



HAL
open science

Particle simulation of space–fractional diffusion equations

Marco Lucchesi, Samer Allouch, Olivier Le Maitre, Kassem Mustapha, Omar Knio

► **To cite this version:**

Marco Lucchesi, Samer Allouch, Olivier Le Maitre, Kassem Mustapha, Omar Knio. Particle simulation of space–fractional diffusion equations. *Computational Particle Mechanics*, 2020, 7 (3), pp.491-507. 10.1007/s40571-019-00275-8 . hal-02327700

HAL Id: hal-02327700

<https://hal.science/hal-02327700v1>

Submitted on 18 Nov 2020

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

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

Particle Simulation of Space-Fractional Diffusion Equations

M. Lucchesi^a, S. Allouch^a, O.P. Le Maître^b, K.A. Mustapha^c, O.M. Knio^a

^aKing Abdullah University of Science and Technology, Thuwal, Saudi Arabia

^bCMAP, CNRS, INRIA, École Polytechnique, Institut Polytechnique de Paris, 91128 Palaiseau, France

^cKing Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia

Abstract

This work explores different particle-based approaches to the simulation of space-fractional diffusion equations in unbounded domains. We rely on smooth particle approximations, and consider five different methods for estimating the fractional diffusion term. The first method is based on a direct differentiation of the particle representation, following the Riesz definition of the fractional derivative, and results in a non-conservative scheme. Three methods follow the particles strength exchange (PSE) methodology and are by construction conservative, meaning that the total particle strength is time-invariant. The first PSE algorithm estimates the fractional diffusion flux using direct differentiation, and uses an integral representation of the divergence operator. The second one relies on the integral representation of the fractional Laplacian to derive a suitable particle strength exchange formula for the diffusion term. The third PSE construction employs the Green's function of the fractional diffusion equation. A fifth method is developed based on the diffusion-velocity approach, where the diffusion term is transformed into a transport term. The performance of all five methods is assessed, for which analytical solutions are known. A detailed analysis is conducted of the various sources of error, namely filtering, quadrature, domain truncation, and time integration. Computational experiments are used to gain insight for the generalization of the present constructions, such as applications in bounded domains or variable diffusivity.

Keywords: fractional diffusion, smooth particle approximation, particle strength exchange, diffusion velocity

1. Introduction

Many transport problems of relevance to industrial and environmental applications involve anomalous diffusion phenomena [1]. A canonical example concerns the well-known problem of the space-fractional diffusion equation, which in its simplest form considers a fixed, uniform in space diffusivity coefficient and fixed fractional order derivatives.

Numerous studies have focused on the numerical solution of this problem, using finite-difference or variational methodologies; see e.g. [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and related references. In contrast to grid-based and Galerkin formulations, works with particle-based methods are scarce, and appear to be limited to random walk type approaches; see e.g. [14, 15, 16, 17, 18, 19].

This work explores the application of smooth particle approximations [20, 21, 22, 23, 24] to space-fractional diffusion models. It is specifically motivated by the desire to capitalize on the advantages offered by particle schemes, particularly for modeling fractional transport in geophysical flows. These advantages include the ability of developing high-order schemes that concentrate computational resources in regions of high gradients, accommodate advection-dominated transport problems with minimal numerical diffusion, deal with situations involving unbounded domains, and naturally accommodate far-field boundary conditions [23].

Email addresses: marco.lucchesi@kaust.edu.sa (M. Lucchesi), samer.allouch@kaust.edu.sa (S. Allouch), olivier.le-maitre@polytechnique.edu (O.P. Le Maître), kassem@kfupm.edu.sa (K.A. Mustapha), Omar.Knio@kaust.edu.sa (O.M. Knio)

In case of constant diffusivity, the space-fractional diffusion equation is associated with a fractional Laplacian operator [25, 26, 27]. In [27] the fractional Laplacian in multiple dimensions is extensively reviewed, and we will adopt the definitions given therein.

In this work, we focus on deriving different smooth particle constructions for solving space fractional diffusion problems in unbounded domains and examining their properties. Starting from the conservation law in divergence form, we provide alternative expressions of the fractional diffusion flux, and further restrict ourselves to the case of a constant in space diffusivity, which enables us to exploit available expressions of the fundamental solution [28]. Different particle schemes are then developed in Section 3. Specifically, we introduce a smooth particle representation of the solution, and use this representation to derive expressions of the fractional diffusion flux. Based on this development, we explore three routes for constructing particle schemes. In the first route, the particle representation is directly differentiated, resulting in a non-conservative approach. To enforce conservation properties, the particle strength exchange (PSE) formalism is considered in the second route. Following this formalism, we develop three alternative PSE formulations. The first PSE formulation is constructed by approximating the divergence of the fractional diffusion flux; this approach is referred to as Flux-PSE (FPSE). Another alternative is explored in which the PSE approximation is applied to the fractional Laplacian operator, leading to what we refer to as the Kernel-PSE (KPSE) method. Third, we propose a PSE variant based on the Green's function of the problem, therefore coining it the GPSE scheme. In the third route, the diffusion operator is rewritten as an advection term, involving a non-uniform particle velocity that is directly linked to the diffusion flux. This leads to a diffusion-velocity method (DVM) [29]. Unlike the other proposed schemes, DVM causes a deformation of the Lagrangian grid even in the absence of a convection field. To guard against the potentially detrimental impact of Lagrangian grid deformation, the implementation of DVM involves a particle redistribution algorithm. In Sections 4 and 5, we analyze the performance of the different constructions in one and two space dimensions, and assess their relative merits. In particular, a computational study is conducted for the impact of discretization and truncation parameters. Major conclusions are summarized in Section 6.

2. The fractional Laplacian

In this section we briefly summarize the definition of the fractional gradient and fractional Laplacian operators. Unless otherwise indicated, all the fundamental concepts and notations are adapted from Ref. [27].

We focus on fractional diffusion equations governed by the conservation law:

$$\frac{\partial u}{\partial t} = -\nabla \cdot \mathbf{Q}^\beta, \quad (1)$$

where \mathbf{Q}^β is the n -dimensional fractional diffusion flux of order $0 < \beta \equiv \alpha - 1 < 1$,

$$\mathbf{Q}^\beta(\mathbf{x}, t) \equiv -\mathcal{D}\nabla^\beta u(\mathbf{x}, t) = -\mathcal{D} \frac{2^{\alpha-1}\Gamma\left(\frac{n+\alpha}{2}\right)}{\pi^{\frac{n}{2}}\Gamma\left(\frac{2-\alpha}{2}\right)} \int_{\mathbb{R}^n} \text{pv} \frac{\mathbf{y}}{|\mathbf{y}|^{\alpha+n}} u(\mathbf{x} + \mathbf{y}, t) dV(\mathbf{y}), \quad (2)$$

where $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $n \geq 1$, $dV(\mathbf{y})$ is the measure in \mathbb{R}^n , \mathcal{D} is the diffusivity, ∇^β denotes the fractional gradient operator of order β , whereas ‘‘pv’’ indicates that the integral should be intended as a principal value integral.

Assuming $\mathcal{D} \equiv 1$ in the following, Eq. (1) reduces to the fractional diffusion equation

$$\frac{\partial u}{\partial t} = \Delta^\alpha u, \quad (3)$$

where $\Delta^\alpha u$ is the fractional Laplacian operator of order α ,

$$\Delta^\alpha u = \frac{\alpha 2^{\alpha-1}\Gamma\left(\frac{n+\alpha}{2}\right)}{\pi^{\frac{n}{2}}\Gamma\left(\frac{2-\alpha}{2}\right)} \int_{\mathbb{R}^n} \text{pv} \frac{u(\mathbf{y}) - u(\mathbf{x})}{|\mathbf{y} - \mathbf{x}|^{n+\alpha}} dV(\mathbf{y}). \quad (4)$$

The fractional Laplacian can alternatively be expressed as a function of the Riesz potential $\mathcal{R}_{n,\alpha}(u, \mathbf{x})$

$$\Delta^\alpha u = -\mathcal{R}_{n,-\alpha}(u, \mathbf{x}) = \Delta \mathcal{R}_{n,2-\alpha}(u, \mathbf{x}). \quad (5)$$

Alternative notations found in the literature for the fractional Laplacian operator are $-(-\Delta)^{\alpha/2}$, which is useful when discussing its spectral properties, and ∇^α , as indicated in [27].

The fractional diffusion equation (3) has a fundamental solution available for the general n -dimensional case. In the one-dimensional case, closed form expressions are available [28], as summarized in Appendix Appendix A. We will take advantage of these expressions to assess the performance of the various approaches considered.

We shall focus on the numerical solution of Eq. (3), and results are presented in one and two dimensions.

3. Particle approximation

In the framework of smooth particle methods, the n -dimensional spatial domain is discretized using a finite collection of particles, defined by their positions, \mathbf{x}_i , and volumes, V_i . We start from a regularization of a generic function $u(\mathbf{x})$ as a convolution with a kernel $\eta_\epsilon(\mathbf{x})$:

$$u(\mathbf{x}) \approx u_\epsilon(\mathbf{x}) \equiv \int_{\mathbb{R}^n} u(\mathbf{y})\eta_\epsilon(\mathbf{x} - \mathbf{y})dV(\mathbf{y}), \quad (6)$$

The particle representation of a function, $u(\mathbf{x})$, is expressed as the discretization of Eq. (6) [20, 21, 22, 23, 24]:

$$u(\mathbf{x}) = \sum_{i \in \mathcal{I}} V_i u_i \eta_\epsilon(\mathbf{x} - \mathbf{x}_i), \quad \eta_\epsilon(\mathbf{x}) = \frac{1}{\epsilon^n} \eta\left(\frac{\mathbf{x}}{\epsilon}\right), \quad (7)$$

where u_i is the strength of i -th particle, $\mathcal{I} \subseteq \mathbb{Z}$ is the particles index set, η is a radial kernel of unit mass, and ϵ is the smoothing parameter. Clearly, η_ϵ tends to the Dirac measure in the sense of distributions as $\epsilon \rightarrow 0$. In this work we mostly rely on the classical second-order, squared exponential kernel [30, 31]

$$\eta(\mathbf{x}) = \frac{1}{\pi^{\frac{n}{2}}} \exp(-|\mathbf{x}|^2). \quad (8)$$

Alternatively, a fourth-order kernel will sometimes be adopted [30, 31]:

$$\eta(\mathbf{x}) = \frac{2}{\pi^{\frac{n}{2}}} \exp(-|\mathbf{x}|^2) - \frac{1}{(2\pi)^{\frac{n}{2}}} \exp(-|\mathbf{x}|^2/2). \quad (9)$$

3.1. Direct Differentiation

As implied by its name, in the direct differentiation (DD) approach one directly differentiates the particle representation of u , and then uses a collocation (midpoint rule) approximation to estimate the time derivative of the particle strength [24]. As explained in Appendix Appendix B, the particle representation of the fractional Laplacian can be written in n space dimensions, leading to the following particle scheme:

$$\frac{\partial u_i}{\partial t} = \frac{1}{\epsilon^\alpha} \sum_{j \in \mathcal{I}} V_j u_j G_\epsilon(\mathbf{x}_i - \mathbf{x}_j), \quad (10)$$

$$G_\epsilon(\mathbf{r}) = \frac{1}{\epsilon^n} G\left(\frac{\mathbf{r}}{\epsilon}\right), \quad G(\mathbf{r}) = -\Gamma_{n,0}(\alpha) {}_1F_1\left(\frac{\alpha+n}{2}, \frac{n}{2}, -|\mathbf{r}|^2\right), \quad (11)$$

where ${}_1F_1(a,b,c)$ denotes the confluent hypergeometric function of the first kind [32], and $\Gamma_{n,b}(\alpha)$ is defined in Eq. (B.14). For $1 < \alpha < 2$, $G(\mathbf{r})$ is a radial function, with negative value at $\mathbf{r} = \mathbf{0}$, a global positive maximum for $1 < |\mathbf{r}| < 2$, and an asymptotic decay $G(\mathbf{r}) \propto |\mathbf{r}|^{-\alpha-n}$ as $|\mathbf{r}| \rightarrow \infty$. For $n = 1$, it is similar to Eq. (1.9.19) in Ref. [27] up to the scaling factor $\pi^{-1/2}\epsilon^{-\alpha}$, and graphical representations are available therein.

For the fourth-order kernel in Eq. (9), the function $G(\mathbf{r})$ becomes

$$G(\mathbf{r}) = -\Gamma_{n,0}(\alpha) \left[2 {}_1F_1\left(\frac{\alpha+n}{2}, \frac{n}{2}, -|\mathbf{r}|^2\right) - \frac{1}{2^{\frac{\alpha+n}{2}}} {}_1F_1\left(\frac{\alpha+n}{2}, \frac{n}{2}, -\frac{|\mathbf{r}|^2}{2}\right) \right]. \quad (12)$$

Note that, similar to the case of Fickian diffusion [24], the DD approach leads to a non-conservative scheme.

3.2. Particle Strength Exchange

This section explores three different approaches to construct a conservative particle scheme based on the PSE formalism [31, 33, 34, 35]. Briefly, PSE methods consist in conservative approximations of various differential operators for the particle representation of any field u . In principle, one may compute $\Delta^\alpha u$ as PSE approximation of Δ applied to the particle representation of $\mathcal{R}_{n,2-\alpha}(u, \mathbf{x})$ (from Eq. (5)), leading to a method called Direct-PSE, or forge a PSE-like approximation of Δ^α and apply it to the particle representation of u , exploiting the particular form of Eq. (4) and leading to the Kernel-PSE method, or apply a PSE approximation of the divergence to the particle representation of $\nabla^\beta u$, leading to the Flux-PSE method.

3.2.1. Direct-Particle Strength Exchange

Discretizing the convolution in Eq. (B.15), the second-order particle representation of $\mathcal{R}_{n,2-\alpha}(u, \mathbf{x})$ is

$$v(\mathbf{x}) = \mathcal{R}_{n,2-\alpha}(u, \mathbf{x}) = \frac{1}{\epsilon^{\beta-1}} \sum_{i \in \mathcal{I}} V_i u_i \kappa_\epsilon^\beta(\mathbf{x} - \mathbf{x}_i) \quad (13)$$

with

$$\kappa_\epsilon^\beta(\mathbf{r}) = \frac{1}{\epsilon^n} \kappa^\beta\left(\frac{\mathbf{r}}{\epsilon}\right), \quad \kappa^\beta(\mathbf{r}) = \Gamma_{n,0}(\beta-1) {}_1F_1\left(\frac{\beta-1+n}{2}, \frac{n}{2}, -|\mathbf{r}|^2\right). \quad (14)$$

Figure 1 shows in the left plot κ^β for $n=1$ and different values of β . It can be seen that it is a monotonically decaying kernel with vanishing derivative at $|\mathbf{r}|=0$, and that $\kappa^\beta(\mathbf{r}) \propto |\mathbf{r}|^{1-n-\beta}$ as $|\mathbf{r}| \rightarrow \infty$ from Eq.(14).

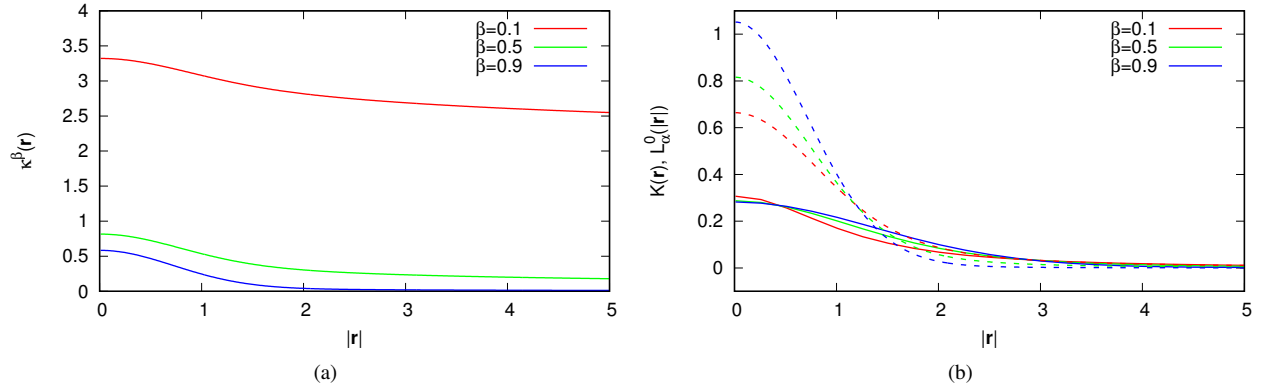


Figure 1: $\kappa_\beta(\mathbf{r})$ from Eq. (14) (a), $L_\alpha^0(r)$ from Eq. (A.1), and $K(\mathbf{r})$ from Eq. (31) (b, solid and dashed line respectively) for three different values of β and $n=1$.

If the fourth-order kernel (9) is used for the particle approximation, one obtains:

$$\kappa^\beta(\mathbf{r}) = \Gamma_{n,0}(\beta-1) \left[2 {}_1F_1\left(\frac{\beta-1+n}{2}, \frac{n}{2}, -|\mathbf{r}|^2\right) - \frac{1}{2^{\frac{\beta-1+n}{2}}} {}_1F_1\left(\frac{\beta-1+n}{2}, \frac{n}{2}, -\frac{|\mathbf{r}|^2}{2}\right) \right]. \quad (15)$$

Our first PSE algorithm thus consists in (i) interpolating $v(\mathbf{x}, t)$ on the same particle grid used to represent u , and (ii) applying the well-known second order PSE approximation of the ordinary Laplacian [31, 33, 34, 35] using the exponentially decaying kernel, η , in (8). This results in the Direct-PSE scheme (DPSE)

$$\frac{\partial u_i}{\partial t} = \frac{2}{\epsilon^2} \sum_{j \in \mathcal{I}} V_j (v_j - v_i) \Phi_\epsilon(\mathbf{x}_j - \mathbf{x}_i) : \quad (16)$$

where

$$\Phi_\epsilon(\mathbf{r}) = \frac{1}{\epsilon^n} \Phi\left(\frac{\mathbf{r}}{\epsilon}\right), \quad \Phi(\mathbf{r}) = -\frac{1}{|\mathbf{r}|} \frac{d\eta}{dr} = \frac{2}{\pi^{\frac{n}{2}}} \exp(-|\mathbf{r}|^2). \quad (17)$$

Note that, because Φ_ϵ is symmetric, the method is conservative in the sense that the total particle strength is time-invariant, i.e.

$$\frac{d}{dt} \left[\sum_{i \in \mathcal{I}} u_i V_i \right] = 0. \quad (18)$$

Note that the method requires two passes over the set of particles, namely for the computation of the function v and then for its differentiation. Although not shown, numerical experiments with DPSE exhibited large approximation errors at the edges of the particle grid, due to the slow decay of $\kappa^\beta(\mathbf{r})$; specifically $\kappa^\beta(\mathbf{r}) \sim |\mathbf{r}|^{1-n-\beta}$ as $|\mathbf{r}| \rightarrow \infty$. This problem could be tackled by considering a different particle grid to represent v , or by seeking an alternative (potentially stretched) kernel that would be more adapted to its distribution. Such approaches are outside the scope of the present work, and are left for a future study.

3.2.2. Flux-Particle Strength Exchange

A possible way to mitigate the issue above is to interpolate the fractional diffusion flux on the particle grid, and then apply a symmetrized formula for computing its divergence. Using the particle representation for $\nabla^\beta u$ shown in Appendix Appendix B, the fractional flux is

$$\mathbf{Q}^\beta(x) = -\nabla^\beta u = -\frac{1}{\epsilon^\beta} \sum_{j \in \mathcal{I}} V_j u_j \mathbf{F}_\epsilon(\mathbf{x} - \mathbf{x}_j), \quad (19)$$

$$\mathbf{F}_\epsilon(\mathbf{r}) = \frac{1}{\epsilon^n} \mathbf{F}\left(\frac{\mathbf{r}}{\epsilon}\right), \quad \mathbf{F}(\mathbf{r}) = -\Gamma_{n,2}(\alpha) {}_1F_1\left(\frac{n+\alpha}{2}, \frac{n+2}{2}, -|\mathbf{r}|^2\right) \mathbf{r}. \quad (20)$$

It is seen (Fig. 3.1.2(a) in [27]) that for $n = 1$ \mathbf{F} is an odd function, has a single minimum at $|\mathbf{r}| \neq 0$, and that it increases slowly to 0 as $\mathbf{F}(\mathbf{r}) \propto |\mathbf{r}|^{-\beta-n}$.

If the fourth-order kernel (9) is adopted, $\mathbf{F}(\mathbf{r})$ becomes

$$\mathbf{F}(\mathbf{r}) = -\Gamma_{n,2}(\alpha) \left[2 {}_1F_1\left(\frac{\alpha+n}{2}, \frac{n+2}{2}, -|\mathbf{r}|^2\right) - \frac{1}{2^{\frac{\alpha+n}{2}}} {}_1F_1\left(\frac{\alpha+n}{2}, \frac{n+2}{2}, -\frac{|\mathbf{r}|^2}{2}\right) \right] \mathbf{r}. \quad (21)$$

We may now approximate the divergence of \mathbf{Q}^β using a conservative PSE form of the divergence [31]

$$\frac{\partial \mathbf{Q}^\beta}{\partial x}(\mathbf{x}_i) = \frac{1}{\epsilon} \sum_{j \in \mathcal{I}} V_j (\mathbf{Q}_j^\beta + \mathbf{Q}_i^\beta) \eta_\epsilon^1(\mathbf{x}_i - \mathbf{x}_j), \quad (22)$$

where $\mathbf{Q}_i^\beta = \mathbf{Q}^\beta(\mathbf{x}_i)$. In the computations, we rely on the second-order kernel [31],

$$\eta_\epsilon^1(\mathbf{r}) = \frac{1}{\epsilon^n} \eta^1\left(\frac{\mathbf{r}}{\epsilon}\right) \quad \eta^1(\mathbf{r}) = -\frac{2\mathbf{r}}{\pi^{\frac{n}{2}}} \exp(-|\mathbf{r}|^2), \quad (23)$$

or on the fourth-order variant [31],

$$\eta^1(\mathbf{r}) = -\frac{(5-2|\mathbf{r}|^2)\mathbf{r}}{\pi^{\frac{n}{2}}} \exp(-|\mathbf{r}|^2), \quad (24)$$

to estimate the flux divergence in (25). Substituting into the governing equations results in the following particle scheme, that we will refer to as FPSE:

$$\frac{\partial u_i}{\partial t} = -\frac{1}{\epsilon} \sum_{j \in \mathcal{I}} V_j (\mathbf{Q}_j^\beta + \mathbf{Q}_i^\beta) \eta_\epsilon^1(\mathbf{x}_i - \mathbf{x}_j). \quad (25)$$

As in the previous scheme (16), we observe that the FPSE approach requires two passes over the particles set, namely to first estimate the flux and then its divergence.

3.2.3. Kernel-Particle Strength Exchange

The integral form of the fractional Laplacian (Eq. (4)) suggests several ways to construct PSE schemes acting directly on the particle representation of u .

We initially considered injecting a regularized version, u_ϵ , of the solution, u , as in Eq. (6), into Eq.(4). Specifically, using the identity

$$\int_{\mathbb{R}^n} \eta_\epsilon(\mathbf{y} - \mathbf{z}) dV(\mathbf{z}) = 1, \quad (26)$$

we substitute the approximation

$$u(\mathbf{y}) - u(\mathbf{x}) \approx \int_{\mathbb{R}^n} (u(\mathbf{z}) - u(\mathbf{x})) \eta_\epsilon(\mathbf{y} - \mathbf{z}) dV(\mathbf{z}) \quad (27)$$

into Eq.(4), and exchange the order of integration. This leads to the following approximation:

$$\Delta^\alpha u(\mathbf{x}) \approx \frac{1}{\epsilon^\alpha} \int_{\mathbb{R}^n} (u(\mathbf{y}) - u(\mathbf{x})) G_\epsilon(\mathbf{x} - \mathbf{y}) dV(\mathbf{y}), \quad (28)$$

where $G_\epsilon(\mathbf{r})$ is found to be equal to Eq. (11), leading to the following PSE scheme:

$$\frac{\partial u_i}{\partial t} = \frac{1}{\epsilon^\alpha} \sum_{j \in \mathcal{I}} V_j (u_j - u_i) G_\epsilon(\mathbf{x}_j - \mathbf{x}_i). \quad (29)$$

This scheme has the unappealing property that it relies on a non-monotonic kernel that changes sign and assumes negative values at the origin. Computational tests using the scheme in Eq. (29) exhibited large errors and signs of instability. Consequently, we decided not to pursue this avenue.

As an alternative to the approach above, we sought to approximate $\Delta^\alpha u(\mathbf{x})$ with a convolution of the form:

$$\mathcal{I}(\epsilon, \mathbf{x}) \equiv \frac{c}{\epsilon^\alpha} \int_{\mathbb{R}^n} (u(\mathbf{y}) - u(\mathbf{x})) K_\epsilon(\mathbf{x} - \mathbf{y}) dV(\mathbf{y}). \quad (30)$$

A suitable template for $K_\epsilon(\mathbf{x})$ is:

$$K_\epsilon(\mathbf{r}) = \frac{1}{\epsilon^n} K\left(\frac{\mathbf{r}}{\epsilon}\right) \quad K(\mathbf{r}) = -\frac{\mathbf{r}}{|\mathbf{r}|^2} \cdot \nabla \kappa^\beta(\mathbf{r}) \quad (31)$$

where $\kappa^\beta(\mathbf{r})$ is defined as in (14) or in (15), depending on the chosen order of the method. A straightforward calculation enables us to relate the kernels K and \mathbf{F} , namely through $\nabla \kappa^\beta(\mathbf{r}) = \mathbf{F}(\mathbf{r})$, for both the second-order and fourth-order approximations. To justify this choice and determine the constant c , we substitute the template (31) into the definition of $\mathcal{I}(\epsilon, \mathbf{x})$:

$$\mathcal{I}(\epsilon, \mathbf{x}) \equiv \frac{c}{\epsilon^\alpha} \int_{\mathbb{R}^n} (u(\mathbf{y}) - u(\mathbf{x})) \times \frac{\Gamma_{n,2}(\alpha)}{\epsilon^n} {}_1F_1\left(\frac{\alpha+n}{2}, \frac{n+2}{2}, -\left(\frac{\mathbf{y}-\mathbf{x}}{\epsilon}\right)^2\right) dV(\mathbf{y}). \quad (32)$$

In order for $\mathcal{I}(\epsilon, \mathbf{x}) \rightarrow \Delta^\alpha u$ as $\epsilon \rightarrow 0$, we must have:

$$\frac{c \Gamma_{n,2}(\alpha)}{\epsilon^n} {}_1F_1\left(\frac{\alpha+n}{2}, \frac{n+2}{2}, -\left(\frac{\mathbf{y}-\mathbf{x}}{\epsilon}\right)^2\right) \rightarrow \frac{\alpha 2^{\alpha-1} \Gamma\left(\frac{n+\alpha}{2}\right)}{\pi^{\frac{n}{2}} \Gamma\left(\frac{2-\alpha}{2}\right)} \frac{1}{|\mathbf{y}-\mathbf{x}|^{n+\alpha}}. \quad (33)$$

Recalling that ${}_1F_1(a, b, -|x|) \rightarrow \Gamma(b)/\Gamma(b-a)|x|^{-a}$ per $|x| \rightarrow -\infty$ [32], we find $c = \alpha$. This results in the following Kernel-PSE scheme (KPSE):

$$\frac{\partial u_i}{\partial t} = \frac{\alpha}{\epsilon^\alpha} \sum_{j \in \mathcal{I}} V_j (u_j - u_i) K_\epsilon(\mathbf{x}_j - \mathbf{x}_i), \quad (34)$$

For $n = 1$ $K(\mathbf{r})$ is plotted in Fig. 1. Note that the kernel K is positive and exhibits the expected asymptotic behavior $K(\mathbf{r}) \propto |\mathbf{r}|^{-n-\alpha}$ as $|\mathbf{r}| \rightarrow \infty$.

Remark. It is interesting to contrast the PSE constructions for Fickian ($\alpha = 2$) and fractional ($1 < \alpha < 2$) diffusion. Starting from the radial, unit mass, rapidly-decaying kernel η in Eq. (8), in the Fickian case we rely on the template $-|\mathbf{r}|^{-1}d\eta/dr$, and convolve the Taylor expansion of u to extract a PSE approximation of the Laplacian. In the fractional diffusion case, we start by convolving η with the kernel $|\mathbf{r}|^{-\beta}$ to first construct the stretched kernel κ^β . We then rely on the template $-|\mathbf{r}|^{-1}d\kappa^\beta/dr$ to approximate the integral representation of the fractional Laplacian. The PSE approximations for both the Fickian and non-Fickian diffusion cases can be captured in a single, order-dependent, expression of the form:

$$\frac{\alpha}{\epsilon^\alpha} \sum_{j \in \mathcal{I}} V_j(u_j - u_i) \mathcal{L}_\epsilon(\mathbf{x}_j - \mathbf{x}_i) \quad 1 < \alpha \leq 2, \quad (35)$$

where \mathcal{L}_ϵ coincides with Φ_ϵ from Eq. (17) in the Fickian case, and with K_ϵ from Eq. (31) for $1 < \alpha < 2$. Note that for the Fickian case, the diffusion kernel (Φ) has finite second moment; in fact for the present choice of η all the moments of Φ are finite. In contrast, for the fractional diffusion case, the kernel K has finite mass and first moment, but infinite higher moments. This highlights the singular nature of the limit $\alpha \rightarrow 2$.

Remark. We also explored an ad-hoc variant of the approach just presented, based on regularizing the Riesz potential directly using a cutoff lengthscale ϵ . In preliminary tests, this classical [21, 23] regularization approach resulted in poor approximations of the diffusion flux, especially at larger values of α . For brevity, these attempts are omitted from the present discussion.

3.2.4. One-dimensional Green-Particle Strength Exchange

The availability of analytical Green's functions offers an alternative means of constructing conservative methods. In the constant diffusivity case, the Green's function in closed-form is given by [28]

$$\mathcal{G}_\alpha^0(x, t) = \frac{1}{t^{1/\alpha}} L_\alpha^0\left(\frac{x}{t^{1/\alpha}}\right), \quad (36)$$

where $L_\alpha^0(r)$ is the so-called reduced Green function (see Appendix Appendix A). For $\Delta t > 0$, using this Green's function, one can express the change in the solution at an arbitrary point x as [36]

$$\begin{aligned} u(x, t + \Delta t) - u(x, t) &= \mathcal{G}_\alpha^0(x, \Delta t) * u(x, t) - u(x, t) \\ &= \int_{-\infty}^{\infty} \mathcal{G}_\alpha^0(x - y, \Delta t) u(y, t) dy - u(x, t) \\ &= \int_{-\infty}^{\infty} \mathcal{G}_\alpha^0(x - y, \Delta t) [u(y, t) - u(x, t)] dy. \end{aligned} \quad (37)$$

Using here superscripts k and $k + 1$ to refer to the time levels, so that $\Delta t = t^{k+1} - t^k$, the previous representation readily leads to the following particle approximation

$$u_i^{k+1} - u_i^k = \sum_{j \in \mathcal{I}} V_j(u_j^k - u_i^k) E_\epsilon(x_j - x_i), \quad (38)$$

where E_ϵ is the radial kernel given by

$$E_\epsilon(x) = \frac{1}{\epsilon} E\left(\frac{x}{\epsilon}\right) \quad E(x) = L_\alpha^0(x) \quad (39)$$

and

$$\epsilon = \Delta t^{1/\alpha}. \quad (40)$$

The kernel L_α^0 is shown in Fig. 1. It is seen to be positive with monotonic decay to 0 as $|r| \rightarrow \infty$. The particle scheme in (38) is clearly conservative, and it can be alternatively expressed as

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \frac{1}{\epsilon^\alpha} \sum_{j \in \mathcal{I}} V_j(u_j^k - u_i^k) E_\epsilon(x_j - x_i), \quad (41)$$

to contrast the present scheme with the PSE formulations above.

3.3. Diffusion velocity method

The DVM is based on expressing the diffusion term as the divergence of a transport term [29, 37, 38, 39]. To this end, we start by rewriting the general n -dimensional conservation equation (1) as:

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{v}u) = 0, \quad \mathbf{v} \equiv \frac{\mathbf{Q}^\beta}{u}, \quad (42)$$

where \mathbf{v} is the diffusion velocity. As suitable particle approximation of (42) simply consists in advecting particles according to

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i \equiv \mathbf{v}(\mathbf{x}_i), \quad (43)$$

while maintaining constant the particle strength $U_i = V_i u_i$. The scheme is clearly conservative, in the sense that the total strength $\sum_{j \in \mathcal{I}} U_j$ is constant. In (43), the diffusion velocity \mathbf{v}_i is approximated as:

$$\mathbf{v}_i \approx \frac{\mathbf{Q}^\beta(\mathbf{x}_i)}{u(\mathbf{x}_i)} \quad (44)$$

where

$$u(\mathbf{x}_i) = \sum_{j \in \mathcal{I}} U_j \eta_\epsilon(\mathbf{x}_i - \mathbf{x}_j), \quad (45)$$

whereas \mathbf{Q}^β is estimated from Eq. (20) or Eq. (21), for the second-order or fourth-order kernels, respectively. Even though the magnitude of the diffusion velocity \mathbf{v}_i can diverge as $u(\mathbf{x}_i) \rightarrow 0$, this is not occurring in practice, unless the strength u_i of the corresponding particle is very small. In the present setup, particles with very small strength are naturally avoided in the initialization stage, namely through suitable selection of the initial domain. In general applications of DVM, the numerical implementation typically eliminates particles whose strength falls below a user specified threshold.

Note that the implementation of DVM inherently induces distortion of the Lagrangian grid, which may affect the accuracy of the reconstruction in Eq. (45), namely by preventing the needed particle overlap [23]. To guard against such effects, a particle redistribution algorithm can be implemented [40]. The application of a 1D DVM scheme utilizing a remeshing algorithm is illustrated in the following section.

4. Results for the 1D case

4.1. Reference problem

In this section, the behavior of the particle schemes is investigated, focusing on the effects of the main numerical parameters: the kernel width ϵ , the number of particles N (that is $\text{Card}(\mathcal{I})$), the half-width D of the discretized domain. As a consequence of both the slow decay of some of the kernels, and of the characteristic length scale associated to the fundamental solution (see Appendix Appendix A), the particles are initialized in a truncated domain centered at the origin. The half-width of the domain is denoted by D , and is specified in terms of the radius R_α (see Appendix Appendix A, Eq. A.3) and the final time, t_f . Unless otherwise noted, we shall use

$$D = 160 t_f^{1/\alpha} R_\alpha. \quad (46)$$

Thus, the extent of the computational domain varies with the fractional diffusion order, reflecting the dependence of the characteristic length on α . The DVM is computed using a much smaller domain coverage of $D = 10 t_f^{1/\alpha} R_\alpha$, due to its intrinsic ability to adapt the Lagrangian grid. We use an odd number of particles uniformly distributed over $[-D, D]$, with the first and last particle centers lying on the extremes of the domain, whereas one of the particles is centered at $x = 0$ to better capture the peak value of the solution. Although computational efficiency is not the concern of the present work, all the simulations were performed on a 56-core workstation, running a parallel implementation of the methods using the shared memory paradigm implemented by the OpenMP library for FORTRAN language.

The methods considered in the numerical tests are summarized in Table 1.

Table 1: Summary of the methods.

	referred	method	kernel
Direct Differentiation	DD	Eq. (10)	Eq. (11)
Flux-PSE	FPSE	Eq. (25)	Eq. (20), (23)
Kernel-PSE	KPSE	Eq. (34)	Eq. (31)
Green-PSE	GPSE	Eq. (41)	Eq. (39)
Diffusion-Velocity	DVM	Sec. 3.3	-
Finite-Differences	FD	Ref. [41]	-

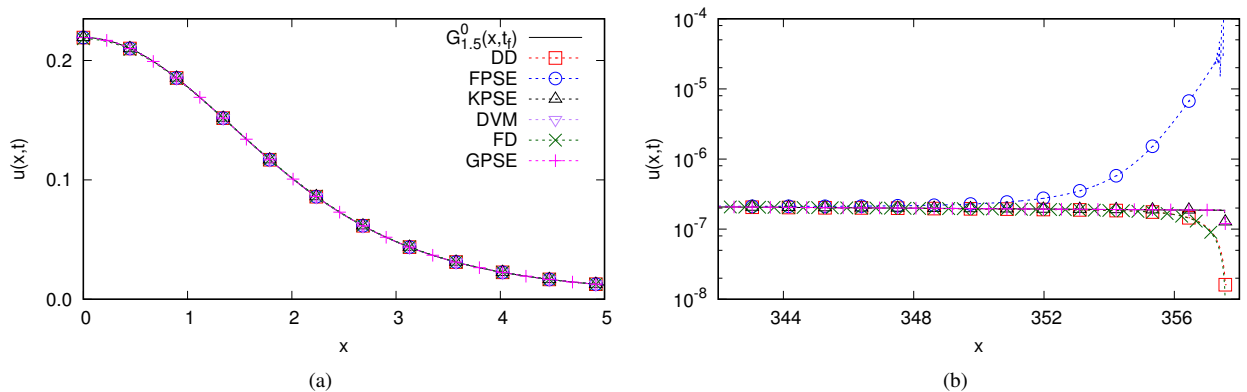
As a reference case, we consider the evolution of the fundamental solution with $\beta = 0.5$, from time $t_0 = 0.5$ to $t_f = 1.5$. Note that the initial condition is selected so as to enable a comparison of the numerical solution and analytical expressions of the fundamental solution, and accordingly to test the accuracy and stability of the computations. The initial condition reflects an even, bell-shaped curve, with a global maximum at $x = 0$, and asymptotic decay as $|x|^{-\alpha}$.

Following the rule given above, for $\beta = 0.5$ we obtain $D = 357.5$, and we use as a reference discretization a grid with $N = 64000$ particles, corresponding to a constant particle volume $V_j = 2D/(N-1) = h \simeq 1.11 \cdot 10^{-2}$. For all the methods but GPSE, for convenience of the analysis we select a kernel size $\epsilon = 2h$, that is, an overlap ratio of 2, and we consider a classical second-order Runge-Kutta scheme (RK2) with time step $\Delta t = 5 \cdot 10^{-5}$ to integrate the evolution of the particle strengths or locations. For the GPSE method, the time step is set to a larger value, $\Delta t = 2.5 \cdot 10^{-3}$, to obtain a smoothing parameter ϵ in Eq. (39) which is roughly equal to $2h$, as in the other methods.

As mentioned in Section 3.3, a distortion of the Lagrangian grid occurs in DVM as particles are transported, and a remeshing strategy is needed to avoid deterioration of the particle representation. In the tests below, a remeshing scheme is incorporated that interpolates the solution on the initial grid using a compact, piecewise cubic, interpolation function [23, 40] preserving the four first moments of the solution. Specifically, remeshing is applied whenever the interparticle spacing exceeds a fixed threshold equal to $1.5h$.

We also contrast the particle predictions with results obtained using a finite-difference scheme recently developed in Ref. [41]. Specifically, assuming homogeneous Dirichlet boundary conditions on $[-D, D]$, we take advantage of the finite-difference approximation of the fractional diffusion operator from [41] using the same regular grid used to initialize the particle distribution, and integrate the resulting discrete system. Note that the scheme in [41] was developed to accommodate variable diffusivity fields and is by construction first-order in space. Nonetheless, we believe that the comparison is instructive.

The exact solution at $t_f = 1.5$, and the five particle solutions are provided in Fig. 2 for $x \in [0, 5]$ (left plot) and $x \in [342, 358]$ (right plot). It is found that all the methods accurately capture the peak value of the solution and the

Figure 2: Fundamental solution at $t = 1.5$ for $\beta = 0.5$ computed with the four proposed methods, in the reference case.

decay as $|x|$ increases. In fact the error is very small, except near the boundaries of the computational domain, and more marked for the DD and FPSE solutions. It is remarked that the absolute point-wise error remains small everywhere, whereas the relative point-wise error increases as $|x| \simeq D$. This behavior does not signal instability, but indicates that domain truncation errors are more pronounced at the computational boundaries. Remarkably, all the particle methods show a boundary error comparable to the finite-difference solution, which imposes homogeneous conditions at the boundaries of the extended grid. This illustrates the ability of particle methods to naturally accommodate far-field conditions [23].

Although this work is devoted to deterministic methods, it is interesting to contrast the present results with previous simulations performed using random walk models. In particular, the solution of our reference case is reported at $t = 1$ in Ref. [14]. The results in Ref. [14] show that the random walk model can closely match the evolution of the fundamental solution, though statistical averaging errors are clearly visible, most notably near the tails of the solution.

4.2. Effect of domain truncation

To gain a better understanding of the effect of domain truncation, and how errors incurred at the computational boundary can affect the solution inside the entire domain, we proceed as follows. First we focus our attention on the relative error measured over a subinterval of half-width D_ε , and consider the relative L_1 -error:

$$\varepsilon_{D_\varepsilon} = \frac{\int_{-D_\varepsilon}^{D_\varepsilon} |u(x, t_f) - G_\alpha^0(x, t_f)| dx}{\int_{-D_\varepsilon}^{D_\varepsilon} |G_\alpha^0(x, t_f)| dx}. \quad (47)$$

We recall that $G_\alpha^0(x, t_f)$ is the fundamental solution of the reference problem given in Appendix Appendix A. We set $D_\varepsilon = 5R_\alpha$, and we consider domain half widths $D = Ct_f^{1/\alpha}R_\alpha$. In the following we use $t_f = 1.5$ and $C \in [20, 160]$ such that $D_\varepsilon < D$ for all C . Note that for this definition of D_ε , the solution at $t_f = 1.5$, as considered in the following, has roughly 97% of its mass in the interval $[-D_\varepsilon, D_\varepsilon]$. Also note that $C = 160$ corresponds to the computational domain of the reference case. In order to isolate the effect of the domain truncation, we use a consistent spatial discretization, with the particle volume h depending solely on β . Specifically, we set $h = 1.21 \cdot 10^{-2}$, $1.11 \cdot 10^{-2}$, and $1.47 \cdot 10^{-2}$, for $\beta = 0.1$, 0.5 , and 0.9 respectively. The smoothing parameter is set as before to be $\epsilon = 2h$; the time-step of the DD, FPSE, KPSE, and DVM methods is fixed to $\Delta t = 5 \cdot 10^{-5}$ in the RK2 scheme, while the GPSE method uses (40) and $\epsilon = 2h$ to obtain $\Delta t = 2.5 \cdot 10^{-2}, 2.5 \cdot 10^{-3}, 1 \cdot 10^{-3}$ for $\beta = 0.1, 0.5$, and 0.9 , respectively. Finally, the relative error $\varepsilon_{D_\varepsilon}$ is approximated by a summation over the particles, namely according to

$$\varepsilon_{D_\varepsilon} \approx \frac{\sum_{i \in I_\varepsilon} V_i |u_i(t_f) - G_\alpha^0(x_i, t_f)|}{\int_{-D_\varepsilon}^{D_\varepsilon} |G_\alpha^0(x, t_f)| dx}, \quad (48)$$

where $I_\varepsilon \subset I$ is the subset of particle indexes such that $|x_i| \leq D_\varepsilon$. As anticipated in the previous section, diffusion-velocity methods intrinsically tackle the problem of domain truncation by suitably stretching the Lagrangian grid. As a consequence, a domain truncation analysis is not insightful, in light of the remeshing strategy use to maintain a suitable particle distribution [23, 40].

In the left plots of Figures 3–5, the relative error is plotted against D , for all four methods considered and for different orders of the fractional derivative. For the DD and FPSE methods it is seen that with $C = Dt_f^{-1/\alpha}/R_\alpha > 20$, the error $\varepsilon_{D_\varepsilon}$ is essentially independent of D . This indicates that for $|x| \leq D_\varepsilon$ the error is then dominated by the space and time discretization errors, and not by the domain truncation error. One can also observe that the relative error increases as β decreases, and based on this metric the DD method seems more accurate than the FPSE method at the same discretization parameters and value of β .

Focusing now on the case of the KPSE and GPSE methods, different behaviors are reported as the size D of the computational domain varies. First, for the KPSE method (solid lines), an initial decay of $\varepsilon_{D_\varepsilon}$ with D is observed, following which the relative error levels off. The saturation of the relative error occurs at a value of C that increases

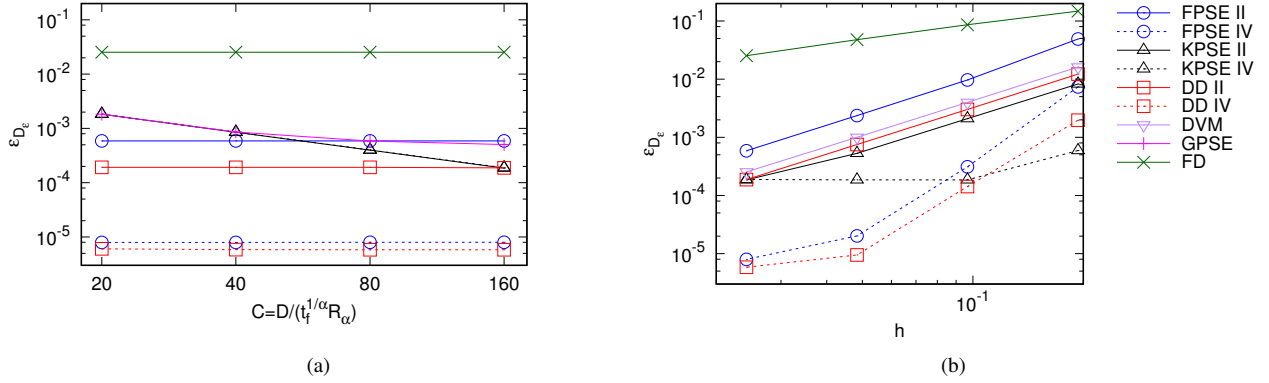


Figure 3: Relative error $\varepsilon_{D_\varepsilon}$ at $t_f = 1.5$ for $\beta = 0.1$, as a function of the domain truncation (a) with $h = 1.21 \cdot 10^{-2}$, and of the particle volume h (b) with $D = 160 t_f^{1/\alpha} R_\alpha$. Roman numerals indicate the order of the method.

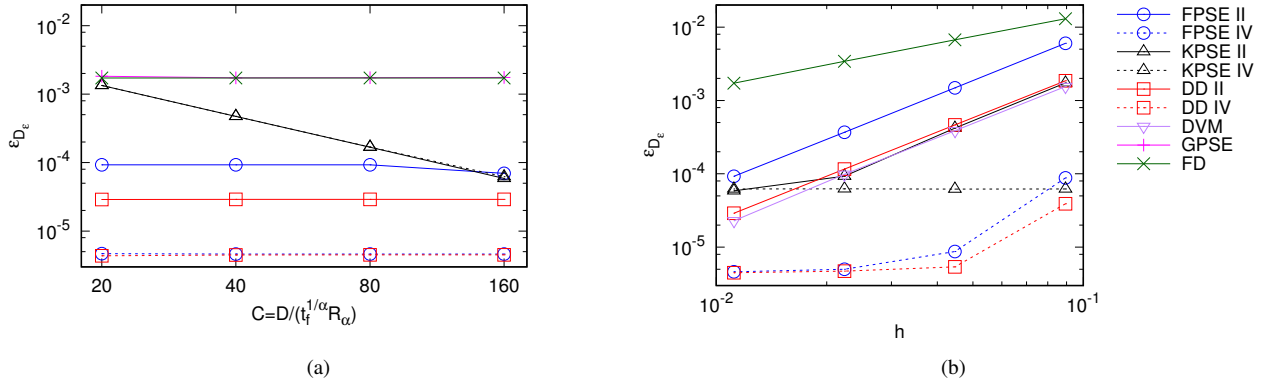


Figure 4: Relative error $\varepsilon_{D_\varepsilon}$ at $t_f = 1.5$ for $\beta = 0.5$, as a function of the domain truncation (a) with $h = 1.11 \cdot 10^{-2}$, and of the particle volume h (b) with $D = 160 t_f^{1/\alpha} R_\alpha$. Roman numerals indicate the order of the method.

as β decreases. In other words, the KPSE method is more sensitive to the truncation of the computational domain when β is small. This finding is consistent with the analysis of the characteristic length scales of the solution. For the GPSE method, one sees that for short computational domains with $C \lesssim 30$ the error decreases with β , whereas for sufficiently large D we observe once again that $\varepsilon_{D_\varepsilon}$ levels off, with higher values for lower β . This behavior suggests a different mechanism for the effect of the domain truncation in the GPSE method.

In all cases, the results presented justify the value of D selected for the reference case, in Section 4.1, for which domain truncation has negligible impact on $\varepsilon_{D_\varepsilon}$. Finally, it is interesting to remark that, before reaching their respective plateau value as D increases, the relative errors $\varepsilon_{D_\varepsilon}$ of the KPSE and GPSE methods seem to behave like $\varepsilon_{D_\varepsilon} = O(C^{-\alpha})$. This behaviour points to the origin of the domain truncation error, which is related to the decay of the kernels involved and not due to space or time discretization parameters.

4.3. Effect of space and time discretization parameters

In this subsection, we investigate the impact of the spatial (h) and time (Δt) discretization parameters on the error incurred by the proposed methods. We do not consider here the Green's function based approach (GPSE) because in order to maintain a constant overlap ratio ϵ/h , the values of h and Δt cannot be varied independently. In the present numerical experiments, the domain half-width D is kept fixed and coincides with the reference value $D = 160 t_f^{1/\alpha} R_\alpha$.

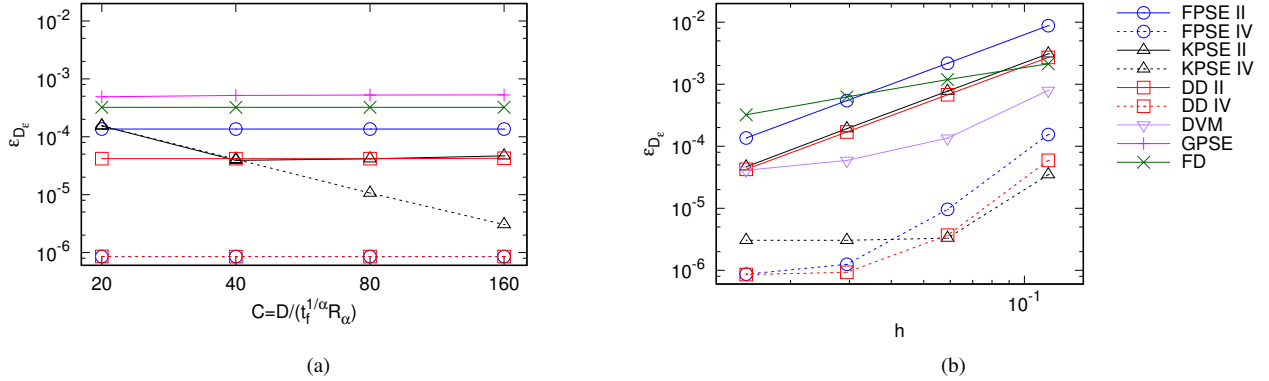


Figure 5: Relative error ϵ_{D_ϵ} at $t_f = 1.5$ for $\beta = 0.9$, as a function of the domain truncation (a) with $h = 1.47 \cdot 10^{-2}$, and of the particle volume h (b) with $D = 160 t_f^{1/\alpha} R_\alpha$. Roman numerals indicate the order of the method.

The right plot in Figures 3, 4, and 5, report ϵ_{D_ϵ} for different values of the particles volume h , and for $\beta = 0.1, 0.5, 0.9$. In these plots, h is varied by changing the number N of particles in the computational domain, while the smoothing parameter is adjusted to have $\epsilon = 2h$; as before, the RK2 time-scheme is used with a fixed time step $\Delta t = 5 \cdot 10^{-5}$. For the coarsest spatial discretizations (the largest values of h), we observe a relative error decreasing as $O(h^2)$ for the second-order methods, as expected from the moment properties of the kernels. The error for the fourth-order implementations of the schemes shows the same behaviour, but decreasing as $O(h^4)$, which is evident for the larger values of h . The FPSE method is seen to have the highest spatial error compared to the others at the same h . For the smallest h tested, the error of the KPSE method is seen to level-off as the time discretization error then becomes dominant, because of the fixed time-step used in all the experiments.

Table 2: Summary of the order of self-convergence in space for different values of β .

β	$h \cdot 10^2$	DD		FPSE		KPSE		DVM
		II ord.	IV ord.	II ord.	IV ord.	II ord.	IV ord.	II ord.
0.1	4.83	2.01	3.88	2.04	3.97	1.98	3.79	2.00
0.5	4.46	2.00	3.97	2.01	3.86	2.00	4.06	2.00
0.9	5.89	2.00	4.00	2.01	3.96	2.00	3.99	2.42

To have a better characterization of the spatial convergence rates of the proposed DD, FPSE and KPSE methods, we estimate their orders of self-convergence. Specifically, the self convergence order $p^{(\ell)}$ is estimated from

$$p^{(\ell+1)} = \log_2 \frac{\sum_{i \in \mathcal{I}^{(\ell)}} |u_i^{(\ell)} - u_i^{(\ell+1)}|}{\sum_{i \in \mathcal{I}^{(\ell)}} |u_i^{(\ell+1)} - u_i^{(\ell+2)}|} \quad (49)$$

where ℓ denotes the discretization level and $\mathcal{I}^{(\ell)}$ the index set of particles belonging to levels ℓ to $\ell + 2$. When going from a level to the next, the numerical parameter investigated is halved. Table 2 summarizes the order of self-convergence in space for the proposed methods. The expected rate of self convergence is indeed observed.

We finally note that all the methods presently considered exhibit almost no sensitivity to the choice of the smoothing parameter, so long as ϵ is larger than h (not shown). In addition, we have examined the convergence in time of DD, KPSE, and FPSE methods, both the RK1 and RK2 schemes. These tests revealed an order of self convergence that is consistent with the order of the time integrator. For brevity, results from the time convergence study are also omitted.

4.4. Time Stability

Because it relies on the analytical Green's function, the GPSE method is unconditionally stable. The DVM is inherently stable as well, because the particle strengths are invariant during time integration and the particle velocities are bounded. However, the other proposed methods, namely DD, FPSE, and KPSE, are generally subject to a stability constraint on the time step when an explicit time integration scheme is employed. It is then interesting to characterize and compare the stability properties of these three methods. To this end, we denote $\mathbf{u}^T = (u_1, \dots, u_N)$ the vector of particles strengths (the superscript T denotes here the transpose). The evolution of the vector \mathbf{u} can be written as

$$\frac{d\mathbf{u}}{dt} = A\mathbf{u}, \quad (50)$$

where A is a $N \times N$ matrix whose entries depend on the method considered. Because the particles have fixed positions x_i in the proposed methods, the matrix A is time-invariant. Furthermore, because of the radial nature of the kernels used in these schemes, A is also a symmetric matrix with real eigenvalues. Denoting $\lambda_{1 \dots N}$ the eigenvalues of A , the condition $\lambda_i \leq 0$ is necessary to ensure that the ‘‘continuous-in-time’’ solution $\mathbf{u}(t) = \exp[A(t - t_0)]\mathbf{u}(t_0)$ does not diverge as $t \rightarrow \infty$. This condition is satisfied for the DD, FPSE and KPSE. In fact, the FPSE and KPSE methods being conservative, the largest eigenvalue of A for these two methods is exactly 0, while one has $\max_{1 \leq i \leq N} \lambda_i < 0$ for the DD method which is leaking mass.

Upon introduction of the explicit time discretization, one can derive a linear relation between the vectors of particle strengths at time t_k and $t_{k+1} = t_k + \Delta t$, of the form

$$\mathbf{u}^{k+1} = \Pi_{\Delta t}(A)\mathbf{u}^k, \quad (51)$$

where $\Pi_{\Delta t}(x)$ is a polynomial with coefficients depending on Δt and on the selected time discretization. For instance, the polynomials of the RK1 and RK2 schemes are given respectively by

$$\Pi_{\Delta t}^{RK1}(x) = 1 + x \Delta t \quad \Pi_{\Delta t}^{RK2}(x) = 1 + x \Delta t + x^2 \frac{\Delta t^2}{2}. \quad (52)$$

Note that $\Pi_{\Delta t}(A)$ is symmetric, because A is symmetric, with real eigenvalues. In fact, if λ is an eigenvalue of A then $\Pi_{\Delta t}(\lambda)$ is an eigenvalue of $\Pi_{\Delta t}(A)$. Therefore, the stability of the scheme requires that $|\Pi_{\Delta t}(\lambda_i)| \leq 1$ for $i = 1, \dots, n$ or explicitly $|1 + \lambda_i \Delta t| \leq 1$, for RK1, and $|1 + \lambda_i \Delta t + \lambda_i^2 \frac{\Delta t^2}{2}| \leq 1$, for RK2. Since all $\lambda_i \leq 0$, the two conditions reduce to

$$0 \leq -\lambda_i \Delta t \leq 2, \quad (53)$$

showing that, as expected, the critical time step for RK1 and RK2 are identical. Consequently, we only need to focus on the lowest eigenvalue of A , denoted λ_{\min} , and we are led to the stability condition

$$\Delta t \leq 2|\lambda_{\min}|^{-1}. \quad (54)$$

Defining the nondimensional parameter a as

$$a \equiv \frac{2\mathcal{D}}{|\lambda_{\min}|h^\alpha}, \quad (55)$$

the stability condition becomes

$$\frac{\mathcal{D}\Delta t}{h^\alpha} \leq a. \quad (56)$$

Table 3 summarizes the value of a for the DD, FPSE and KPSE methods. These values were computed with $h = 1.21 \cdot 10^{-2}$, $1.11 \cdot 10^{-2}$, and $1.47 \cdot 10^{-2}$ for $\beta = 0.1, 0.5$, and 0.9 respectively, using a power iteration method to compute the leading eigenvalue λ_{\min} of A . The table shows that the KPSE method has the most stringent stability requirement on the time step, followed by the DD method, whereas the FPSE method allows a time step that is several times larger without compromising stability. It is also observed that the stability threshold, a , does not depend significantly on the value of β . It is however interesting to note that whereas in the DD and FPSE methods the stability limit becomes less stringent as β increases, the KPSE method exhibits the opposite trend. Overall, the stability limit of all the methods are comparable.

Table 3: Summary of the values of the stability limit a , estimated using Eq. (55), for DD, FPSE and KPSE and for different values of β .

β	DD	FPSE	KPSE
0.1	4.81	7.05	2.25
0.5	5.25	8.83	2.17
0.9	5.43	10.5	2.04

5. Results for the 2D case

In this section we focus on the two-dimensional problem with $\beta = 0.5$, and integrate the space-fractional diffusion equation using as initial condition the fundamental solution at $t = 0.5$, to a final time $t_f = 1.5$.

The fundamental solution in two dimensions is [27]:

$$G_\alpha^0(\mathbf{x}, t) = t^{-\frac{2}{\alpha}} \Phi_{2,\alpha} \left(\frac{|\mathbf{x}|}{t^{\frac{1}{\alpha}}} \right) \quad (57)$$

$$\Phi_{2,\alpha}(r) = \frac{1}{2\pi} \int_0^\infty \omega e^{-\omega^\alpha} J_0(r\omega) d\omega, \quad (58)$$

where $J_0(x)$ is the Bessel function of the first kind [32]. The function $\Phi_{2,\alpha}(r)$ is depicted in Fig. 6.

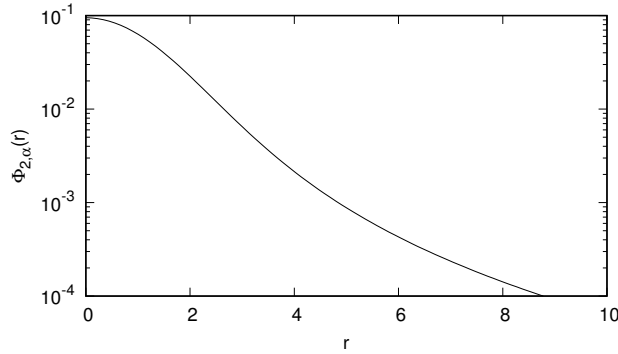


Figure 6: $\Phi_{2,\alpha}(r)$, defined in Eq. (58), versus r .

Numerical simulations are conducted using the DD (Eq. (10)) and FPSE (Eq. (25)) methods, with constant time step $\Delta t = 10^{-3}$. The truncated two-dimensional spatial domain coincides with the square of side $2D$, i.e. $[-D, D] \times [-D, D]$, with $D/(t_f^{1/\alpha} R_\alpha) = 4$. The particles are disposed on a uniform grid with N particles along each side, with grid size $h = 2D/(N - 1)$. Four values of N are considered, $N = 41, 81, 161$ and 321 . Analogously with the one-dimensional case, we fix $\epsilon = 2h$, and we define the particle volume as $V_p = h^2$.

The shape functions $F(\mathbf{r})$ and $G(\mathbf{r})$ depend on the confluent hypergeometric function of the first kind, computation of which can be complex and computationally expensive [42]. Consequently, values of $F(\mathbf{r})$ and $G(\mathbf{r})$ were tabulated, and their value linearly interpolated as needed during the simulations.

Remark. Note that the tabulation of the fractional differentiation kernels F and G must be carefully considered, specifically so that the error associated with their interpolation remains insignificant. For instance, we observed that tabulating F and G with a step $\Delta r = 0.1$ leads to deterioration of the accuracy of the prediction, with an appreciable drop in the rate of convergence (not shown). On the other hand, we found that using a step size $\Delta r = 0.01$, kernel interpolation has an insignificant impact on the prediction. The results presented below are obtained using fine kernel tables, namely using $\Delta r = 0.01$.

To gain an appreciation into the effect of spatial resolution, we computed the radial flux and diffusion term at the initial time, and contrasted the numerical predictions with the analytical expression. Results are reported in Fig. 7

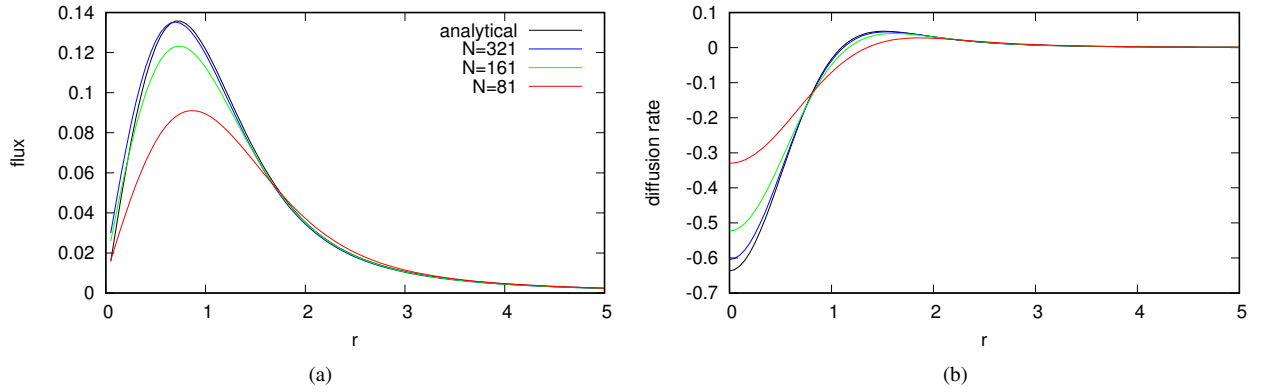


Figure 7: Radial flux (a) and diffusion rate (b) obtained at $t = 0.5$ with $N = 81, 161,$ and $321,$ as indicated. The domain size $D/(t_f^{1/\alpha} R_\alpha) = 4.$ The exact solution is also shown for comparison.

which depicts radial profiles of $\mathbf{Q}^\beta(r, t = 0.5)$ and $\nabla^\alpha u(r, t = 0.5),$ obtained with $N = 81, 161,$ and $321.$ It can be seen that the solutions progressively converge to the analytical value, and the second-order of convergence can be visually anticipated.

Figure 8 shows profiles of u at the final time, $t_f = 1.5,$ along two lines, namely $y = 0$ and $y = x.$ The computed

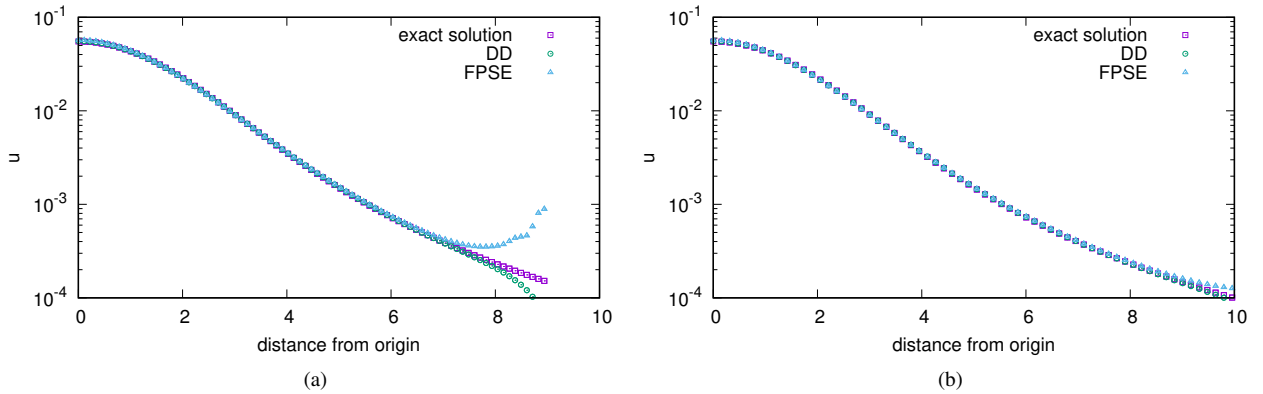


Figure 8: Numerical solution along $y = 0$ (a) and $y = x$ (b). Plotted are results obtained using DD and FPSE, with $N = 161$ and $D/(t_f^{1/\alpha} R_\alpha) = 4,$ in logarithmic scale. The exact solution is also shown for comparison.

solutions closely follow the analytical solution in both directions. Far from the origin, a higher accuracy is achieved in the direction $y = x,$ compared to $y = 0,$ for both methods. This effect is related to the greater extent of the computational domain in the direction $y = x.$ Closer to the boundaries, the DD solution method slightly underestimates the analytical solution, whereas FPSE slightly overestimates it. As in the 1D case, the origin of these effects can be traced to domain truncation and the properties of the approximation. We finally note that both methods reproduce well the radial structure and decay of the solution.

To quantify the accuracy of the predictions, we estimate the relative error at the final time using

$$\varepsilon_{D_\varepsilon} = \frac{\int_{D_\varepsilon} |u(\mathbf{x}, t_f) - G_\alpha^0(\mathbf{x}, t_f)| \, d\mathbf{x}}{\int_{D_\varepsilon} |G_\alpha^0(\mathbf{x}, t_f)| \, d\mathbf{x}}, \quad (59)$$

where D_ε is now the disk of radius 5 centered at the origin. Plotted in Fig. 9 are the computed values of $\varepsilon_{D_\varepsilon}$ for all cases considered. As expected, for both DD and FPSE methods the error decreases as h decreases, reflecting

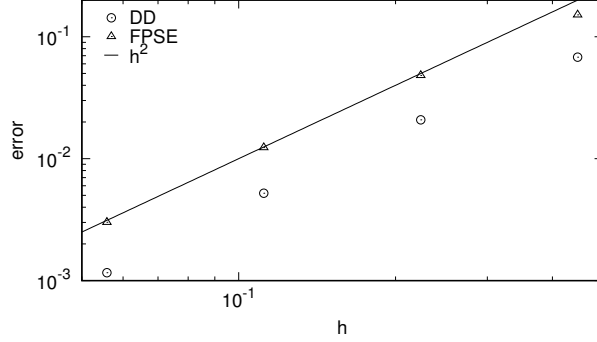


Figure 9: Relative error, computed using Eq. (59), versus h . Plotted are results obtained using DD and FPSE as indicated.

second-order convergence in the entire range of parameters considered. As previously observed, the FPSE method is generating a larger error with respect to the DD method. Although not shown, it was verified that the relative error appears to be weakly affected by the domain truncation for $D/(t_f^{1/\alpha} R_\alpha) > 4$. Because the decay of the kernels depends on the spatial dimension, the domain truncation has a weaker effect with respect to the 1D case. Specifically, with DD and $D/(t_f^{1/\alpha} R_\alpha) = 4$, we only observed a loss of the 1.2% of the initial mass at the end of the simulations.

Similar to the one-dimensional case, the order of self-convergence was assessed using Eq. (49). It was found to be equal to 1.59 and 1.52 at $h = 0.223$, and 1.94 and 1.95 at $h = 0.11$, for DD and FPSE respectively. This is consistent with 1D results, and further supports that the methods achieve second order spatial convergence.

6. Conclusions

In this work different particle schemes have been proposed to simulate one and two-dimensional space-fractional diffusion equations in unbounded domains. This development was motivated by the known reliability and robustness of the particle methods, and their ability to accommodate unbounded domains and effectively treat far-field conditions. Furthermore, we took advantage of the Particle Strength Exchange (PSE) formalism to construct conservative particle schemes, and developed a conservative diffusion velocity method.

The relative merits of the different schemes can be summarized as follows:

- The DD scheme, although nonconservative, has the advantage of being easy to formulate and simple to implement. In particular, it yields the diffusion flux via single convolution sum over the fields of the individual particles.
- The FPSE scheme requires two successive passes over the set of particles to estimate the rate of change of the solution. The first convolution involves long-range interactions, whereas the second approximates a local divergence operator that can be localized to nearby particles. However, it is conservative by construction.
- Compared to FPSE, the KPSE scheme has the advantage of being computationally less intensive, with the source term obtained in a single convolution sum. On the downside, the potential extension of the KPSE scheme to the case of variable diffusivity appears more complicated than for DD and FPSE, which are readily amenable to this generalization.
- The GPSE scheme is conservative and has a moderate computational complexity, but as KPSE it lacks generality. In particular, extension of GPSE to the case of variable diffusivity is not straightforward.
- The DVM method is based on a conservative construction, but this property is not necessarily maintained when a particle redistribution scheme is incorporated that does not strictly satisfy this property. On the other hand,

particle redistribution algorithms offer substantial advantages, including maintaining suitable resolution, and in the present context significantly restricting the simulation domain and accordingly the number of particles.

Another characteristic of GPSE is that it is based on an exact time integration, owing to the knowledge of the fundamental solution. This readily yields a (fractional order dependent) relation between the time step and the spatial discretization parameter ϵ (or alternatively the particle volume h). For the other four schemes (DD, FPSE, KPSE and DVM) we have performed a mesh refinement study to verify that the spatial convergence rate coincided with the order of the kernel selected to represent the solution. We have also shown that this spatial convergence rate is achieved only when the computational domain is sufficiently large. Otherwise, the domain truncation error can substantially affect the solution, mostly at the computational boundaries. Finally, a brief investigation of the stability of the integration scheme was conducted for the DD, FPSE and KPSE schemes. This revealed that all three schemes are subject to comparable time step constraints, which vary weakly with the order of the fractional derivative.

The results of the present work suggest further investigations on the use of particle methods to solve fractional diffusion problems. Specifically, it appears to be worthwhile to develop additional capabilities, particularly to enable tackling fractional diffusion problems in complex settings, such as when diffusivity and/or order are variable, or when the solution is advected by an external field. An attractive avenue would be the introduction of adaptive particle discretizations, elaborate re-meshing or redistribution strategies [43, 44], and multiple resolutions approaches. Another avenue concerns the development of fast methods, which could dramatically reduce the computational effort needed to estimate particle interactions involved in flux and source term expressions. Finally, it would also be interesting to explore stochastic particle approaches [40] for problems with uncertain parameters, and the exploitation of related representations to support inference or calibration problems.

Appendix A. One-dimensional fundamental solution

The Green's function $\mathcal{G}_\alpha^0(x,t)$ of (3) can be expressed as in Eq. (36) [28] and it contains L_α^0 , the so-called reduced Green function. For $1 < \alpha < 2$, L_α^0 and $-\infty < x < +\infty$ is defined as

$$L_\alpha^0(x) = \frac{1}{x\pi} \sum_{m=1}^{\infty} (-x)^m \frac{\Gamma(1+m/\alpha)}{m!} \sin\left[\frac{-m\pi}{2}\right]. \quad (\text{A.1})$$

Note that L_α^0 is even, i.e. $L_\alpha^0(-x) = L_\alpha^0(x)$, and that for $x \rightarrow +\infty$ it admits the following asymptotic expansion [28]:

$$L_\alpha^0(x) \sim \frac{-1}{x\pi} \sum_{m=1}^{\infty} (-x^{-\alpha})^m \frac{\Gamma(1+\alpha m)}{m!} \sin\left[\alpha m \frac{\pi}{2}\right]. \quad (\text{A.2})$$

In Fig. A.10, A.11, and A.12, we use the series representation (A.1) and asymptotic expansion (A.2) to illustrate the behavior of L_α^0 for $\alpha = 1.1, 1.5$, and 1.9 , respectively.

Because L_α^0 does not have a finite second moment, to form a characteristic width of the corresponding distribution we resort to the first moment:

$$R_\alpha \equiv \frac{\int_{-\infty}^{\infty} |x| L_\alpha^0(x) dx}{\int_{-\infty}^{\infty} L_\alpha^0(x) dx}. \quad (\text{A.3})$$

Noting that the denominator in the above definition is equal to 1, and that $L_\alpha^0(x)$ is evenly symmetric, we may rewrite R_α as

$$R_\alpha = 2 \int_0^{\infty} x L_\alpha^0(x) dx. \quad (\text{A.4})$$

To estimate R_α we decompose the integral according to

$$R_\alpha = 2(H_1 + H_2), \quad (\text{A.5})$$

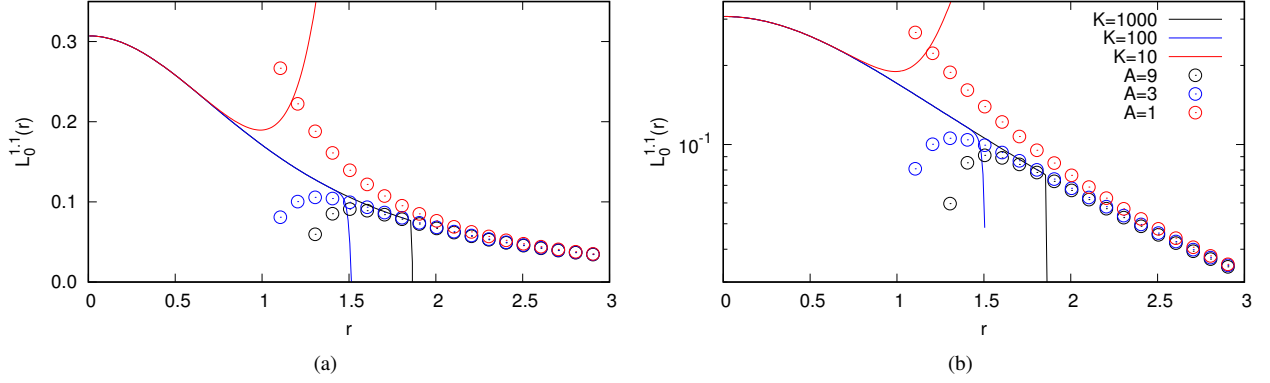


Figure A.10: $L_\alpha^0(r)$ (solid lines) and the asymptotic expansion (symbols) represented with K and A terms respectively, for $\alpha = 1.1$.

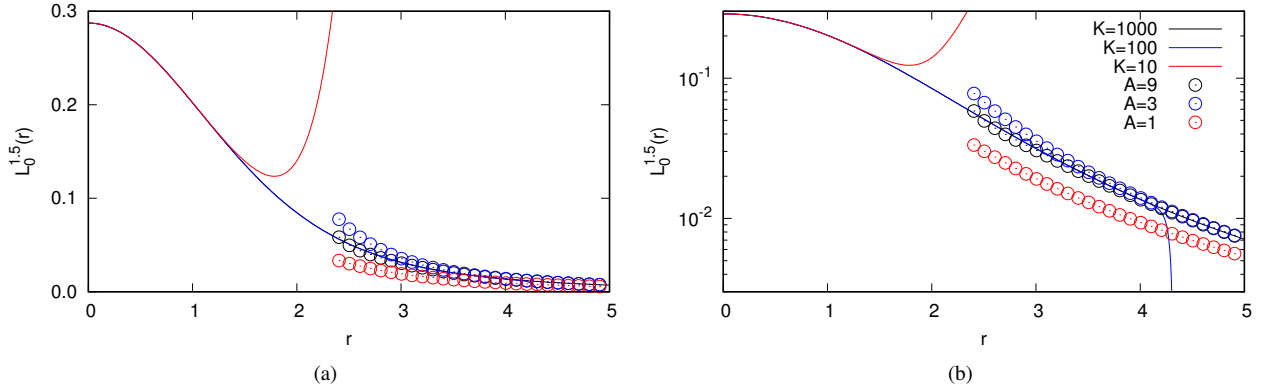


Figure A.11: $L_\alpha^0(r)$ (solid lines) and the asymptotic expansion (symbols) represented with K and A terms respectively, for $\alpha = 1.5$.

where

$$H_1 \equiv \int_0^M x L_\alpha^0(x) dx \text{ and } H_2 \equiv \int_M^\infty x L_\alpha^0(x) dx. \quad (\text{A.6})$$

Substituting (A.1) and (A.2) into (A.6), we obtain:

$$H_1 = \frac{M}{\pi} \sum_{k=0}^{\infty} \frac{(-1)^k \Gamma\left(1 + \frac{2k+1}{\alpha}\right)}{(2k+2)!} M^{2k+1}, \quad (\text{A.7})$$

$$H_2 = \frac{M}{\pi} \sum_{m=1}^{\infty} \frac{(-1)^m \Gamma(1 + \alpha m)}{m!(1 - \alpha m)} \sin\left(\frac{\pi \alpha m}{2}\right) M^{-m\alpha}. \quad (\text{A.8})$$

Estimates of R_α obtained by suitably truncating the above expressions are illustrated in Table A.4.

Table A.4: Estimates of R_α using Eqs. (A.5), (A.7), and (A.8).

α	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
R_α	6.688	3.544	2.512	2.005	1.705	1.509	1.371	1.269	1.190

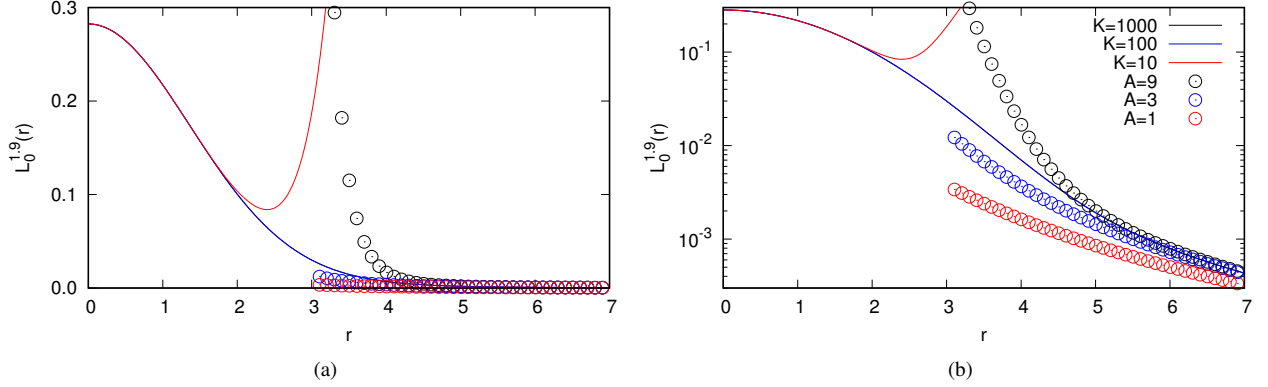


Figure A.12: $L_0^0(r)$ (solid lines) and the asymptotic expansion (symbols) represented with K and A terms respectively, for $\alpha = 1.9$.

Appendix B. Particle representation of the fractional Laplacian, of the fractional gradient, and of the Riesz potential

The methods presented in this work rely on particle representations of specific operators, and results are presented for one and two-dimensional problems. However, the involved particle representations can be given in a general n -dimensional case, where $1 \leq n \leq 3$. One may directly use the available analytical expressions in Section 2, but an easier derivation is possible exploiting the Fourier transform of the various operators involved. Specifically, letting \mathbf{f} denote the frequency vector, with $f = |\mathbf{f}|$, and j the imaginary unit, the fractional Laplacian can be expressed as [27]

$$\mathcal{F} [\Delta^\alpha] = -|2\pi f|^\alpha, \quad (\text{B.1})$$

whereas the fractional gradient is given by

$$\mathcal{F} [\nabla^\beta] = j2\pi\mathbf{f}|2\pi f|^{\beta-1}. \quad (\text{B.2})$$

In the following, we will also make use of the Fourier transform of the n -dimensional Gaussian distribution transform

$$\mathcal{F} \left[\frac{\exp\left(-\frac{\mathbf{x}^2}{\epsilon^2}\right)}{\epsilon^n \pi^{\frac{n}{2}}}\right] = \exp(-\pi^2 \epsilon^2 f^2), \quad (\text{B.3})$$

the Fourier transform of the Riesz potential

$$\mathcal{F} [\mathcal{R}_{n,2-\alpha}] = |2\pi f|^{\alpha-2}, \quad (\text{B.4})$$

as well as the following two general results

$$\int_{\mathbb{R}^n} |\mathbf{y}|^\mu \exp(-a|\mathbf{y}|^2 + j\mathbf{b}\mathbf{x} \cdot \mathbf{y}) dV(\mathbf{y}) = \pi^{\frac{n}{2}} a^{-\frac{\mu+n}{2}} \times \frac{\Gamma\left(\frac{\mu+n}{2}\right)}{\Gamma\left(\frac{n}{2}\right)} {}_1F_1\left(\frac{\mu+n}{2}, \frac{n}{2}, -\frac{\mathbf{b}^2}{4a}|\mathbf{x}|^2\right), \quad (\text{B.5})$$

and

$$\int_{\mathbb{R}^n} \mathbf{y}|\mathbf{y}|^\mu \exp(-a|\mathbf{y}|^2 + j\mathbf{b}\mathbf{x} \cdot \mathbf{y}) dV(\mathbf{y}) = j\mathbf{b}\mathbf{x}\pi^{\frac{n}{2}} a^{-\frac{\mu+n+2}{2}} \times \frac{\Gamma\left(\frac{\mu+n+2}{2}\right)}{2!\Gamma\left(\frac{n+2}{2}\right)} {}_1F_1\left(\frac{\mu+n+2}{2}, \frac{n+2}{2}, -\frac{\mathbf{b}^2}{4a}|\mathbf{x}|^2\right). \quad (\text{B.6})$$

We start from the regularization of the solution $u(\mathbf{x}, t)$, proposed in Eq. (6), here reported in convolution form:

$$u_\epsilon(\mathbf{x}) = u(\mathbf{x}) * \eta_\epsilon(\mathbf{x}), \quad (\text{B.7})$$

where $*$ denotes the convolution symbol. Exploiting the basic properties of the Fourier transform, one readily sees that the Fourier transform of the n -dimensional Riesz potential, of the fractional gradient, and the fractional Laplacian, of $u_\epsilon(\mathbf{x})$, are:

$$\mathcal{F}[\mathcal{R}_{n,2-\alpha}(u_\epsilon, x)] = |2\pi f|^{\alpha-2} \mathcal{F}[\eta_\epsilon] \mathcal{F}[u], \quad (\text{B.8})$$

$$\mathcal{F}[\nabla^\beta u_\epsilon] = j2\pi \mathbf{f} |2\pi f|^{\beta-1} \mathcal{F}[\eta_\epsilon] \mathcal{F}[u], \quad (\text{B.9})$$

$$\mathcal{F}[\Delta^\alpha u_\epsilon] = -|2\pi f|^\alpha \mathcal{F}[\eta_\epsilon] \mathcal{F}[u]. \quad (\text{B.10})$$

Equations (B.8)–(B.10) suggest that these quantities are computed as a convolution of u with specific functions, defined by their Fourier transform:

$$\mathcal{F}[\epsilon^{-n} \kappa_\epsilon^\beta(\mathbf{x})] = |2\pi f|^{\alpha-2} \mathcal{F}[\eta_\epsilon], \quad (\text{B.11})$$

$$\mathcal{F}[\epsilon^{-n} F_\epsilon(\mathbf{x})] = j2\pi \mathbf{f} |2\pi f|^{\beta-1} \mathcal{F}[\eta_\epsilon], \quad (\text{B.12})$$

$$\mathcal{F}[\epsilon^{-n} G_\epsilon(\mathbf{x})] = -|2\pi f|^\alpha \mathcal{F}[\eta_\epsilon]. \quad (\text{B.13})$$

The computation of the inverse Fourier transform of Eq. (B.11)–(B.13) is easily performed using Eq. (B.5) and (B.6), leading directly to Eq. (14), (20), and (11), respectively. Therefore, introducing the notation

$$\Gamma_{n,b}(\alpha) = \frac{2^\alpha \Gamma\left(\frac{\alpha+n}{2}\right)}{\pi^{\frac{n}{2}} \Gamma\left(\frac{n+b}{2}\right) b!}, \quad (\text{B.14})$$

we can write:

$$\mathcal{R}_{n,2-\alpha}(u_\epsilon, x) = \frac{1}{\epsilon^n} \kappa_\epsilon^\beta * u, \quad (\text{B.15})$$

$$\nabla^\beta u_\epsilon = \frac{1}{\epsilon^n} F_\epsilon * u, \quad (\text{B.16})$$

$$\Delta^\alpha u_\epsilon = \frac{1}{\epsilon^n} G_\epsilon * u. \quad (\text{B.17})$$

Applying the midpoint rule to discretize the convolution in Eq. (B.15)–(B.17) leads to Eq. (13), (19), and (10), respectively.

If the function $u(\mathbf{x}, t)$ is represented using the fourth order kernel in Eq. (9), the particle representations of $\mathcal{R}_{n,2-\alpha}(u, \mathbf{x})$, $\nabla^\beta u$, and $\Delta^\alpha u$ can be computed following the same procedure. One may observe that the fourth order kernel is merely the difference between two Gaussian distributions, the first being the second-order kernel multiplied times 2, the second being the same kernel where ϵ is substituted with $\sqrt{2}\epsilon$ everywhere. Therefore, the fourth order particle representations will consist in the difference of two terms, the first being the already obtained second-order representation times 2, and the second term being again the same representation, but where ϵ is substituted with $\sqrt{2}\epsilon$. As a consequence, for the fourth order kernel, Eq. (14), (20), and (11) are modified by substituting ${}_1F_1(a, c, x)$ with:

$$2 {}_1F_1(a, c, x) - 2^{-a} {}_1F_1\left(a, c, \frac{x}{\sqrt{2}}\right). \quad (\text{B.18})$$

Compliance with Ethical Standards

Funding: Research reported in this publication was supported by research funding from King Abdullah University of Science and Technology (KAUST).

Conflict of Interests: On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

- [1] J. Klafter, I.M. Sokolov, *Physics world* **18**(8), 29 (2005)
- [2] V.E. Lynch, B.A. Carreras, D. del Castillo-Negrete, K. Ferreira-Mejias, H. Hicks, *J. Comput. Phys.* **192**(2), 406 (2003)
- [3] C. Tadjeran, M.M. Meerschaert, H.P. Scheffler, *J. Comput. Phys.* **213**(1), 205 (2006)
- [4] H. Zhang, F. Liu, V. Anh, *Appl. Math. Comput.* **217**(6), 2534 (2010)
- [5] X. Li, C. Xu, *Commun. Comput. Phys.* **8**(5), 1016 (2010)
- [6] K. Mustapha, *IMA J. Numer. Anal.* **31**(2), 719 (2010)
- [7] C. Celik, M. Duman, *J. Comput. Phys.* **231**(4), 1743 (2012)
- [8] K. Mustapha, W. McLean, *SIAM J. Numer. Anal.* **51**(1), 491 (2013)
- [9] W. Deng, J.S. Hesthaven, *ESAIM-Math. Model. Num.* **47**(6), 1845 (2013)
- [10] K. Mustapha, B. Abdallah, K. Furati, *SIAM J. Numer. Anal.* **52**(5), 2512 (2014)
- [11] H. Zhou, W. Tian, W. Deng, *J. Sci. Comp.* **56**(1), 45 (2013)
- [12] K. Mustapha, *Numer. Math.* **130**(3), 497 (2015)
- [13] Y. Liu, Y. Yan, M. Khan, *App. Numer. Math.* **115**, 200 (2017)
- [14] R. Gorenflo, F. Mainardi, D. Moretti, G. Pagnini, P. Paradisi, *Chem. Phys.* **284**(1), 521 (2002)
- [15] R. Gorenflo, A. Vivoli, F. Mainardi, *Nonlinear Dynam.* **38**(1), 101 (2004)
- [16] R. Gorenflo, F. Mainardi, A. Vivoli, *Chaos Soliton. Fract.* **34**(1), 87 (2007)
- [17] H. Zhang, F. Liu, V. Anh, *J. Comput. App. Math.* **206**(2), 1098 (2007)
- [18] L. Shi, Z. Yu, Z. Mao, A. Xiao, H. Huang, *Physica A* **392**(23), 5801 (2013)
- [19] Y. Luchko, *Math. Model. Nat. Phen.* **11**(3), 1 (2016)
- [20] A.J. Chorin, *J. Flu. Mech.* **57**(4), 785 (1973)
- [21] A. Leonard, *J. Comput. Phys.* **37**(3), 289 (1980)
- [22] A. Leonard, *Annu. Rev. Flu. Mech.* **17**(1), 523 (1985)
- [23] G.H. Cottet, P.D. Koumoutsakos, *Vortex methods: theory and practice* (Cambridge university press, 2000)
- [24] D. Fishelov, *J. Comput. Phys.* **86**(1), 211 (1990)
- [25] M. Kwaśnicki, *Fractional Calculus and Applied Analysis* **20**(1), 7 (2017)
- [26] A. Lischke, G. Pang, M. Gulian, F. Song, C. Glusa, X. Zheng, Z. Mao, W. Cai, M.M. Meerschaert, M. Ainsworth, et al., arXiv preprint arXiv:1801.09767 (2018)
- [27] C. Pozrikidis, *The Fractional Laplacian* (Chapman and Hall/CRC, 2016)
- [28] F. Mainardi, Y. Luchko, G. Pagnini, *Fract. Calc. Appl. Anal.* **4**, 153 (2001)
- [29] P. Mycek, G. Pinon, G. Germain, E. Rivoalen, *Comp. Appl. Math.* **35**(2), 447 (2016)
- [30] J.T. Beale, A. Majda, *J. Comput. Phys.* **58**(2), 188 (1985)
- [31] J.D. Eldredge, A. Leonard, T. Colonius, *J. Comput. Phys.* **180**(2), 686 (2002)
- [32] K.B. Oldham, J. Myland, J. Spanier, *An atlas of functions: with equator, the atlas function calculator* (Springer Science & Business Media, 2010)
- [33] P.A. Raviart, in *Numerical methods in fluid dynamics* (Springer, 1985), pp. 243–324
- [34] P. Degond, S. Mas-Gallic, *Math. Comput.* **53**(188), 485 (1989)
- [35] O.M. Knio, A.F. Ghoniem, *AIAA J.* **30**(1), 105 (1992)
- [36] J. Choquin, S. Huberson, *Comp. Flu.* **17**(2), 397 (1989)
- [37] J. Fronteau, P. Combis, A Lie-admissible method of integration of Fokker-Planck equations with non-linear coefficients (exact and numerical solutions). Tech. rep., CEA Centre d'Etudes Nucleaires de Saclay (1984)
- [38] P. Degond, F.J. Mustieles, *SIAM J. Sci. Stat. Comp.* **11**(2), 293 (1990)
- [39] A. Beaudoin, S. Huberson, E. Rivoalen, *Journal of Computational Physics* **186**(1), 122 (2003)
- [40] O.P. Le Maître, O.M. Knio, *J. Comput. Phys.* **226**(1), 645 (2007)
- [41] K. Mustapha, K. Furati, O.M. Knio, O. Le Maître, ArXiv e-prints (2017)
- [42] J.W. Pearson, Computation of hypergeometric functions. Ph.D. thesis, University of Oxford (2009)
- [43] D. Wee, A. Ghoniem, *J. Comput. Phys.* **213**, 239 (2006)
- [44] I. Lakkis, A. Ghoniem, *J. Comput. Phys.* **228**, 491 (2009)