A SUPERLINEAR CONVERGENCE ESTIMATE FOR THE PARAREAL SCHWARZ WAVEFORM RELAXATION ALGORITHM *

MARTIN J. GANDER†, YAO-LIN JIANG‡, AND BO SONG§

Abstract. The Parareal Schwarz Waveform Relaxation algorithm is a new space-time parallel algorithm for the solution of evolution partial differential equations. It is based on a decomposition of the entire domain both in space and in time into smaller space-time subdomains, and then computes by an iteration in parallel on all these small subdomains a better and better approximation of the overall solution. The initial conditions in the subdomains are updated using a parareal mechanism, while the boundary conditions are updated using Schwarz waveform relaxation techniques. A first precursor of this algorithm was presented fifteen years ago, and while the method works well in practice, the convergence of the algorithm is not yet understood, and to analyze it is technically difficult. We present in this paper for the first time an accurate superlinear convergence estimate when the algorithm is applied to the heat equation. We illustrate our analysis with numerical experiments including cases not covered by the analysis, which opens up many further research directions.

Key words. Schwarz waveform relaxation, parareal algorithm, Parareal Schwarz Waveform Relaxation, domain decomposition, space-time parallel methods, heat equation

AMS subject classifications. 65M55, 65M22, 65F15

1. Introduction.

Schwarz waveform relaxation algorithms are parallel algorithms for time-dependent partial differential equations (PDEs) based on a spatial domain decomposition. The spatial domain is decomposed into overlapping or non-overlapping subdomains, and an iteration in space-time, based on space-time subdomain solutions, is used to obtain better and better approximations of the underlying global space-time solution. During the iteration, neighboring subdomains are communicating through transmission conditions. The name Schwarz comes from the fact that overlap can be used, like in the classical Schwarz method for elliptic problems [62], and the name waveform relaxation indicates that the iterates are functions in time, like in the classical waveform relaxation method developed for very large scale integration of circuits [48]. Waveform relaxation methods have been analyzed for many different kinds of problems, such as ordinary differential equations (ODEs) [4, 30, 16], differential algebraic equations (DAEs) [46, 41], partial differential equations (PDEs) [50], time-periodic problems [44, 43, 68] and fractional differential equations [45], for further details, see [42]. In the Schwarz waveform relaxation algorithm, the transmission conditions play an important role, and while classical Dirichlet conditions lead to robust, superlinear convergence for diffusive problems [13, 35, 34, 29], optimized transmission conditions based on [21] of Robin or Ventcell type as in the steady case [40] lead to much faster, so called optimized Schwarz waveform relaxation methods, see [20, 3] for diffusive problems, and [22, 19, 38] for wave propagation. These are also

*Submitted to the editors XXX.

Funding: This work was supported by the Natural Science Foundation of China (NSFC) under grant 11871393, 11801449, and the International Science and Technology Cooperation Program of Shaanxi Key Research & Development Plan under grant S2019-YF-GHZD-0003.

†Section of Mathematics, University of Geneva, 1211 Geneva 4, Switzerland (martin.gander@unige.ch).

‡Corresponding author. School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (yljiang@xjtu.edu.cn).

§School of Science, Northwestern Polytechnical University, Xi'an 710072, China (bosong@nwpu.edu.cn).

the same techniques underlying modern time harmonic wave propagation solvers, for an overview, see [33] and references therein.

The parareal algorithm is a time-parallel method that was proposed by Lions, Maday, and Turinici in the context of virtual control to solve evolution problems in parallel, see [49]. In this algorithm, initial value problems are solved on subintervals in time, and through iterations the initial values on each subinterval are corrected to converge to the correct values of the overall solution. The parareal algorithm uses two approximate propagators which are called the fine propagator and the coarse propagator. The fine propagator determines the final precision, while the coarse propagator influences the parallel speedup. In most theoretical analyses of the parareal algorithm, the fine propagator was for simplicity chosen to be the exact solver, and the coarse propagator was a common one-step method such as the Backward Euler method. Precise convergence estimates for the parareal algorithm applied to linear ordinary and partial differential equations can be found in [32]; for the non-linear case, see [14]. The parareal algorithm has also been used in many application areas, like linear and nonlinear parabolic problems [65, 66, 50], molecular dynamics [1], stochastic ordinary differential equations (ODEs) [2, 8], Navier-Stokes equations [67, 10], quantum control problems [56, 57, 55], time periodic problems [25], fractional diffusion equations [72], and low-frequency problems in electrical engineering [61]; for a parallel coarse correction variant, see [70]. Several other new variants of the parareal algorithm have been presented, which use an iterative method, the spectral deferred correction method, for solving ODEs for the coarse and fine propagators rather than traditional methods, see [60, 59], which led to the Parallel Full Approximation Scheme in Space-Time (PFASST) [7]. The parareal algorithm has also been combined with waveform relaxation methods [52, 51, 63, 64]. More recently, new time parallel strategies have also been developed, such as the PARAEXP algorithm [17, 37] and a new full space-time multigrid method [28] with excellent strong and weak scalability properties; for earlier time multigrid approaches, see [53, 68, 69]. There is also MGRIT [11, 9] with a convergence analysis in [27], showing that MGRIT is in fact a multilevel variant of an overlapping parareal algorithm. A further direct approach based on the diagonalization of the time stepping matrix was introduced in [54]. These techniques have been applied to the heat equation [23], the wave equation [12] and the time-periodic fractional diffusion equation [71]. For a complete overview of the historical development of time parallel methods over five decades, see [15].

A first approach to combine Schwarz waveform relaxation and the parareal algorithm for PDEs can be found in [58], where the authors propose to use waveform relaxation solvers for the coarse and fine propagators in the parareal algorithm, see also the PhD thesis [36]. This algorithm can be understood in the sense that if the waveform relaxation algorithms compute the fine and coarse propagators with enough accuracy, the parareal convergence theory applies. In practice it is however more interesting not to iterate to convergence, but just to use one iteration, directly embedded in the parareal updating process, which leads to the so called Parareal Schwarz Waveform Relaxation (PSWR) algorithm that was first proposed in [24]. The implementation of PSWR is not very difficult, but to prove convergence and obtain a convergence estimate is, and we present here for the first time a superlinear convergence result based on detailed kernel estimates, when the method is applied to the one dimensional heat equation.

Our paper is organized as follows. In Section 2, we present the PSWR algorithm for a general parabolic problem. In Section 3, we prove our technical, superlinear convergence estimate for the PSWR algorithm with Dirichlet transmission conditions

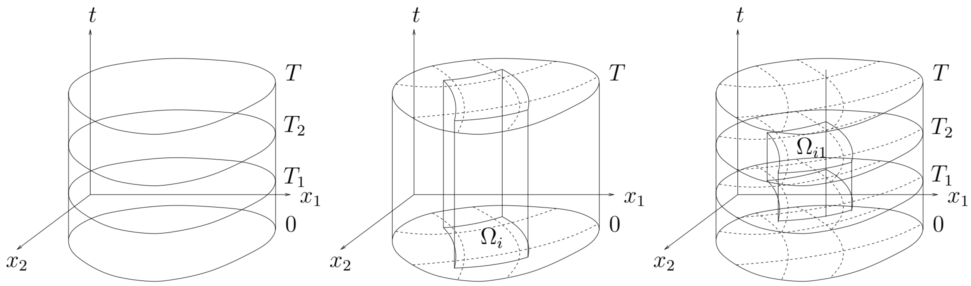

FIG. 1. *Time domain decomposition for parareal (left), space decomposition for Schwarz waveform relaxation showing one overlapping space domain global in time (middle) and space-time decomposition for PSWR showing one smaller space-time domain (right).*

when applied to the heat equation in one spatial dimension with a two subdomain decomposition in space and an arbitrary decomposition in time. We illustrate our analysis with numerical experiments in Section 4, and also test cases not covered by our analysis, like the many spatial subdomain case and optimized transmission conditions. We finally present our conclusions and several open research directions in Section 5.

2. Construction of the PSWR algorithm. We derive the PSWR algorithm for the time dependent parabolic partial differential equation

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \mathcal{L}u + f && \text{in } \Omega\times(0,T),\ \Omega\subset\mathbb{R}^d,\ d=1,2,3,\\
u(x,0) &= u_0(x) && \text{in } \Omega,\\
u &= g && \text{on } \partial\Omega\times(0,T),
\end{aligned}
\tag{2.1}
$$

where $\mathcal{L}$ is a second order elliptic operator, e.g., the Laplace operator. We next describe the parareal algorithm and the Schwarz waveform relaxation algorithm for problem (2.1), before introducing PSWR.

2.1. The parareal algorithm. The parareal algorithm is for the parallelization of the solution of problems like (2.1) in the time direction: by decomposing the time interval $(0,T)$ into N time subintervals (T_n, T_{n+1}) with $0 = T_0 < T_1 < \cdots < T_N = T$, as shown in Figure 1 on the left for the case of $d = 2$ spatial dimensions, we obtain a series of subproblems in the time subintervals (T_n, T_{n+1}) with unknown initial values $u(x, T_n)$, which we denote by $U_n(x)$. In order to obtain the solution of the original problem (2.1), the $\{U_n\}$ have to solve the system of equations

$$
U_0 = u_0, \quad U_{n+1} = S(T_{n+1}, T_n, U_n, f, g), \quad n = 0, 1, \ldots, N-1,
\tag{2.2}
$$

where $S(T_{n+1}, T_n, U_n, f, g)$ denotes the exact solution operator on the time subinterval (T_n, T_{n+1}), i.e. $S(T_{n+1}, T_n, U_n, f, g)$ is the exact solution at T_{n+1} of the evolution problem (2.1) on the time subinterval (T_n, T_{n+1}) with a given initial condition U_n, right hand side source term f and boundary conditions g,

$$
\frac{du_n}{dt} = \mathcal{L}u_n + f \text{ in } \Omega\times(T_n,T_{n+1}), u_n(x,T_n) = U_n(x) \text{ in } \Omega, u_n = g \text{ on } \partial\Omega\times(T_n,T_{n+1}).
\tag{2.3}
$$

The parareal algorithm solves the system of equations (2.2) by iteration using a so called coarse propagator $G(T_{n+1}, T_n, U_n, f, g)$ which provides a rough approximation in time of the solution $u_n(x, T_{n+1})$ of (2.3) with a given initial condition

$u_n(x, T_n) = U_n(x)$, right hand side source term f and boundary conditions g, and a fine propagator $F(T_{n+1}, T_n, U_n, f, g)$, which gives a more accurate approximation in time of the same solution. Starting with a first approximation U_n^0 at the time points $T_0, T_1, T_2, \dots, T_{N-1}$, the parareal algorithm performs for $k = 0, 1, 2, \dots$ the correction iteration

$$U_{n+1}^{k+1} = F(T_{n+1}, T_n, U_n^k, f, g) + G(T_{n+1}, T_n, U_n^{k+1}, f, g) - G(T_{n+1}, T_n, U_n^k, f, g). \tag{2.4}$$

It was shown in [32] that (2.4) is a multiple shooting method in time with an approximate Jacobian in the Newton step, and accurate convergence estimates were derived for the heat and wave equation in [32], see also [18] for similar convergence estimates for the case of nonlinear problems.

2.2. Introduction to Schwarz waveform relaxation. In contrast to the parareal algorithm, the Schwarz waveform relaxation algorithm for the model problem (2.1) is based on a spatial decomposition only, in the most general case into overlapping subdomains $\Omega = \cup_{i=1}^I \Omega_i$, see the middle plot in Figure 1. The Schwarz waveform relaxation algorithm solves iteratively for $k = 0, 1, 2, \dots$ the space-time subdomain problems

$$\begin{array}{rcll} \dfrac{\partial u_i^{k+1}}{\partial t} & = & \mathcal{L} u_i^{k+1} + f, & \text{in } \Omega_i \times (0, T), \\ u_i^{k+1}(x, 0) & = & u_0, & \text{in } \Omega_i, \\ \mathcal{B}_i u_i^{k+1} & = & \mathcal{B}_i \bar{u}^k, & \text{on } \partial\Omega_i \times (0, T). \end{array}$$

Here $\bar{u}^k$ denotes a composed approximate solution from the previous subdomain solutions u_i^k using for example a partition of unity, and an initial guess $\bar{u}^0$ is needed to start the iteration. The operators $\mathcal{B}_i$ are transmission operators, and we did not write the Dirichlet boundary conditions at the outer boundaries for simplicity. If the transmission operators $\mathcal{B}_i$ are the identity, we obtain the classical Schwarz waveform relaxation algorithm, whose convergence was studied for general decompositions in higher space dimensions in [34]; if they represent Robin or higher order transmission conditions, we obtain an optimized Schwarz waveform relaxation algorithm, if the parameters in the transmission conditions are chosen to optimize the convergence factor of the algorithm, see [20, 3] and references therein. A convergence analysis for optimized Schwarz waveform relaxation methods for general decompositions in higher spatial dimensions is however still an open problem, like for optimized Schwarz methods in the steady case.

2.3. Construction of PSWR. We decompose the space-time domain $\Omega \times (0, T)$ into space-time subdomains $\Omega_{i,n} := \Omega_i \times (T_n, T_{n+1})$, $i = 1, 2, \cdots, I$, $n = 0, 1, \cdots, N-1$, as shown in Figure 1 on the right. Like in the parareal algorithm, we introduce a fine subdomain solver $F_{i,n}(U_{i,n}^k, \mathcal{B}_i \bar{u}_n^k)$ and a coarse subdomain solver $G_{i,n}(U_{i,n}^k, \mathcal{B}_i \bar{u}_n^k)$, where we do not explicitly state the dependence of these solvers on the time interval and the right hand side f and original Dirichlet boundary condition g to not increase the complexity of the notation further. There is also a further important notational difference with parareal: here the fine solver F returns the entire solution in space-time, not just at the final time, since this solution is also needed in the transmission conditions of the algorithm. Then for any initial guess of the initial values $U_{i,n}^0$ and the interface values $\mathcal{B}_i \bar{u}_n^0$, the PSWR algorithm for the parabolic problem (2.1) computes for iteration index $k = 0, 1, 2, \dots$ and all spatial and time indices $i = 1, 2, \dots, I$,

$n = 0, 1, \dots, N-1$

$$\begin{aligned} u_{i,n}^{k+1} &= F_{i,n}(U_{i,n}^k, \mathcal{B}_i \bar{u}_n^k), \\ U_{i,n+1}^{k+1} &= u_{i,n}^{k+1}(\cdot, T_{n+1}) + G_{i,n}(U_{i,n}^{k+1}, \mathcal{B}_i \bar{u}_n^{k+1}) - G_{i,n}(U_{i,n}^k, \mathcal{B}_i \bar{u}_n^k), \end{aligned} \tag{2.5}$$

where $\bar{u}_n^k$ is again a composed approximate solution from the subdomain solutions $u_{i,n}^k$ using for example a partition of unity, and an initial guess $\bar{u}_n^0$ and $U_{i,k}^0$ is needed to start the iteration[1]. Note that the first step in (2.5), which is the expensive step involving the fine propagator $F_{i,n}$, can be performed in parallel over all space-time subdomains $\Omega_{i,n}$, since both the initial and boundary data are available from the previous iteration. The cheap second step in (2.5) involving only the coarse propagator $G_{i,n}$ to compute a new initial condition for all space-time subdomains is still in parallel in space, but now sequential in time, like in the parareal algorithm.

It is worthwhile to look at the PSWR (2.5) again before continuing: it is an iteration from initial and boundary data on space-time subdomains to initial and boundary data on space-time subdomains, i.e. it maps traces in space and traces in time to new traces in space and traces in time. There is also a particular choice for the new coarse solver in the middle of the second step of (2.5): it uses the most recent fine approximation for its boundary conditions. This is natural since this can be reused in the second iteration for the old coarse solver on the right in the second line of (2.5), like in the classical parareal algorithm, but using the old iterates would be possible as well. This would however not lead to more parallelism, because of the new initial condition that is needed for the parareal update.

3. Convergence analysis of PSWR. To capture the true convergence behavior of the PSWR algorithm by analysis is technically difficult, and we thus consider from now on the heat equation on an unbounded domain in one spatial dimension,

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} + f(x,t), \quad \text{in } \Omega \times (0,T),\ \Omega := \mathbb{R}, \tag{3.1}$$

with the initial condition $u(x,0) = u_0(x)$, $x \in \Omega$, and only a decomposition into two overlapping subdomains, $\Omega_1 = (-\infty, L)$ and $\Omega_2 = (0, +\infty)$, $L > 0$, and we assume that the algorithm uses Dirichlet transmission conditions, i.e. $\mathcal{B}_i = \mathcal{I}$, the identity in (2.5). We will test the more general case extensively in the numerical experiments in Section 4. We decompose the time interval $(0,T)$ into N equal time subintervals $0 = T_0 \le \cdots \le T_n = n\Delta T \le \cdots \le T_N = T$, $\Delta T = \frac{T}{N}$, and thus our space-time subdomains are $\Omega_{i,n} = \Omega_i \times (T_n, T_{n+1})$, $i = 1, 2$, $n = 0, \dots, N-1$. We also assume that the fine propagator $F_{i,n}$ is exact, like it is often done in the convergence analysis of the parareal algorithm, and that the coarse propagator $G_{i,n}$ is exact in space, and uses Backward Euler in time.

To study the convergence of PSWR, we introduce the error in the space-time subdomains

$$e_{i,n}^k(x,t) := u_{i,n}^k(x,t) - u(x,t) \quad \text{in } \Omega_{i,n}, \tag{3.2}$$

and also the error in the initial values

$$E_{i,n}^k(x) := U_{i,n}^k(x) - u(x, T_n) \quad x \in \Omega_i. \tag{3.3}$$

[1]The latter can for example be computed using the coarse propagator once the former is chosen.

By linearity, it suffices to analyze convergence to the zero solution. Using the definitions of the propagators $F_{i,n}$ and $G_{i,n}$ and their linearity, we get for the error on the first spatial subdomain

$$\begin{aligned} e_{1,n}^{k+1}(x,t) &= F_{1,n}(E_{1,n}^k, e_{2,n}^k(L,\cdot)),\\ E_{1,n+1}^{k+1}(x) &= e_{1,n}^{k+1}(x,T_{n+1}) + G_{1,n}(E_{1,n}^{k+1}, e_{2,n}^{k+1}(L,\cdot)) - G_{1,n}(E_{1,n}^k, e_{2,n}^k(L,\cdot)), \end{aligned} \tag{3.4}$$

and similarly on the second spatial subdomain

$$\begin{aligned} e_{2,n}^{k+1}(x,t) &= F_{2,n}(E_{2,n}^k, e_{1,n}^k(0,\cdot)),\\ E_{2,n+1}^{k+1}(x) &= e_{2,n}^{k+1}(x,T_{n+1}) + G_{2,n}(E_{2,n}^{k+1}, e_{1,n}^{k+1}(0,\cdot)) - G_{2,n}(E_{2,n}^k, e_{1,n}^k(0,\cdot)), \end{aligned} \tag{3.5}$$

where we do not need to use a partition of unity to compose a general approximate solution, since each subdomain must take data directly from its only neighbor, which will simplify the analysis. To study the contraction properties of this iteration, we need estimates of the continuous solution operator represented by the fine propagator F, and of the time discrete solution operator represented by the coarse propagator G. We thus start by computing representation formulas for these solution operators.

3.1. Representation formula for the fine propagator F.

The first step $e_{1,n}^{k+1}(x,t) = F_{1,n}(E_{1,n}^k, e_{2,n}^k(L,\cdot))$ and $e_{2,n}^{k+1}(x,t) = F_{2,n}(E_{2,n}^k, e_{1,n}^k(0,\cdot))$ in the error iteration (3.4), (3.5) requires the solution of homogeneous problems in $\Omega_{i,n}$, $i,= 1,2$, namely

$$\begin{aligned} \frac{\partial e_{1,n}^{k+1}(x,t)}{\partial t} &= \frac{\partial^2 e_{1,n}^{k+1}(x,t)}{\partial x^2}, && (x,t)\in\Omega_{1,n},\\ e_{1,n}^{k+1}(L,t) &= e_{2,n}^k(L,t), && t\in(T_n,T_{n+1}),\\ e_{1,n}^{k+1}(x,T_n) &= E_{1,n}^k(x), && x\in(-\infty,L), \end{aligned} \tag{3.6}$$

and

$$\begin{aligned} \frac{\partial e_{2,n}^{k+1}(x,t)}{\partial t} &= \frac{\partial^2 e_{2,n}^{k+1}(x,t)}{\partial x^2}, && (x,t)\in\Omega_{2,n},\\ e_{2,n}^{k+1}(0,t) &= e_{1,n}^k(0,t), && t\in(T_n,T_{n+1}),\\ e_{2,n}^{k+1}(x,T_n) &= E_{2,n}^k(x), && x\in(0,+\infty). \end{aligned} \tag{3.7}$$

Therefore in Ω_1, the fine propagator has a closed form representation formula giving the solution of problem (3.6) (see [5]),

$$\begin{aligned} e_{1,n}^{k+1}(x,t) = &\int_{-\infty}^0 \left(K(x-L-\xi,t-T_n) - K(x-L+\xi,t-T_n)\right) E_{1,n}^k(\xi)d\xi\\ &+ 2\int_{T_n}^t \frac{\partial K}{\partial x}(x-L,t-T_n-\tau)e_{2,n}^k(L,\tau)d\tau, \end{aligned} \tag{3.8}$$

where the heat kernel is given by

$$K(x,t) = \frac{1}{\sqrt{4\pi t}}e^{-x^2/4t}. \tag{3.9}$$

We now define for the initial value part the linear solution operator $\mathcal{A}_{1,n}$,

$$\left(\mathcal{A}_{1,n}E\right)(x,t) := \int_{-\infty}^0 \left(K(x-L-\xi,t-T_n) - K(x-L+\xi,t-T_n)\right) E(\xi)d\xi, \tag{3.10}$$

and for the boundary value part the linear solution operator $\mathcal{B}_{1,n}$,

$$(\mathcal{B}_{1,n}e)(x,t) := 2\int_{T_n}^{t} \frac{\partial K}{\partial x}(x-L, t-T_n-\tau)e(\tau)d\tau. \tag{3.11}$$

Then (3.8) can be written in the form

$$e_{1,n}^{k+1}(x,t) = (\mathcal{A}_{1,n}E_{1,n}^{k})(x,t) + (\mathcal{B}_{1,n}e_{2,n}^{k}(L,\cdot))(x,t). \tag{3.12}$$

Similarly, we obtain on the second subdomain Ω_2 using the representation formula for the solution of (3.7)

$$e_{2,n}^{k+1}(x,t) = (\mathcal{A}_{2,n}E_{2,n}^{k})(x,t) + (\mathcal{B}_{2,n}e_{1,n}^{k}(0,\cdot))(x,t) \tag{3.13}$$

with the linear solution operators

$$\begin{aligned}(\mathcal{A}_{2,n}E)(x,t) &:= \int_{0}^{\infty} \left(K(x-\xi, t-T_n) - K(x+\xi, t-T_n)\right)E(\xi)d\xi,\\ (\mathcal{B}_{2,n}e)(x,t) &:= -2\int_{T_n}^{t} \frac{\partial K}{\partial x}(x, t-T_n-\tau)e(\tau)d\tau.\end{aligned} \tag{3.14}$$

3.2. Representation formula for the coarse propagator G. Using the Backward Euler time stepping scheme for the coarse propagator G, and denoting by $e_{1,G}(x) := G(E_{1,n}^{k}(x), e_{2,n}^{k}(L, T_{n+1}))$ the term that appears in the error recursion (3.4), we see that $e_{1,G}$ satisfies the equation

$$\begin{aligned}\frac{e_{1,G}(x) - E_{1,n}^{k}(x)}{\Delta T} - \frac{\partial^2 e_{1,G}(x)}{\partial x^2} &= 0, \quad x \in \Omega_1,\\ e_{1,G}(L) &= e_{2,n}^{k}(L, T_{n+1}).\end{aligned}$$

This problem has the closed form solution (see the Appendix)

$$e_{1,G}(x) = e_{2,n}^{k}(L, T_{n+1})e^{\frac{x-L}{\sqrt{\Delta T}}} + (\mathcal{C}_1 E_{1,n}^{k})(x), \tag{3.15}$$

with the linear solution operator $\mathcal{C}_1$ defined by

$$\begin{aligned}(\mathcal{C}_1 E_{1,n}^{k})(x) := -\frac{1}{2\sqrt{\Delta T}}\Bigg(&\int_{-\infty}^{L} e^{\frac{x+\xi-2L}{\sqrt{\Delta T}}} E_{1,n}^{k}(\xi)d\xi - \int_{x}^{L} e^{\frac{x-\xi}{\sqrt{\Delta T}}} E_{1,n}^{k}(\xi)d\xi\\ &- \int_{-\infty}^{x} e^{\frac{-x+\xi}{\sqrt{\Delta T}}} E_{1,n}^{k}(\xi)d\xi\Bigg).\end{aligned}$$

Similarly, denoting by $e_{2,G}(x) := G(E_{2,n}^{k}(x), e_{1,n}^{k}(0, T_{n+1}))$ on Ω_2 the term that appears in the error recursion (3.5), we see that $e_{2,G}$ satisfies the equation

$$\begin{aligned}\frac{e_{2,G}(x) - E_{2,n}^{k}}{\triangle T} - \frac{\partial^2 e_{2,G}(x)}{\partial x^2} &= 0, \quad x \in \Omega_2,\\ e_{2,G}(0) &= e_{1,n}^{k}(0, T_{n+1}),\end{aligned}$$

and we obtain for the solution

$$e_{2,G}(x) = e_{1,n}^{k}(0, T_{n+1})e^{\frac{x}{\sqrt{\Delta T}}} + (\mathcal{C}_2 E_{2,n}^{k})(x), \tag{3.16}$$

with the linear solution operator $\mathcal{C}_2$ defined by

$$(\mathcal{C}_2 E_{2,n}^k)(x) := -\frac{1}{2\sqrt{\Delta T}}\left(\int_0^{+\infty} e^{-\frac{x+\xi}{\sqrt{\Delta T}}} E_{2,n}^k(\xi)d\xi - \int_0^x e^{-\frac{x-\xi}{\sqrt{\Delta T}}} E_{2,n}^k(\xi)d\xi \right.$$
$$\left. - \int_x^{+\infty} e^{\frac{x-\xi}{\sqrt{\Delta T}}} E_{2,n}^k(\xi)d\xi\right).$$

3.3. Matrix Formulation of PSWR. We now rewrite the error recurrence formulation (3.4), (3.5) more explicitly using the representation formulas, and then collect the complete PSWR map from traces in space and time to traces in space and time into a matrix formulation, which is amenable to analysis. We start with Ω_1: the first equation in the the error recursion formula (3.4) can be expressed using the representation formula (3.12) for the fine propagator as

$$e_{1,n}^{k+1}(x,t) = F_{1,n}(E_{1,n}^k, e_{2,n}^k(L,\cdot)) = (\mathcal{A}_{1,n}E_{1,n}^k)(x,t) + (\mathcal{B}_{1,n}e_{2,n}^k(L,\cdot))(x,t). \tag{3.17}$$

For the second equation in (3.4), we have to evaluate (3.17) at $t = T_{n+1}$ and use the representation formula (3.15) for the coarse propagator twice, to obtain

$$\begin{aligned} E_{1,n+1}^{k+1}(x) &= e_{1,n}^{k+1}(x,T_{n+1}) + G_{1,n}(E_{1,n}^{k+1}, e_{2,n}^{k+1}(L,\cdot)) - G_{1,n}(E_{1,n}^k, e_{2,n}^k(L,\cdot)) \\ &= \left(\mathcal{A}_{1,n}E_{1,n}^k\right)(x,T_{n+1}) + \left(\mathcal{B}_{1,n}e_{2,n}^k(L,\cdot)\right)(x,T_{n+1}) \\ &\quad + e_{2,n}^{k+1}(L,T_{n+1})e^{\frac{x-L}{\sqrt{\Delta T}}} + (\mathcal{C}_1 E_{1,n}^{k+1})(x) \\ &\quad - e_{2,n}^k(L,T_{n+1})e^{\frac{x-L}{\sqrt{\Delta T}}} - (\mathcal{C}_1 E_{1,n}^k)(x). \end{aligned} \tag{3.18}$$

In (3.17), we still work with the volume function $e_{1,n}^{k+1}(x,t)$ which is only used in the iteration either traced at $t = T_{n+1}$, i.e. $e_{1,n}^{k+1}(x,T_{n+1})$, as in (3.18), or traced at $x = 0$, i.e. $e_{1,n}^{k+1}(0,t)$ by the second subdomain. We therefore introduce the following linear operators which include taking the trace:

$$\begin{aligned} &\mathcal{A}_{1,n,0}E_{1,n}^k := \left(\mathcal{A}_{1,n}E_{1,n}^k\right)(0,t), && \mathcal{B}_{1,n,0}e_{2,n}^k := \left(\mathcal{B}_{1,n}e_{2,n}^k(L,\cdot)\right)(0,t), \\ &\mathcal{A}_{1,n,\Delta T}E_{1,n}^k := \left(\mathcal{A}_{1,n}E_{1,n}^k\right)(x,T_{n+1}), && \mathcal{B}_{1,n,\Delta T}e_{2,n}^k := \left(\mathcal{B}_{1,n}e_{2,n}^k(L,\cdot)\right)(x,T_{n+1}), \\ &\mathcal{D}_{1,\Delta T}e_{2,n}^k := e_{2,n}^k(L,T_{n+1})e^{\frac{x-L}{\sqrt{\Delta T}}}, \end{aligned} \tag{3.19}$$

and then (3.17) and (3.18) become

$$\begin{aligned} e_{1,n}^{k+1}(0,t) &= (\mathcal{A}_{1,n,0}E_{1,n}^k)(t) + (\mathcal{B}_{1,n,0}e_{2,n}^k)(t), \\ E_{1,n+1}^{k+1}(x) &= (\mathcal{A}_{1,n,\Delta T}E_{1,n}^k)(x) + (\mathcal{B}_{1,n,\Delta T}e_{2,n}^k)(x) \\ &\quad + (\mathcal{D}_{1,\Delta T}e_{2,n}^{k+1})(x) + (\mathcal{C}_1 E_{1,n}^{k+1})(x) - (\mathcal{D}_{1,\Delta T}e_{2,n}^k)(x) - (\mathcal{C}_1 E_{1,n}^k)(x), \end{aligned} \tag{3.20}$$

and we see that the first line represents well a function in time obtained by tracing at $x = 0$ while the second line represents well a function in space. Similarly, we obtain on the second subdomain Ω_2

$$\begin{aligned} e_{2,n}^{k+1}(L,t) &= (\mathcal{A}_{2,n,L}E_{2,n}^k)(t) + (\mathcal{B}_{2,n,L}e_{1,n}^k)(t), \\ E_{2,n+1}^{k+1}(x) &= (\mathcal{A}_{2,n,\Delta T}E_{2,n}^k)(x) + (\mathcal{B}_{2,n,\Delta T}e_{1,n}^k)(x) \\ &\quad + (\mathcal{D}_{2,\Delta T}e_{1,n}^{k+1})(x) + (\mathcal{C}_2 E_{2,n}^{k+1})(x) - (\mathcal{D}_{2,\Delta T}e_{1,n}^k)(x) - (\mathcal{C}_2 E_{2,n}^k)(x), \end{aligned} \tag{3.21}$$

where

$$(3.22)\quad \begin{aligned} &\mathcal{A}_{2,n,L}E_{2,n}^k := \left(\mathcal{A}_{2,n}E_{2,n}^k\right)(L,t), &&\mathcal{B}_{2,n,L}e_{1,n}^k := \left(\mathcal{B}_{2,n}e_{1,n}^k(0,\cdot)\right)(L,t),\\ &\mathcal{A}_{2,n,\Delta T}E_{2,n}^k := \left(\mathcal{A}_2E_{2,n}^k\right)(x,T_{n+1}), &&\mathcal{B}_{2,n,\Delta T}e_{1,n}^k := \left(\mathcal{B}_2e_{1,n}^k(0,\cdot)\right)(x,T_{n+1}),\\ &\mathcal{D}_{2,\Delta T}e_{1,n}^k := e_{1,n}^k(0,T_{n+1})e^{-\frac{x}{\sqrt{\Delta T}}}, \end{aligned}$$

We now collect all the traces in space and time used in the algorithm in the vectors of functions

$$(3.23)\quad \begin{aligned} \boldsymbol{e}_1^{k+1}(0,\cdot) &:= [e_{1,0}^{k+1}(0,\cdot), e_{1,1}^{k+1}(0,\cdot),\dots,e_{1,N-1}^{k+1}(0,\cdot)]^{\mathrm{T}},\\ \boldsymbol{E}_1^{k+1}(x) &:= [E_{1,0}^{k+1}(x), E_{1,1}^{k+1}(x),\dots,E_{1,N-1}^{k+1}(x)]^{\mathrm{T}},\\ \boldsymbol{e}_2^{k+1}(L,\cdot) &:= [e_{2,0}^{k+1}(L,\cdot), e_{2,1}^{k+1}(L,\cdot),\dots,e_{2,N-1}^{k+1}(L,\cdot)]^{\mathrm{T}},\\ \boldsymbol{E}_2^{k+1}(x) &:= [E_{2,0}^{k+1}(x), E_{2,1}^{k+1}(x),\dots,E_{2,N-1}^{k+1}(x)]^{\mathrm{T}}, \end{aligned}$$

and define the matrices

$$\mathbf{I} := \begin{bmatrix} \mathcal{I} & 0 & 0 & \cdots & 0\\ 0 & \mathcal{I} & 0 & \cdots & 0\\ 0 & 0 & \mathcal{I} & & \vdots\\ \vdots & \vdots & \vdots & \ddots & 0\\ 0 & 0 & 0 & 0 & \mathcal{I} \end{bmatrix},\quad \mathbf{I}_{-1} := \begin{bmatrix} 0 & 0 & 0 & \cdots & 0\\ \mathcal{I} & 0 & 0 & \cdots & 0\\ 0 & \mathcal{I} & 0 & & \vdots\\ \vdots & \vdots & \vdots & \ddots & 0\\ 0 & 0 & 0 & \mathcal{I} & 0 \end{bmatrix},$$

where the symbol $\mathcal{I}$ denotes the identity operator. We can then write the recurrence relations for the error in (3.20) and (3.21) in matrix form,

$$(3.24)\quad \begin{bmatrix} \mathbf{I} & 0 & 0 & 0\\ 0 & \mathbf{I}-\mathcal{C}_1\mathbf{I}_{-1} & -\mathcal{D}_{1,\Delta T}\mathbf{I}_{-1} & 0\\ 0 & 0 & \mathbf{I} & 0\\ -\mathcal{D}_{2,\Delta T}\mathbf{I}_{-1} & 0 & 0 & \mathbf{I}-\mathcal{C}_2\mathbf{I}_{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_1^{k+1}(0,\cdot)\\ \boldsymbol{E}_1^{k+1}(x)\\ \boldsymbol{e}_2^{k+1}(L,\cdot)\\ \boldsymbol{E}_2^{k+1}(x) \end{bmatrix} =$$
$$\begin{bmatrix} 0 & \mathcal{P}_{1,0} & \mathcal{Q}_{1,0} & 0\\ 0 & \mathcal{P}_{1,\Delta T}\mathbf{I}_{-1}-\mathcal{C}_1\mathbf{I}_{-1} & \mathcal{Q}_{1,\Delta T}\mathbf{I}_{-1}-\mathcal{D}_{2,\Delta T}\mathbf{I}_{-1} & 0\\ \mathcal{Q}_{2,L} & 0 & 0 & \mathcal{P}_{2,L}\\ \mathcal{Q}_{2,\Delta T}\mathbf{I}_{-1}-\mathcal{D}_{2,\Delta T}\mathbf{I}_{-1} & 0 & 0 & \mathcal{P}_{2,\Delta T}\mathbf{I}_{-1}-\mathcal{C}_2\mathbf{I}_{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{e}_1^{k}(0,\cdot)\\ \boldsymbol{E}_1^{k}(x)\\ \boldsymbol{e}_2^{k}(L,\cdot)\\ \boldsymbol{E}_2^{k}(x) \end{bmatrix},$$

where we also introduced the diagonal matrices of operators

$$(3.25)\quad \begin{aligned} \mathcal{P}_{1,0} &= \mathrm{diag}(\mathcal{A}_{1,0,0},\dots,\mathcal{A}_{1,N-1,0}), &\mathcal{P}_{1,\Delta T} &= \mathrm{diag}(\mathcal{A}_{1,0,\Delta T},\dots,\mathcal{A}_{1,N-1,\Delta T}),\\ \mathcal{P}_{2,L} &= \mathrm{diag}(\mathcal{A}_{2,0,L},\dots,\mathcal{A}_{2,N-1,L}), &\mathcal{P}_{2,\Delta T} &= \mathrm{diag}(\mathcal{A}_{2,0,\Delta T},\dots,\mathcal{A}_{2,N-1,\Delta T}),\\ \mathcal{Q}_{1,0} &= \mathrm{diag}(\mathcal{B}_{1,0,0},\dots,\mathcal{B}_{1,N-1,0}), &\mathcal{Q}_{1,\Delta T} &= \mathrm{diag}(\mathcal{B}_{1,0,\Delta T},\dots,\mathcal{B}_{1,N-1,\Delta T}),\\ \mathcal{Q}_{2,L} &= \mathrm{diag}(\mathcal{B}_{2,0,L},\dots,\mathcal{B}_{2,N-1,L}), &\mathcal{Q}_{2,\Delta T} &= \mathrm{diag}(\mathcal{B}_{2,0,\Delta T},\dots,\mathcal{B}_{2,N-1,\Delta T}). \end{aligned}$$

In order to understand the convergence behavior of the PSWR algorithm, we therefore have to understand the matrix iteration (3.24) where the entries of the matrices are continuous linear operators.

3.4. Tools from Linear Algebra. The analysis of the matrix iteration (3.24) is based on the following three Lemmas from linear algebra:

LEMMA 3.1. *If in the two by two block matrix*

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{3.26}$$

the diagonal submatrices M_{11} and M_{22} are lower triangular, and the off diagonal submatrices M_{12} and M_{21} are strictly lower triangular, and M_{22} is nonsingular, then

$$\det(M) = \det(M_{11}) \det(M_{22}).$$

Proof. Since M_{22} is non-singular, we can write the block matrix M in the factored form

$$M = \begin{bmatrix} I & M_{12}M_{22}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} M_{11} - M_{12}M_{22}^{-1}M_{21} & 0 \\ 0 & M_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ M_{22}^{-1}M_{21} & I \end{bmatrix},$$

and therefore obtain for its determinant the formula

$$\det(M) = \det(M_{11} - M_{12}M_{22}^{-1}M_{21}) \det(M_{22}). \tag{3.27}$$

Now by assumption, the off diagonal matrices are strictly lower triangular, and M_{22} is lower triangular, which implies that $M_{12}M_{22}^{-1}M_{21}$ is a strictly lower triangular matrix, and hence

$$\det(M_{11} - M_{12}M_{22}^{-1}M_{21}) = \det(M_{11}),$$

which concludes the proof of the Lemma. ∎

LEMMA 3.2 (see [39, page 18]). *If the inverse of the block matrix M in (3.26) is nonsingular, then*

$$M^{-1} = \begin{bmatrix} [M_{11} - M_{12}M_{22}^{-1}M_{21}]^{-1} & M_{11}^{-1}M_{12}[M_{21}M_{11}^{-1}M_{12} - M_{22}]^{-1} \\ [M_{21}M_{11}^{-1}M_{12} - M_{22}]^{-1}M_{21}M_{11}^{-1} & [M_{22} - M_{21}M_{11}^{-1}M_{12}]^{-1} \end{bmatrix},$$

assuming that all the relevant inverses exist.

LEMMA 3.3. *For a matrix A with the block structure*

$$A = \begin{bmatrix} B_1 + \Lambda_1 I & B_2 & B_3 & B_4 + \Lambda_2 I \\ B_5 & B_6 & B_7 & B_8 \\ B_9 & B_{10} + \Lambda_3 I & B_{11} + \Lambda_4 I & B_{12} \\ B_{13} & B_{14} & B_{15} & B_{16} \end{bmatrix},$$

where the submatrices B_i ($i = 1, \ldots, 16$) are all strictly lower triangular, and the Λ_i ($i = 1, \ldots, 4$) are scalar values, the spectral radius of A is given by

$$\rho(A) = \max\{|\Lambda_1|, |\Lambda_4|\}.$$

Proof. As in the proof of Lemma 3.1, we use the same block factorization to rewrite the determinant in the form (3.27)

(3.28)
$$\det(A-\lambda I) = \det\left(\begin{bmatrix} B_1+(\Lambda_1-\lambda)I & B_2 & B_3 & B_4+\Lambda_2 I \\ B_5 & B_6-\lambda I & B_7 & B_8 \\ B_9 & B_{10}+\Lambda_3 I & B_{11}+(\Lambda_4-\lambda)I & B_{12} \\ B_{13} & B_{14} & B_{15} & B_{16}-\lambda I \end{bmatrix}\right)$$

$$= \det\left(\begin{bmatrix} B_1+(\Lambda_1-\lambda)I & B_2 \\ B_5 & B_6-\lambda I \end{bmatrix} - \begin{bmatrix} B_3 & B_4+\Lambda_2 I \\ B_7 & B_8 \end{bmatrix} \begin{bmatrix} B_{11}+(\Lambda_4-\lambda)I & B_{12} \\ B_{15} & B_{16}-\lambda I \end{bmatrix}^{-1} \cdot \begin{bmatrix} B_9 & B_{10}+\Lambda_3 I \\ B_{13} & B_{14} \end{bmatrix}\right) \times \det\left(\begin{bmatrix} B_{11}+(\Lambda_4-\lambda)I & B_{12} \\ B_{15} & B_{16}-\lambda I \end{bmatrix}\right).$$

Now for the inverse on the right in (3.28), we obtain using Lemma 3.2 that

$$\begin{bmatrix} B_{11}+(\Lambda_4-\lambda)I & B_{12} \\ B_{15} & B_{16}-\lambda I \end{bmatrix}^{-1} = \begin{bmatrix} C_{11} & C_{12} \\ C_{15} & C_{16} \end{bmatrix},$$

with the block entries in the inverse given by

$$\begin{aligned} C_{11} &= [B_{11}+(\Lambda_4-\lambda)I - B_{12}(B_{16}-\lambda I)^{-1}B_{15}]^{-1}, \\ C_{12} &= (B_{11}+(\Lambda_4-\lambda)I)^{-1}B_{12}[B_{15}(B_{11}+(\Lambda_4-\lambda)I)^{-1}B_{12}-(B_{16}-\lambda I)]^{-1}, \\ C_{15} &= [B_{15}(B_{11}+(\Lambda_4-\lambda)I)^{-1}B_{12}-(B_{16}-\lambda I)]^{-1}B_{15}(B_{11}+(\Lambda_4-\lambda)I)^{-1}, \\ C_{16} &= [(B_{16}-\lambda I)-B_{12}(B_{11}+(\Lambda_4-\lambda)I)^{-1}B_{12}]^{-1}. \end{aligned}$$

We now study the structure of these block entries. For C_{11}, we first observe that $(B_{16}-\lambda I)^{-1}$ is lower triangular, since B_{16} is strictly lower triangular, and hence multiplying on the left and right by the strictly lower triangular matrices B_{12} and B_{15} the result will also be strictly lower triangular. The matrix C_{11} is thus the inverse of a strictly lower triangular matrix plus the diagonal matrix $(\Lambda_4-\lambda)I$, which implies that $C_{11} = B'_{11} + \frac{1}{\Lambda_4-\lambda}I$ for some strictly lower triangular matrix B'_{11}. Similarly, one can also analyze the structure of the other block entries of the inverse, and we obtain

$$\begin{bmatrix} B_{11}+(\Lambda_4-\lambda)I & B_{12} \\ B_{15} & B_{16}-\lambda I \end{bmatrix}^{-1} = \begin{bmatrix} B'_{11}+\frac{1}{\Lambda_4-\lambda}I & B'_{12} \\ B'_{15} & B'_{16}-\frac{1}{\lambda}I \end{bmatrix},$$

where all B'_i $(i = 11, 12, 15, 16)$ are strictly lower triangular matrices. We next study the product on the right in (3.28)

$$\begin{bmatrix} B_3 & B_4+\Lambda_2 I \\ B_7 & B_8 \end{bmatrix} \begin{bmatrix} B_{11}+(\Lambda_4-\lambda)I & B_{12} \\ B_{15} & B_{16}-\lambda I \end{bmatrix}^{-1} \begin{bmatrix} B_9 & B_{10}+\Lambda_3 I \\ B_{13} & B_{14} \end{bmatrix} = \begin{bmatrix} B_{17} & B_{18} \\ B_{19} & B_{20} \end{bmatrix},$$

and find again structurally that the B_i $(i = 17, \ldots, 20)$ are strictly lower triangular matrices. Using Lemma 3.1, the expression for the first determinant in the last line of (3.28) becomes

$$\begin{aligned}
&\det\left(\begin{bmatrix} B_1+(\Lambda_1-\lambda)I & B_2 \\ B_5 & B_6-\lambda I\end{bmatrix} - \begin{bmatrix} B_3 & B_4+\Lambda_2 I \\ B_7 & B_8\end{bmatrix}\right.\\
&\qquad \left.\cdot \begin{bmatrix} B_{11}+(\Lambda_4-\lambda)I & B_{12} \\ B_{15} & B_{16}-\lambda I\end{bmatrix}^{-1} \begin{bmatrix} B_9 & B_{10}+\Lambda_3 I \\ B_{13} & B_{14}\end{bmatrix}\right)\\
&= \det\left(\begin{bmatrix} B_1+(\Lambda_1-\lambda)I & B_2 \\ B_5 & B_6-\lambda I\end{bmatrix} - \begin{bmatrix} B_{17} & B_{18} \\ B_{19} & B_{20}\end{bmatrix}\right)\\
&= \det\left(\begin{bmatrix} \hat{B}_1+(\Lambda_1-\lambda)I & \hat{B}_2 \\ \hat{B}_5 & \hat{B}_6-\lambda I\end{bmatrix}\right)\\
&= \det(\hat{B}_1+(\Lambda_1-\lambda)I)\det(\hat{B}_6-\lambda I) = \lambda^n(\lambda-\Lambda_1)^n,
\end{aligned}$$

if the matrix subblocks are of size $n\times n$, and we used again Lemma 3.1, and here the $\hat{B}_i$ ($i = 1, 2, 5, 6$) are still strictly lower triangular matrices. For the second determinant in (3.28) we get directly using Lemma 3.1 that

$$\begin{aligned}
&\det\left(\begin{bmatrix} B_{11}+(\Lambda_4-\lambda)I & B_{12} \\ B_{15} & B_{16}-\lambda I\end{bmatrix}\right)\\
&\qquad = \det(B_{11}+(\Lambda_4-\lambda)I)\det(B_{16}-\lambda I) = \lambda^n(\lambda-\Lambda_4)^n.
\end{aligned}$$

This yields $\det(A-\lambda I_{(4n)\times(4n)}) = \lambda^{2n}(\lambda-\Lambda_1)^n(\lambda-\Lambda_4)^n$, and hence the spectral radius of A is $\rho(A) = \max\{|\Lambda_1|, |\Lambda_4|\}$. ∎

3.5. Superlinear Convergence of PSWR. We are now ready to prove the main result of this paper, namely the superlinear convergence of PSWR. We collect the norms of the functions appearing in (3.23) into vectors,

$$[\boldsymbol{e}]_t := [\|e_0\|_\infty, \dots, \|e_{N-1}\|_\infty]^T, \quad [\boldsymbol{E}]_x := [\|E_0\|_\infty, \dots, \|E_{N-1}\|_\infty]^T, \tag{3.29}$$

where the infinity norm for a function $g : (a, b) \to \mathbb{R}$ is given by

$$\|g\|_\infty := \sup_{a<s<b} |g(s)|.$$

Note that in $[\boldsymbol{E}]_x$ the infinity norms are in space, indicated by the subscript x, since $\boldsymbol{E}$ represents functions in space, and in $[\boldsymbol{e}]_t$ the infinity norms are in time, indicated by the index t, since $\boldsymbol{e}$ represents functions in time. We also define the matrix of norms of the functions in a matrix $A = [a_{ij}]$ by

$$[A]_t = [\|a_{ij}\|_\infty]. \tag{3.30}$$

THEOREM 3.4 (Superlinear Convergence). *If the fine propagator F is the exact solver, and the coarse propagator G is Backward Euler, then PSWR with Dirichlet transmission conditions and overlap L converges superlinearly on bounded time intervals $(0, T)$, i.e. the errors given by the error recursion formulas (3.4) and (3.5) satisfy the error estimate*

$$\begin{bmatrix} [\boldsymbol{e}_1^{2k}]_t \\ [\boldsymbol{E}_1^{2k}]_x \\ [\boldsymbol{e}_2^{2k}]_t \\ [\boldsymbol{E}_2^{2k}]_x \end{bmatrix} \le \tilde{\mathbb{M}}^{2k} \begin{bmatrix} [\boldsymbol{e}_1^{0}]_t \\ [\boldsymbol{E}_1^{0}]_x \\ [\boldsymbol{e}_2^{0}]_t \\ [\boldsymbol{E}_2^{0}]_x \end{bmatrix}, \tag{3.31}$$

where "$\leq$" denotes the element-by-element comparison, and for each iteration index k, the spectral radius of the iteration matrix $\tilde{\mathbb{M}}^{2k}$ can be bounded by

$$\rho(\tilde{\mathbb{M}}^{2k}) \leq erfc(\frac{kL}{\sqrt{T}}), \tag{3.32}$$

where $erfc(\cdot)$ is the complementary error function with $erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$.

Proof. To obtain a convergence estimate of the matrix iteration (3.24) representing the error recursion formulas (3.4) and (3.5) of the PSWR algorithm with Dirichlet transmission conditions, we first invert the matrix of operators on the left hand side using Lemma 3.2, which leads to

$$\begin{aligned} &\begin{bmatrix} \mathbf{I} & 0 & 0 & 0 \\ 0 & \mathbf{I} - \mathcal{C}_1 \mathbf{I}_{-1} & -\mathcal{D}_{1,\Delta T}\mathbf{I}_{-1} & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ -\mathcal{D}_{2,\Delta T}\mathbf{I}_{-1} & 0 & 0 & \mathbf{I} - \mathcal{C}_2 \mathbf{I}_{-1} \end{bmatrix}^{-1} \\ &\quad = \begin{bmatrix} \mathbf{I} & 0 & 0 & 0 \\ 0 & \mathbf{I} + B_1' & B_2' & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ B_3' & 0 & 0 & \mathbf{I} + B_4' \end{bmatrix}, \end{aligned} \tag{3.33}$$

where B_i' $(i = 1, \ldots, 4)$ are strictly lower triangular matrices of operators. Multiplying the matrix iteration (3.24) on both sides by the inverse (3.33) thus leads to the matrix iteration

$$\begin{bmatrix} \boldsymbol{e}_1^{k+1}(0,\cdot) \\ \boldsymbol{E}_1^{k+1}(x) \\ \boldsymbol{e}_2^{k+1}(L,\cdot) \\ \boldsymbol{E}_2^{k+1}(x) \end{bmatrix} = \mathbb{M} \begin{bmatrix} \boldsymbol{e}_1^{k}(0,\cdot) \\ \boldsymbol{E}_1^{k}(x) \\ \boldsymbol{e}_2^{k}(L,\cdot) \\ \boldsymbol{E}_2^{k}(x) \end{bmatrix}, \tag{3.34}$$

where the iteration matrix $\mathbb{M}$ of operators is given by

$$\mathbb{M} = \begin{bmatrix} 0 & \mathcal{P}_{1,0} & \mathcal{Q}_{1,0} & 0 \\ B_2'\mathcal{Q}_{2,L} & K_1 & K_2 & B_2'\mathcal{P}_{2,L} \\ \mathcal{Q}_{2,L} & 0 & 0 & \mathcal{P}_{2,L} \\ K_3 & B_3'\mathcal{Q}_{1,0} & B_3'\mathcal{P}_{1,0} & K_4 \end{bmatrix},$$

with the new matrices of operators appearing given by

$$\begin{aligned} K_1 &:= (\mathbf{I} + B_1')(\mathcal{P}_{1,\Delta T}\mathbf{I}_{-1} - \mathcal{C}_1\mathbf{I}_{-1}), \\ K_2 &:= (\mathbf{I} + B_1')(\mathcal{Q}_{1,\Delta T}\mathbf{I}_{-1} - \mathcal{D}_{1,\Delta T}\mathbf{I}_{-1}), \\ K_3 &:= (\mathbf{I} + B_4')(\mathcal{Q}_{2,\Delta T}\mathbf{I}_{-1} - \mathcal{D}_{2,\Delta T}\mathbf{I}_{-1}), \\ K_4 &:= (\mathbf{I} + B_4')(\mathcal{P}_{2,\Delta T}\mathbf{I}_{-1} - \mathcal{C}_2\mathbf{I}_{-1}). \end{aligned}$$

The key idea of the proof is now not to estimate the contraction over one step, which would only lead to a linear convergence estimate, but to look at the iteration over all iteration steps at once, i.e.

$$\begin{bmatrix} \boldsymbol{e}_1^{2k}(0,\cdot) \\ \boldsymbol{E}_1^{2k}(x) \\ \boldsymbol{e}_2^{2k}(L,\cdot) \\ \boldsymbol{E}_2^{2k}(x) \end{bmatrix} = \mathbb{M}^{2k} \begin{bmatrix} \boldsymbol{e}_1^{0}(0,\cdot) \\ \boldsymbol{E}_1^{0}(x) \\ \boldsymbol{e}_2^{0}(L,\cdot) \\ \boldsymbol{E}_2^{0}(x) \end{bmatrix}. \tag{3.35}$$

The $2k$-th power of the iteration matrix of operators has the structure

$$\mathbb{M}^{2k} = \begin{bmatrix} L_1 + (\mathcal{Q}_{1,0}\mathcal{Q}_{2,L})^k & L_2 & L_3 & L_4 + (\mathcal{Q}_{1,0}\mathcal{Q}_{2,L})^{k-1}\mathcal{Q}_{1,0}\mathcal{P}_{2,L} \\ L_5 & L_6 & L_7 & L_8 \\ L_9 & L_{10} + (\mathcal{Q}_{2,L}\mathcal{Q}_{1,0})^{k-1}\mathcal{Q}_{2,L}\mathcal{P}_{1,0} & L_{11} + (\mathcal{Q}_{2,L}\mathcal{Q}_{1,0})^k & L_{12} \\ L_{13} & L_{14} & L_{15} & L_{16} \end{bmatrix},$$

where all the new matrices of operators L_i $(i = 1, 2, \dots, 16)$ are strictly lower triangular, as a detailed verification like in the proof of Lemma 3.3 shows. We now take the norms defined in (3.29) in each block row of (3.35), and using the triangle inequality, we obtain the estimate (3.31) shown in the statement of the theorem. Now note that the matrix $\tilde{\mathbb{M}}^{2k}$ has the same structure as the matrix in Lemma 3.3, and we thus get for the spectral radius of $\tilde{\mathbb{M}}^{2k}$

$$\rho(\tilde{\mathbb{M}}^{2k}) = \max\{[(\mathcal{Q}_{1,0}\mathcal{Q}_{2,L})^k]_t, [(\mathcal{Q}_{2,L}\mathcal{Q}_{1,0})^k]_t\}, \tag{3.36}$$

Here $[\cdot]_t$ is defined in (3.30) for the matrices $(\mathcal{Q}_{1,0}\mathcal{Q}_{2,L})^k$ and $(\mathcal{Q}_{2,L}\mathcal{Q}_{1,0})^k$. By the definitions of $\mathcal{Q}_{1,0}$ and $\mathcal{Q}_{2,L}$ in (3.25), and using the definitions of $\mathcal{B}_{1,n,0}$ and $\mathcal{B}_{2,n,L}$ in (3.19) and (3.22), we see that $\mathcal{B}_{1,n,0} = \mathcal{B}_{2,n,L}$, and further $\mathcal{Q}_{1,0} = \mathcal{Q}_{2,L}$. Note that the diagonals of $\mathcal{Q}_{1,0}\mathcal{Q}_{2,L}$ are $\mathcal{B}_{1,n,0}\mathcal{B}_{2,n,L}$, and therefore it suffices to estimate

$$\|(\mathcal{B}_{1,n,0}\mathcal{B}_{2,n,L})^k\|_\infty = \|(\mathcal{B}_{1,n,0})^{2k}\|_\infty \le \|\int_0^t \frac{2kL}{2\sqrt{\pi}(t-\tau)^{3/2}} e^{-\frac{(2kL)^2}{4(t-\tau)}} d\tau\|_\infty,$$

where the infinity norm here is defined for the operator. Using the change of variables $y := kL/\sqrt{t-\tau}$, we obtain

$$\|(\mathcal{B}_{1,n,0}\mathcal{B}_{2,n,L})^k\|_\infty \le \operatorname{erfc}(\frac{kL}{\sqrt{T}}).$$

Therefore the spectral radius of the iteration matrix of operators $\tilde{\mathbb{M}}^{2k}$ can be bounded as shown in (3.32), which concludes the proof. □

Remark 3.5. From Theorem 3.4, we see that the spectral radius of the iteration matrix of operators $\tilde{\mathbb{M}}^{2k}$ can be bounded for each k, which gives a different asymptotic error reduction factor for each k. Our result thus captures the convergence behavior of the PSWR method much more accurately than just an estimate of the decay of the error over one iteration step; it is obtaining this convolved estimate which made the analysis so hard. Estimating over one step, we would just have obtained a classical linear convergence factor, a number less than one. Let us look at an example: let $T := 1, L := 0.1$. Then for $k = 1$, we have $\operatorname{erfc}(0.1) \approx 0.8875$ and thus $\rho(\tilde{\mathbb{M}}^2) \le 0.8875$ and PSWR converges asymptotically at least with the factor 0.8875, i.e the error is asymptotically multiplied at least by 0.8875 every two iterations. This is however only an upper bound, since if we look at $k = 2$, we have $\operatorname{erfc}(0.2) \approx 0.7773$ and thus $\rho(\tilde{\mathbb{M}}^4) \le 0.7773$ and PSWR converges asymptotically at least with the factor 0.7773, i.e the error is asymptotically multiplied at least by 0.7773 every four iterations. So the key result we obtained is much more precise than just an asymptotic linear convergence factor, it proves superlinear asymptotic convergence: if we look at $k = 20$ in our example, we have $\operatorname{erfc}(2) \approx 0.004678 \ll (\operatorname{erfc}(0.1))^{20} \approx 0.09199$ (!) and thus $\rho(\tilde{\mathbb{M}}^{40}) \le 0.004678$, an extremely fast contraction rate. We could also check the equivalent average convergence factor by taking the kth root of $\rho(M^{2k})$. When we choose $k = 1, 2$, and 20, the average convergence factor is 0.8875, 0.8816, and 0.7647, which shows that the average convergence factor decreases as the iteration number k increases. We will see in our numerical experiments, that PSWR algorithm really converges at a superlinear rate, and that our estimate is quite sharp. In order to get a norm estimate, we could also consider the norm of the iteration matrix of operators in the sense induced by the spectral radius, see [47, page 284, Lemma 1] or [73, page 795]: for every $\epsilon > 0$, we can introduce an equivalent norm $\|\cdot\|_\epsilon$ such that the corresponding operator norm satisfies

$$\rho(\tilde{M}^{2k}) \leq \|\tilde{M}^{2k}\|_\epsilon \leq \rho(\tilde{M}^{2k}) + \epsilon,$$

where $\|x\|_\epsilon := \sup_{p\geq 0}(\rho(\tilde{M}^{2k}) + \epsilon)^{-p}\|\tilde{M}^{2kp}x\|_\infty$, $x \in \mathbb{R}^{4N}$. This then implies that our algorithm is also converging superlinearly in the above norm sense.

Remark 3.6. The convergence estimate in Theorem 3.4 depends only on the size of the overlap L and the length of the entire time interval T of simulation, but it does not depend on the number of time subintervals we use in the PSWR algorithm. We will investigate in the next section how sharp this bound is, and if a similar bound would also hold for many subdomains, and optimized transmission conditions, cases which our current analysis does not cover.

4. Numerical experiments.

To investigate numerically how the convergence of the PSWR algorithm depends on the various parameters in the space-time decomposition, we use the 1-dimensional model problem

$$\begin{aligned} \frac{\partial u(x,t)}{\partial t} &= \frac{\partial^2 u(x,t)}{\partial x^2}, && (x,t) \in \Omega \times (0,T), \\ u(x,t) &= 0, && (x,t) \in \partial\Omega \times (0,T), \\ u(x,0) &= u_0, && x \in \Omega, \end{aligned} \tag{4.1}$$

where the domain $\Omega = (0,3)$, and the initial condition is $u_0 = \exp^{-3(1.5-x)^2}$. The model problem (4.1) is discretized by a second-order centered finite difference scheme with mesh size $h = 3/128$ in space and by the Backward Euler method with $\Delta t = T/100$ in time. The time interval is divided into N time subintervals, while the domain Ω is decomposed into J equal spatial subdomains with overlap L. We define the relative error of the infinity norm of the errors along the interface and initial time in the space-time subdomains as the iterative error of our new algorithm.

We first study cases which are very closely related to our analysis, with the only difference that the spatial domain must be bounded in order to perform numerical computations. We thus decompose the domain Ω into 2 spatial subdomains with overlap $L = 2h$. The total time interval length is $T = 1$. We show in Figure 2 on the left the convergence of the PSWR algorithm when the number of time subintervals equals 1 (classical Schwarz waveform relaxation), 2, 4, 10, and 20. This shows that the convergence of the algorithm does indeed not depend on the number of time subintervals, as predicted by Theorem 3.4. We also observe the superlinear convergence behavior predicted by Theorem 3.4, which is typical for waveform relaxation algorithms, see for example [31], and the estimate is asymptotically quite sharp, as one can see from the theoretical bound we also plotted in Figure 2 on the left. Here the theoretical bound is obtained from the spectral radius bound in Theorem 3.4.

We next investigate how the convergence depends on the total time interval length T, with $T \in \{0.1, 0.2, 0.5, 1, 2\}$. We divide the time interval $(0,T)$ each time into 10 time subintervals, and use the same decomposition of the domain Ω into two

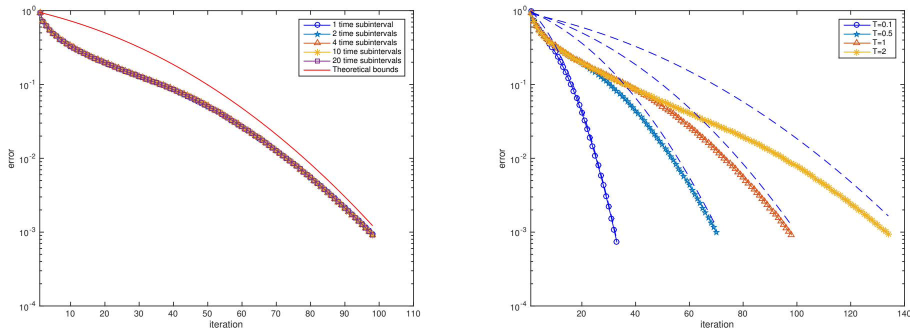

Fig. 2. *Dependence of the PSWR algorithm on the number of time subintervals (left), and the total time window length (right)*

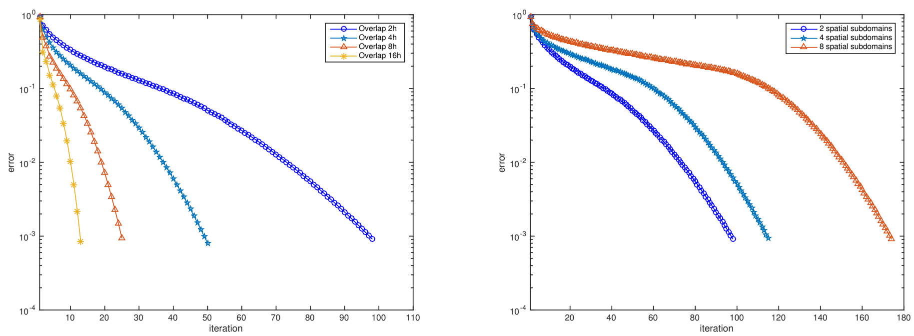

Fig. 3. *Dependence of the PSWR algorithm on the overlap (left), and on the number of spatial subdomains (right).*

subdomains with overlap $L = 2h$ as before. The results are shown in Figure 2 on the right with the corresponding asymptotically rather sharp bounds. We clearly see that the convergence of the PSWR algorithm is much faster on short time intervals, compared to long time intervals, as predicted by Theorem 3.4. We see however also that the initial convergence behavior on long time intervals seems to be linear, and independent of the length of the time interval then, a fact which is not captured by our superlinear convergence analysis.

We next study the dependence on the overlap. We use $L = 2h$, $4h$, $8h$ and $16h$, and divide the time interval $(0, T)$ with $T = 1$ into 10 time subintervals, still using the same two subdomain decomposition of Ω as before. We see on the left in Figure 3 that increasing the overlap substantially improves the convergence speed of the algorithm, as predicted by our convergence estimate in Theorem 3.4. This increases however also the cost of the method, since bigger subdomain problems need to be solved.

We now investigate numerically if a similar convergence result we derived for two subdomains also holds for the case of many subdomains. We decompose the domain Ω into 2, 4, 8 and 16 spatial subdomains, keeping again the overlap $L = 2h$. For each case, we divide the time interval $(0, T)$ with $T = 1$ into 10 time subintervals. We see in Figure 3 on the right that the algorithm on many spatial subdomains still

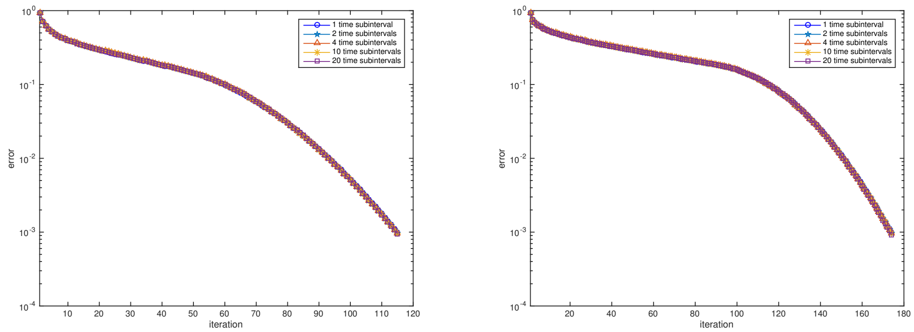

Fig. 4. *Independence of the PSWR algorithm on the number of time subintervals for four spatial subdomains (left), and eight spatial subdomains (right).*

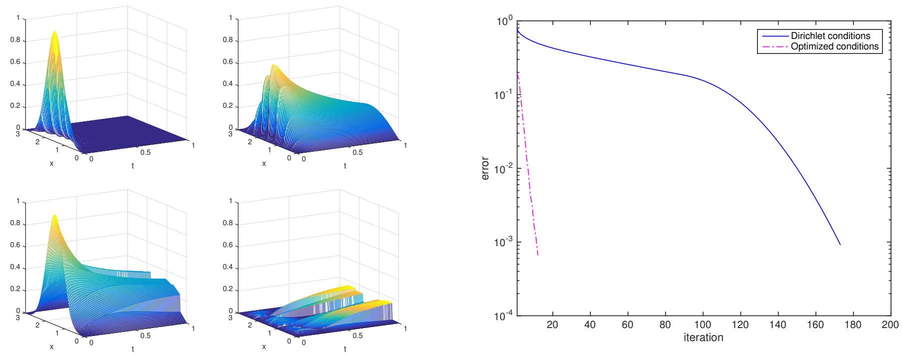

Fig. 5. *Comparison of the PSWR algorithm with Dirichlet and optimized transmission conditions. Left: third iteration and corresponding error for Dirichlet (top) and optimized (bottom) transmission conditions. Right: corresponding convergence curves.*

converges superlinearly, as predicted by our two subdomain analysis, but using more spatial subdomains makes the algorithm converge more slowly, like for the classical Schwarz method for steady problems. This can however be remedied by using smaller global time intervals T, and leads to the so called windowing techniques for waveform relaxation algorithms in general, see [34].

We further investigate whether the convergence of the algorithm still does not depend on the number of time subintervals for the case of many subdomains. We see in Figure 4 that the convergence behavior for four spatial subdomains (left), and eight spatial subdomains (right) is the same as the convergence behavior for two spatial subdomains.

Finally, we compare the convergence behavior of the PSWR algorithm with Dirichlet and optimized transmission conditions. Using optimized transmission conditions leads to much faster, so called optimized Schwarz waveform relaxation methods, see for example [32, 3]. We divide the time interval $(0, T)$ with $T = 1$ into 20 time subintervals, and the domain Ω is decomposed into 8 spatial subdomains. We use first order transmission conditions and choose for the parameters $p = 1$, $q = 1.75$ (for the terminology, see [3]). In Figure 5 we show on the left on top the third iteration

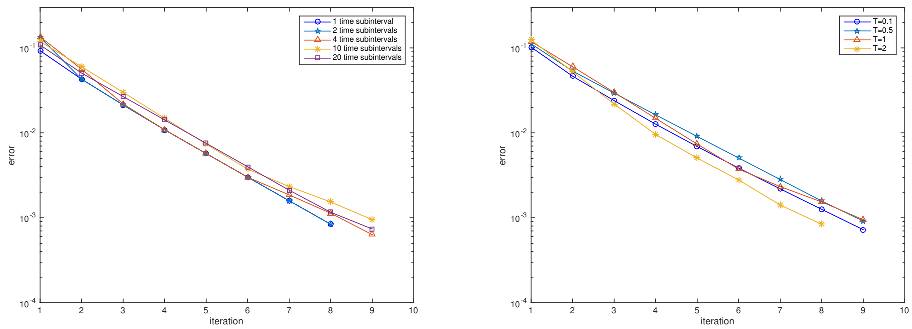

FIG. 6. *Dependence of the PSWR algorithm with optimized transmission conditions on the number of time subintervals (left), and the total time window length (right)*

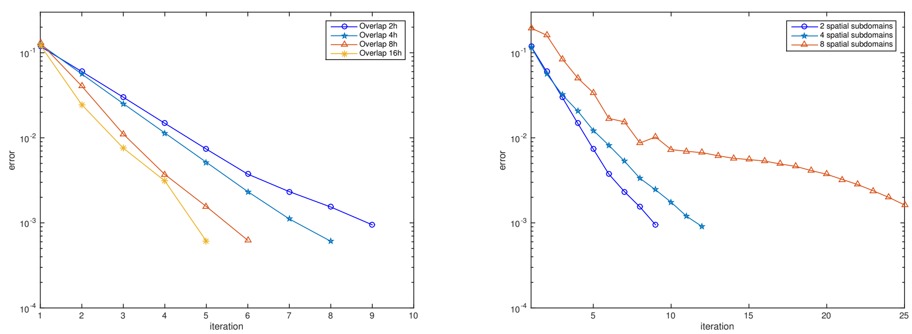

FIG. 7. *Dependence of the PSWR algorithm with optimized transmission conditions on the overlap (left) and the number of spatial subdomains (right).*

and corresponding error using Dirichlet transmission conditions, and below the third iteration and corresponding error using optimized transmission conditions. We clearly see that with optimized transmission conditions, the error is much more effectively eliminated both from the initial line and the spatial boundaries. On the right in Figure 5, the corresponding convergence curves show that using optimized transmission conditions lead to substantially better performance of the algorithm, even better than very generous overlap, and this at no additional cost, since the subdomain size and matrix sparsity is the same as for the case of Dirichlet transmission conditions. We also investigate the dependence on the number of time subintervals (on the left in Figure 6), and the total time interval length T (on the right in Figure 6), where we choose the problem configuration as in the case of the Dirichlet transmission conditions in Figure 2. We observe that convergence is much faster with optimized transmission conditions (less than 10 iterations instead of over 100), and convergence has also become linear, indicating that there is a different convergence mechanism dominating now, due to the optimized transmission conditions. We also observe that in contrast to the Dirichlet transmission condition case, convergence does now not depend any more on the length T of the overall time interval. We also test the dependence on the overlap size L (on the left in Figure 7), and on the number of spatial subdomains J (on the right in Figure 7). Comparing with the Dirichlet transmission condition case in Figure 3, we see again much faster convergence for all overlaps and spatial subdomain numbers, and convergence is also more linear again, except in the case of many spatial subdomains, where after some iterations a superlinear convergence mechanism seems to become active.

5. Conclusion.

We designed and analyzed a new PSWR algorithm for solving time-dependent PDEs. This algorithm is based on a domain decomposition of the entire space-time domain into smaller space-time subdomains, i.e. the decomposition is both in space and in time. The new algorithm iterates on these space-time subdomains using two different updating mechanisms: the Schwarz waveform relaxation approach for boundary condition updates, and the parareal mechanism for initial condition updates. All space time subdomains are solved in parallel, both in space and in time. We proved for the model problem of the one dimensional heat equation and a two subdomain decomposition in space, and arbitrary subdomain decomposition in time that the new algorithm converges superlinearly on bounded time intervals when using Dirichlet transmission conditions in space. We then tested the algorithm numerically and observed that our superlinear theoretical convergence estimate also seems to hold in the case of many subdomains, and as predicted, for fast convergence the overall time interval should not be too large (which can be achieved using a time windowing technique), or the overlap should be not too small. We then showed numerically that both these drawbacks can be greatly alleviated when using optimized transmission conditions, and we also observed that convergence then is more linear. Our results open up the path for many further research directions: is it possible to capture the different, linear convergence mechanism in the case of optimized transmission conditions using a different type of convergence analysis from ours? Can we prove that convergence then becomes independent of the length of the overall time interval? Is it possible to remove the dependence on the number of spatial subdomains using a coarse space correction, like it is done in [6] for optimized transmission conditions in the steady case? What is the convergence behavior when applied to the wave equation? Can one use in space also a Dirichlet-Neumann or Neumann-Neumann iteration, as in [26] without time decomposition? Answering these questions by analysis will be even more challenging than our first convergence estimate for this new algorithm presented here.

Appendix A. Representation formula for the solution of the G propagator.

We derive here the representation formula for the solution of the G propagator using Backward Euler. For the ordinary differential equation

$$\frac{\partial^2 u}{\partial x^2} - a^2 u = f, \quad a > 0,$$

its general solution can be expressed in the form

$$u(x) = C_1 e^{ax} + \int e^{ax-a\tau} \frac{f(\tau)}{2a} d\tau - C_2 \frac{e^{-ax}}{a} - \int e^{a\tau - ax} \frac{df(\tau)}{2a} d\tau.$$

On a bounded domain in the presence of boundary conditions, as in

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} - a^2 u = f, \quad x \in [L_1, L_2], \quad a > 0, \\ u(L_1) = g_1, \quad u(L_2) = g_2, \end{aligned}$$

one can still obtain a closed form solution, namely

$$u(x) = C_1 e^{ax} + \int_{L_1}^{x} e^{ax-a\tau}\frac{f(\tau)}{2a}d\tau - \frac{C_2 e^{-ax}}{a} - \int_{L_1}^{x} e^{a\tau-ax}\frac{f(\tau)}{2a}d\tau,$$

where

$$\begin{aligned} C_1 &= \frac{g_2 - g_1 e^{aL_1-aL_2} - \int_{L_1}^{L_2}(e^{aL_2-a\tau} - e^{a\tau-aL_2})\frac{f(\tau)}{2a}d\tau}{e^{aL_2} - e^{2aL_1-aL_2}}, \\ C_2 &= a\frac{g_2 - g_1 e^{aL_2-aL_1} - \int_{L_1}^{L_2}(e^{aL_2-a\tau} - e^{a\tau-aL_2})\frac{f(\tau)}{2a}d\tau}{e^{aL_2-2aL_1} - e^{-aL_2}}. \end{aligned}$$

Denoting by $\delta L := L_2 - L_1$ we obtain after some simplifications

$$\begin{aligned} u(x) =& \frac{e^{ax-aL_1} - e^{-ax+aL_1}}{e^{a\delta L} - e^{-a\delta L}} g_2 + \frac{e^{aL_2-ax} - e^{-aL_2+ax}}{e^{a\delta L} - e^{-a\delta L}} g_1 \\ &+ \frac{e^{aL_1-ax} - e^{ax-aL_1}}{e^{a\delta L} - e^{-a\delta L}} \int_{L_1}^{L_2} (e^{aL_2-a\tau} - e^{a\tau-aL_2})\frac{f(\tau)}{2a}d\tau \\ &+ \int_{L_1}^{x} (e^{ax-a\tau} - e^{-ax+a\tau})\frac{f(\tau)}{2a}d\tau. \end{aligned}$$

In particular, if $L_1 \to -\infty$, $L_2 = L$ and $g_1 = 0$, then we have

$$\begin{aligned} u(x) =& g_2 e^{a(x-L)} + \int_{-\infty}^{L} e^{a(x+\tau-2L)}\frac{f(\tau)}{2a}d\tau - \int_{x}^{L} e^{a(x-\tau)}\frac{f(\tau)}{2a}d\tau \\ &- \int_{-\infty}^{x} e^{-a(x-\tau)}\frac{f(\tau)}{2a}d\tau, \end{aligned}$$

and if $L_1 = 0$, $L_2 \to +\infty$ and $g_2 = 0$, then we have

$$u(x) = g_1 e^{-ax} + \int_{0}^{+\infty} e^{-a(x+\tau)}\frac{f(\tau)}{2a}d\tau - \int_{0}^{x} e^{-a(x-\tau)}\frac{f(\tau)}{2a}d\tau - \int_{x}^{+\infty} e^{a(x-\tau)}\frac{f(\tau)}{2a}d\tau.$$

Acknowledgments. We would like to thank the anonymous referees for their helpful comments.

REFERENCES

[1] L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zrah, *Parallel-in-time molecular-dynamics simulations*, Phys. Rev. E, 66 (2002), p. 057701.

[2] G. Bal, *Parallelization in time of (stochastic) ordinary differential equations*, Math. Meth. Anal. Num.(submitted), (2003).

[3] D. Bennequin, M. J. Gander, and L. Halpern, *A homographic best approximation problem with application to optimized Schwarz waveform relaxation*, Mathematics of Computation, 78 (2009), pp. 185–223.

[4] K. Burrage, *Parallel and sequential methods for ordinary differential equations*, Clarendon Press, Oxford, 1995.

[5] J. R. Cannon, *The one-dimensional heat equation, volume 23 of Encyclopedia of Mathematics and its Applications*, Cambridge University Press, 1984.

[6] O. Dubois, M. J. Gander, S. Loisel, A. St-Cyr, and D. B. Szyld, *The optimized Schwarz method with a coarse grid correction*, SIAM Journal on Scientific Computing, 34 (2012), pp. A421–A458.

[7] M. Emmett and M. L. Minion, *Toward an efficient parallel in time method for partial differential equations*, Comm. App. Math. and Comp. Sci, 7 (2012), pp. 105–132.

[8] S. Engblom, *Parallel in Time Simulation of Multiscale Stochastic Chemical Kinetics*, Multiscale Modeling & Simulation, 8 (2009), pp. 46–68.

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

[10] P. F. Fischer, F. Hecht, and Y. Maday, *A parareal in time semi-implicit approximation of the Navier-Stokes equations*, in Domain Decomposition Methods in Science and Engineering, R. Kornhuber and et al., eds., vol. 40 of Lecture Notes in Computational Science and Engineering, Berlin, 2005, Springer, pp. 433–440.

[11] S. Friedhoff, R. Falgout, T. Kolev, S. MacLachlan, and J. B. Schroder, *A multigrid-in-time algorithm for solving evolution equations in parallel*, in Sixteenth Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, United States, 2013.

[12] M. Gander, L. Halpern, J. Rannou, and J. Ryan, *A direct time parallel solver by diagonalization for the wave equation*, to appear in SIAM Journal on Scientific Computing, (2018).

[13] M. J. Gander, *A waveform relaxation algorithm with overlapping splitting for reaction diffusion equations*, Numerical Linear Algebra with Applications, 6 (1999), pp. 125–145.

[14] M. J. Gander, *Optimized Schwarz methods*, SIAM Journal on Numerical Analysis, 44 (2006), pp. 699–731.

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

[16] M. J. Gander, M. Al-Khaleel, and A. E. Ruehli, *Waveform relaxation technique for longitudinal partitioning of transmission lines*, in Electrical Performance of Electronic Packaging, IEEE, 2006, pp. 207–210.

[17] M. J. Gander and S. Güttel, *PARAEXP: A parallel integrator for linear initial-value problems*, SIAM Journal on Scientific Computing, 35 (2013), pp. C123–C142.

[18] M. J. Gander and E. Hairer, *Nonlinear convergence analysis for the parareal algorithm*, Lecture Notes in Computational Science and Engineering, 60 (2008), pp. 45–56.

[19] M. J. Gander and L. Halpern, *Absorbing boundary conditions for the wave equation and parallel computing*, Math. of Comp., 74 (2004), pp. 153–176.

[20] M. J. Gander and L. Halpern, *Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems*, SIAM J. Numer. Anal., 45 (2007), pp. 666–697.

[21] M. J. Gander, L. Halpern, and F. Nataf, *Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation*, in Eleventh international Conference of Domain Decomposition Methods, C.-H. Lai, P. Bjørstad, M. Cross, and O. Widlund, eds., ddm.org, 1999.

[22] M. J. Gander, L. Halpern, and F. Nataf, *Optimal Schwarz waveform relaxation for the one dimensional wave equation*, SIAM Journal of Numerical Analysis, 41 (2003), pp. 1643–1681.

[23] M. J. Gander, L. Halpern, J. Ryan, and T. T. B. Tran, *A direct solver for time parallelization*, in Domain Decomposition Methods in Science and Engineering XXII, Springer, 2016, pp. 491–499.

[24] M. J. Gander, Y.-L. Jiang, and R.-J. Li, *Parareal Schwarz waveform relaxation methods*, in Domain Decomposition Methods in Science and Engineering XX, Springer, 2013, pp. 451–458.

[25] M. J. Gander, Y.-L. Jiang, B. Song, and H. Zhang, *Analysis of two parareal algorithms for time-periodic problems*, SIAM Journal on Scientific Computing, 35 (2013), pp. A2393–A2415.

[26] M. J. Gander, F. Kwok, and B. C. Mandal, *Dirichlet-Neumann and Neumann-Neumann waveform relaxation algorithms for parabolic problems*, Electron. Trans. Numer. Anal, 45 (2016), pp. 424–456.

[27] M. J. Gander, F. Kwok, and H. Zhang, *Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT*, Computing and Visualization in Science, (2018).

[28] M. J. Gander and M. Neumuller, *Analysis of a new space-time parallel multigrid algorithm for parabolic problems*, SIAM Journal on Scientific Computing, 38 (2016), pp. A2173–A2208.

[29] M. J. Gander and C. Rohde, *Overlapping Schwarz waveform relaxation for convection-dominated nonlinear conservation laws*, SIAM journal on Scientific Computing, 27 (2005), pp. 415–439.

[30] M. J. Gander and A. E. Ruehli, *Optimized waveform relaxation methods for RC type circuits*, IEEE Transactions on Circuits and Systems I: Regular Papers, 51 (2004), pp. 755–768.

[31] M. J. Gander and A. M. Stuart, *Space-time continuous analysis of waveform relaxation for the heat equation*, SIAM Journal on Scientific Computing, 19 (1998), pp. 2014–2031.

[32] M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, SIAM Journal on Scientific Computing, 29 (2007), pp. 556–578.

[33] M. J. Gander and H. Zhang, *Iterative solvers for the Helmholtz equation: factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods*, SIAM Review, to appear, (2018).

[34] M. J. Gander and H. Zhao, *Overlapping Schwarz waveform relaxation for the heat equation in n-dimensions*, BIT Numerical Mathematics, 42 (2002), pp. 779–795.

[35] E. Giladi and H. B. Keller, *Space-time domain decomposition for parabolic problems*, Numerische Mathematik, 93 (2002), pp. 279–313.

[36] R. Guetat, *Méthode de parallélisation en temps: application aux méthodes de décomposition de domaine*, PhD thesis, Paris 6, 2011.

[37] S. Güttel, *A parallel overlapping time-domain decomposition method for ODEs*, in Domain decomposition methods in science and engineering XX, Springer, 2013, pp. 459–466.

[38] L. Halpern and J. Szeftel, *Nonlinear nonoverlapping Schwarz waveform relaxation for semilinear wave propagation*, Mathematics of Computation, 78 (2009), pp. 865–889.

[39] R. A. Horn and C. R. Johnson, *Matrix analysis*, 2nd ed., Cambridge university press, 2012.

[40] C. Japhet, *Optimized Krylov-Ventcell method. application to convection-diffusion problems*, in Proceedings of the 9th international conference on domain decomposition methods, ddm.org, 1998, pp. 382–389.

[41] Y.-L. Jiang, *On time-domain simulation of lossless transmission lines with nonlinear terminations*, SIAM Journal on Numerical Analysis, 42 (2004), pp. 1018–1031.

[42] Y.-L. Jiang, *Waveform relaxation methods (in Chinese)*, Scientific Press, Beijing, 2010.

[43] Y.-L. Jiang and R. Chen, *Computing periodic solutions of linear differential-algebraic equations by waveform relaxation*, Mathematics of computation, 74 (2005), pp. 781–804.

[44] Y.-L. Jiang, R. M. Chen, and O. Wing, *Periodic waveform relaxation of nonlinear dynamic systems by quasi-linearization*, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 50 (2003), pp. 589–593.

[45] Y.-L. Jiang and X.-L. Ding, *Waveform relaxation methods for fractional differential equations with the Caputo derivatives*, Journal of Computational and Applied Mathematics, 238 (2013), pp. 51–67.

[46] Y.-L. Jiang and O. Wing, *A note on the spectra and pseudospectra of waveform relaxation operators for linear differential-algebraic equations*, SIAM Journal on Numerical Analysis, 38 (2000), pp. 186–201.

[47] Y.-L. Jiang and O. Wing, *A note on convergence conditions of waveform relaxation algorithms for nonlinear differential–algebraic equations*, Applied Numerical Mathematics, 36 (2001), pp. 281–297.

[48] E. Lelarasmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelli, *The waveform relaxation method for time-domain analysis of large scale integrated circuits*, IEEE Trans. on CAD of IC and Syst., 1 (1982), pp. 131–145.

[49] J.-L. Lions, Y. Maday, and G. Turinici, *A "parareal" in time discretization of PDE's*, Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, 332 (2001), pp. 661–668.

[50] J. Liu and Y.-L. Jiang, *Waveform relaxation for reaction–diffusion equations*, Journal of Computational and Applied Mathematics, 235 (2011), pp. 5040–5055.

[51] J. Liu and Y.-L. Jiang, *A parareal waveform relaxation algorithm for semi-linear parabolic partial differential equations*, Journal of Computational and Applied Mathematics, 236 (2012), pp. 4245–4263.

[52] J. Liu and Y.-L. Jiang, *A parareal algorithm based on waveform relaxation*, Mathematics and Computers in Simulation, 82 (2012), pp. 2167–2181.

[53] C. Lubich and A. Ostermann, *Multi-grid dynamic iteration for parabolic equations*, BIT Numerical Mathematics, 27 (1987), pp. 216–234.

[54] Y. Maday and E. M. Rønquist, *Parallelization in time through tensor-product space–time solvers*, Comptes Rendus Mathematique, 346 (2008), pp. 113–118.

[55] Y. Maday, J. Salomon, and G. Turinici, *Monotonic parareal control for quantum systems*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 2468–2482.

[56] Y. Maday and G. Turinici, *A parareal in time procedure for the control of partial differential equations*, Comptes Rendus de l'Académie des Sciences-Series I-Mathematics, 335 (2002), pp. 387–392.

[57] Y. Maday and G. Turinici, *Parallel in time algorithms for quantum control: Parareal time discretization scheme*, Int. J. Quant. Chem., 93 (2003), pp. 223–228.
[58] Y. Maday and G. Turinici, *The parareal in time iterative solver: a further direction to parallel implementation*, Lecture Notes in Computational Science and Engineering, 40 (2005), pp. 441–448.
[59] M. L. Minion, *A hybrid parareal spectral deferred corrections method*, Communications in Applied Mathematics and Computational Science, 5 (2011), pp. 265–301.
[60] M. L. Minion and S. A. Williams, *Parareal and spectral deferred corrections*, in AIP Conference Proceedings, AIP, 2008, pp. 388–391.
[61] S. Schöps, I. Niyonzima, and M. Clemens, *Parallel-in-time simulation of eddy current problems using parareal*, IEEE Trans. Magn., 54 (2018), https://doi.org/10.1109/TMAG.2017.2763090, https://arxiv.org/abs/1706.05750.
[62] H. A. Schwarz, *Über einen Grenzübergang durch alternierendes Verfahren*, Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, 15 (1870), pp. 272–286.
[63] B. Song and Y.-L. Jiang, *Analysis of a new parareal algorithm based on waveform relaxation method for time-periodic problems*, Numerical Algorithms, 67 (2014), pp. 599–622.
[64] B. Song and Y.-L. Jiang, *A new parareal waveform relaxation algorithm for time-periodic problems*, International Journal of Computer Mathematics, 92 (2015), pp. 377–393.
[65] G. Staff, *Convergence and Stability of the Parareal Algorithm*, master's thesis, Norwegian University of Science and Technology, Norway, 2003.
[66] G. A. Staff and E. M. Rønquist, *Stability of the parareal algorithm*, Domain decomposition methods in science and engineering, 40 (2005), pp. 449–456.
[67] J. M. F. Trindade and J. C. F. Pereira, *Parallel-in-time simulation of the unsteady Navier-Stokes equations for incompressible flow*, International Journal for Numerical Methods in Fluids, 45 (2004), pp. 1123–1136.
[68] S. Vandewalle and R. Piessens, *Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations*, SIAM journal on scientific and statistical computing, 13 (1992), pp. 1330–1346.
[69] S. Vandewalle and E. Van de Velde, *Space-time concurrent multigrid waveform relaxation*, Annals of Numerical Mathematics, 1 (1994), pp. 335–346.
[70] S.-L. Wu, *Toward parallel coarse grid correction for the parareal algorithm*, SIAM Journal on Scientific Computing, 40 (2018), pp. A1446–A1472.
[71] S.-L. Wu, H. Zhang, and T. Zhou, *Solving time-periodic fractional diffusion equations via diagonalization technique and multigrid*, Numerical Linear Algebra with Applications, (2018), p. e2178.
[72] S.-L. Wu and T. Zhou, *Fast parareal iterations for fractional diffusion equations*, Journal of Computational Physics, 329 (2017), pp. 210–226.
[73] E. Zeidler, *Nonlinear functional analysis and its applications, Vol 1:Fixed-Point Theorems*, Springer, Berlin, 1986.