NEW TIME DOMAIN DECOMPOSITION METHODS FOR PARABOLIC OPTIMAL CONTROL PROBLEMS II: NEUMANN–NEUMANN ALGORITHMS*

MARTIN J. GANDER† AND LIU-DI LU†

Abstract. We propose to use Neumann–Neumann algorithms for the time parallel solution of unconstrained linear parabolic optimal control problems. We study nine variants, analyze their convergence behavior and determine the optimal relaxation parameter for each. Our findings indicate that while the most intuitive Neumann–Neumann algorithms act as effective smoothers, there are more efficient Neumann–Neumann solvers available. We support our analysis with numerical experiments.

Key words. time domain decomposition, Neumann–Neumann algorithm, parallel in time, parabolic optimal control problems, convergence analysis.

MSC codes. 65M12, 65M55, 65Y05,

1. Introduction.

As our model problem, we consider a parabolic optimal control problem: for a given target function $\hat{y} \in L^2(Q)$, $\gamma > 0$, and $\nu \geq 0$, we want to minimize the cost functional

$$J(y,u) := \frac{1}{2}\|y-\hat{y}\|^2_{L^2(Q)} + \frac{\gamma}{2}\|y(T)-\hat{y}(T)\|^2_{L^2(\Omega)} + \frac{\nu}{2}\|u\|^2_{U_{\mathrm{ad}}}, \tag{1.1}$$

subject to the linear parabolic state equation:

$$\begin{aligned} \partial_t y - \Delta y &= u && \text{in } Q := \Omega \times (0,T), \\ y &= 0 && \text{on } \Sigma := \partial\Omega \times (0,T), \\ y(0) &= y_0 && \text{on } \Sigma_0 := \Omega \times \{0\}, \end{aligned} \tag{1.2}$$

where $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$ is a bounded domain with boundary $\partial\Omega$, and T is the fixed final time. The control u on the right-hand side of the PDE is in an admissible set U_{ad}, and we want to control the solution of the parabolic PDE (1.2) toward a target state $\hat{y}$. For simplicity, we consider homogeneous boundary conditions. The parabolic optimal control problem (1.1)-(1.2) leads to necessary first-order optimality conditions (see e.g., [28, 30]), which include a forward in time primal state equation (1.2), a backward in time dual state equation,

$$\begin{aligned} \partial_t \lambda + \Delta\lambda &= y - \hat{y} && \text{in } Q, \\ \lambda &= 0 && \text{on } \Sigma, \\ \lambda(T) &= -\gamma(y(T) - \hat{y}(T)) && \text{on } \Sigma_T := \Omega \times \{T\}, \end{aligned} \tag{1.3}$$

and an algebraic equation $\lambda = \nu u$ with λ the dual state. This forward-backward system cannot be solved by standard time-stepping methods, and has to be solved either iteratively or at once. Solving at once the space-time discretized system can be challenging, especially for spatial dimension larger than one. To overcome

*Submitted to the editors 2024.01.25.

Funding: This work was funded by the Swiss National Science Foundation Grant 192064.

†Section de Mathématiques, Université de Genève, rue du Conseil-Général 5-7, CP 64, 1205, Geneva, Switzerland (martin.gander@unige.ch, liudi.lu@unige.ch).

this challenge, one can use gradient type methods by solving sequentially forward-backward systems [20, 30]. Multigrid methods [1, 4, 17, 27], tensor product techniques [5, 16, 23, 31], model order reduction [2, 21, 22, 24], can also be applied to solve such problems. Since the role of the time variable in forward-backward optimality systems is key, it is natural to seek efficient solvers through Parallel-in-time techniques. This includes, waveform relaxation [26, 18], Parareal [29], PITA [9], PFASST [6], MGRIT [7], see also the survey paper [11]. Application of such techniques to treat parabolic optimal control problems can be found in [8, 13, 15, 19].

In [14], we considered a new time domain decomposition approach motivated by [12, 25], and analyzed the convergence behavior of Dirichlet–Neumann and Neumann–Dirichlet algorithms within this framework. We have surprisingly discovered different variants of Dirichlet–Neumann and Neumann–Dirichlet algorithms for the parabolic optimal control problem (1.1)-(1.2), when decomposing in time. This is mainly due to the forward-backward structure of the optimality system. The present paper is the sequel of [14]: the goal of the current paper is to investigate Neumann–Neumann algorithms [3] in the context of time domain decomposition and analyze theoretically the convergence behavior of these algorithms. We consider a semidiscretization in space and focus on the time variable. This consists in replacing the spatial operator $-\Delta$ by a matrix $A \in \mathbb{R}^{n\times n}$, for instance using a finite difference discretization in space. If A is symmetric, which is natural for discretizations of $-\Delta$, then it can be diagonalized with $A = PDP^T$, and the diagonalized system reads,

$$(1.4)\qquad \begin{cases} \begin{pmatrix} \dot{z}_i \\ \dot{\mu}_i \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_i \\ \mu_i \end{pmatrix} = \begin{pmatrix} 0 \\ -\hat{z}_i \end{pmatrix} \text{ in } (0,T), \\ z_i(0) = z_{i,0}, \\ \mu_i(T) + \gamma z_i(T) = \gamma \hat{z}_i(T), \end{cases}$$

where d_i is the ith eigenvalue of the matrix A, and z_i, μ_i as well as $\hat{z}_i$ are the ith components of the vectors $\boldsymbol{z}$, $\boldsymbol{\mu}$ and $\hat{\boldsymbol{z}}$. Eliminating μ_i in (1.4), we obtain the second-order ODE

$$(1.5)\qquad \begin{cases} \ddot{z}_i - (d_i^2 + \nu^{-1}) z_i = -\nu^{-1}\hat{z}_i \text{ in } (0,T), \\ z_i(0) = z_{i,0}, \\ \dot{z}_i(T) + (\nu^{-1}\gamma + d_i) z_i(T) = \nu^{-1}\gamma \hat{z}_i(T). \end{cases}$$

We refer to [14, Section 2] for more details about the transition from the PDE-constrained problem (1.1)-(1.2) to the diagonalized reduced problem (1.4).

The rest of the paper is structured as follows. We introduce in Section 2 our new time decomposed Neumann–Neumann algorithms and study their convergence behavior in Section 3. Numerical experiments are shown in Section 4 to support our analysis, and we draw conclusions in Section 5.

2. Neumann–Neumann algorithms.

In this section, we apply the Neumann–Neumann technique (NN) in time to obtain our new time domain decomposition methods to solve the system (1.4), and investigate their convergence behavior. To focus on the error equation, we set both the initial condition $\boldsymbol{y}_0 = 0$ (i.e., $\boldsymbol{z}_0 = 0$) and the target function $\hat{\boldsymbol{y}} = 0$ (i.e., $\hat{\boldsymbol{z}} = 0$). We decompose the time domain $\Omega := (0,T)$ into two nonoverlapping subdomains $\Omega_1 := (0,\alpha)$ and $\Omega_2 := (\alpha, T)$, where α is the interface. And we denote by $z_{j,i}$ and $\mu_{j,i}$ the restriction to Ω_j, $j = 1,2$ of the states z_i and μ_i. Although we will focus on the two-subdomain case in our current

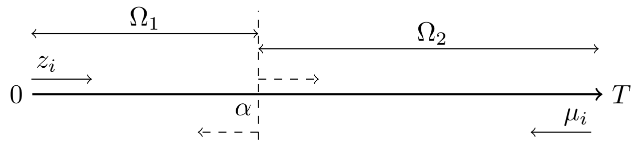

FIG. 1. *Illustration of the forward-backward system.*

study, the results can be extended to N nonoverlapping subdomains $\Omega_j := (\alpha_j, \alpha_{j+1})$, $j = 1, \ldots, N$ with $\alpha_1 = 0$ and $\alpha_{N+1} = T$.

Unlike the name of the NN algorithm suggests, it starts first with a Dirichlet step, which will be corrected by a Neumann step and then updates the transmission condition. As the system (1.4) is a forward-backward system, it appears natural at first glance to keep this property for the decomposed case as illustrated in Figure 1: we expect to have a final condition for the dual state $\mu_{1,i}$ in Ω_1, since we already have an initial condition for $z_{1,i}$; similarly, we expect to have an initial condition for the primal state $z_{2,i}$ in Ω_2, where we already have a final condition for $\mu_{2,i}$. Therefore, for iteration index $k = 1, 2, \ldots$, a natural NN algorithm first solves the Dirichlet step

$$\begin{cases} \begin{pmatrix} \dot{z}_{1,i}^k \\ \dot{\mu}_{1,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{1,i}^k \\ \mu_{1,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_1, \\ z_{1,i}^k(0) = 0, \\ \mu_{1,i}^k(\alpha) = f_{\alpha,i}^{k-1}, \end{cases} \tag{2.1}$$
$$\begin{cases} \begin{pmatrix} \dot{z}_{2,i}^k \\ \dot{\mu}_{2,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{2,i}^k \\ \mu_{2,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_2, \\ z_{2,i}^k(\alpha) = g_{\alpha,i}^{k-1}, \\ \mu_{2,i}^k(T) + \gamma z_{2,i}^k(T) = 0, \end{cases}$$

then corrects the result by solving the Neumann step

$$\begin{cases} \begin{pmatrix} \dot{\psi}_{1,i}^k \\ \dot{\phi}_{1,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} \psi_{1,i}^k \\ \phi_{1,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_1, \\ \psi_{1,i}^k(0) = 0, \\ \dot{\phi}_{1,i}^k(\alpha) = \dot{\mu}_{1,i}^k(\alpha) - \dot{\mu}_{2,i}^k(\alpha), \end{cases} \tag{2.2}$$
$$\begin{cases} \begin{pmatrix} \dot{\psi}_{2,i}^k \\ \dot{\phi}_{2,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} \psi_{2,i}^k \\ \phi_{2,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_2, \\ \dot{\psi}_{2,i}^k(\alpha) = \dot{z}_{2,i}^k(\alpha) - \dot{z}_{1,i}^k(\alpha), \\ \phi_{2,i}^k(T) + \gamma \psi_{2,i}^k(T) = 0, \end{cases}$$

where ψ_i is the primal correction state for z_i and ϕ_i the dual correction state for μ_i. Finally, we update the transmission condition by

$$f_{\alpha,i}^k := f_{\alpha,i}^{k-1} - \theta_1\big(\phi_{1,i}^k(\alpha) + \phi_{2,i}^k(\alpha)\big), \quad g_{\alpha,i}^k := g_{\alpha,i}^{k-1} - \theta_2\big(\psi_{1,i}^k(\alpha) + \psi_{2,i}^k(\alpha)\big), \tag{2.3}$$

with two relaxation parameters $\theta_1, \theta_2 > 0$.

As shown in the algorithm (2.1)-(2.2), both Dirichlet and Neumann steps have the forward-backward structure. However, this structure only appears as being the natural one at first glance. Indeed, isolating the variable in each equation in the systems (2.1) and (2.2), we find the identities

$$\mu_i = \nu(\dot{z}_i + d_i z_i), \quad z_i = \dot{\mu}_i - d_i\mu_i, \quad \phi_i = \nu(\dot{\psi}_i + d_i\psi_i), \quad \psi_i = \dot{\phi}_i - d_i\phi_i. \tag{2.4}$$

To shorten the notation, we define

$$\sigma_i := \sqrt{d_i^2 + \nu^{-1}}, \quad \omega_i := d_i + \gamma\nu^{-1}, \quad \beta_i := 1 - \gamma d_i. \tag{2.5}$$

Using (2.4) and (2.5), we can rewrite the Dirichlet step (2.1) in terms of the primal state z_i,

$$\begin{cases} \ddot{z}_{1,i}^k - \sigma_i^2 z_{1,i}^k = 0 \text{ in } \Omega_1, \\ z_{1,i}^k(0) = 0, \\ \dot{z}_{1,i}^k(\alpha) + d_i z_{1,i}^k(\alpha) = f_{\alpha,i}^{k-1}, \end{cases} \quad \begin{cases} \ddot{z}_{2,i}^k - \sigma_i^2 z_{2,i}^k = 0 \text{ in } \Omega_2, \\ z_{2,i}^k(\alpha) = g_{\alpha,i}^{k-1}, \\ \dot{z}_{2,i}^k(T) + \omega_i z_{2,i}^k(T) = 0. \end{cases} \tag{2.6}$$

Similarly, the Neumann step (2.2) can be rewritten in terms of the primal correction state ψ_i,

$$\begin{cases} \ddot{\psi}_{1,i}^k - \sigma_i^2 \psi_{1,i}^k = 0 \text{ in } \Omega_1, \\ \psi_{1,i}^k(0) = 0, \\ \dot{\psi}_{1,i}^k(\alpha) + \frac{\sigma_i^2}{d_i}\psi_{1,i}^k(\alpha) = \big(\dot{z}_{1,i}^k(\alpha) + \frac{\sigma_i^2}{d_i} z_{1,i}^k(\alpha)\big) - \big(\dot{z}_{2,i}^k(\alpha) + \frac{\sigma_i^2}{d_i} z_{2,i}^k(\alpha)\big), \end{cases}$$
$$\begin{cases} \ddot{\psi}_{2,i}^k - \sigma_i^2 \psi_{2,i}^k = 0 \text{ in } \Omega_2, \\ \dot{\psi}_{2,i}^k(\alpha) = \dot{z}_{2,i}^k(\alpha) - \dot{z}_{1,i}^k(\alpha), \\ \dot{\psi}_{2,i}^k(T) + \omega_i \psi_{2,i}^k(T) = 0, \end{cases} \tag{2.7}$$

and the transmission condition (2.3) becomes

$$\begin{aligned} f_{\alpha,i}^k &= f_{\alpha,i}^{k-1} - \theta_1\big(\dot{\psi}_{1,i}^k(\alpha) + d_i\psi_{1,i}^k(\alpha) + \dot{\psi}_{2,i}^k(\alpha) + d_i\psi_{2,i}^k(\alpha)\big), \\ g_{\alpha,i}^k &= g_{\alpha,i}^{k-1} - \theta_2\big(\psi_{1,i}^k(\alpha) + \psi_{2,i}^k(\alpha)\big). \end{aligned} \tag{2.8}$$

Instead of using (2.1)-(2.3) for our analysis, we will use the equivalent formulation in system (2.6)-(2.8), in which the forward-backward structure has disappeared. Furthermore, the Dirichlet step in (2.1) transforms in the primal state z_i to a Robin–Dirichlet (RD) step (2.6), and the Neumann step in (2.2) transforms in the primal correction state ψ_i to a Robin–Neumann (RN) step (2.7). In other words, we analyze actually a RD step with a RN correction, although it is originally a NN algorithm. We could also have interpreted the NN algorithm (2.1)-(2.3) using the dual state μ_i and the dual correction state ϕ_i, the algorithm would then read differently but the convergence analysis is still the same (see [14]). For the sake of consistency, we keep the interpretation with z_i and ψ_i for all convergence analyses.

The previous transformation reveals that the natural NN algorithm applied to the optimality system (1.4) is certainly not the only option. Since there are three components in a NN algorithm: a Dirichlet step, a Neumann step and an update step, this expands our options when dealing with parabolic optimal control problems, and provides us with more choices within the NN algorithm. More precisely, instead of applying the Dirichlet step to the pair (z_i, μ_i), one can also apply it only to the primal

TABLE 1
Variants of the Neumann-Neumann algorithm.

category	step	Ω_1	Ω_2	algorithm type
category I: (z_i, μ_i)	Dirichlet step	μ_i	z_i	(DD)
		$\dot{z}_i + d_i z_i$	z_i	(RD)
	Neumann step	$\dot{\phi}_i$	$\dot{\psi}_i$	(NN)
		$\ddot{\psi}_i + d_i\dot{\psi}_i$	$\dot{\psi}_i$	(RN)
		$\dot{\psi}_i$	$\dot{\psi}_i$	(NN)
		$\dot{\psi}_i$	$\dot{\psi}_i$	(NN)
		$\dot{\phi}_i$	$\dot{\phi}_i$	(NN)
		$\ddot{\psi}_i + d_i\dot{\psi}_i$	$\ddot{\psi}_i + d_i\dot{\psi}_i$	(RR)
category II: z_i	Dirichlet step	z_i	z_i	(DD)
		z_i	z_i	(DD)
	Neumann step	$\dot{\psi}_i$	$\dot{\psi}_i$	(NN)
		$\dot{\psi}_i$	$\dot{\psi}_i$	(NN)
		$\dot{\phi}_i$	$\dot{\psi}_i$	(NN)
		$\ddot{\psi}_i + d_i\dot{\psi}_i$	$\dot{\psi}_i$	(RN)
		$\dot{\phi}_i$	$\dot{\phi}_i$	(NN)
		$\ddot{\psi}_i + d_i\dot{\psi}_i$	$\ddot{\psi}_i + d_i\dot{\psi}_i$	(RR)
category III: μ_i	Dirichlet step	μ_i	μ_i	(DD)
		$\dot{z}_i + d_i z_i$	$\dot{z}_i + d_i z_i$	(RR)
	Neumann step	$\dot{\phi}_i$	$\dot{\phi}_i$	(NN)
		$\ddot{\psi}_i + d_i\dot{\psi}_i$	$\ddot{\psi}_i + d_i\dot{\psi}_i$	(RR)
		$\dot{\phi}_i$	$\dot{\psi}_i$	(NN)
		$\ddot{\psi}_i + d_i\dot{\psi}_i$	$\dot{\psi}_i$	(RN)
		$\dot{\psi}_i$	$\dot{\psi}_i$	(NN)
		$\dot{\psi}_i$	$\dot{\psi}_i$	(NN)

state z_i or the dual state μ_i. Likewise, the Neumann step can also be applied only to the primal correction state ψ_i or the dual correction state ϕ_i. We list in Table 1 all possible new time domain decomposition NN algorithms we can obtain, together with their equivalent interpretations in terms of the states z_i and ψ_i. According to the Dirichlet step, they can be classified into three main categories. Each category is composed of two blocks, the first block represents the Dirichlet step and the second block the three possible Neumann steps. And each step contains two rows, the first row is the algorithm applied to (1.4), and the second row represents the algorithm applied to (1.5). Note that the update step should also be adapted when modifying the Dirichlet step or the Neumann step. We will further discuss this in the next section, where we investigate the convergence of each algorithm.

Remark 2.1. Although most of the algorithms in Table 1 do not look like having the forward-backward structure, it can always be recovered by using the identities in (2.4). Furthermore, the transmission condition $\ddot{\psi}_i + d_i\dot{\psi}_i$ is actually a Robin type condition, considering the first equation in (2.7).

Remark 2.2. If the order in (2.1)-(2.2) is reversed, and one starts with the Neumann step, followed by the Dirichlet correction, the algorithm is then known under the name FETI (Finite Element Tearing and Interconnecting), invented by Farhat and Roux [10]. Since the two algorithms are very much related, we can also find similar variants as in Table 1 in the context of FETI algorithm.

3. Convergence analysis. In this section, we will study the convergence of each algorithm listed in Table 1. Note that the two systems (2.6) and (2.7) are very similar, the only difference is in the transmission condition at α. We can hence solve these two systems once and for all using the initial and the final condition, and find

(3.1)
$$\begin{aligned} z_{1,i}^k(t) &= A_i^k \sinh(\sigma_i t), \quad z_{2,i}^k(t) = B_i^k\Big(\sigma_i \cosh\big(\sigma_i(T-t)\big) + \omega_i \sinh\big(\sigma_i(T-t)\big)\Big),\\ \psi_{1,i}^k(t) &= C_i^k \sinh(\sigma_i t), \quad \psi_{2,i}^k(t) = D_i^k\Big(\sigma_i \cosh\big(\sigma_i(T-t)\big) + \omega_i \sinh\big(\sigma_i(T-t)\big)\Big).\end{aligned}$$

In general, the solutions (3.1) remain for all algorithms listed in Table 1, and the coefficients A_i^k, B_i^k, C_i^k and D_i^k will be determined by the transmission conditions. To stay in a compact form, we will only present the modified step for each NN variant instead of giving a complete three-step algorithm.

3.1. Category I. This category consists in applying the Dirichlet step to the pair (z_i, μ_i). As illustrated in Table 1, there are three variants according to the Neumann correction step.

3.1.1. Algorithm NN$_{1a}$. This is (2.1)-(2.3), at first glance the most natural NN algorithm, which keeps the forward-backward structure both for the Dirichlet and Neumann steps. To analyze its convergence behavior, we interpret it as (2.6)-(2.8) and solve for the exact iterates. Using (3.1), we determine the coefficients A_i^k, B_i^k through the transmission conditions in (2.6), and find

$$A_i^k = \frac{f_{\alpha,i}^{k-1}}{\sigma_i \cosh(a_i) + d_i \sinh(a_i)}, \quad B_i^k = \frac{g_{\alpha,i}^{k-1}}{\sigma_i \cosh(b_i) + \omega_i \sinh(b_i)}, \tag{3.2}$$

where we let $a_i := \sigma_i \alpha$ and $b_i := \sigma_i(T-\alpha)$ to simplify the notations, and $a_i + b_i = \sigma_i T$. Using once again (3.1), we determine the coefficients C_i^k, D_i^k through the transmission conditions in (2.7)

(3.3)
$$C_i^k = A_i^k - B_i^k \nu^{-1} \frac{\sigma_i \gamma \sinh(b_i) + \beta_i \cosh(b_i)}{\sigma_i \sinh(a_i) + d_i \cosh(a_i)}, \; D_i^k = A_i^k \frac{\cosh(a_i)}{\sigma_i \sinh(b_i) + \omega_i \cosh(b_i)} + B_i^k.$$

We then update the transmission condition (2.8) and find

$$\begin{pmatrix} f_{\alpha,i}^k \\ g_{\alpha,i}^k \end{pmatrix} = \begin{pmatrix} 1 - \theta_1 d_i E_i & \theta_1 \nu^{-1} F_i \\ -\theta_2 E_i & 1 - \theta_2 d_i F_i \end{pmatrix} \begin{pmatrix} f_{\alpha,i}^{k-1} \\ g_{\alpha,i}^{k-1} \end{pmatrix}, \tag{3.4}$$

with $E_i = \frac{\sigma_i \cosh(\sigma_i T) + \omega_i \sinh(\sigma_i T)}{\sigma_i \sinh(b_i) + \omega_i \cosh(b_i)} \frac{1}{\sigma_i \cosh(a_i) + d_i \sinh(a_i)}$ and $F_i = \frac{\sigma_i \cosh(\sigma_i T) + \omega_i \sinh(\sigma_i T)}{\sigma_i \cosh(b_i) + \omega_i \sinh(b_i)} \frac{1}{\sigma_i \sinh(a_i) + d_i \cosh(a_i)}$. The characteristic polynomial associated with the iteration matrix in (3.4) is $X^2 + (\theta_1 d_i E_i + \theta_2 d_i F_i - 2)X + 1 - \theta_1 d_i E_i - \theta_2 d_i F_i + \theta_1 \theta_2 \sigma_i^2 E_i F_i$. We then have the following result.

THEOREM 3.1. *Algorithm NN$_{1a}$ (2.1)-(2.3) converges if and only if*

(3.5)
$$\rho_{NN_{1a}} := \max_{d_i \in \lambda(A)} \left\{ \left| 1 - \frac{d_i(\theta_1 E_i + \theta_2 F_i) \pm \sqrt{d_i^2(\theta_1 E_i + \theta_2 F_i)^2 - 4\theta_1\theta_2 \sigma_i^2 E_i F_i}}{2} \right| \right\} < 1,$$

where $\lambda(A)$ is the spectrum of the matrix A.

To get more insight in the convergence factor (3.5), we consider a few special cases. Supposing no final target (i.e., $\gamma = 0$) and a symmetric decomposition $\alpha = \frac{T}{2}$ (i.e., $a_i = b_i$), we have $E_i = F_i = \frac{2d_i \tanh(a_i)+\sigma_i(1+\tanh^2(a_i))}{(\sigma_i^2+d_i^2)\tanh(a_i)+d_i\sigma_i(1+\tanh^2(a_i))} < \frac{1}{d_i}$. Letting $\theta_1 = \theta_2 = \theta$, the convergence factor (3.5) then becomes $|1 - \theta d_i E_i \pm \theta E_i \sqrt{d_i^2 - \sigma_i^2}|$, where the discriminant is negative due to $d_i^2 - \sigma_i^2 = -\nu^{-1}$. Thus, the convergence factor $\rho_{\mathrm{NN}_{1a}}$ in this case is $\sqrt{1 - 2\theta d_i E_i + \theta^2 \sigma_i^2 E_i^2} > \sqrt{1 - 2\theta + \theta^2 \sigma_i^2 E_i^2} \geq \sqrt{1 - 2\theta}$.

Remark 3.2. For the Laplace operator with homogeneous Dirichlet boundary conditions in our model problem (1.2), there is no zero eigenvalue for its discretization matrix A. For a zero eigenvalue, $d_i = 0$, we have from (2.5) that

$$\sigma_i|_{d_i=0} = \sqrt{\nu^{-1}}, \quad \omega_i|_{d_i=0} = \gamma\nu^{-1}, \quad \beta_i|_{d_i=0} = 1. \tag{3.6}$$

Substituting (3.6) into the convergence factor (3.5), we find $\rho_{\mathrm{NN}_{1a}}|_{d_i=0} = \{|1 \pm \sqrt{-\theta_1\theta_2(E_iF_i)|_{d_i=0}}|\}$ with $(E_iF_i)|_{d_i=0} = 2 + \coth(\sqrt{\nu^{-1}}\alpha)\frac{\coth(\sqrt{\nu^{-1}}(T-\alpha))+\gamma\sqrt{\nu^{-1}}}{1+\gamma\sqrt{\nu^{-1}}\coth(\sqrt{\nu^{-1}}(T-\alpha))} + \tanh(\sqrt{\nu^{-1}}\alpha)\frac{\tanh(\sqrt{\nu^{-1}}(T-\alpha))+\gamma\sqrt{\nu^{-1}}}{1+\gamma\sqrt{\nu^{-1}}\tanh(\sqrt{\nu^{-1}}(T-\alpha))}$. Since $(E_iF_i)|_{d_i=0}$, θ_1, θ_2 are all positive, the discriminant is once again negative, and we have $\rho_{\mathrm{NN}_{1a}}|_{d_i=0} = \sqrt{1 + \theta_1\theta_2(E_iF_i)|_{d_i=0}}$, which is always greater than one. In other words, the convergence behavior of algorithm NN_{1a} for small eigenvalues is not good, and cannot be fixed with relaxation.

Remark 3.3. For large eigenvalues d_i, we have from (2.5) that

$$\sigma_i \sim_\infty d_i, \quad \omega_i \sim_\infty d_i, \quad \beta_i \sim_\infty -d_i, \tag{3.7}$$

and thus obtain $E_i \sim_\infty \frac{1}{d_i}$ and $F_i \sim_\infty \frac{1}{d_i}$. Substituting these into (3.5), we find $\lim_{d_i\to\infty} \rho_{\mathrm{NN}_{1a}} = \{|1 - \theta_1|, |1 - \theta_2|\}$. In other words, high frequency convergence is robust with relaxation, and one can get a good smoother using $\theta_1 = \theta_2 = 1$.

The above analysis reveals the fact that this most natural NN algorithm is a good smoother but not a good solver.

3.1.2. Algorithm $\mathbf{NN_{1b}}$. We apply now the Neumann step only to the primal correction state ψ_i. For $k = 1, 2, \ldots$, we consider the algorithm that first solves the Dirichlet step (2.1), and then corrects it by solving the Neumann step

$$\begin{cases} \begin{pmatrix} \dot\psi_{1,i}^k \\ \dot\phi_{1,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} \psi_{1,i}^k \\ \phi_{1,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_1, \\ \psi_{1,i}^k(0) = 0, \\ \dot\psi_{1,i}^k(\alpha) = \dot z_{1,i}^k(\alpha) - \dot z_{2,i}^k(\alpha), \end{cases}$$
$$\begin{cases} \begin{pmatrix} \dot\psi_{2,i}^k \\ \dot\phi_{2,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} \psi_{2,i}^k \\ \phi_{2,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_2, \\ \dot\psi_{2,i}^k(\alpha) = \dot z_{2,i}^k(\alpha) - \dot z_{1,i}^k(\alpha), \\ \phi_{2,i}^k(T) + \gamma\psi_{2,i}^k(T) = 0. \end{cases} \tag{3.8}$$

As for the update step, let us first consider keeping the same update as (2.3).

Unlike the Dirichlet step (2.1), the Neumann step (3.8) does not have the forward-backward structure in the current form, but this can be recovered using the identities in (2.4). More precisely, we can rewrite the transmission condition $\dot\psi_{1,i}^k(\alpha) = \dot z_{1,i}^k(\alpha) - \dot z_{2,i}^k(\alpha)$ as $\dot\phi_{1,i}^k(\alpha) - \frac{\sigma_i^2}{d_i}\phi_{1,i}^k(\alpha) = (\dot\mu_{1,i}^k(\alpha) - \frac{\sigma_i^2}{d_i}\mu_{1,i}^k(\alpha)) - (\dot\mu_{2,i}^k(\alpha) - \frac{\sigma_i^2}{d_i}\mu_{2,i}^k(\alpha))$, which is a Robin type condition. In other words, when the forward-backward structure is recovered with this interpretation, the Neumann step (3.8) becomes a RN step.

Compared with algorithm NN_{1a}, only the Neumann step is modified, which can be transformed into

(3.9)
$$\begin{cases} \ddot\psi_{1,i}^k - \sigma_i^2\psi_{1,i}^k = 0 \text{ in } \Omega_1, \\ \psi_{1,i}^k(0) = 0, \\ \dot\psi_{1,i}^k(\alpha) = \dot z_{1,i}^k(\alpha) - \dot z_{2,i}^k(\alpha), \end{cases} \qquad \begin{cases} \ddot\psi_{2,i}^k - \sigma_i^2\psi_{2,i}^k = 0 \text{ in } \Omega_2, \\ \dot\psi_{2,i}^k(\alpha) = \dot z_{2,i}^k(\alpha) - \dot z_{1,i}^k(\alpha), \\ \dot\psi_{2,i}^k(T) + \omega_i\psi_{2,i}^k(T) = 0. \end{cases}$$

The convergence analysis is then given by solving explicitly (2.6), (3.9) and (2.8) for one step. In this form, we are actually analyzing here a RD step with a NN correction step. Using (3.1), we can solve (3.9) and determine the coefficients

(3.10)
$$C_i^k = A_i^k + B_i^k\frac{\sigma_i\sinh(b_i)+\omega_i\cosh(b_i)}{\cosh(a_i)},\ D_i^k = A_i^k\frac{\cosh(a_i)}{\sigma_i\sinh(b_i)+\omega_i\cosh(b_i)} + B_i^k.$$

Combining with (3.2), we update the transmission condition (2.8) and find

$$\begin{pmatrix} f_{\alpha,i}^k \\ g_{\alpha,i}^k \end{pmatrix} = \begin{pmatrix} 1-\theta_1 d_i E_i & -\theta_1 d_i F_i \\ -\theta_2 E_i & 1-\theta_2 F_i \end{pmatrix} \begin{pmatrix} f_{\alpha,i}^{k-1} \\ g_{\alpha,i}^{k-1} \end{pmatrix}, \tag{3.11}$$

with $E_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{\sigma_i\sinh(b_i)+\omega_i\cosh(b_i)}\frac{1}{\sigma_i\cosh(a_i)+d_i\sinh(a_i)}$ and $F_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{\sigma_i\cosh(b_i)+\omega_i\sinh(b_i)}\frac{1}{\cosh(a_i)}$. In particular, the eigenvalues of the iteration matrix in (3.11) are 1 and $1-(\theta_1 d_i E_i + \theta_2 F_i)$, meaning that the algorithm (2.1), (3.8), (2.3) stagnates in its current form, and cannot be fixed even with relaxation.

Note that we choose to keep the same Dirichlet and update steps in the algorithm (2.1), (3.8), (2.3), although the Neumann step has been changed comparing to algorithm NN_{1a}. We also observe from the Neumann correction step (3.8) that $\dot\psi_{1,i}^k(\alpha)+\dot\psi_{2,i}^k(\alpha)=0$, which implies that in this case, the update step (2.3) in terms of the primal correction state (2.8) is actually

$$f_{\alpha,i}^k = f_{\alpha,i}^{k-1} - \theta_1 d_i\big(\psi_{1,i}^k(\alpha)+\psi_{2,i}^k(\alpha)\big), \quad g_{\alpha,i}^k = g_{\alpha,i}^{k-1} - \theta_2\big(\psi_{1,i}^k(\alpha)+\psi_{2,i}^k(\alpha)\big). \tag{3.12}$$

In other words, we update both $f_{\alpha,i}^k$ and $g_{\alpha,i}^k$ only by $\psi_i^k(\alpha)$. This observation leads to the idea to consider a modified NN algorithm. More precisely, we first remove d_i in (3.12) as

$$f_{\alpha,i}^k = f_{\alpha,i}^{k-1} - \theta_1(\psi_{1,i}^k(\alpha)+\psi_{2,i}^k(\alpha)), \quad g_{\alpha,i}^k = g_{\alpha,i}^{k-1} - \theta_2(\psi_{1,i}^k(\alpha)+\psi_{2,i}^k(\alpha)). \tag{3.13}$$

In the case when $f_{\alpha,i}^0 = g_{\alpha,i}^0$ and $\theta_1=\theta_2=\theta$, we have $f_{\alpha,i}^k = g_{\alpha,i}^k$, $\forall k\in\mathbb{N}$. In this way, we consider the modified NN algorithm which solves first the Dirichlet step

$$\begin{cases} \begin{pmatrix} \dot z_{1,i}^k \\ \dot\mu_{1,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix}\begin{pmatrix} z_{1,i}^k \\ \mu_{1,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_1, \\ z_{1,i}^k(0) = 0, \\ \mu_{1,i}^k(\alpha) = f_{\alpha,i}^{k-1}, \end{cases}$$
$$\begin{cases} \begin{pmatrix} \dot z_{2,i}^k \\ \dot\mu_{2,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix}\begin{pmatrix} z_{2,i}^k \\ \mu_{2,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_2, \\ z_{2,i}^k(\alpha) = f_{\alpha,i}^{k-1}, \\ \mu_{2,i}^k(T) + \gamma z_{2,i}^k(T) = 0, \end{cases} \tag{3.14}$$

then corrects the result by solving the Neumann step (3.8) and updates the transmission condition by

$$f_{\alpha,i}^k = f_{\alpha,i}^{k-1} - \theta(\psi_{1,i}^k(\alpha) + \psi_{2,i}^k(\alpha)), \quad \theta > 0. \tag{3.15}$$

For this modified NN algorithm, we find the following result.

THEOREM 3.4. *Algorithm NN_{1b} (3.14), (3.8), (3.15) converges if and only if*

$$\rho_{NN_{1b}} := \max_{d_i \in \lambda(A)} \left|1 - \theta(E_i + F_i)\right| < 1. \tag{3.16}$$

Compared to the algorithm (2.1), (3.8), (2.3), algorithm $\mathrm{NN}_{1\mathrm{b}}$ converges with a proper choice of θ. More precisely, for a zero eigenvalue, substituting (3.6) into (3.16), we find $\rho_{\mathrm{NN}_{1\mathrm{b}}}|_{d_i=0} = |1-\theta(\sqrt{\nu}(\tanh(\sqrt{\nu^{-1}}\alpha)+\frac{1+\gamma\sqrt{\nu^{-1}}\tanh(\sqrt{\nu^{-1}}(T-\alpha))}{\gamma\sqrt{\nu^{-1}}+\tanh(\sqrt{\nu^{-1}}(T-\alpha))})+1+\tanh(\sqrt{\nu^{-1}}\alpha)\frac{\gamma\sqrt{\nu^{-1}}+\tanh(\sqrt{\nu^{-1}}(T-\alpha))}{1+\gamma\sqrt{\nu^{-1}}\tanh(\sqrt{\nu^{-1}}(T-\alpha))})|$, meaning that small eigenvalue convergence is good with relaxation. For large eigenvalues d_i, using (3.7), we have $E_i \sim_\infty \frac{1}{d_i}$ and $F_i \sim_\infty 2$. Thus, we obtain $\lim_{d_i\to\infty} \rho_{\mathrm{NN}_{1\mathrm{b}}} = |1-2\theta|$, which is independent of the interface α. So high frequency convergence is robust with relaxation, and one can get a good smoother using $\theta = 1/2$. By equioscillating the convergence factor for small (i.e., $\rho_{\mathrm{NN}_{1\mathrm{b}}}|_{d_i=0}$) and large (i.e., $\rho_{\mathrm{NN}_{1\mathrm{b}}}|_{d_i\to\infty}$) eigenvalues, we obtain

$$\theta^*_{\mathrm{NN}_{1\mathrm{b}}} := \frac{2}{3+\sqrt{\nu}(\tanh(\sqrt{\nu^{-1}}\alpha)+\frac{1+\gamma\sqrt{\nu^{-1}}\tanh(\sqrt{\nu^{-1}}(T-\alpha))}{\gamma\sqrt{\nu^{-1}}+\tanh(\sqrt{\nu^{-1}}(T-\alpha))})+\tanh(\sqrt{\nu^{-1}}\alpha)\frac{\gamma\sqrt{\nu^{-1}}+\tanh(\sqrt{\nu^{-1}}(T-\alpha))}{1+\gamma\sqrt{\nu^{-1}}\tanh(\sqrt{\nu^{-1}}(T-\alpha))}}, \tag{3.17}$$

which is smaller than $2/3$. However, it is not clear under what condition $\theta^*_{\mathrm{NN}_{1\mathrm{b}}}$ is the optimal relaxation parameter. Indeed, the monotonicity of E_i and F_i with respect to d_i may change according to the parameter values α, γ and ν. Thus, the variation of $E_i + F_i$ to d_i is less clear even in the case with $\gamma = 0$. Generally, algorithm $\mathrm{NN}_{1\mathrm{b}}$ is a good smoother and can also be a good solver with a proper relaxation parameter θ.

Remark 3.5. Instead of considering the update step as in (3.13), we could have also modified (3.12) to $f_{\alpha,i}^k = f_{\alpha,i}^{k-1} - \theta_1 d_i(\psi_{1,i}^k(\alpha) + \psi_{2,i}^k(\alpha))$ and $g_{\alpha,i}^k = g_{\alpha,i}^{k-1} - \theta_2 d_i(\psi_{1,i}^k(\alpha) + \psi_{2,i}^k(\alpha))$. Using then the same arguments as above, we end up with $g_{\alpha,i}^k \equiv f_{\alpha,i}^k = f_{\alpha,i}^{k-1}(1 - \theta d_i(E_i + F_i))$. However, the convergence of the algorithm can no longer be guaranteed with this update. More precisely, for a zero eigenvalue $d_i = 0$, the convergence factor is one, and cannot be improved with relaxation. As for large eigenvalues, using once again the equivalence relation of E_i and F_i, we find the convergence factor goes to infinity when d_i is large.

In general, the above analysis shows that the update step should also be adapted when modifying the Neumann step.

3.1.3. Algorithm $\mathbf{NN_{1c}}$. Instead of applying the Neumann step to the primal correction state ψ_i, we can also apply it only to the dual correction state ϕ_i. For $k = 1, 2, \ldots$, we consider the algorithm that first solves the Dirichlet step (2.1), then corrects it by solving the Neumann step

$$
(3.18)\quad
\begin{cases}
\begin{pmatrix}\dot\psi^k_{1,i}\\ \dot\phi^k_{1,i}\end{pmatrix}+\begin{pmatrix}d_i & -\nu^{-1}\\ -1 & -d_i\end{pmatrix}\begin{pmatrix}\psi^k_{1,i}\\ \phi^k_{1,i}\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix} \text{ in } \Omega_1,\\
\psi^k_{1,i}(0)=0,\\
\dot\phi^k_{1,i}(\alpha)=\dot\mu^k_{1,i}(\alpha)-\dot\mu^k_{2,i}(\alpha),
\end{cases}
\qquad
\begin{cases}
\begin{pmatrix}\dot\psi^k_{2,i}\\ \dot\phi^k_{2,i}\end{pmatrix}+\begin{pmatrix}d_i & -\nu^{-1}\\ -1 & -d_i\end{pmatrix}\begin{pmatrix}\psi^k_{2,i}\\ \phi^k_{2,i}\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix} \text{ in } \Omega_2,\\
\dot\phi^k_{2,i}(\alpha)=\dot\mu^k_{2,i}(\alpha)-\dot\mu^k_{1,i}(\alpha),\\
\phi^k_{2,i}(T)+\gamma\psi^k_{2,i}(T)=0.
\end{cases}
$$

Once again, let us first consider keeping the same update step (2.3).

The Neumann step (3.18) does not seem to have the forward-backward structure due to the transmission condition on the second domain Ω_2. Using (2.4), we can rewrite it as $\dot\psi^k_{2,i}(\alpha)+\frac{\sigma_i^2}{d_i}\psi^k_{2,i}(\alpha)=(\dot z^k_{2,i}(\alpha)+\frac{\sigma_i^2}{d_i}z^k_{2,i}(\alpha))-(\dot z^k_{1,i}(\alpha)+\frac{\sigma_i^2}{d_i}z^k_{1,i}(\alpha))$, which then becomes a NR step with the usual forward-backward structure.

Once again, only the Neumann step is modified and can be transformed into

$$
(3.19)\quad
\begin{cases}
\ddot\psi^k_{1,i}-\sigma_i^2\psi^k_{1,i}=0 \text{ in } \Omega_1,\\
\psi^k_{1,i}(0)=0,\\
\dot\psi^k_{1,i}(\alpha)+\dfrac{\sigma_i^2}{d_i}\psi^k_{1,i}(\alpha)=\left(\dot z^k_{1,i}(\alpha)+\dfrac{\sigma_i^2}{d_i}z^k_{1,i}(\alpha)\right)-\left(\dot z^k_{2,i}(\alpha)+\dfrac{\sigma_i^2}{d_i}z^k_{2,i}(\alpha)\right),
\end{cases}
\qquad
\begin{cases}
\ddot\psi^k_{2,i}-\sigma_i^2\psi^k_{2,i}=0 \text{ in } \Omega_2,\\
\dot\psi^k_{2,i}(\alpha)+\dfrac{\sigma_i^2}{d_i}\psi^k_{2,i}(\alpha)=\left(\dot z^k_{2,i}(\alpha)+\dfrac{\sigma_i^2}{d_i}z^k_{2,i}(\alpha)\right)-\left(\dot z^k_{1,i}(\alpha)+\dfrac{\sigma_i^2}{d_i}z^k_{1,i}(\alpha)\right),\\
\dot\psi^k_{2,i}(T)+\omega_i\psi^k_{2,i}(T)=0.
\end{cases}
$$

The convergence analysis is thus given for a RD step (2.6) with a RR correction step (3.19). We can solve (3.19) using (3.1) and determine the coefficients

$$
(3.20)\quad C^k_i=A^k_i-B^k_i\nu^{-1}\frac{\sigma_i\gamma\sinh(b_i)+\beta_i\cosh(b_i)}{\sigma_i\sinh(a_i)+d_i\cosh(a_i)},\; D^k_i=B^k_i-\nu A^k_i\frac{\sigma_i\sinh(a_i)+d_i\cosh(a_i)}{\sigma_i\gamma\sinh(b_i)+\beta_i\cosh(b_i)}.
$$

Combining with (3.2), we update the transmission condition (2.8) and find

$$
(3.21)\quad \begin{pmatrix}f^k_{\alpha,i}\\ g^k_{\alpha,i}\end{pmatrix}=\begin{pmatrix}1-\theta_1E_i & \theta_1\nu^{-1}F_i\\ \theta_2\nu d_iE_i & 1-\theta_2d_iF_i\end{pmatrix}\begin{pmatrix}f^{k-1}_{\alpha,i}\\ g^{k-1}_{\alpha,i}\end{pmatrix},
$$

with $E_i=\frac{\sigma_i\cosh(\sigma_iT)+\omega_i\sinh(\sigma_iT)}{\sigma_i\gamma\sinh(b_i)+\beta_i\cosh(b_i)}\frac{1}{\sigma_i\cosh(a_i)+d_i\sinh(a_i)}$ and $F_i=\frac{\sigma_i\cosh(\sigma_iT)+\omega_i\sinh(\sigma_iT)}{\sigma_i\cosh(b_i)+\omega_i\sinh(b_i)}\frac{1}{\sigma_i\sinh(a_i)+d_i\cosh(a_i)}$. In particular, the eigenvalues of the iteration matrix in (3.21) are 1 and $1-(\theta_1E_i+\theta_2d_iF_i)$. Once again, the algorithm (2.1), (3.18), (2.3) stagnates, and cannot be fixed with relaxation. Similar as in Section 3.1.2, we can adapt the transmission condition (2.3) and make this algorithm converge. More precisely, we first consider the update $f^k_{\alpha,i}=f^{k-1}_{\alpha,i}-\theta(\phi^k_{1,i}(\alpha)+\phi^k_{2,i}(\alpha))$ and $g^k_{\alpha,i}=g^{k-1}_{\alpha,i}-\theta(\phi^k_{1,i}(\alpha)+\phi^k_{2,i}(\alpha))$. In the case when $f^0_{\alpha,i}=g^0_{\alpha,i}$ and $\theta_1=\theta_2=\theta$, we have $g^k_{\alpha,i}=f^k_{\alpha,i}$, $\forall k\in\mathbb{N}$ and

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

This leads to the following result.

THEOREM 3.6. *Algorithm NN_{1c} (3.14), (3.18), (3.22) converges if and only if*

$$\rho_{NN_{1c}} := \max_{d_i\in\lambda(A)} |1-\theta(E_i-\nu^{-1}F_i)| < 1. \tag{3.23}$$

Compared to the algorithm (2.1), (3.18), (2.3), algorithm $\mathrm{NN_{1c}}$ may converge with a proper choice of θ. More precisely, for a zero eigenvalue, $d_i = 0$, we find $\rho_{\mathrm{NN_{1c}}}|_{d_i=0} = |1-\theta(1+\tanh(\sqrt{\nu^{-1}}\alpha)\,\frac{\gamma\sqrt{\nu^{-1}}+\tanh(\sqrt{\nu^{-1}}(T-\alpha))}{\gamma\sqrt{\nu^{-1}}\tanh(\sqrt{\nu^{-1}}(T-\alpha))+1} - \sqrt{\nu^{-1}}(\coth(\sqrt{\nu^{-1}}\alpha) + \frac{\gamma\sqrt{\nu^{-1}}+\tanh(\sqrt{\nu^{-1}}(T-\alpha))}{1+\gamma\sqrt{\nu^{-1}}\tanh(\sqrt{\nu^{-1}}(T-\alpha))}))|$. Depending on the values of ν, γ and α, $(E_i-\nu^{-1}F_i)|_{d_i=0}$ could be negative, then $\rho_{\mathrm{NN_{1c}}}|_{d_i=0}$ would be greater than one since $\theta > 0$. In other words, the convergence for small eigenvalues could be not good, and cannot be fixed even with relaxation. For large eigenvalues d_i, using (3.7), we find $E_i \sim_\infty 2$ and $F_i \sim_\infty \frac{1}{d_i}$. Thus, we obtain $\lim_{d_i\to\infty}\rho_{\mathrm{NN_{1c}}} = |1-2\theta|$, which is independent of the interface α. So large eigenvalue convergence is robust with relaxation, and one can get a good smoother using $\theta = 1/2$. Moreover, we observe that algorithms $\mathrm{NN_{1b}}$ and $\mathrm{NN_{1c}}$ share similar behavior for large eigenvalues. By equioscillating the convergence factor for small (i.e., $\rho_{\mathrm{NN_{1c}}}|_{d_i=0}$) and large (i.e., $\rho_{\mathrm{NN_{1c}}}|_{d_i\to\infty}$) eigenvalues, we obtain

$$\theta^*_{\mathrm{NN_{1c}}} := \frac{2}{3+\tanh(\sqrt{\nu^{-1}}\alpha)\,\frac{\gamma\sqrt{\nu^{-1}}+\tanh(\sqrt{\nu^{-1}}(T-\alpha))}{\gamma\sqrt{\nu^{-1}}\tanh(\sqrt{\nu^{-1}}(T-\alpha))+1} - \sqrt{\nu^{-1}}(\coth(\sqrt{\nu^{-1}}\alpha) + \frac{\gamma\sqrt{\nu^{-1}}+\tanh(\sqrt{\nu^{-1}}(T-\alpha))}{1+\gamma\sqrt{\nu^{-1}}\tanh(\sqrt{\nu^{-1}}(T-\alpha))})}. \tag{3.24}$$

Note that when $(E_i-\nu^{-1}F_i)|_{d_i=0} < 0$, the relaxation cannot improve the convergence for small eigenvalues, thus, (3.24) could also be negative and cannot provide the optimal value of θ in this case. One may use however a negative relaxation parameter θ to make the algorithm converge for small eigenvalues, but this will induce divergence for large eigenvalues. Based on the analysis, algorithm $\mathrm{NN_{1c}}$ is a good smoother but not necessarily a good solver.

Remark 3.7. One could also consider the update step (3.15) instead of (3.22), and the convergence factor (3.23) will be $\max_{d_i\in\lambda(A)}|1-\theta d_i(F_i-\nu E_i)|$. For a similar reason as in Remark 3.5, the algorithm diverges with this choice of update step.

Together with the analysis in Section 3.1.2, we observe that keeping the same update step (2.3) leads to divergent algorithms, when modifying the Neumann step. Thus, we should also adapt the update step according to the Neumann step.

3.2. Category II. We now study the algorithms in Category II which run the Dirichlet step only on the primal state z_i.

3.2.1. Algorithm $\mathbf{NN_{2a}}$. The most natural way is to correct z_i by the primal correction state ψ_i. For $k = 1, 2, \ldots$, algorithm $\mathrm{NN_{2a}}$ first solves the Dirichlet step

$$\begin{cases} \begin{pmatrix}\dot z^k_{1,i}\\ \dot\mu^k_{1,i}\end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1}\\ -1 & -d_i\end{pmatrix}\begin{pmatrix} z^k_{1,i}\\ \mu^k_{1,i}\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix} \text{ in } \Omega_1,\\ z^k_{1,i}(0) = 0,\\ z^k_{1,i}(\alpha) = f^{k-1}_{\alpha,i}, \end{cases}$$
$$\begin{cases} \begin{pmatrix}\dot z^k_{2,i}\\ \dot\mu^k_{2,i}\end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1}\\ -1 & -d_i\end{pmatrix}\begin{pmatrix} z^k_{2,i}\\ \mu^k_{2,i}\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix} \text{ in } \Omega_2,\\ z^k_{2,i}(\alpha) = f^{k-1}_{\alpha,i},\\ \mu^k_{2,i}(T) + \gamma z^k_{2,i}(T) = 0, \end{cases} \tag{3.25}$$

then corrects the result by solving the Neumann step (3.8), and updates the transmission condition by (3.15)

Remark 3.8. Here, it is more natural to consider the transmission condition only for $f_{\alpha,i}^k$. This is due to the continuity of the primal state z_i^k at the interface α. In general, we can show that an update step as (3.22) will lead to divergence for a similar reason as in Remark 3.5. We can also show that a pair of transmission conditions $(f_{\alpha,i}^k, g_{\alpha,i}^k)$ will lead to non-convergent behavior (see Appendix A).

For algorithm $\mathrm{NN_{2a}}$, neither the Dirichlet (3.25) nor the Neumann step (3.8) has the forward-backward structure in its current form. We have seen in Section 3.1.2 that we can recover this structure for the Neumann step (3.8) which becomes a RN step. Using the same idea, we can interpret $z_{1,i}^k(\alpha) = f_{\alpha,i}^{k-1}$ as $\dot{\mu}_{1,i}^k(\alpha) - d_i\mu_{1,i}^k(\alpha) = f_{\alpha,i}^{k-1}$ to recover the forward-backward structure, and the Dirichlet step (3.25) then becomes a ND step.

For the convergence analysis, we transform the Dirichlet step (3.25) using (2.4) and (2.5), and find

$$\text{(3.26)}\qquad \begin{cases} \ddot{z}_{1,i}^k - \sigma_i^2 z_{1,i}^k = 0 \text{ in } \Omega_1,\\ z_{1,i}^k(0) = 0,\\ z_{1,i}^k(\alpha) = f_{\alpha,i}^{k-1}, \end{cases} \qquad \begin{cases} \ddot{z}_{2,i}^k - \sigma_i^2 z_{2,i}^k = 0 \text{ in } \Omega_2,\\ z_{2,i}^k(\alpha) = f_{\alpha,i}^{k-1},\\ \dot{z}_{2,i}^k(T) + \omega_i z_{2,i}^k(T) = 0. \end{cases}$$

The Neumann step becomes (3.9), and we keep the same update step (3.15). In particular, the convergence analysis also proceeds on a NN algorithm (3.26), (3.9), (3.15). Using (3.1), we can solve (3.26) and determine the coefficients,

$$\text{(3.27)}\qquad A_i^k = \frac{f_{\alpha,i}^{k-1}}{\sinh(a_i)}, \quad B_i^k = \frac{f_{\alpha,i}^{k-1}}{\sigma_i\cosh(b_i) + \omega_i\sinh(b_i)}.$$

Combining them with (3.10), we update the transmission condition (3.15) and find $f_{\alpha,i}^k = f_{\alpha,i}^{k-1} - \theta f_{\alpha,i}^{k-1}(E_i + F_i)$, with $E_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{(\sigma_i\sinh(b_i)+\omega_i\cosh(b_i))\sinh(a_i)}$ and $F_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{(\sigma_i\cosh(b_i)+\omega_i\sinh(b_i))\cosh(a_i)}$. This leads to the following result.

THEOREM 3.9. *Algorithm* NN_{2a} (3.25), (3.8), (3.15) *converges if and only if*

$$\text{(3.28)}\qquad \rho_{NN_{2a}} := \max_{d_i\in\lambda(A)} |1 - \theta(E_i + F_i)| < 1.$$

In particular, for a zero eigenvalue, substituting (3.6) into (3.28), we have

$$\text{(3.29)}\qquad \begin{aligned} \rho_{\mathrm{NN_{2a}}}|_{d_i=0} = \Big|1 - \theta\Big(2 + &\coth(\sqrt{\nu^{-1}}\alpha)\frac{\coth\left(\sqrt{\nu^{-1}}(T-\alpha)\right) + \gamma\sqrt{\nu^{-1}}}{1+\gamma\sqrt{\nu^{-1}}\coth\left(\sqrt{\nu^{-1}}(T-\alpha)\right)} \\ &+ \tanh(\sqrt{\nu^{-1}}\alpha)\frac{\tanh\left(\sqrt{\nu^{-1}}(T-\alpha)\right) + \gamma\sqrt{\nu^{-1}}}{1+\gamma\sqrt{\nu^{-1}}\tanh\left(\sqrt{\nu^{-1}}(T-\alpha)\right)}\Big)\Big|. \end{aligned}$$

For large eigenvalues d_i, using (3.7), we find $E_i \sim_\infty 2$ and $F_i \sim_\infty 2$. Thus, we obtain $\lim_{d_i\to\infty}\rho_{\mathrm{NN_{2a}}} = |1-4\theta|$, which is independent of the interface α. So the convergence for high frequencies is robust with relaxation, and one can get a good smoother using $\theta = 1/4$. By equioscillating the convergence factor for small (i.e., $\rho_{\mathrm{NN_{2a}}}|_{d_i=0}$) and large (i.e., $\rho_{\mathrm{NN_{2a}}}|_{d_i\to\infty}$) eigenvalues, we obtain the relaxation parameter

$$\text{(3.30)}\qquad \theta_{\mathrm{NN_{2a}}}^* := \frac{2}{6+\coth(\sqrt{\nu^{-1}}\alpha)\frac{\coth(\sqrt{\nu^{-1}}(T-\alpha))+\gamma\sqrt{\nu^{-1}}}{1+\gamma\sqrt{\nu^{-1}}\coth(\sqrt{\nu^{-1}}(T-\alpha))}+\tanh(\sqrt{\nu^{-1}}\alpha)\frac{\tanh(\sqrt{\nu^{-1}}(T-\alpha))+\gamma\sqrt{\nu^{-1}}}{1+\gamma\sqrt{\nu^{-1}}\tanh(\sqrt{\nu^{-1}}(T-\alpha))}},$$

which is smaller than 1/3. In the case with no final state, i.e., $\gamma = 0$, we have $\theta^*_{\mathrm{NN_{2a}}}|_{\gamma=0} = \frac{2}{6+\coth(\sqrt{\nu^{-1}}\alpha)\coth(\sqrt{\nu^{-1}}(T-\alpha))+\tanh(\sqrt{\nu^{-1}}\alpha)\tanh(\sqrt{\nu^{-1}}(T-\alpha))}$. Using properties of the hyperbolic tangent and cotangent, we find $\coth(\sqrt{\nu^{-1}}\alpha)\coth(\sqrt{\nu^{-1}}(T-\alpha)) + \tanh(\sqrt{\nu^{-1}}\alpha)\tanh(\sqrt{\nu^{-1}}(T-\alpha)) \geq \coth^2(\sqrt{\nu^{-1}}\frac{T}{2}) + \tanh^2(\sqrt{\nu^{-1}}\frac{T}{2}) > 2$, thus $\theta^*_{\mathrm{NN_{2a}}} < \frac{1}{4}$. Based on the analysis, algorithm $\mathrm{NN_{2a}}$ is a good smoother and can also be a good solver. However, it is less clear under what condition $\theta^*_{\mathrm{NN_{2a}}}$ is the optimal relaxation parameter, since the monotonicity of the convergence factor with respect to the eigenvalues d_i is not clear even in the case $\gamma = 0$. This has been observed in our numerical experiments.

3.2.2. Algorithm $\mathbf{NN_{2b}}$. We can also keep the Dirichlet step (3.25), but apply the Neumann step only to the dual correction state ϕ_i as in (3.18). As for the update step, we first consider to take the same update as for algorithm $\mathrm{NN_{2a}}$, i.e., (3.15).

For the convergence analysis, we actually solve a DD step (3.26) and correct by a RR step (3.19). Using (3.27) and (3.20), we update the transmission condition (3.15) and find $f^k_{\alpha,i} = f^{k-1}_{\alpha,i}(1-\theta d_i(F_i - \nu E_i))$ with $E_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{\sigma_i\gamma\sinh(b_i)+\beta_i\cosh(b_i)}\frac{1}{\sinh(a_i)}$ and $F_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{(\sigma_i\cosh(b_i)+\omega_i\sinh(b_i))(\sigma_i\sinh(a_i)+d_i\cosh(a_i))}$. We then obtain the convergence factor

$$\rho_{\mathrm{NN_{2b}}} := \max_{d_i\in\lambda(A)} |1-\theta d_i(F_i-\nu E_i)| < 1. \tag{3.31}$$

To get more insight, we first study the extreme cases. For a zero eigenvalue, $d_i = 0$, substituting (3.6) into (3.31), we have $(F_i - \nu E_i)|_{d_i=0} = 0$. Hence, we find $\rho_{\mathrm{NN_{2b}}}|_{d_i=0} = 1$, which is independent of the relaxation parameter. In other words, the convergence behavior of algorithm $\mathrm{NN_{2b}}$ is not good for small eigenvalues, and the relaxation cannot fix this problem. For large eigenvalues d_i, using (3.7), we find $E_i \sim_\infty 4d_i$ and $F_i \sim_\infty \frac{1}{d_i}$. Thus, we obtain $1-\theta d_i(F_i - \nu E_i) \sim_\infty 4\nu\theta d_i^2$ and $\lim_{d_i\to\infty}\rho_{\mathrm{NN_{2b}}} = \infty$, which is divergent, and cannot be fixed with relaxation. Generally, we have the following result.

THEOREM 3.10. *Algorithm NN_{2b} (3.25) (3.18) (3.15) always diverges.*

Proof. Using the formula of E_i and F_i, we find $F_i - \nu E_i = \frac{-\nu d_i}{\sigma_i\sinh(a_i)+d_i\cosh(a_i)} \frac{(\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T))^2}{\sinh(a_i)(\sigma_i\gamma\sinh(b_i)+\beta_i\cosh(b_i))(\sigma_i\cosh(b_i)+\omega_i\sinh(b_i))}$ which is negative or zero (if $d_i = 0$). Since θ and ν are both positive, $1-\theta d_i(F_i-\nu E_i) \geq 1$ which concludes the proof. ∎

The above result shows that algorithm $\mathrm{NN_{2b}}$ diverges with a positive relaxation parameter θ. Moreover, this divergence cannot be fixed even with a negative θ, since the convergence factor is one for a zero eigenvalue, and is equivalent to $4\nu|\theta|d_i^2$ for large eigenvalues. In general, algorithm $\mathrm{NN_{2b}}$ is neither a good smoother nor a good solver.

Remark 3.11. Compared with algorithm $\mathrm{NN_{2a}}$, we change the Neumann step but keep the same update step. One can also consider the update step (3.22), since the Neumann correction (3.18) is only applied to the dual correction state ϕ_i. Following the same computation, the convergence factor (3.31) then becomes $\max_{d_i\in\lambda(A)}|1-\theta(E_i - \nu^{-1}F_i)|$ with $E_i - \nu^{-1}F_i \geq 0$. However, this does not change the poor convergence behavior for both small and large eigenvalues. Indeed, we still have $(E_i - \nu^{-1}F_i)|_{d_i=0} = 0$, hence $\rho_{\mathrm{NN_{2b}}}|_{d_i=0} = 1$, and $\lim_{d_i\to\infty}\rho_{\mathrm{NN_{2b}}} = \infty$. Thus, the modified algorithm stays divergent. Furthermore, for a similar reason as mentioned in Appendix A, the algorithm is also divergent when considering the update step (2.3) with a pair of transmission conditions $(f^k_{\alpha,i}, g^k_{\alpha,i})$.

Based on the analysis, we cannot find a good NN algorithm when combining the Dirichlet step (3.25) with the Neumann step (3.18).

3.2.3. Algorithm $\mathbf{NN_{2c}}$. If we apply the correction to the pair (ψ_i, ϕ_i), then the Neumann step immediately has the forward-backward structure. In this way, algorithm $\mathrm{NN_{2c}}$ solves first the Dirichlet step (3.25), next the Neumann step (2.2) and updates the transmission condition by (3.15).

For the convergence analysis, we solve a DD step (3.26) followed by a RN correction step (2.7). Using (3.27) and (3.3), we update the transmission condition (3.15) and find $f_{\alpha,i}^k = f_{\alpha,i}^{k-1}(1-\theta(E_i + d_i F_i))$ with $E_i = \frac{\sigma_i \cosh(\sigma_i T)+\omega_i \sinh(\sigma_i T)}{(\sigma_i \sinh(b_i)+\omega_i \cosh(b_i))\sinh(a_i)}$ and $F_i = \frac{\sigma_i \cosh(\sigma_i T)+\omega_i \sinh(\sigma_i T)}{(\sigma_i \cosh(b_i)+\omega_i \sinh(b_i))(\sigma_i \sinh(a_i)+d_i \cosh(a_i))}$. We then obtain the following result.

THEOREM 3.12. *Algorithm NN_{2c}* (3.25), (2.2), (3.15) *converges if and only if*

$$\rho_{NN_{2c}} := \max_{d_i \in \lambda(A)} |1-\theta(E_i + d_i F_i)| < 1. \tag{3.32}$$

For a zero eigenvalue $d_i = 0$, substituting the identities (3.6) into (3.32), we find

$$\rho_{\mathrm{NN_{2c}}}|_{d_i=0} = \Big|1-\theta\big(1+\coth(\sqrt{\nu^{-1}}\alpha)\frac{\coth(\sqrt{\nu^{-1}}(T-\alpha))+\gamma\sqrt{\nu^{-1}}}{1+\gamma\sqrt{\nu^{-1}}\coth(\sqrt{\nu^{-1}}(T-\alpha))}\big)\Big|. \tag{3.33}$$

For large eigenvalues d_i, using (3.7), we find $E_i \sim_\infty 2$ and $F_i \sim_\infty \frac{1}{d_i}$. Thus, we obtain $\lim_{d_i\to\infty} \rho_{\mathrm{NN_{2c}}} = |1-3\theta|$, which is independent of the interface α. So the convergence for high frequencies is robust with relaxation, and one can get a good smoother using $\theta = 1/3$. By equioscillating the convergence factor for small (i.e., $\rho_{\mathrm{NN_{2c}}}|_{d_i=0}$) and large (i.e., $\rho_{\mathrm{NN_{2c}}}|_{d_i\to\infty}$) eigenvalues, we obtain

$$\theta^*_{\mathrm{NN_{2c}}} := \frac{2}{4+\coth(\sqrt{\nu^{-1}}\alpha)\frac{\coth(\sqrt{\nu^{-1}}(T-\alpha))+\gamma\sqrt{\nu^{-1}}}{1+\gamma\sqrt{\nu^{-1}}\coth(\sqrt{\nu^{-1}}(T-\alpha))}}, \tag{3.34}$$

which is smaller than $1/2$. In the case $\gamma = 0$, the relaxation parameter $\theta^*_{\mathrm{NN_{2c}}}$ is bounded by $2/5$. However, it is also not clear under what condition $\theta^*_{\mathrm{NN_{2c}}}$ is the optimal relaxation parameter, since the monotonicity of $E_i + d_i F_i$ with respect to d_i is less clear, and depends on the parameter values α, γ and ν. Generally, algorithm $\mathrm{NN_{2c}}$ is both a good smoother and a good solver with a well-chosen θ.

Remark 3.13. Instead of choosing (3.15) as the update step, one could have considered the update step (3.22). Following the same computation, the convergence factor becomes $\max_{d_i\in\lambda(A)} |1-\theta(d_i E_i - \nu^{-1} F_i)|$, which diverges for large eigenvalues. Furthermore, the algorithm will also be divergent when considering the update step (2.3) with a pair transmission conditions $(f_{\alpha,i}^k, g_{\alpha,i}^k)$ as mentioned in Appendix A.

3.3. Category III. The algorithms in Category III run the Dirichlet step only on the dual state μ_i, and according to the Neumann step, there are three variants.

3.3.1. Algorithm $\mathbf{NN_{3a}}$. As in Section 3.2.1, the most natural way is to correct the dual state μ_i only by the dual correction state ϕ_i. In this way, for $k = 1, 2, ...,$

algorithm $\mathrm{NN}_{3\mathrm{a}}$ first solves the Dirichlet step

$$\begin{cases} \begin{pmatrix} \dot{z}_{1,i}^k \\ \dot{\mu}_{1,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{1,i}^k \\ \mu_{1,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_1, \\ z_{1,i}^k(0) = 0, \\ \mu_{1,i}^k(\alpha) = f_{\alpha,i}^{k-1}, \end{cases}$$
$$\begin{cases} \begin{pmatrix} \dot{z}_{2,i}^k \\ \dot{\mu}_{2,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{2,i}^k \\ \mu_{2,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_2, \\ \mu_{2,i}^k(\alpha) = f_{\alpha,i}^{k-1}, \\ \mu_{2,i}^k(T) + \gamma z_{2,i}^k(T) = 0, \end{cases} \tag{3.35}$$

then corrects the above result by solving the Neumann step (3.18), and updates the transmission condition by (3.22).

Similar to Remark 3.8, we choose here the update step (3.22) because of the continuity of the dual state μ_i^k at the interface α, since other choices of the update step will induce divergence behavior. Regarding the forward-backward structure for the Dirichlet step (3.35), we can recover it by interpreting $\mu_{2,i}^k(\alpha) = f_{\alpha,i}^{k-1}$ as $\dot{z}_{2,i}^k(\alpha) + d_i z_{2,i}^k(\alpha) = f_{\alpha,i}^{k-1}$. The Dirichlet step (3.35) then becomes a NR step.

To analyze algorithm $\mathrm{NN}_{3\mathrm{a}}$, we can rewrite the Dirichlet step (3.35) using (2.4) and (2.5), and find

$$\begin{cases} \ddot{z}_{1,i}^k - \sigma_i^2 z_{1,i}^k = 0 \text{ in } \Omega_1, \\ z_{1,i}^k(0) = 0, \\ \dot{z}_{1,i}^k(\alpha) + d_i z_{1,i}^k(\alpha) = f_{\alpha,i}^{k-1}, \end{cases} \qquad \begin{cases} \ddot{z}_{2,i}^k - \sigma_i^2 z_{2,i}^k = 0 \text{ in } \Omega_2, \\ \dot{z}_{2,i}^k(\alpha) + d_i z_{2,i}^k(\alpha) = f_{\alpha,i}^{k-1}, \\ \dot{z}_{2,i}^k(T) + \omega_i z_{2,i}^k(T) = 0. \end{cases} \tag{3.36}$$

We then correct the above RR step by a RR correction (3.19), which is also the equivalent of the Neumann step (3.18). And the update step (3.22) becomes

$$f_{\alpha,i}^k = f_{\alpha,i}^{k-1} - \theta\big(\dot{\psi}_{1,i}^k(\alpha) + d_i\psi_{1,i}^k(\alpha) + \dot{\psi}_{2,i}^k(\alpha) + d_i\psi_{2,i}^k(\alpha)\big). \tag{3.37}$$

Using (3.1), we can solve explicitly (3.36) and determine the coefficients

$$A_i^k = \frac{f_{\alpha,i}^{k-1}}{\sigma_i\cosh(a_i) + d_i\sinh(a_i)}, \quad B_i^k = -\nu\frac{f_{\alpha,i}^{k-1}}{\sigma_i\gamma\cosh(b_i) + \beta_i\sinh(b_i)}. \tag{3.38}$$

Combining with (3.20), we update the transmission condition (3.37) and obtain $f_{\alpha,i}^k = f_{\alpha,i}^{k-1} - \theta f_{\alpha,i}^{k-1}(E_i + F_i)$ with $E_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{\sigma_i\gamma\sinh(b_i)+\beta_i\cosh(b_i)}\frac{1}{\sigma_i\cosh(a_i)+d_i\sinh(a_i)}$, $F_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{\sigma_i\gamma\cosh(b_i)+\beta_i\sinh(b_i)}\frac{1}{\sigma_i\sinh(a_i)+d_i\cosh(a_i)}$. Thus, we have the following result.

THEOREM 3.14. *Algorithm NN_{3a} (3.35), (3.18), (3.22) converges if and only if*

$$\rho_{NN_{3a}} := \max_{d_i\in\lambda(A)} |1 - \theta(E_i + F_i)| < 1. \tag{3.39}$$

We consider some special cases to get more insight in the convergence factor (3.39). Assuming no final target (i.e., $\gamma = 0$) and a symmetric decomposition $\alpha = \frac{T}{2}$ (i.e., $a_i = b_i$), we find that E_i and F_i are actually the same as for algorithm $\mathrm{NN}_{2\mathrm{a}}$ in Section 3.2.1. Hence, the convergence factor (3.39) is as (3.28) under this assumption, and NN_{2a} and NN_{3a} are actually the same algorithm. Moreover, for a zero eigenvalue, substituting (3.6) into (3.39), we find exactly the same formula as (3.29). Thus, the two algorithms NN_{2a} and NN_{3a} share the same behavior for small eigenvalues. On the other hand, using (3.7) for large eigenvalues d_i, we find $E_i \sim_\infty 2$ and $F_i \sim_\infty 2$. This implies that $\lim_{d_i\to\infty} \rho_{NN_{3a}} = |1-4\theta|$, which is the same as for algorithm NN_{2a}. Once again, the two algorithms NN_{2a} and NN_{3a} share the same behavior for large eigenvalues. Hence, we obtain the same relaxation parameter $\theta^*_{NN_{3a}} = \theta^*_{NN_{2a}}$ as defined in (3.30). In general, algorithm NN_{3a} seems to be very similar to NN_{2a}, and we could also expect it to be a good smoother and solver.

3.3.2. Algorithm NN_{3b}. The second variant in Category III consists in applying the Neumann step to the primal correction state ψ_i. In this way, we consider the algorithm that first solves the Dirichlet step (3.35), followed by the Neumann step (3.8), and updates the transmission condition by (3.22).

For the convergence analysis, we solve a RR step (3.36) and correct by a NN step (3.9). Using (3.38) and (3.10), we can update the transmission condition (3.37) and find $f^k_{\alpha,i} = f^{k-1}_{\alpha,i} - f^{k-1}_{\alpha,i}\theta d_i(E_i - \nu F_i)$ with $F_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{(\sigma_i\gamma\cosh(b_i)+\beta_i\sinh(b_i))\cosh(a_i)}$ and $E_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{(\sigma_i\sinh(b_i)+\omega_i\cosh(b_i))(\sigma_i\cosh(a_i)+d_i\sinh(a_i))}$. This leads to the convergence factor

$$\rho_{NN_{3b}} := \max_{d_i\in\lambda(A)} |1-\theta d_i(E_i-\nu F_i)| < 1. \tag{3.40}$$

We first study the extreme cases. For a zero eigenvalue, substituting the identities (3.6) into (3.40), we find $(E_i - \nu F_i)|_{d_i=0} = 0$, and hence $\rho_{NN_{3b}}|_{d_i=0} = 1$. This is once again independent of the relaxation parameter. In other words, the convergence of this algorithm is not good for small eigenvalues, and the relaxation cannot fix this problem. For large eigenvalues d_i, using (3.7), we find $E_i \sim_\infty \frac{1}{d_i}$ and $F_i \sim_\infty 4d_i$. Thus, we obtain $\rho_{NN_{3b}} \sim_\infty 4\nu\theta d_i^2$ and $\lim_{d_i\to\infty}\rho_{NN_{3b}} = \infty$, which is divergent and cannot be fixed with relaxation. In general, we have the following result.

THEOREM 3.15. *Algorithm NN_{3b} (3.35), (3.8), (3.22) always diverges.*

Proof. Following the same idea as in the proof of Theorem 3.10, we can show that $E_i - \nu F_i$ is always negative or zero, and this concludes the proof. ∎

Remark 3.16. One could have also applied a similar strategy as in Remark 3.11, that is, considering the update step (3.15) instead of (3.22). The convergence factor (3.40) then becomes $\max_{d_i\in\lambda(A)} |1-\theta(E_i-\nu F_i)|$. Once again, this does not change the poor convergence behavior for both small and large eigenvalues.

Similar to algorithm NN_{2b}, algorithm NN_{3b} is neither a good smoother nor a good solver, and other choices of the update step will not change this. Together with Section 3.2.2, we observe that, applying the Dirichlet step to the primal state z_i (resp. dual state μ_i) and correcting the result by a Neumann step to the dual correction state ϕ_i (resp. primal correction state ψ_i), will lead to divergent algorithms, and cannot be fixed even by adapting the update step.

3.3.3. Algorithm NN_{3c}. The last variant consists in applying the Neumann step to the pair (ψ_i, ϕ_i). In this way, the NN_{3b} algorithm solves first the Dirichlet step (3.35), next the Neumann step (2.2) which also has the forward-backward structure. Then it updates the transmission condition by (3.22).

For the convergence analysis, we solve a RR step (3.36) followed by a NR correction (2.7). Using (3.38) and (3.3), we update the transmission condition (3.37) and find $f_{\alpha,i}^k = f_{\alpha,i}^{k-1}(1-\theta(d_iE_i+F_i))$ with $E_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{(\sigma_i\sinh(b_i)+\omega_i\cosh(b_i))(\sigma_i\cosh(a_i)+d_i\sinh(a_i))}$ and $F_i = \frac{\sigma_i\cosh(\sigma_i T)+\omega_i\sinh(\sigma_i T)}{(\sigma_i\gamma\cosh(b_i)+\beta_i\sinh(b_i))(\sigma_i\sinh(a_i)+d_i\cosh(a_i))}$. We thus find the following result.

THEOREM 3.17. *Algorithm NN_{3c} (3.35), (2.2), (3.22) converges if and only if*

$$\rho_{NN_{3c}} := \max_{d_i\in\lambda(A)} |1-\theta(d_iE_i+F_i)| < 1. \quad (3.41)$$

We consider some special cases to get more insight. Assuming no final target (i.e., $\gamma = 0$) and a symmetric decomposition $\alpha = \frac{T}{2}$ (i.e., $a_i = b_i$), we find that E_i is actually the same as the F_i for algorithm $\mathrm{NN_{2c}}$, and F_i is the same as the E_i for algorithm $\mathrm{NN_{2c}}$ in Section 3.2.3. Hence, $\mathrm{NN_{2c}}$ and $\mathrm{NN_{3c}}$ are the same algorithm under this assumption. For a zero eigenvalue, $d_i = 0$, substituting the identities (3.6) into (3.41), we find $\rho_{\mathrm{NN_{3c}}}|_{d_i=0} = \rho_{\mathrm{NN_{2c}}}|_{d_i=0}$ as in (3.32). In other words, algorithms $\mathrm{NN_{2c}}$ and $\mathrm{NN_{3c}}$ have a similar behavior for small eigenvalues. For large eigenvalues d_i, using (3.7), we find $E_i \sim_\infty \frac{1}{d_i}$ and $F_i \sim_\infty 2$. Thus, we obtain $\lim_{d_i\to\infty}\rho_{\mathrm{NN_{3c}}} = |1-3\theta|$, which is independent of the interface α. So the convergence for large eigenvalues is robust with relaxation, and one can get a good smoother using $\theta = 1/3$. Furthermore, we find again similar behavior between algorithms $\mathrm{NN_{2c}}$ and $\mathrm{NN_{3c}}$ for large eigenvalues. Using hence equioscillation, we obtain $\theta^*_{\mathrm{NN_{3c}}} = \theta^*_{\mathrm{NN_{2c}}}$ as defined in (3.34). Based on all these similarities with algorithm $\mathrm{NN_{2c}}$, algorithm $\mathrm{NN_{3c}}$ is also a good smoother and solver. Also for a similar reason as explained in Remark 3.13, other choices of the update step will lead to divergent behavior.

4. Numerical results.

We illustrate now our nine new time domain decomposition algorithms with numerical experiments. As mentioned in the convergence analysis, some algorithms are much more sensitive to the chosen parameters than others. To well illustrate and compare these algorithms, we consider two different test cases,

case A: The time interval $\Omega = (0,1)$ is subdivided into $\Omega_1 = (0,0.5)$, $\Omega_2 = (0.5,1)$ (i.e., symmetric), and the objective function has no explicit final target term ($\gamma = 0$). The regularization parameter is $\nu = 0.1$.

case B: The time interval $\Omega = (0,5)$ is subdivided into $\Omega_1 = (0,1)$, $\Omega_2 = (1,5)$ (i.e., asymmetric), and the objective function has a final target term with $\gamma = 10$. The regularization parameter is $\nu = 10$.

For each test, we will investigate the performance by plotting the convergence factor as a function of the eigenvalues $d_i \in [10^{-2}, 10^2]$.

4.1. Convergence factor of $\mathrm{NN_{2b}}$ and $\mathrm{NN_{3b}}$.

We first illustrate the behavior of $\mathrm{NN_{2b}}$ and $\mathrm{NN_{3b}}$ separately, since their convergence analyses are very similar, and both algorithms are divergent. Figure 2 shows the behavior of the convergence factor as a function of the eigenvalues for these two algorithms. More precisely, both algorithms diverge in the case $\theta = 0.25$. And for both test cases A and B, the two algorithms diverge violently for large eigenvalues with the scale of 10^3 for $\mathrm{NN_{2b}}$ and 10^5 for $\mathrm{NN_{3b}}$. This corresponds to our estimate $4\nu\theta d_i^2$. By applying optimization[1], we find the optimal relaxation parameter is approximately zero for both algorithms in the test cases. As shown in our analysis, the best one can do is to choose $\theta = 0$ to compensate the bad large eigenvalue behavior, yet the algorithms are still divergent. Note that $\mathrm{NN_{2b}}$ and $\mathrm{NN_{3b}}$ in the case $\theta = 0$ are actually a classical Schwarz type algorithm, which does not converge without overlap. Therefore, $\mathrm{NN_{2b}}$ and $\mathrm{NN_{3b}}$ are not good algorithms and cannot be improved with relaxation.

[1] We use in this paper the optimization toolbox *scipy.optimize.fmin* in python.

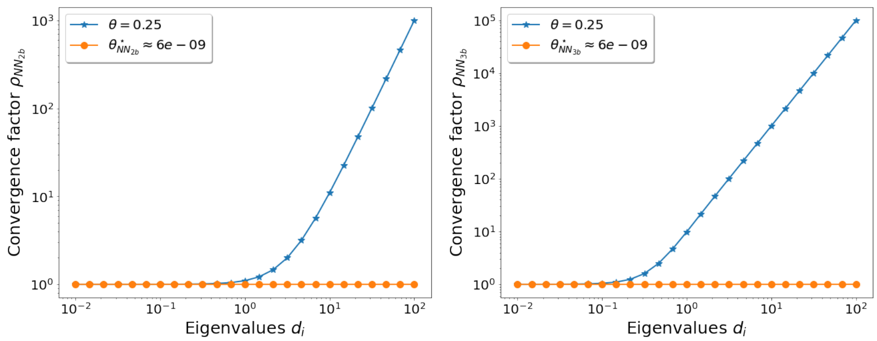

FIG. 2. *Convergence factor with $\theta = 0.25$ of NN_{2b} and NN_{3b} as a function of the eigenvalues $d_i \in [10^{-2}, 10^2]$. Left: case A for NN_{2b}. Right: case B for NN_{3b}.*

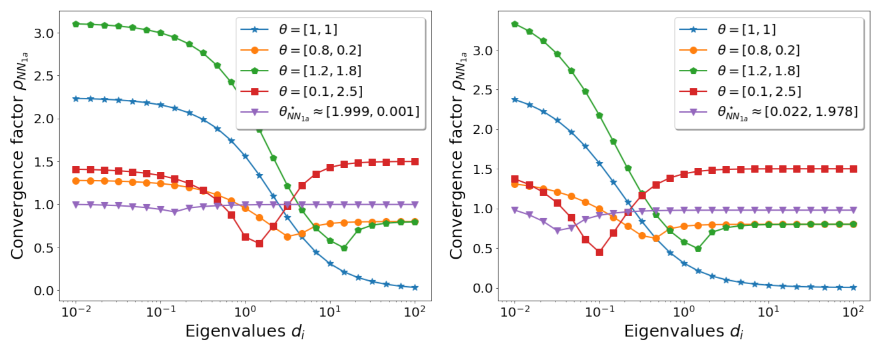

FIG. 3. *Convergence factor with different relaxation parameters θ of NN_{1a} as a function of the eigenvalues $d_i \in [10^{-2}, 10^2]$. Left: case A. Right: case B.*

4.2. Convergence factor of NN_{1a} with different θ. The second test is dedicated to the most natural Neumann–Neumann algorithm NN_{1a}. Based on our analysis, NN_{1a} is only a good smoother but not a good solver. Therefore, we choose some different relaxation parameters θ and show the behavior of the convergence factor as a function of the eigenvalues in Figure 3. For both test cases A and B, NN_{1a} has similar behavior for the tested parameters θ. In the case $\theta = [0.8, 0.2]$ and $\theta = [1.2, 1.8]$, the convergence behavior is the same for large eigenvalues. Indeed, our analysis shows that $\lim_{d_i \to \infty} \rho_{\mathrm{NN}_{1a}} = \{|1 - \theta_1|, |1 - \theta_2|\}$, and in this case equals to 0.8 for both θ. Furthermore, we observe that NN_{1a} is a good smoother with the choice $\theta = [1, 1]$. By using optimization, we find that the optimal relaxation parameter has the form that one goes to zero and the other one goes to two, yet with a poor convergence. Therefore, NN_{1a} can be a good smoother but not a good solver.

4.3. Convergence factor with $\theta = 1/2$. We now focus on the remaining six algorithms NN_{1b}, NN_{1c}, NN_{2a}, NN_{2c}, NN_{3a} and NN_{3c}. Based on our analysis, all six algorithms have shown the potentiel of being a good solver, we thus compare them with a given relaxation parameter $\theta = 1/2$ in two test cases. Figure 4 shows the behavior of the convergence factor as a function of the eigenvalues for the six algorithms. In case A, we observe that NN_{2a} and NN_{3a} have identical behavior, and similar for NN_{2c} and NN_{3c}. Indeed, as explained in our analysis, the convergence factors are the same in case A for NN_{2a} and NN_{3a}, and also for NN_{2c} and NN_{3c}.

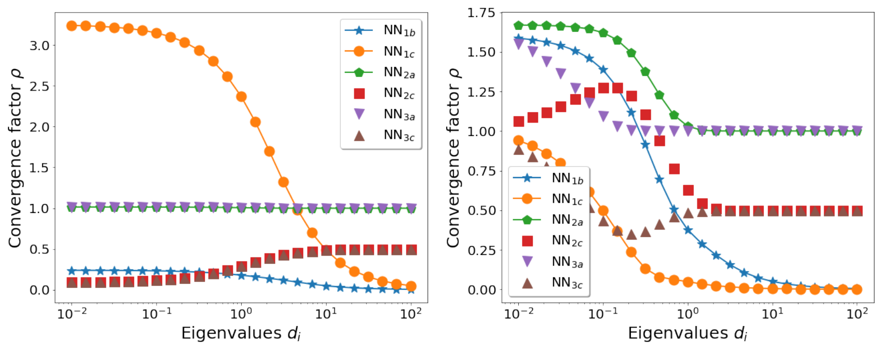

FIG. 4. *Convergence factor with $\theta = 1/2$ of the six algorithms as a function of the eigenvalues $d_i \in [10^{-2}, 10^2]$. Left: case A. Right: case B.*

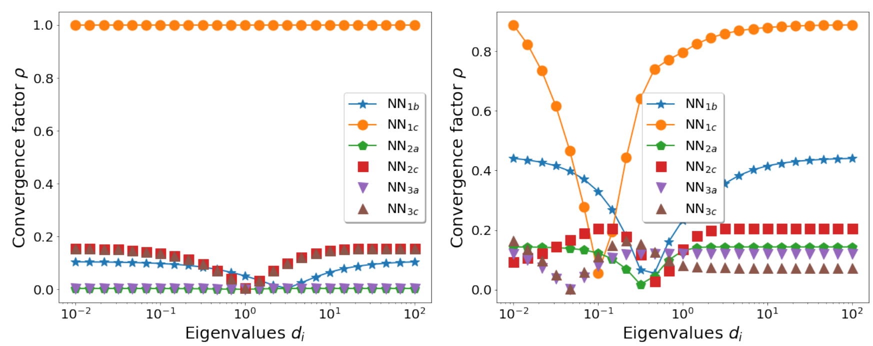

FIG. 5. *Convergence factor with optimal relaxation parameter $\theta^\star$ of the six algorithms as a function of the eigenvalues $d_i \in [10^{-2}, 10^2]$. Left: case A. Right: case B.*

Furthermore, $\mathrm{NN_{1b}}$ and $\mathrm{NN_{1c}}$ have similar behavior for large eigenvalues, which has also been pointed out in our analysis. And as expected, these two algorithms are good smoothers with $\theta = 1/2$. In particular, $\mathrm{NN_{1b}}$ outperforms the other five algorithms in case A, that is both a good smoother and solver. However, this changes in case B. More precisely, $\mathrm{NN_{2a}}$ and $\mathrm{NN_{3a}}$ have rather a symmetric behavior, as well as $\mathrm{NN_{2c}}$ and $\mathrm{NN_{3c}}$. And as shown in our analysis, both $\mathrm{NN_{2a}}$ and $\mathrm{NN_{3a}}$ have the same behavior for large eigenvalues, and also $\mathrm{NN_{2c}}$ and $\mathrm{NN_{3c}}$. Moreover, $\mathrm{NN_{1b}}$ and $\mathrm{NN_{1c}}$ are both good smoothers, and $\mathrm{NN_{1c}}$ has a better performance than $\mathrm{NN_{1b}}$ this time.

4.4. Convergence factor with optimal θ. We then show the convergence behavior of each algorithm using their optimal relaxation parameter $\theta^\star$ determined by optimization. Figure 5 shows the behavior of the convergence factor as a function of the eigenvalues for the six algorithms. In case A, $\mathrm{NN_{2a}}$ and $\mathrm{NN_{3a}}$ have once again identical behavior. Indeed, their convergence factors are the same in case A, and both $\mathrm{NN_{2a}}$ and $\mathrm{NN_{3a}}$ have the same optimal relaxation parameter $\theta^\star_{\mathrm{NN_{2a}}} = \theta^\star_{\mathrm{NN_{3a}}}$, which corresponds to the theoretical value $\theta^*_{\mathrm{NN_{2a}}} \approx 0.249$ as determined by (3.30). For the same reason, we observe the same behavior for $\mathrm{NN_{2c}}$ and $\mathrm{NN_{3c}}$, where the optimal relaxation parameter $\theta^\star_{\mathrm{NN_{2c}}} = \theta^\star_{\mathrm{NN_{3c}}} = \theta^*_{\mathrm{NN_{2c}}} \approx 0.385$ as determined by (3.34). As for $\mathrm{NN_{1b}}$, we find that the optimal relaxation parameter $\theta^\star_{\mathrm{NN_{1b}}} = \theta^*_{\mathrm{NN_{1b}}} \approx 0.446$ as determined by (3.17). However, the optimal relaxation parameter for $\mathrm{NN_{1c}}$ is $\theta^\star_{\mathrm{NN_{1c}}} \approx 0$, which cannot be determined by (3.24). As explained in our analysis,

Table 2
Convergence factor with numerical optimal relaxation parameter.

		NN_{1b}	NN_{1c}	NN_{2a}	NN_{2c}	NN_{3a}	NN_{3c}
case A	ρ	0.104	1.000	**0.004**	0.156	**0.004**	0.156
	$\theta^\star$	0.446	10^{-15}	0.249	0.385	0.249	0.385
case B	ρ	0.440	0.888	**0.143**	0.205	**0.121**	0.165
	$\theta^\star$	0.278	0.944	0.214	0.265	0.220	0.307

the term $E_i - \nu^{-1}F_i$ in (3.23) is negative in case A, thus the best option is to choose $\theta = 0$ which becomes then a Schwarz type algorithm without overlap. In general, all algorithms except NN_{1c} have very good performance in case A, and both NN_{2a} and NN_{3a} outperform the others with a convergence factor around 10^{-3}. Once again, the behavior of the six algorithms becomes much different in case B. While NN_{1c} diverges in case A, it converges in the test case B with the optimal relaxation parameter $\theta^\star_{NN_{1c}} = \theta^*_{NN_{1c}} \approx 0.944$ as determined by (3.24). NN_{1b} rather keeps a similar performance with the optimal relaxation parameter $\theta^\star_{NN_{1b}} = \theta^*_{NN_{1b}} \approx 0.278$ as determined by (3.17). NN_{2a} also has the same optimal relaxation parameter $\theta^\star_{NN_{2a}} = \theta^*_{NN_{2a}} \approx 0.214$ as determined by (3.30), which is slightly different from $\theta^\star_{NN_{3a}} \approx 0.220$ for NN_{3a}. However, for NN_{2c} and NN_{3c}, the optimal relaxation parameter of $\theta^\star_{NN_{2c}} \approx 0.265$ is rather different from $\theta^\star_{NN_{3c}} \approx 0.307$, and both are different from the value determined by (3.34) using equioscillation $\theta^*_{NN_{2c}} \approx 0.285$. Indeed, NN_{2c} rather equioscillates the convergence value between large eigenvalues with some eigenvalues in the interval $[0.1, 1]$, whereas NN_{3c} equioscillates the convergence factor value between small eigenvalues with some eigenvalues in the interval $[0.1, 1]$. In general, all six algorithms converge in case B, NN_{2a} and NN_{3a} still outperform the others with NN_{3a} slightly better than NN_{2a}. We summarize all these results in Table 2.

4.5. Numerical performance of NN_{2a}. Based on our theoretical analysis of the convergence factors, we expect excellent convergence behavior for the algorithm NN_{2a} also in a numerical setting. To illustrate its performance, we now numerically solve the forward-backward problem (1.2)-(1.3) using the algorithm NN_{2a}. We consider the target state $\hat{y}(x,t) = \sin(\pi x)(2t^2 + t)$, the initial condition $y_0(x) = 0$. The problem is discretized using a second order finite-difference scheme with $J_x = J_t = 128$ and $h_t = h_x = \frac{1}{J_x+1}$. Moreover, we choose the relaxation parameter to be $\theta = 0.25$, which is both the theoretical and numerical optimal relaxation parameter in the test case A with a symmetric decomposition. We also keep the same numerical settings as in the test case A and B, except for the subdivision of the time domain. To compare the numerical performance for several subdomains, we equally divide the time domain into N_{sub} subdomains. Figure 6 shows the numerical error decay of NN_{2a} with respect to the iteration number for different values of N_{sub}. We observe that the numerical error decays very fast with 2 subdomains. However, when we increase the number of subdomain N_{sub}, the convergence efficiency decreases for the time domain $(0, 1)$ as the length of each subdomain becomes smaller. Conversely, we still maintain good convergence behavior for the time domain $(0, 5)$ when increasing N_{sub}. Further investigation into how subdomain length affects the results and the potential need of a coarse space is beyond the scope of our present study and will be detailed elsewhere.

5. Conclusion. We introduced and investigated nine new time domain decomposition methods based on Neumann–Neumann algorithms for parabolic optimal control problems. Our analysis indicates that the Neumann correction step and the

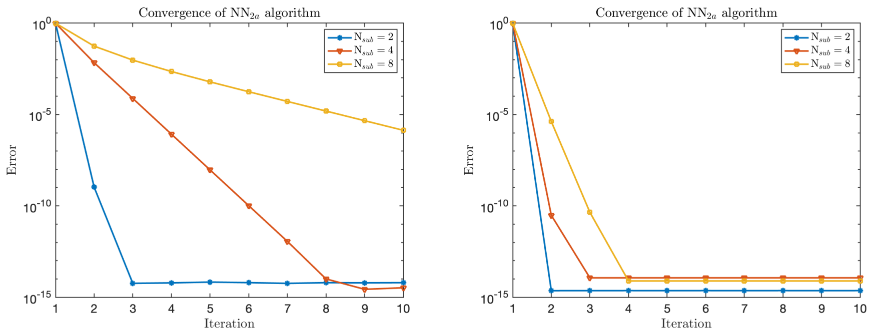

FIG. 6. *Numerical decay of the error of NN_{2a} with relaxation parameter $\theta = 0.25$ and $N_{sub} = 2, 4, 8$ respectively. Left: case A. Right: case B.*

update step must be carefully aligned with the Dirichlet step to prevent potential divergence. Moreover, while it might seem natural at first to maintain the forward-backward structure within the time subdomains, alternative choices exist that result in faster algorithms. These alternatives can still be seen with forward-backward structure through change of variables. Additionally, we discovered several intriguing connections between these algorithms. For instance, algorithms in Categories II and III have rather similar convergence behavior. In terms of the performance, algorithms NN_{2b} and NN_{3b} perform poorly, whereas the most natural algorithm NN_{1a} serves as a good smoother. Algorithms NN_{2a} and NN_{3a}, with optimized relaxation parameter, are much faster than the other algorithms and can be considered as highly efficient solvers. Our theoretical analysis was restricted to the two subdomain case, however our algorithms can all be extended to handle many subdomains as illustrated in our last numerical experiment. A natural extension of this work would involve a detailed investigation of the numerical performance of each algorithm and for many subdomains. Additionally, it would also be interesting to compare these algorithms with other non-overlapping domain decomposition methods.

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), pp. 27–38.

[2] A. Alla and S. Volkwein, *Asymptotic stability of POD based model predictive control for a semilinear parabolic PDE*, Advances in Computational Mathematics, 41 (2015), pp. 1073–1102.

[3] P. E. Bjørstad and O. B. Widlund, *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*, SIAM Journal on Numerical Analysis, 23 (1986), pp. 1097–1120.

[4] A. Borzì and V. Schulz, *Computational Optimization of Systems Governed by Partial Differential Equations*, Society for Industrial and Applied Mathematics, Philadelphia, 2011.

[5] A. Bünger, S. Dolgov, and M. Stoll, *A low-rank tensor method for PDE-constrained optimization with isogeometric analysis*, SIAM Journal on Scientific Computing, 42 (2020), pp. A140–A161.

[6] M. Emmett and M. Minion, *Toward an efficient parallel in time method for partial differential equations*, Communications in Applied Mathematics and Computational Science, 7 (2012), pp. 105 – 132.

[7] R. D. Falgout, S. Friedhoff, T. V. Kolev, S. P. MacLachlan, and J. B. Schroder, *Parallel time integration with multigrid*, SIAM Journal on Scientific Computing, 36 (2014), pp. C635–C661.

[8] L. Fang, S. Vandewalle, and J. Meyers, *A parallel-in-time multiple shooting algorithm for*

large-scale PDE-constrained optimal control problems, Journal of Computational Physics, 452 (2022), p. 110926.

[9] C. Farhat and M. Chandesris, *Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid–structure applications*, International Journal for Numerical Methods in Engineering, 58 (2003), pp. 1397–1434.

[10] C. Farhat and F.-X. Roux, *A method of finite element tearing and interconnecting and its parallel solution algorithm*, International Journal for Numerical Methods in Engineering, 32 (1991), pp. 1205–1227.

[11] M. J. Gander, *50 years of time parallel time integration*, in Multiple Shooting and Time Domain Decomposition Methods, T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, eds., Springer, Heidelberg, 2015, pp. 69–114.

[12] 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, T. Dickopf, M. J. Gander, L. Halpern, R. Krause, and L. F. Pavarino, eds., Cham, 2016, Springer International Publishing, pp. 207–216.

[13] M. J. Gander, F. Kwok, and J. Salomon, *PARAOPT: A parareal algorithm for optimality systems*, SIAM Journal on Scientific Computing, 42 (2020), pp. A2773–A2802.

[14] M. J. Gander and L.-D. Lu, *New time domain decomposition methods for parabolic optimal control problems I: Dirichlet-Neumann and Neumann-Dirichlet algorithms.* To appear in SIAM Journal on Numerical Analysis, 2024.

[15] S. Götschel and M. L. Minion, *An efficient parallel-in-time method for optimization with parabolic PDEs*, SIAM Journal on Scientific Computing, 41 (2019), pp. C603–C626.

[16] M. D. Gunzburger and A. Kunoth, *Space-time adaptive wavelet methods for optimal control problems constrained by parabolic evolution equations*, SIAM Journal on Control and Optimization, 49 (2011), pp. 1150–1170.

[17] W. Hackbusch, *Numerical solution of linear and nonlinear parabolic control problems*, in Optimization and Optimal Control, A. Auslender, W. Oettli, and J. Stoer, eds., Springer Berlin Heidelberg, 1981, pp. 179–185.

[18] L. Halpern and J. Szeftel, *Optimized and quasi-optimal Schwarz waveform relaxation for the one-dimensional Schrödinger equation*, Mathematical Models and Methods in Applied Sciences, 20 (2010), pp. 2167–2199.

[19] M. Heinkenschloss, *A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems*, Journal of Computational and Applied Mathematics, 173 (2005), pp. 169–198.

[20] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, Springer Dordrecht, 2009.

[21] L. Iapichino, S. Trenz, and S. Volkwein, *Reduced-order multiobjective optimal control of semilinear parabolic problems*, in Numerical Mathematics and Advanced Applications ENUMATH 2015, B. Karasözen, M. Manguoğlu, M. Tezer-Sezgin, S. Göktepe, and Ö. Uğur, eds., Cham, 2016, Springer International Publishing, pp. 389–397.

[22] E. Kammann, F. Tröltzsch, and S. Volkwein, *A posteriori error estimation for semilinear parabolic optimal control problems with application to model reduction by POD*, ESAIM: M2AN, 47 (2013), pp. 555–581.

[23] M. Kollmann, M. Kolmbauer, U. Langer, M. Wolfmayr, and W. Zulehner, *A robust finite element solver for a multiharmonic parabolic optimal control problem*, Computers & Mathematics with Applications, 65 (2013), pp. 469–486.

[24] K. Kunisch, S. Volkwein, and L. Xie, *HJB-POD-based feedback design for the optimal control of evolution problems*, SIAM Journal on Applied Dynamical Systems, 3 (2004), pp. 701–722.

[25] F. Kwok, *On the time-domain decomposition of parabolic optimal control problems*, in Domain Decomposition Methods in Science and Engineering XXIII, C.-O. Lee, X.-C. Cai, D. E. Keyes, H. H. Kim, A. Klawonn, E.-J. Park, and O. B. Widlund, eds., Cham, 2017, Springer International Publishing, pp. 55–67.

[26] E. Lelarasmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelli, *The waveform relaxation method for time-domain analysis of large scale integrated circuits*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1 (1982), pp. 131–145.

[27] B. Li, J. Liu, and M. Xiao, *A new multigrid method for unconstrained parabolic optimal control problems*, Journal of Computational and Applied Mathematics, 326 (2017), pp. 358–373.

[28] J.-L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*, 170, Springer-Verlag Berlin Heidelberg, 1 ed., 1971.

[29] J.-L. Lions, Y. Maday, and G. Turinici, *A parareal in time procedure for the control of Partial Differential Equations*, Comptes Rendus Mathematique, 335 (2002), pp. 387–392.

[30] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, vol. 112, Graduate Studies in Mathematics, 2010.

[31] S. R. Ulrich Langer and M. Wolfmayr, *Functional a posteriori error estimates for time-periodic parabolic optimal control problems*, Numerical Functional Analysis and Optimization, 37 (2016), pp. 1267–1294.

Appendix A. Pair transmission conditions.

Let us consider a modified algorithm NN_{2a}, that is, we first solve the Dirichlet step

$$\begin{cases} \begin{pmatrix} \dot{z}_{1,i}^k \\ \dot{\mu}_{1,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{1,i}^k \\ \mu_{1,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_1, \\ z_{1,i}^k(0) = 0, \\ z_{1,i}^k(\alpha) = f_{\alpha,i}^{k-1}, \end{cases}$$

$$\begin{cases} \begin{pmatrix} \dot{z}_{2,i}^k \\ \dot{\mu}_{2,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} z_{2,i}^k \\ \mu_{2,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_2, \\ z_{2,i}^k(\alpha) = g_{\alpha,i}^{k-1}, \\ \mu_{2,i}^k(T) + \gamma z_{2,i}^k(T) = 0, \end{cases}$$

and then correct the result by the Neumann step

$$\begin{cases} \begin{pmatrix} \dot{\psi}_{1,i}^k \\ \dot{\phi}_{1,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} \psi_{1,i}^k \\ \phi_{1,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_1, \\ \psi_{1,i}^k(0) = 0, \\ \dot{\psi}_{1,i}^k(\alpha) = \dot{z}_{1,i}^k(\alpha) - \dot{z}_{2,i}^k(\alpha), \end{cases}$$

$$\begin{cases} \begin{pmatrix} \dot{\psi}_{2,i}^k \\ \dot{\phi}_{2,i}^k \end{pmatrix} + \begin{pmatrix} d_i & -\nu^{-1} \\ -1 & -d_i \end{pmatrix} \begin{pmatrix} \psi_{2,i}^k \\ \phi_{2,i}^k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ in } \Omega_2, \\ \dot{\psi}_{2,i}^k(\alpha) = \dot{z}_{2,i}^k(\alpha) - \dot{z}_{1,i}^k(\alpha), \\ \phi_{2,i}^k(T) + \gamma \psi_{2,i}^k(T) = 0. \end{cases}$$

and update the transmission condition by

$$f_{\alpha,i}^k := f_{\alpha,i}^{k-1} - \theta_1\big(\psi_{1,i}^k(\alpha) + \psi_{2,i}^k(\alpha)\big), \quad g_{\alpha,i}^k := g_{\alpha,i}^{k-1} - \theta_2\big(\psi_{1,i}^k(\alpha) + \psi_{2,i}^k(\alpha)\big),$$

with $\theta_1, \theta_2 > 0$. Following the same analysis as in Section 3.2.1, we find,

$$\begin{pmatrix} f_{\alpha,i}^k \\ g_{\alpha,i}^k \end{pmatrix} = \begin{pmatrix} 1 - \theta_1 E_i & -\theta_1 F_i \\ -\theta_2 E_i & 1 - \theta_2 F_i \end{pmatrix} \begin{pmatrix} f_{\alpha,i}^{k-1} \\ g_{\alpha,i}^{k-1} \end{pmatrix}.$$

In particular, the eigenvalues of the iteration matrix are 1 and $1-(\theta_1 E_i+\theta_2 F_i)$. Thus, the modified algorithm NN_{2a} does not converge in this form. This divergence still stays even by considering the update step (2.3) for the pair transmission conditions. More generally, we have the same behavior for NN_{2b}, NN_{2c}, NN_{3a}, NN_{3b} and NN_{3c}, if we keep a pair of transmission conditions $(f_{\alpha,i}^k, g_{\alpha,i}^k)$.