

# A radial basis functions method for fractional diffusion equations

Cécile Piret <sup>a,\*</sup> and Emmanuel Hanert <sup>b</sup>

<sup>a</sup>*Université catholique de Louvain, Institute of Mechanics, Materials and Civil Engineering (iMMC), 4 Avenue G. Lemaître, B-1348 Louvain-la-Neuve, Belgium*

<sup>b</sup>*Université catholique de Louvain, Earth and Life Institute (ELI), Environmental Sciences, Croix du Sud 2/16, B-1348 Louvain-la-Neuve, Belgium*

---

## Abstract

One of the ongoing issues with fractional diffusion models is the design of an efficient high-order numerical discretization. This is one of the reasons why fractional diffusion models are not yet more widely used to describe complex systems. In this paper, we derive a radial basis functions (RBF) discretization of the one-dimensional space-fractional diffusion equation. In order to remove the ill-conditioning that often impairs the convergence rate of standard RBF methods, we use the RBF-QR method [1,33]. By using this algorithm, we can analytically remove the ill-conditioning that appears when the number of nodes increases or when basis functions are made increasingly flat. The resulting RBF-QR-based method exhibits an exponential rate of convergence for infinitely smooth solutions that is comparable to the one achieved with pseudo-spectral methods. We illustrate the flexibility of the algorithm by comparing the standard RBF and RBF-QR methods for two numerical examples. Our results suggest that the global character of the RBFs makes them well-suited to fractional diffusion equations. They naturally take the global behavior of the solution into account and thus do not result in an extra computational cost when moving from a second-order to a fractional-order diffusion model. As such, they should be considered as one of the methods of choice to discretize fractional diffusion models of complex systems.

*Key words:* Radial basis functions, RBF, fractional diffusion equation, anomalous diffusion

---

\* Corresponding author

*Email addresses:* `cecile.piret@uclouvain.be` (Cécile Piret),  
`emmanuel.hanert@uclouvain.be` (Emmanuel Hanert).

## 1 Introduction

Diffusion processes in complex systems are often observed to deviate from standard laws. The discrepancies can occur both for the time relaxation that can deviate from the classical exponential pattern and for the spatial diffusion that can deviate from Ficks second law. The resulting diffusion processes are no longer Brownian and cannot be represented accurately by a second-order diffusion equation. Instead, fractional-order diffusion models can provide a more realistic description of the system behavior [2]. Models based on fractional-order differential operators have received increasing attention in recent years and have been used to model a wide range of problems in surface and sub-surface hydrology [3–5], plasma turbulence [6,7], finance [8,9], epidemiology [10,11] and ecology [12,13].

One of the key issues with fractional diffusion models is the design of efficient numerical schemes for the space and time discretization. Until now, most models have relied on the finite difference (FD) method to discretize both the fractional-order space diffusion term [14–17] and time derivative [18–20]. Some numerical schemes using low-order finite elements (FE) have also been proposed [21–23]. Since fractional derivatives are non-local operators, FD and FE schemes generate large, full coefficient matrices. They require a computational cost of  $\mathcal{O}(n^3)$  and storage of  $\mathcal{O}(n^2)$ , where  $n$  is the number of degrees of freedom. To reduce the computational burden, Wang et al. recently proposed a fast FD method based on the method of Meerschaert and Tadjeran [14,15] that only requires a storage of  $\mathcal{O}(n)$  and a computational cost of  $\mathcal{O}(n \log^2 n)$  [24,25]. The computational cost has been further improved by Pang and Sun who achieved  $\mathcal{O}(n \log n)$  by using a multigrid method [26].

Another approach to designing an efficient numerical scheme is to discretize these non-local differential operators with non-local numerical methods. Hence, the global behavior of the solution can be naturally taken into account and the computational cost is not substantially increased when moving from a second-order to a fractional-order diffusion model. Following that approach, Hanert has proposed a pseudo-spectral (PS) method based on Chebyshev basis functions in space and Mittag-Leffler basis functions in time to discretize the space-time fractional diffusion equation [27,28]. A similar approach has been followed by Li and Xu to discretize the time-fractional diffusion equation with a Jacobi PS method [29]. All these methods offer spectral accuracy, meaning that the error decreases exponentially fast as the number of nodes increases.

Another numerical method that naturally takes the global behavior of the solution into account is the radial basis function (RBF) method. To our knowledge, the RBF method has never been used to solve the fractional diffusion equation. The RBF method can be seen as a sort of compromise between the

FE and the PS methods. On the one hand, the RBF method is based on an expansion into basis functions that have a spatial location like with the FE method. In that sense, these basis functions can be clustered in a specific region to locally increase the accuracy of the method. On the other hand, the basis functions used in the RBF expansion are high-order functions that span the entire domain like with the PS method. It was shown that RBFs converge to PS methods in their flat radial function limit, making RBFs a generalization to PS methods, for scattered nodes and non-flat radial functions [30]. RBFs have several advantages over PS methods: in addition to offering a flexibility in terms of the shape of the domain, they allow a local node refinement, an easy singularity-free generalization to  $N$  dimensions and a shape parameter allowing us to broaden the solution space outside of the polynomial space, especially vulnerable to the Runge phenomenon [31]. Depending on the smoothness of the solution, a “flatter” rather than “steeper” radial function results in better accuracy (see [32] and references therein). However, the system of discrete equations can become ill-conditioned as radial functions are made increasingly flat. That issue led to the development of the RBF-QR method. It was first introduced on the sphere [1,32] and was later adapted to 1, 2 and 3 dimensions [33].

In this paper, we consider discretizations of the space-fractional diffusion equation on the one-dimensional computational domain  $[a, b]$  with both the direct and QR RBF methods. That equation can be expressed as follows :

$$\frac{\partial f(x, t)}{\partial t} = K_\alpha \left[ \frac{1 + \beta}{2} {}_a D_x^\alpha f(x, t) + \frac{1 - \beta}{2} {}_x D_b^\alpha f(x, t) \right], \quad (1)$$

where  ${}_a D_x^\alpha$  and  ${}_x D_b^\alpha$  are the left and right space-fractional Riemann-Liouville derivatives on  $[a, b]$ , respectively (see for instance [34,35]). The parameter  $\beta \in [-1, 1]$  is a skewness parameter representing a preferential direction of jumps that can be observed in heterogeneous systems. When  $\beta = 0$ , the space derivative reduces to a so-called symmetric Riesz derivative [34]. The coefficient  $K_\alpha$  is a generalized diffusivity whose dimension is  $[K_\alpha] = \text{m}^\alpha \text{s}^{-1}$ . The Riemann-Liouville fractional-order derivatives can be defined as follows:

$${}_a D_x^\alpha v(x) = \frac{1}{\Gamma(2 - \alpha)} \frac{\partial^2}{\partial x^2} \int_a^x \frac{v(y)}{(x - y)^{\alpha-1}} dy,$$

$${}_x D_b^\alpha v(x) = \frac{(-1)^2}{\Gamma(2 - \alpha)} \frac{\partial^2}{\partial x^2} \int_x^b \frac{v(y)}{(y - x)^{\alpha-1}} dy,$$

where  $\Gamma(\cdot)$  is Euler’s gamma function. For simplicity, we have assumed that  $1 < \alpha \leq 2$ . The Riemann-Liouville derivatives being singular at the domain boundaries, they are sometimes replaced by space-fractional Caputo deriva-

tives of order  $\alpha$ :

$${}^C D_x^\alpha v(x) = \frac{1}{\Gamma(2-\alpha)} \int_a^x \frac{\frac{\partial^2 v(y)}{\partial y^2}}{(x-y)^{\alpha-1}} dy,$$

$${}^C D_b^\alpha v(x) = \frac{1}{\Gamma(2-\alpha)} \int_x^b \frac{\frac{\partial^2 v(y)}{\partial y^2}}{(y-x)^{\alpha-1}} dy.$$

Unlike Riemann-Liouville derivatives, Caputo derivatives are not singular on the domain boundaries. That feature makes them particularly appealing for non-local numerical methods, like the PS or RBF methods, as most basis functions take a non-zero value on the boundary. Zhang et al. have also shown that Riemann-Liouville derivatives could cause mass-balance errors on bounded domains [36]. That is not the case with Caputo derivatives as they naturally allow prescribed-flux boundary conditions, which are required for many applications. The RBF methods presented here can be implemented for any type of linear fractional derivative. We give the necessary details for left and right Riemann Liouville and Caputo fractional derivatives.

The paper is organized as follows. Section 2 introduces the RBF method as both an interpolation and a numerical discretization technique, while Section 3 focuses on discretizing fractional differential operators. The RBF-QR is summarized and adapted to the particular problem of discretizing fractional differential operators in Section 4. Numerical results are presented in Section 5 and a conclusion is given in Section 6.

## 2 RBF methodology

Rolland Hardy introduced the RBF methodology in 1971, when he suggested the multiquadric (MQ) method, a meshless interpolation technique using the MQ radial function [37]. This method was popularized in 1982 by Richard Franke with his report on 32 of the most commonly used interpolation methods [38]. He subjected those methods to thorough tests, and found the MQ method overall to be the best one. Franke also conjectured the unconditional non-singularity of the interpolation matrix associated with the MQ radial function, but it was not until a few years later, in 1986, that Charles Micchelli [39] was able to prove it, making use of work by Schoenberg from the 30s and 40s. The main feature of the MQ method is that the interpolant is a linear combination of translations of a basis function which only depends on the Euclidean distance from its center. This basis function is therefore radially symmetric with respect to its center. The MQ method was generalized to other radial functions, such as the thin plate spline or the gaussian, and the

| Name of RBF                     | Abbreviation | Definition                                                                                     |
|---------------------------------|--------------|------------------------------------------------------------------------------------------------|
| <b>Smooth, global</b>           |              |                                                                                                |
| Multiquadric                    | MQ           | $\sqrt{1 + (\varepsilon r)^2}$                                                                 |
| Inverse multiquadric            | IMQ          | $\frac{1}{\sqrt{1 + (\varepsilon r)^2}}$                                                       |
| Gaussian                        | GA           | $e^{-(\varepsilon r)^2}$                                                                       |
| Bessel                          | BE           | $\frac{J_{\frac{d}{2}-1}(\varepsilon r)}{(\varepsilon r)^{\frac{d}{2}-1}}, d = 1, 2, 3, \dots$ |
| <b>Piecewise smooth, global</b> |              |                                                                                                |
| Cubic                           | CU           | $ r ^3$                                                                                        |
| Thin plate spline               | TPS          | $r^2 \ln  r $                                                                                  |

Table 1

Definitions of some types of radial functions. The shape parameter  $\varepsilon$  controls their “flatness”.

method was called the Radial Basis Function method.

### 2.1 The form of an RBF interpolant

In 1D, the basic RBF interpolant takes the form

$$s(x) = \sum_{i=1}^N \lambda_i \phi(|x - x_i|). \quad (2)$$

In order for it to take the values  $f_i$  at locations  $x_i$ ,  $i = 1, 2, \dots, n$ , the expansion coefficients  $\lambda_i$  need to satisfy

$$A \vec{\lambda} = \vec{f} \quad (3)$$

where the entries of the matrix  $A$  are  $A_{i,j} = \phi(|x_i - x_j|)$ . We denote the numerical use of (3) followed by (2) as “RBF-Direct”. Out of all the most commonly used global radial functions  $\phi(r)$  listed in Table 1, we will only consider the smooth ones, because they give rise to spectrally accurate function representations, while the piecewise smooth radial functions only produce algebraically accurate representations. In this study, we will concentrate our attention on the GA radial function. The GA and the whole Bessel (BE) class of functions are the only radial functions so far, for which the RBF-QR method can be adapted in 1, 2 and 3 dimensions [33]. The parameter  $\varepsilon$ , included in all but the piecewise smooth global cases CU and TPS, is known as the *shape parameter*.

## 2.2 RBFs to discretize differential operators

The RBF method was originally introduced to solve interpolation problems, however it was eventually used, first by Ed Kansa in the 90s [40,41], as a discretization technique in the context of solving partial differential equations. Kansa's method consists in solving time dependent PDEs using the method of lines. The differential operators are discretized via RBFs, by developing differentiation matrices in space, and the resulting system of linear ODEs is solved in time using a standard ODE solver.

We wish to find a matrix  $D$  that discretizes the continuous differential operator  $L$  with an RBF expansion. We define an RBF expansion such that it takes the function value  $f_i$  at each node  $x_i$ . We thus require that

$$\sum_{j=1}^N \lambda_j \phi(|x_i - x_j|) = f_i, \quad (4)$$

for all  $x_i$ . It leads to the matrix equation  $A\vec{\lambda} = \vec{f}$ . Analytically applying the differential operator to the radial function gives

$$\sum_{j=1}^N \lambda_j L\phi(|x - x_j|)_{x=x_i} = g_i, \quad (5)$$

where  $g_i$  is the value of the underlying function's derivative at each  $x_i$ . Thus  $B\vec{\lambda} = \vec{g}$  in matrix form, where  $B_{i,j} = L\phi(|x - x_j|)_{x=x_i}$ . The collocation matrix  $A$  being unconditionally nonsingular [42], we can eliminate the expansion coefficient vector  $\vec{\lambda}$  and obtain  $\vec{g} = BA^{-1}\vec{f}$ . The differentiation matrix  $D = BA^{-1}$  thus gives an RBF discretization of  $L$ .

## 3 Evaluating fractional Derivatives of Radial Functions

Finding a closed form analytic expression for the fractional derivative of a function can be challenging. We are often bound to having to represent functions as Taylor series expansions before applying the fractional derivative operator term by term. Here we derive these series expansions for the Riemann-Liouville and Caputo fractional derivatives of the gaussian radial function.

We can thus expand the gaussian in a MacLaurin series and apply the fractional derivative operator  $D^\alpha$  term by term as follows

$$D^\alpha e^{-\varepsilon^2|x-x_i|^2} = \sum_{j=0}^{\infty} \frac{(-1)^j \varepsilon^{2j}}{j!} \left( \sum_{k=0}^{2j} \frac{(2j)!(-1)^k x_i^{2j-k}}{k!(2j-k)!} D^\alpha x^k \right)$$

| Operator Type     | $D^\alpha$          | $D^\alpha x^k, k \in \mathbb{N}, 1 < \alpha \leq 2$                                                                                   |
|-------------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Riemann Liouville | ${}_a D_x^\alpha$   | $\frac{k!(x-a)^{-\alpha}\Gamma(\alpha+1)}{\Gamma(k+1-\alpha)} \sum_{i=0}^k \frac{x^{k-i}(-a)^i}{i!\Gamma(\alpha-i+1)}$                |
|                   | ${}_x D_b^\alpha$   | $\frac{k!(b-x)^{-\alpha}\Gamma(\alpha+1)}{\Gamma(k+1-\alpha)} \sum_{i=0}^k \frac{x^{k-i}(-b)^i}{i!\Gamma(\alpha-i+1)}$                |
| Caputo            | ${}_a^C D_x^\alpha$ | $\frac{(-1)^k k!(x-a)^{2-\alpha}\Gamma(\alpha-1)}{\Gamma(k+1-\alpha)} \sum_{i=0}^{k-2} \frac{(-x)^{k-2-i} a^i}{i!\Gamma(\alpha-i-1)}$ |
|                   | ${}_x^C D_b^\alpha$ | $\frac{(-1)^k k!(b-x)^{2-\alpha}\Gamma(\alpha-1)}{\Gamma(k+1-\alpha)} \sum_{i=0}^{k-2} \frac{(-x)^{k-2-i} b^i}{i!\Gamma(\alpha-i-1)}$ |

Table 2

Explicit forms of  $D^\alpha x^k$ , where  $D^\alpha$  can be either a left/right Riemann Liouville or Caputo fractional derivative of order  $\alpha$ , for  $k \in \mathbb{N}$  and  $1 < \alpha \leq 2$ .

, and truncate the infinite sum once the terms are smaller in magnitude than machine precision. An expression for  $D^\alpha x^k$  is given in Table 2 for left and right Riemann Liouville and Caputo derivatives. Once the fractional derivatives of the Gaussian radial function have been computed, we can build the matrix  $B$  defined in section 2.2 for the fractional-order diffusion term defined in Eq. (1) such that

$$B_{i,j} = \mathcal{D}_\beta^\alpha \phi(|x - x_j|)_{x=x_i}, \quad (6)$$

where  $\mathcal{D}_\beta^\alpha$  is a shorthand notation for the the combination of left and right Riemann-Liouville or Caputo fractional derivatives used in Eq. (1), *i.e.*

$$\mathcal{D}_\beta^\alpha \phi = \frac{1+\beta}{2} {}_a D_x^\alpha \phi + \frac{1-\beta}{2} {}_x D_b^\alpha \phi, \quad (7)$$

$$\text{or } \mathcal{D}_\beta^\alpha \phi = \frac{1+\beta}{2} {}_a^C D_x^\alpha \phi + \frac{1-\beta}{2} {}_x^C D_b^\alpha \phi. \quad (8)$$

The differentiation matrix can then be expressed as  $D = BA^{-1}$ .

## 4 The RBF-QR method applied to fractional differential equation problems

The  $C_\infty$  radial functions have an intrinsic shape parameter which controls the steepness of the radial function translates. The steeper the radial function, the better conditioned the interpolation problem becomes, because the function to interpolate is an element of the function space spanned by highly linearly independent functions. However, the rate at which the interpolant converges to the underlying function is much lower than when the shape parameter is smaller, thus when the radial functions are flatter. However the interpolant is tricky to compute accurately in that instance since the problem becomes highly ill-conditioned when the radial functions are flatter. Fornberg and Driscoll proved that in the limit of  $\varepsilon \rightarrow 0$ , the RBF method reproduces the interpolant generated by any pseudospectral method with matching node set and periodicity [30]. For example, on a periodic domain with equispaced nodes, the RBF interpolant will coincide with the Fourier series. On a 1D non-periodic domain, the RBF interpolant converges to the Lagrange interpolating polynomial in the flat radial function limit. Since the interpolant is well defined in that limit, it became clear that the direct approach is problematic, and not the underlying interpolation problem. In [43], Fornberg and Wright develop and introduce the contour Padé method: the first method capable of stably computing the RBF interpolant in the small shape parameter range. However the algorithm is limited in the number of nodes and is highly sensitive to a set of parameters. The RBF-QR method was introduced in 2007 by Fornberg and Piret [1] and was the second algorithm developed to compute RBF interpolants for small values of  $\varepsilon$ . It was first designed for the interpolation of functions on the surface of the unit sphere, and was later modified for the Gaussian radial function to 1, 2 and 3-dimensions [33]. The RBF-QR method is superior to the Contour-Padé method in that there is no restriction on the number of nodes and no extra set of parameters. Next, we give a brief overview of the RBF-QR method.

### 4.1 The RBF-QR method for interpolation

One of the well-known problems with the Direct-RBF method is the increasing ill-conditioning one encounters when  $N$  gets larger. In [31], it is shown that one can characterize this increase of ill-conditioning in function of the type of domain, and of the node distribution. The pattern of the ill-conditioning increase is given by the number of  $O(\varepsilon^{2j})$  eigenvalues of the collocation matrix, which they display in their Table 5.1 for several types of domains. For example, in a 1D non-periodic setting, we will find that the RBF collocation matrix has 1 eigenvalue of  $O(\varepsilon^0)$ , 1 eigenvalue of  $O(\varepsilon^2)$ , 1 eigenvalue of  $O(\varepsilon^4)$ , etc. This



means also that in this specific circumstance, the ill-conditioning will increase with  $N$  as the condition number  $\kappa \approx O(\varepsilon^{-2N})$ . In [33], equation (4.19) gives the following series expansion for the Gaussian radian function translate in  $1D$  in terms of the Chebyshev polynomials  $T_j(x)$ :

$$e^{-\varepsilon^2(x-x_k)^2} = \sum_{j=0}^{\infty} \varepsilon^{2j} c_j(x_k) e^{-\varepsilon^2 x^2} T_j(x),$$

where  $c_j(x_k) = \frac{2t_j}{j!} e^{-\varepsilon^2 x_k^2} x_k^j {}_0F_1([], j+1, \varepsilon^4 x_k^2)$ , with  $t_0 = \frac{1}{2}$  and  $t_j = 1$  for  $j > 0$  and  ${}_mF_n(z)$  is the hypergeometric function. We can truncate the series after  $p$  terms and express the radial function translates as in the following matrix equation

$$\underbrace{\begin{pmatrix} \phi(|x-x_1|) \\ \phi(|x-x_2|) \\ \vdots \\ \phi(|x-x_n|) \end{pmatrix}}_{\equiv \Phi(x)} = \underbrace{\begin{pmatrix} c_0(x_1) & \varepsilon^2 c_1(x_1) & \varepsilon^4 c_2(x_1) & \cdots & \varepsilon^{2p} c_p(x_1) \\ c_0(x_2) & \varepsilon^2 c_1(x_2) & \varepsilon^4 c_2(x_2) & \cdots & \varepsilon^{2p} c_p(x_2) \\ \vdots & & & \ddots & \vdots \\ c_0(x_n) & \varepsilon^2 c_1(x_n) & \varepsilon^4 c_2(x_n) & \cdots & \varepsilon^{2p} c_p(x_n) \end{pmatrix}}_{\equiv M} \cdot \underbrace{\begin{pmatrix} e^{-\varepsilon^2 x^2} T_0(x) \\ e^{-\varepsilon^2 x^2} T_1(x) \\ e^{-\varepsilon^2 x^2} T_2(x) \\ \vdots \\ e^{-\varepsilon^2 x^2} T_p(x) \end{pmatrix}}_{\equiv \Xi(x)},$$

A  $QR$ -factorization can then be performed on the matrix  $M$ , with a unitary matrix  $Q$  and an upper triangular rectangular matrix  $R$ . Since  $R$  is formed by taking linear combinations of the rows of  $M$ , it retains the same structure of powers of  $\varepsilon$  factors as  $M$ . We can further take advantage of the upper-triangular structure of  $R$  to extract into a matrix  $E$ , the common  $\varepsilon$  factors of each row terms:

$$M = Q \cdot \underbrace{\begin{pmatrix} m_{1,1} & \varepsilon^2 m_{1,2} & \varepsilon^4 m_{1,3} & \cdots & \varepsilon^{2p} m_{1,p} \\ 0 & \varepsilon^2 m_{2,2} & \varepsilon^4 m_{2,3} & \cdots & \varepsilon^{2p} m_{2,p} \\ \vdots & \ddots & & & \vdots \\ 0 & & 0 & & \varepsilon^{2p} m_{n,p} \end{pmatrix}}_{\equiv R} \quad (9)$$

$$= Q \cdot \underbrace{\begin{pmatrix} 1 & & & & \\ & \varepsilon^2 & & & \\ & & \ddots & & \\ & & & \varepsilon^{2(n-1)} & \end{pmatrix}}_{\equiv E} \cdot \underbrace{\begin{pmatrix} m_{1,1} & \varepsilon^2 m_{1,2} & \varepsilon^4 m_{1,3} & \cdots & \varepsilon^{2p} m_{1,p} \\ 0 & m_{2,2} & \varepsilon^2 m_{2,3} & \cdots & \varepsilon^{2(p-1)} m_{2,p} \\ \vdots & \ddots & & & \vdots \\ 0 & & 0 & & \varepsilon^{2(p-n+1)} m_{n,p} \end{pmatrix}}_{\equiv \tilde{R}} \quad (10)$$

where  $m_{i,j}$  are the entries of the matrix  $R$  obtained with the QR factorization of matrix  $M$ .

One can thus rewrite the RBF collocation matrix as follows:

$$A = \begin{pmatrix} \vec{\Phi}^T(x_1) \\ \vec{\Phi}^T(x_2) \\ \vdots \\ \vec{\Phi}^T(x_n) \end{pmatrix} = \begin{pmatrix} \vec{\Xi}^T(x_1) \\ \vec{\Xi}^T(x_2) \\ \vdots \\ \vec{\Xi}^T(x_n) \end{pmatrix} \cdot \tilde{R}^T \cdot E \cdot Q^T$$

$$\equiv \Xi^T \cdot \tilde{R}^T \cdot E \cdot Q^T$$

The interpolation problem can be rewritten as well. We can now incorporate the vector  $\vec{\lambda}$  and the diagonal matrix  $E$  inside  $\vec{\beta}$ , the expansion coefficient vector associated with the new basis. The matrix  $E$  contains *exactly* the sequence of eigenvalues described above. We have thus analytically removed these eigenvalues out of the interpolation equation and hence removed the  $\varepsilon = 0$  singularity. Figure 1 shows the basis terms for both RBF-Direct and the RBF-QR approaches in the case of a small  $\varepsilon$  value. The sequence of operations defining the RBF-QR method can be summarized as follows:

$$A \cdot \vec{\lambda} = \vec{f} \implies \underbrace{(\Xi^T \cdot \tilde{R}^T)}_{\tilde{A}} \cdot \underbrace{(E^T \cdot Q^T \cdot \vec{\lambda})}_{\vec{\beta}} = \vec{f}$$

$$\implies \tilde{A} \cdot \vec{\beta} = \vec{f}$$

We will now apply this technique to stably discretize fractional differentiation operators using RBFs with small  $\varepsilon$  values.

#### 4.2 Fractional Derivative of the RBF-QR basis in 1D

Let us now compute the fractional derivatives of the RBF-QR basis functions. Since we want to compute these derivatives in an arbitrary domain  $[a, b]$ , we must first translate and scale the terms of the new basis from the RBF-QR domain  $[-1, 1]$  to  $[a, b]$ . Assuming that  $y \in [-1, 1]$  and  $x \in [a, b]$ , the mapping simply reads  $y = \frac{2}{b-a}x - \frac{a+b}{b-a}$ . It should be noted that this transformation impacts the value of  $\varepsilon$  for which the RBF-QR method will converge. For  $y \in [-1, 1]$ , the power series in  $\varepsilon_y$ , key feature of the algorithm, converges for  $\varepsilon_y < 1$ . However, with the above linear transformation,

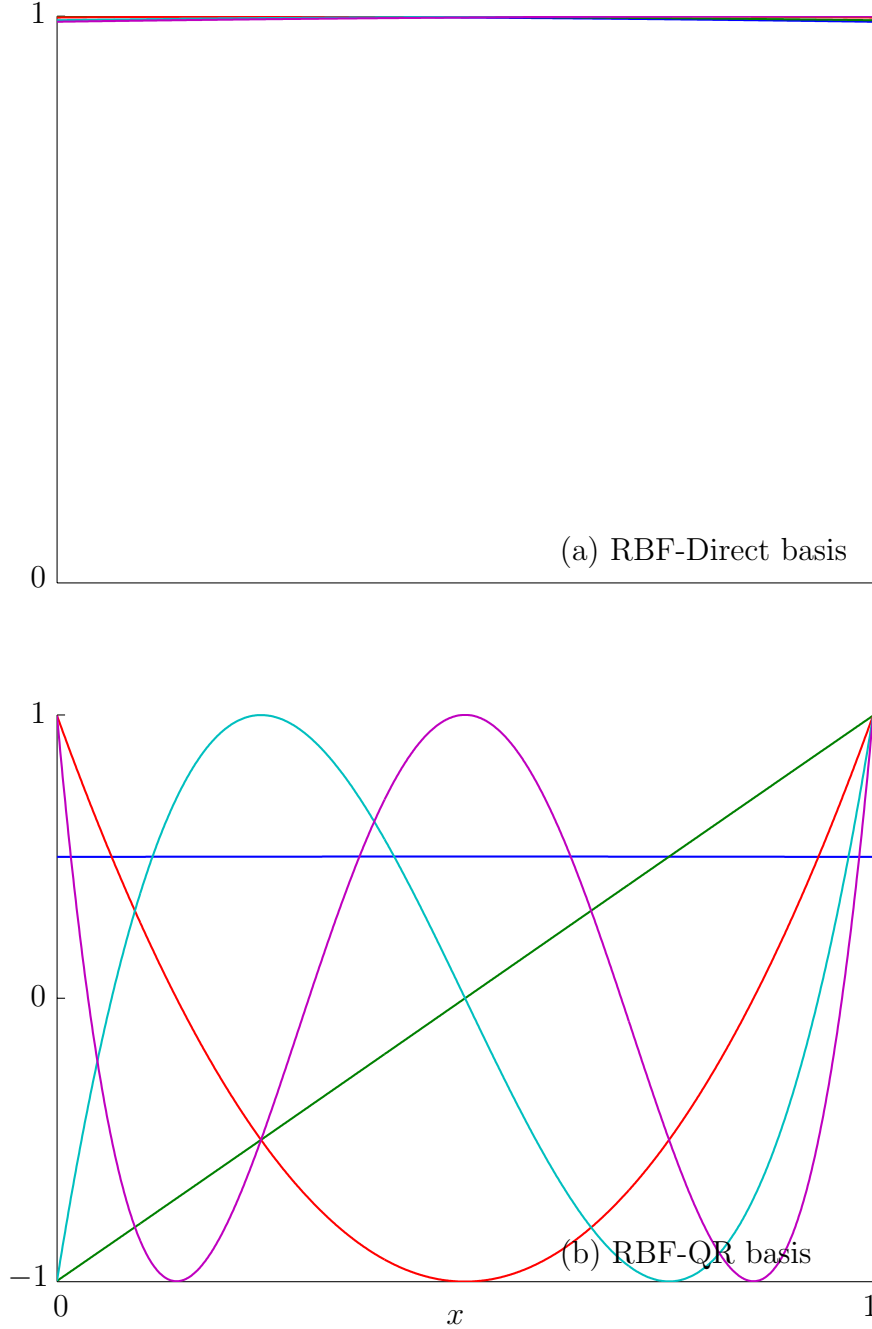


Fig. 1. (a) RBF-Direct and (b) RBF-QR bases spanning the same function space for  $N = 5$  and for a small shape parameter value ( $\varepsilon = 0.1$ ). As the value of  $\varepsilon$  decreases, all the RBF-Direct basis functions become undistinguishable while the RBF-QR basis functions converge toward Chebyshev polynomials. The removable singularity at  $\varepsilon = 0$  has been analytically extracted in the RBF-QR basis.

$$\begin{aligned} \varepsilon_y^2 |y - y_i|^2 &= \left( \frac{2\varepsilon_y}{b-a} \right)^2 |x - x_i|^2 \\ &= \varepsilon^2 |x - x_i|^2, \end{aligned}$$

which implies that  $\varepsilon = \frac{2\varepsilon y}{b-a}$  and that the algorithm will thus converge for  $\varepsilon < \frac{2}{b-a}$ .

The terms of the new basis are linear combinations of  $\Xi_j(y) = e^{-\varepsilon^2 y^2} T_j(y)$  with  $y \in [-1, 1]$ . We can write these functions in terms of  $x$  as  $\Xi_j(y) = e^{-\varepsilon^2 (x - \frac{a+b}{2})^2} T_j\left(\frac{2}{b-a}x - \frac{a+b}{b-a}\right)$ , where  $x \in [a, b]$ . We show how to take the fractional derivative of these functions in what follows. Since no closed form expression for the fractional derivative of  $\Xi_j\left(\frac{2}{b-a}x - \frac{a+b}{b-a}\right)$  is readily available, we start by finding its MacLaurin expansion and then apply the fractional derivative operator term by term. We obtain

$$\begin{aligned} D^\alpha \Xi_j & \left( \frac{2x}{b-a} - \frac{a+b}{b-a} \right) \\ & = \sum_{l=0}^{\infty} (-\varepsilon^2)^j \sum_{i=0}^j \frac{2^i c_i}{(b-a)^i} \sum_{k=0}^{i+2l} \left( \frac{a+b}{-2} \right)^{i+2l-k} \frac{(i+2l)!}{k!(i+2l-k)!} D^\alpha x^k, \end{aligned}$$

in which  $D^\alpha$  can refer to a linear fractional derivative of any type. We give formulas for  $D^\alpha x^k$  in Table 2 for left and right Riemann Liouville and Caputo derivatives.

We can now build the matrix  $B$  defined by Eq. (6), where  $\phi$  is now an RBF-QR basis function. That matrix can be expressed as follows:

$$B = \begin{pmatrix} \mathcal{D}_\beta^\alpha \vec{\Xi}^T(x)_{x=x_1} \\ \mathcal{D}_\beta^\alpha \vec{\Xi}^T(x)_{x=x_2} \\ \vdots \\ \mathcal{D}_\beta^\alpha \vec{\Xi}^T(x)_{x=x_n} \end{pmatrix} \cdot \tilde{R}^T \cdot E \cdot Q^T = \mathcal{D}_\beta^\alpha \vec{\Xi}^T \cdot \tilde{R}^T \cdot E \cdot Q^T,$$

where  $\mathcal{D}_\beta^\alpha$  again represents the combination of left and right Riemann-Liouville or Caputo fractional derivatives used in Eq. (1). The differentiation matrix  $D = BA^{-1}$  can then be written as follows:

$$\begin{aligned} D & = \left( \mathcal{D}_\beta^\alpha \vec{\Xi}^T \cdot \tilde{R}^T \cdot E \cdot Q^T \right) \cdot \left( \vec{\Xi}^T \cdot \tilde{R}^T \cdot E \cdot Q^T \right)^{-1} \\ & = \mathcal{D}_\beta^\alpha \vec{\Xi}^T \cdot \tilde{R}^T \cdot E \cdot Q^{-1} \cdot Q \cdot E^{-1} \cdot \left( \vec{\Xi}^T \cdot \tilde{R}^T \right)^{-1} \\ & = \left( \mathcal{D}_\beta^\alpha \vec{\Xi}^T \cdot \tilde{R}^T \right) \cdot \left( \vec{\Xi}^T \cdot \tilde{R}^T \right)^{-1} \end{aligned}$$

## 5 Numerical Results

In this section, we perform some numerical experiments to compare the RBF-Direct and RBF-QR methods, and highlight the spectral accuracy of the latter. The first test case focuses only on the discretization of the fractional differential operator while the second considers the solution of a fractional-order diffusion equation.

### 5.1 1D Fractional Differential Operator

We show the quality of our method for discretizing fractional derivatives by comparing the true value of the following function's derivative with our RBF-Direct and RBF-QR approximations. Let us consider the function  $f(x) = \cos(jx)$  with  $x \in [0, 1]$  for which the exact left-sided Caputo derivative reads:

$${}_0^C D_x^\alpha \cos(jx) = (ij)^{[\alpha]} x^{[\alpha]-\alpha} {}_1F_1(1, [\alpha]-\alpha+1, ijx) + \frac{{}_1F_1(1, [\alpha]-\alpha+1, -ijx)}{2\Gamma([\alpha]-\alpha+1)}$$

In order to compute the error, we have evaluated the RBF approximations to  ${}_0^C D_t^\alpha \cos(4x)$  with  $\alpha = 1.8$  on a set of a thousand equispaced nodes in  $[0, 1]$ . Figure 2 shows the relative max norm error computed at these nodes. The error is displayed in function of  $N$ , the number of centers used in the RBF expansion, and for different values of  $\varepsilon$ . The Direct-RBF and RBF-QR methods are used depending on this value. It should be noted that the mapping from  $[-1, 1]$  to  $[0, 1]$  has allowed us to use the RBF-QR method for values of  $\varepsilon \in [0, 2]$ . Figure 2 (a) shows the error when the  $N$  nodes are equispaced, while Figure 2 (b) shows the error produced when the  $N$  nodes have a Chebyshev distribution, which provides a denser distribution on the edges of the domain.

It can be seen that the error decays exponentially with  $N$ , at a rate that depends on the value of the shape parameter  $\varepsilon$ . For both node distributions, a faster rate of convergence occurs for smaller values of  $\varepsilon$ , which are achievable only with the RBF-QR method. Figure 2 highlights the error produced when  $\varepsilon = 1.8$  for both the Direct and QR methods. This shape parameter value is large enough for us to use the RBF-Direct method, and small enough to use the RBF-QR method. As expected, both methods coincide exactly for small values of  $N$ , as their respective system is well-conditioned. However, when  $N \approx 14$ , the RBF-Direct becomes too ill-conditioned to give an accurate result and error stops decreasing. The RBF-QR method however continues to converge towards the exact solution. We notice also in Figure 2 (a) that the RBF-QR error grows when  $N \geq 18$  for  $\varepsilon = 1.0$  and  $N \geq 22$  for  $\varepsilon = 1.8$ . As mentioned previously, the RBF interpolant converges towards a PS (and thus

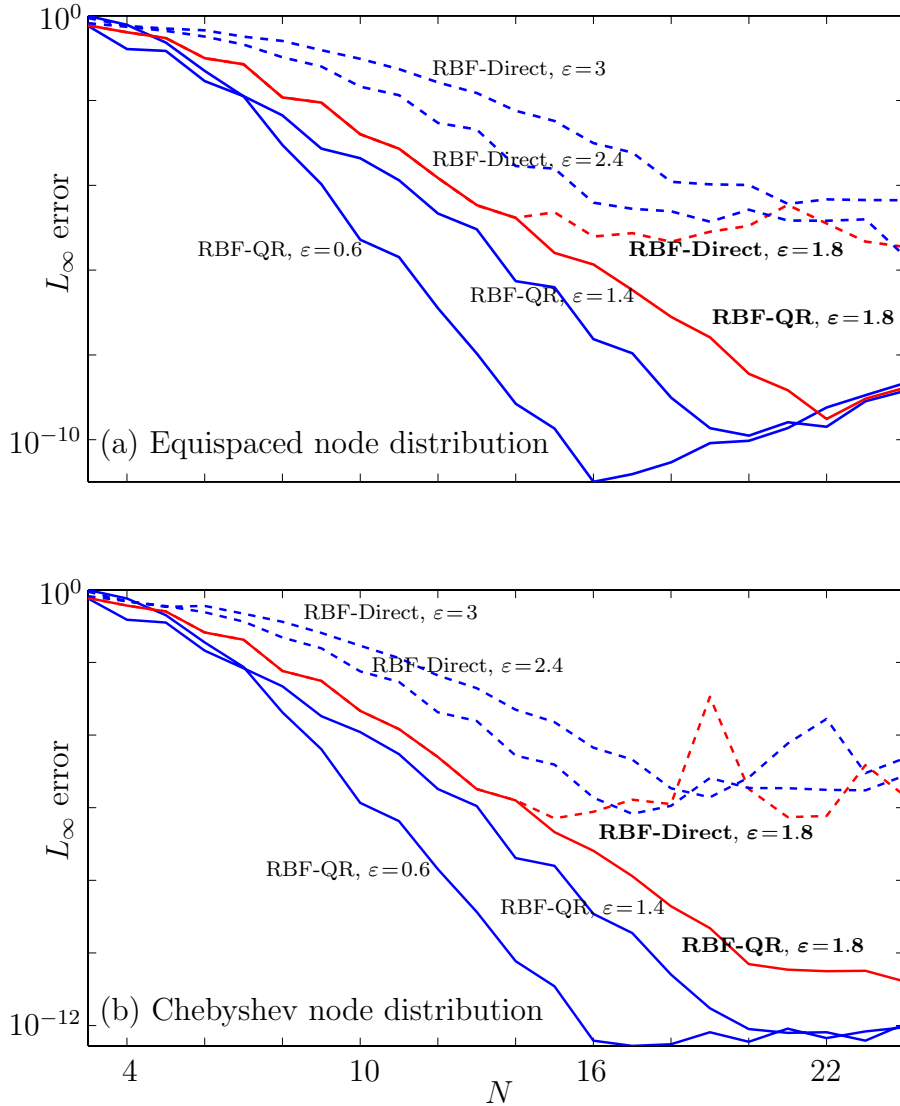


Fig. 2. Relative max-norm error in computing  ${}_0^C D_x^{1.8} \cos(4x)$  with respect to the number of nodes  $N$ , for several values of  $\varepsilon$  and for (a) equispaced and (b) Chebyshev node distributions. We use either the RBF-Direct (dashed line) or RBF-QR method (solid line) depending on the value of  $\varepsilon$ . It can be seen that the curves obtained by both methods for  $\varepsilon = 1.8$  (red line) coincide up until  $N \approx 14$ , at which point the RBF-Direct system becomes too ill-conditioned to produce an accurate result.

polynomial) interpolant as  $\varepsilon \rightarrow 0$ . The resulting interpolation can therefore be crippled by the spurious oscillations when the nodes are equispaced [44]. This issue is resolved when the nodes have a Chebyshev distribution as is shown in Figure 2 (b).

Fig. 3 shows the relative maximum-norm error versus the value of the shape parameter  $\varepsilon$  for different values of  $N$ . Fornberg and Zuev [31] showed that the error has a very common structure as it decreases when the radial functions get flatter (*i.e.* as  $\varepsilon \rightarrow 0$ ) until it reaches a minimum for some “optimal” shape

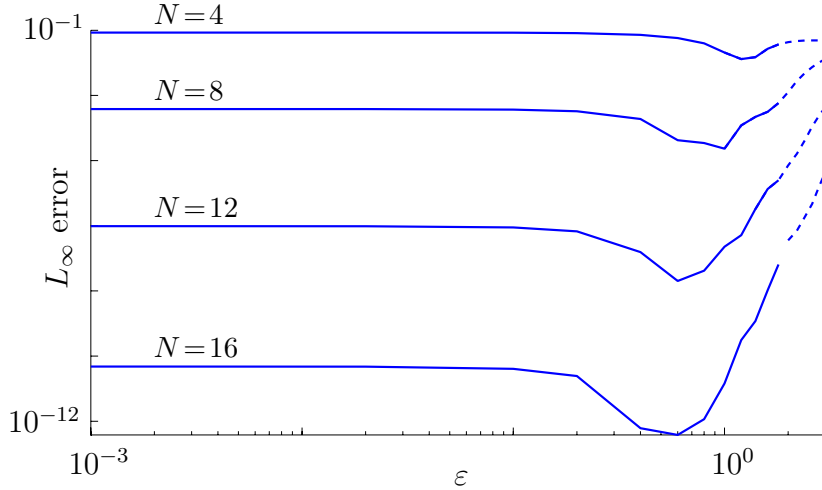


Fig. 3. Relative max-norm error in computing  ${}_0^C D_x^{1.8} \cos(4x)$  with respect to  $\varepsilon$ , for several Chebyshev distributions of  $N$  nodes. The error on the RBF-Direct solution (dashed line) converges only for large values of  $\varepsilon$  and starts diverging as soon as  $\varepsilon$  becomes too small. The error on the RBF-QR solution (solid line) converges as soon as  $\varepsilon < 2$ .

parameter value. The error then increases again as the method converges to the PS method. It can be seen that

- For a same number of nodes  $N$ , the error curves computed via the RBF-Direct and the RBF-QR methods coincide exactly in the intersection of the range of  $\varepsilon$  values for which each method is valid.
- The RBF method is capable of a better accuracy than the PS method.
- The error computed via the RBF-Direct method stagnates because of the ill-conditioning it faces when  $\varepsilon$  is too small or when  $N$  grows.

## 5.2 1D Fractional Diffusion Equation

As a second test case, we consider the following benchmark problem introduced by Sousa [45]. It consists in finding  $f(x, t)$  such that

$$\frac{\partial f(x, t)}{\partial t} = d(x) {}_0 D_x^\alpha f(x, t) + q(x, t) \quad \text{for } x \in [0, 1] \text{ and } t > 0, \quad (11)$$

with  $d(x) = \frac{\Gamma(5-\alpha)}{24} x^\alpha$ ,  $q(x, t) = -2e^{-t}x^4$ ,  $f(x, 0) = x^4$ ,  $f(0, t) = 0$  and  $f(1, t) = e^{-t}$ . In that case, the exact solution of (11) reads:

$$f(x, t) = e^{-t}x^4.$$

Eq. (11) involves only a left-sided Riemann-Liouville fractional derivative and thus amounts to set  $\beta = 1$  in Eq. (1) and add the reaction term  $q(x, t)$ .

Note that both the solution slope and value vanish on the left boundary and hence the left Riemann-Liouville and Caputo derivatives of  $f(x, t)$  are totally equivalent. Eq. (11) is discretized in time with a Crank-Nicolson scheme and a very small time step value is selected such that time discretization errors are negligible.

Fig. 4 shows the relative max-norm error between the numerical and exact solutions at  $t = 1$  with respect to the number of nodes  $N$ , both for equispaced and Chebyshev-distributed RBF centers. It can be seen that the error decays exponentially with the number of centers, however distributed. The rate of convergence increases as the shape parameter  $\varepsilon$  decreases. However, for an equispaced node distribution, the error starts increasing after having reached a minimum value. This is again because RBFs converge to polynomials as  $\varepsilon \rightarrow 0$ . Hence, they tend to suffer from spurious oscillations when the nodes are equispaced and when  $\varepsilon$  becomes too small [44]. A few strategies have been developed to circumvent this problem and are presented in [46]; one can either cluster nodes in problematic regions (the edges in this case) or use other techniques such as the cubic splines' Not-A-Knot. We show in Fig. 4b the error obtained when the nodes are clustered at the edges with a Chebyshev node distribution.

Fig. 5 shows the relative maximum-norm error with respect to the shape parameter  $\varepsilon$ , for different numbers of RBF nodes  $N$ , placed according to a Chebyshev distribution. We used the RBF-QR and RBF-Direct algorithms depending on the value of  $\varepsilon$ . It can be seen that the RBF-QR and RBF-Direct methods give equivalent results when the shape parameter is large enough for the RBF-Direct method to produce an accurate result, and low enough for the RBF-QR method to converge. Further, the error converges to machine precision for all values of  $N$  with the RBF-QR method. Since the exact solution is a 4th-order polynomial, it is exactly represented in the limit of  $\varepsilon \rightarrow 0$  as soon as  $N \geq 5$ .

## 6 Conclusion

The global character of radial functions is often seen as a negative quality of RBFs because of the computational cost and the ill-conditioning problems it causes. Globality, however, is a key feature of fractional derivatives. Only the integer-order derivative is local in essence. We have shown in this work that RBFs can be used to efficiently capture this feature.

The RBF method can be seen as a compromise between the PS and FE methods, depending on whether the RBF radial functions are flatter or steeper. It is not surprising that the best quality of the operator is found for small values



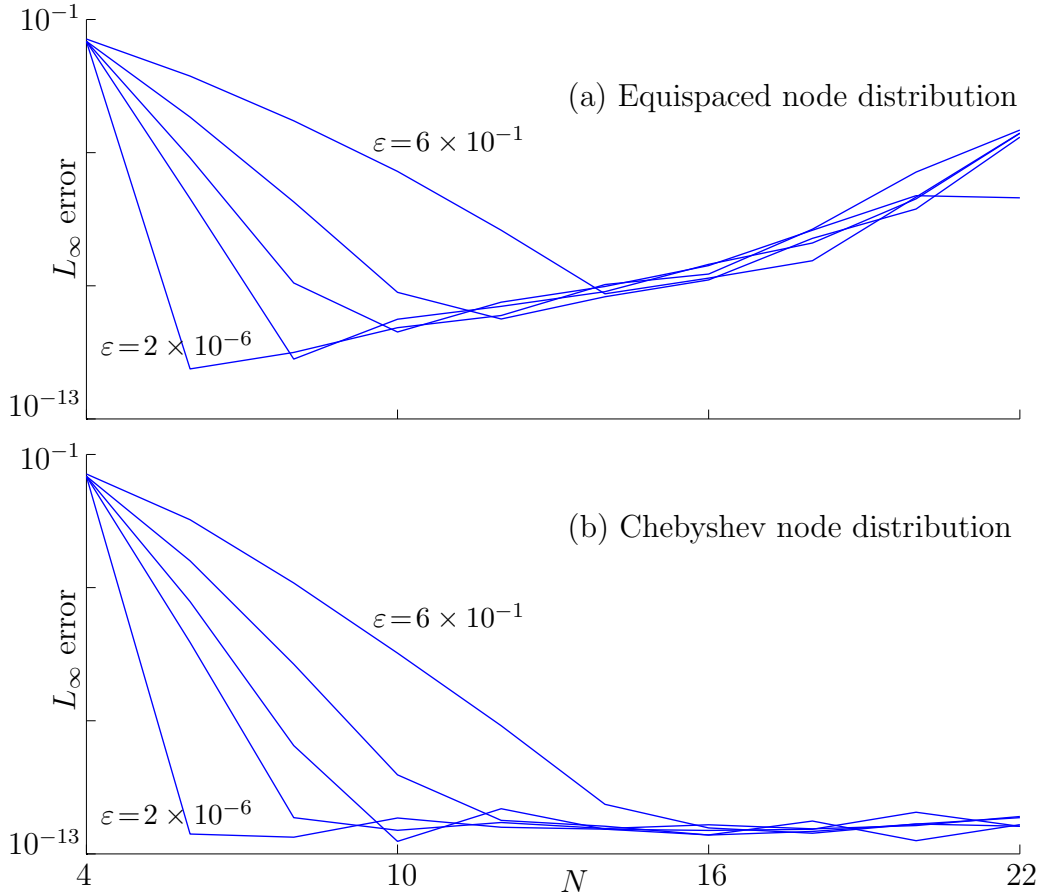


Fig. 4. Relative max-norm error in solving Eq. (11) with respect to the number of nodes  $N$  and for  $\varepsilon = 2 \times 10^{-6}, 9 \times 10^{-3}, 3 \times 10^{-2}, 1 \times 10^{-1}, 6 \times 10^{-1}$  (in ascending order on the graphs). Only the RBF-QR method has been used since the values of  $\varepsilon$  are too small to lead to any sensible result with the RBF-Direct method. The error for an equispaced node distribution (a) diverges after having reached a minimum because of the instability of polynomial approximation of analytic functions on equispaced grids [44]. That problem is circumvented on a Chebyshev node distribution (b).

of the shape parameter, corresponding to relatively flat radial functions which exhibit the strongest global character. It is in that range, however, that the ill-conditioning is the worst. We used the RBF-QR method to analytically remove this ill-conditioning and compute differentiation matrices to discretize fractional derivatives with spectral accuracy. We have also shown that even in the small shape parameter range, RBFs allow us to bypass problems, such as the spurious oscillation on equispaced nodes, that are specific to PS methods, by adapting the node distribution to the problem at hand.

Fractional derivatives are generally impossible to analytically compute in closed form and the common discretization techniques are all limited in either their accuracy or their ill-conditioning. We show, with the help of the RBF-QR algorithm, how RBFs offer an excellent approach for computing fractional derivatives and for solving fractional differential equations, both stably and

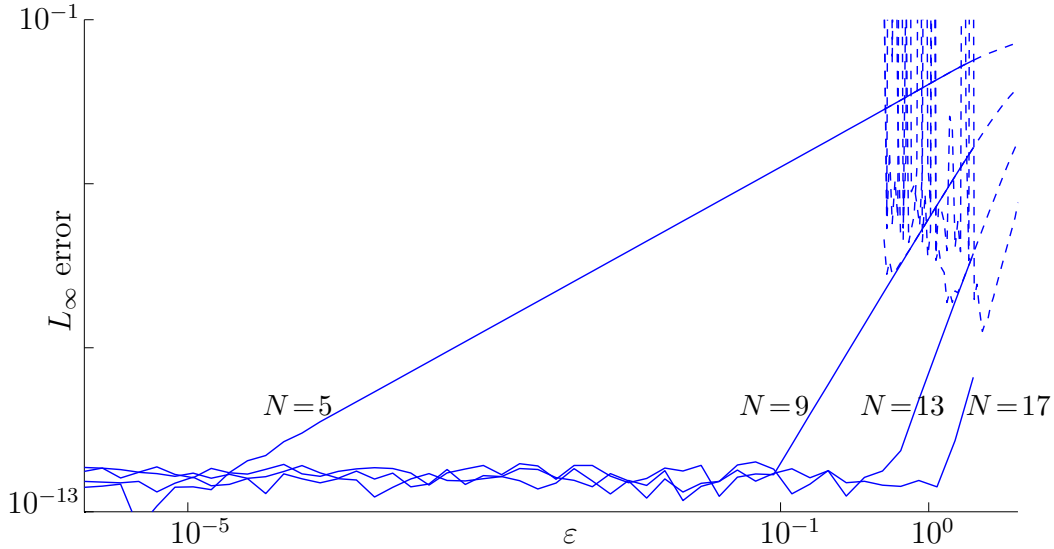


Fig. 5. Relative max-norm error versus  $\varepsilon$ , for several numbers of nodes  $N$ , with a Chebyshev distribution. The error on the RBF-Direct solution (dashed line) converges only for large values of  $\varepsilon$  and starts diverging as soon as  $\varepsilon$  becomes too small. The error on the RBF-QR solution (solid line) converges as soon as  $\varepsilon < 2$ .

accurately. In addition to that, RBFs easily allow extensions to higher dimensions. This will be the subject of a future study.

## 7 Acknowledgments

The authors wish to thank Bengt Fornberg for stimulating discussions and for his careful reading of the manuscript.

## References

- [1] B. Fornberg, C. Piret, A stable algorithm for flat radial basis functions on a sphere, *SIAM J. Sci. Comput* 30 (1) (2007) 60–80.
- [2] R. Metzler, J. Klafter, The random walk’s guide to anomalous diffusion: A fractional dynamics approach, *Physics Reports* 339 (2000) 1–77.
- [3] D.A. Benson, S.W. Wheatcraft, M.M. Meerschaert, Application of a fractional advection-dispersion equation, *Water Resources Research* 36 (2000) 1403–1412.
- [4] Y. Pachepsky, D. Timlin, W. Rawls, Generalized Richards’ equation to simulate water transport in unsaturated soils, *Journal of Hydrology* 272 (2003) 3–13.
- [5] Z.Q. Deng, J.L.M.P. de Lima, M.I.P. de Lima, V.P. Singh, A fractional dispersion model for overland solute transport, *Water Resources Research* 42:W03416.

- [6] D. del Castillo Negrete, B. A. Carreras, V. E. Lynch, Fractional diffusion in plasma turbulence, *Physics of Plasmas* 11 (8) (2004) 3854–3864.
- [7] D. del Castillo Negrete, B. A. Carreras, V. E. Lynch, Nondiffusive transport in plasma turbulence: A fractional diffusion approach, *Physical Review Letters* 94 (065003).
- [8] E. Scalas, R. Gorenflo, F. Mainardi, Fractional calculus and continuous-time finance, *Physica A: Statistical Mechanics and its Applications* 284 (2000) 376–384.
- [9] A. Cartea, D. del Castillo Negrete, Fractional diffusion models of option prices in markets with jumps, *Physica A: Statistical Mechanics and its Applications* 374 (2) (2007) 749–763.
- [10] D. Brockmann, Human mobility and spatial disease dynamics, in: H. G. Schuster (Ed.), *Reviews of Nonlinear Dynamics and Complexity*, Vol. 10, Wiley-VCH, 2009, pp. 1–24.
- [11] E. Hanert, E. Schumacher, E. Deleersnijder, Front dynamics in fractional-order epidemic models, *Journal of Theoretical Biology* 279 (1) (2011) 9–16, doi:10.1016/j.jtbi.2011.03.012.
- [12] S. Das, P.K. Gupta, A mathematical model on fractional Lotka-Volterra equations, *Journal of Theoretical Biology* 277 (1) (2011) 1–6.
- [13] E. Hanert, Front dynamics in a two-species competition model driven by Lévy flights, *Journal of Theoretical Biology* 300 (2012) 134–142, doi:10.1016/j.jtbi.2012.01.022.
- [14] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for fractional advection-diffusion flow equations, *Journal of Computational and Applied Mathematics* 172 (2004) 65–77.
- [15] M.M. Meerschaert, C. Tadjeran, Finite difference approximations for two-sided space-fractional partial differential equations, *Applied Numerical Mathematics* 56 (1) (2006) 80–90.
- [16] C. Tadjeran, M.M. Meerschaert, H.-P. Scheffler, A second-order accurate numerical approximation for the fractional diffusion equation, *Journal of Computational Physics* 213 (2006) 205–213.
- [17] C. Tadjeran, M.M. Meerschaert, A second-order accurate numerical method for the two-dimensional fractional diffusion equation, *Journal of Computational Physics* 220 (2007) 813–823.
- [18] Y. Lin, C. Xu, Finite difference/spectral approximations for the time-fractional diffusion equation, *Journal of Computational Physics* 225 (2) (2007) 1533 – 1552.
- [19] I. Podlubny, A. Chechkin, T. Skovranek, Y. Chen, B.M.V. Jara, Matrix approach to discrete fractional calculus II: Partial fractional differential equations, *Journal of Computational Physics* 228 (2009) 3137–3153.

- [20] M. Cui, Compact alternating direction implicit method for two-dimensional time fractional diffusion equation, *Journal of Computational Physics* 231 (6) (2012) 2621–2633.
- [21] G.J. Fix, J.P. Roop, Least square finite-element solution of a fractional order two-point boundary value problem, *Computers and Mathematics with Applications* 48 (2004) 1017–1033.
- [22] J.P. Roop, Computational aspects of FEM approximation of fractional advection dispersion equations on bounded domains in  $\mathbb{R}^2$ , *Journal of Computational and Applied Mathematics* 193 (2006) 243–268.
- [23] Q. Huang, G. Huang, H. Zhan, A finite element solution for the fractional advection-dispersion equation, *Advances in Water Resources* 31 (2008) 1578–1589.
- [24] H. Wang, K. Wang, T. Sircar, A direct  $O(N\log^2 N)$  finite difference method for fractional diffusion equations, *Journal of Computational Physics* 229 (21) (2010) 8095–8104.
- [25] H. Wang, K. Wang, A direct  $O(N\log^2 N)$  alternating-direction finite difference method for two-dimensional fractional diffusion equations, *Journal of Computational Physics* 230 (2011) 7830–7839.
- [26] H. Pang, H. Sun, Multigrid method for fractional diffusion equations, *Journal of Computational Physics* 231 (2) (2012) 693–703.
- [27] E. Hanert, A comparison of three Eulerian numerical methods for fractional-order transport models, *Environmental Fluid Mechanics* 10 (2010) 7–20, doi:10.1007/s10652-009-9145-4.
- [28] E. Hanert, On the numerical solution of space-time fractional diffusion models, *Computers and Fluids* 46 (2011) 33–39, doi:10.1016/j.compfluid.2010.08.010.
- [29] X. Li, C. Xu, A space-time spectral method for the time fractional diffusion equation, *SIAM Journal on Numerical Analysis* 47 (2009) 2108–2131.
- [30] T.A. Driscoll, B. Fornberg, Interpolation in the limit of increasingly flat radial basis functions, *Computers and Mathematics with Applications* 43 (3-5) (2002) 413–422.
- [31] B. Fornberg, J. Zuev, The Runge phenomenon and spatially variable shape parameters in RBF interpolation, *Computers and Mathematics with Applications* 54 (3) (2007) 379–398.
- [32] B. Fornberg, C. Piret, On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere, *Journal of Computational Physics* 227 (5) (2008) 2758–2780.
- [33] B. Fornberg, E. Larsson, N. Flyer, Stable computations with Gaussian radial basis functions, *SIAM Journal of Scientific Computing* 33 (2011) 869–892.

- [34] I. Podlubny, *Fractional Differential Equations. An Introduction to Fractional Derivatives, Fractional Differential Equations, Some Methods of Their Solution and Some of Their Applications.*, Academic Press, 1999.
- [35] C. Li, W. Deng, Remarks on fractional derivatives, *Applied Mathematics and Computation* 187 (2007) 777–784.
- [36] X. Zhang, M. Lv, J.W. Crawford, I.M. Young, The impact of boundary on the fractional advection-dispersion equation for solute transport in soil: Defining the fractional dispersive flux with the caputo derivatives, *Advances in Water Resources* 30 (5) (2007) 1205–1217.
- [37] R. Hardy, Multiquadric equations of topography and other irregular surfaces, *Journal of Geophysical Research* 76 (8) (1971) 1905–1915.
- [38] R. Franke, Scattered data interpolation: Tests of some methods., *Math. Comput.* 38 (157) (1982) 181–200.
- [39] C. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions, *Constructive Approximation* 2 (1) (1986) 11–22.
- [40] E. Kansa, Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics - II: Solutions to parabolic, hyperbolic and elliptic partial differential equations, *Computers & Mathematics with Applications* 19 (8-9) (1990) 147–161.
- [41] E. Kansa, Multiquadrics - A scattered data approximation scheme with applications to computational fluid-dynamics - I: Surface approximations and partial derivative estimates, *Computers & Mathematics with Applications* 19 (8-9) (1990) 127–145.
- [42] I. Schoenberg, Metric spaces and completely monotone functions, *The Annals of Mathematics Second Series* 39 (4) (1938) 811–841.
- [43] B. Fornberg, G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter, *Comput. Math. Appl.* 48 (5-6) (2004) 853–867.
- [44] R. B. Platte, L. N. Trefethen, A. B. J. Kuijlaars, Impossibility of fast stable approximation of analytic functions from equispaced samples, *SIAM Review* 53 (2) (2011) 308–318.
- [45] E. Sousa, Numerical approximations for fractional diffusion equations via splines, *Computers & Mathematics with Applications* 62 (3) (2011) 938–944.
- [46] B. Fornberg, T. A. Driscoll, G. Wright, R. Charles, Observations on the behavior of radial basis function approximations near boundaries, *Computers Math. Appl.* 43 (3-5) (2002) 473–490.