



Preconditioned Pseudo-Spectral Gradient Flow for Computing the Steady-State of Space Fractional Cahn-Allen Equations With Variable Coefficients

Saleh Mousa Alzahrani ^{1*} and Chniti Chokri ^{2*}

¹Department of Mathematics, University College in Al-Qunfudhah, Umm Al-Qura University, Al-Qunfudhah, Saudi Arabia,

²Department of Mathematics, IPEIN, Carthage University, Nabeul, Tunisia

OPEN ACCESS

Edited by:

Valentina De Simone,
University of Campania Luigi Vanvitelli,
Italy

Reviewed by:

Harun-Or Roshid,
Pabna University of Science and
Technology, Bangladesh
Yang Liu,
Inner Mongolia University, China

*Correspondence:

Saleh Mousa Alzahrani
salzahrani@uqu.edu.sa
Chniti Chokri
chokri.chniti@ipein.mu.tn

Specialty section:

This article was submitted to
Mathematical and Statistical Physics,
a section of the journal
Frontiers in Physics

Received: 27 December 2021

Accepted: 04 March 2022

Published: 08 April 2022

Citation:

Alzahrani SM and Chokri C (2022)
Preconditioned Pseudo-Spectral
Gradient Flow for Computing the
Steady-State of Space Fractional
Cahn-Allen Equations With
Variable Coefficients.
Front. Phys. 10:844294.
doi: 10.3389/fphy.2022.844294

The aim of this paper is to propose some efficient and accurate numerical methods to compute the steady-state of variable coefficients space fractional Cahn-Allen equations. The approach combines an adaptive time stepping semi-implicit gradient flow method to minimize the fractional energy functional and pseudo-spectral approximation schemes. Based on the use of a preconditioned GMRES, the space fractional Cahn-Allen equation is then solved efficiently. The full methodology is supported by the numerical solution of a one-dimensional problem.

Keywords: nonlinear pseudodifferential equation, fractional derivative, dynamics, gradient flow, pseudo-spectral method

1 INTRODUCTION

During the last 2 decades, the mathematical analysis and numerical simulation of fractional partial differential equations became one of the most important hot topics in applied and computational mathematics [1–4]. One of the reasons of this growing interest is related to the fact that these new models provide a better description of some physical phenomena. Indeed, for example, for the heat equation, considering space fractional effects allows to handle anomalous diffusion [5] when the stochastic process driving the medium is a Lévy α -stable flight. Fractional PDEs are also of interest when modeling e.g., the behavior of turbulent flows [6, 7], the chaotic effects in the dynamics of conservative systems [8] but also for the transport of contaminant in groundwater flow [9, 10] or finally in mathematical finance [11]. In [12], a fast algorithm based on time two-mesh finite element scheme, which aims at solving nonlinear problems quickly, is considered to numerically solve the nonlinear space fractional Cahn-Allen equations with smooth and non-smooth solutions. In [13], different schemes has been effectively employed for finding the exact solutions and dynamics of the Cahn-Allen model and the dispersive predator-prey model. The goal of this paper is to contribute to the numerical solution to such systems, and most particularly to a space fractional Cahn-Allen equation [14] with variable coefficients. The integer order equation is usually used to describe the complex phase separation and coarsening phenomena in a solid. Here, the fractional order appears as an exponent $\alpha/2$ ($1 \leq \alpha \leq 2$) in the laplacian operator, $\alpha = 2$ corresponding to the standard situation. The fractional Cahn-Allen equation can then be seen as a more general situation than the standard one. It has been introduced (see e.g., [15–17]) for studying the competing stable phases having an identical Lyapunov functional density for Lévy processes of order $\alpha/2$. This allows to include the situation where the random walk has correlations, non-Gaussian or non-Markovian memory effects which cannot be described by the standard Laplacian.

A recent interesting application [18, 19] considers image segmentation techniques based on the fractional Cahn-Allen equation, with examples in medical imaging. Here, the objective of the paper is to show how the widely used pseudo-spectral approximation schemes which are well-adapted to solve constant coefficients PDEs (see e.g., [20–23]) can be adapted to the variable coefficients case for computing the steady-state of fractional Cahn-Allen equations based on the gradient flow formulation and its suitable discretization.

To this end, we introduce in **Section 2** the problem setting for an example of space fractional Cahn-Allen equation (the approach could be probably extended to other kinds of equations), and we provide a few useful definitions for the pseudo-spectral approximation, in particular concerning the spectral definition of the fractional laplacian. The gradient flow formulation is also given. **Section 3** is devoted to the development of numerical schemes for the gradient flow as well as an adaptive time stepping method. We also explain how to get an efficient implementation of the pseudo-spectral approximation schemes, most particularly regarding the use of a Krylov subspace iterative solver with a well-designed preconditioner. In **Section 4**, we analyze the full methodology in the case of a one-dimensional example. This shows that the newly designed method performs very well for various fractional orders. Finally, we conclude in **Section 5**.

2 PROBLEM SETTING AND DEFINITION

The problem that we propose to solve is the following: find $u: \mathbb{R}^+ \times \Omega \rightarrow \mathbb{R}$ solution to the variable coefficients space fractional Cahn-Allen (SFCA) equation

$$\begin{cases} \partial_t u = -\kappa(\mathbf{x})(\varepsilon(-\Delta)^{\alpha/2} u + f(u)), & (t, \mathbf{x}) \in \mathbb{R} \times \Omega, \\ u(t = 0, \mathbf{x}) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \partial_n u(\mathbf{x}) = 0, & t > 0, \mathbf{x} \in \Gamma := \partial\Omega. \end{cases} \tag{1}$$

In the above system, $u := u(t, \mathbf{x})$ represents the concentration of one of the mixtures at time t and point \mathbf{x} appearing in the modeling of the phase of a binary mixture. The domain of computation is a d -dimensional rectangular box: $\Omega := \prod_{k=1}^d a_k; b_k$, with $-\infty < a_k < b_k < \infty$, for $k = 1, \dots, d$, with $d = 1, 2, 3$. Vector \mathbf{n} denotes the outwardly directed unit normal vector to Ω . The strictly positive function κ is the variable mobility. For simplifications, it is standard to set this function as equal to 1, but we consider also here the nontrivial situation for pseudo-spectral methods where it is \mathbf{x} -dependent. In the standard constant coefficients CA equation with integer order $\alpha = 2$, the (usually small) constant $\varepsilon > 0$ stands for the interfacial width, which captures the dominating effect of reaction kinetics and stays for effective diffusivity.

In this paper, we consider the SFCA equation for a fractional laplacian $(-\Delta)^{\alpha/2}$, $1 < \alpha \leq 2$, with homogeneous Neumann boundary conditions [24]. Nevertheless, homogeneous Dirichlet or periodic boundary conditions can also be included. In the one-dimensional case, we introduce λ_j as the eigenvalues and φ_j , $j \in \mathbb{N}$, as the associated eigenfunctions of the

operator $-\Delta := -\partial_x^2$ with homogeneous Neumann boundary condition:

$$\begin{cases} -\Delta \varphi_j = \lambda_j \varphi_j, & \text{in } \Omega := (a, b), \\ \partial_n \varphi_j = 0, & \text{at } x = a \text{ and } b. \end{cases} \tag{2}$$

These functions, which are explicitly defined by

$$\lambda_j = \left(\frac{j\pi}{b-a}\right)^2, \quad \varphi_j(x) := \sqrt{\frac{2}{b-a}} \cos\left(\frac{j\pi(x-a)}{b-a}\right), \tag{3}$$

form a complete set of orthogonal eigenfunctions. In the case of a homogeneous Dirichlet boundary condition, then one would get

$$\lambda_j = \left(\frac{(j+1)\pi}{b-a}\right)^2, \quad \varphi_j(x) := \sqrt{\frac{2}{b-a}} \sin\left(\frac{(j+1)\pi(x-a)}{b-a}\right). \tag{4}$$

Let us define $u \in \mathcal{U}_{\alpha/2}$ as

$$\mathcal{U}_{\alpha/2} := \left\{ u = \sum_{j=0}^{\infty} \hat{u}_j \varphi_j, \hat{u}_j = \langle u, \varphi_j \rangle, \sum_{j=0}^{\infty} \mu_j^\alpha |\hat{u}_j|^2 < \infty \right\}.$$

Then, if we assume that $u \in \mathcal{U}_{\alpha/2}$, the spectral definition of the fractional laplacian is

$$(-\partial_x^2)^{\alpha/2} u = \sum_{j=0}^{\infty} \mu_j^\alpha \hat{u}_j \varphi_j,$$

with $\mu_j := \sqrt{\lambda_j}$, taking the principal determination of the square-root.

The nonlinear term f is given by $f(u) = F(u)$. The choice of the potential function F , also called Helmholtz free-energy density, is related to the physical situation, e.g., the case of a convex quartic double-well potential or a non-convex logarithmic potential [25, 26]. Here, to fix the ideas, we consider the first choice with a variable space function, for $\mathbf{x} \in \Omega$, $0 < \beta(\mathbf{x}) \leq 1$, generalizing the standard constant situation $\beta = 1$, and setting

$$F(u) = \frac{1}{4\kappa(\mathbf{x})\beta(\mathbf{x})} (\beta(\mathbf{x})u^2 - 1)^2, \tag{5}$$

leading to

$$f(u) = F(u)' = \frac{1}{\kappa(\mathbf{x})} (\beta(\mathbf{x})u^2 - 1)u. \tag{6}$$

Let us introduce the generalized energy functional

$$E_{\alpha/2}(u) := \frac{1}{2} (\varepsilon(-\Delta)^{\alpha/2} u, u)_{0,\Omega} + \int_{\Omega} F(u) d\mathbf{x}, \tag{7}$$

with $(\cdot, \cdot)_{0,\Omega}$ as the $L^2(\Omega)$ -norm. As in [21, 24], we define $\bar{u} = (-\Delta)^{\alpha/2-1} u$, with $1 < \alpha \leq 2$, which leads to the fractional gradient as $\nabla^{\alpha/2} := \nabla(-\Delta)^{\alpha/2-1}$, and then $\nabla^\alpha u = \nabla \bar{u}$. Therefore, one gets

$$E_{\alpha/2}(u)(t) := \frac{\varepsilon}{2} \|\nabla^{\alpha/2} u\|_{0,\Omega}^2 + \int_{\Omega} F(u) d\mathbf{x}, \tag{8}$$

setting $\|u\|_{0,\Omega}^2 := (u, u)_{0,\Omega}$. Based on our notations, then solving **Equation 1** is equivalent to the gradient flow formulation

$$\begin{cases} \partial_t u = -\kappa(\mathbf{x})\nabla_u E_{\alpha/2}(u) = 0, (t, \mathbf{x}) \in \mathbb{R} \times \Omega, \\ u(t = 0, \mathbf{x}) = u_0(\mathbf{x}), \mathbf{x} \in \Omega, \\ \partial_n u(\mathbf{x}) = 0, t > 0, \mathbf{x} \in \Gamma := \partial\Omega. \end{cases} \quad (9)$$

One also can prove that the energy is nonincreasing, after integration over the domain Ω , taking a test-function $v = (\varepsilon(-\Delta)^{\alpha/2}u + f(u))$, and since κ is a positive function

$$\text{for } 1 < \alpha \leq 2 \quad E_{\alpha/2}(t) \leq E_{\alpha/2}(s), \text{ for } s \leq t,$$

3 NUMERICAL SCHEMES

3.1 Minimization Algorithms

When one wants to reach the long time solution to **Eq. 1**, i.e., to compute the steady-state, we can directly discretize the initial boundary-value system **Eq. 9** as a time-dependent problem which is a standard approach. An alternative solution consists in looking at the long-time solution to the gradient flow system **Eq. 9** which cancels the energy gradient (Euler equation), based on an optimization algorithm related to the minimization of the energy functional. A standard way is to use a gradient-type method by building a sequence of minimizers $u^{(n)}$ which is supposed to converge towards the steady-state, hence corresponding to the long-time dynamics solution of the gradient flow, i.e., for large values of n .

Based on this point of view, we can for example build the following simple scheme

$$\begin{cases} \text{iterate on } n \text{ until convergence} \\ u^{(n+1)} = u^{(n)} - \delta^{(n)}\kappa\nabla E_{\alpha/2}(u^{(n)}), \quad \mathbf{x} \in \Omega, \\ u^{(0)} = u_0, \quad \mathbf{x} \in \Omega, \\ \partial_n u^{(n)}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Gamma := \partial\Omega, \\ \text{end iterate } n \end{cases} \quad (10)$$

with

$$\nabla E_{\alpha/2}(u^{(n)}) := \varepsilon(-\Delta)^{\alpha/2}u^{(n)} + f(u^{(n)}),$$

and where $\rho^{(n)} > 0$ is the local descent step of the gradient method which is computed in such a way that the energy is indeed decaying between two successive steps n and $n + 1$.

Let us now make a few remarks. If one considers that the step $\delta^{(n)}$ is a constant step, which means that we use a constant-step minimization gradient algorithm, then scheme **Eq. 10** would correspond to an explicit Euler discretization in time of **Eq. 1** with constant time step $\Delta t = \rho$. In the case of the computation of an optimal step, or at least an adapted gradient step $\rho^{(n)}$, then one gets an explicit Euler scheme with time stepping. It is well-known that the steepest descent method with constant step is not optimal and may suffer from convergence problems, therefore requiring the computation of an optimal step. To this end, we use an optimal linear search method following (the starting value $\rho = 1$ can be changed)

$$\left\{ \begin{array}{l} \rho = 1 \\ \text{compute } \phi^1 = u^{(n)} - \rho\kappa\nabla E_{\alpha/2}(u^{(n)}) \\ \text{compute } \phi^{1/2} = u^{(n)} - \frac{\rho}{2}\kappa\nabla E_{\alpha/2}(u^{(n)}) \\ \text{while } E_{\alpha/2}(\phi^1) > E_{\alpha/2}(\phi^{1/2}) \text{ do} \\ \quad \rho = \rho/2 \\ \quad \phi^1 = \phi^{1/2} \\ \quad \text{compute } \phi^{1/2} = u^{(n)} - \frac{\rho}{2}\kappa\nabla E_{\alpha/2}(u^{(n)}) \\ \text{end while} \\ \rho^{(k)} = \rho \\ u^{(n+1)} = \phi^{1/2} \end{array} \right. \quad (11)$$

which will also be used for all the other alternative schemes below.

Concerning the stopping criterion related to **Eq. 10**, one possibility consists in forcing a maximal value of n . Based on the interpretation of the steepest descent step $\rho^{(n)}$ as a local time step Δt_n , this would correspond to fixing a maximal time of computation t_N for a certain *a priori* known value of N . Nevertheless, N is never *a priori* known and should be determined dynamically into the loop on n thanks to a stopping criterion. Here, we propose the following strong convergence test

$$|E_{\alpha/2}^{(n+1)} - E_{\alpha/2}^{(n)}| \leq \epsilon_1, \quad (12)$$

for a very small value of ϵ_1 , where $E_{\alpha/2}^{(n)} := E_{\alpha/2}(u^{(n)})$. Another possibility is to fix that the uniform norm on Ω between two successive solutions $u^{(n)}$ and $u^{(n+1)}$ is very small (i.e. $\|u^{(n+1)} - u^{(n)}\|_{\infty,\Omega} \leq \epsilon_1$) but this may lead to a criterion which does not provide the minimum.

The stability and convergence properties of the steepest descent algorithm depend on the fact that it is also fully explicit. Here we consider the following alternative scheme where we implicit the energy gradient term

$$\begin{cases} \text{iterate on } n \text{ until convergence} \\ u^{(n+1)} = u^{(n)} - \delta^{(n)}\kappa\nabla E_{\alpha/2}(u^{(n+1)}), \quad \mathbf{x} \in \Omega, \\ u^{(0)} = u_0, \quad \mathbf{x} \in \Omega, \\ \partial_n u^{(n+1)}(\mathbf{x}) = 0, t > 0, \quad \mathbf{x} \in \Gamma := \partial\Omega, \\ \text{end iterate } n \end{cases} \quad (13)$$

which, in terms of PDEs can be rewritten as

$$\begin{cases} \text{iterate on } n \text{ until convergence} \\ u^{(n+1)} = u^{(n)} - \delta^{(n)}\kappa(\mathbf{x})(\varepsilon(-\Delta)^{\alpha/2}u^{(n+1)} + f(u^{(n+1)})), \quad \mathbf{x} \in \Omega, \\ u^{(0)} = u_0, \quad \mathbf{x} \in \Omega, \\ \partial_n u^{(n+1)}(\mathbf{x}) = 0, t > 0, \quad \mathbf{x} \in \Gamma := \partial\Omega, \\ \text{end iterate } n \end{cases} \quad (14)$$

The problem related to the above fully implicit (Euler) scheme is that we have to solve a nonlinear equation at each gradient step, leading to a higher computational cost if for example a fixed-point or Newton-type algorithm is used, most particularly in the framework of pseudo-spectral approximation schemes. Here, we consider the semi-implicit (Euler) (E, for short) scheme

$$\left\{ \begin{array}{l} \text{iterate on } n \text{ until convergence} \\ u^{(n+1)} = u^{(n)} - \delta^{(n)} \kappa(\mathbf{x}) (\varepsilon(-\Delta)^{\alpha/2} u^{(n+1)} + f(u^{(n)})), \quad \mathbf{x} \in \Omega, \\ u^{(0)} = u_0, \quad \mathbf{x} \in \Omega, \\ \partial_n u^{(n+1)}(\mathbf{x}) = 0, t > 0, \quad \mathbf{x} \in \Gamma := \partial\Omega, \\ \text{end iterate } n \end{array} \right. \tag{15}$$

which avoids any nonlinear solution and leads to solving the linear PDE formulation

$$\left\{ \begin{array}{l} \text{iterate on } n \text{ until convergence} \\ \left(\frac{1}{\delta^{(n)}} I + \kappa(\mathbf{x}) \varepsilon(-\Delta)^{\alpha/2} \right) u^{(n+1)} = \frac{1}{\delta^{(n)}} u^{(n)} - \kappa(\mathbf{x}) f(u^{(n)}), \quad \mathbf{x} \in \Omega, \\ u^{(0)} = u_0, \quad \mathbf{x} \in \Omega, \\ \partial_n u^{(n+1)}(\mathbf{x}) = 0, t > 0, \quad \mathbf{x} \in \Gamma := \partial\Omega, \\ \text{end iterate } n \end{array} \right. \tag{16}$$

where I is the identity operator. In fact, it can be seen as the formulation of the fully-implicit (Euler) scheme Eq. 15 with one iteration of the fixed-point algorithm. If one uses the decomposition of the Ginzburg–Landau free energy into a linear kinetic part and a nonlinear part related to the potential function F

$$E_{\alpha/2}(u) := E_{\alpha/2}^{\text{kin}}(u) + E_{\alpha/2}^F(u), \tag{17}$$

setting

$$\left\{ \begin{array}{l} E_{\alpha/2}^{\text{kin}}(u) := \frac{\varepsilon}{2} \|\nabla^{\alpha/2} u\|_{0,\Omega}^2, \\ E_{\alpha/2}^F(u) := \int_{\Omega} F(u) d\mathbf{x}, \end{array} \right. \tag{18}$$

then Eq. 15 can also be seen as an algorithm where the kinetic energy is implicitly approximated while the potential energy is explicit resolved

$$\left\{ \begin{array}{l} \text{iterate on } n \text{ until convergence} \\ u^{(n+1)} = u^{(n)} - \delta^{(n)} \kappa(\mathbf{x}) \nabla E_{\alpha/2}^{\text{kin}}(u^{(n+1)}) - \delta^{(n)} \kappa(\mathbf{x}) \nabla E_{\alpha/2}^F(u^{(n)}), \mathbf{x} \in \Omega, \\ u^{(0)} = u_0, \quad \mathbf{x} \in \Omega, \\ \partial_n u^{(n+1)}(\mathbf{x}) = 0, t > 0, \quad \mathbf{x} \in \Gamma := \partial\Omega, \\ \text{end iterate } n \end{array} \right. \tag{19}$$

An alternative to the semi-implicit Euler scheme is to use a semi-implicit Crank-Nicolson (CN, for brevity) scheme which writes

$$\left\{ \begin{array}{l} \text{iterate on } n \text{ until convergence} \\ \left(\frac{2}{\delta^{(n)}} I + \kappa(\mathbf{x}) \varepsilon(-\Delta)^{\alpha/2} \right) u^{(n+1/2)} = \frac{2}{\delta^{(n)}} u^{(n)} - \frac{\kappa}{2} f(u^{(n)}), \quad \mathbf{x} \in \Omega, \\ u^{(0)} = u_0, \quad \mathbf{x} \in \Omega, \\ \partial_n u^{(n+1/2)}(\mathbf{x}) = 0, t > 0, \quad \mathbf{x} \in \Gamma := \partial\Omega, \\ \text{end iterate } n \end{array} \right. \tag{20}$$

and is similar to Eq. 19 but evaluating the kinetic energy at the mid-point $u^{n+1/2} := (u^{(n+1)} + u^{(n)})/2$ as well as the Neumann

boundary condition. For both the E and CN schemes, the line search algorithm Eq. 11 is adapted thanks to the formulation.

3.2 Pseudo-Spectral Discretization in Space and Linear System Solution

In a practical computation, we approximate the solution u by a J -terms finite-dimensional approximation $u_J := \sum_{j=0}^J \hat{u}_j \varphi_j$, $J \in \mathbb{N}$. To this end, the fast cosine/sine transform can be used for computing the fractional laplacian when Neumann/Dirichlet boundary conditions are used. For the case of periodic boundary conditions, FFT-based methods are applied.

When one uses a fast transform (cosine, sine, FFT), we can proceed to the efficient computation of the application of an operator to a given function. For example, let us consider that one wants to numerically apply the fractional laplacian of order α to a given function v , i.e., we wish to compute $w := (-\Delta)^{\alpha/2} v$, where v is a given field known by its values v_j at the nodes $x_j := -L/2 + (2j + 1)h/2$, for $0 \leq j \leq J - 1$ and where $h = L/J$ denotes the uniform spatial meshsize ($L = b - a$). Then, we get the practical efficient calculation of w by the cosine (dct) and inverse cosine (idct) transforms in one-dimension through the Matlab code

$$\begin{aligned} L &= b - a; \\ J &= 2^{\wedge} 8; \\ h &= L/J; \\ \text{lambda}j &= ((0: J - 1)*\text{pi}/L)^{\wedge} 2.^{\wedge}; \\ w &= \text{idct}(\text{lambda}j.^{\wedge}(\text{alpha}/2).*\text{dct}(v)); \end{aligned} \tag{21}$$

For all the previous schemes presented in section 3.1, and for a constant function κ , the finite-dimensional solution in space can be obtained directly thanks to the above pseudo-spectral discretization. Indeed, even for the semi-implicit schemes, the operators to invert, i.e. $((\delta^{(n)})^{-1} I + \kappa \varepsilon(-\Delta)^{\alpha/2})$ for the E schemes, can be computed directly through their symbols. For the Neumann boundary conditions, this means that we can apply the discrete symbols of $((\delta^{(n)})^{-1} I + \kappa \varepsilon(-\Delta)^{\alpha/2})^{-1}$ in the dual space thanks to its spectral decomposition involving the eigenvalues $((\delta^{(n)})^{-1} I + \kappa \varepsilon \mu_j^{\alpha})^{-1}$, $0 \leq j \leq J - 1$, and in a similar way to Eq. 21.

However, for the more complex situation that we consider in this paper, this is no longer possible. Indeed, a variable coefficient operator cannot be inverted explicitly in the Fourier space. Let us for example consider the situation of a full space representation (no boundary condition). Then, we can use the inverse Fourier representation of the operator $((\delta^{(n)})^{-1} I + \kappa(\mathbf{x}) \varepsilon(-\Delta)^{\alpha/2})$ following

$$\begin{aligned} w &= A(\mathbf{x}, D_{\mathbf{x}})v := \left((\delta^{(n)})^{-1} I + \kappa(\mathbf{x}) \varepsilon(-\Delta)^{\alpha/2} \right) v \\ &= \mathcal{F}^{-1}(a(\mathbf{x}, \xi) \mathcal{F}(u)(\xi)), \end{aligned} \tag{22}$$

where $\xi \in \mathbb{R}^d$ is the dual Fourier variable, the direct and inverse Fourier transform being denoted by \mathcal{F} and \mathcal{F}^{-1} , respectively, and defined by

$$\mathcal{F}(f)(\xi) = \hat{f}(\xi) = \int_{\mathbb{R}^d} e^{-ix \cdot \xi} f(x) dx,$$

and

$$\mathcal{F}^{-1}(f)(\mathbf{x}) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} e^{i\mathbf{x}\cdot\xi} \hat{f}(\xi) d\xi.$$

The symbol of the pseudodifferential operator [27] is given here by the expression

$$a(\mathbf{x}, \xi) := \left((\delta^{(m)})^{-1} + \kappa(\mathbf{x})\varepsilon\|\xi\|^\alpha \right). \tag{23}$$

The inner product of two vectors \mathbf{p} and \mathbf{q} in \mathbb{R}^d is such that: $\mathbf{p} \cdot \mathbf{q} := \sum_{k=1}^d p_k q_k$, and the associated norm is $\|\mathbf{p}\| := \sqrt{\mathbf{p} \cdot \mathbf{p}}$. The symbol a of the operator A , which is of order α , then generates an operator of order α acting between a Sobolev space of order m onto $m - \alpha$. Let us define B as the (left) pseudo-inverse operator of A , i.e., $BA = I$ modulo a smoothing operator. The operator B can be computed recursively through an asymptotic expansion of its total symbol $b := b(\mathbf{x}, \xi)$ by the formula [27].

$$\sigma_{BA} = b \# a(\mathbf{x}, \xi) \sim \sum_{\boldsymbol{\eta} \in \mathbb{N}^d} \frac{1}{\boldsymbol{\eta}!} \partial_\xi^\boldsymbol{\eta} b(\mathbf{x}, \xi) D_{\mathbf{x}}^\boldsymbol{\eta} a(\mathbf{x}, \xi) = 1, \tag{24}$$

where σ_{BA} is the symbol of the operator BA . The operator $D^\boldsymbol{\eta}$ is: $D^\boldsymbol{\eta} = D_1^{\eta_1} \dots D_d^{\eta_d}$, $D = (D_1, \dots, D_d)$, setting $D_k := -i\partial_k = -i\partial_{x_k}$, $k = 1, \dots, d$, and $i := \sqrt{-1}$. The multi-index $\boldsymbol{\eta}$ is defined as a vector $\boldsymbol{\eta} := (\eta_1, \dots, \eta_d)$ with d positive integer components, i.e. $\boldsymbol{\eta} \in \mathbb{N}^d$. The factorial is: $\boldsymbol{\eta}! = \prod_{k=1}^d \eta_k!$. Thanks to these notations, one gets: $\sigma_{(-\Delta)^{\alpha/2}} = \|\xi\|^\alpha$. If κ is a positive constant, then the composition Eq. 24 leads to

$$b(\mathbf{x}, \xi) = b(\xi) = a(\mathbf{x}, \xi)^{-1} = a(\xi)^{-1},$$

since a does not depend on \mathbf{x} and then the infinite sum Eq. 24 reduces to only one term for $\boldsymbol{\eta} = \mathbf{0}$. When a is \mathbf{x} -dependent, then the asymptotic expansion is infinite and b includes an infinite number of terms. The operator $B_{-\alpha/2}$ with symbol $b_{-\alpha/2} := a(\mathbf{x}, \xi)^{-1}$ is a regularizing pseudodifferential operator of order $-\alpha$ which is not the inverse of A . Indeed, we have: $B_{-\alpha/2}A = I + R$, where R is a pseudodifferential operator of negative order which is however usually not an infinitely smoothing operator.

Now, if we come back to Eq. 22, we see that we can rewrite it as

$$w = A(\mathbf{x}, D_{\mathbf{x}})v := \left((\delta^{(m)})^{-1} I + \kappa(\mathbf{x})\varepsilon(-\Delta)^{\alpha/2} \right) v = (\delta^{(m)})^{-1} v + \kappa(\mathbf{x})\varepsilon\mathcal{F}^{-1}(\|\xi\|^\alpha \hat{v}(\xi)), \tag{25}$$

which can be discretized directly thanks to a FFT, fast cosine or sine transform according to the boundary conditions. Let us write then the corresponding discrete operation as

$$\mathbf{w} := [[A_n]]\mathbf{v} := \left((\delta^{(m)})^{-1} I + \varepsilon\kappa[(-\Delta)^{\alpha/2}] \right) \mathbf{v}, \tag{26}$$

where \mathbf{v} and \mathbf{w} denotes the given and unknown vectors, respectively, at the discretization points, I is the identity matrix of size J , κ is the diagonal matrix with the entries $\kappa(x_j)$, for $0 \leq j \leq J-1$, and $[(-\Delta)^{\alpha/2}]$ \mathbf{v} denotes the application of the fast algorithm for evaluating $A\mathbf{v}$ at the discrete level. Therefore, $[[A]]$ is a matrix-free operator given through a fast evaluation procedure (with e.g., a function call $\mathcal{Q}(\mathbf{v})A(\mathbf{v})$ in

Matlab) requiring $\mathcal{O}(J \log J)$ operations and a memory cost of $\mathcal{O}(J)$ entries.

As previously seen, using the E or CN approximations needs the resolution of a linear system. When κ is a constant function, this can be trivially done thanks to the explicit inversion of the operator through the inverse of its symbol. For a function κ depending on \mathbf{x} , then the process for solving one of the linear systems is to use an iterative solver instead of a direct method. Among them, fixed-point methods [28] (Jacobi, Gauss-Seidel, successive approximation) can be used. Nevertheless, it is well-known that they are not robust since their convergence rate strongly depends on the spectral radius of the iteration matrix. More efficient methods are based on Krylov subspace iterative solvers [28, 29]. We choose here the Generalized Minimal Residual (GMRES) method [29] which is extremely robust (other methods, like e.g., BiCGStab, have been tested but are proved to be less efficient within the framework of this paper; we do not report the results for the sake of conciseness). This method can solve matrix-free linear systems by calling an argument which is the function $\mathcal{Q}(\mathbf{v})A(\mathbf{v})$ related to the matrix-vector product $\mathbf{w} := [[A_n]]\mathbf{v}$. The convergence rate of the GMRES is related to the spectral distribution of the matrix of the linear system and the cost of applying it to our problem is therefore $\mathcal{O}(n_n^{iter} J \log J)$, where n_n^{iter} is the number of iterations required for solving the linear system with a tolerance ϵ_2 . To accelerate the computations, one usually includes a preconditioner $[[B_n]]$ which is an approximation of the inverse of $[[A_n]]$. From the above discussion, a natural choice is

$$\mathbf{v} := [[B_n]]\mathbf{w}, \tag{27}$$

where $[[B_n]]$ is the discretization of the operator with symbol a_n^{-1} . This operator is the inverse of $[[A_n]]$ if κ is a constant but is only an approximation for a variable function κ . Its application has a cost of $\mathcal{O}(J \log J)$ operations and the memory cost if $\mathcal{O}(J)$. As it will be seen later during the numerical examples, the use of $[[B_n]]$ may strongly accelerate the convergence rate. Concretely, for the one-dimensional case and the cosine transform (Neumann boundary condition), the Matlab procedure is simply

$$v = \text{idct}(1./(1/\text{deltan} + \text{kappa}.*\text{epsilon}). * \text{lambda}.\text{dct}(\mathbf{w})); \tag{28}$$

where kappa is encoded as a vector representing the diagonal matrix. In the case of a constant parameter κ , the GMRES equipped with this preconditioner solves the linear system in one iteration (as a direct solver would do). A useful alternative preconditioner is $[[B_n^m]]$, which is based on the mean $\kappa^m = J^{-1} \sum_{j=0}^{J-1} \kappa_j$. In practice, $[[B_n^m]]$ is much more robust than $[[B_n]]$.

4 COMPUTATIONAL ASPECTS AND NUMERICAL EXAMPLE

4.1 Computational Aspects

In the following, we call E and CN the semi-implicit Euler and Crank-Nicolson schemes Eqs 16, 20, respectively, without any preconditioner for the GMRES. When the preconditioner $[[B_n^m]]$

is used in the GMRES method, the algorithms are called PE and PCN. The GMRES is considered without restarting and with a tolerance $\epsilon_2 = 10^{-12}$. For each iteration n of the minimization method, the number of iterations of the GMRES is then called $N_{\alpha/2,J}^{m,n}$ ($m = E, CN, PE, \text{ or } PCN$) accordingly to the method, fractional order $\alpha/2$ and number of spatial discretization points J . The total computational cost at iteration n is essentially related to the number of Matrix-Vector (MV) operations based on $[[A_n]]$, and is twice when the preconditioner $[[B_n^m]]$ (or $[[B_n]]$) is applied since it requires an additional function call. Concerning the minimization algorithms, the stopping criterion Eq. 12 is fixed through $\epsilon_1 = 10^{-12}$ and used with the linear line search algorithm. If we call $\bar{N}_{\alpha/2,J}^{m,n}$ the total number of MV products for the whole procedure and for a method m at step n , then we have

$$\bar{N}_{\alpha/2,J}^{m,n} := \sum_{\ell=1}^n N_{\alpha/2,J}^{m,\ell}, \quad \bar{N}_{\alpha/2,J}^m := \bar{N}_{\alpha/2,J}^{m,n} = \sum_{n=1}^{n_{\alpha/2,J}^m} N_{\alpha/2,J}^{m,n}$$

where $n_{\alpha/2,J}^m$ is the iteration number n to reach the tolerance in the minimization algorithm for the method m . As a consequence, the algorithm has a computational cost of $\mathcal{O}(\bar{N}_{\alpha/2,J}^m J \log J)$ operations and needs a storage of $\mathcal{O}(J)$ coefficients. In the case of a higher-dimensional problem, J denotes the total number of unknowns on the d -grid.

Let us introduce the discrete L^2 -norm as: $\forall \mathbf{v} \in \mathbb{R}^J, \|\mathbf{v}\|_0 := h^{1/2} \|\mathbf{v}\|$, with the associated discrete inner product: $(u, v)_0 := h \mathbf{u} \cdot \mathbf{v}$. Then, the converged discrete energy at level J is given by

$$E_{\alpha/2,J}^{(n_{\alpha/2,J}^m)} := E_{\alpha/2,J}(\mathbf{u}_{\alpha/2,J}^{(n_{\alpha/2,J}^m)}) := \frac{\epsilon}{2} \|[[\nabla^{\alpha/2}]] \mathbf{u}_{\alpha/2,J}^{(n_{\alpha/2,J}^m)}\|_0^2 + \left(F(\mathbf{u}_{\alpha/2,J}^{(n_{\alpha/2,J}^m)}), \mathbf{1} \right)_0$$

The converged solution for a method m , the fractional PDE of order $\alpha/2$, a number of spatial grid points J and a number of iterations $n_{\alpha/2,J}^m$ is called $\mathbf{u}_{\alpha/2,J}^{(n_{\alpha/2,J}^m)}$. Finally, the residual energy of a method m at iteration n for a fractional order $\alpha/2$ and spatial discretization J is given by

$$\Delta E_{\alpha/2,J}^{m,n} := |E_{\alpha/2,J}(\mathbf{u}^{m,(n)}) - E_{\alpha/2,J}(\mathbf{u}^{m,(n-1)})|. \tag{29}$$

Therefore, representing $\Delta E_{\alpha/2,J}^{m,n}$ vs. $\bar{N}_{\alpha/2,J}^{m,n}$ provides the correct way of measuring the convergence speed.

4.2 A One-Dimensional Example

To illustrate the methods, we consider an example in dimension one. The computational domain is $\Omega = -1, 1$. The initial data is given by

$$u_0(x) = 0.5 \sin(3\pi x/2) (\cos(\pi x) - 1).$$

The parameter ϵ is fixed to 10^{-2} and β is the function

$$\beta(x) := 0.6 + 0.4 \cos(20\pi x).$$

Concerning the variable function κ , we consider

$$\kappa(x) := 1 + 0.5 \cos(10\pi x). \tag{30}$$

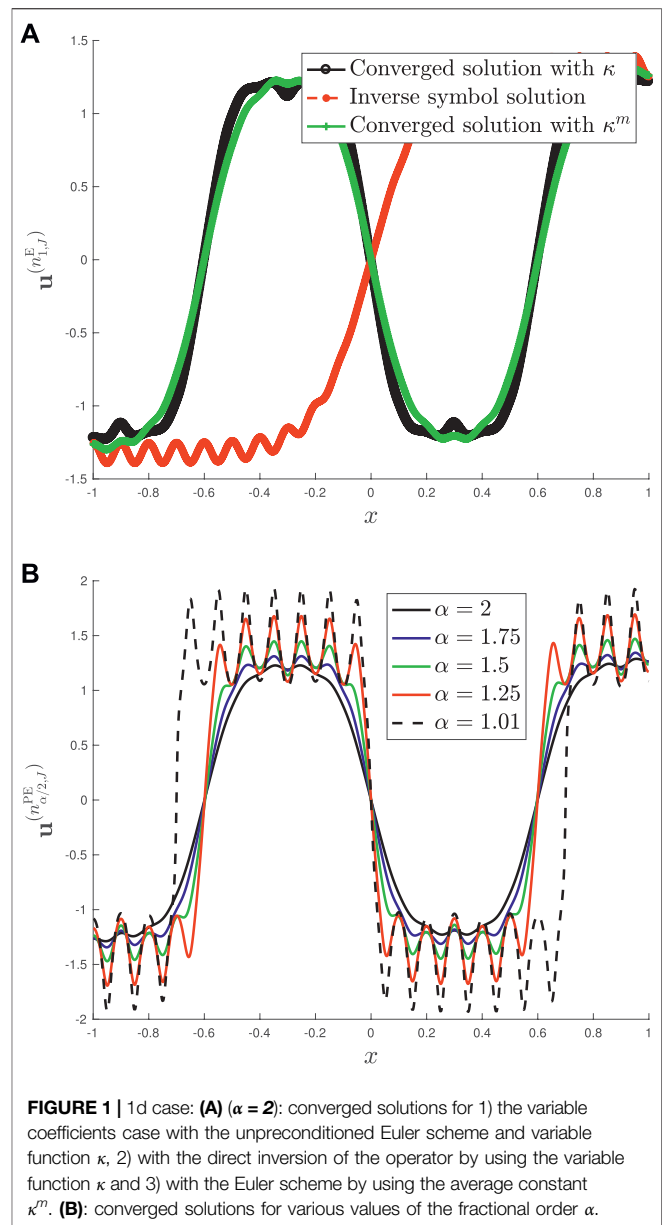
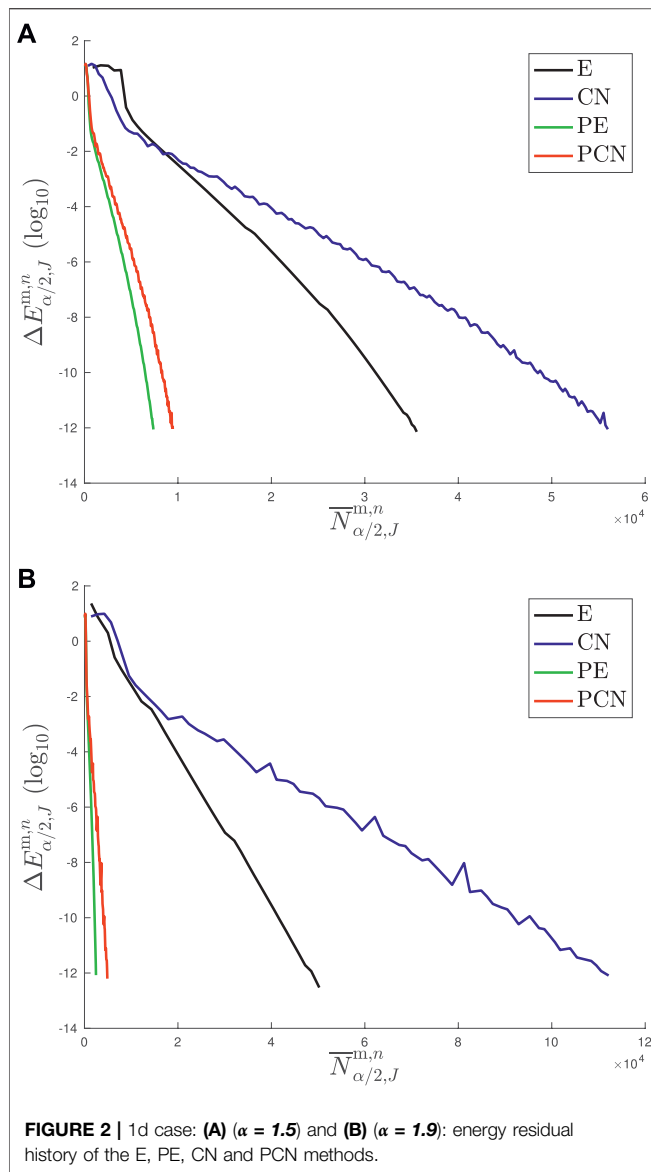


FIGURE 1 | 1d case: **(A)** ($\alpha = 2$): converged solutions for 1) the variable coefficients case with the unpreconditioned Euler scheme and variable function κ , 2) with the direct inversion of the operator by using the variable function κ and 3) with the Euler scheme by using the average constant κ^m . **(B)**: converged solutions for various values of the fractional order α .

Therefore, $\kappa^m = 1$ while κ oscillates. The number of grid points is set to $J = 2^p$, for $p \in \mathbb{N}$, leading to $h = 2^{-p+1}$.

Let us first report on Figure 1 (left) the computed solutions $\mathbf{u}_{\alpha/2,J}^{(n_{\alpha/2,J}^E)}$ when considering κ and $\bar{\kappa}$ based on the Euler formulation Eq. 16, and the solution obtained by the direct inversion of the operator with the inverse symbol of a , i.e. a^{-1} , and the variable function κ . Here, the situation considers the case $\alpha = 2$ which is the integer order CA equation. The number of grid points is $J = 2^{10}$. The solution based on the inverted symbol does not provide a spatially converging solution, which is expected from the previous explanations. As seen on the figure, considering κ^m is not enough to get the true solution, but it provides an idea of the solution. This means that a preconditioner based on κ^m is expected to be robust when included in the GMRES. We now report the converged solution for various fractional orders $\alpha = 2, 1.75,$



1.5, 1.25, 1.01 on **Figure 1** (right). The effect of the oscillating function κ can be clearly observed, and the solution tends to be more singular as α goes to 1. The respective converged energies are: 76.56, 64.90, 53.01, 38.64 and 25.18.

We analyze now the behavior of the global iterative method on **Figure 2**. The spatial discretization parameter is $J = 2^{10}$ for the same function cases as before, but with $\alpha = 1.5$ and $\alpha = 1.9$. On **Figure 2**, we report the residual history for the E, PE, CN and PCN methods, the preconditioner being based on the inverse of the operator with average symbol κ^m . As it can be seen, the semi-implicit Euler iterative scheme converges faster than the CN method, while preconditioning helps a lot for both methods by dividing the number of iterations by a factor 5–6 ($\alpha = 1.5$) to 20 ($\alpha = 1.9$). When α is close to 1, the preconditioner effect is less visible

while it is more important for α close to 2. This can be understood by the fact that the condition number is then larger as α increases, leading to a need for preconditioning $(-\Delta)^{\alpha/2}$. In addition, the preconditioner helps a lot for the convergence when refining the spatial discretization thanks to J , i.e. to get a higher accuracy. As a conclusion, and from other simulations in different configurations (not reported here for the sake of conciseness), it appears that the PE method always provides the best convergence rate.

5 CONCLUSION

In this paper, we introduced some full discretizations of the gradient flow formulation of space fractional Cahn-Allen equations with variable coefficients in space. To this end, we use an Euler or Crank-Nicolson time discretization of the gradient flow formulation, with adaptive time stepping. In addition, a Fourier-based pseudo-spectral discretization scheme in space allows for an efficient and robust computation of each time step, in conjunction with a GMRES solver accelerated by a spectral preconditioner. Some simulations in dimension one show that the method based on the Euler scheme with the spectral preconditioner performs the best. This conclusion should extend to 2D/3D problems since all the ingredients here can be adapted and higher-dimensional grids are cartesian extensions of the one-dimensional case. In some future works, we will address more numerical examples in 2D and an extension of the method to the fractional Cahn-Hilliard equation [30, 31] that is used for example in [32] as fractional inpainting model.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/Supplementary Materials, further inquiries can be directed to the corresponding authors.

AUTHOR CONTRIBUTIONS

SA and CC developed the theoretical formalism, performed the analytic calculations and performed the numerical simulations. Authors contributed equally to the final version of the manuscript.

FUNDING

The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grand code 18-SCI-1-01-0017.

REFERENCES

1. Karniadakis GE, Hesthaven JS, Podlubny I. Special Issue on "Fractional PDEs: Theory, Numerics, and Applications". *J Comput Phys* (2015) 293:1–3. doi:10.1016/j.jcp.2015.04.007
2. Podlubny I. *Fractional Differential Equations*. San Diego: Academic Press (1999).
3. Baleanu D, Diethelm K, Scalas E, Trujillo JJ. *Fractional Calculus, Models and Numerical Methods*. Series on Complexity, Nonlinearity and Chaos, 3. Boston, Mass: World Scientific (2012).
4. Herrmann R. *Fractional Calculus, An Introduction for Physicists*. GigaHedron, Germany: World Scientific (2011).
5. Metzler R, Klafter J. The Random Walk's Guide to Anomalous Diffusion: a Fractional Dynamics Approach. *Phys Rep* (2000) 339(1):1–77. doi:10.1016/s0370-1573(00)00070-3
6. Carreras BA, Lynch VE, Zaslavsky GM. Anomalous Diffusion and Exit Time Distribution of Particle Tracers in Plasma Turbulence Model. *Phys Plasmas* (2001) 8:5096–103. doi:10.1063/1.1416180
7. Shlesinger MF, West BJ, Klafter J. Lévy Dynamics of Enhanced Diffusion: Application to Turbulence. *Phys Rev Lett* (1987) 58:1100–3. doi:10.1103/physrevlett.58.1100
8. Zaslavsky GM, Stevens D, Weitzner H. Self-similar Transport in Incomplete Chaos. *Phys Rev E* (1993) 48:1683–94. doi:10.1103/physreve.48.1683
9. Meerschaert MM, Benson DA, Wheatcraft SW. Subordinated Advection-Dispersion Equation for Contaminant Transport. *Water Resour Res* (2001) 37:1543–50. doi:10.1029/2000WR900409
10. Benson DA, Wheatcraft SW, Meerschaert MM. The Fractional-Order Governing Equation of Lévy Motion. *Water Resour Res* (2000) 36:1413–23. doi:10.1029/2000wr900032
11. Scalas E, Gorenflo R, Mainardi F. Fractional Calculus and Continuous-Time Finance. *Phys A: Stat Mech Appl* (2000) 284:376–84. doi:10.1016/s0378-4371(00)00255-7
12. Yin B, Liu Y, Li H, He S. Fast Algorithm Based on TT-M FE System for Space Fractional Allen-Cahn Equations with Smooth and Non-smooth Solutions. *J Comput Phys* (2019) 379:351–72. doi:10.1016/j.jcp.2018.12.004
13. Ullah MS, Harun-Or-RoshidAli MZ, Noor NFM. Novel Dynamics of Wave Solutions for Cahn-Allen and Diffusive Predator-Prey Models Using MSE Scheme. *Partial Differ Equations Appl Maths* (2021) 3:100017. doi:10.1016/j.padiff.2020.100017
14. Allen SM, Cahn JW. A Microscopic Theory for Antiphase Boundary Motion and its Application to Antiphase Domain Coarsening. *Acta Metal* (1979) 27:1085–95. doi:10.1016/0001-6160(79)90196-2
15. Algahtani OJJ. Comparing the Atangana-Baleanu and Caputo-Fabrizio Derivative with Fractional Order: Allen Cahn Model. *Chaos Solitons Fractals* (2016) 89:552–9. doi:10.1016/j.chaos.2016.03.026
16. Nec Y, Nepomnyashchy AA, Golovin AA. Front-type Solutions of Fractional Allen-Cahn Equation. *Phys D: Nonlinear Phenomena* (2008) 237(24):3237–51. doi:10.1016/j.physd.2008.08.002
17. Lee S, Lee D. The Fractional Allen-Cahn Equation with the Sextic Potential. *Appl Maths Comput* (2019) 351:176–92. doi:10.1016/j.amc.2019.01.037
18. Benes M, Chalupecky V, Mikula K. Geometrical Image Segmentation by the Allen-Cahn Equation. *Appl Numer Maths* (2004) 51(2-3):187–205.
19. Lee D, Lee S. Image Segmentation Based on Modified Fractional Allen-Cahn Equation. *Math Probl Eng* (2019) 2019:3980181. doi:10.1155/2019/3980181
20. Zhai S, Gui D, Zhao J, Feng X. High Accuracy Spectral Method for the Space-Fractional Diffusion Equations. *JMS* (2014) 47(3):274–86. doi:10.4208/jms.v47n3.14.03
21. Bueno-Orovio A, Kay D, Burrage K. Fourier Spectral Methods for Fractional-In-Space Reaction-Diffusion Equations. *Bit Numer Math* (2014) 54(4):937–54. doi:10.1007/s10543-014-0484-2
22. Zhai S, Weng Z, Feng X. Fast Explicit Operator Splitting Method and Time-step Adaptivity for Fractional Non-local Allen-Cahn Model. *Appl Math Model* (2016) 40(2):1315–24. doi:10.1016/j.apm.2015.07.021
23. Alzahrani SS, Khaliq AQM. Fourier Spectral Exponential Time Differencing Methods for Multi-Dimensional Space-Fractional Reaction-Diffusion Equations. *J Comput Appl Maths* (2019) 361:157–75. doi:10.1016/j.cam.2019.04.001
24. Lischke A, Pang G, Gulian M, Song F, Glusa C, Zheng X, et al. What Is the Fractional Laplacian? A Comparative Review with New Results. *J Comput Phys* (2020) 404:109009. doi:10.1016/j.jcp.2019.109009
25. Choi J-W, Lee HG, Jeong D, Kim J. An Unconditionally Gradient Stable Numerical Method for Solving the Allen-Cahn Equation. *Physica A: Stat Mech its Appl* (2009) 388:1791–803. doi:10.1016/j.physa.2009.01.026
26. Dai S, Du Q. Motion of Interfaces Governed by the Cahn–Hilliard Equation with Highly Disparate Diffusion Mobility. *SIAM J Appl Math* (2012) 72(6):1818–41. doi:10.1137/120862582
27. Taylor ME. *Pseudodifferential Operators*. Princeton, NJ: Princeton University Press (1981).
28. Saad Y. *Iterative Methods for Sparse Linear Systems*. Boston: PWS Pub. Co. (1996).
29. Saad Y, Schultz MH. GMRES: a Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J Sci Stat Comput* (1986) 7(3):856–69. doi:10.1137/0907058
30. Ainsworth M, Mao Z. Analysis and Approximation of a Fractional Cahn–Hilliard Equation. *SIAM J Numer Anal* (2017) 55(4):1689–718. doi:10.1137/16m1075302
31. Weng Z, Zhai S, Feng X. A Fourier Spectral Method for Fractional-In-Space Cahn–Hilliard Equation. *Appl Math Model* (2017) 42:462–77. doi:10.1016/j.apm.2016.10.035
32. Bosch J, Stoll M. A Fractional Inpainting Model Based on the Vector-Valued Cahn–Hilliard Equation. *SIAM J Imaging Sci* (2015) 8:2352–82. doi:10.1137/15m101405x

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Alzahrani and Chokri. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.