Multilevel preconditioning technique 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 multilevel Schwarz Waveform Relaxation (SWR) Domain Decomposition Methods (DDM) in real- and imaginary-time for the NonLinear Schrödinger Equation (NLSE). In imaginary-time, it is shown that 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-DDM 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) that models many physics problems, including nonlinear optics and Bose-Einstein condensates [1,

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.

2, 3, 4, 5, 6, 7]. The proposed method is also applicable to the Linear Schrödinger Equation (LSE) in real- and imaginary-time, in particular for solving intense and short laser-molecule interaction including ionization processes. In this framework real-space numerical methods are largely used, see [8, 9, 10, 11], and DDM method is then the center of main interests. Domain decomposition SWR methods for solving wave equations have a long history from the classical SWR method with overlapping zones to optimal version without overlap (see e.g. [12, 13, 8, 14, 15, 16, 17, 18, 19, 20] 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 (Optimal SWR), or Perfectly Matched Layers [21, 8, 22]. We also refer to [21, 23, 24, 25] 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 [26, 27, 15]. In this paper, we focus on multilevel SWR for the NLSE. More specifically, we consider the cubic time-dependent (real-time) NLSE set on $\mathbb{R}^d$, with $d \geq 1$,

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

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, $\mathcal{P}(|\phi|)$ denotes the nonlinear operator

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

Compared to the real-time dynamics, the imaginary-time formulation [1, 2, 3, 4, 5] is used to compute the stationary solutions to the NLSE. The corresponding method is referred to as Continuous Normalized Gradient Flow (CNGF) formulation [1, 3, 4, 5] in the Mathematics literature and imaginary-time method in the Physics literature. The current paper is an extension of [12] 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 acceleration of the convergence 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 multilevel 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 $\Omega_\epsilon^\pm$ such that $\mathbb{R}^d = \Omega_\epsilon^+ \cup \Omega_\epsilon^-$, with overlapping region $\Omega_\epsilon^+ \cap \Omega_\epsilon^-$, where ϵ is a (small) non-negative parameter. In 1-d ($d = 1$), the domains of interest read: $\Omega_\epsilon^+ = \big(-\infty, \epsilon/2\big)$, $\Omega_\epsilon^- = \big(-\epsilon/2, \infty\big)$ and $\mathbb{R} = \Omega_\epsilon^+ \cup \Omega_\epsilon^-$. with $\Omega_\epsilon^+ \cap \Omega_\epsilon^- = \big(-\epsilon/2, \epsilon/2\big)$. We denote by $\phi^\pm$ the solution to the time-dependent GPE in $\Omega_\epsilon^\pm$. Solving the NLSE by a Schwarz waveform domain decomposition [8] requires some transmission conditions at the subdomain interfaces. More specifically, for any Schwarz iteration $k \geq 1$, the equation in $\Omega_\epsilon^\pm$ reads, for a given $T > 0$,

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

where $\Gamma_\epsilon^\pm = \partial\Omega_\epsilon^\pm$. The notation $\phi^{\pm,(k)}$ stands for the solution $\phi^\pm$ in $\Omega_\epsilon^\pm \times (0,T)$ at Schwarz iteration k. Initially, $\phi^{\pm,(0)}$ are two given functions defined in $\Omega_\epsilon^\pm$. 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 optimal SWR algorithm, $\mathcal{B}^\pm$ can be a local or a nonlocal approximation of the DtN operator (see [8, 22]). The convergence criterion for the Schwarz DDM is given by

$$\Big\| \ \|\phi^{+,(k)}_{|\Gamma_\epsilon^+} - \phi^{-,(k)}_{|\Gamma_\epsilon^-}\|_{\infty,\Gamma_\epsilon}\Big\|_{L^2(0,T)} \leq \delta^{\mathrm{Sc}}. \tag{4}$$

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 in real time, is given 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 [5] 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

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

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 $E_\kappa(\phi) := \min_{||\chi||_{L^2(\mathbb{R}^d)}=1} E_\kappa(\chi)$. 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 [1, 3, 4, 5] which consists in solving (1) in imaginary-time, i.e. setting $t \to \mathtt{i}t$. This leads to the iterative algorithm

$$\begin{cases} \partial_t\phi(\mathbf{x},t) = -\nabla_{\phi^*}E_\kappa(\phi) = \triangle\phi(\mathbf{x},t) - V(\mathbf{x})\phi(\mathbf{x},t) - \kappa|\phi|^2\phi(\mathbf{x},t), \ \mathbf{x} \in \mathbb{R}^d, \ t_n < t < t_{n+1}, \\ \phi(\mathbf{x},t_{n+1}) := \phi(\mathbf{x},t^+_{n+1}) = \dfrac{\phi(\mathbf{x},t^-_{n+1})}{||\phi(\cdot,t^-_{n+1})||_{L^2(\mathbb{R}^d)}}, \\ \phi(\mathbf{x},t=0) = \phi_0(\mathbf{x}), \ \mathbf{x} \in \mathbb{R}^d, \text{with } ||\phi_0||^2_{L^2(\mathbb{R}^d)} = 1. \end{cases} \tag{6}$$

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 [5] for a given positive potential V and a positive interaction strength κ.

Within the SWR formalism, we then have to minimize an energy at each Schwarz iteration $k \geq 1$. This is achieved at an imaginary-time, denoted by $T^{(k)} > 0$, where for $t > T^{(k)}$, the error $\|\phi^\pm(\cdot,T^{(k)}) - \phi^\pm(\cdot,t)\|_{L^2(\mathbb{R}^d)}$ is small enough. For a two-subdomains decomposition, we then solve

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

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^-). Regarding the CNGF convergence criterion for a given Schwarz iteration k, we stop the computation when the error between the two numerical solutions $\phi^{n,(k)}$ and $\phi^{n+1,(k)}$ respectively related to two successive time steps t_n and t_{n+1} is small enough

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

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: $T^{(k)} = T^{\text{cvg},(k)} := n^{\text{cvg},(k)}\Delta t$ for a converged solution $\phi^{\text{cvg},(k)}$ reconstructed from the two subdomains solutions $\phi^{\pm,\text{cvg},(k)}$. The convergence criterion for the Schwarz DDM is set by the constraint

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

The convergence of the whole iterative algorithm is obtained at Schwarz iteration k^{cvg}. Then, one gets the converged global solution in imaginary time which is denoted $\phi^{\text{cvg}} := \phi^{\text{cvg},(k^{\text{cvg}})}$ (e.g. with $\delta^{\text{Sc}} = 10^{-14}$). Notice that the converged solution in real and imaginary time are both denoted by ϕ^{cvg}. However the iterative algorithm in imaginary is naturally more complex as it contains a CNFG process. The framework (real or imaginary time) allows for avoiding any confusion about the meaning of ϕ^{cvg}.

2.3. Notations and discretization

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

$$\Omega_\ell = \cup_{(i,j,k)\in\mathbb{Z}^3}\big[ih_\ell, (i+1)h_\ell\big] \times \big[jh_\ell, (j+1)h_\ell\big] \times \big[kh_\ell, (k+1)h_\ell\big].$$

The operator $P_h^{m;\ell}$ designates a projection operator from the grid Ω_m to Ω_ℓ, where $m > \ell$. In practice, $P_h^{m;\ell}$ is an average operator defined iteratively from

Ω_m to Ω_ℓ. Conversely, we introduce $I_h^{\ell;m}$, for $m > \ell$, as a polynomial interpolation operator from Ω_ℓ to Ω_m. In practice, the standard Lagrangian interpolation is used in this paper. Moreover, $\Omega^{\pm}_{\epsilon,\ell}$ now denotes the uniform subgrid such that $\Omega_\ell = \Omega^{+}_{\epsilon,\ell} \cup \Omega^{-}_{\epsilon,\ell}$ on which are defined the semi-discrete time-dependent solutions $\phi_\ell^{\pm,(k)}(t) = \left\{\phi_{\ell,\mathbf{j}}^{\pm,(k)}(t)\right\}_{\mathbf{j}}$ at Schwarz iteration k and where $\mathbf{j}$ is a multi-index in $\mathbb{Z}^d$, which is denoted by $j \in \mathbb{Z}$ when $d = 1$. For the sake of simplicity we also use the notation $\phi_\ell^{\pm,(k)} = \left\{\phi_{\ell,\mathbf{j}}^{\pm,(k)}\right\}_{\mathbf{j}}$ for denoting $\phi_\ell^{\pm,(k)}(t)$ and we now set $\Omega_\ell^{\pm} = \Omega^{\pm}_{\epsilon,\ell}$. The reconstructed solution $\phi_\ell^{(k)}$ on Ω_ℓ is

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

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

3. Multilevel 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 $\Omega_\ell^{\pm}$, with $m > \ell$.

3.1. Two-level SWR method in imaginary-time

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

$$\mathcal{P}_\ell(|f_{p;\ell}|)f_{p;\ell} = \lambda_{p;\ell} f_{p;\ell},$$

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

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

To this end, we use i) $f_{p;\ell}$ and ii) the interpolation operator $I_h^{\ell;m}$. At level $m > \ell$ ($h_m < h_\ell$) and for all $k \geq 0$, the initial guess is then built using $I_h^{\ell;m} f_{p;\ell}$ on $\Omega_m^{\pm}$. 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

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

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

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

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 $f_{p;\ell}$, 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 $m > \ell$, we perform

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

The expected gain is not the convergence acceleration of the SWR algorithm, 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 $m = \ell + 1$.

3.2. Two-level SWR method in real-time

The method proposed above can be directly adapted from the imaginary- to real-time. However, although the relative computational workload is roughly the same compared to imaginary-time, additional data-storage is necessary as seen below. Let us first assume that the converged solution to (1) is computed on $\Omega_\ell^{\pm} \times [0, T]$ and its restriction to the interface $\Gamma_\ell^{\pm}$ is stored in $\mathcal{S}_\ell^{\pm}$, defined by

$$\mathcal{S}_\ell^{\pm} = \big\{\phi_\ell^{\pm,\text{cvg}}\big(t_\ell^n\big) \text{ at } \Gamma_\ell^{\pm}, \forall n \in \{0, \cdots, L_\ell\}\big\}, \tag{11}$$

where i) $\phi_\ell^{\pm,\text{cvg}}$ is the converged solution on $\Omega_\ell^{\pm}$ and ii) $t_\ell^n = n\Delta t_\ell$, with $n \in \{0, \cdots, L_\ell\}$, $\Delta t_\ell L_\ell = T$. These points are used for preconditioning the Schwarz algorithm. More specifically on $\Omega_m^{\pm}$, at Schwarz iteration $k \geq 1$, 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 fixed 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 $\Delta t_\ell = \Delta t_m$, we impose

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

- If $\Delta t_\ell > \Delta t_m$, it is necessary to interpolate in time the converged solution on $\Omega_\ell^\pm$ computed at times t_ℓ^n, for $n \leq L_\ell$, to get an estimate of $I_h^{\ell;m}\phi_\ell^{\pm,\text{cvg}}$ at times t_m^n, with $n \leq L_m$, and where $L_m < L_\ell$. The corresponding interpolation operator is denoted by $I_{\Delta t}^{\ell;m}$ and we then impose

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

- If $\Delta t_\ell < \Delta t_m$, we need to project in time the converged solution on $\Omega_\ell^\pm$ computed at times t_ℓ^n, for $n \leq L_\ell$, to get an estimate of $I_h^{\ell;m}\phi_\ell^{\pm,\text{cvg}}$ at times t_m^n with $n \leq L_m$, and where $L_\ell < L_m$. The corresponding interpolation operator is denoted by $P_{\Delta t}^{\ell;m}$. We then impose

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

Unlike the imaginary-time case, the real-time method then requires the storage of the restriction to $\Gamma_\ell^\pm$ of the converged solution on $\Omega_\ell^\pm$, at any discrete time $t_\ell^n \leq T$. It is however important to notice that, in practice, we only interpolate $\phi_\ell^{\pm,\text{cvg}}$ in the overlapping region $\Gamma_m^\pm$ and not in all $\Omega_m^\pm$. The overall process is summarized as follows. For all $n \leq L_\ell$,

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

Again, this algorithm should preferably be applied to two successive levels ($m = \ell + 1$), 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 $q \geq 3$ levels, where $q = m - \ell$. We define $\Omega_p \to \Omega_{p+1}$: $\phi_p^{\text{cvg}} \to \phi_{p+1}^{\text{cvg}}$, with $p \in \{\ell, \cdots, m-1\}$, $N_\ell < N_{\ell+1} < \cdots < N_m$, and where typically $N_{p+1} \approx 2^d N_p$.

At any level $p \in \{\ell, \cdots, m\}$, the computational complexity for computing a convergent solution is $\mathcal{O}(k_p^{\text{cvg}} N_p^{\alpha_p} K_p)$, 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*. The coefficient α_p, that 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 $k_{p;p-1}^{\text{cvg}}$. It is expected that $k_{p;p-1}^{\text{cvg}} \leq k_p^{\text{cvg}}$, for all $p \leq m-1$. Let us remark that the required 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 $O^r_{\ell;m}$ (from levels ℓ to m), of a q-level preconditioned method by

$$O^r_{\ell;m} = \mathcal{O}\Big(k_\ell^{\mathrm{cvg}} N_\ell^{\alpha_\ell} K_\ell + \sum_{p=\ell+1}^{m} k_{p;p-1}^{\mathrm{cvg}} N_p^{\alpha_p} K_p\Big). \tag{12}$$

In addition, this procedure requires the storage at any level $p \in \{\ell, \cdots, m-1\}$ 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 $O^r_{\ell;m} \ll O^r_m$, where O^r_m denotes the computational complexity of the direct method (1-level) on $\Omega_m^{\pm}$, that is if

$$O^r_{\ell;m} \ll O^r_m = \mathcal{O}\Big(k_m^{\mathrm{cvg}} N_m^{\alpha_m} K_m\Big).$$

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

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

- $k_{p;p-1}^{\mathrm{cvg}} \leq k_p^{\mathrm{cvg}}$, as in real-time, thanks to the transmission conditions,
- $K_{p;p-1}^{(k)} \ll K_p^{(k)}$, if $p \leq m$. 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] $O^i_{\ell;m}$ of a p-level method from levels ℓ to m is given by

$$O^i_{\ell;m} = \mathcal{O}\Big(N_\ell^{\alpha_\ell} \sum_{k=1}^{k_\ell^{\mathrm{cvg}}} K_\ell^{(k)} + \sum_{p=\ell+1}^{m} N_p^{\alpha_p} \sum_{k=1}^{k_{p;p-1}^{\mathrm{cvg}}} K_{p;p-1}^{(k)}\Big), \tag{13}$$

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

$$O^i_{\ell;m} \ll O^i_m = \mathcal{O}\big(N_m^{\alpha_m} \sum_{k=1}^{k_m^{\mathrm{cvg}}} K_m^{(k)}\big).$$

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

[2] The upper index r in $O^r_{\ell;m}$ stands for *real-time.*

[3] The upper index i in $O^i_{\ell;m}$ stands for *imaginary-time.*

4. Numerical examples in 1-d

In the one-dimensional case and for $a > 0$, we introduce $\Omega_a = (-a, a)$, $\Omega^+_{a,\epsilon} = (-a, \epsilon/2)$ and $\Omega^-_{a,\epsilon} = (-\epsilon/2, a)$, 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 $\{x_j\}_{j\in\{1,\cdots,N^+_\epsilon\}}$ the grid nodes in $\Omega^+_{a,\epsilon}$ and $\{y_j\}_{j\in\{1,\cdots,N^-_\epsilon\}}$ those in $\Omega^-_{a,\epsilon}$. In the following tests, the domains overlap on o nodes such that: $x_{N^+_\epsilon} = y_{1+o}$ and $x_{N^+_\epsilon - o} = y_1$. The spatial mesh size $h = h_0$ is assumed to be constant and then $\epsilon = (o-1)h$, which is the length of the overlapping zone.

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

At a given level and in real-time, we consider the following Crank-Nicolson scheme [28]. Denoting $\phi^{\pm,n,(k)}$ the approximate solution in $\Omega^\pm$ at time t_n with $n \geq 0$ and at Schwarz iteration $k \geq 0$, we get

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

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

- For the SWR-DDM with Robin-type transmission conditions

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

  for a 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-order spatial derivative is approximated by using a 3 point-scheme. We again refer to [12] 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 $n \geq 0$,

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

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 strength $\kappa = 250$. The initial data is given by $\phi_0(x) = \exp(-x^2/2)\pi^{-1/4}$. The two subdomains are $\Omega^+_{a,\epsilon} = (-a, \epsilon/2)$ and $\Omega^-_{a,\epsilon} = (-\epsilon/2, a)$, 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 $\ell = 0$. We successively compare the rate of convergence or residual history (8), at level $m > 0$, i.e. on upper levels Ω_m, with $h_m \approx h_0/2^m$ and $m = 1, 2, 3$. We then have $N_m = 2^m N_0$, $N^+_m = o + 2^{m-1}N_0$ and $N^-_m = 2^{m-1}N_0$. The overlapping region covers two nodes ($o = 2$): $\epsilon_m = (o-1)h_m = h_m$, $m = 1, 2, 3$. The time steps are equal to $\Delta t_1 = 5 \times 10^{-2}$, $\Delta t_2 = 2 \times 10^{-2}$ and $\Delta t_3 = 5 \times 10^{-3}$. 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 $\ell = 0$ (or in Ω_0) $f_{g;0}$ is first computed. As proposed in Section 2.2, we impose $\phi^\pm_0 = I_h^{0;m} f^\pm_{p;0}$ at the upper level m, with $m = 1, 2$, i.e. at $\{\pm\epsilon_m/2\} \times \{t > 0\}$, we impose

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

where $f_{g;0}$ is the ground state computed on $\Omega^\pm_0$. In the Test-case 1, the size of the overlapping region is always reduced to $\epsilon = h_m$, $m = 1, 2$. Convergence results (residual history) are reported in Fig. 1 (top). 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 (down) the convergence times $T^{(k)}$ of the CNGF per Schwarz iteration, i.e. for $m = 1, 2$,

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

We observe on Fig. 1 (top) that the preconditioning from lower levels $m - 1$ or $m - 2$ 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.

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

$$\phi_m^{\pm,(0)} = I_h^{m-1;m} f^\pm_{g;m-1}.$$

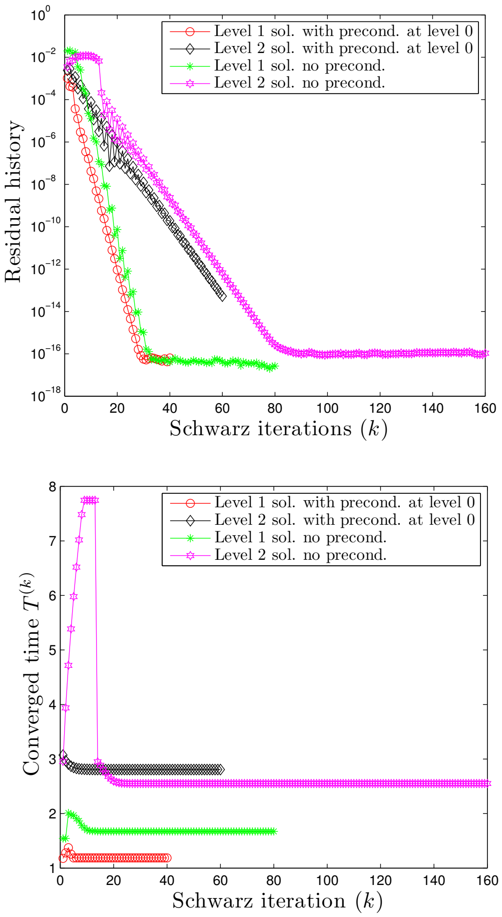


Figure 1: Levels 1 and 2. Top: Comparison of the residual history (8) vs. k, with/without preconditioning at the coarse level 0. Down: Minimization time (convergence time of the CNGF) $T^{(k)}$ per Schwarz iteration k, for $N_0 = 2^7$.

In other words, we compare $k^{\mathrm{cvg}}_{m;m-1}$ with k^{cvg}_{m}, for $m = 1, 2, 3$. We also study the convergence times $T^{(k)}$ of the minimization algorithm i) $K^{(k)}_{m;m-1}\Delta t_m$ (with preconditioning) and $K^{(k)}_{m}\Delta t_m$ (without preconditioning). In the previous expressions, $K^{(k)}_{m;m-1}$ (resp. $K^{(k)}_{m}$) is the number of iterations of the CNGF to converge at level m with (resp. without) preconditioning, for $m = 1, 2, 3$ (see Section 3.4). At the coarse level $\ell = m - 1 \geq 0$, $\phi^{\pm}_{m-1}(0)$ is chosen as the projection on $\Omega^{\pm}_{m-1}$ of $\phi_0(x) = \pi^{-1/4}e^{-x^2/2}$ for computing the ground state of the NLSE [5]. We consider the standard homogeneous Dirichlet boundary conditions when $k = 0$: $\phi^{\pm,(0)}_{m} = 0$ at $\{\pm\epsilon_m/2\}$. We report on Fig. 2 the residual history (8) with respect to the lower level preconditioning, for different values of grid points $N_m = 2^{7+m}$. We observe that $k^{\mathrm{cvg}}_{1;0} \leq k^{\mathrm{cvg}}_1$, $k^{\mathrm{cvg}}_{2;1} \leq k^{\mathrm{cvg}}_2$, $k^{\mathrm{cvg}}_{3;2} \leq k^{\mathrm{cvg}}_3$. However, the Schwarz acceleration is moderate and seems independent of the number of grid points.

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

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

$$\big(\partial_x + \gamma\big)\phi^{\pm,(0)}_m(t^n_m) = \big(\partial_x + \gamma\big) I_h^{m-1;m} f^{\pm}_{g;m-1}$$

at $\big\{ \pm \epsilon_m/2 \big\}$. 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.

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 $\Omega^{+}_{a,\epsilon} = (-a, 5/2 + \epsilon/2)$ and $\Omega^{-}_{a,\epsilon} = (5/2 - \epsilon/2, a)$, with $\epsilon > 0$ 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

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

Test-case 1. In this first test-case, the numerical data are as follows ($\ell = 0$, $m = 1$): $N_1 = 2N_0$, $N_1^+ = o + N_0$, $N_1^- = N_0$, with $N_0 = 400$. The overlapping region covers respectively 20, 10 and 2 nodes. The time step is fixed to $\Delta t_1 = \Delta t_0 = 1 \times 10^{-2}$.

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 $\epsilon = (o-1)h_1$. 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 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. We therefore expect a faster convergence compared to the CSWR method [22]. 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: $N_1 = 2N_0$, $N_1^+ = o + N_0$ and $N_1^- = N_0$, with overlap $o = 2$ and $\epsilon = (o-1)h_1$. The time step is fixed to $\Delta t_1 = \Delta t_0 = 1 \times 10^{-2}$. We impose at the subdomain interfaces, and for all $n \geq 0$

$$\big(\partial_x \pm \gamma\big)\phi_1^{\pm,(0)} = \big(\partial_x \pm \gamma\big) I_h^{0;1} \phi_0^{\pm,\mathrm{cvg}}, \qquad \text{at } \big\{ \pm \epsilon_1/2 \big\},$$

with $\gamma = 20$. We compare the residual history (8) for the CSWR and Robin-based SWR algorithms with and without preconditioning. The algorithm is tested respectively with $N_0 = 200$ and $N_0 = 400$. 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%.

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 in using approximate solutions computed on coarser grids (lower levels), and 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 be easily 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, R. Duboscq, GPELab, a Matlab toolbox to solve Gross-Pitaevskii equations I: Computation of stationary solutions, Comput. Phys. Comm. 185 (11) (2014) 2969–2991.

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

[3] X. Antoine, R. Duboscq, Modeling and computation of Bose-Einstein condensates: stationary states, nucleation, dynamics, stochasticity, no. 2146 in Nonlinear Optical and Atomic Systems: at the Interface of Mathematics and Physics, CEMPI Subseries, 1st Volume, Lecture Notes in Mathematics, Springer, 2015.

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

[5] W. Bao, Q. Du, Computing the ground state solution of Bose-Einstein condensates by a normalized gradient flow, SIAM J. Sci. Comput. 25 (5) (2004) 1674–1697. doi:10.1137/S1064827503422956.

[6] G. Fibich, The Nonlinear Schrödinger Equation. Singular Solutions and Optical Collapse, no. 192 in Applied Mathematical Sciences, Springer-Verlag, New York, 2015.

[7] C. Sulem, P. Sulem, The Nonlinear Schrödinger Equation. Self-Focusing and Wave Collapse, no. 139 in Applied Mathematical Sciences, Springer-Verlag, New York, 1999.

[8] X. Antoine, E. Lorin, A. 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. Sci. Comput. 64 (3) (2015) 620–646.

[9] E. Cancès, M. Defranceschi, W. Kutzelnigg, C. Le Bris, Y. Maday, Computational quantum chemistry: a primer, in: Handbook of numerical analysis, Vol. X, Handb. Numer. Anal., X, North-Holland, Amsterdam, 2003, pp. 3–270.

[10] A. Bandrauk, F. Fillion-Gourdeau, E. Lorin, Atoms and molecules in intense laser fields: Gauge invariance of theory and models, Journal of Physics B: Atomic, Molecular and Optical Physics 46 (15).

[11] A. Bandrauk, M. C. Delfour, C. Le Bris (Eds.), High-dimensional partial differential equations in science and engineering, Vol. 41 of CRM Proceedings & Lecture Notes, American Mathematical Society, Providence, RI, 2007, papers from the conference held at the Université de Montréal, Montréal, QC, August 7–12, 2005.

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

[13] X. Antoine, E. Lorin, Lagrange-Schwarz waveform relaxation domain decomposition methods for linear and nonlinear quantum wave problems, Appl. Math. Lett. 57 (2016) 38–45.

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

[15] M. Gander, L. Halpern, Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems, SIAM J. Num. Anal. 45 (2) (2007) 666–697.

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

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

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

[19] M. Gander, L. Halpern, F. Nataf, Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation, in: Eleventh International Conference of Domain Decomposition Methods, 1999, pp. 27–36.

[20] E. Lorin, X. Yang, X. Antoine, Frozen Gaussian approximation based domain decomposition methods for the linear Schrödinger equation beyond the semi-classical regime, J. Comput. Phys. 315 (2016) 221–237.

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

[22] 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.

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

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

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

[26] V. Dolean, M. J. Gander, L. Gerardo-Giorda, Optimized Schwarz methods for Maxwell's equations, SIAM J. Sci. Comput. 31 (3) (2009) 2193–2213. doi:10.1137/080728536.

[27] M. El Bouajaji, V. Dolean, M. J. Gander, S. Lanteri, Optimized Schwarz methods for the time-harmonic Maxwell equations with damping, SIAM J. Sci. Comput. 34 (4) (2012) A2048–A2071. doi:10.1137/110842995.

[28] X. Antoine, C. Besse, S. Descombes, Artificial boundary conditions for one-dimensional cubic nonlinear Schrödinger equations, SIAM J. Numer. Anal. 43 (6) (2006) 2272–2293 (electronic). doi:10.1137/040606983.

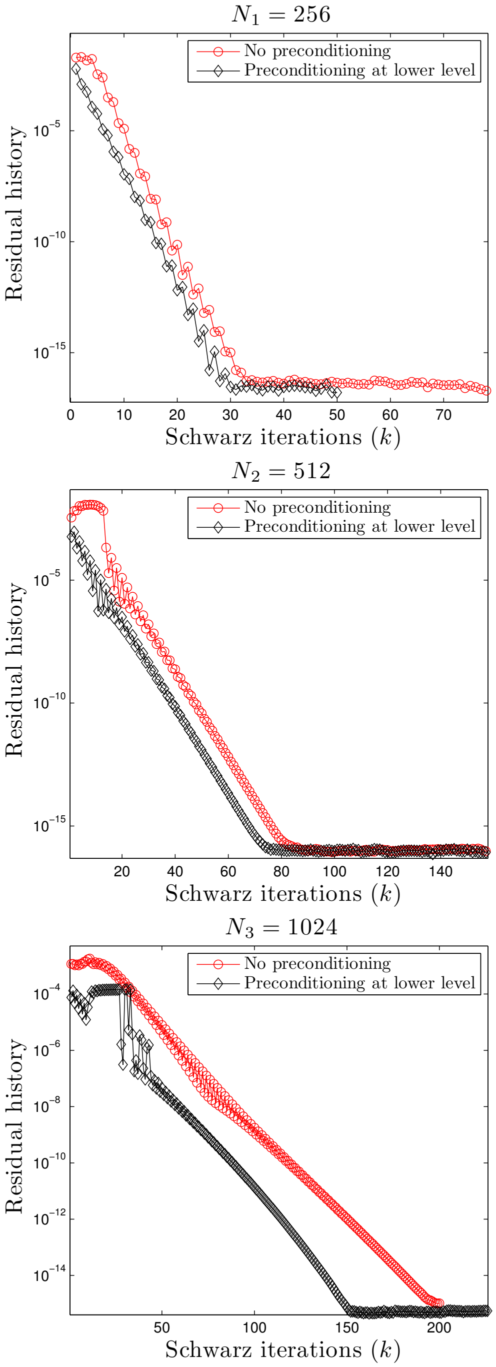


Figure 2: Comparison of the residual history (8) vs. k, of the CSWR method at level m, for a solution with preconditioning at level $\ell = m - 1$, and without preconditioning. Top: $m = 1$. Middle: $m = 2$. Down: $m = 3$.

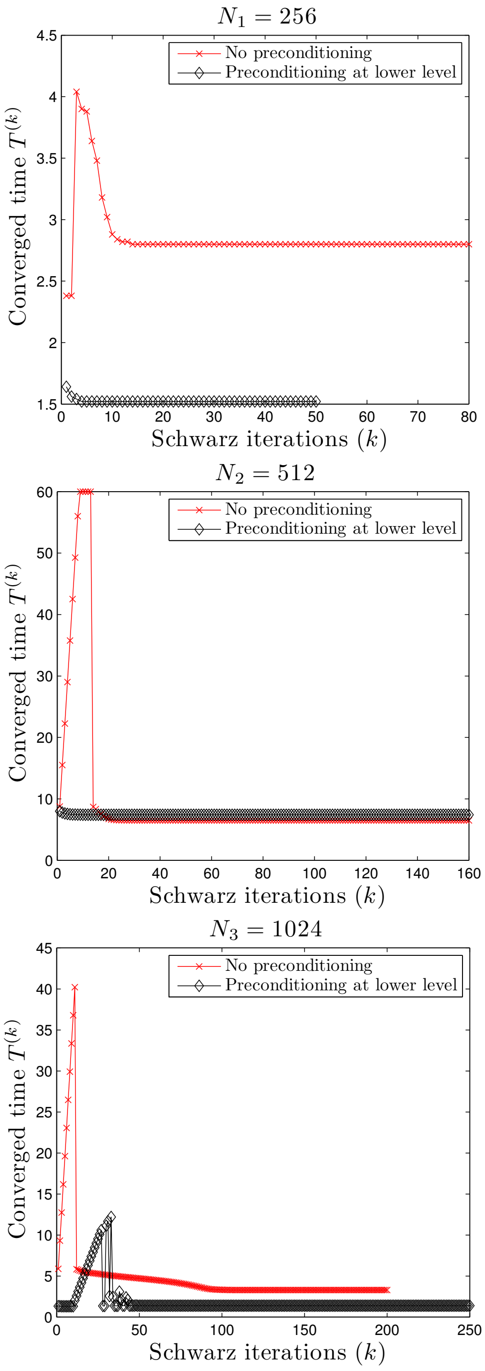
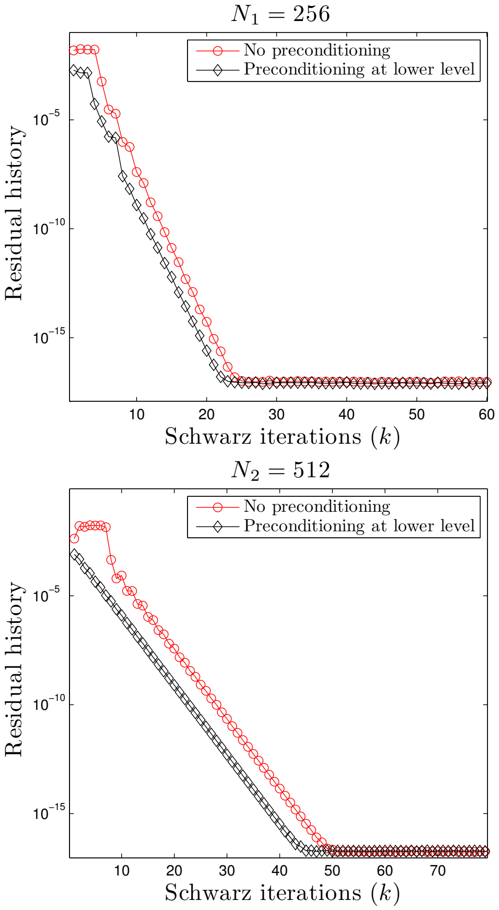


Figure 3: Minimization time (CNGF convergence time) $T^{(k)}$: comparison for the CSWR method at level m between the solution with preconditioning at level $m-1$, and without preconditioner. Top: $m = 1$. Middle: $m = 2$. Down: $m = 3$.


$N_1 = 256$

$N_2 = 512$

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

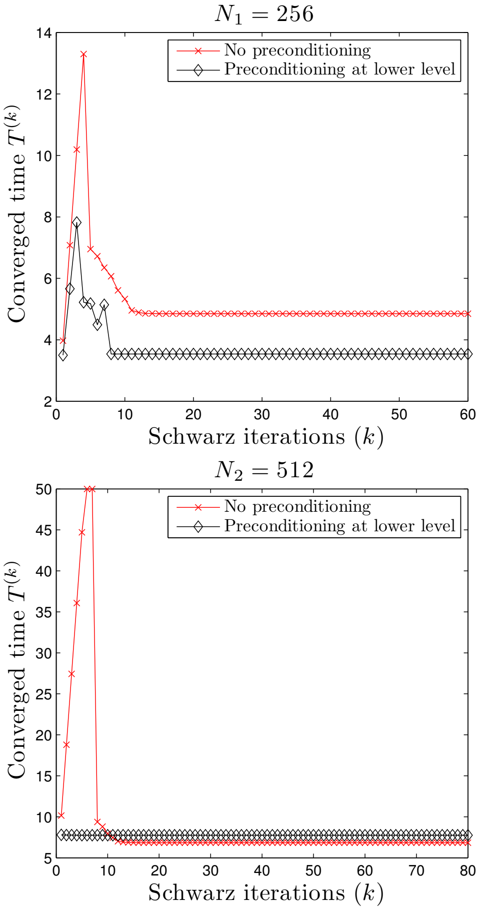


Figure 5: Minimization time (CNGF convergence time) $T^{(k)}$ comparison for Robin-based SWR at level m between solution with preconditioning at level $m-1$, and without preconditioning. Top: $m = 1$. Down: $m = 2$.

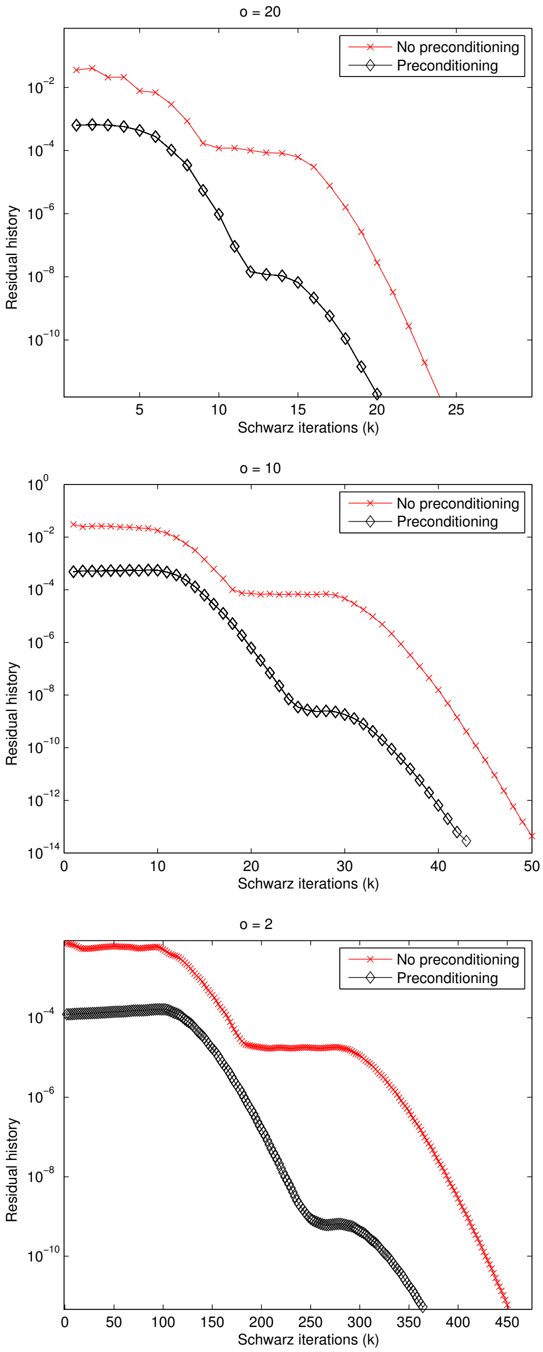


Figure 6: Comparison of the residual history (4) vs k, for the CSWR method, with and without preconditioning for $N_0 = 400$. Top: $o = 20$. Middle: $o = 10$. Down: $o = 2$.

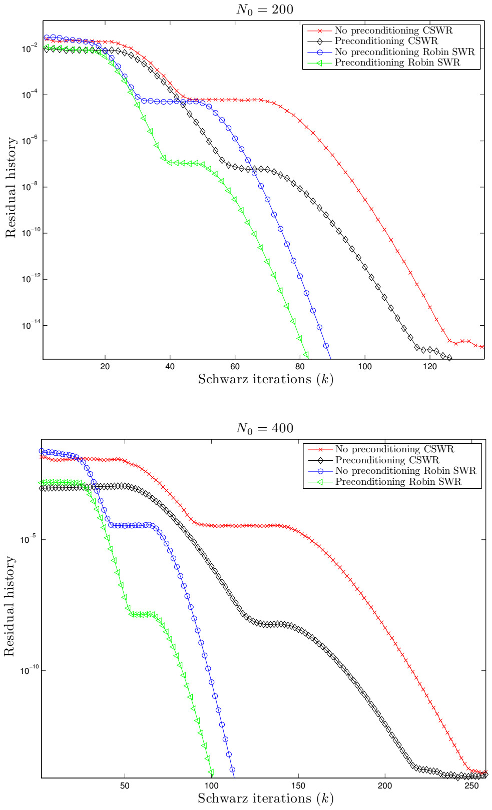


Figure 7: Comparison of the residual history (4) vs. k, of the CSWR and Robin-based SWR methods, with and without preconditioning, for $o = 2$. Top: $N_0 = 200$. Down: $N_0 = 400$.