

Computational Physics

Quasi-optimal domain decomposition method for neural network-based computation of the time-dependent Schrödinger equation [☆]

Emmanuel Lorin ^{a,b,*}, Xu Yang ^c^a School of Mathematics and Statistics, Carleton University, Ottawa, K1S 5B6, Canada^b Centre de Recherches Mathématiques, Université de Montréal, Montréal, H3T 1J4, Canada^c Department of Mathematics, University of California, Santa Barbara, CA 93106, USA

ARTICLE INFO

Keywords:

Schrödinger equation
 Schwarz waveform relaxation
 Dirichlet-to-Neumann operator
 Pseudo-differential calculus
 Physics-informed neural networks

ABSTRACT

In this paper, we derive and analyze the performance of optimal/quasi-optimal Schwarz Waveform Relaxation (SWR) domain decomposition methods (DDM) for the time-dependent Schrödinger equation when implemented with neural network-based Partial Differential Equations (PDE) solvers. Optimal SWR methods, which are based on Dirichlet-to-Neumann operators, are known to have a higher convergence rate than classical or optimized SWR methods. However, they are usually considered prohibitive due to their computational costs with standard PDE solvers. Thanks to Physics Informed Neural Network acceleration within the Schwarz waveform relaxation process and an efficient computation of Dirichlet-to-Neumann transmission operators, we demonstrate that optimal and quasi-optimal SWR methods can be performed almost as efficiently as classical or optimized SWR methods while maintaining a faster convergence rate. We present a few numerical examples to illustrate the performance and convergence of the proposed method.

1. Introduction

In this paper, our focus is on computing the time-dependent Schrödinger equation (TDSE) using the optimal or quasi-optimal Schwarz Waveform Relaxation (SWR) domain decomposition method (DDM) [1], which relies on Dirichlet-to-Neumann-like transmission conditions. At the PDE level, optimal and quasi-optimal SWR methods are known to converge much faster than classical (based on Dirichlet transmission conditions) or optimized (based on Robin transmission conditions) SWR methods. However, the approximation of the corresponding local initial boundary value problems with standard Schrödinger equation solvers is i) more computationally complex (ill-conditioned linear systems), ii) requires large data storage due to the nonlocality of Dirichlet-to-Neumann (DtN) operators, and iii) may lead to numerical instabilities. For these reasons, classical SWR or Optimized SWR methods are usually preferred, even if they provide a slower convergence rate than optimal SWR methods. Additional details can be found in [2–5]. Here, we show that, thanks to Physics Informed Neural Networks (PINN) [6], optimal and quasi-optimal Schwarz Waveform

Relaxation (SWR) methods can be numerically performed almost as efficiently as classical and optimized SWR methods. PINN algorithms offer two main advantages: i) they introduce learning into the Schwarz waveform relaxation process, as observed in [7], and ii) they enable the efficient computation of Dirichlet-to-Neumann transmission operators through automatic differentiation (and possibly integration, currently under investigation) of neural networks. It is important to note that the term *learning* refers to the acceleration of the optimization algorithms within the Schwarz process, specifically to the initialization of the optimization algorithms within the Schwarz algorithm. We should also mention that other types of neural network-based algorithms for solving Partial Differential Equations (PDEs) can be considered instead of PINN [8–11]. To clarify this concept, we provide some preliminary information about the SWR method and PINN algorithms. We insist on the fact that although PINN methods enable the efficient performance of optimal SWR methods, in general and particularly in low dimensions, standard PDE solvers usually remain much more computationally efficient. The central idea of this paper is to demonstrate that within the PINN framework, unlike standard Schrödinger equation solvers,

[☆] The review of this paper was arranged by Prof. Weigel Martin.

* Corresponding author.

E-mail addresses: elarin@math.carleton.ca (E. Lorin), xuyang@math.ucsb.edu (X. Yang).

optimal and quasi-optimal SWR methods can be performed almost as efficiently as classical or Robin SWR methods while maintaining a faster convergence rate.

1.1. Introductory remarks

Optimal and quasi-optimal SWR methods are domain decomposition methods for evolutionary PDEs, which are based on transparent transmission operators (typically such as Dirichlet-to-Neumann-like operators), and which are known to provide a very fast convergence of the Schwarz process [1,12,13], as less as two Schwarz iterations at the continuous level, and in the most simple configuration. Optimal Schwarz waveform relaxation methods are more generally constructed using Nirenberg's factorization of the Schrödinger operator (and more generally evolution wave operators) at the subdomain interfaces [1,14]. In one dimension, it simply consists of the following factorization,

$$\partial_t - i\partial_{xx} = (\sqrt{\partial_t} + e^{i\pi/4}\partial_x)(\sqrt{\partial_t} - e^{i\pi/4}\partial_x),$$

and allowing an incoming/outgoing wave decomposition at the subdomain interface. The pseudo-differential (fractional operator) $\partial_t^{1/2}$ is a nonlocal operator which is, for instance, defined by Riemann-Liouville integrals [3]. More specifically, for a real integrable function f , we define

$$D_t^{1/2} f(t) = D_t \int_0^t \frac{f(\tau)}{\sqrt{\pi(t-\tau)}} d\tau,$$

where D_t^α is the so-called α -derivative with respect to t . It is well-known that the approximation of Dirichlet-to-Neumann operators is far from trivial, and often leads to numerical stability, ill-posed linear systems with accuracy and storage issues, as observed in the framework of absorbing boundary conditions [15,2,4] or Schwarz waveform relaxation [12,13]. Hence for TDSE, even if the optimal Schwarz waveform relaxation method allows for an "optimal" convergence, much faster than classical Schwarz (based on Dirichlet transmission conditions) or even Optimized Schwarz (based on optimized Robin transmission conditions), they are often considered as prohibitive from the computational point of view. Practically, an optimized Robin transmission operator usually provides the best compromise between computational efficiency, accuracy, and SWR convergence rate. We summarize the difficulties related to standard discretizations (finite element/difference/volume methods) of the DtN-like transmission conditions (see [2,16,17]):

- Storage of the solution at any time at the subdomain interfaces;
- Stability and accuracy issues;
- Loss of efficiency due to the numerical computation of ill-posed linear systems.

PINN algorithms are a "new" type of PDE solvers which consist in i) searching the PDE solution in the form of a neural network (that is a parameterized given function), ii) optimizing the neural network (NN) parameters by minimizing a loss function written as a continuous PDE residual and iii) including (experimental or numerical) data. The *space-and-time* approximate solution is then given by a neural network evaluated at a set of optimized parameters. This approach benefits in particular from i) the use of automatic differentiation which allows for exact computation of partial derivatives (no differential operator approximation), and ii) the use of efficient stochastic methods for optimizing a loss function constructed as the norm of PDE residuals at randomly chosen space&time points. On the other hand, as far as we know, automatic integration is not rigorously treated in the literature, and as a consequence, as in this paper, fractional time-derivatives will be computed using appropriate quadratures. We do not delve into the discussion of the interest/relevance of using NN-based algorithms for

solving PDEs, but rather, we refer to [6,18,19]. Also, note that neural networks are now widely used in quantum chemistry for solving high-dimensional eigenvalue problems in connection with the energy states of large molecules [9,20].

In this paper, we are interested in the consequences of using optimal SWR-DDM in combination with PINN algorithms:

1. At Schwarz iteration k , we initiate the optimization algorithm using the set of converged parameters obtained at the previous Schwarz iteration $k-1$, corresponding to the approximate solution at iteration $k-1$. Consequently, we anticipate an acceleration of the convergence of the optimization algorithm as k increases. In comparison, achieving a similar acceleration with standard numerical solvers involving time-stepping would require storing the approximate solutions *at every time step*.
2. The discretization of time-derivatives in standard evolution PDE solvers is usually the source of numerical linear instabilities. While automatic differentiation applied to time-derivatives may, at first sight, circumvent this issue (see, for instance, [21] for the Dirac equation), the discretization and minimization of the loss function by Monte Carlo integration may potentially introduce some numerical instability. As far as we know, this interesting question has not been rigorously addressed in the literature.
3. The minimization of the loss functions may be complexified due to the nonlocal operator involved in the residual.

In theory, we expect an improvement for Points 1. and 2. using PINN compared to standard Real Space Methods (RSM). The key point is then to study the overall efficiency of the optimization with nonlocal (DtN-type) boundary conditions, Point 3. Let us mention that domain decomposition in the framework of PINN algorithms has been studied in several recent works, such as [22], where space-time decomposition is directly implemented in the minimization of the loss function. In [23], Schwarz domain decomposition is proposed for stationary equations, and evolution PDEs are considered in [24].

For the sake of simplicity, in this paper, we will mainly work in a one-dimensional framework, although the concepts, algorithms, and implementations discussed in Subsections 2.3 and 3.2 are identical in higher dimensions.

1.2. PINN for the time-dependent Schrödinger equation

In this paper, we consider the following TDSE:

$$\begin{aligned} i\partial_t u + \partial_{xx} u + V(x)u &= 0, \quad x \in \mathbb{R}, t \geq 0, \\ |u(x, t)| &\rightarrow_{x \rightarrow \pm\infty} 0, \quad t \geq 0, \\ u(x, 0) &= u_0(x), \quad x \in \mathbb{R}, \end{aligned} \quad (1)$$

where u_0 denotes the Cauchy data, and V is a smooth, real-valued potential that depends on space and is positive (respectively negative) for attractive (respectively repulsive) interactions. To solve the Schrödinger equation using PINN, we first consider a bounded spatial domain denoted by Ω , which is assumed to contain the support of u_0 . We impose the boundary conditions $Mu = 0$ at $\Gamma := \partial\Omega$, where M is a boundary operator. Therefore, we consider the following TDSE on $\Omega \times [0, T]$, for some $T > 0$,

$$\begin{aligned} i\partial_t u + \partial_{xx} u + V(x)u &= 0 && \text{in } \Omega \times [0, T], \\ Mu &= 0 && \text{in } \Gamma \times [0, T], \\ u(\cdot, 0) &= u_0 && \text{in } \Omega. \end{aligned} \quad (2)$$

In practice, M is often chosen as the DtN-like operator (absorbing or transparent conditions) to avoid artificial wave reflections; alternatively the Robin operator or the identity operator (for Dirichlet conditions) can be used for large enough domains.

PINN algorithms generalize Lagaris' work [25] on differential equation computation and involve approximating the solution to (2) using a parameterized neural network denoted by $N(\theta, x, t)$. The parameters

of the network are optimized by minimizing a discrete version of the following loss function:

$$\mathcal{L}(\theta) = \lambda \|\mathbf{i} \partial_t + \partial_{xx} + V(x)\mathbf{N}(\theta, \cdot, \cdot)\|_{L^2(\Omega \times [0, T])} + \mu \|\mathbf{M}\mathbf{N}(\theta, \cdot, \cdot)\|_{L^2(\Gamma \times [0, T])} + \kappa \|\mathbf{N}(\theta, \cdot, 0) - u_0\|_{L^2(\Omega)},$$

where λ , μ and κ are positive parameters and $\theta \in \Theta \in \mathbb{R}^P$ for some “large” P . The notation $\|\cdot\|_{L^2(\Omega \times [0, T])}$ (resp. $\|\cdot\|_{L^2(\Gamma \times [0, T])}$) denotes the L^2 -norm over $\Omega \times [0, T]$ (resp. $\Gamma \times [0, T]$). To numerically construct the loss function by Monte Carlo integration, a large number of space-time input data points $\{(x_j, t_n)\}_{j,n}$ are randomly chosen. Karniadakis et al. have developed numerous techniques to improve of PINN algorithms for direct and inverse PDEs problems; see [6,18,19] for more details.

1.3. Organization of the paper

In Section 2, we discuss the combination of PINN with (quasi-)optimal SWR methods. We examine some important convergence properties of this solver, particularly in comparison with combined Robin-SWR methods. In Section 3, we provide additional details on the implementation of the PINN and SWR algorithms and present a complexity analysis. We then present some numerical experiments in Section 4, and in Section 5.

2. Optimal and quasi-optimal SWR methods with PINN

The purpose of this paper is to demonstrate the relevance of optimal (or quasi-optimal) SWR methods when used in conjunction with PINN algorithms. In this section, we introduce the basics of PINN-SWR algorithms.

2.1. Optimal and quasi-optimal SWR methods

We recall here the basics of SWR methods, particularly its optimal version based on (transparent or absorbing) Dirichlet-to-Neumann-like operators. As this paper focuses on fundamental principles and understanding, we consider a simple setting of two subdomains with or without overlap. We denote these subdomains as $\Omega_\varepsilon^+ = (-a, +\varepsilon/2)$ and $\Omega_\varepsilon^- = (-\varepsilon/2, a)$ for some $\varepsilon \geq 0$ and $a \in \mathbb{R}_+^*$. We recall that SWR methods on two subdomains consist in solving:

$$\begin{aligned} \mathbf{i} \partial_t u^{\pm(k)} &= -\partial_{xx} u^{\pm(k)} - V(x) u^{\pm(k)}, \quad \text{in } \Omega_\varepsilon^\pm \times [0, T], \\ u^{\pm(k)}(\cdot, 0) &= u_0^\pm, \quad \text{in } \Omega_\varepsilon^\pm, \\ \mathcal{T}_\pm(x, t) u^{\pm(k)} &= \mathcal{T}_\pm(x, t) u^{\mp(k-1)}, \quad \text{on } \Gamma_\varepsilon^\pm \times [0, T], \\ u^{\pm(k)} &= 0, \quad \text{on } \Lambda_\varepsilon^\pm \times [0, T], \end{aligned} \quad (3)$$

where $\mathcal{T}_\pm(x, t)$ is a boundary operator, and where we have denoted $\Gamma_\varepsilon^\pm = \{\pm\varepsilon/2\}$, and $\Lambda_\varepsilon^\pm = \partial\Omega_\varepsilon^\pm \setminus \Gamma_\varepsilon^\pm = \{\mp a\}$. The well-posedness and the convergence of this method and its rate of convergence were established in [1,5,12,13] for different types of transmission conditions applied to the Schrödinger equation. In this paper, we focus on the optimal SWR-methods which rely on transparent/absorbing transmission operators obtained by Nirenberg’s factorization at the subdomain interfaces [14,12]. Below we recall some fundamental results relative to the construction of optimal and quasi-optimal Schwarz waveform relaxation methods. Let us start by recalling the principle of Nirenberg’s factorization.

$$\mathbf{i} \partial_t + \partial_{xx} + V(x) = (\partial_x + \mathbf{i} L^-)(\partial_x + \mathbf{i} L^+) + \mathcal{R}, \quad (4)$$

where $\mathcal{R} \in \text{OPS}^{-\infty} = \bigcap_m \text{OPS}^m$ (with OPS^m denoting the set of order m differential operators) is a smooth pseudo-differential operator and $L^\pm$ are pseudo-differential operators of order $1/2$ in time and order 0 in space, which can be constructed by expanding its symbol $\lambda^\pm$ in the form

$$\lambda^\pm \sim \sum_{j=0}^{\infty} \lambda_{1/2-j/2}^\pm \quad (5)$$

where $\lambda_{1/2-j/2}^\pm$ are elementary symbols corresponding to operators of order $1/2 - j/2$, $j \in \mathbb{N}$.

- when V is constant, $\lambda^\pm = \mp \sqrt{-\tau + V}$.
- When V is not constant, we construct *quasi-optimal SWR methods of order p* (p-OSWR) involving DtN operators at the subdomain interfaces with the following operators:

$$\mathcal{T}^\pm(x, t) = \partial_x + \mathbf{i} L^{\pm,p}(x, t, \partial_x, \partial_t), \quad (6)$$

where for $p = 0, 1, 2$, $L^{\pm,p}$ is given by

$$\begin{aligned} L^{\pm,0}(x, t, \partial_x, \partial_t) u &= \pm e^{\mathbf{i}\pi/4} \partial_t^{1/2} u, \\ L^{\pm,1}(x, t, \partial_x, \partial_t) u &= \pm e^{\mathbf{i}\pi/4} e^{\mathbf{i}w(x,t)} \partial_t^{1/2} (e^{-\mathbf{i}w(x,t)} u(x, t)), \\ L^{\pm,2}(x, t, \partial_x, \partial_t) u &= L^{\pm,1} w(x, t) + \pm \frac{1}{4} V'(x) e^{\mathbf{i}w(x,t)} I_t (e^{-\mathbf{i}w(x,t)} u(x, t)), \end{aligned} \quad (7)$$

and where the function w is defined in the linear case by $w(x, t) = tV(x)$.

2.2. PINN-SWR methods

Instead of using standard approximation methods such as finite elements/differences or pseudo-spectral methods [13,5], we propose to solve the system using PINN. While other types of NN-based solvers exist (see [8,10,11]), we choose PINN for their simplicity and flexibility. The generic NN to optimize is denoted by $\mathbf{N}(\theta, x, t)$, where θ (belongs to a vector space Θ) represents the unknown parameters. The PINN-SWR allows for adaptability in the depth of the neural networks, depending on the local structure of the solution. We consider

$$\begin{aligned} \mathbf{i} \partial_t N^{\pm(k)} &= -\partial_{xx} N^{\pm(k)} - V(x) N^{\pm(k)}, \quad \text{in } \Omega_\varepsilon^\pm \times [0, T], \\ N^{\pm(k)}(\cdot, 0) &= u_0^\pm, \quad \text{in } \Omega_\varepsilon^\pm, \\ \mathcal{T}_\pm(x, t) N^{\pm(k)} &= \mathcal{T}_\pm(x, t) N^{\mp(k-1)}, \quad \text{on } \Gamma_\varepsilon^\pm \times [0, T], \\ N^{\pm(k)} &= 0, \quad \text{on } \Lambda_\varepsilon^\pm \times [0, T], \end{aligned} \quad (8)$$

where $N^{\pm(k)}$ denotes the local neural network in $\Omega_\varepsilon^\pm$ at Schwarz iteration k , and where i) $\mathcal{T}_\pm(x, t) = I$ is the identity operator for the classical SWR method, ii) $\mathcal{T}_\pm(x, t) = \partial_x + rI$ for some constant r for the Robin/optimized SWR method, and iii) $\mathcal{T}_\pm(x, t) = \partial_x + \mathbf{i} \Lambda^{\pm,p}(x, t)$ for the optimal or quasi-optimal SWR method. For instance at $x = \pm\varepsilon/2$ (for $\Gamma_\varepsilon^\pm = \{\pm\varepsilon/2\}$) and $p = 0$, the quasi-optimal SWR method corresponds to

$$(\partial_x + \mathbf{i} \Lambda^{\pm,0}) N^{\pm}(\theta, \pm\varepsilon/2, t) = \partial_x u \pm e^{-\mathbf{i}\pi/4} \frac{\partial}{\partial t} \int_0^t \frac{N^{\pm}(\theta, \pm\varepsilon/2, \tau)}{\sqrt{\pi(t-\tau)}} d\tau.$$

At Schwarz iteration k , we minimize at $(\pm\varepsilon/2, t)$

$$\begin{aligned} \mathcal{L}^\pm(\theta^{\pm(k)}) &= \lambda^\pm \left\| \mathbf{i} \partial_t N^{\pm(k)}(\theta^{\pm(k)}, \cdot, \cdot) + \partial_{xx} N^{\pm(k)}(\theta^{\pm(k)}, \cdot, \cdot) \right. \\ &\quad \left. + V(x) N^{\pm(k)}(\theta^{\pm(k)}, \cdot, \cdot) \right\|_{L^2(\Omega_\varepsilon^\pm \times [0, T])} \\ &\quad + \mu^\pm \left\| \mathcal{T}_\pm(\pm\varepsilon/2, t) N^{\pm(k)}(\theta^{\pm(k)}, \pm\varepsilon/2, \cdot) \right. \\ &\quad \left. - \mathcal{T}_\pm(\pm\varepsilon/2, t) N^{\mp(k-1)}(\bar{\theta}^\pm, \pm\varepsilon/2, \cdot) \right\|_{L^2(0, T)} \\ &\quad + \kappa^\pm \left\| N^{\pm(k)}(\theta^{\pm(k)}, \cdot, \cdot) \right\|_{L^2(\Lambda_\varepsilon^\pm \times [0, T])} \\ &\quad + \zeta^\pm \left\| N^{\pm(k)}(\theta^{\pm(k)}, 0, \cdot) - u_0^\pm(\cdot) \right\|_{L^2(\Omega_\varepsilon^\pm)}, \end{aligned} \quad (9)$$

where we have denoted $\bar{\theta}^\pm = (\bar{\theta}^{\pm(k-1)})$ computed at the Schwarz iteration $k-1$. The convergence criterion of the PINN-SWR algorithm reads as follows

$$\lim_{k \rightarrow +\infty} \left\| N^+(\bar{\theta}^{+, (k)}, \cdot, \cdot) - N^-(\bar{\theta}^{-, (k)}, \cdot, \cdot) \right\|_{\infty, \bar{\Omega}_\varepsilon^+ \cap \bar{\Omega}_\varepsilon^-} \Big\|_{L^2(0, T)} = 0, \quad (10)$$

where $\|\cdot\|_{\infty, \Omega_\varepsilon^+ \cap \Omega_\varepsilon^-}$ denotes L^∞ -norm over the spatial domain $\Omega_\varepsilon^+ \cap \Omega_\varepsilon^-$. Moreover, the approximate solution ($\tilde{N}$) to the Schrödinger equation as is hence defined by:

$$\tilde{N} = \begin{cases} N^{+, (k^{cvg})}(\bar{\theta}^+, \cdot, \cdot), & \text{in } \Omega_\varepsilon^+ \times [0, T], \\ N^{-, (k^{cvg})}(\bar{\theta}^-, \cdot, \cdot), & \text{in } \Omega_\varepsilon^- \times [0, T], \end{cases}$$

where k^{cvg} denotes the number of Schwarz iterations to reach convergence and $\bar{\theta}^\pm = \bar{\theta}^{\pm, (k^{cvg})}$ denotes the corresponding converged set of parameters. The loss function is computed by evaluating the equation at a large number of randomly chosen input points $\{(x_j^\pm, t_n)\}_{j,n}$ in $\Omega_\varepsilon^\pm \times [0, T]$. From the optimization point of view, the method now requires the minimizing of two local loss functions. At each iteration k , we optimize the loss functions in $\Omega_\varepsilon^\pm$ with updated boundary conditions. Although standard SWR methods require the computation of IBVP “from scratch” (but with updated boundary conditions), PINN-SWR allow to initialize the local neural networks at iteration k , using the parameters $\bar{\theta}^{\pm, (k-1)}$ parameterizing to the space-time approximation solution $N^\pm(\bar{\theta}^{\pm, (k)}, \cdot, \cdot)$ in $\Omega_\varepsilon^\pm$. That is at Schwarz iteration k and for $\ell \geq 0$,

$$\theta_{\ell+1}^{\pm, (k)} = \theta_\ell^{\pm, (k)} - \nu \nabla \mathcal{L}^\pm(\theta_\ell^{\pm, (k)}),$$

where $\theta_0^{\pm, (k)} = \bar{\theta}^{\pm, (k-1)}$, ℓ denotes the optimization algorithm iteration index, and ν denotes the *learning rate*. In other words, we expect the optimization algorithm accelerates the convergence at least close to SWR where $\mathcal{L}(\bar{\theta}^{\pm, (k-1)}) \approx \mathcal{L}(\bar{\theta}^{\pm, (k)})$. By analogy, the use of standard TDSE solvers would correspond to randomly choose the initial parameters $\theta_0^{\pm, (k)} = \theta_{\text{random}}^{\pm, (k)}$ and with $\mathcal{L}(\bar{\theta}^{\pm, (k-1)}) \ll \mathcal{L}(\theta_{\text{random}}^{\pm, (k)})$.

Remark 2.1. By rewriting SWR algorithms as a fixed-point problem and employing microlocal analysis arguments from [5,13], the following error estimate can be directly established:

$$\|u^{\pm, (k)} - u_{|\Omega_\varepsilon^\pm}\|_{L^2(\Omega_\varepsilon^\pm \times [0, T])} \leq C(\tau, \varepsilon, V)^k \|u^{\pm, (0)} - u_{|\Omega_\varepsilon^\pm}\|_{L^2(\Omega_\varepsilon^\pm \times [0, T])},$$

where $C(\tau, \varepsilon, V)$ is dependent on the type of transmission conditions with $1 > C_{\text{CSWR}} > C_{\text{RSWR}} > C_{\text{p-OSWR}}$ and where CSWR refers to classical SWR based on Dirichlet transmission conditions, RSWR refers to (optimized) Robin-SWR based on Robin transmission conditions, and p-OSWR refers to optimal (or p-optimal) SWR methods. Rather than computing (8), it may be relevant to consider the following system

- For $k = 1$, we consider (3) with $u^{\pm, (0)}$ given.
- For $k \geq 2$, we set $w^{\pm, (1)} = u^{\pm, (1)} - u^{\pm, (0)}$ and consider

$$\begin{aligned} i\partial_t w^{\pm, (k)} &= -\partial_{xx} w^{\pm, (k)} - V(x)w^{\pm, (k)}, & \text{on } \Omega_\varepsilon^\pm \times [0, T], \\ w^{\pm, (k)}(\cdot, 0) &= w^{\pm, (k-1)}(\cdot, 0), & \text{in } \Omega_\varepsilon^\pm, \\ \mathcal{T}_\pm(x, t)w^{\pm, (k)} &= \mathcal{T}_\pm(x, t)w^{\mp, (k-1)}, & \text{on } \Gamma_\varepsilon^\pm \times [0, T], \\ w^{\pm, (k)} &= 0, & \text{on } \Lambda_\varepsilon^\pm \times [0, T]. \end{aligned} \quad (11)$$

Practically, we can then construct neural networks $N^\pm : (x, t) \mapsto N^\pm(\theta^{\pm, (k)}, x, t)$ which are eventually convergent to the null function. From a PINN point of view, $N^\pm(\theta^{\pm, (k)}, \cdot, \cdot)$ is an approximation of $w^{\pm, (k)}$ at Schwarz iteration k . We denote by $\bar{\theta}^{\pm, (k^{cvg})}$ the parameters corresponding to the converged SWR solution. Adapting the analysis of [7], we obtain the following stability result for k large enough and for some positive $D > 0$:

$$\|\bar{\theta}^{\pm, (k^{cvg})} - \bar{\theta}^{\pm, (k)}\|_2 \lesssim D |N^\pm(\bar{\theta}^{\pm, (k)}, \cdot, \cdot)|_{L^\infty(\Omega_\varepsilon^\pm \times [0, T])},$$

where $N^\pm(\bar{\theta}^{\pm, (k)}, \cdot, \cdot)$ is close to the null function and $|u|_{L^\infty(\Omega_\varepsilon^\pm \times [0, T])} = \sup_{x \in \Omega_\varepsilon^\pm, t \in [0, T]} |u(x, t)|$.

2.3. Multi-dimensional PINN-SWR algorithm

In this paper, the SWR-learning concepts are proposed in a one-dimensional framework. However, the multi-dimensional extension of the algorithms and concepts remains valid. Hereafter, we present some details about the extension to d spatial dimensions. The neural networks read $N^\pm(\theta, \mathbf{x}, t)$, for $\mathbf{x} \in \Omega_\varepsilon^\pm \subset \mathbb{R}^d$ with smooth boundary and such that $\Omega = \Omega_\varepsilon^- \cup \Omega_\varepsilon^+$. Hence considering the IVBP

$$\begin{aligned} i\partial_t u + \Delta u + V(\mathbf{x})u &= 0, & \text{in } \Omega \times [0, T], \\ Mu &= 0, & \text{in } \Gamma \times [0, T], \\ u(\cdot, 0) &= u_0, & \text{in } \Omega, \end{aligned} \quad (12)$$

the corresponding SWR at Schwarz iteration k , reads

$$\begin{aligned} i\partial_t N^{\pm, (k)} &= -\Delta N^{\pm, (k)} - V(\mathbf{x})N^{\pm, (k)}, & \text{in } \Omega_\varepsilon^\pm \times [0, T], \\ N^{\pm, (k)}(\cdot, 0) &= u_0^\pm, & \text{in } \Omega_\varepsilon^\pm, \\ \mathcal{T}_\pm(\mathbf{x}, t)N^{\pm, (k)} &= \mathcal{T}_\pm(\mathbf{x}, t)N^{\mp, (k-1)}, & \text{on } \Gamma_\varepsilon^\pm \times [0, T], \\ N^{\pm, (k)} &= 0, & \text{on } \Lambda_\varepsilon^\pm \times [0, T], \end{aligned} \quad (13)$$

where $\mathcal{T}_\pm(\mathbf{x}, t)$ is a boundary operator. For $d = 2$ and null potential, we have for instance:

- Assuming that $\Gamma_\varepsilon^\pm$ are smooth interfaces with outward normal vectors $\mathbf{n}^\pm$. Quasi-optimal transmission operators typically read for

$$\begin{aligned} 0\text{-optimal (Robin)} & \partial_{\mathbf{n}^\pm} + i\tau, & \text{on } \Gamma_\varepsilon^\pm, \\ 1\text{-optimal} & \partial_{\mathbf{n}^\pm} + e^{-i\pi/4} \partial_t^{1/2}, & \text{on } \Gamma_\varepsilon^\pm, \\ 2\text{-optimal} & \partial_{\mathbf{n}^\pm} + e^{-i\pi/4} \partial_t^{1/2} - e^{i\pi/4} \frac{1}{2} \Delta_{\Gamma_\varepsilon^\pm} I_t^{1/2}, & \text{on } \Gamma_\varepsilon^\pm, \end{aligned}$$

where $\Delta_{\Gamma_\varepsilon^\pm}$ is the second-order derivative (Laplace-Beltrami) operator over $\Gamma_\varepsilon^\pm$ and I_t^α

$$I_t^\alpha f(t) = \int_0^t f(\tau) (\pi(t-\tau))^{-\alpha} d\tau.$$

See [26] for the details.

- In the case of curved interface

$$\begin{aligned} \partial_{\mathbf{n}^\pm} + e^{-i\pi/4} \partial_t^{1/2} &+ \frac{\kappa(s)}{2} - e^{i\pi/4} \left(\frac{\kappa(s)^2}{8} + \frac{1}{2} \Delta_{\Gamma_\varepsilon^\pm} \right) I_t^{1/2} \\ &+ i \left(\frac{\kappa(s)^3}{8} + \frac{1}{2} \partial_s (\kappa(s) \partial_s) + \frac{\Delta_{\Gamma_\varepsilon^\pm} \kappa(s)}{8} \right) I_t, & \text{on } \Gamma_\varepsilon^\pm, \end{aligned}$$

where we have denoted κ the local curvature, and where s is the curvilinear abscissa along $\Gamma_\varepsilon^\pm$.

The derivation and implementation of PINN algorithms in higher spatial dimension is straightforward. At Schwarz iteration k , we hence minimize the following local loss functions

$$\begin{aligned} \mathcal{L}^\pm(\theta^{\pm, (k)}) &= \lambda^\pm \left\| i\partial_t N^{\pm, (k)}(\theta^{\pm, (k)}, \cdot, \cdot) + \Delta N^{\pm, (k)}(\theta^{\pm, (k)}, \cdot, \cdot) \right. \\ &\quad \left. + V(\cdot) N^{\pm, (k)}(\theta^{\pm, (k)}, \cdot, \cdot) \right\|_{L^2(\Omega_\varepsilon^\pm \times [0, T])} \\ &\quad + \mu^\pm \left\| \mathcal{T}_\pm(\cdot, t) N^{\pm, (k)}(\theta^{\pm, (k)}, \cdot, \cdot) \right. \\ &\quad \left. - \mathcal{T}_\pm(\cdot, t) N^{\mp, (k-1)}(\bar{\theta}^\mp, \cdot, \cdot) \right\|_{L^2(\Gamma_\varepsilon^\pm \times [0, T])} \\ &\quad + \kappa^\pm \left\| N^{\pm, (k)}(\theta^{\pm, (k)}, \cdot, \cdot) \right\|_{L^2(\Lambda_\varepsilon^\pm \times [0, T])} \\ &\quad + \zeta^\pm \left\| N^{\pm, (k)}(\theta^{\pm, (k)}, 0, \cdot) - u_0^\pm(\cdot) \right\|_{L^2(\Omega_\varepsilon^\pm)}, \end{aligned}$$

where $\bar{\theta}^\pm$ are the optimized parameters defining the approximate local space-time solutions in $\Omega_\varepsilon^\pm$ at iteration $k-1$ and where $\lambda^\pm, \mu^\pm$ and $\kappa^\pm$ are some positive parameters.

3. Numerical schemes

In this section, we provide some details about the schemes used to solve the TDSE using SWR-DDM. We will perform several tests and comparisons including:

- Comparison of Classical/Robin-SWR with Optimal SWR in terms of number of Schwarz iterations using finite difference (FD) solvers and physics-informed neural network (PINN) solvers. We do not expect any improvement in terms of Schwarz iterations when using PINN over FD for the same type of transmission conditions. However, we will illustrate that Optimal SWR has a much faster convergence rate than CSWR/Robin-SWR, as previously proven and observed in [13].
- Comparison of the normalized CPU time for CSWR/Robin-SWR/Optimal-SWR convergence with FD and PINN, as a function of Schwarz iterations.
- Comparison of loss values for Robin-SWR and Optimal SWR using PINN, where at each Schwarz iteration k , (i) the parameters are initialized with the converged parameters at iteration $k-1$ (SWR-learning), or (ii) the parameters are randomly initialized (no learning).

3.1. Finite difference set-up

We consider two bounded subdomains $\Omega_{a,\varepsilon}^+ = (-a, b + \varepsilon/2)$, $\Omega_{a,\varepsilon}^- = (b - \varepsilon/2, a)$, $a \in \mathbb{R}_+^*$ and with $\varepsilon > 0$ a (small) parameter characterizing the overlapping region $\Gamma_{a,\varepsilon} = \Omega_{a,\varepsilon}^+ \cap \Omega_{a,\varepsilon}^- = (b - \varepsilon/2, b + \varepsilon/2)$, and $\Omega_a = \Omega_{a,\varepsilon}^+ \cup \Omega_{a,\varepsilon}^- = (-a, a)$. The interfaces are located at $b \pm \varepsilon/2$. The Crank-Nicolson scheme which is used here is fully described in [13]. Denoting $u^{\pm,n,(k)}$ the approximate solution in $\Omega_{a,\varepsilon}^\pm$ at Schwarz iteration k and time iteration n , the convergence criterion for the Schwarz DDM is given by

$$\left\| u_{|\Gamma_\varepsilon}^{+,n_T,(k)} - u_{|\Gamma_\varepsilon}^{-,n_T,(k)} \right\|_{\infty,\Gamma_\varepsilon} \leq \delta^{\text{Sc}}, \quad (14)$$

with $\delta^{\text{Sc}} = 10^{-14}$ ("Sc" for Schwarz) and where $n_T = T/\Delta t$. When the convergence of the full iterative algorithm is obtained at Schwarz iteration k^{cvg} , one gets the converged global solution $u^{\text{cvg}} := u^{(k^{\text{cvg}})}$ in Ω_a .

CSWR algorithm. The CSWR method is based on Dirichlet transmission conditions implemented as follows. At $x_{N+}^+ = b + \varepsilon/2$, we impose $u_{N+}^{+,n+1,(k)} + u_{N+}^{+,n,(k)} = u_{j_0}^{-,n+1,(k-1)} + u_{j_0}^{-,n,(k-1)}$, and j_0 denotes the number of overlapping nodes, i.e. $\varepsilon = (j_0 - 1)\Delta x$. At $x_1^- = b - \varepsilon/2$, we fix $u_1^{-,n+1,(k)} + u_1^{-,n,(k)} = u_{N+}^{+,n+1,(k-1)} + u_{N+}^{+,n,(k-1)}$. Finally at $x_1^+ = -a$ and $x_{N-}^- = a$, we set null Dirichlet boundary conditions.

Robin-SWR algorithm. The Robin-SWR method is based on Robin transmission conditions $\partial_x \pm ir$ (with non-null constant r). Say at $x_{N+}^+ = b + \varepsilon/2$, we impose $(u_{N+}^{+,n+1,(k)} + u_{N+}^{+,n,(k)} - u_{N+}^{+,n+1,(k-1)} - u_{N+}^{+,n,(k-1)}) + \Delta x r (u_{N+}^{+,n+1,(k)} + u_{N+}^{+,n,(k)}) = (u_{j_0+1}^{-,n+1,(k-1)} + u_{j_0+1}^{-,n,(k-1)} - u_{j_0}^{-,n+1,(k-1)} - u_{j_0}^{-,n,(k-1)}) + \Delta x r (u_{j_0+1}^{-,n+1,(k-1)} + u_{j_0+1}^{-,n,(k-1)})$, and j_0 denotes the number of overlapping nodes, i.e. $\varepsilon = (j_0 - 1)\Delta x$.

Standard p-OSWR solver. The chosen discretization of the nonlocal time Riemann-Liouville operator was derived from [3], and allows in 1-d, for unconditional stability in the framework of TDSE discretization. We refer to [13] for a full description of the algorithm.

3.2. On the PINN-SWR approximation

Let us provide some details about the discretization of the PINN-SWR algorithm (8) in the case of DtN transmission conditions (quasi-optimal or optimal SWR methods). The local loss functions are numerically evaluated by Monte Carlo integration using $(x_j, t_n)_{j,j}$, randomly chosen points in space-time. It is important to note that the non-locality

(in time) of the DtN operator is not a concern in the PINN approach. At each Schwarz iteration, $\theta^{(k)}$ parameterizes the complete space-time solution. In contrast to standard PDE solvers, which require storing the solution at the boundary at each previous iteration, this information is contained in the converged parameters $\bar{\theta}^{(k-1)}$. The transmission conditions in quasi-optimal SWR methods involve the fractional derivative $\partial_t^{1/2}$

$$\partial_t^{1/2} N(\theta, x, t) = \partial_t \left(N(\theta, x, t) * \frac{1}{\sqrt{\pi t}} \right) = \partial_t \int_0^t \frac{N(\theta, x, \tau)}{\sqrt{\pi(t-\tau)}} d\tau,$$

for any θ and x . The fractional derivative can be estimated using the convolution product of neural networks and automatic differentiation. This is particularly convenient as it is well-known that accurate and efficient approximation of fractional derivatives is hard to achieve, especially in a FPDE framework. The transmission operator at $(\pm\varepsilon/2, t)$ applied to a neural network N reads

$$\partial_t^{1/2} N^{\pm,(k)} \pm e^{i\pi/4} \partial_x N^{\pm,(k)} = \partial_t^{1/2} N^{\mp,(k-1)} \pm e^{i\pi/4} \partial_x N^{\mp,(k-1)},$$

Automatic integration is still at an early stage of investigation, so that the approximation of $\partial_t^{1/2}$ is here preferred; see [27,28] for instance. Introducing discrete times $0 < t_1 < \dots < t_n$, we define the following quadrature, at $t = t_n$ for any u with enough regularity

$$\tilde{\partial}_t^{1/2} u(t_n) = u(t_n) + \sum_{i=1}^n \Delta t_i w_i^{(\alpha)} u(t_{n-i}), \quad (15)$$

where the weights read

$$w_i^{(\alpha)} = \sum_{l=1}^i \frac{\Gamma(l-\alpha)}{\Gamma(-\alpha)\Gamma(l+1)},$$

and where $\Delta t_i = t_i - t_{i-1}$. Higher order quadratures are discussed in [27, 28]. In order to include the contribution of the transmission conditions in the loss function, we then proceed as follows.

- Within the PINN framework, the discrete times $\{t_n\}_n$ are randomly selected.
- At any $(\pm\varepsilon/2, t_n)$, we impose

$$\begin{aligned} \tilde{\partial}_t^{1/2} N(\theta, \pm\varepsilon/2, t_n) \pm e^{i\pi/4} \partial_x N(\theta, \pm\varepsilon/2, t_n) \\ = \pm e^{i\pi/4} \partial_x N(\theta, \pm\varepsilon/2, t_n) + N(\theta, \pm\varepsilon/2, t_n) \\ + \sum_{i=1}^n \Delta t_i w_i^{(\alpha)} N(\theta, \pm\varepsilon/2, t_n). \end{aligned}$$

The above condition does not require any additional storage, as all the corresponding information is all encoded in θ .

- Finally the transmission condition which is implemented at $(\theta, \pm\varepsilon/2, t)$ reads

$$\tilde{\partial}_t^{1/2} N^{\pm,(k)} \pm e^{i\pi/4} \partial_x N^{\pm,(k)} = \tilde{\partial}_t^{1/2} N^{\mp,(k-1)} \pm e^{i\pi/4} \partial_x N^{\mp,(k-1)},$$

when CSWR (resp. RSWR) simply involves the transmission conditions $N^{\pm,(k)} = N^{\mp,(k-1)}$ (resp. $rN^{\pm,(k)} + \partial_x N^{\pm,(k)} = rN^{\mp,(k-1)} + \partial_x N^{\mp,(k-1)}$, for some constant r).

Remark 3.1. Dirichlet-to-Neumann boundary conditions are widely used to prevent artificial reflections in various types of wave equations (such as Maxwell, wave, Schrödinger, Dirac, etc.) on complex domains, as discussed in [26]. As mentioned earlier, their stable, accurate, and efficient approximation can however be challenging. It is interesting to notice that practically, the fast convergence of optimal or quasi-optimal SWR methods does not necessarily require a highly accurate approximation of the transmission operators. For example, for the Schrödinger operator, the Robin operator can be viewed as an approximation of the DtN operator, where the fractional operator in time is approximated by a constant algebraic operator, and still provides rapid convergence of the corresponding SWR algorithm (then referred to as Robin-SWR).

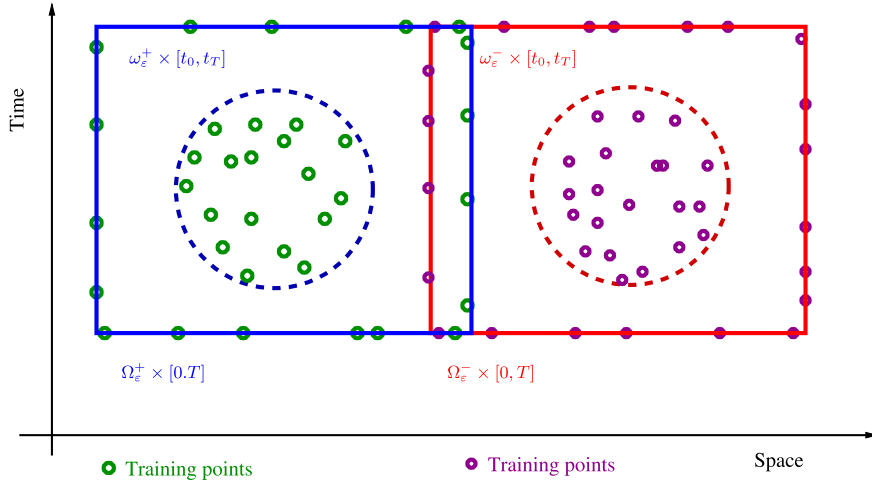


Fig. 1. Second level of learning.

In future works, we will specifically explore transparent/high-order absorbing boundary conditions using PINN algorithms in detail.

In order to improve the efficiency of the overall algorithm on $(\Omega_\varepsilon^\pm \cup \Gamma_\varepsilon^\pm) \times [0, T]$, a second level of (traditional) learning involves restricting the training zone to $(\omega_\varepsilon^\pm \cup \Gamma_\varepsilon^\pm) \times [t_0, t_T]$, where $\omega_\varepsilon^\pm \subset \Omega_\varepsilon^\pm$ and $0 < t_0 < t_T < T$. This relies on the trained networks to approximate the local solutions in $\Omega_\varepsilon^\pm \setminus \omega_\varepsilon^\pm \times ([0, t_0] \cup [t_T, T])$, as illustrated in Fig. 1.

3.3. Practical implementation

Usual neural network libraries, like `jax`, which is utilized in this paper, are designed to handle real-valued neural networks. It is noteworthy, however, that `pytorch`, starting from version 1.7, possesses the capability to directly manage complex neural networks. Below, we rewrite the Schrödinger equation as a real 2-equation system on the real and imaginary part of the approximate complex wavefunction $N = N_R + iN_I$. Denoting $\overline{N}_{R,I}^{\pm,(k-1)} = N_{R,I}^{\pm,(k-1)}(\overline{\theta}^{\pm,(k-1)}, \cdot, \cdot)$, the (quasi)-optimal SWR method involving the DtN-like transmission condition $\partial_t^{1/2} \pm e^{\pm\pi/4} \partial_x$ reads as follows:

$$\begin{aligned} \partial_t N_R^{\pm,(k)} &= -\partial_{xx} N_I^{\pm,(k)} - V(x) N_I^{\pm,(k)}, \text{ in } \Omega_\varepsilon^\pm \times [0, T], \\ \partial_t N_I^{\pm,(k)} &= \partial_{xx} N_R^{\pm,(k)} - V(x) N_R^{\pm,(k)}, \text{ in } \Omega_\varepsilon^\pm \times [0, T], \\ N_R^{\pm,(k)}(\cdot, 0) &= \text{Re}(u_0^\pm), \text{ in } \Omega_\varepsilon^\pm, \\ N_I^{\pm,(k)}(\cdot, 0) &= \text{Im}(u_0^\pm), \text{ in } \Omega_\varepsilon^\pm, \\ \partial_t^{1/2} N_R^{\pm,(k)} &\pm \frac{1}{\sqrt{2}} \partial_x (N_R^{\pm,(k)} - N_I^{\pm,(k)}) \\ &= \partial_t^{1/2} \overline{N}_R^{\mp,(k-1)} \pm \frac{1}{\sqrt{2}} \partial_x (\overline{N}_R^{\mp,(k-1)} - \overline{N}_I^{\mp,(k-1)}), \text{ on } \Gamma_\varepsilon^\pm \times [0, T], \quad (16) \\ \partial_t^{1/2} N_I^{\pm,(k)} &\pm \frac{1}{\sqrt{2}} \partial_x (N_R^{\pm,(k)} + N_I^{\pm,(k)}) \\ &= \partial_t^{1/2} \overline{N}_I^{\mp,(k-1)} \pm \frac{1}{\sqrt{2}} \partial_x (\overline{N}_R^{\mp,(k-1)} + \overline{N}_I^{\mp,(k-1)}), \text{ on } \Gamma_\varepsilon^\pm \times [0, T], \\ N_R^{\pm,(k)} &= 0, \text{ on } \Lambda_\varepsilon^\pm \times [0, T], \\ N_I^{\pm,(k)} &= 0, \text{ on } \Lambda_\varepsilon^\pm \times [0, T]. \end{aligned}$$

Notice that N_R and N_I can be taken with the same set of parameters (or not). The local loss functions (assuming for simplicity that the initial condition is encoded in the neural network) which are defined at Schwarz iteration k , and at any $(\pm\varepsilon/2, t)$ read

$$\mathcal{L}^\pm(\theta^{(k)}) = \lambda^\pm \left\| \partial_t N_R^{\pm,(k)} + \partial_{xx} N_I^{\pm,(k)} + V(x) N_I^{\pm,(k)} \right\|_{L^2(\Omega_\varepsilon^\pm \times [0, T])}$$

$$\begin{aligned} &+ \lambda^\pm \left\| \partial_t N_I^{\pm,(k)} + \partial_{xx} N_R^{\pm,(k)} + V(x) N_R^{\pm,(k)} \right\|_{L^2(\Omega_\varepsilon^\pm \times [0, T])} \\ &+ \mu^\pm \left\| \partial_t^{1/2} N_R^{\pm,(k)} \pm \frac{1}{\sqrt{2}} \partial_x (N_R^{\pm,(k)} - N_I^{\pm,(k)}) \right. \\ &\quad \left. - \partial_t^{1/2} \overline{N}_R^{\mp,(k-1)} \mp \frac{1}{\sqrt{2}} \partial_x (\overline{N}_R^{\mp,(k-1)} - \overline{N}_I^{\mp,(k-1)}) \right\|_{L^2(\Gamma_\varepsilon^\pm \times [0, T])} \\ &+ \mu^\pm \left\| \partial_t^{1/2} N_I^{\pm,(k)} \pm \frac{1}{\sqrt{2}} \partial_x (N_R^{\pm,(k)} + N_I^{\pm,(k)}) \right. \\ &\quad \left. - \partial_t^{1/2} \overline{N}_I^{\mp,(k-1)} \mp \frac{1}{\sqrt{2}} \partial_x (\overline{N}_R^{\mp,(k-1)} + \overline{N}_I^{\mp,(k-1)}) \right\|_{L^2(\Gamma_\varepsilon^\pm \times [0, T])} \\ &+ \kappa^\pm \left\| N_R^{\pm,(k)} \right\|_{L^2(\Lambda_\varepsilon^\pm \times [0, T])} + \kappa^\pm \left\| N_I^{\pm,(k)} \right\|_{L^2(\Lambda_\varepsilon^\pm \times [0, T])}. \end{aligned}$$

3.4. Computational complexity

Let us discuss the computational complexity of SWR methods when using RSM (typically finite element or finite difference) from one hand, and PINN algorithm on the other hand. In this context, we denote by $n_x^\pm$ the number of degrees of freedom in each subdomain $\Omega_\varepsilon^\pm$ for RSM (and the number of spatial training points for PINN), and n_t as the number of time steps for RSM (and the number of training points in-time for PINN). Additionally, we denote by $k_{\text{SWR}}^{\text{cvg}}$ the number of Schwarz iterations required to reach a given tolerance δ^{cv} with a SWR algorithm. Finally, $n_\theta^\pm$ represents the number of neural network parameters to optimize in the PINN algorithms. The computational complexities for SWR+PINN and SWR+RSM methods read

$$C_{\text{PINN+SWR}} = O(n_\theta^\pm n_t n_x^\pm \sum_{k=1}^{k_{\text{SWR}}^{\text{cvg}}} p_{\text{SWR}}^\pm(k)),$$

where $p_{\text{SWR}}^\pm(k) (\rightarrow_{k \rightarrow +\infty} 0$ and coming from the acceleration of the optimization algorithm within SWR iterations) denotes the number of gradient descent iterations at Schwarz iteration k . Moreover,

$$C_{\text{RSM+SWR}} = O(k_{\text{SWR}}^{\text{cvg}} n_t (n_x^\pm)^{\alpha_{\text{SWR}}}),$$

for some $\alpha_{\text{SWR}} > 1$, depending on the sparsity and structure of the linear systems involved in implicit RSM. The coefficient α_{SWR} also depends on the type of transmission conditions, where typically $\alpha_{\text{p-OSWR}} > \alpha_{\text{RSM}} > \alpha_{\text{CSWR}}$. Additionally, for RSM DtN-like operators require large data storage, and it is well-known that using DtN-transmission conditions usually leads to stability issues (see [2–4]), necessitating smaller time steps than those for Dirichlet conditions (CSWR); or equivalently, $n_t^{(\text{CSWR})} < n_t^{(\text{p-OSWR})}$. On the other hand, as recalled in Section 2, in

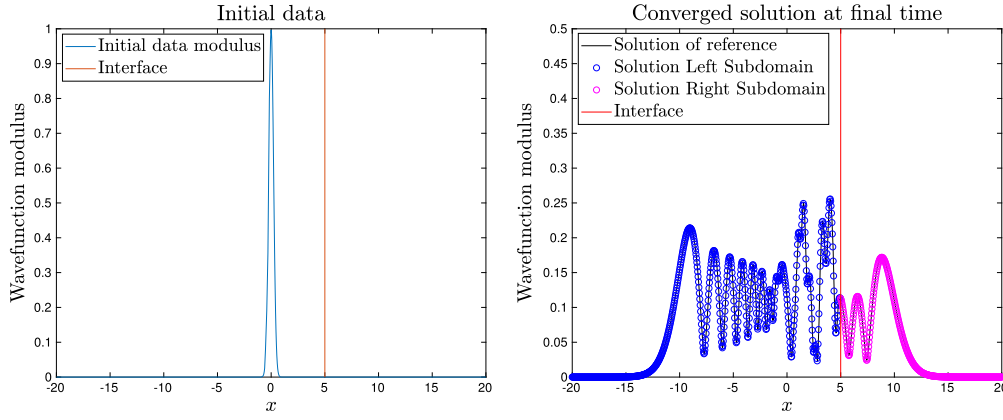


Fig. 2. Experiment 1. (Left) Initial data. (Right) Converged solution at final time.

general, we have $k_{\text{p-OSWR}}^{\text{cvg}} < k_{\text{RSM}}^{\text{cvg}} \ll k_{\text{CSWR}}^{\text{cvg}}$. In the PINN-framework, the minimization of the loss functions with CSWR and p-OSWR algorithms at each Schwarz iteration k respectively requires $p_{\text{CSWR}}(k)$ and $p_{\text{pOSWR}}(k)$ gradient descent iterations. Unlike RSM, in the framework of PINN algorithms, we do not predict an efficiency discrepancy due to stability or storage issues when using DtN compared to Dirichlet or Robin transmission conditions. Hence,

- We *can not state* that overall $C_{\text{RSM+pOSWR}} < C_{\text{RSM+CSWR}}$, although in general $C_{\text{RSM+RSM}} < C_{\text{RSM+RSM}} < C_{\text{RSM+pOSWR}}$; see [1,13,12].
- The above discussion however suggests that we *can* expect that $C_{\text{PINN+pOSWR}} < C_{\text{PINN+CSWR}}$ and $C_{\text{PINN+pOSWR}} < C_{\text{PINN+RSM}}$.

This latter point will then be numerically studied.

4. Numerics

This section is devoted to numerical experiments illustrating the above discussions. We first present preliminary tests using real space methods (RSM) then PINN algorithms. Hereafter, we shall use the notations from Section 3.

4.1. RSM-SWR numerical experiments

Experiment 1. The first test is devoted to a wave propagating in the direction of the subdomain interface, from initial time 0 to final time $T = 1$. We follow the experiment given in [13]. The numerical parameters for this test are the following: $a = 20$, $b = 5$, with $N^+ = 640$ and $N^- = 385$. The size of the overlapping zone is $\varepsilon = \Delta x$, corresponding to $j_0 = 2$. The time step is equal to $\Delta t = 0.1$. The initial data is given by $u_0(x) = \exp(-10x^2)$. We report in Fig. 2 (Left) the initial condition, and the converged solution at final time $T = 1$, Fig. 2 (Right). We compare on Fig. 3 (Left) the residual history $\{k, \|u_{\Gamma_\varepsilon}^{+,n_T,(k)} - u_{\Gamma_\varepsilon}^{-,n_T,(k)}\|_{\infty,\Gamma_\varepsilon}\|_{L^2(0,T)}\}$ of CSWR, Robin-SWR (with $r = 5$), and 1-optimal SWR. In Fig. 3 (Middle), we report the CPU-time as a function of Schwarz iteration. In order to provide a fair comparison, we also propose to normalize the CPU-time as a function of Schwarz iteration / CPU for the 100th iteration (the first iterations require additional basic computations) and report $t_{\text{SWR}}^{(k)}$ as a function of k , for $k \geq 2$:

$$t_{\text{SWR}}^{(k)} \leftarrow \frac{t_{\text{SWR}}^{(k)}}{t_{\text{SWR}}^{(100)}}.$$

This example shows in particular that the cost of the local IBVP computations remains roughly constant as a function of the Schwarz iteration, at least in the asymptotic regime of convergence [12,5].

4.2. PINN-SWR numerical experiments

In the following series of experiments, we consider

$$i\partial_t u + \frac{1}{2}\Delta u + V(x)u = 0,$$

where V is a given smooth potential. We solve this equation using SWR algorithms on $\Omega_\varepsilon^+ = (-a, \varepsilon/2)$ and $\Omega_\varepsilon^- = (-\varepsilon/2, a)$ and the time interval $(0, T)$ with null Dirichlet conditions $\pm a$. A PINN algorithm is implemented on each subdomain allowing the optimization at each Schwarz iteration k , of the parameters $\theta^{\pm,(k)}$. The initial condition is a Gaussian function

$$u_0(x) = \exp(-\alpha(x - x_0)^2 + ik_0 x),$$

with $k_0 = 5$, $\alpha = 25$ and $x_0 = -7/8$ and in the following tests we take $a = 3/2$. We are specifically interested in:

- illustrating the acceleration process which is introduced in the PINN algorithm by initializing at Schwarz iteration k , the local loss functions $\mathcal{L}^\pm$ with $\bar{\theta}^{\pm,(k-1)}$. In other words, at iteration k the parameters $(\theta_{\ell=0}^{\pm,(k)})$ are initialized with the converged sets of parameters (in the optimization sense) at the previous Schwarz iteration: $\theta_0^{\pm,(k)} = \bar{\theta}^{\pm,(k-1)}$ allowing to define approximate solutions to (3). We then compare the convergence of the loss functions as a function of Schwarz and optimization iterations with randomly chosen parameters $\theta_0^{\pm,(k)}$.
- Comparison of the rate of convergence as well as the computational efficiency of PINN-SWR algorithm with different types of transmission conditions.

In the following experiment the local neural networks are constituted by 3 hidden layers with 5 neurons each. The number of training points in space and time is set by default to 25×25 . Practically, the 2 subdomains are respectively defined by $(-a, \Delta x)$ and $(-\Delta x, a)$ with $\Delta x = a/25$.

Experiment 2a. This experiment is dedicated to the illustration of PINN-SWR convergence on the space-time domain $(-7/4, 7/4) \times (0, 1/4)$. We present the modulus of the reconstructed wavefunction after one Schwarz iteration in Fig. 4. Specifically, in the right subdomain, the global solution is nearly “null”. Fig. 5 displays the modulus of the reconstructed wavefunction at Schwarz convergence across the entire space-time domain. For improved visualization, we also include the reconstructed solution on a truncated domain in space-time.

Experiment 2b. We here compare the rate of convergence of PINN-SWR algorithms using Robin-SWR ($\partial_x \pm ir$) and Optimal-SWR algorithms with transmission operator $\partial_t^{1/2} \pm e^{i\pi/4} \partial_x / \sqrt{2}$. The Robin constant is fixed to $r = 10$. The Robin constant can be optimized to achieve

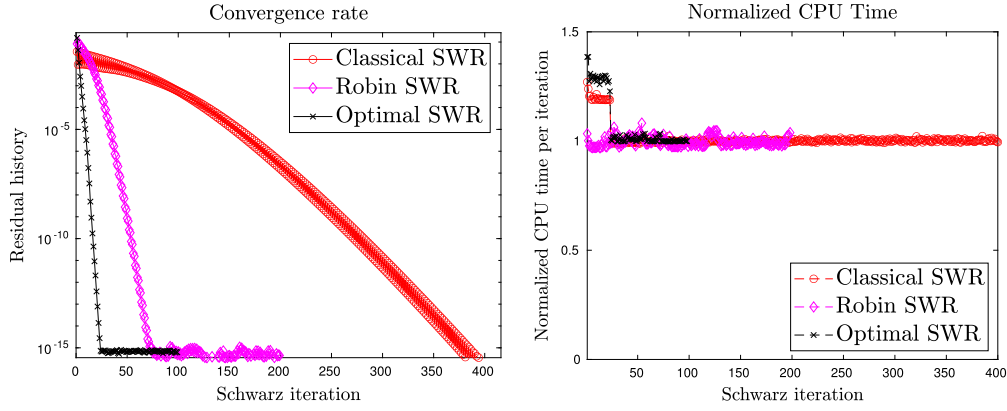


Fig. 3. Experiment 1. (Left) Residual history for Quasi-optimal SWR, Robin-SWR and Classical SWR methods. (Right) Relative CPU time as function of Schwarz iteration.

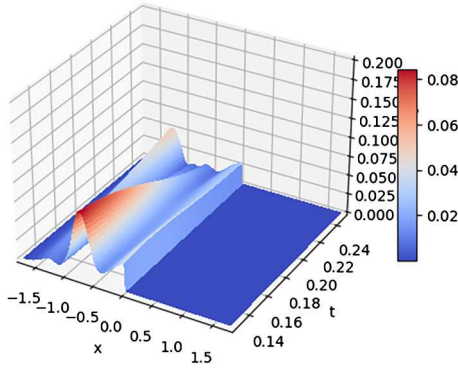


Fig. 4. Experiment 2a. (Left) Loss function as a function of Schwarz iterations. (Right) Reconstructed wavefunction after one Schwarz iteration.

the fastest possible convergence. This optimization can be accomplished using analytical arguments or by dynamically adapting its values to enhance the convergence rate from one Schwarz iteration to the next. In this scenario, the method is referred to as optimized Schwarz. Learning the optimized constant within the combined PINN-SWR algorithm could also be investigated, but it is not in the scope of this study. We report in Fig. 6 (Left), the residual history as a function of the Schwarz iterations in subdomain $\Omega_\varepsilon^\pm$:

$$\left\{ (k, \left\| N^{\pm(k)}(\theta_{\pm}^k, \pm\varepsilon/2, \cdot) - N^{\mp(k)}(\theta_{\mp}^k, \pm\varepsilon/2, \cdot) \right\|_{L^\infty(0,T)}), k \geq 0 \right\}.$$

The transmission is encoded within the loss function and is hence satisfied up to i) a pre-defined optimization error, as well as ii) a quadrature error for DtN-based transmission conditions. To illustrate the convergence of SWR algorithms traditionally, the norm in space and time of the difference of the local solutions on the overlapping zones/interfaces is reported. In order to illustrate the acceleration of the optimization algorithm and to compare the efficiency of the Optimal-SWR and Robin-SWR (RSWR) methods, such a criterion is no longer appropriate. The key point to report is the increasingly faster convergence of the optimization algorithm within the SWR process. The chosen criterion is the evaluation of the loss function for a fixed number of gradient descent iterations as a function of Schwarz iterations. We expect a decrease in the overall local loss functions along the Schwarz iterations, with naturally smaller values for Optimal-SWR than for Robin-SWR. In Fig. 7 (resp. Fig. 8), we report the loss function values as function of k in the Domains $\Omega_\varepsilon^\pm$, when the neural network parameters are “learnt” or randomly chosen in the case of RSRW (resp. OSWR) algorithm, after a fixed number N_0 of iterations of the optimization algorithm, that is we report:

$$\left\{ (k, \mathcal{L}_{SWR}^\pm(\theta_{N_0}^{\pm(k)})), k \geq 1 \right\},$$

with and without “learning”. We notice that when the optimization parameters are randomly (resp. learnt from previous Schwarz iteration) chosen, the value of the loss function at $\ell = N_0$ is overall not decreasing (resp. decreasing) as a function of k . Let us mention that it is naturally possible to reduce the loss function values on those tests by increasing N_0 , the number of iterations of the optimization algorithm. As expected and theoretically proven in [13,29], the convergence of optimal SWR is much faster than Robin-SWR, and observed using standard PDE solvers [13]. As discussed above, Dirichlet-to-Neumann transmission operators usually deteriorate the stability, efficiency and sometimes accuracy of SWR computational solvers. Thanks to automatic differentiation PINN does not (directly) introduce stability issues related to the time/space discretization. For completeness, we also present the loss function as a function of the optimization iteration after three Schwarz iterations ($k = 3$) in Ω_ε^+ with and without acceleration in Fig. 9 (Left). To illustrate the acceleration of the optimization algorithm along the Schwarz process, we additionally provide the normalized loss values as a function of the optimization iteration for different Schwarz iterations k ($k = 1, 3, 10, 25, 50$) without learning (Fig. 9, Middle) and with learning (Fig. 9, Right). Specifically, we represent the graph in a semi-logarithmic scale, normalizing it by $\mathcal{L}^+(\theta^{+(1)})$ for various values of k :

$$\left\{ (\ell, \mathcal{L}(\theta_\ell^{+(k)})/\mathcal{L}(\theta_\ell^{+(1)})), \ell \geq 1 \right\}, \quad (17)$$

with and without “learning”. As expected, initializing the optimization algorithm at iteration $k > 1$ using the converged set of parameters from Schwarz iteration $k - 1$ allows for the acceleration of the optimization algorithm. We notice that the larger k , the smaller the ratio (17). This test hence illustrates the acceleration of the convergence thanks to the learning provided by PINN algorithms.

The next important question to address, is the performance of the optimal vs Robin (or classical) SWR algorithm within the PINN framework. We report in Fig. 10 (Left), the CPU-time for minimizing a local loss functions with a fixed number of epochs, and within one Schwarz iteration and different values of training data in time (from 12 to 150). The number of training data in space is fixed to 25. We also report in Fig. 10 (Right), the relative cost Optimal-SWR vs Robin-SWR $(t_{\text{OptSWR}} - t_{\text{RSWR}})/t_{\text{OptSWR}}$. We observe the computational complexity of both PINN-SWR methods is linear. As expected, optimal-SWR computational complexity is shown to be higher than Robin-SWR, but the difference is actually relatively small in particular for a large number of training points in time; the slight additional computational cost is negligible in comparison with the gain in terms of convergence acceleration of the SWR algorithm. Overall, the PINN-Optimal SWR is then expected to be more efficient than PINN-RSWR or PINN-CSWR. Another illustration of this acceleration is also illustrated in the next experiment.

Experiment 3. We propose to compare the convergence of PINN-SWR algorithm in terms of the optimization algorithm, with and without

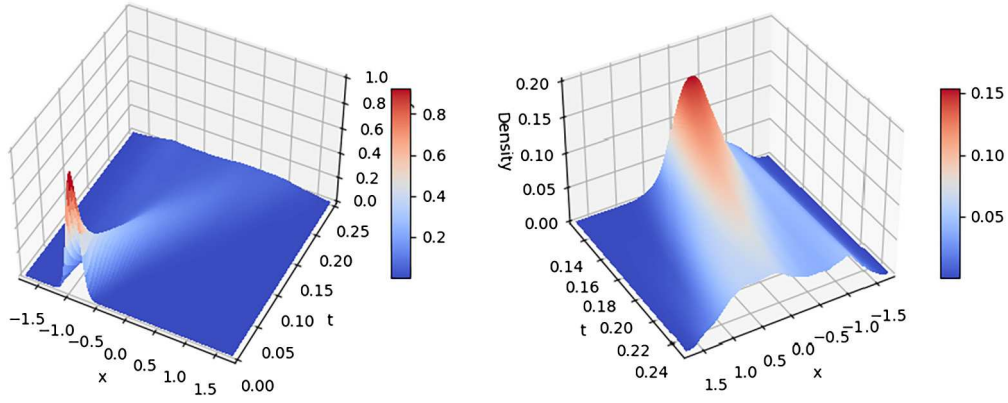


Fig. 5. Experiment 2a. Reconstructed wavefunction. (Left) Global space-time domain. (Right) Truncated domain.

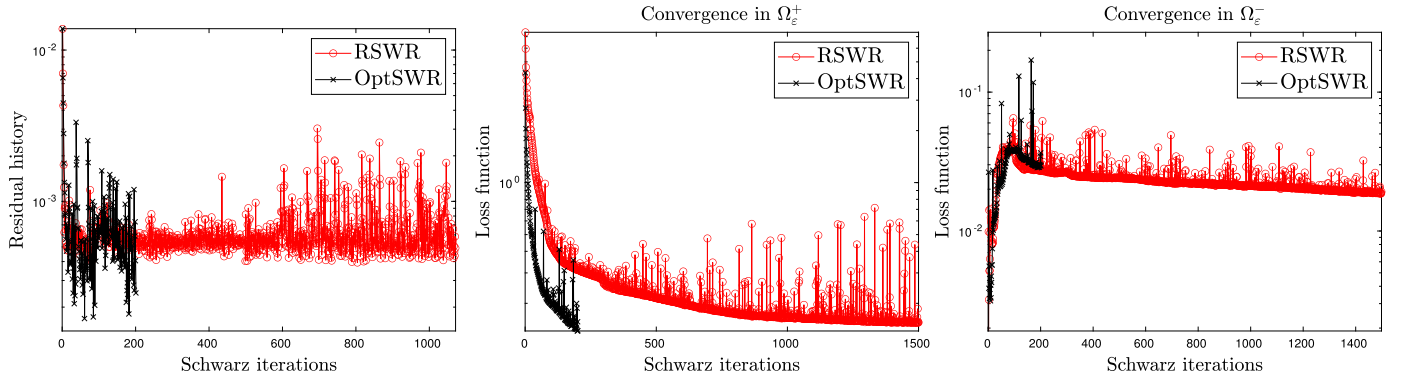


Fig. 6. Experiment 2b. Residual history comparison between Robin-SWR and optimal-SWR methods in Ω_ε^+ . (Left). (Middle) Local loss function $\mathcal{L}^+$ as function of k . (Right) Local loss function $\mathcal{L}^-$ as function of k .

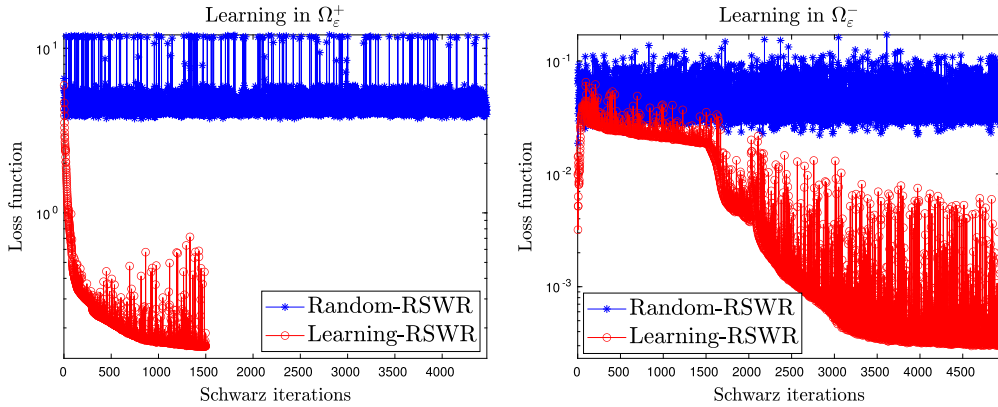


Fig. 7. Experiment 2b. Local loss function $\mathcal{L}^\pm$ as function of k for random and learnt parameters in RSWR-algorithm.

learning. The test is similar as above, except that the number of training data are 40 in both time and space, and the neural networks contain 10 neurons and 10 hidden layers. We have fixed to $N_0 = 10^4$ the maximal possible number of optimization iterations per Schwarz iteration. At each Schwarz iteration k for RSWR (Robin constant r is taken equal to 10) and Optimal SWR with and without learning in each subdomains $\Omega_\varepsilon^\pm$, we report in logscale the *normalized* number of optimization iterations for reaching the loss value $\eta = 0.3$ in Ω_ε^+ (resp. $\eta = 0.02$ in Ω_ε^-),

that is $\{(k, \ell_\eta^{\pm(k)}), k \geq 1\}$ where $\ell_\eta^{\pm(k)} = \frac{1}{N_0} \min_{\ell} \{\ell \geq 1, : \mathcal{L}(\theta_\ell^{\pm(k)}) \leq \eta\}$ for Optimal SWR Fig. 11 (Left) and Robin-SWR in Fig. 11 (Right). The normalized CPU-time per Schwarz iteration can be deduced from these graphs, considering that each optimization iteration requires roughly speaking a constant CPU-time. As the PINN-SWR algorithm depends on

many parameters (related to the optimization algorithms, neural network structures, etc.), it was preferred to report a normalized number of optimization iterations rather than normalized CPU-time. We also report in Fig. 12, for Optimal-SWR the loss function values as a function of Schwarz iterations in logscale. Selecting the number of optimization iterations *a priori* is non-standard, but it allows us to illustrate the SWR-learning effect provided by PINN, as well as the fact that Optimal-SWR allows for faster convergence than Robin-SWR.

Experiment 4. In this experiment, we aim to study the effect of choosing hyperparameters on the relative convergence of the accelerated PINN-RSWR and PINN-OptSWR algorithms. To achieve this goal, our focus is on the relative selection of hyperparameters λ (associated with the PDE residual) and μ (related to transmission conditions), as well as the convergence threshold within the optimization algorithm:

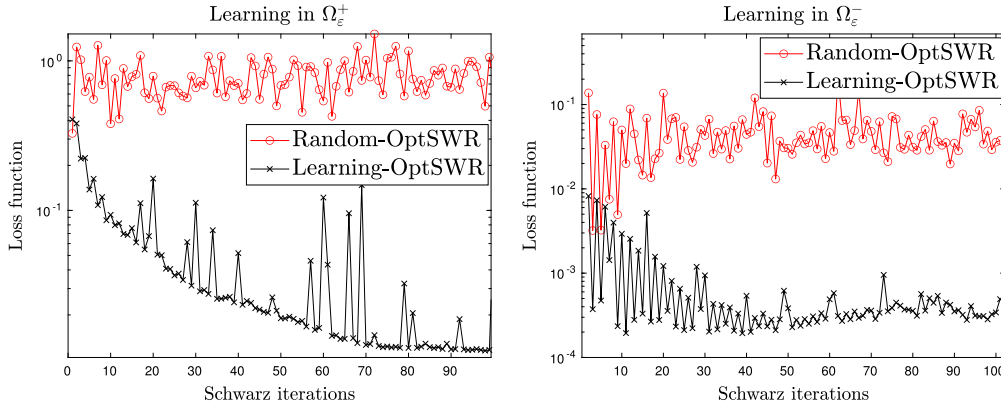


Fig. 8. Experiment 2b. Local loss function $\mathcal{L}^\pm$ as function of k for random and learnt parameters in optimal-SWR algorithms.

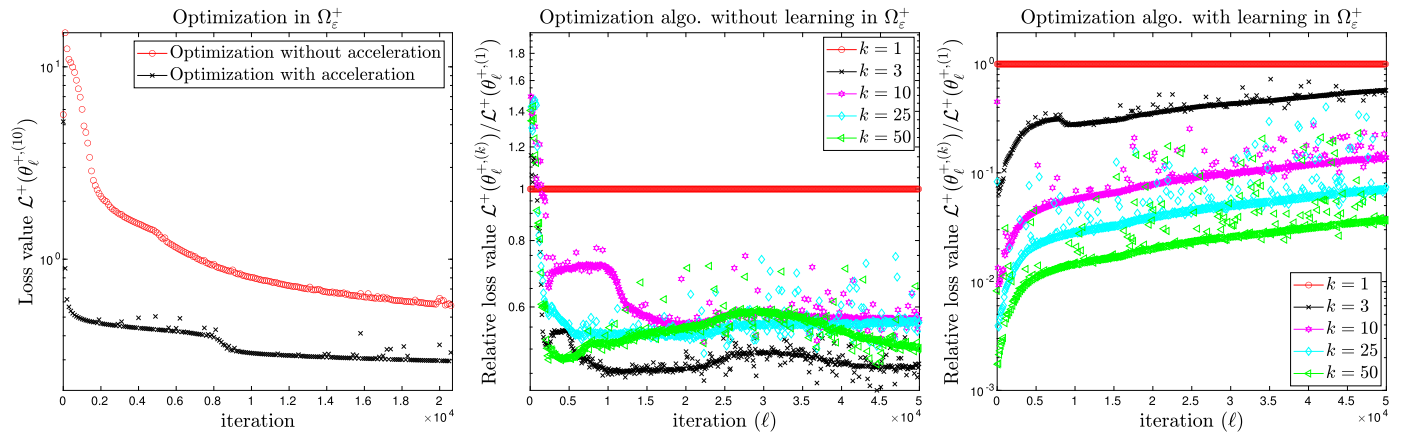


Fig. 9. Experiment 2b. (Left) Local loss function at iteration $k = 3$ with and without learning in Ω_ε^+ . (Middle) Normalized loss function (17) without learning. (Right) Normalized loss function (17) with learning.

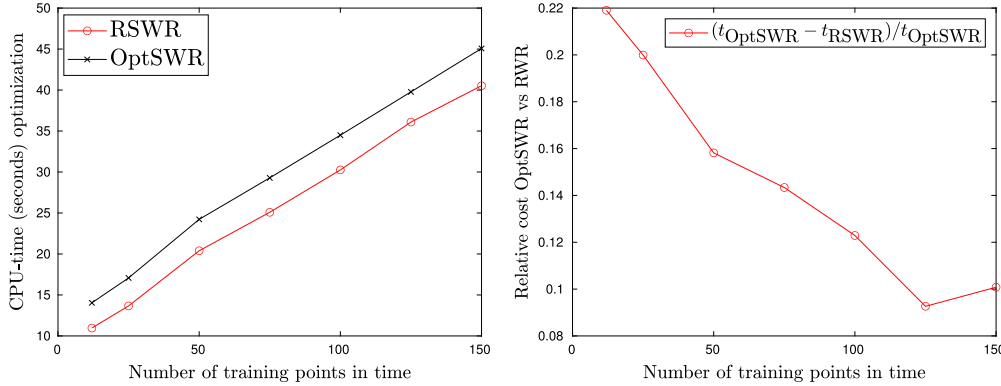


Fig. 10. Experiment 2b. (Left) CPU-time with PINN algorithm: Robin-SWR vs Optimal-SWR per Schwarz iteration. (Right) Relative cost Optimal-SWR vs Robin-SWR.

$$\begin{aligned} \mathcal{L}^\pm(\theta^{\pm(k)}) = & \lambda \left\| \mathbb{1}_{\partial_t} N^{\pm(k)}(\theta^{\pm(k)}, \cdot, \cdot) + \partial_{xx} N^{\pm(k)}(\theta^{\pm(k)}, \cdot, \cdot) \right. \\ & + V(x) N^{\pm(k)}(\theta^{\pm(k)}, \cdot, \cdot) \Big\|_{L^2(\Omega_\varepsilon^\pm \times [0, T])} \\ & + \mu \left\| \mathcal{T}_\pm(\pm \varepsilon/2, t) N^{\pm(k)}(\theta^{\pm(k)}, \pm \varepsilon/2, \cdot) \right. \\ & \left. - \mathcal{T}_\pm(\pm \varepsilon/2, t) N^{\mp(k-1)}(\theta^\mp, \pm \varepsilon/2, \cdot) \right\|_{L^2(0, T)}. \end{aligned} \quad (18)$$

The space-time domain is $[-7/4, 7/4] \times [0, 1/4]$ and the initial data is $u_0(x) = \exp(-\alpha(x - x_0)^2 + ik_0x)$, with $k_0 = 5$, $\alpha = 25$ and $x_0 = -7/8$. We consider two hidden layers with ten neurons each and report below the relative convergence of RSWR and Optimal-SWR algorithms for various hyperparameters.

Experiment 4.a This experiment focuses on the relative number of optimization iterations (OptSWR/RSWR) at convergence, for a fixed tol-

erance ε of the optimization algorithm, as a function of the Schwarz iterations, while varying the hyperparameters $\lambda_\pm$ in (18). We set $\rho_i := \lambda_\pm / \mu_\pm = 2^i$ for $i = 1, \dots, 4$, and fix $\mu_\pm = 10$. For fixed ρ_i , we denote

$$r_i^{(k)} = \mathcal{L}_{\text{RSWR}}^+(\theta_i^{+,(\ell_k)}), \quad o_i^{(k)} = \mathcal{L}_{\text{OptSWR}}^+(\theta_i^{+,(\ell_k)}),$$

such that $\mathcal{L}^+(\theta_i^{+,(\ell_k)}) < \varepsilon < \mathcal{L}^+(\theta_i^{+,(\ell_{k-1})})$. In Fig. 13, we present the number of iterations for the optimization algorithm for Optimal-SWR as a function of the Schwarz iterations for different values of the parameters: $\{(k, o_i^{(k)}), k = 1, \dots, 100\}$. In Fig. 13 (Right), we report the relative (OptSWR/RSWR) number of iterations $\{(k, o_i^{(k)} / r_i^{(k)}), k = 1, \dots, 100\}$. It is observed that, like any PINN-based PDE solver, the overall convergence of the PINN-SWR algorithms is largely dependent of the choice of

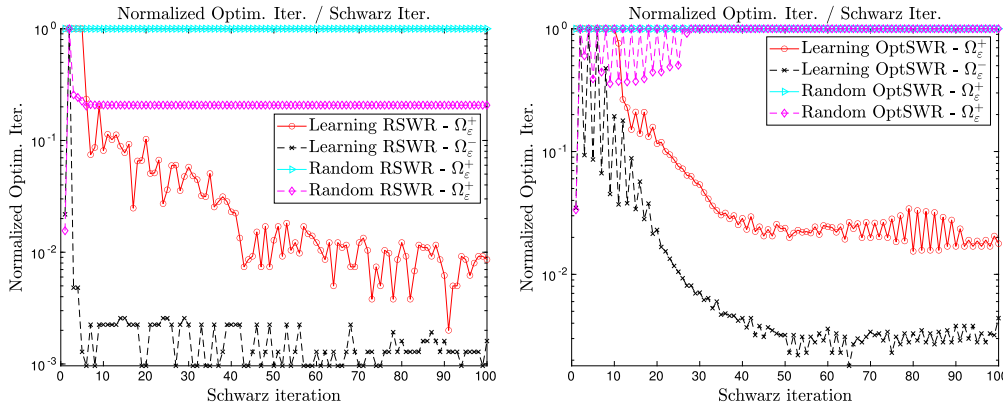


Fig. 11. Experiment 3. Normalized number of optimization iterations as a function of Schwarz iteration in logscale: Robin-SWR (Left) and Optimal-SWR (Right) with and without learning.

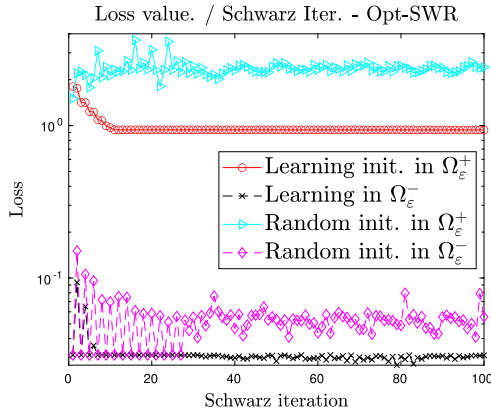


Fig. 12. Experiment 3. Loss function as a function of Schwarz iteration for Optimal-SWR with and without learning.

the hyper-parameters. Furthermore, PINN-OptSWR demonstrates better convergence, particularly for properly chosen parameters.

Experiment 4.b In this experiment, we set the parameters $\lambda = 5$ and $\mu = 10$, and vary the convergence tolerance of the optimization algorithm. Specifically, we use $\varepsilon_i = \varepsilon_0/2^i$ for $i = 0, 1, 2, 3$, with $\varepsilon_0 = 1$. The relative number of optimization iterations is then reported as a function of the tolerance ε_i for both RSWR and OptSWR. More specifically, we denote:

$$R_i^{(k)} := \mathcal{L}_{\text{RSWR}}^+(\theta_i^{+,(\ell_k)}), \quad O_i^{(k)} := \mathcal{L}_{\text{OptSWR}}^+(\theta_i^{+,(\ell_k)}),$$

such that $\mathcal{L}^+(\theta_i^{+,(\ell_k)}) < \varepsilon_i < \mathcal{L}^+(\theta_i^{+,(\ell_{k-1})})$. We report in Fig. 14 (Left), the number of iterations of optimization algorithm with OptSWR as a function of Schwarz iterations, for different tolerances $\varepsilon_i: \{(k, O_i^{(k)}), k = 1, \dots, 50\}$ $\{(k, O_i^{(k)}/R_i^{(k)}), k = 1, \dots, 50\}$. In Fig. 14, we report the relative (OptSWR/RSWR) number of iterations: $\{(k, O_i^{(k)}/R_i^{(k)}), k = 1, \dots, 50\}$. Notice that the smaller ε_i the more accurate, $N^{\pm, (k)}$, corresponding to the local approximate solutions to the Schrödinger equation.

These experiments highlight the crucial role of hyperparameter selection for achieving overall convergence in PINN-based algorithms, including PINN-SWR. Furthermore, they suggest that the results and conclusions presented in this paper, particularly regarding the relative convergence rates of PINN-RSWR vs. PINN-OptSWR, remain largely valid regardless of the chosen hyperparameters; at least within the parameter range that ensures overall convergence of the PINN-SWR algorithms.

Experiment 5. In this final experiment, our focus is specifically on the approximation of the transmission/boundary condition within the loss function—an important aspect for validating the conclusions presented in this paper. This particular question will also be addressed in high dimensions in a forthcoming paper on absorbing boundary conditions with neural network-based algorithms. To achieve this, we consider the following local IBVP within the domain $\Omega = (a, b)$ and for $t \in [0, T]$:

$$\begin{aligned} i\partial_t u + \partial_{xx} u + V(x)u &= 0, \quad x \in \Omega, t \in [0, T], \\ \mathcal{T}u &= 0, \quad x = b^-, t \in [0, T], \\ u &= 0 \quad x = a^+, t \in [0, T], \\ u(x, 0) &= u_0(x), \quad x \in (a, b), \end{aligned} \quad (19)$$

where the boundary operator $\mathcal{T}$ is defined as the Dirichlet-to-Neumann (DtN) operator $\mathcal{T}_{\text{DtN}} = \partial_t^{1/2} + e^{i\pi/4}\partial_x$ (used in OptSWR) or Robin operator $\mathcal{T}_{\text{Robin}} = r + i\partial_x$ (used in Robin SWR), for some fixed Robin constant. We assume hereafter that the initial and Dirichlet boundary conditions at $x = a^+$ are intrinsically integrated within the neural networks ($N_R(\cdot, 0) = \text{Re}(u_0)$ and $N_I(\cdot, 0) = \text{Im}(u_0)$ and $N_{R,I} = 0$ at $x = a^+$). Hence, we simply consider

$$\begin{aligned} \partial_t N_R &= -\partial_{xx} N_I - V(x)N_I, \quad \text{in } \Omega \times [0, T], \\ \partial_t N_I &= \partial_{xx} N_R - V(x)N_R, \quad \text{in } \Omega \times [0, T], \end{aligned}$$

either coupled with i) Dirichlet-to-Neumann boundary conditions

$$\begin{aligned} \partial_t^{1/2} N_R + \frac{1}{\sqrt{2}} \partial_x (N_R - N_I) &= 0, \quad \text{on } \{b^-\} \times [0, T], \\ \partial_t^{1/2} N_I + \frac{1}{\sqrt{2}} \partial_x (N_R + N_I) &= 0, \quad \text{on } \{b^-\} \times [0, T], \end{aligned}$$

where

$$\partial_t^{1/2} N_{R,I}(\theta, x, t) = \partial_t \int_0^t \frac{N_{R,I}(\theta, x, \tau)}{\sqrt{\pi(t-\tau)}} d\tau,$$

or coupled with ii) Robin boundary conditions

$$\begin{aligned} -rN_I + \partial_x N_R &= 0, \quad \text{on } \{b^-\} \times [0, T], \\ rN_R + \partial_x N_I &= 0, \quad \text{on } \{b^-\} \times [0, T]. \end{aligned}$$

We set

$$\begin{aligned} \mathcal{L}_{\text{Intern}}(\theta) &= \left\| \partial_t N_R + \partial_{xx} N_I + V(x)N_I \right\|_{L^2(\Omega_\varepsilon^\pm \times [0, T])} \\ &\quad + \left\| \partial_t N_I + \partial_{xx} N_R + V(x)N_R \right\|_{L^2(\Omega_\varepsilon^\pm \times [0, T])}. \end{aligned}$$

We then consider the following loss contribution for Dirichlet-to-Neumann boundary conditions

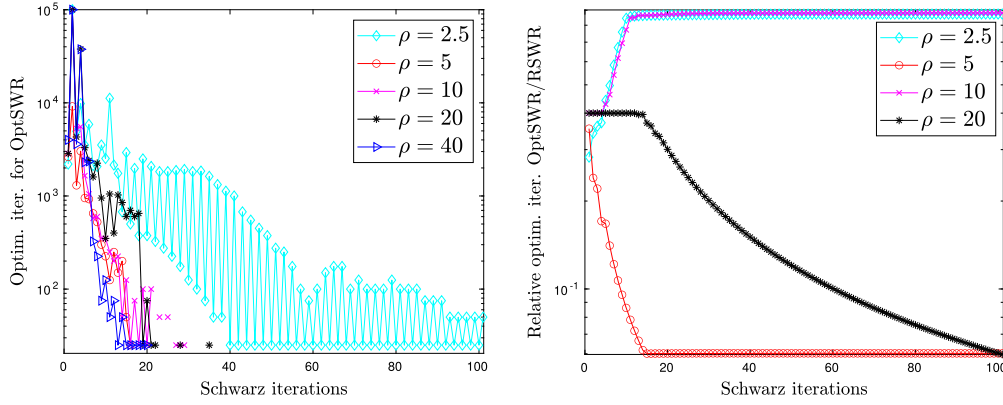


Fig. 13. Experiment 4. (Left) Number of optimization iterations for OptSWR, as function of Schwarz iterations for different values of ρ_i and fixed optimization algorithm tolerance: $\{(k, o_i^{(k)}), k = 1, \dots, 100\}$. (Right) Relative number of optimization iterations (OptSWR/RSWR) as function of Schwarz iterations for different values of ρ_i and fixed optimization algorithm tolerance: $\{(k, o_i^{(k)}/r_i^{(k)}), k = 1, \dots, 100\}$.

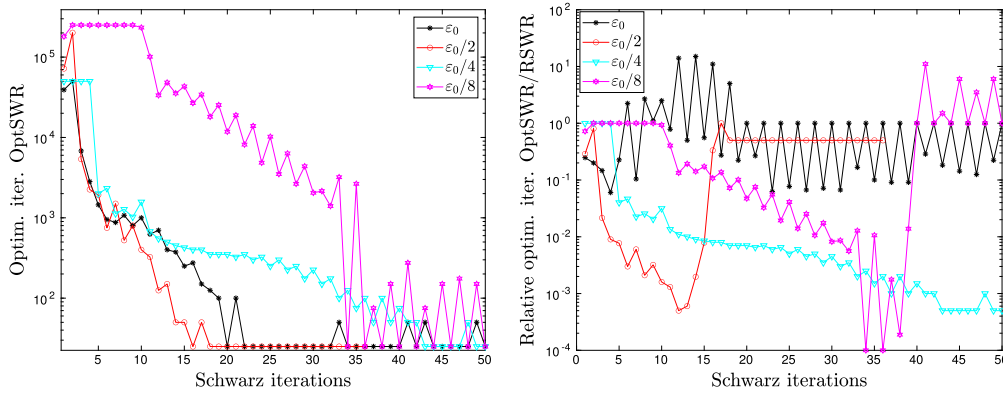


Fig. 14. Experiment 4.b (Left) Optimization iterations as function Schwarz iteration for different tolerance ε_i : $\{(k, O_i^{(k)}/R_i^{(k)}), k = 1, \dots, 50\}$ (Right) Relative number of optimization iterations as function Schwarz iterations for different tolerance ε_i : $\{(k, O_i^{(k)}/R_i^{(k)}), k = 1, \dots, 50\}$.

$$\mathcal{L}_{\text{DtN}}(\theta) = \left\| \partial_t^{1/2} N_R + \frac{1}{\sqrt{2}} \partial_x (N_R - N_I) \right\|_{L^2(\Gamma_\varepsilon^\pm \times [0, T])} + \left\| \partial_t^{1/2} N_I + \frac{1}{\sqrt{2}} \partial_x (N_R + N_I) \right\|_{L^2(\Gamma_\varepsilon^\pm \times [0, T])},$$

as well as a loss contribution for Robin boundary conditions

$$\mathcal{L}_{\text{Robin-BC}}(\theta) = \left\| -r N_I + \partial_x N_R \right\|_{L^2(\Gamma_\varepsilon^\pm \times [0, T])} + \left\| r N_R + \partial_x N_I \right\|_{L^2(\Gamma_\varepsilon^\pm \times [0, T])}.$$

Finally we set

$$\mathcal{L}_{\text{DtN}} = \lambda \mathcal{L}_{\text{Intern}} + \mu \mathcal{L}_{\text{DtN-BC}}, \quad \mathcal{L}_{\text{Robin}} = \lambda \mathcal{L}_{\text{Intern}} + \mu \mathcal{L}_{\text{Robin-BC}}. \quad (20)$$

Practically, the convolution product in DtN-operator is approximated using a discrete convolution product (function `convolve` from `numpy`). We then proceed as follows:

1. minimization of the global loss function $\mathcal{L}_{\text{DtN}}$ (resp. $\mathcal{L}_{\text{Robin}}$), that is construction of a sequence of parameters $\{\theta_{\text{DtN};\ell}\}_\ell$ (resp. $\{\theta_{\text{Robin};\ell}\}_\ell$);
2. report $\{(\ell, \mathcal{L}_{\text{DtN}}(\theta_{\text{DtN};\ell}), \ell \geq 0\}$, as well as $\{(\ell, \mathcal{L}_{\text{Robin}}(\theta_{\text{Robin};\ell}), \ell \geq 0\}$;
3. report $\{(\ell, \mathcal{L}_{\text{DtN-BC}}(\theta_{\text{DtN};\ell}), \ell \geq 0\}$, as well as $\{(\ell, \mathcal{L}_{\text{Robin-BC}}(\theta_{\text{Robin};\ell}), \ell \geq 0\}$.

Step 2 corresponds to the convergence of the PINN algorithm, while Step 3 specifically focuses on the contribution of the boundary conditions within the optimization algorithm. In this experiment, we consider neural networks with 2 hidden layers, each containing 10 neurons. The computational domain is such that $a = -2$ and $b = 0$, with null po-

tential, and initial condition given by $\exp(15x) \exp(-30(x+1)^2)$. The Robin constant is taken equal to $r = 1$. Within the loss functions, we take $\lambda = 0.9$ and $\mu = 0.1$, and for consistency the same set of randomly chosen points for both IBVP (Robin and DtN). In Fig. 15, we report the loss functions for Robin $\{(\ell, \mathcal{L}_{\text{DtN}}(\theta_{\text{Robin};\ell}), \ell \geq 0\}$ and DtN $\{(\ell, \mathcal{L}_{\text{DtN}}(\theta_{\text{DtN};\ell}), \ell \geq 0\}$ as well as the corresponding contribution of the boundary condition $\{(\ell, \mu \mathcal{L}_{\text{Robin-BC}}(\theta_{\text{Robin};\ell}), \ell \geq 0\}$ and $\{(\ell, \mu \mathcal{L}_{\text{DtN-BC}}(\theta_{\text{DtN};\ell}), \ell \geq 0\}$. This allows to specifically illustrate the convergence of the boundary conditions. Although the DtN operator is more complex and, as a consequence, less accurately approximated than the Robin operator, the overall choice of the boundary condition does not have a significant impact on the convergence of the PDE solver.

5. Conclusion

In this paper, we have studied PINN algorithms for solving the time-dependent Schrödinger equation using (quasi)-optimal SWR domain decomposition methods. In [7], it was shown that PINN solvers introduce some learning/acceleration within SWR algorithms. Specifically, regardless of the transmission conditions chosen, the optimization algorithm is accelerated from one Schwarz iteration to the next thanks to “learned” initial neural network parameters. On the other hand, optimal SWR algorithms are based on non-local transparent operators (Dirichlet-to-Neumann) and provide the fastest convergence rate among SWR algorithms. Dirichlet-to-Neumann operators are known to deteriorate the efficiency, stability, and potentially the accuracy of standard Schrödinger equation solvers. We have shown in this paper that the use of PINN largely circumvents this issue. In particular, we have observed that the overall efficiency of optimal SWR is higher than that of

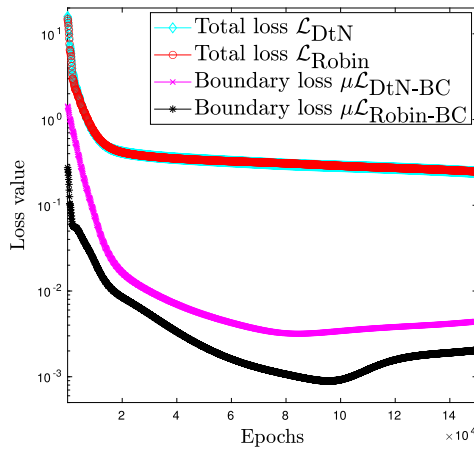


Fig. 15. Experiment 5. Global loss functions with Robin and DtN boundary conditions $\{(\ell, \mathcal{L}_{\text{DtN}}(\theta_{\text{Robin}; \ell}), \ell \geq 0)\}$ and $\{(\ell, \mathcal{L}_{\text{DtN}}(\theta_{\text{DtN}; \ell}), \ell \geq 0)\}$ as well as the corresponding contribution of the boundary condition $\{(\ell, \mu \mathcal{L}_{\text{Robin-BC}}(\theta_{\text{Robin}; \ell}), \ell \geq 0)\}$ and $\{(\ell, \mu \mathcal{L}_{\text{DtN-BC}}(\theta_{\text{DtN}; \ell}), \ell \geq 0)\}$.

CSWR/Robin-SWR, thanks to the acceleration property offered by the PINN-SWR approach and the automatic differentiation (and potentially integration) of neural networks, allowing for efficient computation of Dirichlet-to-Neumann transmission conditions. In future works, we will study the performance of PINN-SWR methods in high-dimensional PDEs, as well as NN-based approximations of Dirichlet-to-Neumann-like operators for absorbing boundary conditions for different types of wave equations.

CRedit authorship contribution statement

Emmanuel Lorin: Formal analysis, Methodology, Software, Writing – original draft, Writing – review & editing. **Xu Yang:** Formal analysis, Methodology, Software, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

- [1] L. Halpern, J. Szeftel, Optimized and quasi-optimal Schwarz waveform relaxation for the one-dimensional Schrödinger equation, *Math. Models Methods Appl. Sci.* 20 (12) (2010) 2167–2199.
- [2] X. Antoine, C. Besse, P. Klein, Absorbing boundary conditions for the one-dimensional Schrödinger equation with an exterior repulsive potential, *J. Comput. Phys.* 228 (2) (2009) 312–335.
- [3] X. Antoine, C. Besse, Unconditionally stable discretization schemes of non-reflecting boundary conditions for the one-dimensional Schrödinger equation, *J. Comput. Phys.* 188 (1) (2003) 157–175.
- [4] X. Antoine, C. Besse, P. Klein, Absorbing boundary conditions for the two-dimensional Schrödinger equation with an exterior potential. Part I: Construction

- and a priori estimates, *Math. Models Methods Appl. Sci.* 22 (10) (2012) 1250026, 38.
- [5] X. Antoine, F. Hou, E. Lorin, Asymptotic estimates of the convergence of classical Schwarz waveform relaxation domain decomposition methods for two-dimensional stationary quantum waves, *ESAIM: Math. Model. Numer. Anal. (M2AN)* 52 (4) (2018) 1569–1596.
- [6] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.
- [7] E. Lorin, X. Yang, Schwarz waveform relaxation-learning for advection-diffusion-reaction equations, *J. Comput. Phys.* 473 (2023) 111657.
- [8] Ji. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, *Proc. Natl. Acad. Sci. USA* 115 (34) (2018) 8505–8510.
- [9] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, L. Zdeborová, Machine learning and the physical sciences, *Rev. Mod. Phys.* 91 (Dec 2019) 045002.
- [10] H. Gao, M.J. Zahr, J.-X. Wang, Physics-informed graph neural Galerkin networks: a unified framework for solving PDE-governed forward and inverse problems, *Comput. Methods Appl. Mech. Eng.* 390 (2022) 114502, 18.
- [11] J. Sirignano, K. Spiliopoulos, DGM: a deep learning algorithm for solving partial differential equations, *J. Comput. Phys.* 375 (2018) 1339–1364.
- [12] X. Antoine, E. Lorin, An analysis of Schwarz waveform relaxation domain decomposition methods for the imaginary-time linear Schrödinger and Gross-Pitaevskii equations, *Numer. Math.* 137 (4) (2017) 923–958.
- [13] X. Antoine, E. Lorin, On the rate of convergence of Schwarz waveform relaxation methods for the time-dependent Schrödinger equation, *J. Comput. Appl. Math.* 354 (2019) 15–30.
- [14] L. Nirenberg, *Lectures on Linear Partial Differential Equations*, American Mathematical Society, Providence, R.I., 1973.
- [15] X. Antoine, C. Besse, Construction, structure and asymptotic approximations of a microdifferential transparent boundary condition for the linear Schrödinger equation, *J. Math. Pures Appl.* (9) 80 (7) (2001) 701–738.
- [16] A. Modave, A. Royer, X. Antoine, C. Geuzaine, A non-overlapping domain decomposition method with high-order transmission conditions and cross-point treatment for Helmholtz problems, *Comput. Methods Appl. Mech. Eng.* 368 (2020) 113162.
- [17] I. Badia, B. Caudron, X. Antoine, C. Geuzaine, A well-conditioned weak coupling of boundary element and high-order finite element methods for time-harmonic electromagnetic scattering by inhomogeneous objects, *SIAM J. Sci. Comput.* 44 (3) (2022) B640–B667.
- [18] G. Pang, L. Lu, G.E. Karniadakis, fPINNs: fractional physics-informed neural networks, *SIAM J. Sci. Comput.* 41 (4) (2019) A2603–A2626.
- [19] L. Yang, D. Zhang, G.E. Karniadakis, Physics-informed generative adversarial networks for stochastic differential equations, *SIAM J. Sci. Comput.* 42 (1) (2020) A292–A317.
- [20] G. Carleo, M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* 355 (6325) (2017) 602–606.
- [21] E. Lorin, X. Yang, Time-dependent Dirac equation with physics-informed neural networks: computation and properties, *Comput. Phys. Commun.* 280 (2022) 108474, 15.
- [22] K. Shukla, A.D. Jagtap, G.E. Karniadakis, Parallel physics-informed neural networks via domain decomposition, *J. Comput. Phys.* 447 (2021) 110683, 19.
- [23] H.H. Kim, H.J. Yang, Domain decomposition algorithms for physics-informed neural networks, in: Susanne C. Brenner, Eric Chung, Axel Klawonn, Felix Kwok, Jinchao Xu, Jun Zou (Eds.), *Domain Decomposition Methods in Science and Engineering XXVI*, Springer International Publishing, Cham, 2022, pp. 697–704.
- [24] W. Wu, X. Feng, H. Xu, Improved deep neural networks with domain decomposition in solving partial differential equations, *J. Sci. Comput.* 93 (1) (2022) 20, 34.
- [25] I.E. Lagaris, A. Likas, D.I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Trans. Neural Netw.* 9 (5) (1998) 987–1000.
- [26] X. Antoine, E. Lorin, Q. Tang, A friendly review to absorbing boundary conditions and perfectly matched layers for classical and relativistic quantum wave equations, *Mol. Phys.* 115 (2017).
- [27] R. Gorenflo, Fractional calculus: some numerical methods, in: *CISM Courses and Lectures*, vol. 378, Springer, 1997.
- [28] N.J. Ford, J.A. Connolly, Comparison of numerical methods for fractional differential equations, *Commun. Pure Appl. Anal.* 5 (2) (2006) 289–307.
- [29] M. Gander, L. Halpern, Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems, *SIAM J. Numer. Anal.* 45 (2) (2007) 666–697.