot{\mu}^k_{1,(i)}(0)-d_i\mu^k_{1,(i)}(0)=\dot{\mu}^k_{2,(i)}(T)-d_i\mu^k_{2,(i)}(T),\\
\mu^k_{1,(i)}(\alpha)=\mu^k_{2,(i)}(\alpha),
\end{cases}\\
&\begin{cases}
\ddot{\mu}^k_{2,(i)}-\sigma_i^2\mu^k_{2,(i)}=0 \quad \text{in } I_2,\\
\dot{\mu}^k_{2,(i)}(\alpha)=f^{k-1}_{\alpha,i},\\
\mu^k_{2,(i)}(T)=\mu^{k-1}_{1,(i)}(0),
\end{cases}
\end{aligned} \tag{3.42}$$

and then update the value along the interface using

$$f^k_{\alpha,i}:=(1-\theta)f^{k-1}_{\alpha,i}+\theta\dot{\mu}^k_{1,(i)}(\alpha).$$

Similar to the ND_1 algorithm, the solutions (3.19) are used to determine the coefficients $A_{1,i}^k, A_{2,i}^k, B_{1,i}^k, B_{2,i}^k$. And we still use the definition of m_i^k in (3.20) to simplify the following analysis. Substituting identities (3.19) into (3.20), we obtain relations as follows:

$$\mu_{1,(i)}^k(0) = A_{2,i}^k = m_i^k, \quad \mu_{2,(i)}^{k+1}(T) = B_{2,i}^{k+1} = m_i^k.$$

Furthermore, according to the initial and final conditions in (3.42), the expressions of $A_{1,i}^k, A_{2,i}^k, B_{1,i}^k$ are obtained as follows:

$$\begin{aligned}
A_{1,i}^k &= \frac{(d_i \sinh(b_i) - \sigma_i \cosh(a_i))B_{1,i}^k + d_i(\cosh(b_i) - \cosh(a_i))m_i^{k-1}}{\sigma_i \cosh(a_i) + d_i \sinh(a_i)}, \\
A_{2,i}^k &= \frac{\sigma_i(\sinh(a_i) + \sinh(b_i))B_{1,i}^k + (\sigma_i \cosh(b_i) + d_i \sinh(a_i))m_i^{k-1}}{\sigma_i \cosh(a_i) + d_i \sinh(a_i)}, \\
B_{1,i}^k &= -\frac{f_{\alpha,i}^{k-1} + \sigma_i \sinh(b_i) m_i^{k-1}}{\sigma_i \cosh(b_i)}.
\end{aligned}$$

Thus, through the above expressions, we get the expression of m_i^k with $f_{\alpha,i}^{k-1}$ and m_i^{k-1} as follows:

$$\begin{aligned}
m_i^k = A_{2,i}^k = &-\frac{\sinh(a_i) + \sinh(b_i)}{\cosh(b_i)e_i} f_{\alpha,i}^{k-1} \\
&+ \frac{\sigma_i + \sinh(a_i)(d_i \cosh(b_i) - \sigma_i \sinh(b_i))}{\cosh(b_i)e_i} m_i^{k-1}.
\end{aligned} \tag{3.43}$$

For $f_{\alpha,i}^k$, we obtain

$$\begin{aligned}
f_{\alpha,i}^k &= (1-\theta) f_{\alpha,i}^{k-1} + \theta \dot{\mu}_{1,(i)}^k(\alpha) \qquad (3.44) \\
&= (1-\theta) f_{\alpha,i}^{k-1} + \theta \sigma_i \left(\cosh(a_i) A_{1,i}^k + \sinh(a_i) A_{2,i}^k\right) \\
&= \left(1 - \theta \frac{h_i - \sigma_i}{\cosh(b_i)e_i}\right) f_{\alpha,i}^{k-1} + \frac{\theta \sigma_i}{\cosh(b_i)e_i}\big(\sigma_i \sinh(b_i) - d_i \cosh(b_i) + E_i\big) m_i^{k-1}.
\end{aligned}$$

Combining (3.43) and (3.44), we obtain the iterative relation as follows:

$$\begin{pmatrix} f_{\alpha,i}^k \\ m_i^k \end{pmatrix} = Q_{DN_3,i} \begin{pmatrix} f_{\alpha,i}^{k-1} \\ m_i^{k-1} \end{pmatrix},$$

where

$$\begin{aligned}
Q_{DN_3,i} &:= \begin{pmatrix} Q_i^1 & Q_i^2 \\ Q_i^3 & Q_i^4 \end{pmatrix} \\
&:= \begin{pmatrix} \dfrac{\cosh(b_i)e_i + \theta(\sigma_i - h_i)}{\cosh(b_i)e_i} & \dfrac{\theta \sigma_i(\sigma_i \sinh(b_i) - d_i \cosh(b_i) + E_i)}{\cosh(b_i)e_i} \\ -\dfrac{\sinh(a_i) + \sinh(b_i)}{\cosh(b_i)e_i} & \dfrac{\sigma_i + \sinh(a_i)(d_i \cosh(b_i) - \sigma_i \sinh(b_i))}{\cosh(b_i)e_i} \end{pmatrix}.
\end{aligned}$$

Finally, we get the following convergence theorem of the new DN_3 algorithm.

Theorem 3.5. *Algorithm* DN_3 (3.40)-(3.41) *converges if and only if*

$$\rho_{DN_3} := \max_{i=1,2,\ldots,m} |\rho_{Q_{DN_3,i}}| < 1,$$

where $\rho_{Q_{DN_3,i}}$ *is the spectrum of the matrix* $Q_{DN_3,i}$.

Remark 3.9. If the time decomposition is symmetric $\alpha = T/2$, i.e., $a_i = b_i$, substituting $a_i = b_i$ into the expressions of $Q_{ND_3,i}$ and $Q_{ND_2,i}$, we obtain the characteristic polynomials of $Q_{DN_3,i}$ and $Q_{ND_2,i}$ are equal. Therefore, if $a_i = b_i$, the DN_3 algorithm and the ND_2 algorithm have the same convergence factor $\rho_{DN_3} = \rho_{ND_2}$.

Remark 3.10. If d_i goes to infinity, we can get $Q_i^2 Q_i^3 \to 0$, $Q_i^1 \to 1 - 2\theta$, $Q_i^4 \to 0$, and then we obtain

$$\lim_{d_i \to \infty} \rho_{Q_{DN_3,i}} = |1 - 2\theta|.$$

Thus, when the relaxation parameter $\theta = 0.5$, the convergence performance of the DN_3 algorithm is good for large eigenvalues d_i.

3.3.2. ND_3

For the iteration index $k = 1, 2, \ldots$, we can get the ND_3 algorithm based on Table 1 as follows:

$$\begin{cases} \begin{pmatrix} \dot{z}_{1,(i)}^k \\ \dot{\mu}_{1,(i)}^k \end{pmatrix} + \begin{pmatrix} d_i & -\dfrac{1}{r} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{1,(i)}^k \\ \mu_{1,(i)}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{in } I_1, \\ z_{1,(i)}^k(0) = z_{2,(i)}^k(T), \\ \dot{\mu}_{1,(i)}^k(\alpha) = \dot{\mu}_{2,(i)}^k(\alpha), \end{cases}$$
$$\begin{cases} \begin{pmatrix} \dot{z}_{2,(i)}^k \\ \dot{\mu}_{2,(i)}^k \end{pmatrix} + \begin{pmatrix} d_i & -\dfrac{1}{r} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{2,(i)}^k \\ \mu_{2,(i)}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \text{in } I_2, \\ \mu_{2,(i)}^k(\alpha) = f_{\alpha,i}^{k-1}, \\ \mu_{2,(i)}^k(T) = \mu_{1,(i)}^{k-1}(0), \end{cases} \tag{3.45}$$

and then update the value along the interface using

$$f_{\alpha,i}^k := (1-\theta) f_{\alpha,i}^{k-1} + \theta \mu_{1,(i)}^k(\alpha). \tag{3.46}$$

Like the DN_2 algorithm, the equations in I_2 of the ND_3 algorithm seem to lose the forward-backward structure, this question can be explained using the method similar to Remark 3.5. Then for the analysis of convergence, based on (3.3), we obtain the systems as follows:

$$\begin{cases} \ddot{\mu}_{1,(i)}^k - \sigma_i^2 \mu_{1,(i)}^k = 0 \quad \text{in } I_1, \\ \dot{\mu}_{1,(i)}^k(0) - d_i \mu_{1,(i)}^k(0) = \dot{\mu}_{2,(i)}^k(T) - d_i \mu_{2,(i)}^k(T), \\ \dot{\mu}_{1,(i)}^k(\alpha) = \dot{\mu}_{2,(i)}^k(\alpha), \end{cases}$$

$$\begin{cases}\ddot{\mu}^k_{2,(i)} - \sigma_i^2 \mu^k_{2,(i)} = 0 \quad \text{in } I_2,\\ \mu^k_{2,(i)}(\alpha) = f^{k-1}_{\alpha,i},\\ \mu^k_{2,(i)}(T) = \mu^{k-1}_{1,(i)}(0),\end{cases}$$

and then update the value along the interface using

$$f^k_{\alpha,i} := (1-\theta) f^{k-1}_{\alpha,i} + \theta \mu^k_{1,(i)}(\alpha).$$

Similar to the ND_1 algorithm, the solutions (3.19) are used to determine the coefficients $A^k_{1,i}, A^k_{2,i}, B^k_{1,i}, B^k_{2,i}$. And we still use the definition of m^k_i (3.20) to simplify the following analysis. Substituting identities (3.19) into (3.20), we can get relations as follows:

$$\mu^k_{1,(i)}(0) = A^k_{2,i} = m^k_i, \quad \mu^{k+1}_{2,(i)}(T) = B^{k+1}_{2,i} = m^k_i.$$

Furthermore, according to the initial and final conditions in (3.42), the expressions of $A^k_{1,i}, A^k_{2,i}, B^k_{1,i}$ are obtained as follows:

$$\begin{aligned}
A^k_{1,i} &= -\frac{(\sigma_i \sinh(a_i) + d_i \cosh(b_i)) B^k_{1,i} + d_i(\sinh(a_i) + \sinh(b_i)) m^{k-1}_i}{\sigma_i \sinh(a_i) + d_i \cosh(a_i)},\\
A^k_{2,i} &= \frac{\sigma_i(\cosh(a_i) - \cosh(b_i)) B^k_{1,i} + (d_i \cosh(a_i) - \sigma_i \sinh(b_i)) m^{k-1}_i}{\sigma_i \sinh(a_i) + d_i \cosh(a_i)},\\
B^k_{1,i} &= \frac{f^{k-1}_{\alpha,i} - \cosh(b_i) m^{k-1}_i}{\sinh(b_i)}.
\end{aligned}$$

Thus, through the above expressions, we get the expression of m^k_i with $f^{k-1}_{\alpha,i}$ and m^{k-1}_i as follows:

$$\begin{aligned}
m^k_i = A^k_{2,i} = &\frac{\sigma_i(\cosh(a_i) - \cosh(b_i))}{\sinh(b_i) E_i} f^{k-1}_{\alpha,i}\\
&+ \frac{\sigma_i + \cosh(a_i)(d_i \sinh(b_i) - \sigma_i \cosh(b_i))}{\sinh(b_i) E_i} m^{k-1}_i.
\end{aligned} \tag{3.47}$$

For $f^k_{\alpha,i}$, the following relation is obtained:

$$\begin{aligned}
f^k_{\alpha,i} &= (1-\theta) f^{k-1}_{\alpha,i} + \theta \mu^k_{1,(i)}(\alpha)\\
&= (1-\theta) f^{k-1}_{\alpha,i} + \theta \left(\sinh(a_i) A^k_{1,i} + \cosh(a_i) A^k_{2,i}\right)\\
&= \left(1 - \theta \frac{h_i - \sigma_i}{\sinh(b_i) E_i}\right) f^{k-1}_{\alpha,i} + \frac{\theta}{\sinh(b_i) E_i}\big(d_i \sinh(b_i) - \sigma_i \cosh(b_i) + e_i\big) m^{k-1}_i.
\end{aligned} \tag{3.48}$$

Combining (3.47) and (3.48), we get the iterative relation as follows:

$$\begin{pmatrix} f^k_{\alpha,i} \\ m^k_i \end{pmatrix} = Q_{ND_3,i} \begin{pmatrix} f^{k-1}_{\alpha,i} \\ m^{k-1}_i \end{pmatrix},$$

where

$$Q_{ND_3,i} := \begin{pmatrix} Q_i^1 & Q_i^2 \\ Q_i^3 & Q_i^4 \end{pmatrix} \tag{3.49}$$
$$:= \begin{pmatrix} \dfrac{\sinh(b_i)E_i + \theta(\sigma_i - h_i)}{\sinh(b_i)E_i} & \dfrac{\theta(d_i \sinh(b_i) - \sigma_i \cosh(b_i) + e_i)}{\sinh(b_i)E_i} \\ \dfrac{\sigma_i(\cosh(a_i) - \cosh(b_i))}{\sinh(b_i)E_i} & \dfrac{\sigma_i + \cosh(a_i)(d_i \sinh(b_i) - \sigma_i \cosh(b_i))}{\sinh(b_i)E_i} \end{pmatrix}.$$

Finally, we obtain the following convergence theorem for the ND_3 algorithm.

Theorem 3.6. *Algorithm ND_3 (3.45)-(3.46) converges if and only if*

$$\rho_{ND_3} := \max_{i=1,2,\ldots,m} |\rho_{Q_{ND_3,i}}| < 1,$$

where $\rho_{Q_{ND_3,i}}$ is the spectrum of the matrix $Q_{ND_3,i}$.

If the time decomposition is symmetric, i.e., $a_i = b_i$, the ND_3 algorithm and the DN_2 algorithm have the same convergence factor $\rho_{ND_3} = \rho_{DN_2}$. Then for the ND_3 algorithm, we can obtain the following corollary, just like the DN_2 algorithm.

Corollary 3.4. *If the time domain decomposition is symmetric, i.e., $\alpha = T/2$, there is $\rho_{ND_3}(\theta) = \rho_{ND_3}(1-\theta)$. And if the matrix A is not singular and $\theta \neq 1$, the ND_3 algorithm (3.45)-(3.46) converges for any initial guess.*

Proof. Substituting $a_i = b_i$ into the expression of $Q_{ND_3,i}$, we obtain $Q_{ND_3,i} = Q_{DN_2,i}$. Therefore, if $a_i = b_i$, the ND_3 algorithm and the DN_2 algorithm have the same convergence factor $\rho_{ND_3} = \rho_{DN_2}$, which concludes the proof. □

Remark 3.11. If $d_i = 0$, based on the expression of (3.49), we obtain

$$\begin{aligned} Q_i^1|_{d_i=0} &= \frac{\sinh(a_i)\sinh(b_i) + \theta(1 - \cosh(\sigma_i T))}{\sinh(a_i)\sinh(b_i)}, \\ Q_i^2 Q_i^3|_{d_i=0} &= \theta\left(\frac{\cosh(a_i) - \cosh(b_i)}{\sinh(a_i)\sinh(b_i)}\right)^2, \\ Q_i^4|_{d_i=0} &= \frac{1 - \cosh(a_i)\cosh(b_i)}{\sinh(a_i)\sinh(b_i)}. \end{aligned}$$

We find that the convergence factors of DN_2 and ND_3 are the same for $d_i = 0$. Thus, we conclude that $\rho_{Q_{ND_3,i}}|_{d_i=0} \geq 1$, and $\rho_{Q_{ND_3,i}}|_{d_i=0} = 1$ if and only if $a_i = b_i$. Then the convergence behavior of the ND_3 algorithm is poor when the eigenvalues are small, and when the decomposition is asymmetric, the ND_3 algorithm may even diverge.

Remark 3.12. If d_i goes to infinity, we can get $Q_i^2 Q_i^3 \to 0$, $Q_i^1 \to 1-2\theta$, $Q_i^4 \to 0$, and then we obtain

$$\lim_{d_i \to \infty} \rho_{Q_{ND_3,i}} = |1-2\theta|.$$

Thus, when the relaxation parameter $\theta = 0.5$, the convergence performance of the ND_3 algorithm is good for large eigenvalues d_i.

Remark 3.13. In Section 3, we derive the convergence factors of all proposed algorithms and analyze their convergence in some special cases. In this remark, we will summarize and compare the obtained convergence results for all proposed algorithms, and the details are as follows.

Firstly, if the time domain decomposition is symmetric $\alpha = T/2$, i.e., $a_i = b_i$, we have the following results:

- Algorithms DN_1 and ND_1, algorithms DN_2 and ND_3, and algorithms DN_3 and ND_2 have the same convergence factors respectively, i.e., $\rho_{DN_1} = \rho_{ND_1}$, $\rho_{DN_2} = \rho_{ND_3}$, $\rho_{DN_3} = \rho_{ND_2}$.
- For algorithms DN_1 and ND_1, if the matrix A is not singular, the algorithms converge for any initial guess.
- For algorithms DN_2 and ND_3, there is $\rho_{DN_2}(\theta) = \rho_{DN_2}(1-\theta)$. And if the matrix A is not singular and $\theta \neq 1$, the DN_2 and ND_3 algorithms converge for any initial guess.

Secondly, if eigenvalue $d_i = 0$, for algorithms DN_1, ND_1, DN_2, and ND_3, it can be concluded that $\rho \geq 1$. And if eigenvalue $d_i \to \infty$, for the convergence factors of all proposed algorithms, we obtain the following conclusions:

- For algorithms DN_1 and ND_1, as $d_i \to \infty$, we obtain $\rho \to 1-\theta$.
- For algorithms DN_2, ND_2, DN_3, and ND_3, as $d_i \to \infty$, we obtain $\rho \to |1-2\theta|$.

4. Numerical experiments

Numerical experiments in this section are performed to measure the actual convergence of six new time domain decomposition algorithms for the model problem (2.1)-(2.2). To well illustrate and compare the convergence among the algorithms, we divide the time interval $I = (0,1)$ into two non-overlapping subdomains $I_1 = (0,\alpha)$, $I_2 = (\alpha, 1)$ with the interface α, and select the parameter $r = 0.1$. We will study the performance of convergence factors for six proposed algorithms by considering the convergence factor ρ as a function of the eigenvalues d.

4.1. Convergence factor for symmetric and asymmetric decompositions

We show the convergence factors of six proposed algorithms for the symmetric and asymmetric time decompositions. Firstly, Fig. 1 shows the convergence factors of six proposed algorithms in the case of symmetric time decomposition, and the relaxation parameter $\theta = 1$ on the left while $\theta = 0.5$ on the right. From Fig. 1, we find that $\rho_{DN_1} = \rho_{ND_1}$, $\rho_{DN_2} = \rho_{ND_3}$ and $\rho_{DN_3} = \rho_{ND_2}$, which is mutually confirmed by the theoretical analysis in Section 3. For $\theta = 1$, we find that the convergence of the DN_1 and ND_1 algorithms is obviously better than the other four algorithms, and the convergence factors $\rho \to 0$ as $d \to \infty$. Meanwhile, the convergence factors of the DN_2 and ND_3 algorithms are equal to 1, and then DN_2 and ND_3 don't converge. In the case of $\theta = 0.5$, all six proposed algorithms have good convergence behavior, and the ND_2 and DN_3 algorithms are better than others.

For the asymmetric decomposition, we choose $\theta = 1$ and $\theta = 0.5$ as the relaxation parameters, and compare the numerical performance of convergence factors of each algorithm. In the case of $\theta = 1$, and the time domain decomposition is asymmetric with $\alpha = 0.3$ and $\alpha = 0.7$, the convergence of six proposed algorithms is shown in Fig. 2. We find the DN_2 and ND_3 algorithms diverge in both cases, and the convergence of ND_2 and DN_3 is poor. Furthermore, for $\alpha = 0.3$, the DN_1 algorithm is slightly better than ND_1, and this performance is opposite for $\alpha = 0.7$.

In the case of $\theta = 0.5$, we know that the convergence factors of Category II and Category III go to zero if eigenvalues $d \to \infty$ based on our convergence analysis. Then we investigate the convergence behavior of all six proposed algorithms with $\theta = 0.5$ in Fig. 3. Both for $\alpha = 0.3$ and $\alpha = 0.7$, the convergence factors ρ of DN_2 and ND_3 are larger than 1 if $d \to 0$. Moreover, the convergence performance of ND_2 and DN_3 is better than the other four algorithms in general. Meanwhile, in the case of $\alpha = 0.3$, the DN_3 algorithm is slightly better than the ND_2 algorithm, and this situation is reversed for $\alpha = 0.7$.

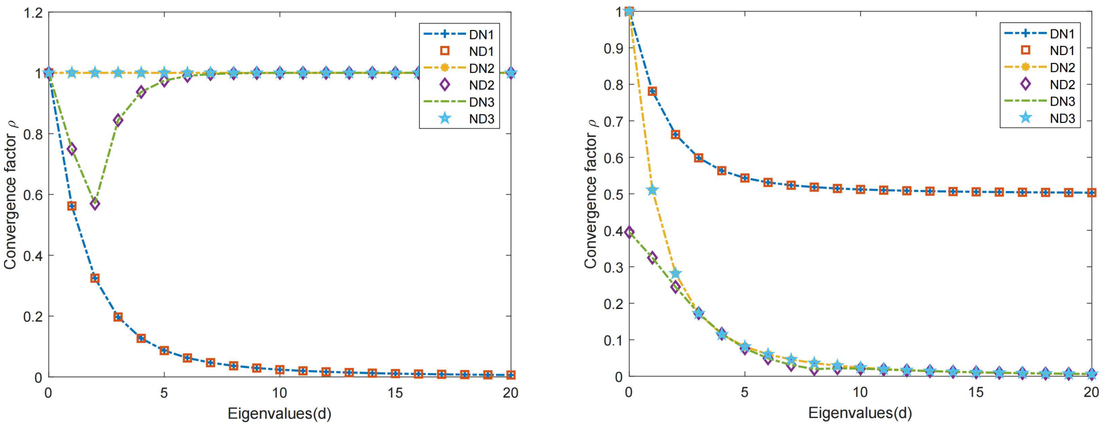

Figure 1: Convergence factors with $\alpha = T/2$ of six proposed algorithms as a function of the eigenvalues d. Left: $\theta = 1$. Right: $\theta = 0.5$.

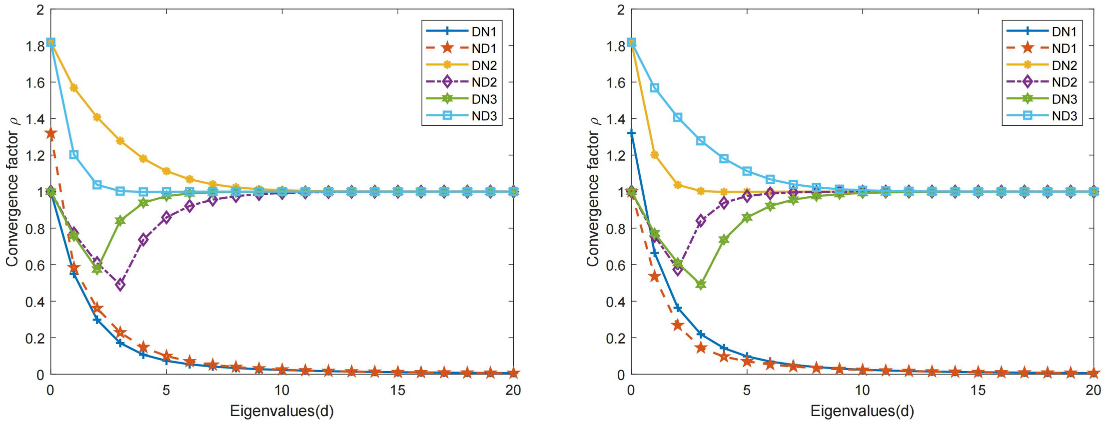

Figure 2: Convergence factors with $\theta = 1$ of six proposed algorithms as a function of the eigenvalues d. Left: $\alpha = 0.3$. Right: $\alpha = 0.7$.

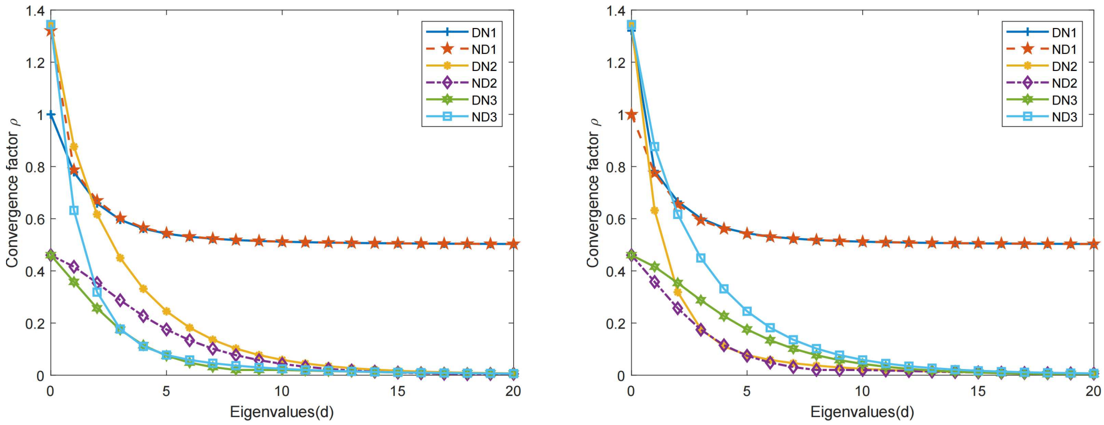

Figure 3: Convergence factors with $\theta = 0.5$ of six proposed algorithms as a function of the eigenvalues d. Left: $\alpha = 0.3$. Right: $\alpha = 0.7$.

4.2. Convergence factor for different θ

In this part, we investigate the influence of different relaxation parameters θ for all six proposed algorithms, and try to find the optimal θ for each algorithm in the case of symmetric and asymmetric decomposition. Firstly, for algorithms of Category I, i.e., DN_1 and ND_1, we know that they are only good smoothers but not good solvers based on the analysis in Section 3. Then we explore the influence of the relaxation parameter θ on the convergence factors of the DN_1 and ND_1 algorithms. And the convergence performance of DN_1 and ND_1 under different θ for $\alpha = 0.3$, $\alpha = 0.5$ and $\alpha = 0.7$ is shown in Fig. 4. As shown in Fig. 4, the convergence factors ρ of DN_1 and ND_1 decrease with the increase of θ for $\alpha = 0.3$, $\alpha = 0.5$ and $\alpha = 0.7$. Then we find that $\theta = 1$ is the best choice when we choose algorithms of Category I. Furthermore, combined with Fig. 4 and Remarks 3.1, 3.3 in theoretical convergence

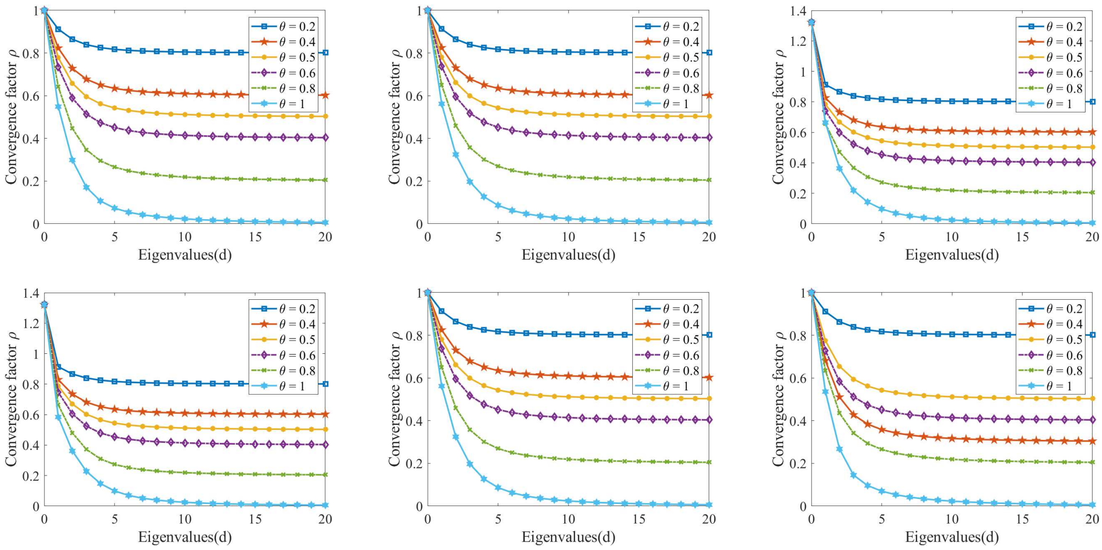

Figure 4: Convergence factors with different relaxation parameters θ of DN_1 and ND_1 as a function of the eigenvalues d for $\alpha = 0.3, 0.5, 0.7$ (left to right). Top: DN_1. Bottom: ND_1.

analysis, we conclude that when choosing the DN_1 algorithm, the parameter α should satisfy $\alpha \leq T/2$, because for small eigenvalues (i.e., $d \to 0$), the convergence factors $\rho_{DN_1} > 1$ for $\alpha > T/2$. And this situation is just opposite for the ND_1 algorithm, then for ND_1, we choose $\alpha \geq T/2$ to ensure the convergence of ND_1.

For the DN_2 and ND_3 algorithms, their convergence performance with different relaxation parameters θ is shown in Fig. 5. The two sub-figures in the middle of Fig. 5

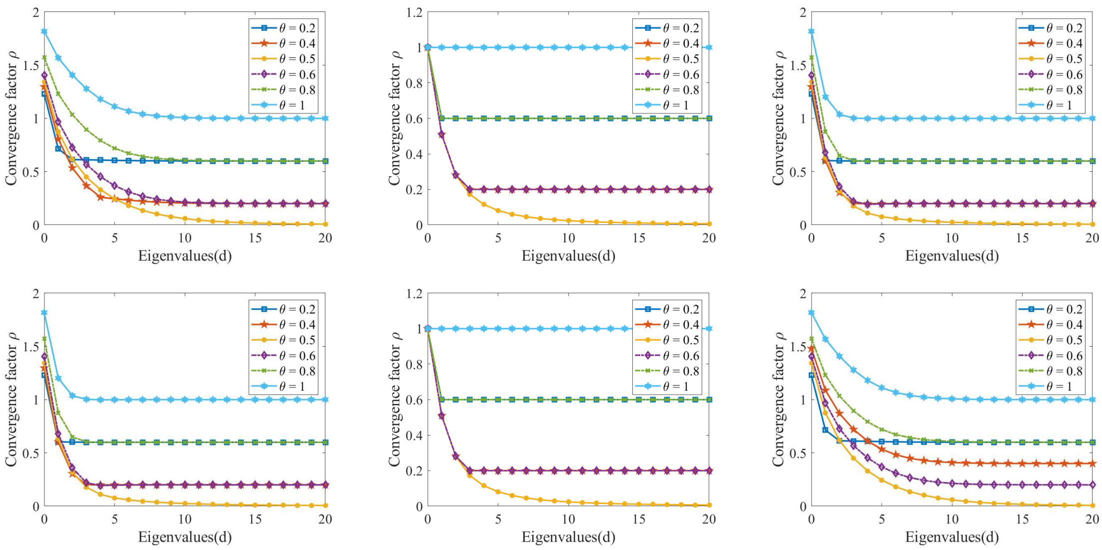

Figure 5: Convergence factors with different relaxation parameters θ of DN_2 and ND_3 as a function of the eigenvalues d for $\alpha = 0.3, 0.5, 0.7$ (left to right). Top: DN_2. Bottom: ND_3.

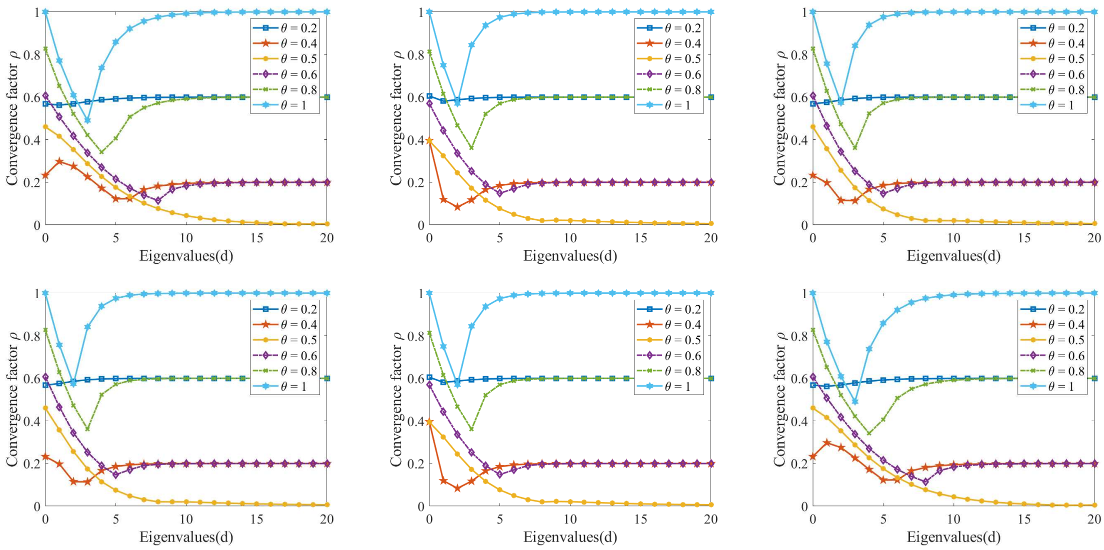

Figure 6: Convergence factors with different relaxation parameters θ of ND_2 and DN_3 as a function of the eigenvalues d for $\alpha = 0.3, 0.5, 0.7$ (left to right). Top: ND_2. Bottom: DN_3.

show the convergence performance of DN_2 and ND_3 when the decomposition is symmetric. We observe the convergence factors of DN_2 and ND_3 satisfy $\rho(\theta) = \rho(1-\theta)$, which corresponds to Corollaries 3.3 and 3.4 in theoretical analysis, especially when $\theta = 0.5$, the convergence of DN_2 and ND_3 are better than other selected θ. Meanwhile, in the case of the asymmetric decomposition, the convergence performance of DN_2 and ND_3 is shown in the left and right of Fig. 5. And we discover the convergence factors of DN_2 and ND_3 are greater than 1 when $d \to 0$, which is consistent with Remarks 3.6 and 3.11. Thus, the DN_2 and ND_3 algorithms may be divergent for the asymmetric decomposition.

Fig. 6 shows the convergence behavior of the ND_2 and DN_3 algorithms with different relaxation parameters θ both for the symmetric and asymmetric decompositions. We observe the convergence of ND_2 and DN_3 with $\theta = 0.4$ and $\theta = 0.5$ are better than the other selected parameters θ in Fig. 6 both for the symmetric and asymmetric decompositions. Moreover, the convergence performance with $\theta = 0.4$ is superior to $\theta = 0.5$ for eigenvalues $d \to 0$, and the convergence behavior with $\theta = 0.5$ is better than $\theta = 0.4$ for large eigenvalues. Finally, combining the previous analysis in this subsection, we can discover that DN_2 and ND_3 with appropriate parameters θ have better convergence performance compared with the other four algorithms, i.e., the DN_1, ND_1, DN_2 and ND_3 algorithms.

4.3. Convergence factor for different r

From all six proposed algorithms, we choose the DN_1, DN_2, and DN_3 algorithms to investigate the dependence of the convergence factors on the parameter r through

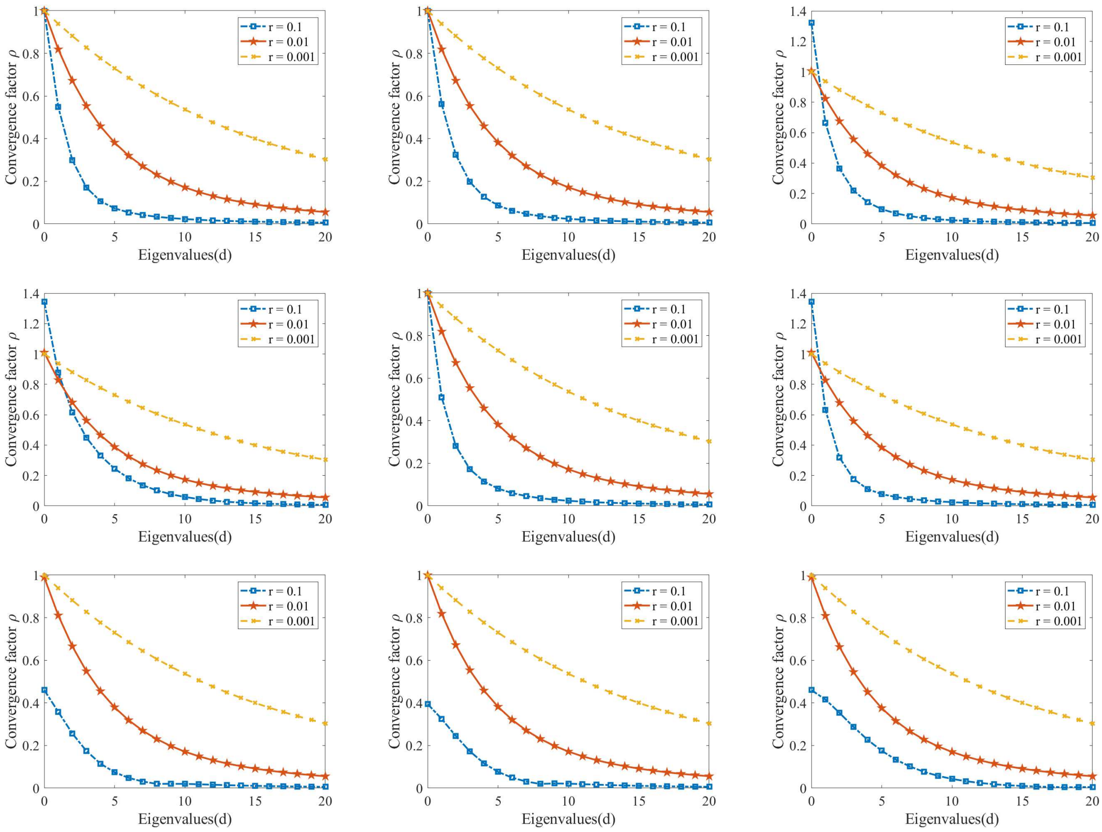

Figure 7: Convergence factors of DN_1, DN_2, DN_3 (top to bottom) with different r as a function of the eigenvalues d for $\alpha = 0.3, 0.5, 0.7$ (left to right).

their numerical performances for $\alpha = 0.3, 0.5, 0.7$ in Fig. 7. Meanwhile, we choose the relaxation coefficient θ which makes the convergence factor $\rho \to 0$ as the eigenvalues $d \to \infty$ for the selected algorithms. As shown in Fig. 7, it can be observed that the convergence factors increase as r decreases.

4.4. Numerical performance of six proposed algorithms with optimal θ

This subsection explores the convergence of six proposed algorithm in a practical numerical example. In this work, the system (2.5) is discretized with the grid points $t_s = s\Delta t, (s = 0, 1, 2, \ldots, N, N = T/\Delta t)$ using the Crank-Nicolson scheme,

$$\begin{cases} \dfrac{Y_{s+1} - Y_s}{\Delta t} + \dfrac{A}{2}(Y_s + Y_{s+1}) - \dfrac{1}{2r}(\Lambda_s + \Lambda_{s+1}) = 0, \\ Y_0 = Y_N, \end{cases}$$

$$\begin{cases} \dfrac{\Lambda_{s+1} - \Lambda_s}{\Delta t} - \dfrac{1}{2}(Y_s + Y_{s+1}) - \dfrac{A}{2}(\Lambda_s + \Lambda_{s+1}) = \dfrac{1}{2}\big((Y_d)_s + (Y_d)_{s+1}\big), \\ \Lambda_N = \Lambda_0, \end{cases}$$

where $Y_s = \boldsymbol{y}(t_s), \Lambda_s = \boldsymbol{\lambda}(t_s), (Y_d)_s = \boldsymbol{y_d}(t_s)$. The error of algorithm is regarded as the maximum error of all grid points. According to the theoretical analysis and numerical performance of the convergence factors ρ mentioned above, we know the convergence factors of the proposed algorithms are related to the eigenvalues of matrix A, then the optimal relaxation parameter $\theta^\star$ is determined by the range of eigenvalues of A in practical applications. Then, for the forward-backward system (2.5), we consider the following two numerical examples, which are located in one-dimensional and two-dimensional space respectively.

- **Case A:** The space interval is $\Omega = (-4, 4)$, and the target state is
$$y_d(t,x) = \Big[(1+t)\sin(\pi t)\big(e^{-8(x-1)^2} + e^{-8(x+1)^2} - e^{-8} - e^{-72}\big)\Big]^+,$$
where $[\cdot]^+ := \max\{\cdot, 0\}$. The spatial and temporal discrete mesh sizes are $\Delta x = \Delta t = 0.01$.

- **Case B:** The space interval is $\Omega = (0,1) \times (0,1) \in \mathbb{R}^2$, and the target state is
$$y_d(t,\boldsymbol{x}) = x_1^2(1-x_1)^2 x_2^2(1-x_2)^2 \sin(2\pi t).$$
The spatial discrete mesh sizes are $\Delta x_1 = \Delta x_2 = 0.1$, and temporal discrete mesh size is $\Delta t = 0.01$.

For each test, we consider the time interval $I = (0,T) = (0,1)$, and the parameter $r = 0.1$ is chosen. Based on the above parameters setting, the optimal relaxation parameters $\theta^\star$ with different α are obtained by the numerical optimization for Case A and Case B, which are shown in detail in Tables 2 and 3 respectively. In these tables, ρ is the maximum of convergence factors with the optimal relaxation parameter $\theta^\star$, and the bold part of each row has the best convergence for the selected α.

For Case A, the range of eigenvalues of matrix A is $[0.1542, 4 \times 10^4]$, based on the Table 2, we can see that all six proposed algorithms are convergent for the symmetric decomposition, and the convergence of ND_2 (DN_3) is the best. However, when the decomposition is asymmetric, the ND_1, DN_2, ND_3 algorithms still diverge under

Table 2: Convergence factors with the numerical optimal relaxation parameter $\theta^\star$ for Case A.

		DN_1	ND_1	DN_2	ND_2	DN_3	ND_3
$\alpha = 0.3$	$\theta^\star$	1	10^{-5}	10^{-5}	0.371	**0.394**	10^{-5}
	ρ	0.914	1.094	1.080	0.259	**0.214**	1.049
$\alpha = 0.5$	$\theta^\star$	1	1	0.5	**0.421**	**0.421**	0.5
	ρ	0.914	0.914	0.900	**0.186**	**0.186**	0.900
$\alpha = 0.7$	$\theta^\star$	10^{-5}	1	10^{-5}	**0.394**	0.371	10^{-5}
	ρ	1.171	0.914	1.049	**0.214**	0.259	1.080

Table 3: Convergence factors with the numerical optimal relaxation parameter $\theta^\star$ for Case B.

		DN_1	ND_1	DN_2	ND_2	DN_3	ND_3
$\alpha = 0.3$	$\theta^\star$	1	1	0.497	0.501	0.503	0.497
	ρ	0.0064	0.0064	0.0074	0.0051	0.0064	0.0064
$\alpha = 0.5$	$\theta^\star$	1	1	0.497	0.503	0.503	0.497
	ρ	0.0064	0.0064	0.0064	0.0064	0.0064	0.0064
$\alpha = 0.7$	$\theta^\star$	1	1	0.497	0.503	0.501	0.497
	ρ	0.0064	0.0064	0.0064	0.0064	0.0051	0.0074

the optimal relaxation parameters $\theta^\star$ for $\alpha = 0.3$ ($\alpha < T/2$), and the DN_3 algorithm demonstrates the best convergence performance among DN_1, ND_2 and DN_3 which are convergent. If $\alpha = 0.7$ ($\alpha > T/2$), the convergence factors of DN_1, DN_2, ND_3 still fail to converge under $\theta^\star$, and the convergence performance of the ND_2 algorithm are superior to the ND_1, ND_2 and DN_3 algorithms which are convergent. And the error convergence performance of the algorithms for the symmetric and asymmetric decompositions with the optimal relaxation parameter $\theta^\star$ for Case A is shown in Fig. 8. As shown in Fig. 8, for Case A, the convergence of ND_2 and DN_3 are similar both for the symmetric and asymmetric decompositions, and they outperform the other four algorithms whose convergence rates are still very slow with the optimal $\theta^\star$. In particular, for the asymmetric decomposition, the ND_1, DN_2 and ND_3 algorithms diverge for $\alpha = 0.3$, and if $\alpha = 0.7$ the DN_1, DN_2 and ND_3 algorithms diverge which is consistent with the results in Table 2. Moreover, it can be observed that the convergence of the ND_2 and DN_3 algorithms for the symmetric decomposition is better than the asymmetric decomposition by comparing the three sub-figures in Fig. 8.

For Case B, the range of eigenvalues of matrix A is $[19.5774, 780.4226]$. From Table 3, it can be seen that six proposed algorithms with the optimal relaxation parameter $\theta^\star$ are convergent for all selected α, and the corresponding convergence factors ρ are very small and close. Then the convergence performance of each algorithm with the optimal relaxation parameter $\theta^\star$ is similar, which is also verified by the error convergence performance in Fig. 9.

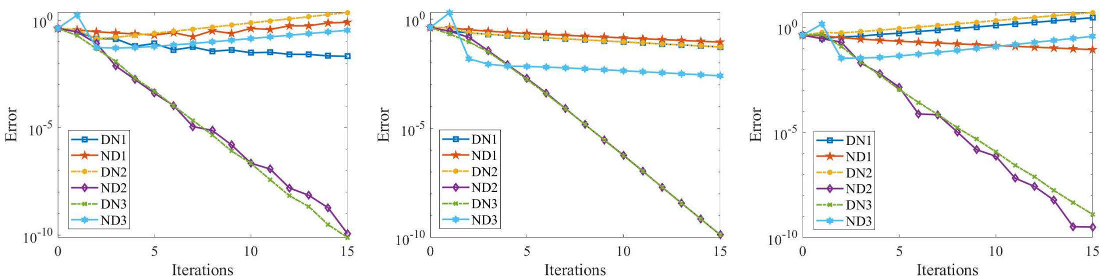

Figure 8: Error decay of six proposed algorithms for Case A with $\alpha = 0.3, 0.5, 0.7$ (left to right).

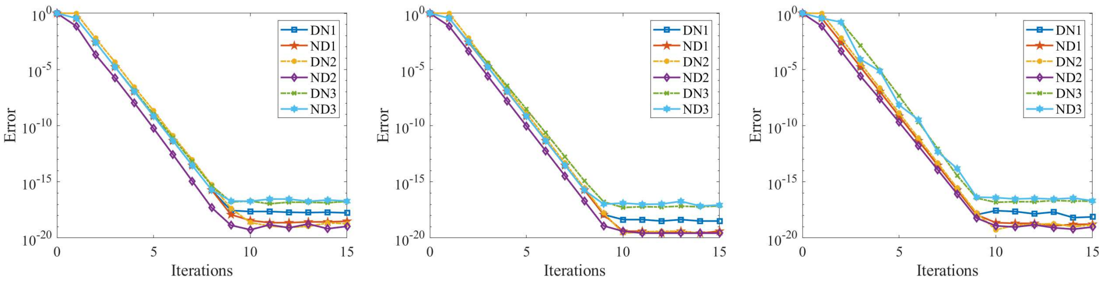

Figure 9: Error decay of six proposed algorithms for Case B with $\alpha = 0.3, 0.5, 0.7$ (left to right).

5. Conclusions

In this work, we proposed Dirichlet-Neumann and Neumann-Dirichlet methods and their variants for the time-periodic parabolic optimal control problems. Different from the initial-value optimal control problems, by the Lagrange multiplier approach, a coupled forward-backward system was obtained with the time-periodic conditions which is more complex to solve. Then for the time-periodic parabolic optimal control problems, the treatment of the time-periodic conditions is the key and difficult point when designing the DN and ND algorithms in time. We presented the convergence analysis for the proposed DN and ND algorithms. By comparing and analyzing the convergence of all proposed algorithms, for the relaxation coefficient $\theta = 1$, the DN_1 and ND_1 algorithms of Category I are good smoothers but not good solvers. And the DN_2 and ND_3 algorithms for the symmetric decomposition with $\theta = 0.5$ are good smoothers but not good solvers. For the parameter r, we can find that the convergence factors of proposed algorithms increase as r decreases from their numerical performances. And we expect to get further conclusions in future research for the theoretical dependency on r of convergence factors. Meanwhile, we find that the convergence performance of the algorithm is closely related to the eigenvalue of matrix A obtained by spatial discretization, especially the minimum eigenvalue of A. Furthermore, compared with other algorithms, the ND_2 and DN_3 algorithms with optimal relaxation parameter $\theta^\star$ always have good convergence performance both for Case A and Case B. Then ND_2 and DN_3 are high-efficiency solvers for the time-periodic parabolic optimal control problems, and several numerical experiments finally illustrate our theoretical results. Moreover, the ND_2 and DN_3 algorithms can be considered applying to the multiple subdomains decomposition and implementing in parallel. In the future research, the Neumann-Neumann method which can be implemented in parallel is also an interesting topic for the time-periodic parabolic optimal control problems.

Acknowledgements

The authors would like to thank the anonymous referees for their constructive criticisms and helpful comments to improve the paper significantly.

This work was funded by the National Key Research and Development Program of China under grant 2022YFA1004600, by the Natural Science Foundation of China (NSFC) under grant 12271426, and by the Natural Science Basic Research Plan in Shaanxi Province of China under grant 2025JC-YBMS-023.

References

[1] D. Abbeloos, M. Diehl, M. Hinze, and S. Vandewalle, *Nested multigrid methods for time-periodic, parabolic optimal control problems*, Comput. Visual Sci. 14 (2011), 27–38.

[2] O. Axelsson, S. Farouq, and M. Neytcheva, *A preconditioner for optimal control problems, constrained by stokes equation with a time-harmonic control*, J. Comput. Appl. Math. 310 (2017), 5–18.

[3] O. Axelsson and Z. Z. Liang, *A note on preconditioning methods for time-periodic eddy current optimal control problems*, J. Comput. Appl. Math. 352 (2019), 262–277.

[4] J.-D. Benamou, *A domain decomposition method with coupled transmission conditions for the optimal control of systems governed by elliptic partial differential equations*, SIAM J. Numer. Anal. 33 (1996), 2401–2416.

[5] J.-D. Benamou, *A domain decomposition method for control problems*, in: The 9th International Conference on Domain Decomposition Methods, (1998), 266–273.

[6] A. Borzì and V. Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, SIAM, 2012.

[7] G. Ciaramella, L. Halpern, and L. Mechelli, *Convergence analysis and optimization of a Robin Schwarz waveform relaxation method for time-periodic parabolic optimal control problems*, J. Comput. Phys. 496 (2024), Paper No. 112572.

[8] V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation*, SIAM, 2015.

[9] X. Du, M. Sarkis, C. E. Schaerer, and D. B. Szyld, *Inexact and truncated parareal-in-time Krylov subspace methods for parabolic optimal control problems*, Electron. Trans. Numer. Anal. 40 (2013), 36–57.

[10] M. Emmett and M. L. Minion, *Toward an efficient parallel in time method for partial differential equations*, Commun. Appl. Math. Comput. Sci. 7 (2012), 105–132.

[11] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM J. Sci. Comput. 36 (2014), C635–C661.

[12] P. F. Fischer, F. Hecht, and Y. Maday, *A parareal in time semi-implicit approximation of the Navier-Stokes equations*, in: Domain Decomposition Methods in Science and Engineering. Lecture Notes in Computational Science and Engineering, Springer, 40 (2005), 433–440.

[13] S. Friedhoff, R. D. Falgout, T. Kolev, S. MacLachlan, and J. B. Schroder, *A Multigrid-in-Time Algorithm for Solving Evolution Equations in Parallel*, Lawrence Livermore National Lab, 2012.

[14] M. J. Gander, *50 years of time parallel time integration*, in: Multiple Shooting and Time Domain Decomposition Methods, Springer, (2015), 69–113.

[15] M. J. Gander, Y. L. Jiang, B. Song, and H. Zhang, *Analysis of two parareal algorithms for time-periodic problems*, SIAM J. Sci. Comput. 35 (2013), A2393–A2415.

[16] M. J. Gander and F. Kwok, *Schwarz methods for the time-parallel solution of parabolic control problems*, in: Domain Decomposition Methods in Science and Engineering XXII. Lecture Notes in Computational Science and Engineering, Springer, (2016), 207–216.

[17] M. J. GANDER, F. KWOK, AND J. SALOMON, *PARAOPT: A parareal algorithm for optimality systems*, SIAM J. Sci. Comput. 42 (2020), A2773–A2802.

[18] M. J. GANDER, F. KWOK, AND G. WANNER, *Constrained Optimization: From Lagrangian Mechanics to Optimal Control and PDE Constraints*, Springer International Publishing, 2014.

[19] M. J. GANDER, J. LIU, S. L. WU, X. YUE, AND T. ZHOU, *ParaDiag: Parallel-in-time algorithms based on the diagonalization technique*, arXiv:2005.09158, 2020.

[20] M. J. GANDER AND L. D. LU, *New time domain decomposition methods for parabolic optimal control problems I: Dirichlet-Neumann and Neumann-Dirichlet algorithms*, SIAM J. Numer. Anal. 62 (2024), 2048–2070.

[21] M. J. GANDER AND L. D. LU, *New time domain decomposition methods for parabolic optimal control problems II: Neumann-Neumann algorithms*, SIAM J. Numer. Anal. 62 (2024), 2588–2610.

[22] S. GÖTSCHEL AND M. L. MINION, *Parallel-in-time for parabolic optimal control problems using PFASST*, in: Domain Decomposition Methods in Science and Engineering XXIV. Lecture Notes in Computational Science and Engineering, Springer, (2018), 363–371.

[23] S. GÖTSCHEL AND M. L. MINION, *An efficient parallel-in-time method for optimization with parabolic PDEs*, SIAM J. Sci. Comput. 41 (2019), C603–C626.

[24] B. HOUSKA AND M. DIEHL, *Robustness and stability optimization of power generating kite systems in a periodic pumping mode*, in: 2010 IEEE International Conference on Control Applications, IEEE, (2010), 2172–2177.

[25] M. KOLMBAUER AND U. LANGER, *A robust preconditioned MinRes solver for distributed time-periodic eddy current optimal control problems*, SIAM J. Sci. Comput. 34 (2012), B785–B809.

[26] R. KRUG, G. LEUGERING, A. MARTIN, M. SCHMIDT, AND D. WENINGER, *Time-domain decomposition for optimal control problems governed by semilinear hyperbolic systems*, SIAM J. Control Optim. 59 (2021), 4339–4372.

[27] F. KWOK, *On the time-domain decomposition of parabolic optimal control problems*, in: Domain Decomposition Methods in Science and Engineering XXIII. Lecture Notes in Computational Science and Engineering, Springer, (2017), 55–67.

[28] F. KWOK AND D. N. TOGNON, *A parallel in time algorithm based on ParaExp for optimal control problems*, in: 2024 IEEE 63rd Conference on Decision and Control, IEEE, (2024), 6685–6690.

[29] J. E. LAGNESE AND G. LEUGERING, *Time-domain decomposition of optimal control problems for the wave equation*, Syst Control Lett, 48 (2003), 229–242.

[30] J. E. LAGNESE AND G. LEUGERING, *Domain Decomposition Methods in Optimal Control of Partial Differential Equations*, Birkhäuser Basel, 2004.

[31] G. LEUGERING, *Space-time-domain decomposition for optimal control problems governed by linear hyperbolic systems*, J. Optim. Differ. Equ. their Appl. 29 (2021), 24.

[32] X. L. LIN AND S. L. WU, *A parallel-in-time preconditioner for Crank-Nicolson discretization of a parabolic optimal control problem*, J. Comput. Appl. Math. 451 (2024), Paper No. 116106.

[33] J. L. LIONS, Y. MADAY, AND G. TURINICI, *A "parareal" in time discretization of PDE's*, C. R. Acad. Sci., Paris, Sér. I Math. 332 (2001), 661–668.

[34] Y. MADAY AND E. M. RONQUIST, *Parallelization in time through tensor-product space–time solvers*, C. R. Math. 346 (2008), 113–118.

[35] B. C. MANDAL, *Substructuring waveform relaxation methods for parabolic optimal control problems*, in: Bansal, J., Das, K., Nagar, A., Deep, K., Ojha, A. (Eds) Soft Computing for

Problem Solving. Advances in Intelligent Systems and Computing, Springer, Singapore, (2019), 485–494.

[36] T. P. MATHEW, M. SARKIS, AND C. E. SCHAERER, *Analysis of block parareal preconditioners for parabolic optimal control problems*, SIAM J. Sci. Comput. 32 (2010), 1180–1200.

[37] D. L. RUSSELL, J. L. LIONS, AND S. K. MITTER, *Optimal Control of Systems Governed by Partial Differential Equations*, Am. Math. Mon., 1972.

[38] F. TROLTZSCH AND J. SPREKELS, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, AMS, 2010.

[39] S. L. WU AND J. LIU, *A parallel-in-time block-circulant preconditioner for optimal control of wave equations*, SIAM J. Sci. Comput. 42 (2020), A1510–A1540.

[40] S. L. WU, Z. Y. WANG AND T. ZHOU, *PinT preconditioner for forward-backward evolutionary equations*, SIAM J. Matrix Anal. Appl. 44 (2023), 1771–1798.

[41] H. YANG AND X. C. CAI, *Two-level space time domain decomposition methods for flow control problems*, J. Sci. Comput. 70 (2016), 717–743.