Noname manuscript No.
(will be inserted by the editor)

Derivation and analysis of computational methods for fractional Laplacian equations with absorbing layers

X. Antoine · E. Lorin · Y. Zhang

the date of receipt and acceptance should be inserted later

Abstract This paper is devoted to the derivation and analysis of accurate and efficient Perfectly Matched Layers (PML) or efficient absorbing layers for solving fractional Laplacian equations within Initial Boundary Value Problems (IBVP). Two main approaches are derived: we first propose a Fourier-based pseudospectral method, and then present a real space method based on an efficient computation of the fractional Laplacian with PML. Some numerical experiments and analytical results are proposed along the paper to illustrate the presented methods.

Keywords fractional partial differential equations; perfectly matched layers; Fourier pseudospectral approximation; time splitting scheme; finite-difference

1 Introduction

During the last decades, there was a growing interest in the computational physics community to get access to efficient and simple fractional equation solvers due to recent development of differential models involving the fractional Laplacian, like e.g. in quantum physics, fluid dynamics, solid mechanics, epidemiology. We refer to [17,18,23,27,29,35,37,44] for some applications. This paper deals with the derivation and analysis of accurate and efficient numerical methods for fractional Laplacian equations with Perfectly Matched Layers (PML) within Initial Boundary Value Problems (IBVP). The main focus on this paper is hence the discretization of fractional equations *on bounded domains or truncated infinite domains, avoiding spurious wave reflections thanks to PML or accurate absorbing layers.* Two approaches are proposed. First, we develop a pseudospectral scheme, extending some of the ideas presented in [2,4,5,8] to fractional Laplacian equations with PML (see hereafter some technical details and references on PML). We next introduce a relatively simple real space scheme based on an efficient evaluation of the fractional Laplacian with PML. The main two difficulties addressed below for the approximation of the fractional Laplacian equations are i) the fact that fractional Laplacians are non-local operators which make their approximation potentially computationally expensive, ii) the fact that, on bounded domains, the combination with absorbing layers/absorbing boundary conditions require some special cares. We are simultaneously interested in the derivation of methods which are efficient (in this regard, absorbing boundary conditions may not be appropriate [9]) and accurate, justifying the use of PML (see e.g. [12,13,14,15,19,20,32,33,

X. Antoine
Institut Elie Cartan de Lorraine, Université de Lorraine, UMR 7502, Inria Nancy-Grand Est, F-54506 Vandoeuvre-lès-Nancy Cedex, France.
E-mail: xavier.antoine@univ-lorraine.fr

E. Lorin
Centre de Recherches Mathématiques, Université de Montréal, Montréal, Canada, H3T 1J4.
School of Mathematics and Statistics, Carleton University, Ottawa, Canada, K1S 5B6.
E-mail: elorin@math.carleton.ca

Y. Zhang
Center for Applied Mathematics, Tianjin University, Tianjin 300072, P.R. China.
E-mail: sunny5zhang@163.com

42,43]) and pseudospectral methods (see [2,3,4,5,8,10,34,38,40,41]). On unbounded domains, the latter allows for writing fractional Laplacians as power functions in Fourier space. Moreover, following simple ideas developed in [8] for solving PDEs with non-constant coefficients by using FFTs, it is possible to combine pseudospectral methods with PML. Let us remark that finite-difference/volume/element methods can also be implemented. The latter may however require to compute the real power of a matrix approximating the Laplace operator, at least when Dirichlet boundary conditions are imposed. The real power of a matrix is often numerically computed thanks to the approximation of a Cauchy integral having a contour enclosing the matrix spectrum [6,25,30,36]. Alternatively, an efficient differential method [7, 25] is proposed. Moreover the combination with PML is easily achieved thanks to rational approximants such as Padé's approximants [31].

We consider two-dimensional time-dependent Fractional Laplace Equations (FLE) [26] of the following form

$$\begin{cases} \mathtt{i}\partial_t u(t,\boldsymbol{x}) + \displaystyle\sum_{\alpha\in\mathfrak{K}} v_\alpha(t,\boldsymbol{x})(-\triangle)^\alpha u(t,\boldsymbol{x}) = 0, \quad (t,\boldsymbol{x}) \in [0,T]\times\mathbb{R}^2, \\ u(t=0,\boldsymbol{x}) = u_0(\boldsymbol{x}), \quad \boldsymbol{x}\in\mathbb{R}^2. \end{cases} \tag{1}$$

In the above system, $\{v_\alpha\}_{\alpha\in\mathfrak{K}}$ are supposed to be regular real- or purely complex-valued functions and $\mathfrak{K}$ is a finite set of strictly positive real numbers. System (1) includes e.g. fractional diffusion and Schrödinger-like equations. The proposed methods can be easily be extended in 3-d. Among the many definitions of fractional derivatives and fractional Laplacians [35], we select the Fourier spectral definition (also referred to as the Riesz derivative) of the fractional Laplacian, which reads

$$(-\triangle)^\alpha u = \mathcal{F}^{-1}(|\boldsymbol{\xi}|^{2\alpha}\mathcal{F}(u)(\boldsymbol{\xi})),$$

where ξ_x (respectively ξ_y) is the Fourier dual variable in direction x (respectively y), $\boldsymbol{\xi} = (\xi_x,\xi_y)$, $|\boldsymbol{\xi}|^2 := |\xi_x|^2 + |\xi_y|^2$, $\mathcal{F}(u)$ denotes the two-dimensional Fourier transform of u and $\mathcal{F}^{-1}$ is the associated inverse Fourier transform. For any $u\in\mathcal{S}(\mathbb{R}^2)$ (i.e. the Schwartz's space of rapidly decaying C^∞-functions [39]) and $\alpha\in(0,1)$, we have $(-\triangle)^\alpha u \in L^2(\mathbb{R}^2)$. An equivalent definition [26] can be stated, for $\alpha\in(0,1)$ and any $u\in\mathcal{S}(\mathbb{R}^2)$, as

$$(-\triangle)^\alpha u(\boldsymbol{x}) = C(\alpha)\text{p.v.}\int_{\mathbb{R}^2}\frac{u(\boldsymbol{x})-u(\boldsymbol{y})}{|\boldsymbol{x}-\boldsymbol{y}|^{2+2\alpha}}d\boldsymbol{y} = C(\alpha)\lim_{\varepsilon\to 0^+}\int_{\mathbb{R}^2\setminus B_\varepsilon(\boldsymbol{x})}\frac{u(\boldsymbol{x})-u(\boldsymbol{y})}{|\boldsymbol{x}-\boldsymbol{y}|^{2+2\alpha}}d\boldsymbol{y},$$

where $B_\varepsilon(\boldsymbol{x})$ is the ball of radius ε and center $\boldsymbol{x}$, $C(\alpha)$ is the constant defined by

$$C(\alpha) := \Big(\int_{\mathbb{R}^2}\frac{1-\cos(\xi_x)}{|\boldsymbol{\xi}|^{2+2\alpha}}d\boldsymbol{\xi}\Big)^{-1}, \tag{2}$$

and p.v. denotes the principal value. In fact, it is proven in [26] that the fractional Laplacian can also be rewritten, for $\alpha\in(0,1)$ and any $u\in\mathcal{S}$, as

$$\begin{aligned}(-\triangle)^\alpha u(\boldsymbol{x}) &= -\frac{1}{2}C(\alpha)\text{p.v.}\int_{\mathbb{R}^2}\frac{u(\boldsymbol{x}+\boldsymbol{y})-2u(\boldsymbol{x})+u(\boldsymbol{x}-\boldsymbol{y})}{|\boldsymbol{y}|^{2+2\alpha}}d\boldsymbol{y} \\ &= C(\alpha)\lim_{\varepsilon\to 0^+}\int_{\mathbb{R}^2\setminus B_\varepsilon(\boldsymbol{x})}\frac{u(\boldsymbol{x})-u(\boldsymbol{y})}{|\boldsymbol{x}-\boldsymbol{y}|^{2+2\alpha}}d\boldsymbol{y}\,.\end{aligned}$$

Although nonlocal, it could be interesting to numerical investigate this equality. In the following, the equation under consideration is approximated on an open two-dimensional bounded rectangular physical domain denoted by $\mathcal{D}_{\text{Phy}}$. Hereafter, we present the standard PML-framework for solving PDE on a bounded domain [12,13]. In order to absorb the waves at the boundary of the computational domain, we add a layer $\mathcal{D}_{\text{PML}}$ surrounding $\mathcal{D}_{\text{Phys}}$. The overall computational domain is denoted: $\mathcal{D} = \overline{\mathcal{D}_{\text{Phy}}\cup\mathcal{D}_{\text{PML}}}$. In two dimensions, one gets $\mathcal{D} = [-L_x,L_x]\times[-L_y,L_y]$ and $\mathcal{D}_{\text{Phys}} = [-L_x^*,L_x^*]\times[L_y^*,L_y^*]$, where $L_{x,y} > L_{x,y}^*$. Standard Perfectly Matched Layers (PMLs) methods [9] require a stretching of the real spatial coordinates using the following change of variables

$$\widetilde{\nu} = \nu + e^{\mathtt{i}\theta}\int_{L_\nu^*}^{\nu}\widetilde{\sigma}_\nu(s)ds, \tag{3}$$

with $\nu = x, y$, and where the absorbing functions are defined by

$$\widetilde{\sigma}_\nu(\nu) = \begin{cases} \sigma_\nu(|\nu| - L_\nu),\ L_\nu^* \leqslant |\nu| < L_\nu, \\ 0, \qquad\qquad\quad |\nu| < L_\nu^*. \end{cases}$$

The function S_ν is such that: $S_\nu(\nu) := 1 + e^{\mathtt{i}\theta_\nu}\widetilde{\sigma}(\nu)$ for some angle $\theta_\nu \in (0, \pi/2)$. The function S_ν then allows for an absorption of the waves by the way of the PML. We then define the (possibly fractional) differential operator along the ν-direction as

$$\partial_\nu^\gamma \mapsto \frac{1}{S_\nu^\gamma(\nu)}\partial_\nu^\gamma = \frac{1}{\big(1 + e^{\mathtt{i}\theta}\widetilde{\sigma}(\nu)\big)^\gamma}\partial_\nu^\gamma, \tag{4}$$

where γ is a given derivation order. In the following, we use the following standard polynomial absorption profiles

$$\text{Type I: } \sigma_0(\nu + \delta_\nu), \text{ Type II: } \sigma_0(\nu + \delta_\nu)^2. \tag{5}$$

Alternative techniques can be found such as the ones developed in [16,21,22]. The goal of the paper is to focus on how to implement PML and absorbing layers into the numerical schemes for FLE *but not to have a full numerical study of the optimization of the tuning parameters. In particular, we propose efficient and accurate methods for solving fractional Laplace-based equations on bounded domains.* In particular, in all the examples the parameters are fixed in order to show how the methods perform, but the optimization study is not addressed. Let us now define the Laplacian operator with PML through its total symbol [4] in the Fourier space

$$\begin{aligned} \sigma(-\triangle_{\mathrm{PML}}) &= -\sigma\Big(\frac{1}{S_x}\partial_x\Big(\frac{1}{S_x}\partial_x\Big) + \frac{1}{S_y}\partial_y\Big(\frac{1}{S_y}\partial_y\Big)\Big) \\ &= -\sigma\Big(\frac{1}{S_x^2}\partial_x^2 + \frac{1}{S_x}\partial_x\Big(\frac{1}{S_x}\Big)\partial_x + \frac{1}{S_y^2}\partial_y^2 + \frac{1}{S_y}\partial_y\Big(\frac{1}{S_y}\Big)\partial_y\Big) \\ &= \frac{1}{S_x^2}|\xi_x|^2 + \mathtt{i}\frac{1}{S_x}\partial_x\Big(\frac{1}{S_x}\Big)\xi_x + \frac{1}{S_y^2}|\xi_y|^2 + \mathtt{i}\frac{1}{S_y}\partial_y\Big(\frac{1}{S_y}\Big)\xi_y\,. \end{aligned} \tag{6}$$

By analogy, we also introduce the 2-d PML (also called Fractional PML in [8]) Laplacian operator $(-\triangle)^\alpha_{\mathrm{PML}}$ with variable coefficients based on the pseudodifferential operator definition [39]

$$(-\triangle)^\alpha_{\mathrm{PML}} u := \mathcal{F}^{-1}(a^\alpha(\boldsymbol{x}, \boldsymbol{\xi})\mathcal{F}(u)(\boldsymbol{\xi})) := \mathrm{Op}(a^\alpha)u,$$

with the symbol

$$a^\alpha(\boldsymbol{x}, \boldsymbol{\xi}) = (-\sigma(\triangle_{\mathrm{PML}}))^\alpha,$$

and where $P = \mathrm{Op}(\sigma)$ designates the pseudodifferential operator with symbol $\sigma := \sigma(P)$. However, from a practical point of view, in particular when α is not an integer, it is no longer possible to *efficiently* split the real space and Fourier variables in order to reconstruct a "simple" absorbing operator. In the latter case, a direct implementation would indeed require several non-trivial approximations including convolution products.

In this paper, we will propose two strategies to solve IBVP for fractional Laplacian equations using PML-techniques, which allow for i) efficient and accurate computations (spectral convergence when using FFTs), ii) while avoiding artificial wave reflections at the domain boundary and possible side-effects due to periodic boundary conditions.

This paper is organized as follows. A Fourier-based method for solving fractional Laplacian equations on bounded domains using PMLs is derived in Section 2. While the equation is computed using FFTs, PMLs are evaluated thanks to Padé's approximants in Subsection 2.2. Section 3 is devoted to the derivation and analysis of a finite-difference approach for solving fractional Laplacian and Schrödinger-type equations. The Laplace operator is here approximated by a standard finite-difference scheme, and fractional Laplace operators are computed explicitly through the real power of discrete Laplace operator/matrix based on the solution to a differential system (Subsection 3.1). This leads to some well-adapted schemes for solving FLE in Subsection 3.2. The computational complexity of the algorithm is analyzed in Subsection 3.3 while the implementation of the PML is detailed in Subsection 3.4. We conclude the paper in Section 4. In addition, we provide some details about our Padé's approximants approach for the PML in Appendix A. An alternative approach based on the approximation of a Cauchy integral is also discussed in Appendix B.

2 Pseudospectral method for fractional Laplace equations with PMLs

In this section, we derive a study of a very efficient Fourier-based method for solving IBVP with fractional Laplacian. The counterpart of Fourier methods, as we are interested in IBVP (thus bounded domains), is i) periodic effects and ii) artificial wave reflections at the domain boundary. In order to deal with *both* issues, we propose to introduce PML-techniques which moreover do not complexify much the overall computation of the IBVP. Regarding boundary issues, we refer e.g. to [16,21,24].

2.1 Fractional Laplacian equations with time-dependent coefficients

Let us first assume that the coefficients are only time-dependent, which allows us to perform a simple transformation on the equation and to introduce classical PMLs for the Laplacian. We consider

$$\begin{cases} \mathtt{i}\partial_t u + c(t)(-\triangle)^{\alpha} u = 0,\ (t,\boldsymbol{x}) \in [0,T]\times\mathbb{R}^2, \\ u(0,\boldsymbol{x}) = u_0(\boldsymbol{x}),\ \boldsymbol{x}\in\mathbb{R}^2, \end{cases} \tag{7}$$

with $\alpha \in (0,1)$, $(t,\boldsymbol{x})$ in $[0,T]\times\mathbb{R}^2$ and $c : t \mapsto c(t)$ being a smooth real time-dependent function. We formally compose the equation with $(-\triangle)^{1-\alpha}$, leading to

$$\begin{cases} \mathtt{i}\partial_t (-\triangle)^{1-\alpha} u - c(t)\triangle u = 0,\ (t,\boldsymbol{x}) \in [0,T]\times\mathbb{R}^2, \\ u(0,\boldsymbol{x}) = u_0(\boldsymbol{x}),\ \boldsymbol{x}\in\mathbb{R}^2. \end{cases} \tag{8}$$

A PML-Laplacian formulation is introduced in $[0,T]\times[-L_x,L_x]\times[-L_y,L_y]$ with periodic boundary conditions, that is

$$\begin{cases} \mathtt{i}\partial_t (-\triangle)^{1-\alpha} u - c(t)\triangle_{\mathrm{PML}} u = 0,\ (t,\boldsymbol{x}) \in [0,T]\times\mathcal{D}, \\ u(0,\boldsymbol{x}) = u_0(\boldsymbol{x}),\ \boldsymbol{x}\in\mathcal{D}, \\ u(t,L_x,y) = u(t,-L_x,y),\ t\in[0,T], \\ u(t,x,L_y) = u(t,x,-L_y),\ t\in[0,T]\,. \end{cases} \tag{9}$$

Let us denote the set of grid-points [8] by

$$\begin{aligned} \mathcal{D}_{N_x,N_y} &= \big\{(x_{k_1},y_{k_2})\big\}_{(k_1,k_2)\in\mathcal{O}_{N_x,N_y}}, \\ \mathcal{O}_{N_x,N_y} &= \big\{(k_1,k_2)\in\mathbb{N}^2 / \, k_1 = 0,\cdots,N_x-1,\, k_2 = 0,\cdots,N_y-1\big\}, \end{aligned}$$

and the uniform mesh size by $h_x := x_{k_1+1} - x_{k_1} = 2L_x/N_x$ and $h_y := y_{k_2+1} - y_{k_2} = 2L_y/N_y$ (for the entire domain $\mathcal{D}$). The corresponding discrete wave numbers are defined by $\xi_{\nu;p} = p\pi/L_\nu$, for $p \in \{-N_\nu/2,\cdots,N_\nu/2-1\}$ with $\nu = x,y$. Regarding the pseudospectral approximations [8,11,28] , we use the following notation

$$\widehat{u}^{(x)}_{p,k_2}(t) = \sum_{k_1=0}^{N_x-1} u_{k_1,k_2}(t) e^{-\mathtt{i}\xi_{x;p}(x_{k_1}+L_x)}, \qquad \widetilde{u}^{(x)}_{k_1,k_2}(t) = \frac{1}{N_x}\sum_{p=-N_x/2}^{N_x/2-1} \widehat{u}^{(x)}_{p,k_2}(t) e^{\mathtt{i}\xi_{x;p}(x_{k_1}+L_x)}\,, \tag{10}$$

and

$$\begin{aligned} \widehat{u}^{(x,y)}_{p,q}(t) &= \sum_{k_1=0}^{N_x-1}\sum_{k_2=0}^{N_y-1} u_{k_1,k_2}(t) e^{-\mathtt{i}(\xi_{x;p}(x_{k_1}+L_x)+\xi_{y;q}(y_{k_2}+L_y))}, \\ \widetilde{u}^{(x,y)}_{k_1,k_2}(t) &= \frac{1}{N_xN_y}\sum_{p=-N_x/2}^{N_x/2-1}\sum_{q=-N_y/2}^{N_y/2-1} \widehat{u}^{(x,y)}_{p,q}(t) e^{\mathtt{i}(\xi_{x;p}(x_{k_1}+L_x)+\xi_{y;q}(y_{k_2}+L_y))}. \end{aligned} \tag{11}$$

More precisely [11], $\widehat{u}^{(x,y)}_{p,q}(t)$ is an approximate discrete Fourier transform and $\widetilde{u}^{(x,y)}_{k_1,k_2}(t)$ is an approximation to $u(t,x_{k_1},y_{k_2})$ obtained through a discrete inverse Fourier transform with for some $c>0$

$$\max_{(k_1,k_2)\in\mathcal{O}_{N_x,N_y}} |\widetilde{u}_{k_1,k_2}(t) - \mathcal{F}^{-1}(u)(t,x_{k_1},y_{k_2})| \leqslant c(N_xN_y)^{1-s}\|u\|_{H^s},$$

for $s > 1$ and $u(t,\cdot) \in L^1 \cap H^s$ periodic on the torus $\mathbb{T}^2$, where L^1 (resp. H^s) is the standard Lebesgue (resp. Sobolev) space. Typically, the high modes which are neglected in the above approximation leads to the following aliasing error estimates: for $u(t,\cdot) \in H^r$, there exists $c > 0$ such that

$$\|\widetilde{\widehat{u}}(t,\cdot) - u(t,\cdot)\|_{H^s} \leqslant c(N_x N_y)^{s-r} \|u(t,\cdot)\|_{H^r} ,$$

for some $r > s > 1$ (in 2-d) and $u(t,\cdot) \in L^1 \cap H^r$-periodic. We next introduce the following discrete x-directional fractional operator $[[(-\triangle_x)^\alpha]]u_{k_1,k_2}(t_n)$ approximating $(-\triangle_x)^\alpha u(t_n, x_{k_1}, y_{k_2})$ such that

$$[[(-\triangle_x)^\alpha]]u_{k_1,k_2}(t_n) := \frac{1}{N_x} \sum_{p=-N_x/2}^{N_x/2-1} \left|\xi_{x;p}\right|^{2\alpha} e^{\mathtt{i}\alpha\pi} \widehat{\widetilde{u}}^{(x)}_{p,k_2}(t_n) e^{\mathtt{i}\xi_{x;p}(x_{k_1}+L_x)} \tag{12}$$

and the approximation $\partial_x^\alpha u(t_n, x_{k_1}, y_{k_2})$ to $[[\partial_x^\alpha]]u_{k_1,k_2}(t_n)$ given by

$$[[\partial_x^\alpha]]u_{k_1,k_2}(t_n) := \frac{1}{N_x} \sum_{p=-N_x/2}^{N_x/2-1} (\mathtt{i}\xi_{x;p})^\alpha e^{\mathtt{i}\alpha\pi} \widehat{\widetilde{u}}^{(x)}_{p,k_2}(t_n) e^{\mathtt{i}\xi_{x;p}(x_{k_1}+L_x)}. \tag{13}$$

Similar operators can be defined along the y-direction. Then, an approximation of the fractional Laplacian $\triangle^\alpha u(t_n, x_{k_1}, y_{k_2})$ on the grid is

$$\begin{aligned}[[(-\triangle)^\alpha]]u_{k_1,k_2}(t_n) := \frac{1}{N_x N_y} \sum_{p=-N_x/2}^{N_x/2-1} \sum_{q=-N_y/2}^{N_y/2-1} \left(\left|\xi_{x;p}\right|^2 + \left|\xi_{y;q}\right|^2\right)^\alpha \\ \times e^{\mathtt{i}\alpha\pi} \widehat{\widetilde{u}}^{(x,y)}_{p,q}(t_n) e^{\mathtt{i}(\xi_{x;p}(x_{k_1}+L_x)+\xi_{y;q}(y_{k_2}+L_y))} .\end{aligned} \tag{14}$$

We denote by $\boldsymbol{u}_h^n = \{u^n_{k_1,k_2}\}_{(k_1,k_2)\in\mathcal{O}_{N_x,N_y}}$ the approximation of u on $\mathcal{D}_{N_x,N_y}$ at time t_n and by $\boldsymbol{S}_{\nu;h}$ (resp. $\boldsymbol{S}'_{h,\nu}$) the projection of S_ν (resp. $S'_{h,\nu}$) on the grid set $\mathcal{D}_{N_x,N_y}$. Starting from $\boldsymbol{u}_h^0$, a natural implicit scheme in time $t_n < t_{n+1} = t_n + \Delta t$ (with $\Delta t > 0$) then reads

$$[[(-\triangle)^{1-\alpha}]]\boldsymbol{u}_h^{n+1} + \mathtt{i}c(t_{n+1})\Delta t[[\triangle_{\mathrm{PML}}]]\boldsymbol{u}_h^{n+1} = [[(-\triangle)^{1-\alpha}]]\boldsymbol{u}_h^n , \tag{15}$$

where $\triangle_{\mathrm{PML}}$ is approximated by

$$[[\triangle_{\mathrm{PML}}]] := \frac{\mathbf{I}_h}{\boldsymbol{S}^2_{x,h}}[[\triangle_x]] - \frac{\boldsymbol{S}'_{x,h}}{\boldsymbol{S}^3_{x,h}}[[\partial_x]] + \frac{\mathbf{I}_h}{\boldsymbol{S}^2_{y,h}}[[\triangle_y]] - \frac{\boldsymbol{S}'_{y,h}}{\boldsymbol{S}^3_{y,h}}[[\partial_y]]. \tag{16}$$

At each time iteration, we then solve (15)-(16) by using a standard iterative linear system solver:

$$A^{n+1}\boldsymbol{u}_h^{n+1} = B\boldsymbol{u}_h^n .$$

This methodology was also used in [2,5] for the Dirac and the Schrödinger equation.

To illustrate the approach, we consider the following system (with $\alpha = 9/10$)

$$\begin{cases} \mathtt{i}\partial_t u(t,\boldsymbol{x}) + (-\triangle)^{9/10} u(t,\boldsymbol{x}) = 0, & (t,\boldsymbol{x}) \in [0,T]\times\mathbb{R}^2, \\ u(t=0,\boldsymbol{x}) = u_0(\boldsymbol{x}), & \boldsymbol{x}\in\mathbb{R}^2, \end{cases} \tag{17}$$

with the initial data

$$u_0(x,y) = 10e^{-10((x+40/13)^2+(y-1)^2)-5\mathtt{i}x+2\mathtt{i}y}. \tag{18}$$

We rewrite the equation in the form (8), which is approximated by (15) and (16), i.e.

$$[[(-\triangle)^{9/10}]]\boldsymbol{u}_h^{n+1} + \mathtt{i}\Delta t[[\triangle_{\mathrm{PML}}]]\boldsymbol{u}_h^{n+1} = [[(-\triangle)^{9/10}]]\boldsymbol{u}_h^n. \tag{19}$$

The computational domain is $\mathcal{D} = [-4,4]^2$ and the final time is $T = 40$. In this example, we take $S_\nu(\nu) = 1 + 10^{-2}e^{\mathtt{i}\theta}\widetilde{\sigma}(\nu)$, where $\sigma(\nu) = \sigma_0(\nu+\delta)^3$ [4] and $\sigma_0 = 5$, $\theta = \pi/4$ and $\delta_\nu = 0.025L_\nu$. We implement a semi-implicit discretization scheme to solve the corresponding IBVP (9), i.e. we first solve the equation in the x- and then the y-direction using an ADI-type algorithm. The real space grid contains $N_x \times N_y = 75^2$ points and we fix the time step to $\Delta t = 10^{-1}$. We report in Fig. 1 the amplitude of the initial data, the solution with periodic boundary conditions (without PML), the reference numerical

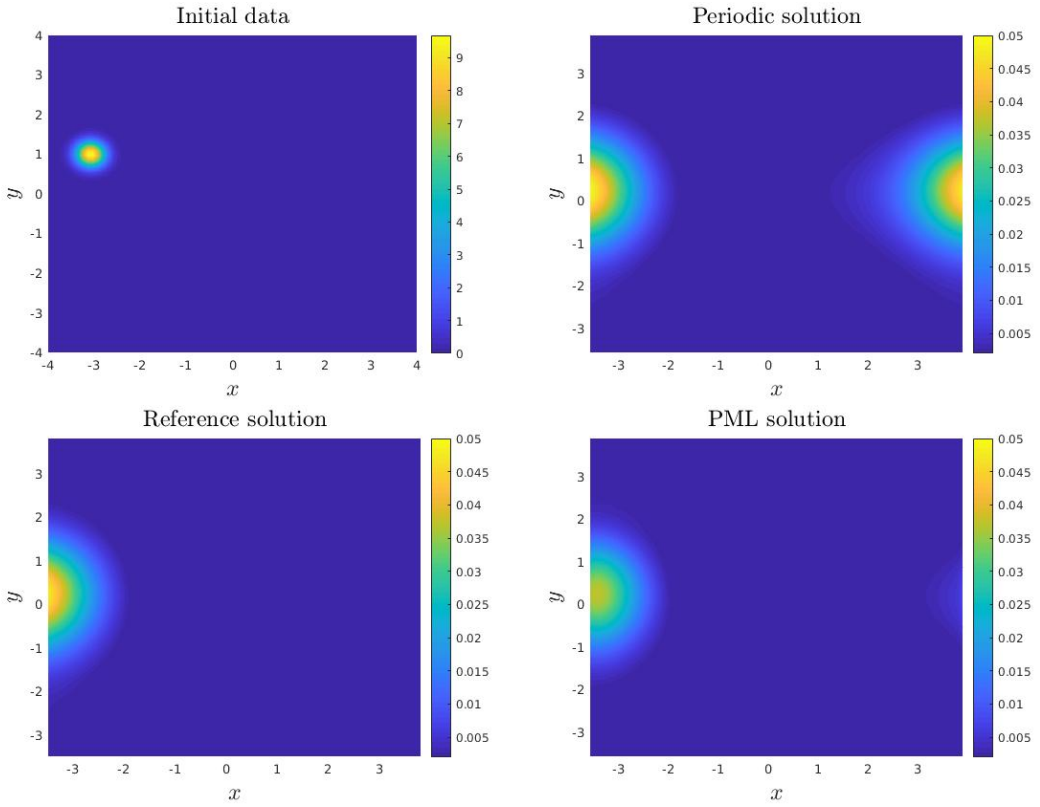

Fig. 1 (Top-left) amplitude of the initial data, (Top-right) periodic solution, (Bottom-left) reference solution and (Bottom-right) of the PML solution.

solution (computed on a larger domain) and the PML-based solution. We observe that the wave is mainly absorbed at $y = -4$.

We can directly generalize this approach to equations of the form

$$\begin{cases} \mathtt{i}\partial_t u(t,\boldsymbol{x}) + \displaystyle\sum_{\alpha\in\mathfrak{K}} v_\alpha(t)(-\triangle)^\alpha u(t,\boldsymbol{x}) = 0, \qquad (t,\boldsymbol{x}) \in [0,T]\times\mathbb{R}^2, \\ u(0,\boldsymbol{x}) = u_0(\boldsymbol{x}), \quad \boldsymbol{x}\in\mathbb{R}^2. \end{cases} \tag{20}$$

Indeed, for $m \in \mathbb{N}^*$ fixed, we consider the following problem on $\mathcal{D}$ with periodic boundary conditions at $\partial\mathcal{D}$. We propose to use a simple operator-splitting by successively solving m single fractional equations similar to (7), for any α_i in $\mathfrak{K} = \{\alpha_1, \cdots, \alpha_m\}$. More precisely, we have the following proposition.

Proposition 1 *Let us assume that* (20) *is well-posed. Then, it is equivalent to solve successively the following set of equations*

$$\begin{cases} \mathtt{i}\partial_t(-\triangle)^{1-\alpha_1}u^{(1)}(t,\boldsymbol{x}) - v_{\alpha_1}(t)\triangle u^{(1)}(t,\boldsymbol{x}) = 0, \qquad u^{(1)}(0,\cdot) = u_0, \qquad t\in[0,T], \\ \qquad\qquad \cdots \\ \mathtt{i}\partial_t(-\triangle)^{1-\alpha_k}u^{(k)}(t,\boldsymbol{x}) - v_{\alpha_k}(t)\triangle u^{(k)}(t,\boldsymbol{x}) = 0, \;\; u^{(k)}(0,\cdot) = u^{(k-1)}(T,\cdot), \;\; t\in[0,T], \\ \qquad\qquad \cdots \\ \mathtt{i}\partial_t(-\triangle)^{\alpha_m}u^{(m)}(t,\boldsymbol{x}) - v_{\alpha_m}(t)\triangle u^{(m)}(t,\boldsymbol{x}) = 0, \; u^{(m)}(0,\cdot) = u^{(m-1)}(T,\cdot), \; t\in[0,T]. \end{cases} \tag{21}$$

Proof. Since we have $[\triangle^\alpha, \triangle^{\alpha'}] = 0$ for any $\alpha, \alpha' \in \mathfrak{K}^2$, the operator splitting is error-free. Therefore, we exactly have

$$e^{\mathtt{i}t((-\triangle)^\alpha + (-\triangle)^{\alpha'})} = e^{\mathtt{i}t(-\triangle)^\alpha} e^{\mathtt{i}t(-\triangle)^{\alpha'}},$$

implying that (20) is equivalent to

$$\left\{\begin{array}{ll} \mathtt{i}\partial_t u^{(1)}(t,\boldsymbol{x}) + v_{\alpha_1}(t)(-\triangle)^{\alpha_1} u(t,\boldsymbol{x}) = 0, & u^{(1)}(0,\cdot) = u_0, \quad t \in [0,T], \\ \cdots & \\ \mathtt{i}\partial_t u^{(k)}(t,\boldsymbol{x}) + v_{\alpha_k}(t)(-\triangle)^{\alpha_k} u^{(k)}(t,\boldsymbol{x}) = 0, & u^{(k)}(0,\cdot) = u^{(k-1)}(T,\cdot), \ t \in [0,T], \\ \cdots & \\ \mathtt{i}\partial_t u^{(m)}(t,\boldsymbol{x}) + v_{\alpha_m}(t)(-\triangle)^{\alpha_m} u^{(m)}(t,\boldsymbol{x}) = 0, & u^{(m)}(0,\cdot) = u^{(m-1)}(T,\cdot), \ t \in [0,T]. \end{array}\right.$$

This concludes the proof. □

From (21) and for t_n to t_{n+1}, let us introduce the discrete times $t_{n_k} = t_n + \Delta t$ for any $k \in \{1, \cdots, m-1\}$ and define $u^{n_k} = u(t_{n_k}, \cdot)$. We then propose the following modified system of fractional equations on $(0,T) \times \mathcal{D}$ containing the PML contribution

$$\left\{\begin{array}{l} \mathtt{i}\partial_t (-\triangle)^{1-\alpha_1} u(t,\boldsymbol{x}) - v_{\alpha_1}(t)\triangle_{\mathrm{PML}} u(t,\boldsymbol{x}) = 0, \quad u(t_n,\cdot) = u^n, \quad t \in [t_n, t_{n_1}], \\ \cdots \\ \mathtt{i}\partial_t (-\triangle)^{1-\alpha_k} u(t,\boldsymbol{x}) - v_{\alpha_k}(t)\triangle_{\mathrm{PML}} u(t,\boldsymbol{x}) = 0, \ u(t_n,\cdot) = u^{n_{k-1}}, \ t \in [t_n, t_{n_k}], \\ \cdots \\ \mathtt{i}\partial_t (-\triangle)^{\alpha_m} u(t,\boldsymbol{x}) - v_{\alpha_m}(t)\triangle_{\mathrm{PML}} u(t,\boldsymbol{x}) = 0, \quad u(t_n,\cdot) = u^{n_{m-1}}, \ t \in [t_n, t_{n+1}]. \end{array}\right.$$

By using the same notations as [8] and for $t_n < t_{n+1} = t_n + \Delta t$, a straightforward implicit scheme in time consists is successively solving the following system of linear equations

$$\left\{\begin{array}{l} [[(-\triangle)^{1-\alpha_1}]]\boldsymbol{u}_h^{n_1} - \mathtt{i}c(t_{n_1})\Delta t[[\triangle_{\mathrm{PML}}]]\boldsymbol{u}_h^{n_1} = [[(-\triangle)^{1-\alpha_1}]]\boldsymbol{u}_h^{n}, \\ \cdots \\ [[(-\triangle)^{1-\alpha_k}]]\boldsymbol{u}_h^{n_k} - \mathtt{i}c(t_{n_k})\Delta t[[\triangle_{\mathrm{PML}}]]\boldsymbol{u}_h^{n_k} = [[(-\triangle)^{1-\alpha_k}]]\boldsymbol{u}_h^{n_{k-1}}, \\ \cdots \\ [[(-\triangle)^{1-\alpha_m}]]\boldsymbol{u}_h^{n+1} - \mathtt{i}c(t_{n+1})\Delta t[[\triangle_{\mathrm{PML}}]]\boldsymbol{u}_h^{n+1} = [[(-\triangle)^{1-\alpha_m}]]\boldsymbol{u}_h^{n_{m-1}}, \end{array}\right.$$

where $\boldsymbol{u}_h^{n_k}$ is an approximation to u on $\mathcal{D}_{N_x,N_y}$ at time t_{n_k}, and where $[[\triangle]]$ is the pseudospectral approximation (14) of $\triangle$ with $\alpha = 1$. Higher order splitting schemes can easily be derived to increase the accuracy of the solver and to reduce the numerical dispersion.

We can extend the above approach to problems with space-dependent coefficients. However, some additional work is necessary. We consider

$$\left\{\begin{array}{l} \mathtt{i}\partial_t u(t,x,y) + \displaystyle\sum_{\alpha\in\mathfrak{K}} v_\alpha(t,x,y)(-\triangle)^\alpha u(t,x,y) = 0, \quad (t,x,y) \in [0,T]\times\mathbb{R}^2, \\ u(t=0,x,y) = u_0(x,y), \quad (x,y) \in \mathbb{R}^2, \end{array}\right. \tag{22}$$

where $\{v_\alpha\}_{\alpha\in\mathfrak{K}}$ correspond to smooth real- or purely complex-valued functions, and $\mathfrak{K}$ is a given finite set of strictly positive numbers. Naturally, we can no more directly apply the above transformation. Instead, we proceed as follows. For any exponent $\alpha \in \mathfrak{K}$, we set $\psi_\alpha = (-\triangle)^{\alpha-1}u$, and then now consider

$$\left\{\begin{array}{l} \mathtt{i}\partial_t u(t,x,y) + \displaystyle\sum_{\alpha\in\mathfrak{K}} v_\alpha(t,x,y)(-\triangle)\psi_\alpha(t,x,y) = 0, \quad (t,x,y) \in [0,T]\times\mathbb{R}^2, \\ u(t,x,y) = (-\triangle)^{1-\alpha}\psi_\alpha(t,x,y)\,, \\ u(t=0,x,y) = u_0(x,y)\,. \end{array}\right.$$

For the equation on a bounded domain $\mathcal{D}$ with periodic boundary conditions and using PML, we analyze

$$\left\{\begin{array}{l} \mathtt{i}\partial_t u(t,x,y) + \displaystyle\sum_{\alpha\in\mathfrak{K}} v_\alpha(t,x,y)(-\triangle_{\mathrm{PML}})\psi_\alpha(t,x,y) = 0, \quad (t,x,y) \in [0,T]\times\mathcal{D}, \\ u(t,x,y) = (-\triangle)^{1-\alpha}\psi_\alpha(t,x,y)\,, \\ u(t=0,x,y) = u_0(x,y)\,. \end{array}\right.$$

Starting from $k = 0$ with $u^{(0)}$ given, we iteratively construct a sequence $\{u^{(k)}\}_k$ such that for $k \geqslant 1$

$$\left\{\begin{array}{l} \mathtt{i}\partial_t u^{(k)}(t,x,y) + \displaystyle\sum_{\alpha\in\mathfrak{K}} v_\alpha(t,x,y)(-\triangle_{\mathrm{PML}})\psi_\alpha^{(k)}(t,x,y) = 0, \quad (t,x,y) \in [0,T]\times\mathcal{D}, \\ \psi_\alpha^{(k)}(t,x,y) = (-\triangle)^{\alpha-1}u^{(k-1)}(t,x,y)\,, \\ u(t=0,x,y) = u_0(x,y)\,. \end{array}\right. \tag{23}$$

Using the same notations as in Subsection 2.1, from t_n to t_{n+1}, the corresponding scheme then reads until convergence ($k \to +\infty$)

$$\begin{array}{l} \boldsymbol{u}_h^{(k),n+1} - \mathtt{i}\Delta t \sum_{\alpha\in\mathfrak{R}} \boldsymbol{v}_{\alpha,h}^{n+1}[[\triangle_{\mathrm{PML}}]]\boldsymbol{\psi}_{\alpha,h}^{(k),n+1} = \boldsymbol{u}_h^{(k),n} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \boldsymbol{\psi}_{\alpha,h}^{(k),n+1} = [[(-\triangle)^{\alpha-1}]]\boldsymbol{u}_h^{(k-1),n+1}\,, \end{array} \tag{24}$$

where $\boldsymbol{u}_h^{(k),n}$ (resp. $\boldsymbol{\psi}_{\alpha,h}^{(k),n}$) is an approximation to u (resp. ψ_α) on $\mathcal{D}_{N_x,N_y}$ at time t_n and iteration k, and where $[[\triangle_{\mathrm{PML}}]]$ is defined by (16).

Although the idea developed in the above remark may work, we rather explore a methodology which does not require the use of an iterative process. To this end, an alternative approach based on Padé's approximants is proposed for equations with space-dependent coefficients.

2.2 Padé's approximants for fractional equations with space-dependent coefficients

The iterative approach developed above may have slow convergence (and possibly divergence). We propose a direct methodology based on approximants of the fractional PML-Laplacian symbol for rational values of the exponents α. Different types of approximants exist, among which Padé's are the most common ones [31]. We first introduce

$$\begin{array}{l} \mathtt{i}\partial_t u(t,x,y) + \displaystyle\sum_{\alpha\in\mathfrak{R}} v_\alpha(x,y)(-\triangle)^\alpha u(t,x,y) = 0, \quad (t,x,y)\in[0,T]\times\mathbb{R}^2, \\ u(t=0,x,y) = u_0(x,y), \quad (x,y)\in\mathbb{R}^2, \end{array} \tag{25}$$

where $\{v_\alpha\}_{\alpha\in\mathfrak{R}}$ are smooth real- or purely complex-valued functions. Hence, we do not consider fractional Schrödinger equations with t-dependent coefficients. However, as long as the corresponding equation is well-posed, there is nothing that prevents the use of the method developed below in this case. For $\alpha\in\mathfrak{R}$ (which is supposed to be a finite set of strictly positive numbers), we search for an approximation to the fractional operator symbol

$$\sigma\big((-\triangle_{\mathrm{PML}})^\alpha\big) = \Big(\frac{1}{S_x^2}|\xi_x|^2 + \mathtt{i}\frac{1}{S_x}\partial_x(\frac{1}{S_x})\xi_x + \frac{1}{S_y^2}|\xi_y|^2 + \mathtt{i}\frac{1}{S_y}\partial_y(\frac{1}{S_y})\xi_y\Big)^\alpha. \tag{26}$$

The chosen approximant to $\sigma\big((-\triangle_{\mathrm{PML}})^\alpha\big)$ at order M (in the sense (29)), is denoted by $p_M^{(\alpha)}(x,y,\xi_x,\xi_y)$ and is assumed to be a power series/sum in ξ_x and ξ_y, such that $\big(-\triangle_{\mathrm{PML}}\big)^\alpha$ is approximated by $P_M^{(\alpha)}$ given by

$$\mathrm{Op}\big(p_M^{(\alpha)}(x,y,\cdot,\cdot)\big) = P_M^{(\alpha)}(x,y,\partial_x,\partial_y),$$

where

$$p_M^{(\alpha)}(x,y,\xi_x,\xi_y) = \big(|\xi_x|^2+|\xi_y|^2\big)^\alpha, \text{ for } (x,y)\in\mathcal{D}_{\mathrm{Phy}}\,.$$

We next consider the equation on a bounded domain $\mathcal{D} = [-L_x,L_x]\times[-L_y,L_y]$ with periodic boundary conditions:

$$\begin{cases} \mathtt{i}\partial_t u(t,x,y) + \displaystyle\sum_{\alpha\in\mathfrak{R}} v_\alpha(x,y)(-\triangle_{\mathrm{PML}})^\alpha u(t,x,y) = 0,\ (t,x,y)\in[0,T]\times\mathcal{D}, \\ u(t=0,x,y) = u_0(x,y), \quad (x,y)\in\mathcal{D}\,, \\ u(t,\pm L_x,y) = u(t,\mp L_x,y),\ u(t,x,\pm L_y) = u(t,x,\mp L_y)\,. \end{cases} \tag{27}$$

More specifically, for fixed $M\in\mathbb{N}^*$, we use the approximate problem

$$\begin{cases} \mathtt{i}\partial_t u(t,x,y) + \displaystyle\sum_{\alpha\in\mathfrak{R}} v_\alpha(x,y)P_M^{(\alpha)}u(t,x,y) = 0, \quad (t,x,y)\in[0,T]\times\mathcal{D}\,, \\ u(t=0,x,y) = u_0(x,y), \quad (x,y)\in\mathcal{D}\,, \\ u(t,L_x,y) = u(t,-L_x,y),\ u(t,x,L_y) = u(t,x,-L_y)\,. \end{cases} \tag{28}$$

Since $\mathcal{D}_{\rm Phy} \subsetneq \mathcal{D}$, we then get

$$\mathtt{i}\partial_t u(t,x,y) + \sum_{\alpha\in\mathfrak{R}} v_\alpha(x,y)(-\triangle)^\alpha u(t,x,y) = 0, \qquad \text{for } (t,x,y)\in[0;T]\times\mathcal{D}_{\rm Phy}\,.$$

From a practical point of view, constructing the approximants is not an easy task. We detail the methodology for $\alpha = 1/2$, and for $\alpha \in \mathbb{N}/2^{\mathbb{N}}$ in A.

We consider the case of the square-root Laplacian (i.e. for $\alpha = 1/2$). It is shown in [1] that the θ-rotated Padé approximation at order $M\in\mathbb{N}$ of $\sqrt{z}$ is given by

$$\sqrt{z} \approx p_M^{(1/2)}(z) = \sum_{k=0}^{M} a_k^{(M)} - \sum_{k=1}^{M} \frac{a_k^{(M)} d_k^{(M)}}{z + d_k^{(M)}}, \tag{29}$$

where the coefficients are, for $\theta\in[0,\pi/2)$,

$$a_0^{(M)} = 0, \qquad a_k^{(M)} = \frac{e^{\mathtt{i}\theta}}{M\cos^2\Big(\dfrac{(2k+1)\pi}{4M}\Big)}, \qquad d_k^{(M)} = \tan^2\Big(\frac{(2k+1)\pi}{4M}\Big)e^{\mathtt{i}\theta}\,. \tag{30}$$

Then formally, in (27) we define

$$(-\triangle_{\rm PML})^{1/2} := \sum_{k=0}^{M} a_k^{(M)} + \sum_{k=1}^{M} a_k^{(M)} d_k^{(M)} \Big(\frac{1}{S_x^2}\partial_x^2 - \frac{S_x'}{S_x^3}\partial_x + \frac{1}{S_y^2}\partial_y^2 - \frac{S_y'}{S_y^3}\partial_y - d_k^{(M)}\Big)^{-1}. \tag{31}$$

Remark 1 The discretization of the RHS in (31) requires the solution to a low dimensional linear system (since it is only applied in the PML). More specifically, an equation of the form $(-\triangle_{\rm PML})^{1/2}u = f$ would be approximated by

$$\begin{cases} \sum_{k=0}^{M} a_k^{(M)} u + \sum_{k=1}^{M} a_k^{(M)} d_k^{(M)} \varphi_k = f\,, \\ \Big(\dfrac{1}{S_x^2}\partial_x^2 - \dfrac{S_x'}{S_x^3}\partial_x + \dfrac{1}{S_y^2}\partial_y^2 - \dfrac{S_y'}{S_y^3}\partial_y - d_k^{(M)}\Big)\varphi_k = u\,. \end{cases}$$

Let us consider the equation with periodic conditions at $\{(\pm L_x, y) : y\in[-L_y,L_y]\}$ and $\{(x,\pm L_y) : x\in[-L_x,L_x]\}$

$$\begin{cases} \mathtt{i}\partial_t u(t,x,y) + v(x,y)(-\triangle_{\rm PML})^{1/2}u(t,x,y) = 0, \quad (t,x,y)\in[0,T]\times\mathcal{D}\,, \\ u(t=0,x,y) = u_0(x,y), \quad (x,y)\in\mathcal{D}\,, \end{cases}$$

which then becomes the following system, still with periodic boundary conditions at $\{(\pm L_x, y) : y\in[-L_y,L_y]\}$ and $\{(x,\pm L_y) : x\in[-L_x,L_x]\}$

$$\begin{cases} \mathtt{i}\partial_t u + v(x,y)(-\triangle)^{1/2}u = 0, & (x,y)\in\mathcal{D}_{\rm Phy}\,, \\ \mathtt{i}\partial_t u + v(x,y)\big(\sum_{k=0}^{M} a_k^{(M)} u + \sum_{k=1}^{M} a_k^{(M)} d_k^{(M)}\varphi_k\big) = 0, & (x,y)\in\mathcal{D}_{\rm PML}\,, \\ \Big(\dfrac{1}{S_x^2}\partial_x^2 - \dfrac{S_x'}{S_x^3}\partial_x + \dfrac{1}{S_y^2}\partial_y^2 - \dfrac{S_y'}{S_y^3}\partial_y - d_k^{(M)}\Big)\varphi_k = u, & (x,y)\in\mathcal{D}_{\rm PML}\,. \end{cases} \tag{32}$$

The implicit time semi-discrete version then reads

$$\begin{cases} u^{n+1} - \mathtt{i}\Delta t v(x,y)(-\triangle)^{1/2}u^{n+1} = u^n,\ (t,x,y)\in[t_n,t_{n+1}]\times\mathcal{D}_{\rm Phy}\,, \\ u^{n+1} - \mathtt{i}\Delta t v(x,y)\big(\sum_{k=0}^{M} a_k^{(M)} u^{n+1} + \sum_{k=1}^{M} a_k^{(M)} d_k^{(M)}\varphi_k^{n+1/2}\big) = u^n,\ (t,x,y)\in[t_n,t_{n+1}]\times\mathcal{D}_{\rm PML}\,, \\ \Big(\dfrac{1}{S_x^2}\partial_x^2 - \dfrac{S_x'}{S_x^3}\partial_x + \dfrac{1}{S_y^2}\partial_y^2 - \dfrac{S_y'}{S_y^3}\partial_y - d_k^{(M)}\Big)\varphi_k^{n+1/2} = u^n,\ (x,y)\in\mathcal{D}_{\rm PML}\,. \end{cases}$$

Finally, for $t_n < t_{n+1} = t_n + \Delta t$, a fully discrete version reads as follows. For $\varphi_h = \{\varphi_{i,j}\}_{(i,j)\in\mathbb{Z}^2}$, we propose the following second-order approximation of the Laplace operator, with fixed space steps Δx and Δy,

$$\begin{cases} \partial_{x,h}\varphi_{i,j} = \dfrac{\varphi_{i+1,j} - \varphi_{i-1,j}}{2\Delta x},\ \partial_{y,h}\varphi_{i,j} = \dfrac{\varphi_{i,j+1} - \varphi_{i,j-1}}{2\Delta y}, \\ -\triangle_{x,h}\varphi_{i,j} = \dfrac{\varphi_{i+2,j} - 16\varphi_{i+1,j} + 30\varphi_{i,j} - 16\varphi_{i-1,j} + \varphi_{i-2,j}}{12\Delta x^2}, \\ -\triangle_{y,h}\varphi_{i-2,j} = \dfrac{\varphi_{i,j+2} - 16\varphi_{i,j+1} + 30\varphi_{i,j} - 16\varphi_{i,j-1} + \varphi_{i,j-2}}{12\Delta y^2}. \end{cases}$$

We introduce below a second-order scheme (in space) in the PML, so that a 3-point discretization of the Laplacian in each direction is sufficient. A fourth-order approximation of the Laplacian such as $-\triangle_{x,h}\varphi_{i,j}$ is hence not required. Setting

$$\boldsymbol{w}_h = \boldsymbol{u}_{h|\mathcal{D}_{\rm PML}},\ \boldsymbol{\nu}_h = \boldsymbol{u}_{h|\mathcal{D}_{\rm Phy}}\,,$$

we consider the following numerical scheme approximating (32), from t_{n+1} to t_n,

$$\begin{cases} \boldsymbol{\nu}_h^{n+1} - \mathtt{i}\Delta t \boldsymbol{v}_h\big([[\triangle]]\boldsymbol{u}_h^{n+1}\big)_{|\rm Phy} = \boldsymbol{\nu}_h^n, \ \text{ in } \mathcal{D}_{\rm Phy}\,, \\ \Big(\mathbf{I}_h - \mathtt{i}\Delta t \boldsymbol{v}_h \sum_{k=0}^{M} a_k^{(M)}\Big)\boldsymbol{w}_h^{n+1} = \mathtt{i}\Delta t \boldsymbol{v}_h \sum_{k=1}^{M} a_k^{(M)} d_k^{(M)} \boldsymbol{\varphi}_{k,h}^{n+1/2}\,, \\ \Big(\dfrac{1}{\boldsymbol{S}_{x,h}^2}\triangle_{x,h} + \dfrac{1}{\boldsymbol{S}_{y,h}^2}\triangle_{y,h} - \dfrac{\boldsymbol{S}'_{x,h}}{\boldsymbol{S}_{x,h}^3}\partial_{x,h} - \dfrac{\boldsymbol{S}'_{y,h}}{\boldsymbol{S}_{y,h}^3}\partial_{y,h} - d_k^{(M)}\mathbf{I}_h\Big)\boldsymbol{\varphi}_{k,h}^{n+1/2} = \boldsymbol{w}_h^n \ \text{ in } \mathcal{D}_{\rm PML}\,, \end{cases} \tag{33}$$

where $\boldsymbol{S}_{\nu,h}$ denotes $\{S_\nu(\nu_i)\}_i$. Let us remark that the third system in (33), can be implemented independently as

$$\boldsymbol{A}_h^{(k)}\boldsymbol{\varphi}_{k,h}^{n+1} = \boldsymbol{w}_h^{n+1}\,,$$

with $\boldsymbol{A}_h^{(k)} = \{A_{i,j}^{(k)}\}_{i,j}$ such that

$$A_{i,i}^{(k)} = 2\big(\frac{1}{S_x^2(x_i)\Delta x^2} + \frac{1}{S_y^2(y_j)\Delta y^2}\big) - d_k^{(M)}\,,$$

$$A_{i,i+1}^{(k)} = -\frac{1}{S_x^2(x_i)\Delta x^2} + \frac{S'_x(x_i)}{S_x^3(x_i)\Delta x},\ A_{i-1,i}^{(k)} = -\frac{1}{S_x^2(x_i)\Delta x^2} - \frac{S'_x(x_i)}{S_x^3(x_i)\Delta x},$$

and finally

$$A_{i,i+N_x}^{(k)} = -\frac{1}{S_y^2(y_j)\Delta y^2} + \frac{S'_y(y_j)}{S_y^3(y_j)\Delta y},\ A_{i-N,i}^{(k)} = -\frac{1}{S_y^2(y_j)\Delta y^2} - \frac{S'_y(y_j)}{S_y^3(y_j)\Delta y}.$$

The other entries are zero. We trivially get that the matrices $\{\boldsymbol{A}_h^{(k)}\}_{1\leqslant k\leqslant M}$ are symmetric and invertible. The two first equations can be rewritten in the form

$$\mathcal{A}_h\boldsymbol{u}_h^{n+1} = \boldsymbol{\mathcal{F}}_h^{n+1/2}, \qquad \boldsymbol{\mathcal{F}}_h^{n+1/2} := \big(\boldsymbol{\nu}_h^n, \mathtt{i}\Delta t\boldsymbol{v}_h\sum_{k=1}^{M} a_k^{(M)} d_k^{(M)}\boldsymbol{\varphi}_h^{n+1/2}\big)^T.$$

A numerical example is proposed to illustrate the approach based on Padé's approximants in the 1-d case. The computational domain is $\mathcal{D} = [-10, 10]$, with $\mathcal{D}_{\rm Phy} = [-9.5, 9.5]$. We take $\Delta x = 5\times10^{-2}$, $\Delta t = 7.5\times10^{-2}$ and $N_T = 500$. The initial data is given by

$$u_0(x) = 10\exp\big(-(x-20/3)^2 + \mathtt{i}k_0 x\big),$$

with $k_0 = 10$. We consider a PML-solution with $\sigma(x) = \exp(5x)$, $\Delta t = 2\times10^{-2}$, and compare it in Fig. 2, with a reference solution computed on a larger domain and a solution with periodic boundary

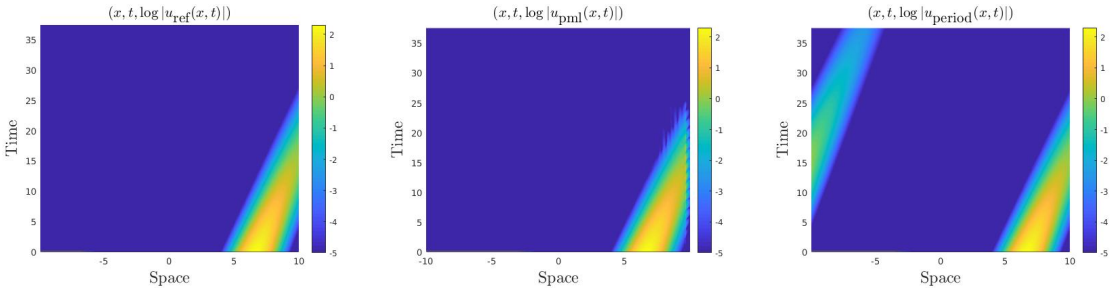

Fig. 2 Solutions of reference (Left), PML (Middle) and solution computed with periodic boundary conditions (Right) : $\{(x,t,\log|u(x,t)|),(x,t)\in\mathcal{D}\times[0,T]\}$.

conditions (we represent $\{(x,t,\log|u(x,t)|),(x,t)\in\mathcal{D}\times[0,T]\}$). This shows the absorbing effect of the PML, which mainly avoids the periodic transmission of the outgoing waves.

In the next example, we compare the quality of the PML absorption for increasing orders of the Padé's approximation with $M=4,8,12$, for $\theta=2\pi/7$. The same test is next performed, for $M=12$ but for different values of the rotation angle: $\theta=0,\pi/8,\pi/4,2\pi/7,\pi/3$, see Fig. 3 (Left) and (Right). These experiments illustrate the importance of properly selecting the PML parameters to enhance the absorption and to optimize the PML. Clearly, the full optimization of the tuning parameters is an important question which has to be addressed in a future and is known to be nontrivial. We also check the stability and absorption of the method with $M=12$ and $\theta=7\pi/2$, by representing the ℓ^2-norm of the PML/Reference/Periodic-BC solutions as a function of time in Fig. 4 (Left). A convergence graph as a function of M is proposed in Fig. 4 (Right). More specifically, in the latter we report the ℓ^2-norm error (PML-solution vs solution of reference) at final time, $\|\boldsymbol{u}_h^{n_T}-\boldsymbol{u}_{\text{Ref}}^{n_T}\|_2$, as a function of the number M of Padé approximants. Let us also mention [23] for the mathematical analysis of the stability of the numerical scheme in the fractional framework.

This generalization of the derivation of the PMLs for $\alpha=p/2^k$, $k\in\mathbb{N}^*$ and $p\in\mathbb{N}^*$, is detailed in Appendix A.

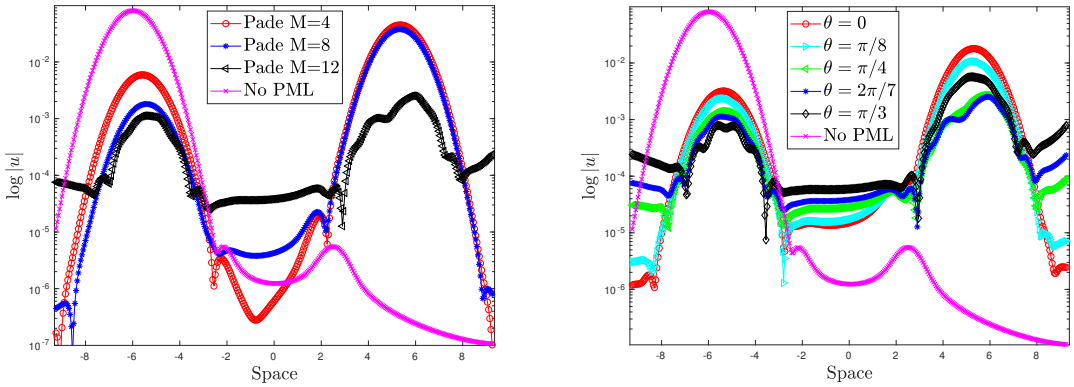

Fig. 3 (Left) PML resolution in logscale $\{(x,\log|u(x,T)|),x\in\mathcal{D}\}$ at $T=37.5$ for different values M of Padé functions 4, 8, 12, and $\theta=2\pi/7$; (Middle) for different values of θ: $0,\pi/8,\pi/4,2\pi/7,\pi/3$. (Right) ℓ^2-norm of $\boldsymbol{u}_h^n$ as a function of n for PML, Periodic BC and reference solution.

3 Approximation of fractional Laplacian equations using a system of ODEs

Fourier-based methods are particularly convenient to work with when dealing with fractional derivatives, due to the simple expression of fractional operators in the Fourier space based on the spectral definition.

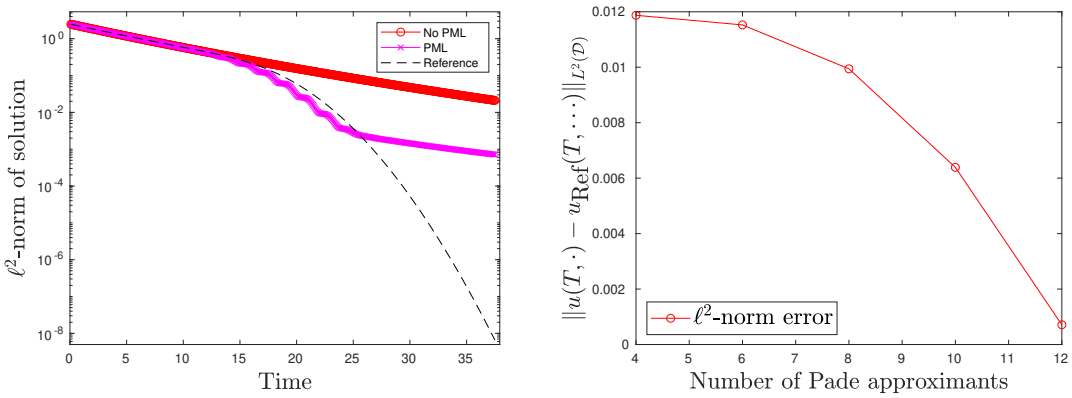

Fig. 4 (Left) ℓ^2-norm of $\boldsymbol{u}_h^n$ as a function of n for PML, Periodic BC and reference solution. (Right) ℓ^2-norm error at final time $\|\boldsymbol{u}_h^{n_T} - \boldsymbol{u}_{\mathrm{Ref}}^{n_T}\|_2$ as a function of number of Padé approximants.

However, the coupling with PMLs on bounded domains is not simple, in particular due to the complexity to implement Padé's approximants. Alternative approaches such as real space methods should be developed. The aim of this section is to address such a contribution. According for instance to [25], it is possible to approximate the fractional Laplacian by using a strategy based on differential equations to estimate the real power of a matrix. This idea was in particular developed in [7]. We detail below this approach, including absorbing layers into the formulation.

3.1 Differential equations formulation to approximate the fractional Laplacian

Let us recall first that for any matrix $A \in \mathbb{R}^{N\times N}$ with no negative eigenvalues, and for $\alpha \in \mathbb{R}$, the N-dimensional system

$$y'(\tau) = \alpha(A-I)\big(\tau(A-I)+I\big)^{-1} y(\tau),\ y(0) = b, \tag{34}$$

is such that $y(\tau) = \big(\tau(A-I)+I\big)^{\alpha} b$, $y(1) = A^{\alpha} b$, where I is the identity matrix in $\mathbb{R}^{N\times N}$. The latter is a standard method for solving fractional algebraic linear systems. The approximation of $(-\triangle)^{\alpha} u$ for any $u \in W^{\alpha,p}(\mathbb{R}^N)$ is computed by discretizing (34), for $p \in [1,\infty)$ and $\alpha \in (0,1)$. Later on, A will represent an approximate one- (resp. two-) dimensional Laplacian $-\Delta$, for $N = N_x$ (resp. $N = N_x N_y$) spatial discretization points.

We denote by $\mathit{\Pi}_h$ a projector from $C(\mathbb{R}^2;\mathbb{C}) \cap L^2(\mathbb{R}^2;\mathbb{C})$ to $\ell^2(h\mathbb{Z}\times h\mathbb{Z})$, where h is a spatial discretization step. We define $\boldsymbol{u}_h \in \mathbb{C}^N$ as a finite-difference approximation of u on $\ell^2(h\mathbb{Z}\times h\mathbb{Z})$, and $\boldsymbol{A}_h \in \mathbb{R}^{N\times N}$ as a finite-difference approximation of the Laplacian for solving (34). To compute $\mathbf{u}_h$, we can proceed as follows. We first construct the sequence $\{\boldsymbol{w}_h^k\}_k$ such that

$$\Big(\mathbf{I}_h - \delta\tau\alpha(\boldsymbol{A}_h - \mathbf{I}_h)\boldsymbol{B}_h^{(k)}\Big)\boldsymbol{w}_h^{k+1} = \boldsymbol{w}_h^k, \qquad \boldsymbol{w}_h^0 = \boldsymbol{u}_h, \tag{35}$$

where $\tau_k = k\delta\tau$, for $k \in \mathbb{N}$, and defining

$$\boldsymbol{C}_h^{(k)} = \big(\boldsymbol{B}_h^{(k)}\big)^{-1} = \big(\tau_{k+1}\boldsymbol{A}_h + (1-\tau_{k+1})\mathbf{I}_h\big) = \tau_{k+1}\big(\boldsymbol{A}_h + (\tau_{k+1}^{-1} - 1)\mathbf{I}_h\big)\,. \tag{36}$$

System (35) can also be written

$$\boldsymbol{C}_h^{(k)}\Big(\mathbf{I}_h - \delta\tau\alpha(\boldsymbol{A}_h - \mathbf{I}_h)\Big)\boldsymbol{B}_h^{(k)}\boldsymbol{w}_h^{k+1} = \boldsymbol{w}_h^k,$$

leading to the alternative formulation

$$\begin{aligned} \boldsymbol{C}_h^{(k)}\Big(\mathbf{I}_h - \delta\tau\alpha(\boldsymbol{A}_h - \mathbf{I}_h)\Big)\boldsymbol{\omega}_h^{k+1,*} &= \boldsymbol{w}_h^k\,, \\ \boldsymbol{w}_h^{k+1} = \boldsymbol{C}_h^{(k)}\boldsymbol{\omega}_h^{k+1,*}, &\qquad \boldsymbol{w}_h^0 = \boldsymbol{u}_h\,. \end{aligned} \tag{37}$$

As a consequence, for $k_T = \lfloor 1/\delta\tau \rfloor$, $\mathit{\Pi}_h\big((-\triangle)^\alpha u\big)$ is approximated by $\boldsymbol{w}_h^{k_T}$. We get the error estimate

$$\|\mathit{\Pi}_h\big((-\triangle)^\alpha u\big) - \boldsymbol{w}_h^{k_T}\|_2 = O(\delta\tau^p + h^q)\,,$$

where p is the order of the ODE solver for (34) ($p = 1$ here) and q is the order of the finite-difference approximation of the Laplace operator. The main issue in the above scheme is the need to solve a linear system for evaluating $\boldsymbol{w}_h^{(k+1)}$. For instance, if $\boldsymbol{A}_h$ is a five-point stencil finite-difference operator approximating the two-dimensional Laplace operator, it has the following eigenvalues and corresponding eigenvectors [1]

$$\begin{aligned}
\lambda_h^{(l_x,l_y)} &= \frac{4}{\Delta x^2}\sin^2\Big(\frac{l_x\pi}{2(N_x+1)}\Big) + \frac{4}{\Delta y^2}\sin^2\Big(\frac{l_y\pi}{2(N_y+1)}\Big),\\
\mathrm{v}_{h,j_x,j_y}^{(l_x,l_y)} &= \sin\Big(j_x\frac{l_x\pi}{N_x+1}\Big)\sin\Big(j_y\frac{l_y\pi}{N_y+1}\Big),\ (l_x,l_y)\in\{1,\cdots,N_x\}\times\{1,\cdots,N_y\}.
\end{aligned}$$

The eigenvalues of $\boldsymbol{B}_h^{(k)}$ are denoted by $\mu_h^{(l_x,l_y;k)} = \tau_{k+1}^{-1}\Big((\tau_{k+1}^{-1}-1) + \lambda_h^{(l_x,l_y)}\Big)^{-1}$. Then, the following proposition holds.

Proposition 2 *Let us assume that $\boldsymbol{A}_h$ is the five-point stencil finite-difference operator approximating the two-dimensional Laplace operator. Then, for any $\alpha\in\mathbb{R}$, we have*

$$\boldsymbol{A}_h^\alpha \boldsymbol{u}_h^{(0)} = \boldsymbol{P}_h \boldsymbol{\Lambda}_h^{(\alpha,k_T)} \boldsymbol{P}_h^{-1}\boldsymbol{u}_h^{(0)}\,, \tag{38}$$

where $\boldsymbol{\Lambda}_h^{(\alpha,k_T)} = diag\big(\zeta_h^{(1,1;k_T)},\cdots,\zeta_h^{(N_x,N_y;k_T)}\big)$ is a diagonal matrix with

$$\zeta_h^{(l_x,l_y;p)} = \big(\mathit{\Pi}_{p=0}^{k_T-1}\nu_h^{(l_x,l_y;p)}\big)^{-1},\ \nu_h^{(l_x,l_y;p)} = 1-\delta t\alpha(\lambda_h^{(l_x,l_y)}-1)\mu_h^{(l_x,l_y;p)},$$

and $\boldsymbol{P}_h$ is the matrix of the eigenvectors. Equivalently, one gets

$$\boldsymbol{A}_h^\alpha = \boldsymbol{P}_h\boldsymbol{\Lambda}_h^\alpha\boldsymbol{P}_h^{-1}\,, \tag{39}$$

with $\Lambda_h := diag\big(\lambda_h^{(1,1)},\cdots,\lambda_h^{(N_x,N_y)}\big)$.

Proof. The vector $\boldsymbol{A}_h^\alpha\boldsymbol{u}_h$ is approximated by $\boldsymbol{w}_h^{k_T}$, which is obtained iteratively by

$$\boldsymbol{w}_h^k = \Big(\mathbf{I}_h - \delta t\alpha\boldsymbol{P}_h(\boldsymbol{\Lambda}_h-\mathbf{I}_h)\,\tau_{k+1}^{-1}(\boldsymbol{\Lambda}_h + (\tau_{k+1}^{-1}-1)\mathbf{I}_h)^{-1}\,\boldsymbol{P}_h^{-1}\Big)\boldsymbol{w}_h^{k+1},\qquad \boldsymbol{w}_h^0 = \boldsymbol{u}_h.$$

For $0\leqslant p\leqslant k_T-1$, we have

$$\boldsymbol{A}_h^\alpha\boldsymbol{u}_h^{(0)} = \boldsymbol{u}_h^{(k_T)} = \boldsymbol{P}_h\boldsymbol{\Lambda}_h^{(\alpha,k_T)}\boldsymbol{P}_h^{-1}\boldsymbol{u}_h^{(0)}\,. \tag{40}$$

Since $\boldsymbol{A}_h$ is diagonalizable, i.e. $A_h = \boldsymbol{P}_h\boldsymbol{\Lambda}_h\boldsymbol{P}_h^{-1}$, we conclude that

$$\begin{aligned}
\boldsymbol{A}_h^\alpha &= (2\pi\mathtt{i})^{-1}\boldsymbol{A}_h\int_\Gamma z^{\alpha-1}(z\mathbf{I}_h-\boldsymbol{A}_h)^{-1}dz = \big(\boldsymbol{P}_h\boldsymbol{\Lambda}_h\boldsymbol{P}_h^{-1}\big)^\alpha\\
&= \boldsymbol{P}_h(2\pi\mathtt{i})^{-1}\boldsymbol{\Lambda}_h\int_\Gamma z^{\alpha-1}(z\mathbf{I}_h-\boldsymbol{\Lambda}_h)^{-1}dz\boldsymbol{P}_h = \boldsymbol{P}_h\boldsymbol{\Lambda}_h^\alpha\boldsymbol{P}_h^{-1}\,.
\end{aligned}$$

□

Let us remark that the differential equation approach is more general than (39), as the latter is restricted to diagonalizable matrices. The extension of this proposition to the 3-d case or to higher order finite-difference approximations with Dirichlet boundary conditions is straightforward. Regarding, the accuracy of order p of the approximation of $(-\triangle)^\alpha u$, let us recall that, for u regular enough, we have

$$-\triangle_h u = -\triangle u + \mathcal{O}\big(h^p R_1(u)\big),$$

so that, for homogeneous Dirichlet boundary conditions [35], we get at least

$$(-\triangle)_h^\alpha u = (-\triangle)^\alpha u + \mathcal{O}\big(h^{p\alpha}R_\alpha(u)\big),$$

where R_1 and R_α are some smooth differential operators. This result justifies the computation of the real power α of $\boldsymbol{A}_h$. In principle, this is no longer valid for nonzero Dirichlet boundary conditions (see e.g. [35]). Let us now state the following proposition.

Proposition 3 *Let us consider the scheme* (35)-(36) *approximating* (34) *for computing* $-(-\triangle)^\alpha u$ *in one-dimension, for* $u \in H^{2\alpha}(\mathcal{D})$. *We assume that the Laplace operator is approximated by a* K*-point stencil finite-difference scheme using the matrix* $\boldsymbol{A}_h$, *on a uniform finite spatial grid* $\{x_j\}_{-J\leqslant j\leqslant J}$. *Then, there exists a real-valued positive constant* $C > 0$ *such that*

$$\max_{-J\leqslant j\leqslant J} \left|(-\triangle)^\alpha u_0(x_j) - \boldsymbol{A}_h^\alpha \boldsymbol{u}_h^{(0)}\right| \leqslant C(\Delta x^{K-1} + \delta t).$$

Proof. First, in (35)-(36), the computation of $\boldsymbol{B}_h^{(k)}$ and $\boldsymbol{w}_h^{k+1}$ is performed exactly. The Δx contribution comes from the $(K-1)$th order approximation of $-\triangle$ by a K-point stencil finite-difference scheme. The first-order error in time is finally due to the approximate solution of the differential system based on the backward Euler scheme (35). □

To illustrate this approach, we approximately compute $(-\triangle)^\alpha u_0$ in 1-d with $\alpha = 1/2$ on $\mathcal{D} = [-10, 10]$, with null Dirichlet boundary conditions, and

$$u_0(x) = \exp(-x^2/10)\cos(2x)/\mathcal{N},$$

where $\mathcal{N}$ is the L^2-normalization constant. We discretize the Laplace operator by using a three-point stencil finite-difference scheme on a grid with 1001 points. We initially project u_0 on the finite-difference grid, leading to $\boldsymbol{u}_h^{(0)} = \{u_0(x_j)\}_{1\leqslant j\leqslant 1001}$. We compare $\boldsymbol{u}_h^{(k_T)}$ with $\boldsymbol{A}_h^\alpha \boldsymbol{u}_h^{(0)}$ for different values of k_T (i.e. thanks to δt since $k_T \delta t = 1$), reporting in Fig. 5 (Left) the error $\|\boldsymbol{A}_h^\alpha \boldsymbol{u}_h^{(0)} - \boldsymbol{u}_h^{(k_T)}\|_{\ell^2(\mathcal{D})}$ (in logscale) as a function of $\delta t_i = 0.125/2^i$, for $i = 1, \cdots, 6$. In particular, we observe the first-order accuracy of the ODE solver. We also plot $\boldsymbol{A}_h^\alpha \boldsymbol{u}_h^{(0)}$ and $\boldsymbol{u}_h^{(k_T)}$, for $k_T = 256$ and $N = 1001$ in Fig. 5 (Right).

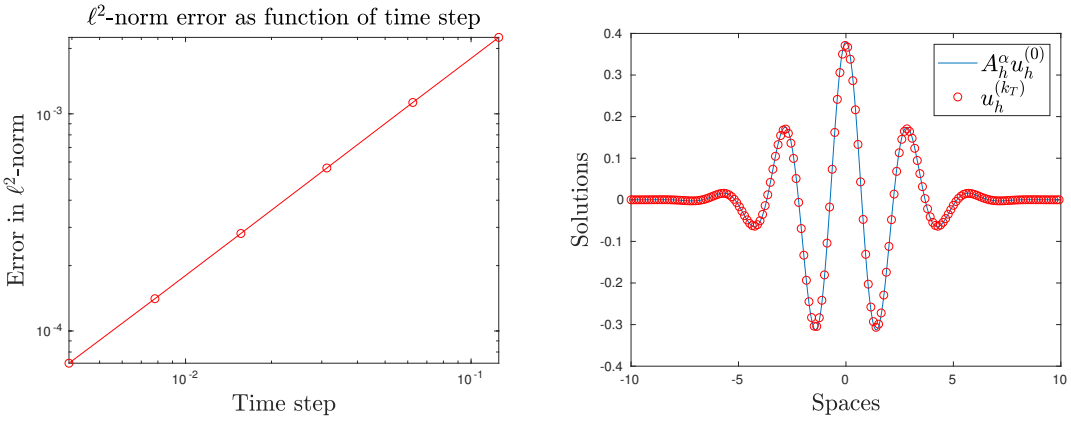

Fig. 5 (Left) error in ℓ^2-norm (logscale), i.e. $\|\boldsymbol{A}_h^\alpha \boldsymbol{u}_h^{(0)} - \boldsymbol{u}_h^{(k_T)}\|_{\ell^2(\mathcal{D})}$, with respect to Δt. (Right) Solutions $\boldsymbol{A}_h^\alpha \boldsymbol{u}_h^{(0)}$ and $\boldsymbol{u}_h^{(k_T)}$, for $k_T = 256$ and $N = 1001$.

We propose a second simple test focusing on the computation of $\boldsymbol{A}_h^\alpha$. We report in Fig. 6 (logscale) the CPU-time *in seconds* for different methods to compute $(-\triangle)^\alpha u_0$, with a $L^2(\mathcal{D})$-normalized data $u_0(x) = \exp(-10x^2)\cos(2x)/\mathcal{N}$ over $\mathcal{D} = [-2, 2]$, with $N = 1 + 2^{6+p}$ $(p = 1, \cdots, 7)$ grid points. We compare the *Direct Finite-Difference* (DFD) method where the solution to (37) is obtained using GMRES, and the *Optimized Finite-Difference* (OFD) method using (40) to efficiently compute the matrix power. In the latter case, the Laplace operator is approximated by a three-point stencil scheme and $k_T = 1, 50, 100$ in (35). These numerical results show that the OFD method is much faster than the DFD approach, and is almost independent of k_T. Indeed, it only requires to solve one linear system unlike the DFD method which requires the computation of k_T linear systems/

3.2 Application to the numerical solution of time-dependent fractional Laplacian equations

Based on the previous approximations, we can solve time-dependent fractional Laplacian equations. Compared to FFT-based approaches, it is necessary to solve a differential system at each time iteration

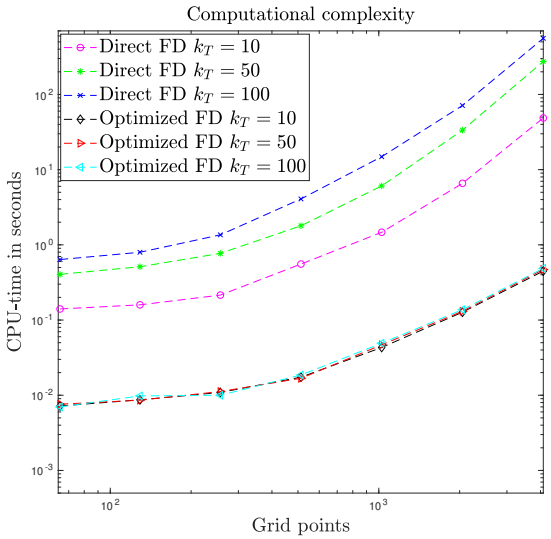

Fig. 6 CPU time vs. the number of grid points (logscale) to compute $(-\triangle)^{\alpha} u_0$. Comparison between the Direct Finite Difference Method (37) using GMRES and the Optimized Finite Difference based on (40).

to update the value of the fractional Laplacian. More specifically, we consider

$$\begin{cases} \partial_t u(t,x,y) + \displaystyle\sum_{\alpha \in \mathfrak{R}} v_\alpha(x,y)(-\triangle)^{\alpha} u(t,x,y) = 0, \quad \text{for } (t,x,y) \in [0,T] \times \mathbb{R}^2, \\ u(t=0,x,y) = u_0(x,y), \quad \text{for } (x,y) \in \mathbb{R}^2, \end{cases} \tag{41}$$

where $\{v_\alpha\}_{\alpha \in \mathfrak{R}}$ are smooth positive real-valued functions, and $\mathfrak{R}$ is *a finite set of strictly positive real numbers*. We propose the following implicit time-discretization

$$u^{n+1} + \Delta t \sum_{\alpha \in \mathfrak{R}} v_\alpha(x,y)(-\triangle_{\mathrm{PML}})^{\alpha} u^{n+1} = u^n, \ (t,x,y) \in [t_n, t_{n+1}] \times \mathcal{D}\,.$$

For two vectors $\mathbf{v} = \{v_j\}_{1 \leqslant j \leqslant N}$, $\mathbf{w} = \{w_j\}_{1 \leqslant j \leqslant N}$ in $\mathbb{R}^N$, we define $\mathbf{v} \otimes \mathbf{w} \in \mathbb{R}^N$ as:

$$\mathbf{v} \otimes \mathbf{w} = \{v_j w_j\}_{1 \leqslant j \leqslant N}. \tag{42}$$

Then, for each $\alpha \in \mathfrak{R} \cap \mathbb{R} \backslash \mathbb{N}$, $(-\triangle)^{\alpha} u^{n+1}$ is approximated by $\boldsymbol{A}_h^{\alpha} \boldsymbol{u}_h^{n+1}$ solution to

$$\begin{cases} \boldsymbol{u}_h^{n+1} + \Delta t \displaystyle\sum_{\alpha \in \mathfrak{R}} \boldsymbol{v}_{\alpha,h} \otimes \boldsymbol{A}_h^{\alpha} \boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n, \\ \boldsymbol{A}_h^{\alpha} \boldsymbol{u}_h^{n+1} = \boldsymbol{w}_h^{(\alpha),n,k_T}, \end{cases} \tag{43}$$

where, *for each* α, $\boldsymbol{w}_h^{(\alpha),n,k_T}$ is computed from

$$\Big(\mathbf{I}_h - \Delta t \alpha (\boldsymbol{A}_h - \mathbf{I}_h) \boldsymbol{B}_h^{(k)}\Big) \boldsymbol{w}_h^{(\alpha),n,k+1} = \boldsymbol{w}_h^k, \qquad \boldsymbol{w}_h^{(\alpha),n,0} = \boldsymbol{u}_h^{n+1}. \tag{44}$$

According to Proposition 2, imposing some Dirichlet boundary conditions at $x = \pm L_x$ and $y = \pm L_y$, we can rewrite the scheme as the following linear system (see (42))

$$\Big(\mathbf{I}_h + \Delta t \sum_{\alpha \in \mathfrak{R}} \boldsymbol{v}_{\alpha,h} \otimes \boldsymbol{P}_h \boldsymbol{\Lambda}_h^{(\alpha,k_T)} \boldsymbol{P}_h^{-1}\Big) \boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n\,. \tag{45}$$

If the equation has constant coefficients $\{v_\alpha\}_{\alpha \in \mathfrak{R}}$, it is not necessary to use a linear system solver *at each time iteration*. Indeed, in the latter case the scheme is simply to

$$\Big(\mathbf{I}_h + \Delta t \sum_{\alpha \in \mathfrak{R}} v_\alpha \boldsymbol{P}_h \boldsymbol{\Lambda}_h^{(\alpha,k_T)} \boldsymbol{P}_h^{-1}\Big) \boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n\,. \tag{46}$$

Setting $\boldsymbol{v}_h^{(k)} = \boldsymbol{P}_h^{-1}\boldsymbol{u}_h^{(k)}$, we obtain the explicit expression

$$\boldsymbol{v}_h^{n+1} = \Big(\mathbf{I}_h + \Delta t \sum_{\alpha\in\mathfrak{K}} v_\alpha \boldsymbol{\Lambda}_h^{(\alpha,k_T)}\Big)^{-1} \boldsymbol{v}_h^n\,. \tag{47}$$

Then, $\boldsymbol{u}_h^{(n+1)} = \boldsymbol{P}_h\boldsymbol{v}_h^{(n+1)}$ can be solved if needed. Thus, we can state the

Proposition 4 *For any $\alpha \in \mathbb{R}$, the scheme (45) approximating (41) is unconditionally ℓ^2-stable.*

Proof. The entries of $\boldsymbol{\Lambda}_h^{(\alpha,k_T)}$ as well as the functions $\{v_\alpha\}_{\alpha\in\mathfrak{K}}$ are positive, so that

$$\rho\Big(\mathbf{I}_h + \Delta t \sum_{\alpha\in\mathfrak{K}} \boldsymbol{v}_{\alpha,h}\otimes \boldsymbol{P}_h\boldsymbol{\Lambda}_h^{(\alpha,k_T)}\boldsymbol{P}_h^{-1}\Big) \geqslant 1,$$

where $\rho(A)$ designates the spectral radius of a given matrix A. We then deduce from (45) that we have the inequality $\|\boldsymbol{u}_h^{n+1}\|_{\ell^2} \leqslant \|\boldsymbol{u}_h^n\|_{\ell^2}$. □

In fact, we can extend the above proposition to fractional Schrödinger equations.

Proposition 5 *Consider the fractional Schrödinger equation with homogeneous Dirichlet boundary conditions*

$$\begin{cases} \mathtt{i}\partial_t u(t,x,y) + \displaystyle\sum_{\alpha\in\mathfrak{K}} v_\alpha(x,y)(-\triangle)^\alpha u(t,x,y) = 0, \quad \text{for } (t,x,y)\in[0,T]\times\mathcal{D},\\ u(t=0,x,y) = u_0(x,y), \quad \text{for } (x,y)\in\mathcal{D}\,,\\ u(t,\pm L_x,y) = u(t,x,\pm L_y) = 0, \end{cases} \tag{48}$$

where $\{v_\alpha\}_{\alpha\in\mathfrak{K}}$ are smooth real-valued functions. We further assume that the problem is well-posed. Then, the following schemes are unconditionally stable

– *Implicit Euler scheme*

$$\boldsymbol{u}_h^{n+1} - \mathtt{i}\Delta t \sum_{\alpha\in\mathfrak{K}} \boldsymbol{v}_{\alpha,h}\otimes \boldsymbol{P}_h\boldsymbol{\Lambda}_h^{(\alpha,k_T)}\boldsymbol{P}_h^{-1}\boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n\,. \tag{49}$$

– *Crank-Nicolson scheme*

$$\begin{aligned}\Big(\mathbf{I}_h - \mathtt{i}\frac{\Delta t}{2}\sum_{\alpha\in\mathfrak{K}} \boldsymbol{v}_{\alpha,h}\otimes \boldsymbol{P}_h\boldsymbol{\Lambda}_h^{(\alpha,k_T)}\boldsymbol{P}_h^{-1}\Big)\boldsymbol{u}_h^{n+1}\\ = \Big(\mathbf{I}_h + \mathtt{i}\frac{\Delta t}{2}\sum_{\alpha\in\mathfrak{K}} \boldsymbol{v}_{\alpha,h}\otimes \boldsymbol{P}_h\boldsymbol{\Lambda}_h^{(\alpha,k_T)}\boldsymbol{P}_h^{-1}\Big)\boldsymbol{u}_h^{n}\,.\end{aligned} \tag{50}$$

In addition, the Crank-Nicolson scheme preserves the ℓ^2-norm.

As recalled in Proposition 2, we can alternatively use (39) to compute the power of $\boldsymbol{A}_h$. Then, we have

Proposition 6 *The scheme*

$$\boldsymbol{u}_h^{n+1} - \mathtt{i}\Delta t \sum_{\alpha\in\mathfrak{K}} \boldsymbol{v}_{\alpha,h}\otimes \boldsymbol{P}_h\boldsymbol{\Lambda}_h^{\alpha}\boldsymbol{P}_h^{-1}\boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n \tag{51}$$

is unconditionally stable.

Let us remark that, in presence of a potential V, the above analysis remains valid by simply slightly modifying the differential equation to solve. In particular, the following proposition is of interest for approximating $-(-\triangle)^\alpha + V$.

Proposition 7 *For any vector $V \in \mathbb{R}^N$, matrix $A \in \mathbb{R}^{N\times N}$ with no negative eigenvalues, and any $\alpha\in\mathbb{R}$, the N-dimensional system*

$$y'(\tau) = \alpha(A-I)\big(\tau(A-I)+I\big)^{-1}\big(y(\tau)+(VI)b\big),\ y(0)=b \tag{52}$$

is such that

$$y(\tau) = \big(\tau(A-I)+I\big)^\alpha b + \tau(VI)b, y(1) = (A^\alpha + VI)b,$$

where I is the identity matrix in $\mathbb{R}^{N\times N}$.

3.3 Computational complexity analysis

In this Subsection, we analyze the computational complexity of the above schemes for solving the two-dimensional time-dependent fractional Laplacian equations. Let us focus on (45), while the complexity for the other schemes could easily be deduced. The main computation for solving fractional Laplacian equations is related to the approximation of $(-\triangle)^{\alpha}u$ when α is not an integer. In the following, we assume for the sake of simplicity that $\mathfrak{K} \subset \mathbb{R}\backslash\mathbb{N}$ and we define $\mathfrak{k} := \#\mathfrak{K}$. We denote by k_T the number of iterations for solving (44) to compute $\boldsymbol{A}_h^{\alpha}\boldsymbol{u}_h^n$ for any given $n \geqslant 1$, and N is the total number of grid points ($\approx \sqrt{N}$ points in each direction x and y for the 2-d case). For each $\alpha \in \mathfrak{K} \cap \mathbb{R}\backslash\mathbb{N}$ and time iteration, k_T sparse linear systems (44) are solved, each requiring $O(N^{\gamma_1})$ operations for some $1 < \gamma_1 < 2$, i.e. $O(k_T \mathfrak{k} N^{\gamma_1})$ operations at any time iteration. Finally, for a total of N_T time iterations, the solution to (45) needs $O(N_T N^{\gamma_2})$ operations, with $1 < \gamma_2 < 2$. This leads to the following Proposition.

Proposition 8 *We consider the two-dimensional system* (41) *approximated by* (45) *in combination with* (44) *on a two-dimensional grid with N points. The overall computational complexity, for N_T time iterations (with k_T fixed in* (44)*), is $\mathcal{O}\big(N_T(N^{\gamma_2} + k_T \mathfrak{k} N^{\gamma_1})\big)$, for some exponents $1 < \gamma_1, \gamma_2 < 2$.*

Let us note that the computation of $\{\boldsymbol{w}^{(\alpha),n,k_T}\}_{\alpha\in\mathfrak{K}}$ is embarrassingly parallel. The main drawback of the scheme is the need to solve $\mathfrak{k}k_T$ systems (44) at each time iteration. However in the framework of Proposition 2, we directly get an evaluation of $\tau_{k+1}^{-1}\big((\boldsymbol{A}_h - \mathbf{I}_h)(\boldsymbol{A}_h + (\tau_{k+1}^{-1} - 1)\mathbf{I}_h)\big)^{-1}$ which would be otherwise highly complex. Moreover, the use of an implicit scheme for constructing $\boldsymbol{A}_h^{\alpha}\boldsymbol{u}_h^n$ allows for reducing k_T. When the equation has constant coefficients, the global complexity can even be drastically improved as explained in the following Proposition.

Proposition 9 *For constant coefficients v_α, we assume that we approximate the 2-d system* (41) *by* (44)-(45)*, based on N spatial grid points. Then, the global complexity is given by $\mathcal{O}\big(N_T N \log N\big)$ operations, plus a pre-computational cost of $\mathcal{O}\big(\mathfrak{k}\, k_T N\big)$ operations, where the linear system* (45)*, fractional Laplacian* (44)*, are both explicitly evaluated and accelerated via the Discrete Sine Transform (DST). By comparison, a direct implicit implementation of* (41)*, combined with* (44)*, requires $\mathcal{O}\big(N_T(\mathfrak{k}k_T N^{\gamma_1} + N^{\gamma_2})\big)$ operations, for some exponents $1 < \gamma_1, \gamma_2 < 2$ with a larger prefactor.*

Proof. If (41) has constant coefficients, the scheme is reduced to (46). By setting $\boldsymbol{v}_h^{(0)} = \boldsymbol{P}_h^{-1}\boldsymbol{u}_h^{(0)}$, we deduce that

$$\boldsymbol{v}_h^n = \Big(\mathbf{I}_h + \Delta t \sum_{\alpha\in\mathfrak{K}} v_\alpha \boldsymbol{\Lambda}_h^{(\alpha,k_T)}\Big)^{-n} \boldsymbol{v}_h^0 \,.$$

Since $\mathbf{I}_h + \Delta t \sum_{\alpha\in\mathfrak{K}} v_\alpha \Pi_{p=0}^{k_T-1} \boldsymbol{\Lambda}_h^{(\alpha,p)}$ is a diagonal matrix, its explicit construction (once for all) and evaluation at power $-n$ only requires $\mathcal{O}(nN + \mathfrak{k}\, k_T N)$ operations. Finally, at any time iteration n, it is possible to evaluate $\boldsymbol{u}_h^n = \boldsymbol{P}_h \boldsymbol{v}_h^n$ within $\mathcal{O}(N \log N)$ operations *via* the DST. To compute explicitly all the vectors $\boldsymbol{u}_h^n$ requires $\mathcal{O}(N_T N \log N)$ operations. If only $u_h^{N_T}$ is needed, the overall cost is $\mathcal{O}\big(N_T N + N \log(N)\big)$. A direct implicit implementation of (44) would require $\mathcal{O}(\mathfrak{k}k_T N^{\gamma_1})$ operations, for $1 < \gamma_1 < 2$, since linear systems must be solved at each iteration, leading to a larger complexity. Once $(-\triangle)^{\alpha}u$ is approximated for each $\alpha \in \mathfrak{K}$, then N_T linear systems must be solved, requiring $\mathcal{O}(N_T N^{\gamma_2})$ operations, with $1 < \gamma_2 < 2$. □

3.4 PML implementation

The inclusion of a PML in the computational equation yields additional difficulties. Let us consider e.g. the following fractional Schrödinger equation

$$\begin{cases} \mathtt{i}\partial_t u(t,x,y) + \displaystyle\sum_{\alpha\in\mathfrak{K}} v_\alpha(x,y)(-\triangle)^{\alpha} u(t,x,y) = 0, \quad (t,x,y) \in [0,T] \times \mathbb{R}^2, \\ u(t=0,x,y) = u_0(x,y), \quad (x,y) \in \mathbb{R}^2\,, \end{cases} \tag{53}$$

where $(x,y) \in \mathbb{R}^2$ and $t > 0$, $\{v_\alpha\}_{\alpha \in \mathfrak{K}}$ are smooth real-valued functions. In addition, $\mathfrak{K}$ is a finite set of strictly positive real numbers. According to Section 2, the corresponding equation with PML reads

$$\begin{cases} \mathtt{i}\partial_t u(t,x,y) + \displaystyle\sum_{\alpha \in \mathfrak{K}} v_\alpha(x,y)(-\triangle_{\mathrm{PML}})^\alpha u(t,x,y) = 0, \qquad (t,x,y) \in [0,T] \times \mathcal{D}, \\ u(t=0,x,y) = u_0(x,y), \quad (x,y) \in \mathcal{D}\,, \\ u(t,\pm L_x,y) = u(t,x,\pm L_y) = 0, \quad t \in [0,T]\,, \end{cases}$$

with

$$\triangle_{\mathrm{PML}} := \frac{1}{S_x^2}\partial_x^2 + \frac{1}{S_x}\partial_x(\frac{1}{S_x})\partial_x + \frac{1}{S_y^2}\partial_y^2 + \frac{1}{S_y}\partial_y(\frac{1}{S_y})\partial_y\,. \tag{54}$$

A direct implicit discretization is then

$$\boldsymbol{u}_h^{n+1} + \Delta t \sum_{\alpha \in \mathfrak{K}} \boldsymbol{v}_{\alpha,h} \otimes \boldsymbol{A}_h^\alpha \boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n,$$

where $\boldsymbol{A}_h$ is a 3-point finite-difference approximation of $-\triangle_{\mathrm{PML}}$. The main issue is to efficiently compute $\boldsymbol{A}_h^\alpha$, which will be addressed later on.

Let us first present a simple 1-d case considering

$$\mathtt{i}\partial_t u + (-\triangle)^{3/4} u/10 = 0\,, (t,x) \in [0,T] \times \mathbb{R}\,.$$

The computational domain is $\mathcal{D} = [-L,L]$, with $L = 10$. We then solve

$$\begin{cases} \mathtt{i}\partial_t u + (-\triangle_{\mathrm{PML}})^{3/4} u/10 = 0\,, (t,x) \in [0,T] \times \mathcal{D}, \\ u(t,\pm L) = 0. \end{cases}$$

We fix $\Delta x = 5 \times 10^{-2}$ ($N = 401$), $\Delta t = 10^{-1}$ and $N_T = 300$. The operator $\triangle_{\mathrm{PML}}$ is approximated by a three-point stencil finite-difference scheme. The initial data is

$$u_0(x) = 10 \exp\big(-(x-5)^2/2 + \mathtt{i}k_0 x\big)\,,$$

with $k_0 = 7$. The PML absorbing function is chosen as follows: $\sigma(x) = -\sigma_0(x+\delta)$, with $\sigma_0 = 0.5$, $\delta = 1$ and $\theta = 2\pi/7$. We compare in Fig. 7 the amplitude of the PML solution (in logscale) with a solution of reference computed on a larger domain, as well as the solution with homogeneous Dirichlet boundary conditions. We observe the effectiveness of the PML as absorbing layer. Let us remark that the question of optimizing the PML is again not addressed here, which may improve the absorption quality.

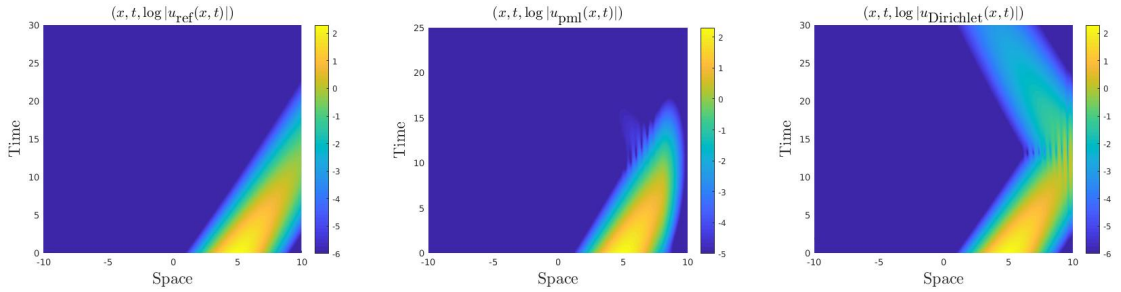

Fig. 7 Reference solution (Left), PML solution (Middle) and solution with Dirichlet boundary conditions (Right) $\{(x,t,\log|u(x,t)|), (x,t) \in \mathcal{D} \times [0,T]\}$.

As mentioned above, the efficient computation of $\boldsymbol{A}_h^\alpha$ is challenging. Unlike the Dirichlet boundary conditions case, the matrix $\boldsymbol{A}_h$ associated with PML is not as simple. Based on Padé's approximation, we propose a decomposition (for simplicity, we consider $\mathfrak{K} = \{\alpha\}$) as follows

$$\begin{cases} \mathtt{i}\partial_t u + v(x,y)(-\triangle)^\alpha u = 0, & (x,y) \in \mathcal{D}_{\mathrm{Phy}}\,, \\ \mathtt{i}\partial_t u + v(x,y)\big(\sum_{k=0}^M a_k^{(M)} u + \sum_{k=1}^M a_k^{(M)} d_k^{(M)} \varphi_k\big) = 0, & (x,y) \in \mathcal{D}_{\mathrm{PML}}\,, \\ \Big(\dfrac{1}{S_x^2}\partial_x^2 - \dfrac{S_x'}{S_x^3}\partial_x + \dfrac{1}{S_y^2}\partial_y^2 - \dfrac{S_y'}{S_y^3}\partial_y - d_k^{(M)}\Big)\varphi_k = u, & (x,y) \in \mathcal{D}_{\mathrm{PML}}\,, \\ u(t,\pm L_x,y) = 0, \qquad u(t,x,\pm L_y) = 0, & t \in [0,T]\,, \\ u(t=0,x,y) = u_0(x,y), & (x,y) \in \mathcal{D}. \end{cases}$$

Let us recall that the Padé's coefficients are α-dependent [31]. In $\mathcal{D}_{\rm Phy}$, we apply the method developed in Subsection 3.2. Alternatively, the approximation of $(-\triangle_{\rm PML})^\alpha$ can be performed by using a Cauchy integral representation. The latter may be interesting as the PML is much smaller and the corresponding Cauchy integral cheaper than applied to the whole domain $\mathcal{D}$. Denoting by $\boldsymbol{A}_{\rm PML}$ a finite-difference approximation of $-\triangle_{\rm PML}$, one would get

$$\boldsymbol{A}_{\rm PML}^\alpha = (2\pi\mathtt{i})^{-1}\boldsymbol{A}_{\rm PML}\int_{\Gamma_{A_{\rm PML}}} z^{\alpha-1}(zI-\boldsymbol{A}_{\rm PML})^{-1}dz\,. \tag{55}$$

We refer to B for a simple and efficient preconditioning of the Cauchy integral computation.

Let us detail the numerical scheme in 1-d, assuming that $\{v_\alpha\}_{\alpha\in\mathfrak{R}}$ are constant

$$\boldsymbol{u}_h^{n+1} - \mathtt{i}\Delta t\sum_{\alpha\in\mathfrak{R}} v_{\alpha,h}\boldsymbol{C}_h^\alpha\boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n + \mathtt{i}\Delta t\boldsymbol{F}_h^n\,.$$

We denote by N the total number of grid points, $N_{\rm PML^\pm}$ the number of grid points in $\mathcal{D}_{\rm PML^\pm}$ (left and right PML regions) with $\mathcal{D}_{\rm PML} := \mathcal{D}_{\rm PML^+}\cup\mathcal{D}_{\rm PML^-}$. We also introduce $N_{\rm Phy}$ as the number of interior grid points ($N = N_{\rm Phy}+N_{\rm PML^-}+N_{\rm PML^+}$), setting

$$\boldsymbol{C}_h^\alpha = \begin{pmatrix} \sum_{k=0}^M a_k^{(M)}\mathbf{I}_{\rm PML^-} & \mathbf{0}_{\rm Phy^-}^T & \mathbf{0}_{\rm PML}^T \\ \mathbf{0}_{\rm Phy^-} & \widetilde{\boldsymbol{A}}_{\rm Phy}^\alpha & \mathbf{0}_{\rm Phy^+}^T \\ \mathbf{0}_{\rm PML} & \mathbf{0}_{\rm Phy^+} & \sum_{k=0}^M a_k^{(M)}\mathbf{I}_{\rm PML^+} \end{pmatrix} \in \mathbb{R}^{N\times N},$$

where $\widetilde{\boldsymbol{A}}_{\rm Phy}\in\mathbb{R}^{N_{\rm Phy}\times N_{\rm Phy}}$ is a three-point stencil discretization of the Laplace operator $-\triangle$ restricted to $\mathcal{D}_{\rm Phy}$. The matrices $\mathbf{I}_{\rm PML^\pm}\in\mathbb{R}^{N_{\rm PML^\pm}\times N_{\rm PML^\pm}}$ are the identity matrices corresponding to the points on $\mathcal{D}_{\rm PML^\pm}$, $\mathbf{0}_{\rm PML}\in\mathbb{R}^{N_{\rm PML^+}\times N_{\rm PML^-}}$ and $\mathbf{0}_{\rm Phy^\pm}\in\mathbb{R}^{N_{\rm Phy^\pm}\times N_{\rm PML^\pm}}$ designate the zero matrices. Moreover $\boldsymbol{F}_h^n$ is a vector with zero components in $\mathcal{D}_{\rm phy}$ and approximates $\sum_{k=1}^M a_k^M d_k^{(M)}\varphi_k^n$ in $\mathcal{D}_{\rm PML}$, where

$$\Big(\frac{1}{S_x^2}\partial_x^2 - \frac{S_x'}{S_x^3}\partial_x - d_k^{(M)}\Big)\varphi_k^n = u^n, \qquad (x,y)\in\mathcal{D}_{\rm PML}\,.$$

The corresponding scheme can be rewritten in the form

$$\Big(\mathbf{I}_h - \mathtt{i}\Delta t\textstyle\sum_{\alpha\in\mathfrak{R}} v_\alpha\boldsymbol{P}_h\boldsymbol{\Lambda}_h^{(\alpha,k_T)}\boldsymbol{P}_h^{-1}\Big)\boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n + \mathtt{i}\Delta t\boldsymbol{F}_h^n\,, \tag{56}$$

where we have now

$$\boldsymbol{\Lambda}_h^{(\alpha,k)} = \begin{pmatrix} \sum_{l=0}^M a_l^{(M)}\mathbf{I}_{\rm PML^-} & \mathbf{0}_{\rm PML^-}^T & \mathbf{0}_{\rm PML}^T \\ \mathbf{0}_{\rm Phy^-} & \widetilde{\boldsymbol{\Lambda}}_{\rm Phy}^{(\alpha,k)} & \mathbf{0}_{\rm Phy^+}^T \\ \mathbf{0}_{\rm PML} & \mathbf{0}_{\rm Phy^+} & \sum_{l=0}^M a_l^{(M)}\mathbf{I}_{\rm PML^+} \end{pmatrix} \in \mathbb{R}^{N\times N},$$

$\widetilde{\boldsymbol{\Lambda}}_h^{(\alpha,k_T)}$ is a diagonal matrix and, for $1\leqslant l_x\leqslant N_{\rm Phy}$, $\zeta_h^{(l_x;k_T)} = \big(\Pi_{p=0}^{k_T-1}\nu_h^{(l_x;p)}\big)^{-1}$. The matrix $\boldsymbol{P}_h$ is defined as

$$\boldsymbol{P}_h = \begin{pmatrix} \mathbf{I}_{\rm PML^-} & \mathbf{0}_{\rm PML^-}^T & \mathbf{0}_{\rm PML}^T \\ \mathbf{0}_{\rm Phy^-} & \boldsymbol{P}_{\rm Phy} & \mathbf{0}_{\rm Phy} \\ \mathbf{0}_{\rm PML} & \mathbf{0}_{\rm Phy^+} & \mathbf{I}_{\rm PML^+} \end{pmatrix} \in \mathbb{R}^{N\times N},$$

where

$$\boldsymbol{A}_h^\alpha = \boldsymbol{P}_h\boldsymbol{\Lambda}_h^{(\alpha,k_T)}\boldsymbol{P}_h^{-1}, \qquad \widetilde{\boldsymbol{A}}_{\rm Phy}^\alpha = \widetilde{\boldsymbol{P}}_{\rm Phy}\widetilde{\boldsymbol{\Lambda}}_{\rm Phy}^{(\alpha,k_T)}\widetilde{\boldsymbol{P}}_{\rm Phy}^{-1}\,.$$

For real $\{v_\alpha\}_\alpha$, the scheme (56) is trivially ℓ^2-stable. Notice that we even have an explicit knowledge of the eigenvalues of $\boldsymbol{A}_h^\alpha$ thanks to those of $\widetilde{\boldsymbol{A}}_{\rm Phy}^\alpha$. The latter also trivially provides stability conditions for complex sequence $\{v_\alpha\}_\alpha$. In this new following illustrative example, we consider the initial problem

$$\mathtt{i}\partial_t u + (-\triangle)^{1/2}u/10 = 0\,,(t,x)\in[0,T]\times\mathbb{R}\,.$$

The bounded computational domain is $\mathcal{D} = [-10, 10]$ and we solve the following system

$$\begin{cases} \mathtt{i}\partial_t u + (-\triangle)^{1/2} u/10 & = 0, \ (t,x) \in [0,T] \times \mathcal{D}_{\text{Phy}}, \\ \mathtt{i}\partial_t u + \big(\sum_{k=0}^{M} a_k^{(M)} u/10 + \sum_{k=1}^{M} a_k^{(M)} d_k^{(M)} \varphi_k\big) & = 0, \qquad x \in \mathcal{D}_{\text{PML}}\,, \\ \Big(\dfrac{1}{S_x^2}\partial_x^2 - \dfrac{S_x'}{S_x^3}\partial_x - d_k^{(M)}\Big)\varphi_k & = u/10, \quad x \in \mathcal{D}_{\text{PML}}\,, \\ u(t, \pm L) & = 0, \qquad t \in [0,T]\,. \end{cases}$$

We fix $\Delta x = 10^{-1}$ ($N = 101$), $\Delta t = 10^{-1}$ and $N_T = 1100$ and $M = 2$. The coefficients $\{a_k^{(M)}\}_k$ and $\{d_k^{(M)}\}_k$ are defined in (30). The initial data is

$$u_0(x) = \exp\big(-(x+5)^2/2 + \mathtt{i}k_0 x\big)/\mathcal{N}\,,$$

with $k_0 = -8$ and $\mathcal{N}$ is the normalization constant for the L^2-norm. The PML absorbing function is such that $\sigma(x) = -\sigma_0(x+\delta)^2$, with the parameters $\sigma_0 = 0.5$, $\delta = 0.5$ and $\theta = 2\pi/11$. We compare in Fig. 8 the amplitude of the PML-solution (in logscale) with a solution of reference computed on a larger domain $[-20, 20]$ as well as the solution with zero Dirichlet boundary conditions. We observe only some small scale reflections back into the computational domain which confirm the behavior of the absorbing layers. We also report in Fig. 9, the solutions in semilogscale at final time $T = 30$ to show the absorbing property of the PML.

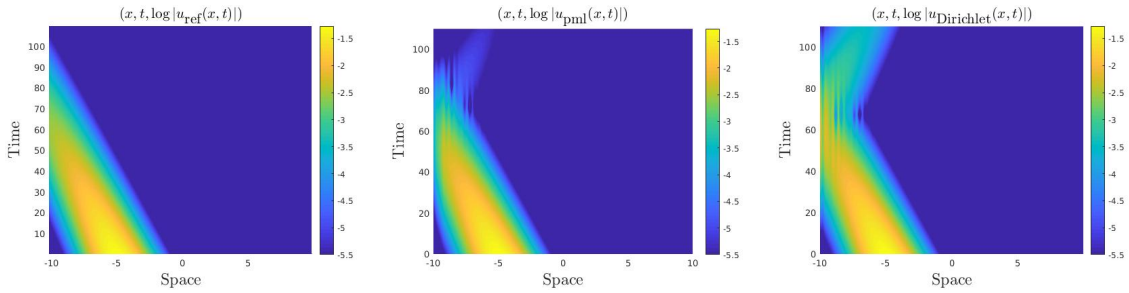

Fig. 8 Solution of reference (Left), PML (Middle) and with Dirichlet boundary conditions (Right) $\{(x,t,\log|u(x,t)|), (x,t) \in \mathcal{D} \times [0,T]\}$.

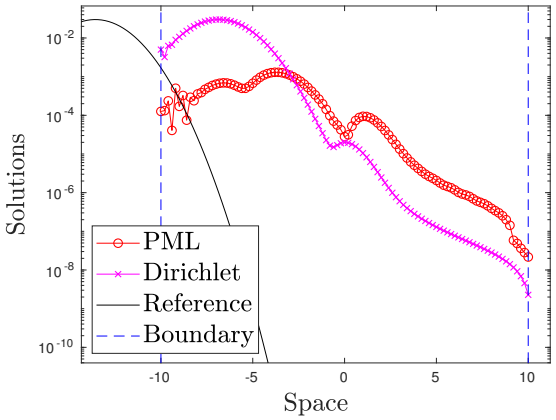

Fig. 9 Solution in logscale computed with Dirichlet boundary conditions, PML and reference solutions $\{(x,T,\log|u(x,T)|), x \in \mathcal{D}\}$.

In the two-dimensional case, the derivation of the numerical scheme is slightly more technical but is based on similar ideas. Let us assume that $N := N_x = N_y$. In the case of Dirichlet boundary conditions at

$\{(\pm L_x, y) : y \in [-L_y, L_y]\} \cup \{(x, \pm L_y) : x \in [-L_x, L_x]\}$, the standard 2-d second-order finite-difference approximate Laplacian is denoted by

$$\mathbb{C}_h = \begin{pmatrix} \mathbb{A}_{\rm Phy} & \boldsymbol{E}_{\rm Phy} & \mathbf{0}_{\rm Phy} & \cdots & \cdots & \mathbf{0}_{\rm Phy} \\ \boldsymbol{E}_{\rm Phy} & \mathbb{A}_{\rm Phy} & \boldsymbol{E}_{\rm Phy} & \mathbf{0}_{\rm Phy} & \cdots & \mathbf{0}_{\rm Phy} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0}_{\rm Phy} & \cdots & \mathbf{0}_{\rm Phy} & \boldsymbol{E}_{\rm Phy} & \mathbb{A}_{\rm Phy} & \boldsymbol{E}_{\rm Phy} \\ \mathbf{0}_{\rm Phy} & \cdots & \cdots & \mathbf{0}_{\rm Phy} & \boldsymbol{E}_{\rm Phy} & \mathbb{A}_{\rm Phy} \end{pmatrix} \in \mathbb{R}^{N^2 \times N^2},$$

where $\mathbb{A}_{\rm Phy} \in \mathbb{R}^{N \times N}$ and $\boldsymbol{E}_{\rm Phy} \in \mathbb{R}^{N \times N}$. When including the PML on a $N \times N$ grid, we decompose the PML region into South, North, West, East regions as

$$\mathcal{D}_{\rm PML} = \mathcal{D}_{{\rm PML}^{(S)}} \cup \mathcal{D}_{{\rm PML}^{(N)}} \cup \mathcal{D}_{{\rm PML}^{(E)}} \cup \mathcal{D}_{{\rm PML}^{(W)}}.$$

We denote by $N_{{\rm PML}^{(S,N,W,E)}}$ the number of grid points is each direction South, North, West, East, respectively. Formally, the scheme still reads

$$\boldsymbol{u}_h^{n+1} + \Delta t \sum_{\alpha \in \mathfrak{R}} v_{\alpha,h} \boldsymbol{C}_h^{\alpha} \boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n - \mathtt{i} \Delta t \boldsymbol{F}_h^n, \tag{57}$$

where $\boldsymbol{F}_h^n$ is a vector, which is zero on $\mathcal{D}_{\rm phy}$ and provides an approximation to $\sum_{k=1}^M d_k^{(M)} \varphi_k^n$ on $\mathcal{D}_{\rm PML}$, where

$$\Big(\frac{1}{S_x^2} \partial_x^2 - \frac{S_x'}{S_x^3} \partial_x + \frac{1}{S_y^2} \partial_y^2 - \frac{S_y'}{S_y^3} \partial_y - d_k^{(M)} \Big) \varphi_k^n = u^n, \qquad (x,y) \in \mathcal{D}_{\rm PML}.$$

We prove now a proposition which is useful in practice, regarding the "diagonalization of the scheme", leading to an efficient implementation (see Proposition 9).

Proposition 10 *The scheme* (57) *can be analytically rewritten in the form:*

$$\Big(\mathbf{I}_h + \Delta t \sum_{\alpha \in \mathfrak{R}} v_\alpha \boldsymbol{P}_h \boldsymbol{\Lambda}_h^{(\alpha,k_T)} \boldsymbol{P}_h^{-1} \Big) \boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n - \mathtt{i} \Delta t \boldsymbol{F}_h^n, \tag{58}$$

where $\boldsymbol{\Lambda}_h^{(\alpha,k_T)}$ *is a diagonal matrix with* $N^2 - N_{Phy}^2$ *eigenvalues* $\sum_{k=0}^M a_k^{(M)}$*, and, for* $1 \leqslant l_x \leqslant N_{Phy}$*,* $1 \leqslant l_y \leqslant N_{Phy}$*, such that*

$$\zeta_h^{(l_x,l_y;k_T)} = \big(\Pi_{p=1}^{k_T} \nu_h^{(l_x,l_y;p)} \big)^{-1}.$$

Moreover, this scheme is unconditionally ℓ^2*-stable.*

Proof. Let us first notice that the matrix $\boldsymbol{C}_h^{\alpha} \in \mathbb{R}^{N^2 \times N^2}$ is block diagonal

$$\boldsymbol{C}_h^{\alpha} = \begin{pmatrix} \sum_{k=0}^M a_k^{(M)} \mathbf{I}_{{\rm PML}^{(S)}} & \mathbf{0}_{{\rm PML}^{(S)}}^T & \cdots & \cdots & \cdots & \mathbf{0}_{{\rm PML}^{(S)}}^T & \mathbf{0}_{{\rm PML}^{(SE)}} \\ \mathbf{0}_{{\rm PML}^{(W)}} & \boldsymbol{A}_{\rm Phy}^{\alpha} & \boldsymbol{E}_{\rm Phy} & \cdots & \cdots & \cdots & \mathbf{0}_{{\rm PML}^{(E)}}^T \\ \mathbf{0}_{{\rm PML}^{(W)}} & \boldsymbol{E}_{\rm Phy} & \boldsymbol{A}_{\rm Phy}^{\alpha} & \boldsymbol{E}_{\rm Phy} & \cdots & \cdots & \mathbf{0}_{{\rm PML}^{(E)}}^T \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0}_{{\rm PML}^{(W)}} & \cdots & \cdots & \boldsymbol{E}_{\rm Phy} & \boldsymbol{A}_{\rm Phy}^{\alpha} & \boldsymbol{E}_{\rm Phy} & \mathbf{0}_{{\rm PML}^{(E)}}^T \\ \mathbf{0}_{{\rm PML}^{(W)}} & \cdots & \cdots & \cdots & \boldsymbol{E}_{\rm Phy} & \boldsymbol{A}_{\rm Phy}^{\alpha} & \mathbf{0}_{{\rm PML}^{(E)}}^T \\ \mathbf{0}_{{\rm PML}^{(N)}} & \mathbf{0}_{{\rm PML}^{(N)}} & \cdots & \cdots & \cdots & \mathbf{0}_{{\rm PML}^{(N)}} & \sum_{k=0}^M a_k^{(M)} \mathbf{I}_{{\rm PML}^{(N)}} \end{pmatrix},$$

where $\boldsymbol{I}_{{\rm PML}^{(S)}} \in \mathbb{R}^{NN_{{\rm PML}^{(S)}} \times NN_{{\rm PML}^{(W)}}}$, $\boldsymbol{I}_{{\rm PML}^{(N)}} \in \mathbb{R}^{NN_{{\rm PML}^{(N)}} \times NN_{{\rm PML}^{(N)}}}$ and $\mathbf{0}_{{\rm PML}^{(S,W,E)}} \in \mathbb{R}^{N \times NN_{{\rm PML}^{(S,W,E)}}}$. The matrix $\boldsymbol{A}_h^{\alpha}$ reads

$$\boldsymbol{A}_h^{\alpha} = \begin{pmatrix} \sum_{k=0}^M a_k^{(M)} \mathbf{I}_{{\rm PML}^{(WW)}} & \mathbf{0}_{{\rm Phy}^{(W)}}^T & \mathbf{0}_{{\rm PML}^{(EW)}}^T \\ \mathbf{0}_{{\rm Phy}^{(W)}} & \widetilde{\mathbb{A}}_{\rm Phy}^{\alpha} & \mathbf{0}_{{\rm Phy}^{(E)}}^T \\ \mathbf{0}_{{\rm PML}^{(EW)}} & \mathbf{0}_{{\rm Phy}^{(E)}} & \sum_{k=0}^M a_k^{(M)} \mathbf{I}_{{\rm PML}^{(EE)}} \end{pmatrix} \in \mathbb{R}^{N^2 \times N^2},$$

where $\widetilde{\mathbb{A}}_h \in \mathbb{R}^{N_{\rm Phy}\times N_{\rm Phy}}$ is the matrix $\mathbb{A}_h \in \mathbb{R}^{N^2\times N^2}$ restricted to $\mathcal{D}_{\rm Phy}$, and $\mathbf{I}_{{\rm PML}^{(WW)}}$ is the identity matrix in $\mathbb{R}^{N_{{\rm PML}^{(W)}}\times N_{{\rm PML}^{(W)}}}$, $\mathbf{I}_{{\rm PML}^{(EE)}}$ is the identity matrix in $\mathbb{R}^{N_{{\rm PML}^{(E)}}\times N_{{\rm PML}^{(E)}}}$, and $\mathbf{0}_{{\rm Phy}^{(W)}}$ is the null matrix in $\mathbb{R}^{N_{\rm Phy}\times N_{{\rm PML}^{(W)}}}$, $\mathbf{0}_{{\rm Phy}^{(E)}}$ is the null matrix in $\mathbb{R}^{N_{{\rm PML}^{(E)}}\times N_{\rm Phy}}$. Finally, we obtain

$$\Big(\mathbf{I}_h + \Delta t \sum_{\alpha\in\mathfrak{K}} v_\alpha \boldsymbol{P}_h \boldsymbol{\Lambda}_h^{(\alpha,k_T)} \boldsymbol{P}_h^{-1}\Big)\boldsymbol{u}_h^{n+1} = \boldsymbol{u}_h^n - \mathtt{i}\Delta t \boldsymbol{F}_h^n\,, \tag{59}$$

where now

$$\boldsymbol{\Lambda}_h^{(\alpha,k)} = \begin{pmatrix} \sum_{k=0}^M a_k^{(M)}\mathbf{I}_{{\rm PML}^{(S)}} & \mathbf{0}_{{\rm PML}^{(S)}} & \mathbf{0}_{{\rm PML}^{(S)}} \\ \mathbf{0}_{\rm Phy} & \widetilde{\boldsymbol{\Lambda}}_{\rm Phy}^{(\alpha,k)} & \mathbf{0}_{\rm Phy} \\ \mathbf{0}_{{\rm PML}^{(N)}} & \mathbf{0}_{{\rm PML}^{(N)}} & \sum_{k=0}^M a_k^{(M)}\mathbf{I}_{{\rm PML}^{(N)}} \end{pmatrix} \in \mathbb{R}^{N^2\times N^2}\,,$$

and $\widetilde{\boldsymbol{\Lambda}}_h^{(\alpha,k_T)}$ is a diagonal matrix with $N^2 - N_{\rm Phy}^2$ eigenvalues $\sum_{k=0}^M a_k^{(M)}$ (corresponding to the regions $\mathcal{D}_{{\rm PML}^{(W,E)}}$), such that, for $1\leqslant l_x\leqslant N_{\rm Phy}$, $1\leqslant l_y\leqslant N_{\rm Phy}$, we have

$$\zeta_h^{(l_x,l_y;k_T)} = \big(\Pi_{p=1}^{k_T}\nu_h^{(l_x,l_y;p)}\big)^{-1}.$$

The matrix $\boldsymbol{P}_h$ is defined as

$$\boldsymbol{P}_h = \begin{pmatrix} \mathbf{I}_{{\rm PML}^{(S)}} & \widetilde{\mathbf{0}}_{\rm PML}^T & \mathbf{0}_{{\rm PML}^{(NS)}}^T \\ \widetilde{\mathbf{0}}_{\rm PML} & \widetilde{\boldsymbol{P}}_h & \mathbf{0}_{{\rm PML}^{(N)}}^T \\ \mathbf{0}_{{\rm PML}^{(NS)}} & \mathbf{0}_{{\rm PML}^{(N)}} & \mathbf{I}_{{\rm PML}^{(N)}}\,, \end{pmatrix} \in \mathbb{R}^{N^2\times N^2}$$

and is such that

$$\boldsymbol{A}_h^\alpha = \boldsymbol{P}_h\boldsymbol{\Lambda}_h^{(\alpha,k_T)}\boldsymbol{P}_h^{-1}, \qquad \widetilde{\boldsymbol{A}}_h^\alpha = \widetilde{\boldsymbol{P}}_h\widetilde{\boldsymbol{\Lambda}}_h^{(\alpha,k_T)}\widetilde{\boldsymbol{P}}_h^{-1}.$$

In the above equations, we set $\widetilde{\mathbf{0}}_{{\rm PML}^{(S)}}^T$ as the null matrix in $\mathbb{R}^{N^2\times NN_{{\rm PML}^{(S)}}}$, $\widetilde{\mathbf{0}}_{{\rm PML}^{(N)}}^T$ is the zero matrix in $\mathbb{R}^{NN_{{\rm PML}^{(N)}}\times N^2}$, and $\widetilde{\mathbf{0}}_{{\rm PML}^{(NS)}}^T$ is the null matrix in $\mathbb{R}^{NN_{{\rm PM}^{(N)}}\times NN_{{\rm PML}^{(S)}}}$. All the eigenvalues of $\boldsymbol{\Lambda}_h^{(\alpha,k)}$ are positive, implying that the implicit scheme is ℓ^2-stable for any Δt, concluding hence the proof. □

As in the one-dimensional case, the ℓ^2-stability analysis can easily be established from (59) for complex sequences $\{v_\alpha\}_\alpha$ thanks to the explicit knowledge to the eigenvalues of $\boldsymbol{A}_h^\alpha$, while the real case is trivial.

To illustrate the approach, we fix the following initial boundary-value problem (for $\alpha = 1/2$), with homogeneous Dirichlet boundary conditions

$$\begin{cases} \mathtt{i}\partial_t u(t,x,y) + (-\triangle_{\rm PML})^{1/2}u(t,x,y)/10 = 0, & (t,x,y)\in[0,T]\times\mathcal{D}, \\ u(t=0,x,y) = e^{-((x-13/4)^2+y^2)/2+5\mathtt{i}x}, & (x,y)\in\mathcal{D}, \\ u(t,\pm L_x,y) = 0,\ t\in[0,T], \\ u(t,x,\pm L_y) = 0,\ t\in[0,T]\,. \end{cases} \tag{60}$$

The computational domain is $\mathcal{D} = [-4.8, 4.8]^2$ and the final time is $T = 4$. Following our strategy, the modified fractional linear equation with PML consists in approximating $(-\triangle)^{1/2}$ as in (17). In this example, we take $S_{x,y}(\cdot) = 1 + 10^{-2}e^{\mathtt{i}\theta}\widetilde{\sigma}(\cdot)$ in (16), even if more optimized profiles could be used. We apply a Crank-Nicolson scheme in time to solve the corresponding IBVP (60). The real space grid involves $N_x \times N_y = 51^2$ points and the time discretization is $\Delta t = 5\times 10^{-2}$. We report in Fig. 10 the amplitude of the initial data, of the solution with homogeneous Dirichlet boundary conditions (without PML), the reference solution (computed on a larger domain) and the PML-based solution, excluding the PML region. In this case, we consider a quadratic absorbing profile (Type II) $\sigma(\nu) = \sigma_0(\nu+\delta)^2$ [4], setting $\sigma_0 = 5\times 10^{-3}$, $\theta = 2\pi/7$, and $\delta_\nu = 0.15L_\nu$. From the simulation, we can see that only some small reflections occur in the physical domain.

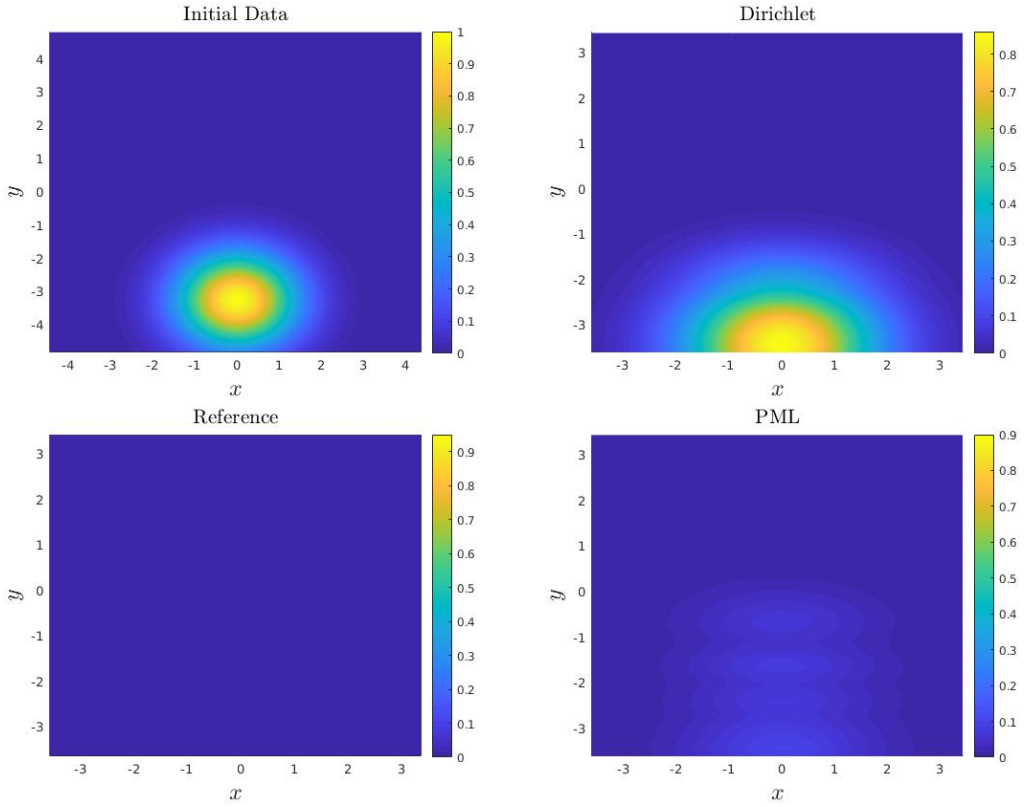

Fig. 10 (Top-left) Amplitude of the initial data, (Top-right) periodic solution, (Bottom-left) reference solution and (Bottom-right) of the PML solution (with Type II profile).

4 Conclusion

In this paper, we introduced and explored several computational methods for solving fractional Laplacian equations, including fractional Schrödinger equations. Two main methodologies were proposed. The first one is a pseudospectral method based on discrete Fourier transforms with periodic boundary conditions coupled with PMLs. The latter were approximated by using Padé's approximants as well as finite-difference methods in the absorbing layers. Alternatively, a finite-difference methodology combined with i) a differential equation solver for computing power of matrices, as well as ii) PMLs was then proposed for solving fractional equations with homogeneous Dirichlet boundary conditions. A basic mathematical analysis has been developed and numerical experiments illustrating the accuracy and efficiency of these methods were reported. In future works, we plan to implement these methods in high-dimension and to address realistic physical problems. Finally, the optimization of the PML parameters for fractional PDEs needs to be further investigated.

Acknowledgments. X. Antoine was supported by the ANR project NABUCO, ANR-17-CE40-0025, and the LIAFSMA (Université de Lorraine). X. Antoine acknowledges the support from the Inria associate team BEC2HPC (Bose-Einstein Condensates: Computation and HPC simulation). E. Lorin thanks NSERC through the Discovery Grant program. This work was partially done while the authors were visiting the Institute for Mathematical Sciences in 2019, National University of Singapore.

References

1. X. Antoine, C. Besse, and P. Klein. Absorbing boundary conditions for the two-dimensional Schrödinger equation with an exterior potential. Part I: Construction and *a priori* estimates. Math. Models Methods Appl. Sci., 22(10):1250026, 38, 2012.
2. X. Antoine, C. Besse, and V. Rispoli. High-order IMEX-spectral schemes for computing the dynamics of systems of nonlinear Schrödinger/Gross–Pitaevskii equations. J. Comput. Phys., 327:252–269, 2016.

3. 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. Comput. Phys., 258:509–523, 2014.
4. X. Antoine, C. Geuzaine, and Q. Tang. Coupling spectral methods and perfectly matched layer for simulating the dynamics of nonlinear Schrödinger equations. Application to rotating Bose-Einstein condensates. Submitted, 2019.
5. X. Antoine and E. Lorin. Computational performance of simple and efficient sequential and parallel Dirac equation solvers. Comput. Phys. Commun., 220:150–172, 2017.
6. X. Antoine and E. Lorin. Double-preconditioning for fractional linear systems. Application to fractional Poisson equations. Submitted, 2019.
7. X. Antoine and E. Lorin. ODE-based double-preconditioning for solving linear systems $A^\alpha x = b$ and $f(A)x = b$. Submitted, 2019.
8. X. Antoine and E. Lorin. Towards perfectly matched layers for time-dependent space fractional PDEs. J. Comput. Phys., 391:59–90, 2019.
9. X. Antoine, E. Lorin, and Q. Tang. A friendly review of absorbing boundary conditions and perfectly matched layers for classical and relativistic quantum waves equations. Molecular Physics, 115(15-16):1861–1879, 2017.
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. C. Bardos and E. Tadmor. Stability and spectral convergence of Fourier method for nonlinear problems: on the shortcomings of the 2/3 de-aliasing method. Numer. Math., 129(4):749–782, 2015.
12. J.-P. Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. J. Comput. Phys., 114(2):185–200, 1994.
13. J.-P. Bérenger. Three-dimensional perfectly matched layer for the absorption of electromagnetic waves. J. Comput. Phys., 127(2):363–379, 1996.
14. A. Bermúdez, L. Hervella-Nieto, A. Prieto, and R. Rodríguez. An optimal perfectly matched layer with unbounded absorbing function for time-harmonic acoustic scattering problems. J. Comput. Phys., 223(2):469–488, 2007.
15. A. Bermúdez, L. Hervella-Nieto, A. Prieto, and R. Rodríguez. An exact bounded perfectly matched layer for time-harmonic scattering problems. SIAM J. Sci. Comput., 30(1):312–338, 2007/08.
16. M. Chandru, P. Das, and H. Ramos. Numerical treatment of two-parameter singularly perturbed parabolic convection diffusion problems with non-smooth data. Mathematical Methods in the Applied Sciences, 41(14):5359–5387, 2018.
17. A.V. Chechkin, R. Gorenflo, and I.M. Sokolov. Retarding subdiffusion and accelerating superdiffusion governed by distributed-order fractional diffusion equations. Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics, 66(4):7, 2002.
18. A.V. Chechkin, R. Gorenflo, and I.M. Sokolov. Fractional diffusion in inhomogeneous media. Journal of Physics A: Mathematical and General, 38(42):L679–L684, 2005.
19. F. Collino and P. Monk. The perfectly matched layer in curvilinear coordinates. SIAM J. Sci. Comput., 19(6):2061–2090, 1998.
20. T. Colonius. Modeling artificial boundary conditions for compressible flow. In Annual review of fluid mechanics., volume 36 of Annu. Rev. Fluid Mech., pages 315–345. 2004.
21. P. Das. A higher order difference method for singularly perturbed parabolic partial differential equations. J. Difference Equ. Appl., 24(3):452–477, 2018.
22. P. Das and V. Mehrmann. Numerical solution of singularly perturbed convection-diffusion-reaction problems with two small parameters. BIT, 56(1):51–76, 2016.
23. P. Das, S. Rana, and H. Ramos. A perturbation-based approach for solving fractional-order Volterra-Fredholm integrodifferential equations and its convergence analysis. International Journal of Computer Mathematics, 2019.
24. P. Das, S. Rana, and J. Vigo-Aguiar. Higher order accurate approximations on equidistributed meshes for boundary layer originated mixed type reaction diffusion systems with multiple scale nature. Appl. Numer. Math., 148:79–97, 2020.
25. P. I. Davies and N. J. Higham. Computing $f(A)b$ for matrix functions f. In QCD and numerical analysis III, volume 47 of Lect. Notes Comput. Sci. Eng., pages 15–24. Springer, Berlin, 2005.
26. E. Di Nezza, G. Palatucci, and E. Valdinoci. Hitchhiker's guide to the fractional Sobolev spaces. Bulletin des Sciences Mathématiques, 136(5):521–573, 2012.
27. M. A. Ezzat, A. S. El-Karamany, and A. A. El-Bary. Thermo-viscoelastic materials with fractional relaxation operators. Appl. Math. Model., 39(23-24):7499–7512, 2015.
28. J. Goodman, T. Hou, and E. Tadmor. On the stability of the unsmoothed Fourier method for hyperbolic equations. Numer. Math., 67(1):93–129, 1994.
29. R. Gorenflo, F. Mainardi, D. Moretti, and P. Paradisi. Time fractional diffusion: A discrete random walk approach. Nonlinear Dynamics, 29(1-4):129–143, 2002.
30. N. Hale, N. J. Higham, and L. N. Trefethen. Computing $\mathbf{A}^\alpha$, $\log(\mathbf{A})$, and related matrix functions by contour integrals. SIAM J. Numer. Anal., 46(5):2505–2523, 2008.
31. N.J. Higham. Evaluating Padé approximants of the matrix logarithm. SIAM J. Matrix Anal. Appli., 22(4):1126–1135, 2001.
32. F. Q. Hu. On absorbing boundary conditions for linearized Euler equations by a perfectly matched layer. J. Comput. Phys., 129(1):201–219, 1996.
33. F. Q. Hu. A stable, perfectly matched layer for linearized Euler equations in unsplit physical variables. J. Comput. Phys., 173(2):455–480, 2001.
34. X. Li and C. Xu. Existence and uniqueness of the weak solution of the space-time fractional diffusion equation and a spectral method approximation. Commun. Comput. Phys., 8(5):1016–1051, 2010.
35. A. Lischke, G. Pang, M. Gulian, F. Song, C. Glusa, X. Zheng, Z. Mao, W. Cai, M. Meerschaert, M. Ainsworth, and G. E. Karniadakis. What is the fractional laplacian? a comparative review with new results. Journal of Computational Physics, 404:109009, 2020.
36. Y. Saad and M.H. Schultz. GMRES - A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. SIAM J. Sc. Stat. Comput., 7(3):856–869, 1986.

37. E. Scalas, R. Gorenflo, and F. Mainardi. Fractional calculus and continuous-time finance. Physica A: Statistical Mechanics and its Applications, 284(1):376–384, 2000.
38. J. Shen, T. Tang, and L.-L. Wang. Spectral methods, volume 41 of Springer Series in Computational Mathematics. Springer, Heidelberg, 2011. Algorithms, analysis and applications.
39. M. E. Taylor. Partial differential equations I. Basic theory, volume 115 of Applied Mathematical Sciences. Springer, New York, second edition, 2011.
40. B.E. Treeby, J. Jaros, A.P. Rendell, and B.T. Cox. Modeling nonlinear ultrasound propagation in heterogeneous media with power law absorption using a k-space pseudospectral method. Journal of the Acoustical Society of America, 131(6):4324–4336, 2012.
41. B.E. Treeby, J. Jaros, D. Rohrbach, and B.T. Cox. Modelling elastic wave propagation using the k-wave matlab toolbox. IEEE International Ultrasonics Symposium, IUS, pages 146–149, 2014.
42. S. V. Tsynkov. Numerical solution of problems on unbounded domains. A review. Appl. Numer. Math., 27(4):465–532, 1998. Absorbing boundary conditions.
43. E. Turkel and A. Yefet. Absorbing PML boundary layers for wave-like equations. Appl. Numer. Math., 27(4):533–557, 1998. Absorbing boundary conditions.
44. P. Veeresha, H.M. Baskonus, D.G. Prakasha, W. Gao, and G. Yel. Regarding new numerical solution of fractional schistosomiasis disease arising in biological phenomena. Chaos, Solitons and Fractals, 133, 2020.

A Padé approximant-based PML

In the following, we detail the procedure for deriving PMLs by using Padé's approximants with $\alpha = p/2^k$, $k \in \mathbb{N}^*$ and $p \in \mathbb{N}^*$.

Case $\alpha = 1/2^k$, $k \in \mathbb{N}^*$. The idea developed for $\alpha = 1/2$ can easily be extended to coefficients of the form $1/2^k$. We can iteratively repeat the process described above, by simply using

$$(-\triangle_{\mathrm{PML}})^{1/2^k} = \sqrt{(-\triangle_{\mathrm{PML}})^{1/2^{k-1}}}\,.$$

$$\mathrm{Op}\Big(\sqrt{\sigma\big((-\triangle_{\mathrm{PML}})^{1/2^k}\big)}\Big) \approx \mathrm{Op}\Big(\sum_{k=0}^{M} a_k^{(M)} - \sum_{k=1}^{M} \frac{a_k^{(M)} d_k^{(M)}}{\sigma\big((-\triangle_{\mathrm{PML}})^{1/2^{k-1}}\big) + d_k^{(M)}}\Big)\,.$$

This leads to long calculations, which however have to be done once for all for any given $\alpha = 1/2^k$ for $k \in \mathbb{N}^*$.

Case $\alpha \in \mathbb{N}^*/2^k$, $k \in \mathbb{N}^*$. We extend the above ideas to rational numbers α in the form $p/2^k$, for $p \in \mathbb{N}^*$. In fact, thanks to the above discussion, we simply need to detail the case $\alpha = p/2$, for $p \in \mathbb{N}^*$. Although the expressions look quite complex, in practice simplifications and approximations are possible

$$\begin{cases} \mathtt{i}\partial_t u(t,x,y) + \displaystyle\sum_{\alpha\in\mathfrak{K}} v_\alpha(x,y)(-\triangle)^\alpha u(t,x,y) = 0, \quad (t,x,y) \in [0,T]\times\mathbb{R}^2, \\ u(t=0,x,y) = u_0(x,y), \quad (x,y)\in\mathbb{R}^2. \end{cases} \tag{61}$$

In the above system, we assume that $\{v_\alpha\}_{\alpha\in\mathfrak{K}}$ designates some smooth real- or purely complex-valued functions and that $\mathfrak{K}$ is a finite set of strictly positive numbers in $\mathbb{N}^*/2$. We then consider the corresponding IBVP

$$\begin{cases} \mathtt{i}\partial_t u(t,x,y) + \displaystyle\sum_{\alpha\in\mathfrak{K}} v_\alpha(x,y)(-\triangle_{\mathrm{PML}})^\alpha u(t,x,y) = 0, \quad (t,x,y) \in [0,T]\times\mathcal{D}, \\ u(t=0,x,y) = u_0(x,y), \quad (x,y)\in\mathcal{D}, \\ u(t,\pm L_x,y) = u(t,\mp L_x,y),\ u(t,x,\pm L_y) = u(t,x,\mp L_y),\ t\in[0,T]\,. \end{cases} \tag{62}$$

We can formally rewrite the symbol of $(-\triangle_{\mathrm{PML}})^\alpha$ [1] as

$$\sigma\big((-\triangle_{\mathrm{PML}})^{p/2}\big) = \sigma\big(\sqrt{-\triangle_{\mathrm{PML}}}^{\,p}\big) = \sigma\big(\sqrt{-\triangle_{\mathrm{PML}}}\big)\#\sigma\big(\sqrt{-\triangle_{\mathrm{PML}}}\big)\cdots\#\sigma\big(\sqrt{-\triangle_{\mathrm{PML}}}\big)\,,$$

where we recall that

Proposition 11 *[1] For two pseudodifferential operators A and B with C^∞-coefficients, $\alpha = (\alpha_1,\alpha_2)$ with $|\alpha| = \alpha_1+\alpha_2$ and $\alpha! = \alpha_1!\alpha_2!$, the symbol to the composed operator AB is given by*

$$\sigma(AB) = \sigma(A)\#\sigma(B) \sim \sum_{|\alpha|=0}^{\infty} \frac{(-\mathtt{i})^{|\alpha|}}{\alpha!}\partial_x^{\alpha_1}\partial_y^{\alpha_2}\sigma(A)\partial_{\xi_x}^{\alpha_1}\partial_{\xi_y}^{\alpha_2}\sigma(B)\,.$$

From a practical point of view, the computation of these symbols and the approximation of the corresponding operators can be complex. Instead, we can proceed as follows:

– If $p \in 2\mathbb{N}^*$, and denoting $q = p/2 \in \mathbb{N}^*$, then the corresponding *differential* operator simply reads

$$(-\triangle_{\mathrm{PML}})^{p/2} = \mathrm{Op}\Big(\Big(\frac{1}{S_x^2}|\xi_x|^2 + \mathtt{i}\frac{S_x'}{S_x^3}\xi_x + \frac{1}{S_y^2}|\xi_y|^2 + \mathtt{i}\frac{S_y'}{S_y^3}\xi_y\Big)^q\Big)$$

and can easily be analytically computed and numerically approximated, as a standard differential operator.

– If $p \in 2\mathbb{N}+1$, with $p = 2q+1$ and $q \in \mathbb{N}$, then we rewrite

$$\begin{aligned}\sigma\big((-\triangle_{\mathrm{PML}})^{p/2}\big) &= \sigma\big((-\triangle_{\mathrm{PML}})^{q}\sqrt{-\triangle_{\mathrm{PML}}}\big)\\ &= \sigma\big((-\triangle_{\mathrm{PML}})^{q}\big)\#\sigma\big(\sqrt{-\triangle_{\mathrm{PML}}}\big)\\ &= \sum_{|\beta|=0}^{\infty}\frac{(-\mathtt{i})^{|\beta|}}{\beta!}\partial_x^{\beta_1}\partial_y^{\beta_2}\sigma\big((-\triangle_{\mathrm{PML}})^{q}\big)\partial_{\xi_x}^{\beta_1}\partial_{\xi_y}^{\beta_2}\sigma\big(\sqrt{-\triangle_{\mathrm{PML}}}\big)\,,\end{aligned}$$

where $\beta = (\beta_1,\beta_2) \in \mathbb{N}^2$ denotes a 2-index. Regarding $\sigma\big(\sqrt{-\triangle_{\mathrm{PML}}}\big)$, we use Padé's approximants, so that

$$\begin{aligned}\sigma\big((-\triangle_{\mathrm{PML}})^{q+1/2}\big) \approx \sum_{|\beta|=0}^{\infty}\frac{(-\mathtt{i})^{|\beta|}}{\beta!}\partial_x^{\beta_1}\partial_y^{\beta_2}\Big(\frac{1}{S_x^2}|\xi_x|^2 - \mathtt{i}\frac{S_x'}{S_x^3}\xi_x + \frac{1}{S_y^2}|\xi_y|^2 - \mathtt{i}\frac{S_y'}{S_y^3}\xi_y\Big)^q\\ \times\partial_{\xi_x}^{\beta_1}\partial_{\xi_y}^{\beta_2}\Big(\textstyle\sum_{k=0}^{M}a_k^{(M)} - \sum_{k=1}^{M}a_k^{(M)}d_k^{(M)}\big(\frac{1}{S_x^2}|\xi_x|^2 + \frac{1}{S_y^2}|\xi_y|^2\\ +\mathtt{i}\frac{S_x'}{S_x^3}\xi_x + \mathtt{i}\frac{S_y'}{S_y^3}\xi_y + d_k^{(M)}\big)^{-1}\Big)\,.\end{aligned}$$

From a practical point of view, we define

$$\sigma\big((-\triangle_{\mathrm{PML}})_m^{q+1/2}\big) := \sum_{|\beta|=0}^{m}\lambda_\beta^{(m)}(x,y,\xi_x,\xi_y)\,,$$

where

$$\begin{aligned}\lambda_\beta^{(m)} := \partial_x^{\beta_1}\partial_y^{\beta_2}\Big(\frac{1}{S_x^2}|\xi_x|^2 - \mathtt{i}\frac{S_x'}{S_x^3}\xi_x + \frac{1}{S_y^2}|\xi_y|^2 - \mathtt{i}\frac{S_y'}{S_y^3}\xi_y\Big)^q\\ \times\partial_{\xi_x}^{\beta_1}\partial_{\xi_y}^{\beta_2}\Big(\textstyle\sum_{k=0}^{M}a_k^{(M)} - \sum_{k=1}^{M}a_k^{(M)}d_k^{(M)}\big(\frac{1}{S_x^2}|\xi_x|^2 + \frac{1}{S_y^2}|\xi_y|^2\\ +\mathtt{i}\frac{S_x'}{S_x^3}\xi_x + \mathtt{i}\frac{S_y'}{S_y^3}\xi_y + d_k^{(M)}\big)^{-1}\Big)\,.\end{aligned}$$

Tedious computations allow for an explicit expression of $\{\lambda_\beta^{(m)}\}_\beta$. We get

$$\begin{aligned}\lambda_{(0,0)}^{(m)} = \Big(\frac{1}{S_x^2}|\xi_x|^2 - \mathtt{i}\frac{S_x'}{S_x^3}\xi_x + \frac{1}{S_y^2}|\xi_y|^2 - \mathtt{i}\frac{S_y'}{S_y^3}\xi_y\Big)^q\\ \times\Big(\sum_{k=0}^{M}a_k^{(M)} - \sum_{k=1}^{M}a_k^{(M)}d_k^{(M)}\big(\frac{1}{S_x^2}|\xi_x|^2 + \frac{1}{S_y^2}|\xi_y|^2 + \mathtt{i}\frac{S_x'}{S_x^3}\xi_x + \mathtt{i}\frac{S_y'}{S_y^3}\xi_y + d_k^{(M)}\big)^{-1}\Big)\,.\end{aligned}$$

Next, we obtain

$$\begin{aligned}\lambda_{(1,0)}^{(m)} = -q\Big(\partial_x\Big(\frac{1}{S_x^2}\Big)|\xi_x|^2 - \mathtt{i}\partial_x\Big(\frac{S_x'}{S_x^3}\Big)\xi_x\Big)\Big(\frac{1}{S_x^2}|\xi_x|^2 - \mathtt{i}\frac{S_x'}{S_x^3}\xi_x + \frac{1}{S_y^2}|\xi_y|^2 - \mathtt{i}\frac{S_y'}{S_y^3}\xi_y\Big)^{q-1}\\ \times\frac{\sum_{k=1}^{M}a_k^{(M)}d_k^{(M)}\big(\frac{1}{S_x^2}\partial_{\xi_x}|\xi_x|^2 + \mathtt{i}\frac{S_x'}{S_x^3}\big)}{\sum_{k=0}^{M}a_k^{(M)} - \sum_{k=1}^{M}a_k^{(M)}d_k^{(M)}\big(\frac{1}{S_x^2}|\xi_x|^2 + \frac{1}{S_y^2}|\xi_y|^2 + \mathtt{i}\frac{S_x'}{S_x^3}\xi_x + \mathtt{i}\frac{S_y'}{S_y^3}\xi_y + d_k^{(M)}}\,,\end{aligned}$$

and

$$\begin{aligned}\lambda_{(0,1)}^{(m)} = -q\Big(\partial_y\Big(\frac{1}{S_y^2}\Big)|\xi_y|^2 - \mathtt{i}\partial_y\Big(\frac{S_y'}{S_y^3}\Big)\xi_x\Big)\Big(\frac{1}{S_x^2}|\xi_x|^2 - \mathtt{i}\frac{S_x'}{S_x^3}\xi_x + \frac{1}{S_y^2}|\xi_y|^2 - \mathtt{i}\frac{S_y'}{S_y^3}\xi_y\Big)^{q-1}\\ \times\frac{\sum_{k=1}^{M}a_k^{(M)}d_k^{(M)}\big(\frac{1}{S_x^2}\partial_{\xi_y}|\xi_y|^2 + \mathtt{i}\frac{S_y'}{S_y^3}\big)}{\sum_{k=0}^{M}a_k^{(M)} - \sum_{k=1}^{M}a_k^{(M)}d_k^{(M)}\big(\frac{1}{S_x^2}|\xi_x|^2 + \frac{1}{S_y^2}|\xi_y|^2 + \mathtt{i}\frac{S_x'}{S_x^3}\xi_x + \mathtt{i}\frac{S_y'}{S_y^3}\xi_y + d_k^{(M)}}\,.\end{aligned}$$

For $|\beta| = 1$, we have to construct $\lambda_{(1,1)}^{(m)}$, $\lambda_{(2,0)}^{(m)}$, and $\lambda_{(0,2)}^{(m)}$.

B Cauchy integral approximation

In this Appendix, we discuss the approximation of A^α by using the Cauchy integral representation, for $\alpha \in \mathbb{R}$ and $A \in \mathbb{R}^{N\times N}$. We recall that

$$A^\alpha = (2\pi\mathtt{i})^{-1}A\int_{\Gamma_A} z^{\alpha-1}(zI-A)^{-1}dz\,, \tag{63}$$

where Γ_A is a closed contour in the complex plane enclosing the spectrum of matrix A, where the latter is assumed to have its spectrum in $\mathbb{C}\backslash\mathbb{R}_-$. This approach can be quite inefficient if the spectrum of the matrix A has a large radius. This leads to a straightforward approximation of the Cauchy integral based on a quadrature rule

$$A_h^\alpha = (2\pi\mathtt{i})^{-1} A \sum_j \Delta z_j \theta_j z_j^{\alpha-1} (z_j I - A)^{-1},$$

where $\{\theta_j\}_j$ are interpolation weights and $\{z_j\}_j \in \Gamma_A \subset \mathbb{C}$ are the interpolation nodes on Γ_A. There are many ways to reduce the computational complexity [6,7,25,31]. Among others, we propose in [6] the following possible approach based on the use of a traditional preconditioner M for the linear system. Typically, $M \approx A^{-1}$, and MA has a spectrum clustering at the point $(1,0)$ in the complex plane. Thus

$$(MA)^\alpha = (2\pi\mathtt{i})^{-1} MA \int_{\Gamma_{MA}} z^{\alpha-1} (zI - MA)^{-1} dz, \tag{64}$$

where $\ell(\Gamma_M) \ll \ell(\Gamma)$, ℓ denoting the length of a curve in the complex plan. In particular computing (64) is cheaper than (63). However, the connection between $(MA)^\alpha$ and A^α is not necessarily simple.

Proposition 12 *Assume that A is symmetric and M is a preconditioner commuting with A. Then, we have [6]*

$$A^\alpha = M^{-\alpha} (MA)^\alpha.$$

In other words, the polynomial preconditioning allows for an efficient computation of matrix powers.

Practically, the proposed preconditioning allows for a reduction of the length of the contour enclosing the spectrum of the precontioned matrix MA, as long as $M^{-\alpha}$ can be efficiently computed. We refer to [6] for additional details.