

A MULTIRATE NEUMANN–NEUMANN WAVEFORM RELAXATION METHOD FOR HETEROGENEOUS COUPLED HEAT EQUATIONS*

AZAHAR MONGE[†] AND PHILIPP BIRKEN[†]

Abstract. An important challenge when coupling two different time dependent problems is to increase parallelization in time. We suggest a multirate Neumann–Neumann waveform relaxation algorithm to solve two heterogeneous coupled heat equations. In order to fix the mismatch produced by the multirate feature at the space-time interface a linear interpolation is constructed. The heat equations are discretized using a finite element method in space, whereas two alternative time integration methods are used: implicit Euler and SDIRK2. We perform a one-dimensional convergence analysis for the nonmultirate fully discretized heat equations using implicit Euler to find the optimal relaxation parameter in terms of the material coefficients, the step size, and the mesh resolution. This gives a very efficient method which needs only two iterations. Numerical results confirm the analysis and show that the one-dimensional nonmultirate optimal relaxation parameter is a very good estimator for the multirate one-dimensional case and even for multirate and nonmultirate two-dimensional examples using both implicit Euler and SDIRK2.

Key words. fluid-structure interaction, coupled problems, iterative solvers, multirate, domain decomposition, transmission problem

AMS subject classifications. 68Q25, 68R10, 68U05

DOI. 10.1137/18M1187878

1. Introduction. The main goal of this work is to describe a partitioned algorithm to solve two heterogeneous coupled heat equations allowing parallelization in time. In a partitioned approach different codes for the subproblems are reused and the coupling is done by a master program which calls interface functions of the segregated codes [6, 7]. These algorithms are currently an active research topic driven by certain multiphysics applications where multiple physical models or multiple simultaneous physical phenomena involve solving coupled systems of partial differential equations (PDEs). An example of this is fluid structure interaction (FSI) [34, 4]. Moreover, we want that the time parallelization performed at the subsolvers works for different time grids. This is handled through multirate methods which are a classical field of research (see [9]).

Our prime motivation here is thermal interaction between fluids and structures, also called conjugate heat transfer. There are two domains with jumps in the material coefficients across the connecting interface. Conjugate heat transfer plays an important role in many applications and its simulation has proved essential [1]. Examples for thermal fluid structure interaction are cooling of gas-turbine blades, thermal anti-icing systems of airplanes [5], supersonic reentry of vehicles from space [28, 17], gas quenching, which is an industrial heat treatment of metal workpieces [16, 32], or the cooling of rocket nozzles [20, 21].

*Received by the editors May 17, 2018; accepted for publication (in revised form) March 25, 2019; published electronically October 29, 2019.

<https://doi.org/10.1137/18M1187878>

Funding: This work was supported by the eSENCE project 3M: Multiphysics, multicore, multirate solution of coupled dynamic problems.

[†]Centre for Mathematical Sciences, Lund University, Sweden (azahar.monge@na.lu.se, <http://www.maths.lu.se/staff/azahar-monge/>, philipp.birken@na.lu.se).

The classical way of parallelizing the numerical solution of PDEs is to use domain decomposition (DD) methods. These split the computational domain into subdomains and coordinate the coupling between the subdomains in an iterative manner. For an introduction to DD methods and their basic convergence results see [31, 33]. The Dirichlet–Neumann iteration is a standard DD method to find solutions of the coupled problem. The PDEs are solved sequentially using a Dirichlet, respectively, Neumann, boundary with data given from the solution of the other problem. Previous numerical experiments [2] showed that this iteration is fast for thermal FSI, and a convergence analysis of two heterogeneous linear heat equations showed that the fast behavior was a consequence of the strong jumps in the material coefficients [29].

In spite of that, it has two main disadvantages. First, the subsolvers wait for each other, and therefore, they perform sequentially. Second, in the time dependent case the Dirichlet–Neumann iteration is used at each time step and consequently, both fields are solved with a common time resolution. Using instead a multirate scheme that allows for different time resolutions on each subdomain would be more efficient.

The aim of this work is to present a high order, parallel, multirate method for two heterogeneous coupled heat equations which could be applied to FSI problems. We use the Neumann–Neumann waveform relaxation (NNWR) method of Gander and others [22, 12], which is a waveform relaxation (WR) method based on the classical Neumann–Neumann iteration. Here, we describe a multirate version of it in detail. For time discretization we consider two alternatives, the implicit Euler method and a second order singly diagonally implicit Runge–Kutta (SDIRK2) method. WR methods were originally introduced by [23] for ordinary differential equation (ODE) systems and used for the first time to solve time dependent PDEs in [13, 14]. They allow the use of different spatial and time discretizations for each subdomain which is specially useful in problems with strong jumps in the material coefficients [11] or the coupling of different models for the subdomains [10]. A time adaptive partitioned approach for thermal FSI was presented in [3]. In [27], two new iterative partitioned coupling methods that allow for the simultaneous execution of flow and structure solvers were introduced. Furthermore, multirate approaches for the coupling of heterogeneous materials using the optimized Schwarz algorithm were presented in [15] for the advection-diffusion equation, in [24] for coupled ocean-atmospheric problems, and in [19] for fracture models in mixed formulation. However, parallelization in time for the coupling of heterogeneous materials using the NNWR method was not considered.

Our algorithm has to take care of two aspects. On one hand, an interpolation procedure needs to be chosen to communicate data between the subdomains through the space-time interface in the multirate case. We want that the interpolation preserves a second order numerical solution of the coupled problem when using SDIRK2. On the other hand, the choice of the relaxation parameter for the NNWR method is crucial because when choosing the relaxation parameter right, two iterations are sufficient for two equispaced subdomains. In [22], a one-dimensional analysis based on Laplace transform at the continuous level shows that $\Theta = 1/4$ is the optimal relaxation parameter for two homogeneous coupled heat equations on two identical subdomains.

In this paper, we perform a fully discrete analysis of the NNWR algorithm for two heterogeneous coupled one-dimensional heat equations to find the optimal relaxation parameter in terms of the material coefficients. More specifically, we derive the iteration matrix of the fully discrete NNWR algorithm with respect to the interface

unknowns. Then, we calculate the spectral radius of the iteration matrix through its eigendecomposition in order to estimate the optimal relaxation parameter Θ_{opt} which is dependent on the material coefficients, the time, and space resolutions. In the case of homogeneous materials, $\Theta_{opt} = 1/4$ recovering the result in [22]. Furthermore, the asymptotic optimal relaxation parameters when approaching the continuous case in either time or space are also determined.

In addition, we include numerical results where it is shown that the parallel, multirate method for two heterogeneous coupled heat equations introduced in this paper is extremely fast when choosing the right relaxation parameter. Moreover, we also show that the one-dimensional formula is a very good estimate for the multirate one-dimensional case and even for multirate and nonmultirate two-dimensional examples using both implicit Euler and SDIRK2. Finally, we also include a numerical comparison that shows that the NNWR method is a more efficient choice than the Dirichlet–Neumann waveform relaxation (DNWR) in the multirate case.

2. Model problem. The unsteady transmission problem reads as follows, where we consider a domain $\Omega \subset \mathbb{R}^d$ which is cut into two subdomains $\Omega = \Omega_1 \cup \Omega_2$ with transmission conditions at the interface $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$:

$$(2.1) \quad \begin{cases} \alpha_m \frac{\partial u_m(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_m \nabla u_m(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_m \subset \mathbb{R}^d, \quad m = 1, 2, \\ u_m(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_m \setminus \Gamma, \\ u_1(\mathbf{x}, t) = u_2(\mathbf{x}, t), & \mathbf{x} \in \Gamma, \\ \lambda_2 \frac{\partial u_2(\mathbf{x}, t)}{\partial \mathbf{n}_2} = -\lambda_1 \frac{\partial u_1(\mathbf{x}, t)}{\partial \mathbf{n}_1}, & \mathbf{x} \in \Gamma, \\ u_m(\mathbf{x}, 0) = u_m^0(\mathbf{x}), & \mathbf{x} \in \Omega_m, \end{cases}$$

where $t \in [T_0, T_f]$ and \mathbf{n}_m is the outward normal to Ω_m for $m = 1, 2$.

The constants λ_m describe the thermal conductivities of the materials on Ω_m . D_m represent the thermal diffusivities of the materials and they are defined by $D_m = \lambda_m / \alpha_m$ with $\alpha_m = \rho_m c_{p_m}$, where ρ_m represents the density and c_{p_m} the specific heat capacity. In this work we consider the case of constant material coefficients. For variable material coefficients $\alpha_m(\mathbf{x}, t)$ and $\lambda_m(\mathbf{x}, t)$, we expect from previous work on the Dirichlet–Neumann iteration [29] that also here, it is the jump directly at the interface that dictates the convergence behavior.

3. The Dirichlet–Neumann waveform relaxation algorithm. The DNWR method is a basic iterative substructuring method in domain decomposition. The PDEs are solved sequentially using Dirichlet, respectively, Neumann, boundary with data given from the solution of the other problem introduced in [25, 26].

It starts with an initial guess $g^0(\mathbf{x}, t)$ on the interface $\Gamma \times (T_0, T_f]$, and then performs a three-step iteration. At each iteration k , imposing continuity of the solution across the interface, one first finds the local solution $u_1^{k+1}(\mathbf{x}, t)$ in Ω_1 by solving the Dirichlet problem:

$$(3.1) \quad \begin{cases} \alpha_1 \frac{\partial u_1^{k+1}(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_1 \nabla u_1^{k+1}(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_1, \\ u_1^{k+1}(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_1 \setminus \Gamma, \\ u_1^{k+1}(\mathbf{x}, t) = g^k(\mathbf{x}, t), & \mathbf{x} \in \Gamma, \\ u_1^{k+1}(\mathbf{x}, 0) = u_1^0(\mathbf{x}), & \mathbf{x} \in \Omega_1. \end{cases}$$

Then, imposing continuity of the heat fluxes across the interface, one finds the local solution $u_2^{k+1}(\mathbf{x}, t)$ on Ω_2 by solving the Neumann problem:

$$(3.2) \quad \begin{cases} \alpha_2 \frac{\partial u_2^{k+1}(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_2 \nabla u_2^{k+1}(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_2, \\ u_2^{k+1}(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_2 \setminus \Gamma, \\ \lambda_2 \frac{\partial u_2^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_2} = -\lambda_1 \frac{\partial u_1^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_1}, & \mathbf{x} \in \Gamma, \\ u_2^{k+1}(\mathbf{x}, 0) = u_2^0(\mathbf{x}), & \mathbf{x} \in \Omega_2. \end{cases}$$

Finally, the interface values are updated with

$$(3.3) \quad g^{k+1}(\mathbf{x}, t) = \Theta u_2^{k+1}(\mathbf{x}, t) + (1 - \Theta)g^k(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma,$$

where $\Theta \in (0, 1]$ is the relaxation parameter. The optimal relaxation parameter for the DNWR algorithm has been proved to be $\Theta = 1/2$ in [12] for the choice $\lambda_1/\alpha_1 = \lambda_2/\alpha_2 = \text{constant}$.

4. The Neumann–Neumann waveform relaxation algorithm. We now describe the NNWR algorithm [22], [18, Chap. 2]. The main advantage of the NNWR method with respect to the DNWR method is that it allows for spatial parallelization even though the DNWR method can also be parallelized to some extent. For instance, in [30] the entire space-time domain is subdivided into smaller blocks. Then, rearranging the order of operations, different iterates can be simultaneously computed in a pipeline-parallel fashion.

The NNWR algorithm starts with an initial guess $g^0(\mathbf{x}, t)$ on the space-time interface $\Gamma \times (T_0, T_f]$, and then performs a three-step iteration. At each iteration k , one first solves two Dirichlet problems on Ω_1 and Ω_2 simultaneously, then two Neumann problems are solved simultaneously again on Ω_1 and Ω_2 , and finally, an update is performed to get a new guess $g^{k+1}(\mathbf{x}, t)$ on the interface $\Gamma \times (T_0, T_f]$.

More specifically, imposing continuity of the solution across the interface (i.e., given a common initial guess $g^0(\mathbf{x}, t)$ on $\Gamma \times (T_0, T_f]$), one can find the local solutions $u_m^{k+1}(\mathbf{x}, t)$ on Ω_m , $m = 1, 2$, through the following Dirichlet problems:

$$(4.1) \quad \begin{cases} \alpha_m \frac{\partial u_m^{k+1}(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_m \nabla u_m^{k+1}(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_m, \\ u_m^{k+1}(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_m \setminus \Gamma, \\ u_m^{k+1}(\mathbf{x}, t) = g^k(\mathbf{x}, t), & \mathbf{x} \in \Gamma, \\ u_m^{k+1}(\mathbf{x}, 0) = u_m^0(\mathbf{x}), & \mathbf{x} \in \Omega_m. \end{cases}$$

We now add into the framework the second coupling condition which is the continuity of the heat fluxes. To this end, one solves two simultaneous Neumann problems to get the correction functions $\psi_m^{k+1}(\mathbf{x}, t)$ on Ω_m , $m = 1, 2$, where the Neumann boundary condition at the interface $\Gamma \times (T_0, T_f]$ is prescribed by the continuity of the heat fluxes of the solutions $u_m^{k+1}(\mathbf{x}, t)$ given by the Dirichlet problems:

$$(4.2) \quad \begin{cases} \alpha_m \frac{\partial \psi_m^{k+1}(\mathbf{x}, t)}{\partial t} - \nabla \cdot (\lambda_m \nabla \psi_m^{k+1}(\mathbf{x}, t)) = 0, & \mathbf{x} \in \Omega_m, \\ \psi_m^{k+1}(\mathbf{x}, t) = 0, & \mathbf{x} \in \partial\Omega_m \setminus \Gamma, \\ \lambda_m \frac{\partial \psi_m^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_m} = \lambda_1 \frac{\partial u_1^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_1} + \lambda_2 \frac{\partial u_2^{k+1}(\mathbf{x}, t)}{\partial \mathbf{n}_2}, & \mathbf{x} \in \Gamma, \\ \psi_m^{k+1}(\mathbf{x}, 0) = 0, & \mathbf{x} \in \Omega_m. \end{cases}$$

Finally, the interface values are updated with

$$(4.3) \quad g^{k+1}(\mathbf{x}, t) = g^k(\mathbf{x}, t) - \Theta(\psi_1^{k+1}(\mathbf{x}, t) + \psi_2^{k+1}(\mathbf{x}, t)), \quad \mathbf{x} \in \Gamma,$$

where $\Theta \in (0, 1]$ is the relaxation parameter. Note that choosing an appropriate relaxation parameter is crucial to get a great rate of convergence of the NNWR algorithm [12]. If one uses the optimal relaxation parameter, two iterations are enough when the lengths of the subdomains are equal.

5. Semidiscrete method. We now describe a rather general space discretization of the problem (4.1)–(4.3). We assume that the meshes of Ω_1 and Ω_2 share the same nodes on Γ . Furthermore, we assume that there is a specific set of unknowns associated with the interface nodes. Otherwise, we allow at this point for arbitrary meshes on both sides.

Then, letting $\mathbf{u}_I^{(m)}, \psi_I^{(m)} : [T_0, T_f] \rightarrow \mathbb{R}^{R_m}$, where R_m is the number of grid points on Ω_m , $m = 1, 2$, and $\mathbf{u}_\Gamma, \psi_\Gamma^{(1)}, \psi_\Gamma^{(2)} : [T_0, T_f] \rightarrow \mathbb{R}^s$, where s is the number of common grid points at the interface Γ , we can write a general discretization of the first equation in (4.1) and (4.2), respectively, in a compact form as

$$(5.1) \quad \mathbf{M}_{II}^{(m)} \dot{\mathbf{u}}_I^{(m),k+1}(t) + \mathbf{A}_{II}^{(m)} \mathbf{u}_I^{(m),k+1}(t) = -\mathbf{M}_{I\Gamma}^{(m)} \dot{\mathbf{u}}_\Gamma^k(t) - \mathbf{A}_{I\Gamma}^{(m)} \mathbf{u}_\Gamma^k(t),$$

$$(5.2) \quad \mathbf{M}_{II}^{(m)} \dot{\psi}_I^{(m),k+1}(t) + \mathbf{M}_{I\Gamma}^{(m)} \dot{\psi}_\Gamma^{(m),k+1}(t) + \mathbf{A}_{II}^{(m)} \psi_I^{(m),k+1}(t) + \mathbf{A}_{I\Gamma}^{(m)} \psi_\Gamma^{(m),k+1}(t) = \mathbf{0},$$

where the initial conditions $\mathbf{u}_I^{(m)}(T_0), \psi_I^{(m)}(T_0) \in \mathbb{R}^{R_m}$ and $\mathbf{u}_\Gamma(T_0), \psi_\Gamma^{(m)}(T_0) \in \mathbb{R}^s$ for $m = 1, 2$ are known.

To close the system, we need an approximation of the normal derivatives on Γ . Those can be written as a linear functional using Green's formula [33, p. 3]. Thus, the equation

$$(5.3) \quad \begin{aligned} & \mathbf{M}_{\Gamma\Gamma}^{(m)} \dot{\psi}_\Gamma^{(m),k+1}(t) + \mathbf{M}_{\Gamma I}^{(m)} \dot{\psi}_I^{(m),k+1}(t) + \mathbf{A}_{\Gamma\Gamma}^{(m)} \psi_\Gamma^{(m),k+1}(t) + \mathbf{A}_{\Gamma I}^{(m)} \psi_I^{(m),k+1}(t) \\ &= \sum_{i=1}^2 \left(\mathbf{M}_{\Gamma\Gamma}^{(i)} \dot{\mathbf{u}}_\Gamma^k(t) + \mathbf{M}_{\Gamma I}^{(i)} \dot{\mathbf{u}}_I^{(i),k+1}(t) + \mathbf{A}_{\Gamma\Gamma}^{(i)} \mathbf{u}_\Gamma^k(t) + \mathbf{A}_{\Gamma I}^{(i)} \mathbf{u}_I^{(i),k+1}(t) \right), \quad m = 1, 2, \end{aligned}$$

is a discrete version of the third equation in (4.2) and completes the system (5.2). Check [29] for details.

We can now write a semidiscrete version of the NNWR algorithm using an ODE system. At each iteration k , one first solves the two Dirichlet problems in (5.1) obtaining $\mathbf{u}_I^{(m),k+1}(t)$ for $m = 1, 2$. Then, for the vector of unknowns $\psi_m^{k+1}(t) = (\psi_I^{(m),k+1}(t)^T \psi_\Gamma^{(m),k+1}(t)^T)^T$, one solves the following two Neumann problems in parallel that correspond to (5.2)–(5.3):

$$(5.4) \quad \mathbf{M}_m \dot{\psi}_m^{k+1}(t) + \mathbf{A}_m \psi_m^{k+1}(t) = \mathbf{b}^k, \quad m = 1, 2,$$

where

$$(5.5) \quad \mathbf{M}_m = \begin{pmatrix} \mathbf{M}_{II}^{(m)} & \mathbf{M}_{I\Gamma}^{(m)} \\ \mathbf{M}_{\Gamma I}^{(m)} & \mathbf{M}_{\Gamma\Gamma}^{(m)} \end{pmatrix}, \quad \mathbf{A}_m = \begin{pmatrix} \mathbf{A}_{II}^{(m)} & \mathbf{A}_{I\Gamma}^{(m)} \\ \mathbf{A}_{\Gamma I}^{(m)} & \mathbf{A}_{\Gamma\Gamma}^{(m)} \end{pmatrix}, \quad \mathbf{b}^k = \begin{pmatrix} \mathbf{0} \\ \mathbf{F}^k \end{pmatrix}$$

with

$$(5.6) \quad \mathbf{F}^k = \sum_{i=1}^2 \left(\mathbf{M}_{\Gamma\Gamma}^{(i)} \dot{\mathbf{u}}_\Gamma^k(t) + \mathbf{M}_{\Gamma I}^{(i)} \dot{\mathbf{u}}_I^{(i),k+1}(t) + \mathbf{A}_{\Gamma\Gamma}^{(i)} \mathbf{u}_\Gamma^k(t) + \mathbf{A}_{\Gamma I}^{(i)} \mathbf{u}_I^{(i),k+1}(t) \right).$$

Finally, the interface values are updated by

$$(5.7) \quad \mathbf{u}_\Gamma^{k+1}(t) = \mathbf{u}_\Gamma^k(t) - \Theta \left(\psi_\Gamma^{(1),k+1}(t) + \psi_\Gamma^{(2),k+1}(t) \right).$$

The iteration starts with some initial condition $\mathbf{u}_\Gamma^0(t)$ and a termination criterion must be chosen. One option would be $\|\mathbf{u}_\Gamma^{k+1}(t) - \mathbf{u}_\Gamma^k(t)\| \leq TOL$, where TOL is a user defined tolerance. However, this option is memory consuming because it saves the solutions for all $t \in [T_0, T_f]$. Moreover, an extra interpolation step is needed in the multirate case, i.e., when having two nonconforming time grids. As we expect the error to be largest at the end point T_f and because it simplifies the analysis to be presented for finding the optimal relaxation parameter, we propose the criterion $\|\mathbf{u}_\Gamma^{k+1}(T_f) - \mathbf{u}_\Gamma^k(T_f)\| \leq TOL$, where T_f is the synchronization endpoint of the macrostep.

6. Space-time interface interpolation. The NNWR algorithm for parabolic problems was first introduced in [22, 12]. They only briefly mention the possibility of using different step sizes on the two subdomains and provide one set of numerical results without going into details. In addition, their analysis does not include the coupling of two different materials. For those reasons, the goal of this paper is to introduce a parallel multirate method for the coupling of two heterogeneous heat equations and analyze its performance in the fully discrete case. This would be especially useful when coupling two different materials, where typically the field with higher heat conductivity needs a finer resolution than the other and therefore, efficiency will be gained by using a multirate method.

Both the Dirichlet and the Neumann problems (5.1) and (5.4) allow the use of independent time discretization on each of the subdomains. Therefore, in the case of mismatched time grids, there exists the need to define an interface interpolation.

Consider $\tau_1 = \{t_1, t_2, \dots, t_{N_1}\}$ and $\tau_2 = \{t_1, t_2, \dots, t_{N_2}\}$ to be two partitions of the time interval $[T_0, T_f]$. To exchange data at the space-time interface between the different time grids, we use linear interpolation. Given the local discrete solutions $F \in \mathbb{R}^{s \times N_1}$ and $G \in \mathbb{R}^{s \times N_2}$ at the space-time interface $\Gamma \times [T_0, T_f]$, with s being the number of grid points at Γ , we use the following procedure: For each $k = 1, 2, \dots, s$ and for each $t_i \in \tau_2$, find the subinterval in τ_1 such that $t_i \in [t_j, t_{j+1}]$. Linear interpolation through the points $(t_j, F(x_k, t_j))$ gives $G(x_k, t_j)$. We denote this by the interpolation function $G = I(\tau_2, \tau_1, F)$ and conversely, $F = I(\tau_1, \tau_2, G)$.

7. Time integration. In this section we present a time discretized version of the NNWR method presented in equations (5.1), (5.4), and (5.7). In order to get a multirate algorithm we use a certain time integration method with time step $\Delta t_1 := (T_f - T_0)/N_1$ on Ω_1 and with time step $\Delta t_2 := (T_f - T_0)/N_2$ on Ω_2 and the interpolation presented in the previous section will be used to transfer data from one domain to the other. We let $n_m := 1, 2, \dots, N_m$ be the time integration indices with respect to Ω_m and t_{n_m} defines any time point of the grid for $m = 1, 2$. We have chosen two alternative time integration schemes as a basis to construct the multirate algorithm: the implicit Euler method and an SDIRK2 method.

7.1. Implicit Euler. Applying the implicit Euler method with time step Δt_1 on Ω_1 and with time step Δt_2 on Ω_2 , we can write the systems (5.1), (5.4), and (5.7) in a fully discrete form.

The local approximations and the solutions at the space-time interface are given by the vectors $\mathbf{u}_\Gamma^{(m),k,n_m} \approx \mathbf{u}_\Gamma^{(m),k}(t_{n_m}) \in \mathbb{R}^{R_m}$ and $\mathbf{u}_\Gamma^{k,n_m} \approx \mathbf{u}_\Gamma^k(t_{n_m}) \in \mathbb{R}^s$, respectively. Remember that R_m is the number of spatial grid points on Ω_m and s is the num-

ber of spatial grid points at the interface Γ . Similarly, the corrections both in the subdomains and at the interface are given by the vectors $\psi_I^{(m),k,n_m} \approx \psi_I^{(m),k}(t_{n_m}) \in \mathbb{R}^{R_m}$ and $\psi_\Gamma^{(m),k,n_m} \approx \psi_\Gamma^{(m),k}(t_{n_m}) \in \mathbb{R}^s$, respectively.

At each iteration k , one first solves the two Dirichlet problems from (5.1) for $n_m = 1, 2, \dots, N_m$ with $\mathbf{u}_I^{(m),k+1,0} \approx \mathbf{u}_I^{(m)}(T_0)$, $m = 1, 2$, and $\mathbf{u}_\Gamma^{k+1,0} \approx \mathbf{u}_\Gamma(T_0)$ simultaneously:

$$(7.1) \quad \begin{aligned} \left(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t_m} + \mathbf{A}_{II}^{(m)} \right) \mathbf{u}_I^{(m),k+1,n_m+1} = & - \left(\frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t_m} + \mathbf{A}_{I\Gamma}^{(m)} \right) \mathbf{u}_\Gamma^{k,n_m+1} \\ & + \frac{\mathbf{M}_{II}^{(m)}}{\Delta t_m} \mathbf{u}_I^{(m),k+1,n_m} + \frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t_m} \mathbf{u}_\Gamma^{k,n_m} \end{aligned}$$

for $m = 1, 2$. Note that interpolation is not needed to solve the Dirichlet problems because $\mathbf{u}_I^{(1),k+1,n_1+1}$ in (7.1) is only dependent on terms related to Ω_1 . In the same way, $\mathbf{u}_I^{(2),k+1,n_2+1}$ in (7.1) only depends on n_2 .

We compute now the fluxes $\tilde{\mathbf{F}}_1^{k,\tau_1} := \tilde{\mathbf{f}}_1^{k,\tau_1} + I(\tau_1, \tau_2, \tilde{\mathbf{f}}_2^{k,\tau_2})$ and $\tilde{\mathbf{F}}_2^{k,\tau_2} := \tilde{\mathbf{f}}_2^{k,\tau_2} + I(\tau_2, \tau_1, \tilde{\mathbf{f}}_1^{k,\tau_1})$ in (5.6) with

$$(7.2) \quad \begin{aligned} \tilde{\mathbf{f}}_m^{k,n_m} = & \left(\frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t_m} + \mathbf{A}_{I\Gamma}^{(m)} \right) \mathbf{u}_\Gamma^{k,n_m+1} + \left(\frac{\mathbf{M}_{\Gamma I}^{(m)}}{\Delta t_m} + \mathbf{A}_{\Gamma I}^{(m)} \right) \mathbf{u}_I^{(m),k+1,n_m+1} \\ & - \frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t_m} \mathbf{u}_\Gamma^{k,n_m} - \frac{\mathbf{M}_{\Gamma I}^{(m)}}{\Delta t_m} \mathbf{u}_I^{(m),k+1,n_m}, \end{aligned}$$

where $n_m = 1, \dots, N_m$ and $\tau_m = \{t_1, t_2, \dots, t_{N_m}\}$ are the corresponding time grids on Ω_m for $m = 1, 2$. Note that unlike in the Dirichlet problems, we need to use the interpolation described in the previous section. We use it to calculate $\tilde{\mathbf{F}}_1^{k,\tau_1}$ and $\tilde{\mathbf{F}}_2^{k,\tau_2}$ because their components run over different time integrations (one indicated by n_1 and the other by n_2).

One can now rewrite the Neumann problems in (5.4) in terms of the vector of unknowns

$$\psi_m^{k+1,n_m+1} := \left(\psi_I^{(m),k+1,n_m+1T} \psi_\Gamma^{(m),k+1,n_m+1T} \right)^T.$$

solves the two Neumann problems for $n_m = 1, 2, \dots, N_m$ with $\psi_m^{k+1,0} \approx \psi_m(T_0)$, $m = 1, 2$, in parallel:

$$(7.3) \quad \left(\frac{\mathbf{M}_m}{\Delta t_m} + \mathbf{A}_m \right) \psi_m^{k+1,n_m+1} = \frac{\mathbf{M}_m}{\Delta t_m} \psi_m^{k+1,n_m} + \tilde{\mathbf{b}}^{k,n_m},$$

where

$$\tilde{\mathbf{b}}^{k,n_m} = \left(\mathbf{0}^T \tilde{\mathbf{F}}_m^{k,n_mT} \right)^T.$$

Then, the interface values are updated, respectively, by

$$(7.4) \quad \mathbf{u}_\Gamma^{k+1,\tau_1} = \mathbf{u}_\Gamma^{k,\tau_1} - \Theta \left(\psi_\Gamma^{(1),k+1,\tau_1} + I \left(\tau_1, \tau_2, \psi_\Gamma^{(2),k+1,\tau_2} \right) \right),$$

$$(7.5) \quad \mathbf{u}_\Gamma^{k+1,\tau_2} = \mathbf{u}_\Gamma^{k,\tau_2} - \Theta \left(\psi_\Gamma^{(2),k+1,\tau_2} + I \left(\tau_2, \tau_1, \psi_\Gamma^{(1),k+1,\tau_1} \right) \right).$$

Here, interpolation is needed to perform the additions because $\psi_{\Gamma}^{(1),k+1,\tau_1}$ and $\psi_{\Gamma}^{(2),k+1,\tau_2}$ correspond to different time integrations. Note that (7.4) and (7.5) correspond to the same interface function, but evaluated on different time grids τ_1 or τ_2 .

Finally, if the termination criteria $\|\mathbf{u}_{\Gamma}^{k+1,N_m} - \mathbf{u}_{\Gamma}^{k,N_m}\| \approx \|\mathbf{u}_{\Gamma}^{k+1}(T_f) - \mathbf{u}_{\Gamma}^k(T_f)\|$ is not small enough, one starts the process from (7.1) once more. Figure 1 sketches the communication needed for the NNWR algorithm just explained.

7.2. SDIRK2. As an alternative, we introduce here a higher order version of the multirate algorithm presented above. Specifically, we consider SDIRK2 as a basis to discretize the systems (5.1), (5.4), and (5.7) in time. Consider an autonomous initial value problem

$$(7.6) \quad \dot{\mathbf{u}}(t) = \mathbf{f}(\mathbf{u}(t)), \quad \mathbf{u}(0) = \mathbf{u}_0.$$

An SDIRK method is then defined as

$$(7.7) \quad \begin{aligned} \mathbf{U}^i &= \mathbf{u}^n + \Delta t_n \sum_{k=1}^i a_{ik} \mathbf{f}(\mathbf{U}^k), \quad i = 1, \dots, j, \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \Delta t_n \sum_{i=1}^j b_i \mathbf{f}(\mathbf{U}^i) \end{aligned}$$

with given coefficients a_{ik} and b_i . The two-stage method SDIRK2 is defined by the coefficients in the following Butcher array:

$$\begin{array}{c|cc} a & a & 0 \\ 1 & 1-a & a \\ \hline & 1-a & a \end{array}$$

with $a = 1 - \frac{1}{2}\sqrt{2}$. As the coefficients a_{2i} and b_i for $i = 1, 2$ are identical, the second equation in (7.7) is superfluous because $\mathbf{u}^{n+1} = \mathbf{U}^2$. The vectors $\mathbf{k}_i = \mathbf{f}(\mathbf{U}^i)$ are called stage derivatives and j is the number of stages. Since the starting vector

$$(7.8) \quad \mathbf{s}_i = \mathbf{u}^n + \Delta t_n \sum_{k=1}^{i-1} a_{ik} \mathbf{k}_k, \quad i = 1, \dots, j-1,$$

is known, (7.7) is just a sequence of implicit Euler steps.

Applying SDIRK2 with time step Δt_1 on Ω_1 and with time step Δt_2 on Ω_2 we can write the systems (5.1), (5.4), and (5.7) in a fully discrete form. This algorithm preserves more or less the same structure as the one presented above for implicit Euler. The main difference lies in the fact that now both the Dirichlet and the Neumann solvers have to take into account the two stages of SDIRK2 and the interpolation has to be applied for each stage.

Therefore, at each fixed point iteration k , let $\mathbf{s}_1^{(m)} = \mathbf{u}_I^{(m),k+1,n_m}$ and $\mathbf{s}_2^{(m)} = \mathbf{u}_I^{(m),k+1,n_m} + \Delta t_m(1-a)\mathbf{k}_1^{(m)}$ be the starting vectors. Then, one first solves the two Dirichlet problems for $n_m = 1, 2, \dots, N_m$ with $\mathbf{u}_I^{(m),k+1,0}$, $m = 1, 2$, $\mathbf{u}_{\Gamma}^{k+1,0}$, simultaneously:

$$(7.9) \quad \begin{aligned} \left(\frac{\mathbf{M}_{II}^{(m)}}{a\Delta t_m} + \mathbf{A}_{II}^{(m)} \right) \mathbf{U}_j^{(m)} &= \frac{\mathbf{M}_{II}^{(m)}}{a\Delta t_m} \mathbf{s}_j^{(m)} - \mathbf{M}_{I\Gamma}^{(m)} \dot{\mathbf{u}}_{\Gamma}^{k,n_m+j-1+(2-j)a} \\ &\quad - \mathbf{A}_{I\Gamma}^{(m)} \mathbf{u}_{\Gamma}^{k,n_m+j-1+(2-j)a}, \quad j = 1, 2, \\ \mathbf{u}_I^{(m),k+1,n_m+1} &= \mathbf{U}_2^{(m)}, \end{aligned}$$

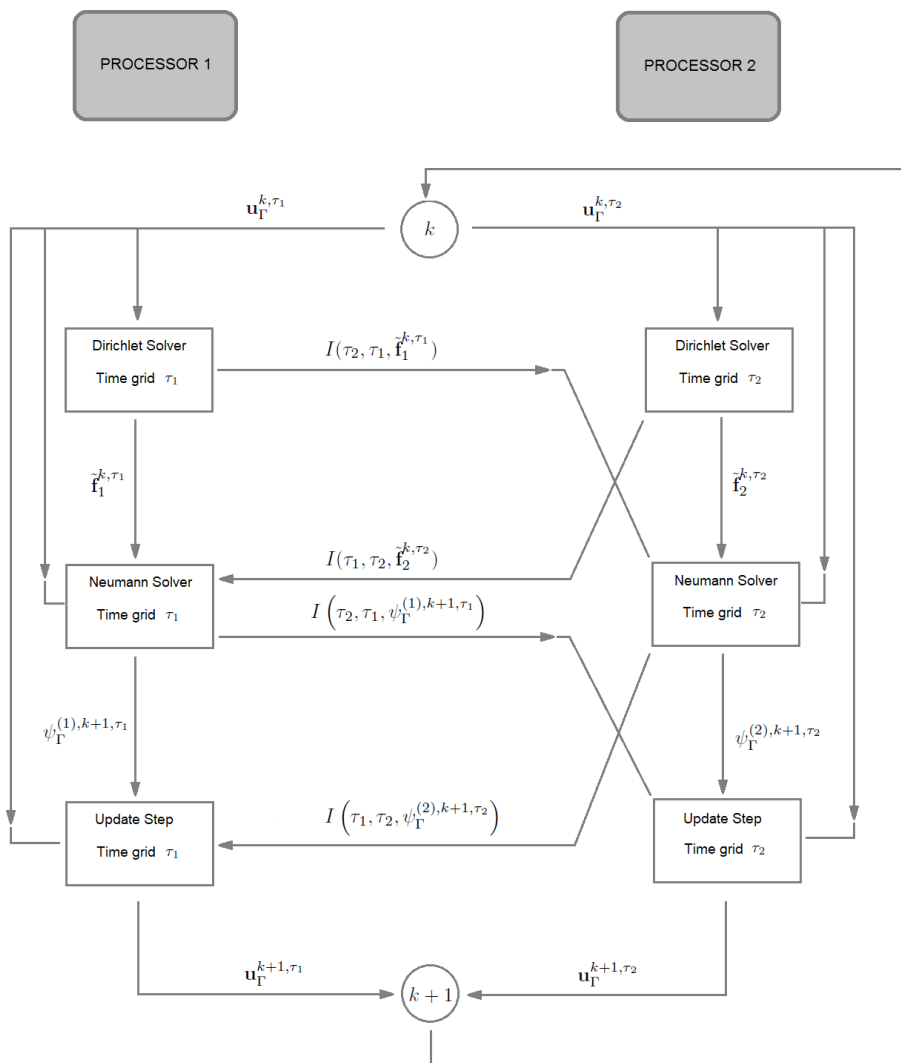


FIG. 1. Illustration of the NNWR algorithm using implicit Euler. The process starts with the space-time interface functions u_Γ^{k,τ_m} , $\tau_m = \{t_1, t_2, \dots, t_{N_m}\}$ for $m = 1, 2$, corresponding to the two nonconforming time grids. Those are needed to run the Dirichlet solvers in parallel getting $u_\Gamma^{(m),k+1,\tau_m}$, $m = 1, 2$. In order to run the Neumann solvers for the corrections of the solution, one needs to provide the fluxes \tilde{f}_1^{k,τ_1} , \tilde{f}_2^{k,τ_2} and their corresponding interpolations $I(\tau_2, \tau_1, \tilde{f}_1^{k,\tau_1})$, $I(\tau_1, \tau_2, \tilde{f}_2^{k,\tau_2})$. One can then run the Neumann problems in parallel getting the corrections $\psi_\Gamma^{(1),k+1,\tau_1}$ and $\psi_\Gamma^{(2),k+1,\tau_2}$ at the space-time interface. Finally, those and their interpolations $I(\tau_1, \tau_2, \psi_\Gamma^{(2),k+1,\tau_2})$ and $I(\tau_2, \tau_1, \psi_\Gamma^{(1),k+1,\tau_1})$ are used to update the space-time interface values. If needed, the process is restarted.

where $u_\Gamma^{(m),k,n_m}$, $U_j^{(m)}$, $s_j^{(m)}$, $k_j^{(m)} \in \mathbb{R}^{R_m}$, and $u_\Gamma^{k,n_m} \in \mathbb{R}^s$. The stage derivatives are given by $k_j^{(m)} = \frac{1}{a\Delta t_m}(U_j^{(m)} - s_j^{(m)})$. Note that the index $m = 1, 2$ denotes the subdomain and the index $j = 1, 2$ denotes the stage.

We compute now the fluxes $\mathbf{F}_j^{(1),k,\tau_1} := \mathbf{f}_j^{(1),k,\tau_1} + I(\tau_1, \tau_2, \mathbf{f}_j^{(2),k,\tau_2})$, $\mathbf{F}_j^{(2),k,\tau_2} := \mathbf{f}_j^{(2),k,\tau_2} + I(\tau_2, \tau_1, \mathbf{f}_j^{(1),k,\tau_1})$ in (5.6) with

$$(7.10) \quad \begin{aligned} \mathbf{f}_j^{(m),k,n_m} = & \mathbf{M}_{\Gamma\Gamma}^{(m)} \dot{\mathbf{u}}_{\Gamma}^{k,n_m+j-1+(2-j)a} + \mathbf{M}_{\Gamma I}^{(m)} \mathbf{k}_j^{(m)} \\ & + \mathbf{A}_{\Gamma\Gamma}^{(m)} \mathbf{u}_{\Gamma}^{k,n_m+j-1+(2-j)a} + \mathbf{A}_{\Gamma I}^{(m)} \mathbf{U}_j^{(m)} \end{aligned}$$

for $m = 1, 2$. Note that interpolation here is needed because the components of $\mathbf{F}_j^{(1),k,\tau_1}$ and $\mathbf{F}_j^{(2),k,\tau_2}$ for the two stages $j = 1, 2$ correspond to different time integrations.

One can now rewrite the Neumann problems in (5.4) in terms of the vector of unknowns $\psi_m^{k+1,n_m+1} := (\psi_I^{(m),k+1,n_m+1T} \psi_{\Gamma}^{(m),k+1,n_m+1T})^T$, where $\psi_I^{(m),k+1,n_m+1} \in \mathbb{R}^{R_m}$ and $\psi_{\Gamma}^{(m),k+1,n_m+1} \in \mathbb{R}^s$. Let $\mathbf{s}_1^{(m)} = \psi_m^{k+1,n_m}$ and $\mathbf{s}_2^{(m)} = \psi_m^{k+1,n_m} + \Delta t_m(1 - a)\mathbf{k}_1^{(m)}$ be the starting vectors. One then solves the two Neumann problems for $n_m = 1, 2, \dots, N_m$ with $\psi_m^{k+1,0} = \psi_m^{k+1}(T_0)$, $m = 1, 2$, in parallel:

$$(7.11) \quad \begin{aligned} \left(\frac{\mathbf{M}_m}{a\Delta t_m} + \mathbf{A}_m \right) \mathbf{Y}_j^{(m)} &= \frac{\mathbf{M}_m}{a\Delta t_m} \mathbf{s}_j^{(m)} + \mathbf{b}_j^{(m),k,n_m}, \quad j = 1, 2, \\ \psi_m^{k+1,n_m+1} &= \mathbf{Y}_2^{(m)}, \end{aligned}$$

where $\mathbf{Y}_j^{(m)}$, $\mathbf{s}_j^{(m)}$, $\mathbf{b}_j^{(m),k,n_m}$, $\mathbf{k}_j^{(m)} \in \mathbb{R}^{R_m+s}$, $\mathbf{k}_j^{(m)} = \frac{1}{a\Delta t_m}(\mathbf{Y}_j^{(m)} - \mathbf{s}_j^{(m)})$, and $\mathbf{b}_j^{(m),k,n_m} = (\mathbf{0}^T \mathbf{F}_j^{(m),k,n_mT})^T$.

Then, the interface values are updated, respectively, by

$$(7.12) \quad \mathbf{u}_{\Gamma}^{k+1,\tau_1} = \mathbf{u}_{\Gamma}^{k,\tau_1} - \Theta \left(\psi_{\Gamma}^{(1),k+1,\tau_1} + I \left(\tau_1, \tau_2, \psi_{\Gamma}^{(2),k+1,\tau_2} \right) \right),$$

$$(7.13) \quad \mathbf{u}_{\Gamma}^{k+1,\tau_2} = \mathbf{u}_{\Gamma}^{k,\tau_2} - \Theta \left(\psi_{\Gamma}^{(2),k+1,\tau_2} + I \left(\tau_2, \tau_1, \psi_{\Gamma}^{(1),k+1,\tau_1} \right) \right).$$

Here, interpolation is needed because $\psi_{\Gamma}^{(1),k+1,\tau_1}$ and $\psi_{\Gamma}^{(2),k+1,\tau_2}$ are nonconforming.

Finally, if the termination criteria $\|\mathbf{u}_{\Gamma}^{k+1,N_m} - \mathbf{u}_{\Gamma}^{k,N_m}\|$ is not small enough, one starts the process from (7.9) once more.

A linear interpolation through $(t_{n_m}, \mathbf{u}_{\Gamma}^{k,n_m})$ and $(t_{n_m} + \Delta t_m, \mathbf{u}_{\Gamma}^{k,n_m+1})$ is used in order to approximate $\mathbf{u}_{\Gamma}^{k,n_m+a}$ in the first equation of (7.9) and in the fluxes (7.10), i.e.,

$$(7.14) \quad \mathbf{u}_{\Gamma}^{k,n_m+a} \approx \mathbf{u}_{\Gamma}^{k,n_m} + a \left(\mathbf{u}_{\Gamma}^{k,n_m+1} - \mathbf{u}_{\Gamma}^{k,n_m} \right).$$

Furthermore, there are first order time derivatives in the first equation of (7.9) and in (7.10). We use forward differences to approximate all the remaining first order derivatives:

$$(7.15) \quad \dot{\mathbf{u}}_{\Gamma}^{k,n_m+j-1+(2-j)a} \approx \frac{\mathbf{u}_{\Gamma}^{k,n_m+1} - \mathbf{u}_{\Gamma}^{k,n_m}}{\Delta t_m}$$

for $j = 1, 2$ and $m = 1, 2$. Summarizing, the SDIRK2-NNWR algorithm just presented has the same structure as the implicit Euler NNWR algorithm described previously and sketched in Figure 1. The main difference is that the whole procedure is repeated twice, once for each stage.

8. Derivation of the iteration matrix. We are interested in the performance of the NNWR algorithm. As the rate of convergence of a linear iteration is given by the spectral radius of its iteration matrix, we derive in this section the iteration matrix with respect to the set of unknowns at the space-time interface for implicit Euler. A similar analysis to find the convergence rates of the Dirichlet–Neumann iteration for the unsteady transmission problem can be found in [29]. We intentionally avoid a derivation for SDIRK2 and we will show in the numerical results section that NNWR-SDIRK2 behaves as predicted by the analysis of implicit Euler. Similarly, we assume that we have conforming time grids, i.e., $\Delta t := \Delta t_1 = \Delta t_2$, because we do not want the space-time interpolation to interfere in the analysis. We will see later how far the analysis performed for the nonmultirate case is applicable to the multirate case.

The goal now is to find the iteration matrix Λ with respect to the final synchronization point $\mathbf{u}_\Gamma^{N_m} \approx \mathbf{u}_\Gamma(T_f)$ because the global error over the time window $[T_0, T_f]$ is assumed to be increasing, having its maximum at the final time T_f . Thus, we will find Λ such that

$$(8.1) \quad \mathbf{u}_\Gamma^{k+1, N_m} = \Lambda \mathbf{u}_\Gamma^{k, N_m} + \sum_{i=1}^2 (\varphi^{k+1, \tilde{\tau}_i} + \varphi^{k, \tau_i}),$$

where φ^{k, τ_m} are terms dependent on solutions at the previous fixed point iteration k for the time grids $\tau_m = \{t_1, t_2, \dots, t_{N_m}\}$, $m = 1, 2$, and $\varphi^{k+1, \tilde{\tau}_m}$ are terms dependent on solutions at the current iteration $k+1$ but for the time grids $\tilde{\tau}_m = \{t_1, t_2, \dots, t_{N_m-1}\} \subset \tau_m$, $m = 1, 2$. To perform the analysis, we neglect all the solutions at previous time steps (indicated by $\varphi^{k+1, \tilde{\tau}_m}$). Thus, we do not find the exact rate of convergence when having more than one single time step, but instead a good estimate.

We now rewrite (7.1), (7.3), and (7.4)–(7.5) as an iteration for $\mathbf{u}_\Gamma^{k+1, N_m}$. As we chose above, we omit all the terms in (8.1) except for the first two. We isolate the term $\mathbf{u}_I^{(m), k+1, N_m}$ from (7.1) and $\psi_I^{(m), k+1, N_m}$ from the first equation in (7.3) leading to

$$(8.2) \quad \mathbf{u}_I^{(m), k+1, N_m} = - \left(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)} \right)^{-1} \left(\frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t} + \mathbf{A}_{I\Gamma}^{(m)} \right) \mathbf{u}_\Gamma^{k, N_m},$$

$$(8.3) \quad \psi_I^{(m), k+1, N_m} = - \left(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)} \right)^{-1} \left(\frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t} + \mathbf{A}_{I\Gamma}^{(m)} \right) \psi_\Gamma^{(m), k+1, N_m}.$$

Inserting (8.2) and (8.3) into the second equation of the system (7.3) we get

$$(8.4) \quad \psi_\Gamma^{(m), k+1, N_m} = \mathbf{S}^{(m)-1} \sum_{i=1}^2 \mathbf{S}^{(i)} \mathbf{u}_\Gamma^{k, N_m},$$

$$(8.5) \quad \mathbf{S}^{(m)} := \left(\frac{\mathbf{M}_{\Gamma\Gamma}^{(m)}}{\Delta t} + \mathbf{A}_{\Gamma\Gamma}^{(m)} \right) - \left(\frac{\mathbf{M}_{\Gamma I}^{(m)}}{\Delta t} + \mathbf{A}_{\Gamma I}^{(m)} \right) \left(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)} \right)^{-1} \left(\frac{\mathbf{M}_{I\Gamma}^{(m)}}{\Delta t} + \mathbf{A}_{I\Gamma}^{(m)} \right).$$

Finally, inserting (8.4) into (7.4) or (7.5) one gets $\mathbf{u}_\Gamma^{k+1, N_m} = \Lambda \mathbf{u}_\Gamma^{k, N_m}$ with

$$(8.6) \quad \Lambda = \mathbf{I} - \Theta \left(2\mathbf{I} + \mathbf{S}^{(1)-1} \mathbf{S}^{(2)} + \mathbf{S}^{(2)-1} \mathbf{S}^{(1)} \right).$$

In the one-dimensional case, the iteration matrix Λ is just a real number and thus its spectral radius is its modulus. Then, the optimal relaxation parameter Θ_{opt} in 1D is given by

$$(8.7) \quad \Theta_{opt} = \frac{1}{2 + \mathbf{S}^{(1)-1} \mathbf{S}^{(2)} + \mathbf{S}^{(2)-1} \mathbf{S}^{(1)}}.$$

9. One-dimensional convergence analysis. So far, the derivation was performed for a rather general discretization. In this section, we study the iteration matrix Λ for a specific finite element (FE) discretization in 1D. We will give a formula for the convergence rates. The behavior of the rates when approaching both the continuous case in time and space is also given.

Specifically, we use $\Omega_1 = [-1, 0]$, $\Omega_2 = [0, 1]$. For the FE discretization, we use the standard piecewise-linear polynomials as test functions. Here we discretize Ω_m into $N + 1$ equal sized cells of size $\Delta x = 1/(N + 1)$ for $m = 1, 2$.

With $\mathbf{e}_j = (0 \ \cdots \ 0 \ 1 \ 0 \ \cdots \ 0)^T \in \mathbb{R}^N$, where the only nonzero entry is located at the j th position, the discretization matrices are given by

$$\mathbf{A}_{II}^{(m)} = \frac{\lambda_m}{\Delta x^2} \begin{pmatrix} 2 & -1 & & 0 \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{pmatrix}, \quad \mathbf{M}_{II}^{(m)} = \frac{\alpha_m}{6} \begin{pmatrix} 4 & 1 & & 0 \\ 1 & 4 & \ddots & \\ & \ddots & \ddots & 1 \\ 0 & & 1 & 4 \end{pmatrix},$$

$\mathbf{M}_{\Gamma\Gamma}^{(m)} = 2\alpha_m/6$, $\mathbf{A}_{\Gamma\Gamma}^{(m)} = \lambda_m/\Delta x^2$, $\mathbf{A}_{\Gamma I}^{(1)} = -(\lambda_1/\Delta x^2)\mathbf{e}_N$, $\mathbf{A}_{\Gamma I}^{(2)} = -(\lambda_2/\Delta x^2)\mathbf{e}_1$, $\mathbf{M}_{\Gamma I}^{(1)} = (\alpha_1/6)\mathbf{e}_N$, $\mathbf{M}_{\Gamma I}^{(2)} = (\alpha_2/6)\mathbf{e}_1$, $\mathbf{M}_{\Gamma I}^{(1)} = (\alpha_1/6)\mathbf{e}_N^T$, $\mathbf{M}_{\Gamma I}^{(2)} = (\alpha_2/6)\mathbf{e}_1^T$, $\mathbf{A}_{\Gamma I}^{(1)} = -(\lambda_1/\Delta x^2)\mathbf{e}_N^T$, $\mathbf{A}_{\Gamma I}^{(2)} = -(\lambda_2/\Delta x^2)\mathbf{e}_1^T$. Here, $\mathbf{A}_{II}^{(m)}, \mathbf{M}_{II}^{(m)} \in \mathbb{R}^{N \times N}$, $\mathbf{A}_{\Gamma I}^{(m)}, \mathbf{M}_{\Gamma I}^{(m)} \in \mathbb{R}^{N \times 1}$ and $\mathbf{A}_{\Gamma\Gamma}^{(m)}, \mathbf{M}_{\Gamma\Gamma}^{(m)} \in \mathbb{R}^{1 \times N}$ for $m = 1, 2$.

One computes $\mathbf{S}^{(m)}$ for $m = 1, 2$, by inserting the corresponding matrices specified above in (8.5) obtaining

$$(9.1) \quad \mathbf{S}^{(m)} = \left(\frac{\alpha_m}{3\Delta t} + \frac{\lambda_m}{\Delta x^2} \right) - \left(\frac{\alpha_m}{6\Delta t} - \frac{\lambda_m}{\Delta x^2} \right)^2 \mathbf{e}_X^T \left(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)} \right)^{-1} \mathbf{e}_X,$$

where $X = N$ for $m = 1$ and $X = 1$ for $m = 2$. Rewriting the matrices $(\frac{\mathbf{M}_{II}^{(m)}}{\Delta t} + \mathbf{A}_{II}^{(m)})^{-1}$ in terms of their eigendecomposition, lengthy computations give the following explicit expressions for $\mathbf{S}^{(m)}$ (see [29, sec. 5] for details):

$$(9.2) \quad \mathbf{S}^{(m)} = \frac{6\Delta t \Delta x (\alpha_m \Delta x^2 + 3\lambda_m \Delta t) - (\alpha_m \Delta x^2 - 6\lambda_m \Delta t)^2 w_m}{18\Delta t^2 \Delta x^3},$$

where

$$(9.3) \quad w_m = \sum_{i=1}^N \frac{3\Delta t \Delta x^2 \sin^2(i\pi \Delta x)}{2\alpha_m \Delta x^2 + 6\lambda_m \Delta t + (\alpha_m \Delta x^2 - 6\lambda_m \Delta t) \cos(i\pi \Delta x)}.$$

With this we obtain an explicit formula for the optimal relaxation parameter Θ_{opt} in (8.7):

$$(9.4) \quad \Theta_{opt} = \left(2 + \frac{6\Delta t \Delta x (\alpha_2 \Delta x^2 + 3\lambda_2 \Delta t) - (\alpha_2 \Delta x^2 - 6\lambda_2 \Delta t)^2 w_2}{6\Delta t \Delta x (\alpha_1 \Delta x^2 + 3\lambda_1 \Delta t) - (\alpha_1 \Delta x^2 - 6\lambda_1 \Delta t)^2 w_1} + \frac{6\Delta t \Delta x (\alpha_1 \Delta x^2 + 3\lambda_1 \Delta t) - (\alpha_1 \Delta x^2 - 6\lambda_1 \Delta t)^2 w_1}{6\Delta t \Delta x (\alpha_2 \Delta x^2 + 3\lambda_2 \Delta t) - (\alpha_2 \Delta x^2 - 6\lambda_2 \Delta t)^2 w_2} \right)^{-1}.$$

We could not find a way of simplifying the finite sum (9.3) because Δx depends on N (i.e., $\Delta x = 1/(N+1)$). However, (9.4) is a computable expression that gives the optimal relaxation parameter Θ_{opt} of the NNWR algorithm using implicit Euler for given Δx , Δt , α_m , and λ_m , $m = 1, 2$.

We are now interested in the asymptotics of (9.4) with respect to both spatial and temporal resolutions. To this end, we rewrite (9.4) in terms of $c := \Delta t/\Delta x^2$:

$$(9.5) \quad \Theta_{opt} = \left(2 + \frac{6\Delta t(\alpha_2 + 3\lambda_2 c) - \Delta x(\alpha_2 - 6\lambda_2 c)^2 w'_2}{6\Delta t(\alpha_1 + 3\lambda_1 c) - \Delta x(\alpha_1 - 6\lambda_1 c)^2 w'_1} + \frac{6\Delta t(\alpha_1 + 3\lambda_1 c) - \Delta x(\alpha_1 - 6\lambda_1 c)^2 w'_1}{6\Delta t(\alpha_2 + 3\lambda_2 c) - \Delta x(\alpha_2 - 6\lambda_2 c)^2 w'_2} \right)^{-1},$$

where

$$(9.6) \quad w'_m = \sum_{i=1}^N \frac{3\Delta t \sin^2(i\pi\Delta x)}{2\alpha_m + 6\lambda_m c + (\alpha_m - 6\lambda_m c) \cos(i\pi\Delta x)}$$

for $m = 1, 2$.

Finally, we compute the limits $c \rightarrow 0$ and $c \rightarrow \infty$ of the iteration matrix Λ . The limit $c \rightarrow 0$ leads to the continuous case with respect to time and the limit $c \rightarrow \infty$ leads to the continuous case with respect to space. On one hand, $\lim_{c \rightarrow 0} \Lambda = 1 - \Theta(2 + \alpha_2/\alpha_1 + \alpha_1/\alpha_2) = 1 - \Theta((\alpha_1 + \alpha_2)^2/\alpha_1\alpha_2)$. On the other hand, $\lim_{c \rightarrow \infty} \Lambda = 1 - \Theta(2 + \lambda_2/\lambda_1 + \lambda_1/\lambda_2) = 1 - \Theta((\lambda_1 + \lambda_2)^2/\lambda_1\lambda_2)$. Consequently,

$$(9.7) \quad \Theta_{\{c \rightarrow 0\}} = \frac{\alpha_1 \alpha_2}{(\alpha_1 + \alpha_2)^2},$$

$$(9.8) \quad \Theta_{\{c \rightarrow \infty\}} = \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)^2}.$$

The result obtained in (9.8) is consistent with the one-dimensional semidiscrete analysis performed in [22]. There, a convergence analysis for the NNWR method in (4.1), (4.2), and (4.3) with constant coefficients using Laplace transforms shows that $\Theta_{opt} = 1/4$ when the two subdomains Ω_1 and Ω_2 are identical. Their result is recovered by our analysis when one approaches the continuous case in space in (9.8) for constant coefficients, i.e., $\lambda_1 = \lambda_2$. In that case, one gets

$$(9.9) \quad \Theta_{opt} = \frac{\lambda_1 \lambda_2}{(\lambda_1 + \lambda_2)^2} = \frac{\lambda_1^2}{4\lambda_1^2} = \frac{1}{4}.$$

10. Numerical results. All the results in this section have been produced by implementing algorithms 4 and 7 in Python using the FE discretization specified in the previous section. First, we show numerically that the NNWR algorithm using implicit Euler preserves first order and using SDIRK2 second order. Second, we show

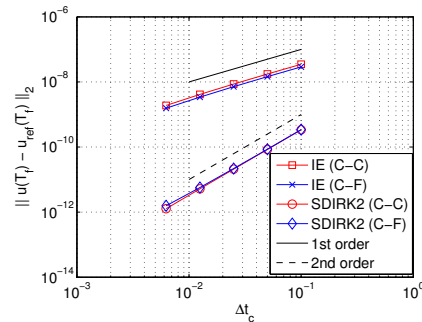


FIG. 2. Error plot of the coupled solution at the final time T_f for three different time grids. $\Delta x = 1/200$, $[T_0, T_f] = [0, 1]$, and $TOL = 1e-15$. Air-steel thermal interaction using implicit Euler and air-water thermal interaction using SDIRK2.

TABLE 1

Physical properties of the materials. λ is the thermal conductivity, ρ the density, and c_p the specific heat capacity and $\alpha = \rho c_p$.

Material	λ (W/mK)	ρ (kg/m ³)	c_p (J/kgK)	α (J/K m ³)
Air	0.0243	1.293	1005	1299.5
Water	0.58	999.7	4192.1	4.1908e6
Steel	48.9	7836	443	3471348

the validity of (9.4) as an estimator for the optimal relaxation parameter Θ_{opt} of the NNWR algorithm using implicit Euler. We also show that (9.4) is a good estimator for the multirate case both using implicit Euler and SDIRK2 and also for two-dimensional examples. Furthermore, we also show that the theoretical asymptotics deduced both in (9.7) and (9.8) match the numerical experiments. Finally, a comparison between the Dirichlet–Neumann and the Neuman–Neumann methods is included.

10.1. NNWR results. Figure 2 shows the error plots of the discrete coupled solution at the final time T_f for the coupling of different materials using both implicit Euler and SDIRK2. Physical properties of the materials are shown in Table 1. We have considered two initial time grids (for $\Delta t_c = 1/10$ and $\Delta t_f = 1/100$ given), which we then refine several times by a factor of 2:

- (C-C): Two coarse conforming time grids with $\Delta t_1 = \Delta t_2 = \Delta t_c$.
- (C-F): Nonconforming time grids with $\Delta t_1 = \Delta t_c$ and $\Delta t_2 = \Delta t_f$.

When coupling two different materials in the multirate case, we always assign the finer grid to the material that has higher heat conductivity because it performs the heat changes faster. In space, we fix $\Delta x = 1/200$ and we compute a reference solution by solving (2.1) directly on a very fine time grid, with $\Delta t = 1/1000$. One observes in Figure 2 that first and second order convergence are obtained in the nonconforming case for implicit Euler and SDIRK2, respectively. Moreover, the errors obtained in the multirate case (C-F) are nearly the same as in the coarser nonmultirate case (C-C). Thus, the accuracy of the multirate case is determined by its coarser rate. This is consistent with [35, 8], where the convergence of the discrete multirate WR algorithm is independent of the ratio of time steps.

Figure 3 compares the behavior of the algorithm described in this paper using implicit Euler (left plot) and SDIRK2 (right plot). It shows the convergence rates in terms of the relaxation parameter Θ for the one-dimensional thermal coupling

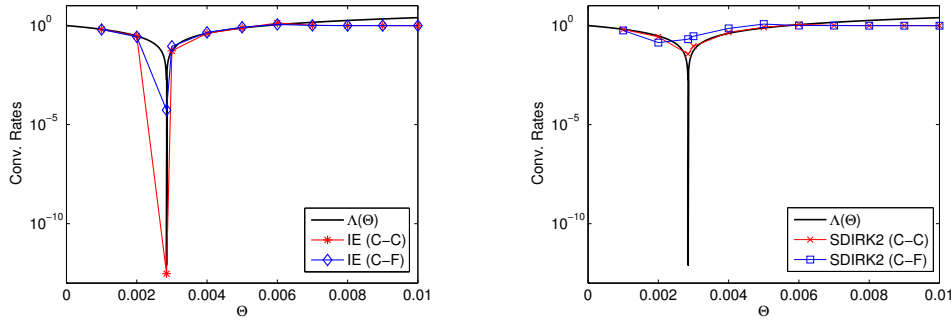


FIG. 3. Air-water convergence rates as a function of the relaxation parameter Θ in 1D. $\Delta x = 1/100$, $\Delta t_c = 100$, and $\Delta t_f = 1$. Left: $\Lambda(\Theta)$ in (8.6) and the experimental convergence rates both for the multirate (C-F) and nonmultirate (C-C) cases using implicit Euler. Right: $\Lambda(\Theta)$ in (8.6) using implicit Euler and the experimental convergence rates both for the multirate (C-F) and nonmultirate (C-C) cases using SDIRK2.

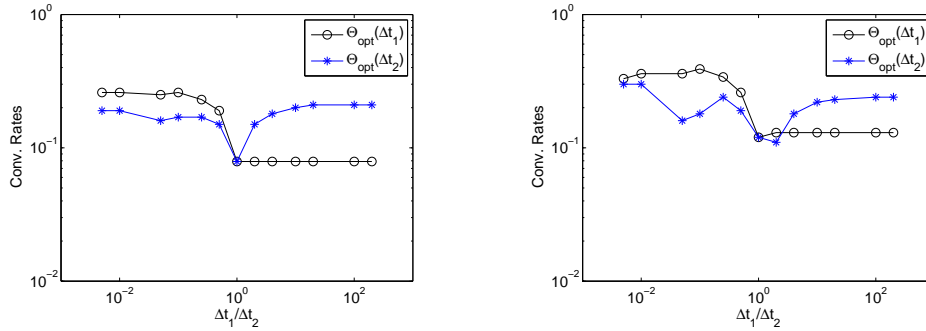


FIG. 4. Convergence rates as a function of the temporal ratio $\Delta t_1/\Delta t_2$ for the air-water coupling in 1D. We plot the convergence rates in the multirate case ($\Delta t_1 \neq \Delta t_2$) using the relaxation parameters $\Theta_{opt}(\Delta t_1)$ and $\Theta_{opt}(\Delta t_2)$ in (9.4). $\Delta t_1/\Delta t_2 = 1e - 3/2e - 1, 2e - 3/2e - 1, 1e - 2/2e - 1, 2e - 2/2e - 1, 5e - 2/2e - 1, 1e - 1/2e - 1, 2e - 1/2e - 1, 2e - 1/1e - 1, 2e - 1/5e - 2, 2e - 1/2e - 2, 2e - 1/1e - 2, 2e - 1/2e - 3, 2e - 1/1e - 3$, and $\Delta x = 1/100$. Left: Implicit Euler. Right: SDIRK2.

between air and water. We have plotted $\Lambda(\Theta)$ in (8.6) with the one-dimensional space discretization specified in section 9 and the experimental convergence rates for both the multirate and nonmultirate cases. The relevance of the analysis presented above is illustrated in Figure 3 because the algorithm is extremely fast at Θ_{opt} (converging in two iterations), but if one deviates slightly from Θ_{opt} , we get a much slower method. As can be seen in the left plot, the experimental convergence rates for the nonmultirate case (C-C) are exactly predicted by the theory. Moreover, our formula also predicts where the convergence rate of the NNWR algorithm in the multirate case (C-F) is minimal. They are not identical because the linear interpolation performed at the space-time interface in the multirate case is neglected in (8.1) to simplify the theoretical analysis. One can also observe in the right plot that $\Lambda(\Theta)$ using implicit Euler estimates quite well the optimal relaxation parameter of the NNWR algorithm using SDIRK2 for both the multirate and nonmultirate cases.

In order to illustrate the behavior of the NNWR method in the multirate case ($\Delta t_1 \neq \Delta t_2$), we have plotted in Figure 4 the convergence rates using the relaxation parameters $\Theta_{opt}(\Delta t_1)$ and $\Theta_{opt}(\Delta t_2)$ in (9.4) with respect to the variation of $\Delta t_1/\Delta t_2$

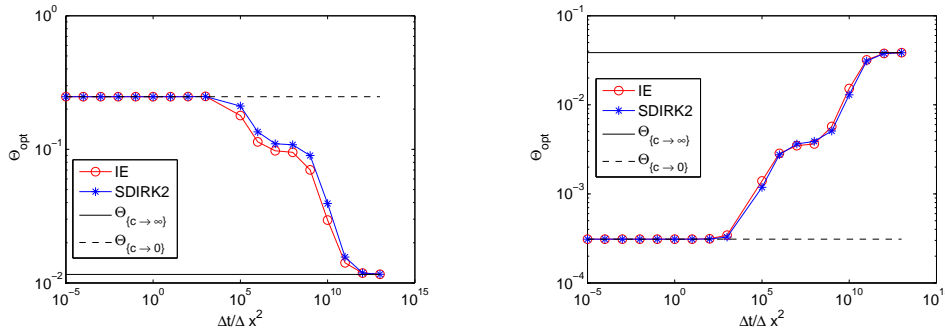


FIG. 5. Optimal relaxation parameter Θ_{opt} as a function of the parameter $c := \Delta t/\Delta x^2$ for both implicit Euler and SDIRK2 in 1D. The constant lines $\Theta_{\{c \rightarrow \infty\}}$ and $\Theta_{\{c \rightarrow 0\}}$ represent the spatial and temporal asymptotics of Θ_{opt} in (9.5). $\Delta t = 1e-9, 1e-8, \dots, 1e8, 1e9$, and $\Delta x = 1/100$. Left: Water-steel coupling. Right: Air-Water coupling.

for the air-water coupling. In Figure 4 we have chosen $\Delta t_1/\Delta t_2 = 1e-3/2e-1, 2e-3/2e-1, 1e-2/2e-1, 2e-2/2e-1, 5e-2/2e-1, 1e-1/2e-1, 2e-1/2e-1, 2e-1/1e-1, 2e-1/5e-2, 2e-1/2e-2, 2e-1/1e-2, 2e-1/2e-3, 2e-1/1e-3$, and $\Delta x = 1/100$. One observes that the multirate method converges fast using any of the two relaxation parameters for both implicit Euler and SDIRK2. Nevertheless, one can also see in Figure 4 that even though we have not performed a specific analysis for the optimal relaxation parameter in the multirate case, the Θ_{opt} in (9.4) can be used as an estimator. More specifically, we conclude from Figure 4 that one can use $\Theta_{opt}(\Delta t_2)$ when $\Delta t_1 < \Delta t_2$ and $\Theta_{opt}(\Delta t_1)$ when $\Delta t_1 > \Delta t_2$. Furthermore, one can also observe in Figure 4 that the convergence rate in the multirate case $\Delta t_1 > \Delta t_2$ using $\Theta_{opt}(\Delta t_1)$ does not deviate from the result derived for the nonmultirate case.

Figure 5 shows the optimal relaxation parameter Θ_{opt} with respect to the parameter $c := \Delta t/\Delta x^2$ using both implicit Euler and SDIRK2. We have chosen $\Delta t = 1e-9, 1e-8, \dots, 1e8, 1e9$, and $\Delta x = 1/100$. For implicit Euler, we have plotted the function $\Theta_{opt}(c)$ in (9.5). For SDIRK2, we have plotted a sister function $\Theta_{opt}(c)$ that can be found applying exactly the derivation presented in sections 8 and 9 to the discretized SDIRK2-NNWR method introduced in section 7.2. One can see that the two time discretization methods have a similar behavior when varying c . This illustrates why the optimal relaxation parameter Θ_{opt} computed in (9.4) for implicit Euler is also valid for SDIRK2 as observed in Figure 3. Furthermore, in Figure 5 we observe that the optimal relaxation parameter for any given Δt and Δx is always between the bounds of the theoretical asymptotics deduced both in (9.7) and (9.8), tending to them in the temporal and spatial limits, respectively.

We now want to demonstrate that the one-dimensional formula (9.4) is a good estimator for the optimal relaxation parameter Θ_{opt} in 2D. Thus, we now consider a two-dimensional version of the model problem consisting of two coupled linear heat equations on two identical unit squares, e.g., $\Omega_1 = [-1, 0] \times [0, 1]$ and $\Omega_2 = [0, 1] \times [0, 1]$.

Figure 6 shows the convergence rates in terms of the relaxation parameter Θ for two-dimensional examples. On the left we have the thermal coupling between air and steel and on the right between air and water. One can observe that the convergence rates of the NNWR method using Θ_{opt} from (9.4) in the two-dimensional examples are worse than in 1D, but still optimal. Hence, we suggest to use Θ_{opt} in 2D as well; otherwise the method is much slower.

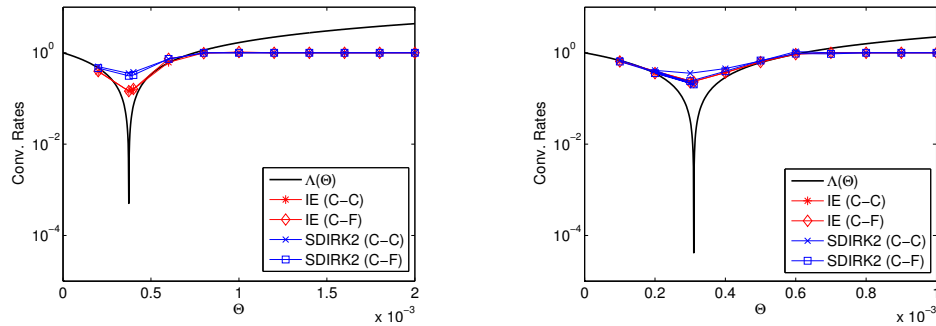


FIG. 6. Convergence rates as a function of the relaxation parameter Θ in 2D. $\Lambda(\Theta)$ in (9.4) using implicit Euler and the experimental convergence rates both for the multirate (C-F) and the nonmultirate (C-C) cases using implicit Euler and SDIRK2. $\Delta x = 1/10$, $\Delta t_c = 1/10$, and $\Delta t_f = 1/50$. Left: Air-steel coupling. Right: Air-water coupling.

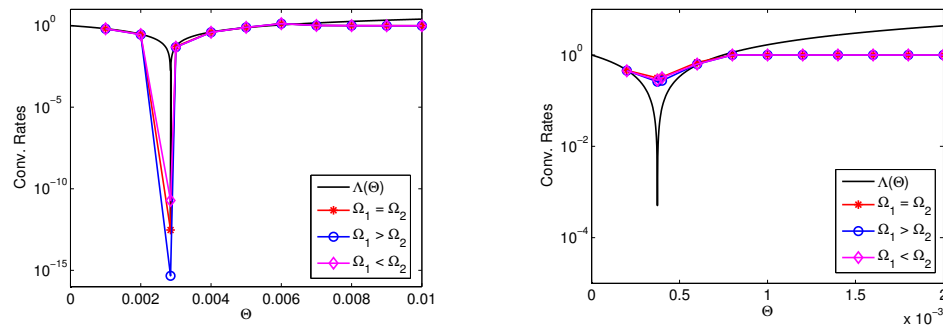


FIG. 7. Convergence rates as a function of the relaxation parameter Θ for different distributions of the subdomains. $\Lambda(\Theta)$ in (9.4) using implicit Euler and the experimental convergence rates for two equivalent subdomains ($\Omega_1 = \Omega_2$), $\Omega_1 > \Omega_2$, and $\Omega_1 < \Omega_2$ using implicit Euler. $\Delta x = 1/10$ and $\Delta t_1 = \Delta t_2 = 1/10$. Left: Air-water coupling in 1D. Right: Air-steel coupling in 2D.

To conclude this section Figure 7 shows that the convergence rates of the NNWR algorithm do not get affected when changing the length subdomains in 1D and 2D. In particular, Figure 7 shows the one-dimensional air-water coupling and the two-dimensional air-steel coupling for equal subdomains ($\Omega_1 = [0, 1] \equiv [1, 2] = \Omega_2$ in 1D or $\Omega_1 = [0, 1] \times [0, 1] \equiv [1, 2] \times [0, 1] = \Omega_2$ in 2D), for Ω_1 larger than Ω_2 ($\Omega_1 = [0, 1.5] > [1.5, 2] = \Omega_2$ in 1D or $\Omega_1 = [0, 1.5] \times [0, 1] > [1.5, 2] \times [0, 1] = \Omega_2$ in 2D), and for Ω_2 larger than Ω_1 ($\Omega_1 = [0, 0.5] < [0.5, 2] = \Omega_2$ in 1D or $\Omega_1 = [0, 0.5] \times [0, 1] < [0.5, 2] \times [0, 1] = \Omega_2$ in 2D)). One can see that the dimension of the subdomains does not have a significative impact in the behavior of the algorithm presented above.

10.2. NNWR-DNWR comparison. Finally, we compare the Dirichlet-Neumann and the Neumann-Neumann couplings. We consider the FE discretization specified in section 9 and the implicit Euler as a time integration method for both DNWR and NNWR with $\Delta x = 1/500$. In addition, we will use $\Theta = 1/2$ as the optimal relaxation parameter for the DNWR algorithm as suggested in [12] for constant material coefficients. Note that the optimal relaxation parameter for the NNWR method is $\Theta_{opt} = 1/4$ (see (9.4)) in the case of constant coefficients because $\lambda_1 = \lambda_2$ and $\alpha_1 = \alpha_2$.

TABLE 2

Computational effort of DNWR and NNWR for the one-dimensional steel-steel coupling in the nonmultirate case. $\Delta x = 1/500$ and $TOL = 1e - 8$. Number of total iterations in brackets.

Δt	Comp. Time - NNWR (s)	Comp. Time - DNWR (s)
1	0.07 (2 iterations)	0.05 (2 iterations)
1/10	0.42 (2 iterations)	0.3 (2 iterations)
1/50	1.86 (2 iterations)	1.25 (2 iterations)
1/100	3.74 (2 iterations)	2.47 (2 iterations)

TABLE 3

Computational effort comparison of DNWR and NNWR for the one-dimensional steel-steel coupling in the multirate case. $\Delta x = 1/500$ and $TOL = 1e - 8$. Number of total iterations in brackets.

$\Delta t_1 - \Delta t_2$	Comp. Time - NNWR (s)	Comp. Time - DNWR (s)
1/5 - 1/10	0.48 (3 iterations)	0.70 (6 iterations)
1/5 - 1/50	1.5 (3 iterations)	26.98 (71 iterations)
1/5 - 1/100	2.74 (3 iterations)	Not convergent

TABLE 4

Computational effort comparison of DNWR and NNWR for the one-dimensional air-steel coupling in the multirate case. $\Delta x = 1/500$ and $TOL = 1e - 8$. Number of total iterations in brackets.

$\Delta t_1 - \Delta t_2$	Comp. Time - NNWR (s)	Comp. Time - DNWR (s)
1/5 - 1/10	0.47 (3 iterations)	1.24 (12 iterations)
1/5 - 1/50	2.20 (4 iterations)	4.65 (12 iterations)
1/5 - 1/100	3.95 (4 iterations)	8.77 (12 iterations)

Table 2 shows the time needed to solve the one-dimensional steel-steel coupling in the nonmultirate case. The number of total iterations needed to achieve a chosen tolerance of $1e - 8$ is also given. One can see that the DNWR method is slightly more efficient than the NNWR method. Moreover, the NNWR algorithm runs in parallel on two different processors using double the amount of computational power as the DNWR. Thus, the DNWR method is a better option for this case because it is cheaper and faster.

However, the NNWR algorithm beats the DNWR algorithm by far when we move to the multirate environment. This is illustrated in Table 3, where the computational effort used to solve the one-dimensional steel-steel coupling in the multirate case is shown. There, one can see how the number of total iterations needed to achieve a tolerance of $1e - 8$ using DNWR grows exponentially when the difference between Δt_1 and Δt_2 increases. On the contrary, the NNWR method is very efficient even when there is a huge difference between Δt_1 and Δt_2 . Thus, we recommend the NNWR algorithm when coupling two fields with nonconforming time grids.

Finally, we have included a comparison for the one-dimensional air-steel coupling in the multirate case. This interaction between air and steel has the particularity of strong jumps in the material coefficients across the space interface. In this case, we have chosen $\Theta = 1/2$ for DNWR because even though in [12] it is only proved optimal for the case of constant coefficients, they show in the numerical results section that it also applies to an example where the diffusion coefficient varies spatially. Moreover, Θ_{opt} in (9.4) is chosen for the NNWR method. Table 4 shows a comparison of the computational time employed to solve the one-dimensional air-steel coupling in the multirate case. One can see that the NNWR method is more efficient than the DNWR method because it needs many fewer iterations to achieve the same tolerance. Note

that the number of iterations increases when the time resolution decreases for the NNWR method. This happens because the analysis performed in section 8 to find the optimal relaxation parameter takes into account only one single time step (see (8.1)). Therefore, in the case of multiple time steps, Θ_{opt} in (9.4) is a very good choice, but it is not optimal. Besides that, the large amount of iterations performed by the DNWR algorithm hints that $\Theta = 1/2$ might not be the optimal relaxation parameter when having strong jumps in the material coefficients for the fully discrete problem. Thus, performing a specific analysis to find the optimal relaxation parameter of the DNWR algorithm is left for future research.

11. Summary and conclusions. We suggested a new high order parallel NNWR method with nonconforming time grids for two heterogeneous coupled heat equations and studied the optimal relaxation parameter in terms of the material coefficients and the temporal and spatial resolutions Δt and Δx . To this end, we considered the coupling of two heat equations on two identical domains. We assumed structured spatial grids and conforming time grids on both subdomains to derive a formula for the optimal relaxation parameter Θ_{opt} in 1D using implicit Euler. Furthermore, we determined the limits of the optimal relaxation parameter when approaching the continuous case either in space ($\lambda_1 \lambda_2 / (\lambda_1 + \lambda_2)^2$) or time ($\alpha_1 \alpha_2 / (\alpha_1 + \alpha_2)^2$). The method using Θ_{opt} converges extremely fast, typically within two iterations. This was confirmed by numerical results, where we also demonstrated that the nonmultirate one-dimensional case gives excellent estimates for the multirate one-dimensional case and even for multirate and nonmultirate two-dimensional examples using both implicit Euler and SDIRK2. In addition, we have shown that the NNWR method is a more efficient choice than the classical DNWR in the multirate case.

REFERENCES

- [1] A. BANKA, *Practical applications of CFD in heat processing*, Heat Treating Progr., 5 (2005), pp. 44–49.
- [2] P. BIRKEN, T. GLEIM, D. KUHL, AND A. MEISTER, *Fast solvers for unsteady thermal fluid structure interaction*, Internat. J. Numer. Methods Fluids, 79 (2015), pp. 16–29.
- [3] P. BIRKEN, K. QUINT, S. HARTMANN, AND A. MEISTER, *A time-adaptive fluid-structure interaction method for thermal coupling*, Comput. Vis. Sci., 13 (2011), pp. 331–340.
- [4] S. BREMICKER-TRÜBELHORN AND S. ORTLEB, *On multirate GARK schemes with adaptive micro step sizes for fluid-structure interaction: Order conditions and preservation of the geometric conservation law*, Aerospace, 4 (2017).
- [5] J. BUCHLIN, *Convective heat transfer and infrared thermography*, J. Appl. Fluid Mech., 3 (2010), pp. 55–62.
- [6] P. CAUSIN, J. GERBEAU, AND F. NOBILE, *Added-mass effect in the design of partitioned algorithms for fluid-structure problems*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 4506–4527.
- [7] A. CRISTIANO, I. MALOSSO, P. BLANCO, S. DEPARIS, AND A. QUARTERONI, *Algorithms for the partitioned solution of weakly coupled fluid models for cardiovascular flows*, Numer. Methods Biomed. Eng., 27 (2011), pp. 2035–2057.
- [8] M. CROW AND M. ILIC, *The waveform relaxation method for systems of differential/algebraic equations*, in Proceedings of the 29th IEEE Conference on Decision and Control, Vol. 2, 1990, pp. 453–458.
- [9] M. 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, Berlin, 2015, pp. 69–114.
- [10] M. GANDER, L. HALPERN, C. JAPHET, AND V. MARTIN, *Advection diffusion problems with pure advection approximation in subregions*, in Domain Decomposition Methods in Science and Engineering XVI, Lect. Notes Comput. Sci. Eng. 50, Springer, Berlin, 2007, pp. 239–246.
- [11] M. GANDER, L. HALPERN, AND F. NATAF, *Optimized Schwarz waveform relaxation for the one dimensional wave equation*, SIAM J. Numer. Anal., 41 (2003), pp. 1643–1681.

- [12] M. GANDER, F. KWOK, AND B. MANDAL, *Dirichlet-Neumann and Neumann-Neumann waveform relaxation algorithms for parabolic problems*, ETNA, 45 (2016), pp. 424–456.
- [13] M. GANDER AND A. STUART, *Space-time continuous analysis of waveform relaxation for the heat equation*, SIAM J. Sci. Comput., 19 (1998), pp. 2014–2031.
- [14] E. GILADI AND H. KELLER, *Space-time domain decomposition for parabolic problems*, Numer. Math., 93 (2002), pp. 279–313.
- [15] L. HALPERN, C. JAPHET, AND J. SZEFTTEL, *Optimized Schwarz waveform relaxation and discontinuous Galerkin time stepping for heterogeneous problems*, SIAM J. Numer. Anal., 50 (2012), pp. 2588–2611.
- [16] U. HECK, U. FRITSCHING, AND K. BAUCKHAGE, *Fluid flow and heat transfer in gas jet quenching of a cylinder*, Internat. J. Numer. Methods Heat Fluid Flow, 11 (2001), pp. 36–49.
- [17] M. HINDERKS AND R. RADESPIEL, *Investigation of hypersonic gap flow of a reentry nose cap with consideration of fluid structure interaction*, in Proceedings of AIAA, 2006, pp. 2708–3741.
- [18] T. HOANG, *Space-Time Domain Decomposition Methods for Mixed Formulations of Flow and Transport Problems in Porous Media*, Ph.D. thesis, Université Pierre et Marie Curie, Paris, France, 2013.
- [19] T. HOANG, C. JAPHET, M. KERN, AND J. ROBERTS, *Space-time domain decomposition for reduced fracture models in mixed formulation*, SIAM J. Numer. Anal., 54 (2016), pp. 288–316.
- [20] D. KOWOLLIK, P. HORST, AND M. HAUPT, *Fluid-structure interaction analysis applied to thermal barrier coated cooled rocket thrust chambers with subsequent local investigation of delamination phenomena*, Progr. Propul. Phys., 4 (2013), pp. 617–636.
- [21] D. KOWOLLIK, V. TINI, S. REESE, AND M. HAUPT, *3D fluid-structure interaction analysis of a typical liquid rocket engine cycle based on a novel viscoplastic damage model*, Internat. J. Numer. Methods Engrg., 94 (2013), pp. 1165–1190.
- [22] F. KWOK, *Neumann-Neumann waveform relaxation for the time-dependent heat equation*, in Domain Decomposition in Science and Engineering XXI, J. Erhel, M.J. Gander, L. Halpern, G. Pichot, T. Sassi and O.B. Widlund, eds., Lect. Notes Comput. Sci. Eng. 98, Springer, Berlin, 2014.
- [23] E. LELARASMEE, A. RUEHLI, AND A. SANGIOVANNI-VINCENTELLI, *The waveform relaxation method for time-domain analysis of large scale integrated circuits*, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., 1 (1982), pp. 131–145.
- [24] F. LEMARIÉ, L. DEBREU, AND E. BLAYO, *Toward an optimized global-in-time Schwarz algorithm for diffusion equations with discontinuous and spatially variable coefficients. Part 1: The constant coefficients case*, ETNA, 40 (2013), pp. 148–169.
- [25] B. MANDAL, *A time-dependent Dirichlet-Neumann method for the heat equation*, in Domain Decomposition in Science and Engineering XXI, J. Erhel, M.J. Gander, L. Halpern, G. Pichot, T. Sassi and O.B. Widlund, eds., Lect. Notes Comput. Sci. Eng. 98, Springer, Berlin, 2014.
- [26] B. MANDAL, *Convergence Analysis of Substructuring Waveform Relaxation Methods for Space-Time Problems and Their Application to Optimal Control Problems*, Ph.D. thesis, University of Geneva, Geneva, Switzerland, 2014.
- [27] M. MEHL, B. UEKERMANN, H. BIJL, D. BLOM, B. GATZHAMMER, AND A. VAN ZUIJLEN, *Parallel coupling numerics for partitioned fluid-structure interaction simulations*, Comput. Math. Appl., 71 (2016), pp. 869–891.
- [28] R. MEHTA, *Numerical computation of heat transfer on reentry capsules at Mach 5*, in Proceedings of AIAA, 2005.
- [29] A. MONGE AND P. BIRKEN, *On the convergence rate of the Dirichlet-Neumann iteration for unsteady thermal fluid-structure interaction*, Comput. Mech., 62 (2018), pp. 525–541.
- [30] B. ONG AND B. MANDAL, *Pipeline implementations of Neumann-Neumann and Dirichlet-Neumann waveform relaxation methods*, Numer. Algorithms, 78 (2018), pp. 1–20.
- [31] A. QUARTERONI AND A. VALLI, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science, Oxford, 1999.
- [32] P. STRATTON, I. SHEDLETSKY, AND M. LEE, *Gas Quenching with Helium*, Solid State Phenom., 118 (2006), pp. 221–226.
- [33] A. TOSELLI AND O. WIDLUND, *Domain Decomposition Methods—Algorithms and Theory*, Springer, Berlin, 2004.
- [34] E. VAN BRUMMELEN, *Partitioned iterative solution methods for fluid-structure interaction*, Internat. J. Numer. Methods Fluids, 65 (2011), pp. 3–27.
- [35] J. K. WHITE, *Multirate Integration Properties of Waveform Relaxation with Application to Circuit Simulation and Parallel Computation*, Ph.D. thesis, EECS Department, University of California, Berkeley, 1985.