Multilevel preconditioning techniques for Schwarz waveform relaxation domain decomposition methods for real-and imaginary-time nonlinear Schrödinger equations

X Antoine, E Lorin

▶ **To cite this version:**

X Antoine, E Lorin. Multilevel preconditioning techniques for Schwarz waveform relaxation domain decomposition methods for real-and imaginary-time nonlinear Schrödinger equations. Applied Mathematics and Computation, 2018, 336 (1), pp.403-417. 10.1016/j.amc.2018.04.075 . hal-01266021

HAL Id: hal-01266021
https://hal.science/hal-01266021

Submitted on 2 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multilevel preconditioning techniques for Schwarz waveform relaxation domain decomposition methods for real- and imaginary-time nonlinear Schrödinger equations

X. Antoine[a,b], E. Lorin[c,d]

[a] *Institut Elie Cartan de Lorraine, Université de Lorraine, F-54506 Vandoeuvre-lès-Nancy Cedex, France*
[b] *Inria Nancy Grand-Est/IECL - SPHINXS.*
[c] *Centre de Recherches Mathématiques, Université de Montréal, Montréal, Canada, H3T 1J4*
[d] *School of Mathematics and Statistics, Carleton University, Ottawa, Canada, K1S 5B6*

Abstract

This paper is dedicated to the derivation of a multilevel Schwarz Waveform Relaxation (SWR) Domain Decomposition Method (DDM) in real- and imaginary-time for the NonLinear Schrödinger Equation (NLSE). In imaginary-time, it is shown that the use of the multilevel SWR-DDM accelerates the convergence compared to the one-level SWR-DDM, resulting in an important reduction of the computational time and memory storage. In real-time, the method requires in addition the storage of the solution in overlapping zones at any time, but on coarser discretization levels. The method is numerically validated on the Classical SWR and Robin-based SWR methods but can however be applied to any SWR approach.

Keywords: Domain decomposition method, Schwarz waveform relaxation algorithm, multilevel preconditioning, nonlinear Schrödinger equation, dynamics, stationary states

1. Introduction

This paper is devoted to the derivation of a multilevel Schwarz Waveform Relaxation (SWR) method for computing both in real- and imaginary-time the solution to the NonLinear Schrödinger Equation (NLSE) [4, 5, 6, 10, 11]. Domain decomposition SWR methods for solving wave equations have a long history from the classical SWR method with overlapping zones to optimized version without overlap (see e.g. [7, 9, 12, 15, 16, 17, 18, 19, 22, 8] as well as `http://www.ddm.org`, for a complete review and references about this method). Basically in SWR methods, the transmission conditions at the subdomain interfaces are derived from the solution to the corresponding wave equation, usually using Dirichlet boundary conditions (Classical SWR), Robin boundary conditions, transparent or high-order Absorbing Boundary Conditions (ABCs) including Dirichlet-to-Neumann (DtN) transmitting conditions (Optimized SWR), or Perfectly Matched Layers [1, 9, 21]. We also refer to [1, 2, 20, 23] for some reviews on truncation techniques for quantum wave equations in infinite domains. SWR methods can be *a priori* applied to any type of wave equation [13, 14, 15].

Email addresses: `xavier.antoine@univ-lorraine.fr` (X. Antoine), `elorin@math.carleton.ca` (E. Lorin)

X. Antoine thanks the support of the french ANR grants "Bond" (ANR-13-BS01-0009-01) and ANR-12-MONU-0007-02 BECASIM (Modèles Numériques call). E. Lorin thanks NSERC for the financial support via the Discovery Grant program.

In this paper, we will focus on multilevel SWR for the NLSE. More specifically, we consider the cubic time-dependent (real-time) NLSE set on $\mathbb{R}^d$, with $d \geqslant 1$,

$$\begin{cases} \mathtt{i}\partial_t u = -\triangle u + V(\mathbf{x})u + \kappa|u|^2 u, \, \mathbf{x} \in \mathbb{R}^d, \, t > 0, \\ \\ u(\mathbf{x}, 0) = \phi_0(\mathbf{x}), \, \mathbf{x} \in \mathbb{R}^d. \end{cases} \tag{1}$$

The real-valued space-dependent smooth potential V is positive for attractive interactions and negative for repulsive interactions. The nonlinearity strength κ is a real-valued constant which is positive for a focusing nonlinearity and negative for a defocusing nonlinearity. The function ϕ_0 is a given initial data. In the sequel of the paper, $\mathcal{P}(|u|)$ denotes the nonlinear operator

$$\mathcal{P}(|u|)u = \Big(\mathtt{i}\partial_t + \triangle - V(\mathbf{x}) - \kappa|u|^2\Big)u. \tag{2}$$

By using pseudodifferential operator calculus, rates of convergence for the CSWR and OSWR methods can be derived [7] in imaginary-time. Concretely, the imaginary-time formulation [4, 5, 6, 10, 11] is used to compute the stationary solutions to the NLSE. The corresponding method is referred to as Continuous Normalized Gradient Flow (CNGF) formulation [7, 11] in the Mathematics literature and imaginary-time method in the Physics literature. The current paper is an extension of [7] where we focus on multilevel *preconditioning*. In the imaginary-time framework (stationary state computation), we refer to as *preconditioning* the storage and use of a converged solution at a lower (coarser) level for i) initializing the CNGF algorithm (Cauchy data selection) and for ii) deriving the transmission conditions in the overlapping zone interfaces at an upper (finer) level. In real-time (computation of the dynamics), *preconditioning* also includes the storage of the converged solution in the overlapping zones, *at any time*, for accurately deriving the transmission conditions. We numerically show that the convergence of the SWR method is improved in both cases. Although, the convergence acceleration is moderate in imaginary-time, it is however shown that the computational cost per Schwarz iteration, that is the CNGF convergence, is strongly accelerated compared to unpreconditioned SWR methods.

The paper is organized as follows. In Subsections 2.1 and 2.2, we recall some results about SWR methods in real- and imaginary-time. In Subsection 2.2, we provide some informations about the Continuous Normalized Gradient Flow (CNGF) method for solving the stationary NLSE. Subsection 2.3 gives some notations about the multi-level approximation. In Section 3, we describe the two-level SWR method in imaginary-time and next in real-time. A discussion on the computational complexity is also addressed. Section 4 is devoted to some numerical experiments where two types of results are presented: i) convergence rates for Schwarz algorithms and ii) CNGF convergence time in imaginary-time. We finally conclude in Section 5.

2. SWR methods in real- and imaginary-time; notations

2.1. SWR algorithms in real-time

We recall the basics of SWR algorithms which are presented for two subdomains for the sake of conciseness. We introduce two open sets $\Omega_\varepsilon^\pm$ such that $\mathbb{R}^d = \Omega_\varepsilon^+ \cup \Omega_\varepsilon^-$, with overlapping region $\Omega_\varepsilon^+ \cap \Omega_\varepsilon^-$, where ε is a (small) non-negative parameter. In 1d ($d = 1$), the domains of interest read: $\Omega_\varepsilon^+ = \big(-\infty, \varepsilon/2\big)$, $\Omega_\varepsilon^- = \big(-\varepsilon/2, \infty\big)$ and $\mathbb{R} = \Omega_\varepsilon^+ \cup \Omega_\varepsilon^-$. with $\Omega_\varepsilon^+ \cap \Omega_\varepsilon^- = \big(-\varepsilon/2, \varepsilon/2\big)$. We denote by $\phi^\pm$ the solution to the time-dependent GPE in $\Omega_\varepsilon^\pm$. Solving the NLSE by a Schwarz

waveform domain decomposition [9] requires transmission conditions at the subdomain interfaces. More specifically, for any Schwarz iteration $k \geqslant 1$, the equation in $\Omega_\varepsilon^\pm$ reads, for a given $T > 0$,

$$\left\{\begin{array}{lcl} \left(\mathtt{i}\partial_t + \triangle - V - \kappa\left|\phi^{\pm,(k)}\right|^2\right)\phi^{\pm,(k)} & = & 0, \text{ on } \Omega_\varepsilon^\pm \times (0,T), \\ \mathcal{B}^\pm\phi^{\pm,(k)} & = & \mathcal{B}^\pm\phi^{\mp,(k-1)}, \text{ on } \Gamma_\varepsilon^\pm \times (0,T), \\ \phi^{\pm,(k)}(\cdot,0) & = & \phi_0(\cdot) \text{ on } \Omega_\varepsilon^\pm, \end{array}\right. \tag{3}$$

where $\Gamma_\varepsilon^\pm = \partial\Omega_\varepsilon^\pm$. The notation $\phi^{\pm,(k)}$ stands for the solution $\phi^\pm$ in $\Omega_\varepsilon^\pm \times (0,T)$ at Schwarz iteration k. Initially, $\phi^{\pm,(0)}$ are two given functions defined in $\Omega_\varepsilon^\pm$. We denote by $\mathcal{B}^\pm$ an operator characterizing the type of SWR algorithm. In the CSWR case, $\mathcal{B}^\pm$ is the identity operator and $\mathcal{B}^\pm = \partial_{\mathbf{n}^\pm} + \gamma\mathrm{Id}$ ($\gamma \in \mathbb{R}_+^*$) for the Robin-like SWR method. For the Optimized SWR algorithm, $\mathcal{B}^\pm$ can be a local or a nonlocal approximation of the DtN operator (see [9, 21]). The convergence criterion for the Schwarz DDM is fixed by the constraint

$$\left\| \ \|\phi^{+,(k)}_{|\Gamma_\varepsilon^+} - \phi^{-,(k)}_{|\Gamma_\varepsilon^-}\|_{\infty,\Gamma_\varepsilon}\right\|_{L^2(0,T)} \leqslant \delta^{\mathrm{Sc}}. \tag{4}$$

typically with $\delta^{\mathrm{Sc}} = 10^{-14}$ ("Sc" is added for Schwarz). The convergence occurs at an iteration denoted by k^{cvg}, and the converged global solution is denoted by $\phi^{\mathrm{cvg}} := \phi^{(k^{\mathrm{cvg}})}$.

2.2. SWR algorithms in imaginary-time

The computation of stationary states, e.g. ground states and excited states, corresponds [11] to computing a real number μ and a spatially varying eigenfunction ϕ satisfying the equation

$$\mu\phi(\mathbf{x}) = -\triangle\phi(\mathbf{x}) + V(\mathbf{x})\phi(\mathbf{x}) + \kappa\phi(\mathbf{x}), \mathbf{x} \in \mathbb{R}^d,$$

with the L^2-norm constraint

$$||\phi||_0^2 := \int_{\mathbb{R}^d} |\phi(\mathbf{x})|^2 d\mathbf{x} = 1.$$

The total energy of the system is defined as

$$E_\kappa(\chi) := \int_{\mathbb{R}^d} |\nabla\chi(\mathbf{x})|^2 + V(\mathbf{x})|\chi(\mathbf{x})|^2 + \frac{\kappa}{2}|\chi(\mathbf{x})|^4 d\mathbf{x}. \tag{5}$$

A stationary state is then such that $E_\kappa(\phi) := \min_{||\chi||_0=1} E_\kappa(\chi)$. Once ϕ is obtained, the eigenvalue μ is given by

$$\mu := \mu_\kappa(\phi) = E_\kappa(\phi) + \int_{\mathbb{R}^d} \frac{\kappa}{2}|\phi(\mathbf{x})|^4 d\mathbf{x}.$$

To determine μ and ϕ, a standard method is the imaginary-time/CNGF method [11] which consists in solving (1) in imaginary-time, i.e. setting $t \to \mathtt{i}t$. This leads to

$$\left\{\begin{array}{l} \partial_t\phi(\mathbf{x},t) = -\nabla_{\phi^*}E_\kappa(\phi) = \triangle\phi(\mathbf{x},t) - V(\mathbf{x})\phi(\mathbf{x},t) - \kappa|\phi|^2\phi(\mathbf{x},t),\ \mathbf{x} \in \mathbb{R}^d,\ t_n < t < t_{n+1}, \\ \phi(\mathbf{x},t_{n+1}) := \phi(\mathbf{x},t_{n+1}^+) = \dfrac{\phi(\mathbf{x},t_{n+1}^-)}{||\phi(\cdot,t_{n+1}^-)||_0}, \\ \phi(\mathbf{x},t) = \phi_0(\mathbf{x}),\ \mathbf{x} \in \mathbb{R}^d, \text{with } ||\phi_0||_0^2 = 1. \end{array}\right. \tag{6}$$

In the above equation, $t_0 := 0 < t_1 < ... < t_{n+1} < ...$ are the discretization times (that we assume to be equally spaced here), ϕ_0 is an initial guess for the time marching algorithm discretizing the projected gradient method and $\lim_{t\to t_n^{\pm}} \phi(\mathbf{x},t) = \phi(\mathbf{x},t_n^{\pm})$. The corresponding semi-discrete energy is diminishing [11] for given positive potential V and interaction strength κ.

Within the SWR formalism, we then have to minimize an energy problem at each Schwarz iteration $k \geqslant 1$. This is achieved at an imaginary-time, denoted by $T^{(k)} > 0$, where for $t > T^{(k)}$, $\|\phi^{\pm}(\cdot,T^{(k)}) - \phi^{\pm}(\cdot,t)\|_0$ is small enough. The convergence time of the CNGF at iteration k grows with $T^{(k)}$. For a two-subdomains decomposition, we then solve

$$\left\{\begin{array}{rcl} \partial_t \phi^{\pm,(k)} & = & \triangle \phi^{\pm,(k)} - V(\cdot)\phi^{\pm,(k)} - \kappa|\phi^{\pm,(k)}|^2\phi^{\pm,(k)}, \text{ on } \Omega_{\varepsilon}^{\pm}\times(t_n,t_{n+1}), \\ \mathcal{B}^{\pm}\phi^{\pm,(k)} & = & \mathcal{B}^{\pm}\phi^{\mp,(k-1)}, \text{ on } \Gamma_{\varepsilon}^{\pm}\times(t_n,t_{n+1}), \\ \phi^{\pm,(k)}(\cdot,0) & = & \phi_0(\cdot), \text{ on } \Omega_{\varepsilon}^{\pm}, \\ \phi^{\pm,(k)}(\cdot,t_{n+1}) & = & \phi^{\pm,(k)}(\cdot,t_{n+1}^{+}) = \frac{\phi^{\pm,(k)}(\cdot,t_{n+1}^{-})}{||\widetilde{\phi}^{-,(k)}(\cdot,t_{n+1}^{-})+\widetilde{\phi}^{+,(k)}(\cdot,t_{n+1}^{-})||_0}, \text{ in } \Omega_{\varepsilon}^{\pm}, \end{array}\right. \tag{7}$$

where again, $t_0 := 0 < t_1 < ... < t_{n+1} < ...$ are uniformly spaced discrete times, with constant time step Δt, and $\widetilde{\phi}^{+}$ (resp. $\widetilde{\phi}^{-}$) denotes the extension to $\mathbb{R}^d$ of ϕ^{+} (resp. ϕ^{-}). Concerning the CNGF convergence criterion for a given Schwarz iteration k, we stop the computation when

$$||\phi^{n+1,(k)} - \phi^{n,(k)}||_{\infty} \leqslant \delta,$$

where δ is a small parameter and $\|\phi\|_{\infty} := \sup_{\mathbf{x}\in\mathbb{R}^d}|\phi(\mathbf{x})|$. At the CNGF convergence, the stopping time is such that: $T^{(k)} = T^{\mathrm{cvg},(k)} := n^{\mathrm{cvg},(k)}\Delta t$ for a converged solution $\phi^{\mathrm{cvg},(k)}$ reconstructed from the two subdomains solutions $\phi^{\pm,\mathrm{cvg},(k)}$. The convergence criterion for the Schwarz DDM is fixed by the constraint

$$\Big\| \ \|\phi^{+,\mathrm{cvg},(k)}_{|\Gamma_{\varepsilon}^{+}} - \phi^{-,\mathrm{cvg},(k)}_{|\Gamma_{\varepsilon}^{-}}\|_{\infty,\Gamma_{\varepsilon}}\Big\|_{L^2(0,T^{(k^{\mathrm{cvg}})})} \leqslant \delta^{\mathrm{Sc}}. \tag{8}$$

The convergence of the whole iterative algorithm is obtained at Schwarz iteration k^{cvg}. Then, one gets the converged global solution $\phi^{\mathrm{cvg}} := \phi^{\mathrm{cvg},(k^{\mathrm{cvg}})}$, typically with $\delta^{\mathrm{Sc}} = 10^{-14}$ ("Sc" is added for Schwarz).

2.3. Notations and discretization

We introduce here some important notations. The domain $\mathbb{R}^d$ ($d \geqslant 1$) is approximated by a uniform finite volume/difference grids Ω_l with cell volume h_l^d, where $h_l = h_0/2^l$ ($l \geqslant 1$) and $h_0 \in \mathbb{R}^*$ corresponds to the coarsest grid one-dimensional space step. For instance, for $d = 3$, we have

$$\Omega_l = \cup_{i,j,k}\Omega_l^{(i,j,k)} = \cup_{i,j,k}\big[ih_l,(i+1)h_l\big]\times\big[jh_l,(j+1)h_l\big]\times\big[kh_l,(k+1)h_l\big].$$

The operator $P_h^{m;l}$ designates a projection operator from the grid Ω_m to Ω_l, where $m > l$. In practice, $P_h^{m;l}$ is an average operator defined iteratively from Ω_m to Ω_l. Conversely, we introduce $I_h^{l;m}$, for $m > l$, as a polynomial interpolation operator from Ω_l to Ω_m. In practice, the standard Lagrangian interpolation is be used. Moreover, $\Omega_{\varepsilon,l}^{\pm}$ is defined as the uniform subgrid such that $\Omega_l = \Omega_{\varepsilon,l}^{+}\cup\Omega_{\varepsilon,l}^{-}$ on which are defined the semi-discrete time-dependent solutions $\phi_l^{\pm,(k)}(t) = \big\{\phi_{l,j}^{\pm,(k)}(t)\big\}_j$

at Schwarz iteration k. For the sake of simplicity we also use the notation $\phi_l^{\pm,(k)} = \left\{\phi_{l,j}^{\pm,(k)}\right\}_j$ for denoting $\phi_l^{\pm,(k)}(t)$, and we now set $\Omega_l^{\pm} = \Omega_{\varepsilon,l}^{\pm}$. The reconstructed solution $\phi_l^{(k)}$ on Ω_l is

$$\phi_l^{(k)} := \frac{\widetilde{\phi}_l^{+,(k)} + \widetilde{\phi}_l^{-,(k)}}{||\widetilde{\phi}_l^{+,(k)} + \widetilde{\phi}_l^{-,(k)}||_{\ell^2(\Omega_l)}}, \tag{9}$$

where $\widetilde{\phi}_l^{\pm,(k)}$ denotes the extension by 0 of $\phi_l^{\pm,(k)}$ to $\Omega_l^{\mp}$ and $||\widetilde{\phi}_l||_{\ell^2(\Omega_l)}$ is the (discrete) ℓ^2-norm on Ω_l.

3. Multi-level SWR methods

A two-level preconditioning technique is derived by using i) the converged solution computed at a lower (coarser) level and ii) interpolation operations from $\Omega_m^{\pm}$ to $\Omega_l^{\pm}$, with $m > l$.

3.1. Two-level SWR method in imaginary-time

Let us assume that an approximation $f_{p;l} = \left\{f_{p;l,j}\right\}_j$ of the eigenfunction f_p associated to the p^{th} eigenvalue λ_p approximated by $\lambda_{p;l}$ has been computed at level l on Ω_l, where

$$\mathcal{P}_l(|f_{p;l}|)f_{p;l} = \lambda_{p;l} f_{p;l},$$

where $\mathcal{P}_l$ designates a discrete approximation of $\mathcal{P}$ on Ω_l (see Section 4.1). For $m > l$, using a domain decomposition on $\Omega_m^{\pm}$, as described above, leads to computing time-dependent local wavefunctions $\phi_l^{\pm,(k)} = \left\{\phi_{l,j}^{\pm,(k)}\right\}_j$. This requires

- to choose a discrete initial guess $\phi_m^{(k)}(t=0) = \left\{\phi_{m,j}^{(k)}(t=0)\right\}_j$, for all $k \geqslant 0$,
- and to impose a transmission condition on $\phi_m^{\pm,(k)}(t)$ at the interface $\Gamma_m^{\pm}$ for all $t > 0$.

To this end, we use i) $f_{p;l}$ and ii) the interpolation operator $I_h^{l;m}$. At level $m > l$ ($h_m < h_l$) and for all $k \geqslant 0$, the initial guess is built as

$$\phi_m^{(k)}(0) = I_h^{l;m} f_{p;l}, \qquad \text{on } \Omega_m^{\pm}.$$

Therefore, this is useful since the initial function in the minimization process at level m is expected to be close to the converged solution $f_{p;m}$, for m close to l. We then impose

$$\phi_m^{\pm,(k)}(0) = I_h^{l;m} f_{p;l}^{\pm}, \qquad \text{on } \Omega_m^{\pm}.$$

We also want to benefit from the knowledge of $f_{p;l}$ for designing the transmission conditions. We then impose at the subdomain interfaces, for all $t \geqslant 0$

$$\mathcal{B}_m^{\pm}\phi_m^{\pm,(0)}(t) = \mathcal{B}_m^{\pm} I_h^{l;m} f_{p;l}^{\pm}, \qquad \text{on } \Gamma_m^{\pm}, \tag{10}$$

where the operator $\mathcal{B}_m^{\pm}$ is an approximation of the operator $\mathcal{B}^{\pm}$ on $\Gamma_m^{\pm}$. From now on, all the necessary conditions to perform the Algorithm (7) on Ω_m are available. The preconditioning step ensures simultaneously that the initial guess and transmission conditions are already close to the

exact solution on the grids $\Omega_m^\pm$. Compared to a direct computation of $f_{p;m}$, the additional workload consists of computing and storing $f_{p;l}$, which makes this approach quite simple and attractive. The computational complexity aspects will be detailed in Section 3.4.

To summarize, from level l to level $m > l$, we perform:

1. At the lower level l: computation of $f_{p;l}$, starting from an initial guess $\phi_l^\pm(0) = \phi_{0;l}^\pm$.
2. At the upper level m: computation of $f_{p;m}$, starting from $\phi_m^\pm(0) = I_h^{l;m} f_{p;l}^\pm$ and with transmission conditions at $\Gamma_m^\pm$ by using $I_h^{l;m} f_{p;l}$.

The expected gain is not the acceleration of the SWR algorithm through an improved convergence, but rather the acceleration of the CNGF algorithm at each Schwarz iteration k. Typically, this procedure is used between two successive levels, i.e. with $m = l + 1$.

3.2. Two-level SWR method in real-time

It is possible to directly adapt the method proposed above from imaginary-time to real-time. However, although the relative computational workload is roughly the same compared to imaginary-time, additional data-storage is necessary as detailed below. Let us first assume that the converged solution to (1) is computed on $\Omega_l^\pm \times [0, T]$ and its restriction to $\Gamma_l^\pm$ is stored in $\mathcal{S}_l^\pm$, defined by

$$\mathcal{S}_l^\pm = \big\{ \phi_l^{\pm,\mathrm{cvg}}\big(t_l^n\big) \text{ at } \Gamma_l^\pm,\ \forall n \in \{0, \cdots, L_l\} \big\}, \tag{11}$$

where i) $\phi_l^{\pm,\mathrm{cvg}}$ is the converged solution on $\Omega_l^\pm$ and ii) $t_l^n = n\Delta t_l$, with $n \in \{0, \cdots, L_l\}$, $\Delta t_l L_l = T$. This set is used for preconditioning the Schwarz algorithm. More specifically on $\Omega_m^\pm$, at Schwarz iteration $k \geqslant 1$, we need to impose transmission conditions at any time $t \in [0, T]$. Note that unlike the imaginary-time situation, the Cauchy data is a given data, which restricts the flexibility of the method in the real-time context. We then set the following transmission conditions at any time t_m^n on $\Gamma_m^\pm$.

- If $\Delta t_l = \Delta t_m$, we impose:

$$\mathcal{B}^\pm \phi_m^{\pm,(0)}(t_m^n) = \mathcal{B}^\pm I_h^{l;m} \phi_l^{\pm,\mathrm{cvg}}\big(t_m^n\big), \qquad \text{on } \Gamma_m^\pm.$$

- If $\Delta t_l > \Delta t_m$, it is necessary to interpolate in time the converged solution on $\Omega_l^\pm$ computed at times t_l^n, for $n \leqslant L_l$, to get an estimate of $I_h^{l;m}\phi_l^{\pm,\mathrm{cvg}}$ at times $t_m^{(n)}$, with $n \leqslant L_m$, and where $L_m < L_l$. The corresponding interpolation operator is denoted by $I_{\Delta t}^{l;m}$ and we then impose

$$\mathcal{B}^\pm \phi_m^{\pm,(0)}\big(t_m^n\big) = \mathcal{B}^\pm \big(I_{\Delta t}^{l;m} I_h^{l;m}\big) \phi_l^{\pm,\mathrm{cvg}}\big(t_m^n\big), \qquad \text{on } \Gamma_m^\pm.$$

- If $\Delta t_l < \Delta t_m$, we need to project in time the converged solution on $\Omega_l^\pm$ computed at times t_l^n, for $n \leqslant L_l$, to get an estimate of $I_h^{l;m}\phi_l^{\pm,\mathrm{cvg}}$ at times $t_m^{(n)}$ with $n \leqslant L_m$, and where $L_l < L_m$. The corresponding interpolation operator is denoted $P_{\Delta t}^{l;m}$. We then impose

$$\mathcal{B}^\pm \phi_{m|\Gamma_m^\pm}^{(0)}\big(t_m^n\big) = \mathcal{B}^\pm \big(P_{\Delta t}^{l;m} I_h^{l;m}\big) \phi_l^{\pm,\mathrm{cvg}}\big(t_m^n\big), \qquad \text{on } \Gamma_m^\pm.$$

Unlike the imaginary-time case, the real-time method then requires the storage of the restriction to $\Gamma_l^{\pm}$ of the converged solution on $\Omega_l^{\pm}$, at any discrete time $t_l^n \leqslant T$. It is however important to notice that, in practice, we only interpolate $\phi_l^{\pm,\mathrm{cvg}}$ in the overlapping region $\Gamma_m^{\pm}$ and not in all $\Omega_m^{\pm}$. The overall process is summarized as follows. For all $n \leqslant L_l$,

1. At the lower level l: compute $\phi_l^{\pm,\mathrm{cvg}}(t_l^n)$, starting from the Cauchy data $\phi_l^{\pm}(0) = \phi_{0;l}^{\pm}$.
2. Store $\mathcal{S}_l^{\pm}$ as defined in (11).
3. At upper level $m > l$: compute $\phi_m^{\pm,\mathrm{cvg}}(t_m^n)$, with initial data $\phi_m^{\pm}(0) = \phi_{0;m}^{\pm}$ and transmission conditions $I_h^{l;m}\phi_l^{\pm,\mathrm{cvg}}$ and imposed on $\Gamma_m^{\pm}$.

Again, this algorithm should preferably be applied to two successive levels ($m = n + 1$), as it is numerically shown in Section 4.

3.3. Multilevel method and computational complexity in real-time

The methodology presented above can easily be iteratively extended to $q \geqslant 3$ levels, where $q = m - l$. We define $\Omega_p \to \Omega_{p+1}$: $\phi_p^{\mathrm{cvg}} \to \phi_{p+1}^{\mathrm{cvg}}$, with $p \in \{l, \cdots, m-1\}$, $N_l < N_{l+1} < \cdots < N_m$, and where typically $N_{p+1} \approx 2^d N_p$.

At any level $p \in \{l, \cdots, m\}$, the computational complexity for computing a convergent solution is $\mathcal{O}(k_p^{\mathrm{cvg}} N_p^{\alpha_p} K_p)$, where K_p is the number of time iterations to reach T and k_p^{cvg} is the number of Schwarz iterations to converge *without preconditioning*. Coefficient α_p typically belongs to $(1, 3)$ is related to the complexity for solving the induced sparse linear system. Assume now that the computation at level p was preconditioned by using the converged solution at level $p-1$, according to the procedure described in subsections 2.1 and 3.2. In this case, the number of Schwarz iterations to converge at the upper level p, is denoted by $k_{p;p-1}^{\mathrm{cvg}}$. It is expected that $k_{p;p-1}^{\mathrm{cvg}} \leqslant k_p^{\mathrm{cvg}}$, for all $p \leqslant m - 1$. Let us remark that the space and time interpolations from one level to another have a negligible computational complexity compared with any NLSE solution and Schwarz iterations. We can now estimate the overall complexity in real-time, which is designated[2] by $O_{l;m}^r$ (from levels l to m), of a q-level preconditioned method by

$$O_{l;m}^r = \mathcal{O}\Big(k_l^{\mathrm{cvg}} N_l^{\alpha_l} K_l + \sum_{p=l+1}^{m} k_{p;p-1}^{\mathrm{cvg}} N_p^{\alpha_p} K_p\Big). \tag{12}$$

In addition, this procedure requires the storage at any level $p \in \{l, \cdots, m-1\}$ of the converged solution $\phi_p^{\pm,\mathrm{cvg}}$ at any time and in the overlapping region of $\Omega_p^{\pm}$. The overall procedure is relevant as long as $O_{l;m}^r \ll O_m^r$, where O_m^r denotes the computational complexity of the direct method (1-level) on $\Omega_m^{\pm}$, that is if

$$O_{l;m}^r \ll O_m^r = \mathcal{O}\Big(k_m^{\mathrm{cvg}} N_m^{\alpha_m} K_m\Big).$$

We recall that $N_m = N_l/2^{d(m-l)}$ and $\alpha_m > 1$.

[2]Upper index r in $O_{l;m}^r$ stands for *real-time*.

3.4. Multilevel method and computational complexity in imaginary-time

In imaginary-time, the overall gain is expected to be higher compared to real-time. First, at any Schwarz iteration k, let us denote by $K_p^{(k)}$ the number of imaginary-time iterations for the CNGF algorithm to converge at any *unpreconditioned* level p. We also denote by $K_{p;p-1}^{(k)}$ the number of imaginary-time iterations for the CNGF algorithm to converge at level p with *preconditioning* at the lower level $p-1$, as described in subsections 2.2 and 3.1. Then, from one level $p-1$ to p, we expect that

- $k_{p;p-1}^{\text{cvg}} \leqslant k_p^{\text{cvg}}$, as in real-time, thanks to the transmission conditions,
- $K_{p;p-1}^{(k)} \ll K_p^{(k)}$, if $p \leqslant m$. This additional outstanding property is due to the fact that the interpolated solution at lower level $p-1$ is taken as the initial guess at the upper level p.

In conclusion, in imaginary-time, the overall complexity[3] $O_{l;m}^i$ of a p-level method from levels l to m is given by

$$O_{l;m}^i = \mathcal{O}\Big(N_l^{\alpha_l} \sum_{k=1}^{k_l^{\text{cvg}}} K_l^{(k)} + \sum_{p=l+1}^{m} N_p^{\alpha_p} \sum_{k=1}^{k_{p;p-1}^{\text{cvg}}} K_{p;p-1}^{(k)}\Big), \tag{13}$$

where O_m^i is the computational complexity of the direct method (1-level) on $\Omega_m^{\pm}$, i.e.

$$O_{l;m}^i \ll O_m^i = \mathcal{O}\big(N_m^{\alpha_m} \sum_{k=1}^{k_m^{\text{cvg}}} K_m^{(k)}\big).$$

We again recall that $N_m = N_l/2^{d(m-l)}$ and $\alpha_m > 1$.

4. Numerical examples

In the one-dimensional and for $a > 0$, we introduce $\Omega_a = (-a, a)$, $\Omega_{a,\varepsilon}^+ = (-a, \varepsilon/2)$ and $\Omega_{a,\varepsilon}^- = (-\varepsilon/2, a)$, where ε is a (small compared to a) parameter equal to the size of the overlapping region. Homogeneous Dirichlet boundary conditions are imposed at $\pm a$. We denote by $\{x_j\}_{j\in\{1,\cdots,N_\varepsilon^+\}}$ the grid nodes in $\Omega_{a,\varepsilon}^+$ and $\{y_j\}_{j\in\{1,\cdots,N_\varepsilon^-\}}$ those in $\Omega_{a,\varepsilon}^-$. In the following tests, the domains overlap on o nodes such that: $x_{N_\varepsilon^+} = y_{1+o}$ and $x_{N_\varepsilon^+-o} = y_1$. The spatial mesh size $h = h_0$ is assumed to be constant and then $\varepsilon = (o-1)h$.

4.1. Discrete SWR methods in real- & imaginary-time

At a given level and in real-time, we consider the following Crank-Nicolson scheme [3]. Denoting $\phi^{\pm,n,(k)}$ the approximate solution in $\Omega^{\pm}$ at time t_n with $n \geqslant 0$ and at Schwarz iteration $k \geqslant 0$, we get

[3] Upper index i in $O_{l;m}^i$ stands for *imaginary-time*.

- For the SWR DDM (denoted by CSWR-DDM)

$$\left\{\begin{array}{rcl} \mathtt{i}\dfrac{\phi^{\pm,n+1,(k)}-\phi^{\pm,n,(k)}}{\Delta t} & = & -\partial_x^2\dfrac{\phi^{\pm,n+1,(k)}+\phi^{\pm,n,(k)}}{2}+V(x)\dfrac{\phi^{\pm,n+1,(k)}+\phi^{\pm,n,(k)}}{2} \\ & & +\kappa|\phi^{\pm,n+1,(k)}+\phi^{\pm,n,(k)}|^2\dfrac{\phi^{\pm,n+1,(k)}+\phi^{\pm,n,(k)}}{8}=0, \ \text{ in } \Omega^{\pm}_{a,\varepsilon}, \\ \phi^{\pm,n+1,(k)}_{\pm\varepsilon/2} & = & \phi^{\mp,n+1,(k-1)}_{\pm\varepsilon/2}, \ \text{ on } \big\{\pm\varepsilon/2\big\}. \end{array}\right.$$

- For the SWR DDM with Robin-type transmission conditions

$$\big(\partial_{\mathbf{n}^\pm}+\gamma\big)\phi^{n+1,(k)}_{\pm,\varepsilon/2}=\big(\partial_{\mathbf{n}^\pm}+\gamma\big)\phi^{\mp,n+1,(k-1)}_{\varepsilon/2}, \ \text{ on } \big\{\pm\varepsilon/2\big\},$$

  for some given parameter $\gamma\in\mathbb{R}^*$ and where $\mathbf{n}^\pm$ denotes the time index.

In imaginary-time, the basic principle consists in replacing $t\to\mathtt{i}t$ and $\Delta t\to\mathtt{i}\Delta t$ in the above scheme. The second derivative is approximated by using a 3 point-scheme. We again refer to [7] for details about SWR methods in imaginary-time. As the CNGF method is a minimization technique, it is important to note that at each Schwarz iteration a convergence criterion has to be imposed to ensure the convergence of the minimizer. More specifically, denoting by $\widetilde{\phi}^{(k)}$ the imaginary-time solution of (3) at Schwarz iteration k in Ω_a, the imaginary-time iterations are stopped when, for $n\geqslant 0$

$$\|\widetilde{\phi}^{(k)}(\cdot,t^-_{n+1})-\widetilde{\phi}^{(k)}(\cdot,t_n)\|_{L^2(\Omega_a)}\leqslant\delta, \tag{14}$$

where δ is a very small positive parameter.

4.2. Numerical tests in imaginary-time

In this series of tests, we consider a one-dimensional optical lattice with potential $V(x)=x^2/2+25\sin^2(\pi x/2)$ and nonlinearity strenght $\kappa=250$. The initial data is given by $\phi_0(x)=\exp(-x^2/2)\pi^{-1/4}$. The two subdomains are $\Omega^+_{a,\varepsilon}=(-a,\varepsilon/2)$ and $\Omega^-_{a,\varepsilon}=(-\varepsilon/2,a)$, with $a=16$. The DDM algorithms are the CSWR and Robin-type SWR methods. The coarsest level has $N_0=2^7$ nodes and $h_0=(b-a)/(N_0-1)$. This level is refereed to as level 0. We successively compare the rate of convergence or residual history (8), at level $i>0$, i.e. on upper levels Ω_i, with $h_i\approx h_0/2^i$ and $i=1,2,3$. We then have $N_i=2^iN_0$, $N_i^+=o+2^{i-1}N_0$ and $N_i^-=2^{i-1}N_0$. The overlapping region covers two nodes ($o=2$): $\varepsilon_i=(o-1)h_i=h_i$, $i=1,2,3$. The time steps are equal to $\Delta t_1=5\times10^{-2}$, $\Delta t_2=2\times10^{-2}$ and $\Delta t_3=5\times10^{-3}$. The convergence parameter in (14) for the CNGF is fixed to $\delta=10^{-9}$.

Test-case 1. In the first test-case dedicated to the CSWR method, we assume that the preconditioning is derived from level 0, i.e., at level 0 (or in Ω_0) $f_{g;0}$ is first computed. As proposed in Section 2.2, we impose $\phi_0^\pm=I_h^{0;i}f^\pm_{p;0}$ at the upper level i, with $i=1,2$, i.e. at $\{\pm\varepsilon_i/2\}\times\{t>0\}$, we impose

$$\phi_i^{\pm,(0)}=I_h^{0;i}f^\pm_{p;0},$$

where $f_{g;0}$ is the ground state computed on $\Omega_0^{\pm}$. In Test 1, the size of the overlapping region is always reduced to $\varepsilon = h_i$, $i = 1, 2$. Convergence results (residual history) are reported in Fig. 1 (left). These correspond to estimates of k_1^{cvg}, k_2^{cvg}, $k_{1;0}^{\text{cvg}}$ and $k_{2;0}^{\text{cvg}}$ defined in (8). In addition, we provide in Fig. 1 (right) the convergence times $T^{(k)}$ of the CNGF per Schwarz iteration, i.e. for $i = 1, 2$,

- $K_{i;0}^{(k)} \Delta t_i$: time step $\times$ number of CNGF iterations $K_{i;0}^{(k)}$ to converge at level i $(= 1, 2)$ with preconditioning at level 0,
- $K_i^{(k)} \Delta t_i$: time step $\times$ number of CNGF iterations $K_i^{(k)}$, without preconditioning.

We observe on Fig. 1 (left) that the preconditioning from lower levels $i-1$ or $i-2$ has only a weak effect on the acceleration of the convergence of the CSWR method. Regarding the convergence time of the CNGF method, we however notice that i) for all k, $K_{1;0}^{(k)} \ll K_1^{(k)}$, and that ii) $K_{2;0}^{(k)} \ll K_2^{(k)}$ for the first CSWR iterations and then $K_{2;0}^{(k)} \approx K_2^{(k)}$. This test illustrates that the convergence acceleration of the DDM-CNGF method thanks to the preconditionning at a lower level.

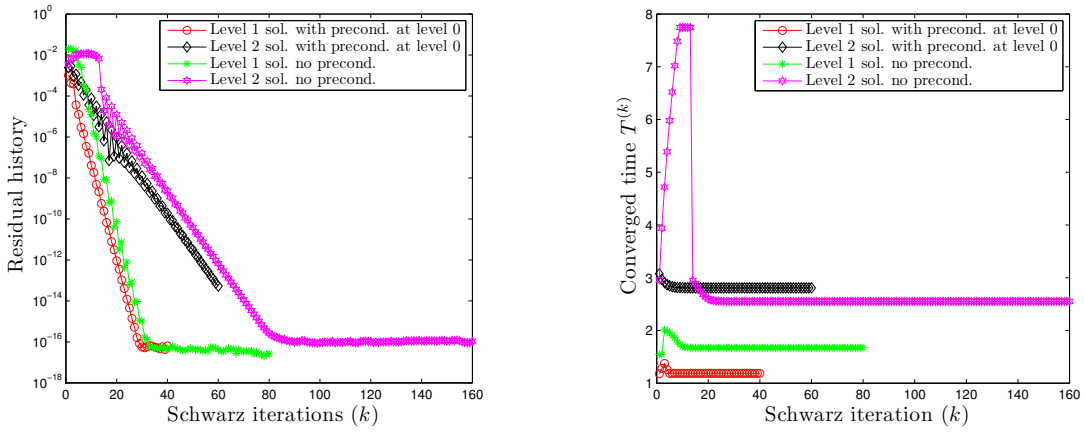

Figure 1: Levels 1 and 2. Left: Comparison of the residual history (8) vs k, with/without preconditioning at the coarse level 0. Right: Minimization time (convergence time of the CNGF) $T^{(k)}$ per Schwarz iteration k, for $N_0 = 2^7$.

Test-case 2. In the second test-case, we compare the residual history (8) of the CSWR method at level $i \geqslant 1$, that is for $\Omega_i^{\pm}$, with and without preconditioning at level $i-1$ for different spatial discretization step sizes. We impose i) initially $\phi_i(0) = I_h^{i-1;i} f_{g;i-1}$, and ii) at $\{\pm\varepsilon_i/2\}$, we force

$$\phi_i^{\pm,(0)} = I_h^{i-1;i} f_{g;i-1}^{\pm}.$$

In other words, we compare $k_{i;i-1}^{\text{cvg}}$ with k_i^{cvg}, for $i = 1, 2, 3$. We also study the convergence times $T^{(k)}$ of the minimization algorithm i) $K_{i;i-1}^{(k)} \Delta t_i$ (with preconditioning) and $K_i^{(k)} \Delta t_i$ (without preconditioning). In the previous expressions, $K_{i;i-1}^{(k)}$ (resp. $K_i^{(k)}$) is the number of iterations of the CNGF to converge at level i with (resp. without) preconditioning, for $i = 1, 2, 3$ (see Section 3.4). At the coarse level $i-1 \geqslant 0$, $\phi_{i-1}^{\pm}(0)$ is chosen as the projection on $\Omega_{i-1}^{\pm}$ of $\phi_0(x) = \pi^{-1/4} e^{-x^2/2}$ for

computing the ground state of the NLSE [11]. We consider the standard homogeneous Dirichlet boundary conditions when $k=0$: $\phi_i^{\pm,(0)}=0$ at $\{\pm\varepsilon_i/2\}$. We report on Fig. 2 the residual history (8) with respect to the lower level preconditioning, for different values of grid points $N_i=2^{7+i}$. We observe that $k_{1;0}^{\mathrm{cvg}}\leqslant k_1^{\mathrm{cvg}}$, $k_{2;1}^{\mathrm{cvg}}\leqslant k_2^{\mathrm{cvg}}$, $k_{3;2}^{\mathrm{cvg}}\leqslant k_3^{\mathrm{cvg}}$. However, the Schwarz acceleration is moderate and seems independent of the number of grid points.

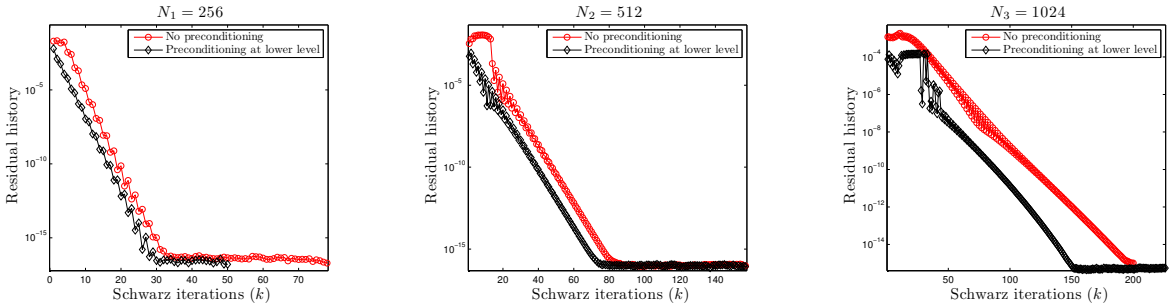

Figure 2: Comparison of the residual history (8) vs k, of the CSWR method at level i, for a solution with preconditioning at level $i-1$, and without preconditioning. Left: $i=1$. Middle: $i=2$. Right: $i=3$.

The genuine gain is again the acceleration of the CNGF algorithm. Indeed, the minimization algorithm at fixed Schwarz iteration k and finer level i is strongly accelerated as it can be observed in Fig. 3 and as it was expected from Section 3: $K_{i;i-1}^{(k)}\ll K_i^{(k)}$, with convergence times given by $K_i^{(k)}\Delta t_i$ without preconditioning, or $K_{i;i-1}^{(k)}\Delta t_i$ with preconditioning.

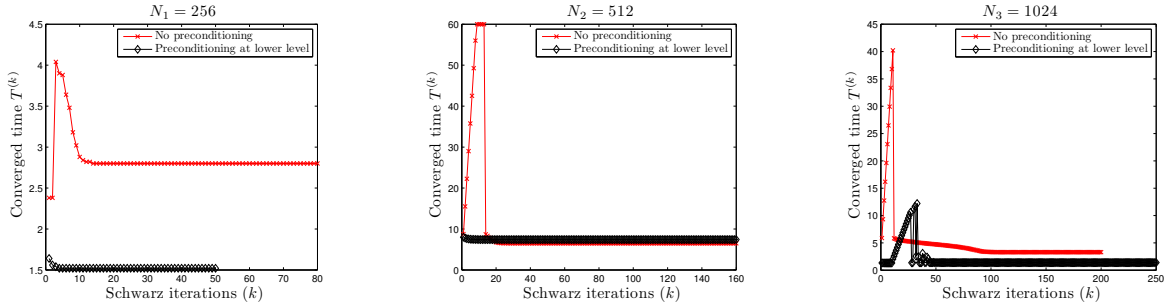

Figure 3: Minimization time (CNGF convergence time) $T^{(k)}$: comparison for the CSWR method at level i between the solution with preconditioning at level $i-1$, and without preconditioner. Left: $i=1$. Middle: $i=2$. Right: $i=3$.

Test-case 3. Now, the preconditioning technique is applied to the Robin-based Schwarz Waveform Relaxation algorithm, taking $\gamma=20$, which provides a fast convergence. The methodology and numerical data are the same as for the test case 2. We compare i) the residual history (8) at level $i\geqslant 1$, that is on $\Omega_i^{\pm}$, with preconditioning at level $i-1$. The number of grid points at level i is given by $N_i=2^{7+i}$. Therefore, we i) initially ($t=0$) take $\phi_i^{\pm}(0)=I_h^{i-1;i}f_{g;i-1}^{\pm}$ and ii) we impose

$$\big(\partial_x+\gamma\big)\phi_i^{\pm,(0)}(t_i^n)=\big(\partial_x+\gamma\big)I_h^{i-1;i}f_{g;i-1}^{\pm}$$

at $\big\{\pm\varepsilon_i/2\big\}$. We also study the convergence times $T^{\mathrm{cvg},k}$ of the minimization algorithm, with and without preconditioning. The results are summarized in Figs. 4 and 5.

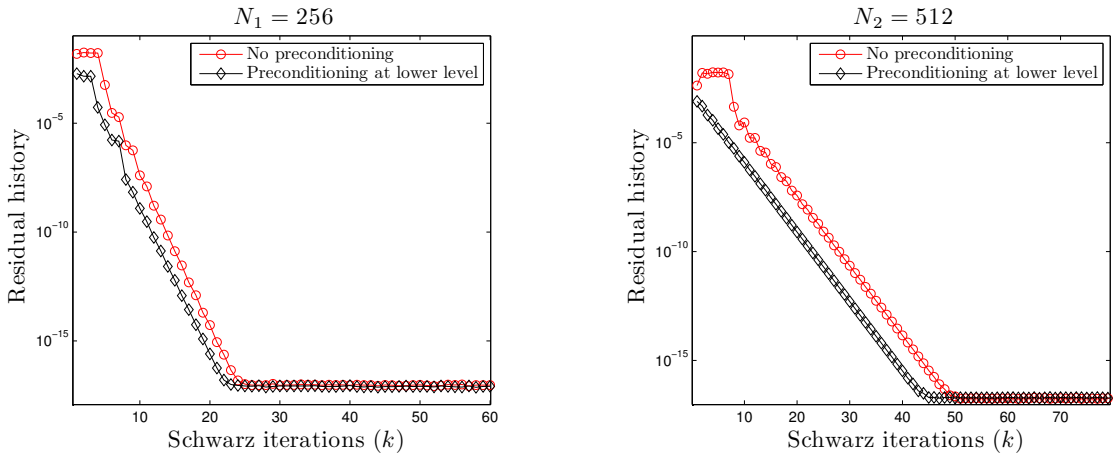

Figure 4: Comparison of the residual history (8) vs k, for the Robin-based SWR method at level i, between the solution with preconditioning at level $i-1$ and without preconditioning. Left: $i = 1$. Right: $i = 2$.

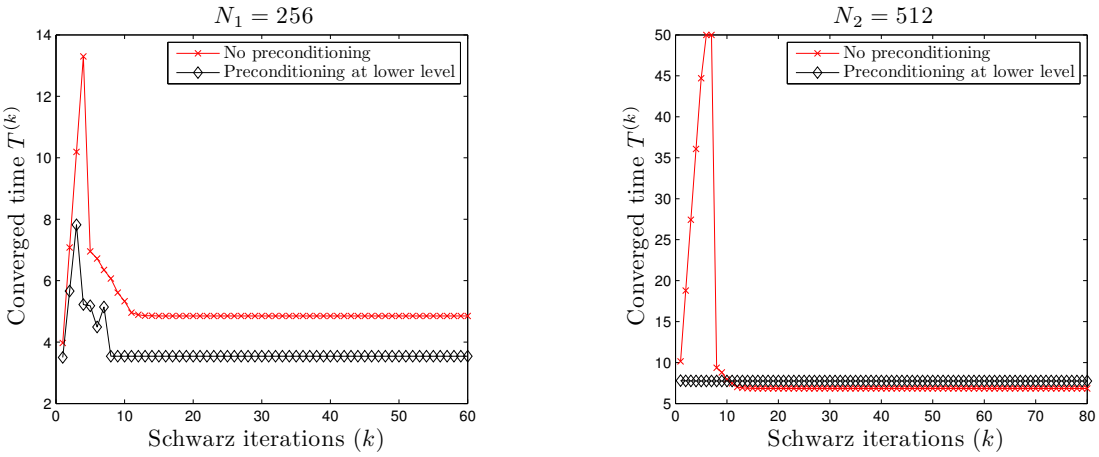

Figure 5: Minimization time (CNGF convergence time) $T^{(k)}$ comparison for Robin-based SWR at level i between solution with preconditioning at level $i-1$, and without preconditioning. Left: $i = 1$. Right: $i = 2$.

As for the CSWR method, it is numerically observed that the preconditioning technique applied to the Robin SWR method has a strong positive effect on the convergence of the CNGF, but a moderate one from the SWR convergence point of view.

4.3. Numerical tests in real-time

This section is devoted to experiments in real time. We consider $\Omega^+_{a,\varepsilon} = (-a, 5/2 + \varepsilon/2)$ and $\Omega^-_{a,\varepsilon} = (5/2 - \varepsilon/2, a)$, with $\varepsilon > 0$ and $a = 10$. Homogeneous Dirichlet boundary conditions are again imposed at $\pm a$. The final real time is $T = 0.5$. In the equation, we have chosen $\kappa = 50$ and $V = 0$, corresponding to a standard cubic NLSE. In addition, the initial data is given by a gaussian

profile

$$\phi_0(x) = \exp\Big(-\frac{1}{5}\big(\frac{b+2a}{4} - x\big)^2\Big)\exp(2ix).$$

Test-case 1. In this first test case, the numerical data are as follows: $N_1 = 2N_0$, $N_1^+ = o + N_0$, $N_1^- = N_0$, with $N_0 = 400$. The overlapping region covers respectively 20, 10 and 2 nodes. The time step is fixed to $\Delta t_1 = \Delta t_0 = 1 \times 10^{-2}$.

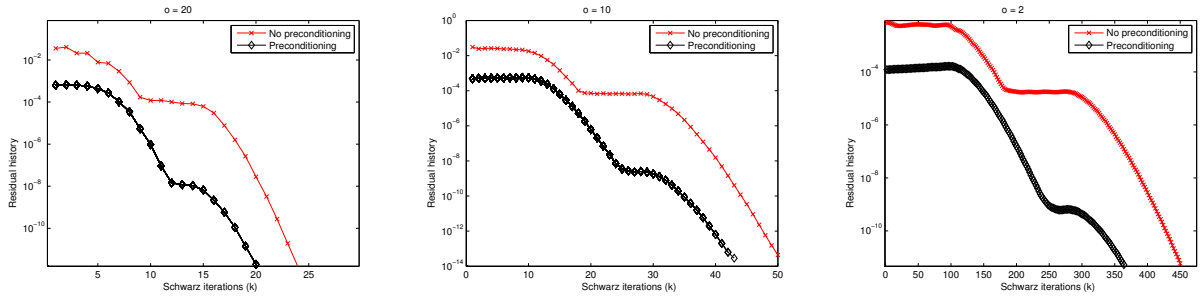

Figure 6: Comparison of the residual history (4) vs k, for the CSWR method, with and without preconditioning for $N_0 = 400$. Left: $o = 20$. Middle: $o = 10$. Right: $o = 2$.

Figs. 6 illustrate the effect of the acceleration of the preconditioning on the convergence of the CSWR (8), in the three studied cases $N_0 = 400$ and: $o = 20$, 10, 2 and $\varepsilon = (o-1)h_i$. Let us note that with or without preconditioning, the convergence graphs have two plateaux and two decreasing regions. More specifically, we notice that the main effect of the preconditioning is to extend the first decreasing zone and to reduce the length of the second plateau. We finally remark that the preconditioning allows for a reduction of the number of iterations to reach the machine tolerance of about 20%, whatever the size of the overlapping region is (see Fig. 6).

Test-case 2. In the second test-case, we consider the SWR method with Robin-based transmission conditions. In this case, we expect a faster convergence compared to the CSWR method [21]. Notice that for this type of transmission condition, it is necessary to reconstruct the normal derivatives at the subdomain interfaces. The numerical data are as follows: $N_i = 2^i N_0$, $N_i^+ = o + 2^{i-1}N_0$ and $N_i^- = 2^{i-1}N_0$, with overlap $o = 2$ and $N_0 = 200$ and $\varepsilon = (o-1)h_i$. The time step is fixed to $\Delta t_1 = \Delta t_0 = 1 \times 10^{-2}$, for all $i = 1, 2$. We impose at the subdomain interfaces, and for all $n \geqslant 0$

$$\big(\partial_x \pm \gamma\big)\phi_m^{\pm,(0)} = \big(\partial_x \pm \gamma\big) I_h^{n;m} \phi_l^{\pm,\mathrm{cvg}}, \qquad \text{at } \big\{\pm \varepsilon_i/2\big\},$$

with $\gamma = 20$. We compare the residual history (8) for the CSWR and Robin-based SWR algorithms with and without preconditioning. As expected, the convergence is faster for the Robin-based SWR methods than for the CSWR algorithm. The results on Fig. 7 also show that preconditioning the Robin SWR method also improves the convergence. As for the CSWR method, the effect of preconditioning is to extend the first decay zone and to reduce the length of the second plateau. The global gain is not as high as for the CSWR method and is about 15%.

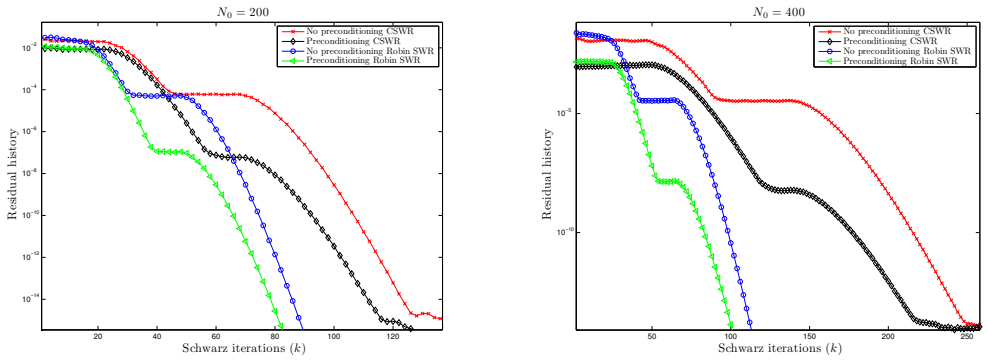

Figure 7: Comparison of the residual history (4) vs k, of the CSWR and Robin-based SWR methods, with and without preconditioning, for $o = 2$. Left: $N_0 = 200$. Right: $N_0 = 400$.

5. Conclusion

We proposed and numerically tested a simple preconditioning technique for accelerating SWR algorithms applied to the solution of NLSE both in real- and imaginary-time. The general principle consists, by using approximate solutions computed on coarser grids (lower levels), in designing i) suitable SWR transmission conditions, as well as ii) adapted initial data in imaginary-time. Due to its simplicity and efficiency, the presented approach can easily be included in a parallel SWR-DDM solver for the NLSE in real- or imaginary-time. In a forthcoming paper, the procedure developed in this work will be implemented in higher dimensions and tested on more realistic situations.

References

[1] X. Antoine, A. Arnold, C. Besse, M. Ehrhardt, and A. Schädle. A review of transparent and artificial boundary conditions techniques for linear and nonlinear Schrödinger equations. *Commun. Comput. Phys.*, 4(4):729–796, 2008.

[2] X. Antoine, W. Bao, and C. Besse. Computational methods for the dynamics of the nonlinear Schrödinger/Gross-Pitaevskii equations. *Comput. Phys. Comm.*, 184(12):2621–2633, 2013.

[3] X. Antoine, C. Besse, and S. Descombes. Artificial boundary conditions for one-dimensional cubic nonlinear Schrödinger equations. *SIAM J. Numer. Anal.*, 43(6):2272–2293 (electronic), 2006.

[4] X. Antoine and R. Duboscq. GPELab, a Matlab toolbox to solve Gross-Pitaevskii equations I: Computation of stationary solutions. *Comput. Phys. Comm.*, 185(11):2969–2991, 2014.

[5] X. Antoine and R. Duboscq. Robust and efficient preconditioned Krylov spectral solvers for computing the ground states of fast rotating and strongly interacting Bose-Einstein condensates. *J. of Comput. Phys.*, 258C:509–523, 2014.

[6] X. Antoine and R. Duboscq. *Modeling and computation of Bose-Einstein condensates: stationary states, nucleation, dynamics, stochasticity.* in Nonlinear Optical and Atomic Systems:

at the Interface of Mathematics and Physics, CEMPI Subseries, 1st Volume, Lecture Notes in Mathematics. Springer, to appear, 2015.

[7] X. Antoine and E. Lorin. An analysis of Schwarz waveform relaxation domain decomposition methods for the imaginary-time linear Schrödinger and Gross-Pitaevskii equations. *Submitted*, 2015.

[8] X. Antoine and E. Lorin. Lagrange-Schwarz waveform relaxation domain decomposition methods for linear and nonlinear quantum wave problems. *Appl. Math. Lett.*, 2016.

[9] X. Antoine, E. Lorin, and A.D. Bandrauk. Domain decomposition method and high-order absorbing boundary conditions for the numerical simulation of the time dependent Schrödinger equation with ionization and recombination by intense electric field. *J. of Sc. Comput.*, 64(3):620–646, 2015.

[10] W. Bao and Y. Cai. Mathematical theory and numerical methods for Bose-Einstein condensation. *Kinetic and Related Models*, 6(1):1–135, 2013.

[11] W. Bao and Q. Du. Computing the ground state solution of Bose-Einstein condensates by a normalized gradient flow. *SIAM J. Sci. Comput.*, 25(5):1674–1697, 2004.

[12] C. Besse and F. Xing. Schwarz waveform relaxation method for one-dimensional Schrödinger equation with general potential. *submitted*, 2015.

[13] V. Dolean, M. J. Gander, and L. Gerardo-Giorda. Optimized Schwarz methods for Maxwell's equations. *SIAM J. Sci. Comput.*, 31(3):2193–2213, 2009.

[14] M. El Bouajaji, V. Dolean, M. J. Gander, and S. Lanteri. Optimized Schwarz methods for the time-harmonic Maxwell equations with damping. *SIAM J. Sci. Comput.*, 34(4):A2048–A2071, 2012.

[15] M. Gander and L. Halpern. Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems. *SIAM J. Num. Anal.*, 45(2), 2007.

[16] M.J. Gander. Optimal Schwarz waveform relaxation methods for the one-dimensional wave equation. *SIAM J. Numer. Anal.*, 41:1643–1681, 2003.

[17] M.J. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44:699–731, 2006.

[18] M.J. Gander. Optimized Schwarz waveform relaxation methods for advection diffusion problems. *SIAM J. Numer. Anal.*, pages 666–697, 2007.

[19] M.J. Gander, L. Halpern, and F. Nataf. Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation. page 1999.

[20] D. Givoli. High-order local non-reflecting boundary conditions: a review. *Wave Motion*, 39(4):319–326, 2004.

[21] L. Halpern and J. Szeftel. Optimized and quasi-optimal Schwarz waveform relaxation for the one-dimensional Schrödinger equation. *Math. Models Methods Appl. Sci.*, 20(12):2167–2199, 2010.

[22] E. Lorin, X. Yang, and X. Antoine. Frozen gaussian approximation based domain decomposition methods for the linear and nonlinear Schrödinger equation beyond the semi-classical regime. *submitted*, 2015.

[23] S. Tsynkov. Numerical solution of problems on unbounded domains. A review. *Applied Numerical Mathematics*, 27(4):465–532, 1998.