Solving mixed classical and fractional partial differential equations using short–memory principle and approximate inverses

Daniele Bertaccini & Fabio Durastante

Numerical Algorithms

ISSN 1017-1398
Volume 74
Number 4

Numer Algor (2017) 74:1061-1082
DOI 10.1007/s11075-016-0186-8

Springer

Your article is protected by copyright and all rights are held exclusively by Springer Science +Business Media New York. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".

Springer

Author's personal copy

Numer Algor (2017) 74:1061–1082
DOI 10.1007/s11075-016-0186-8

ORIGINAL PAPER

Solving mixed classical and fractional partial differential equations using short–memory principle and approximate inverses

Daniele Bertaccini[1] · Fabio Durastante[2]

Received: 21 March 2016 / Accepted: 28 July 2016 / Published online: 11 August 2016
© Springer Science+Business Media New York 2016

Abstract The efficient numerical solution of the large linear systems of fractional differential equations is considered here. The key tool used is the *short–memory principle*. The latter ensures the decay of the entries of the inverse of the discretized operator, whose inverses are approximated here by a sequence of sparse matrices. On this ground, we propose to solve the underlying linear systems by these approximations or by iterative solvers using sequence of preconditioners based on the above mentioned inverses.

Keywords Preconditioners · Fractional calculus · Iterative methods

Mathematics Subject Classification (2010) 65F10 · 65F08 · 26A33 · 65Y10

1 Introduction

Partial differential equations provide tools for modeling phenomena in many areas of science. Nonetheless, there exist phenomena for which this kind of modeling is not as effective. For example, the processes of anomalous diffusion, the dynamics

✉ Daniele Bertaccini
bertaccini@mat.uniroma2.it

Fabio Durastante
fdurastante@uninsubria.it

1 Dipartimento di Matematica, Università di Roma "Tor Vergata", via della Ricerca Scientifica 1, 00133 Roma, Italy

2 Dipartimento di Scienza e Alta Tecnologia, Università dell'Insubria, via Valleggio 11, 22100 Como, Italy

Author's personal copy

of viscoelastic and polymeric materials (see [30, 37] for details and a list of other possible applications). Indeed, most of the processes associated with these have non-local dynamics, for which the use of *fractional partial derivatives* seems to be much more effective. For physical interpretation (see, e.g., [32]).

To deal with the simulation of these models, we use the matrix approach framework as suggested in [31, 33, 34] to try to transfer some of computational techniques developed for ordinary partial differential equations to differential equations with fractional partial derivative, called also *fractional differential equations* or *FDEs* for brevity. In recent years, there has been some important contributions in this field; see, e.g., [10, 23–26, 39].

We propose the use of a structural property of the fractional derivative known as the *short–memory principle*. We are dealing with non-local operators, but their structure permit to observe a decay of correlations towards the extremes of the interval of integration. As have been observed in [24, Chap. 2.6], "Up to now, the short memory principal has not been thoroughly studied so is seldom used in the real applications." As an example, we can consider the work on the predictor-corrector approach in [14] and the work in [35]. For the solution with Krylov subspace methods have been also developed strategies with approximate inverse preconditioners. The latter build structured approximation of the inverse of the discretization matrix in the fashion of an inverse circulant-plus-diagonal preconditioner (see the work in [27, 29]).

The novelty of our proposal is mainly in the use of *short–memory principle* as a means to generate sequences of approximations for the inverse of the discretization matrix with a low computational effort. With the aid of this precious property, we can solve the underlying discrete problems effectively by preconditioned Krylov iterative methods. In this way, there is no loss of accuracy in the discretization of the differential model, because the decay properties of the operators are used to approximate their inverses. Use of this solution framework also allows to exploit strategies for updating approximate inverses to treat problems with coefficients that vary over time, or to apply methods of integration with variable time step, in the style of [1, 3, 4, 16]. Finally, we note that from the computational point of view this also allows us to use GPUs, for which these techniques have been recently specialized (see [6, 11, 12, 17] for the implementation details and Section 4 for our contribution).

2 Matrix approach

This section is devoted to the approximation of the fractional integral and differential operator in matrix form. In accord with [30], we start recalling the notation for the fractional operators we are interested in. From now on with the notation $\Gamma(\cdot)$ we mean the *Euler gamma function*, the usual analytic continuation to all complex numbers (except the non-positive integers) of the convergent improper integral function

$$\Gamma(t) = \int_0^{+\infty} x^{t-1} e^{-x} dx. \tag{1}$$

Definition 1 (Fractional Operators) Given a function $y(t)$, we define

Author's personal copy

Fractional Integral given $\alpha > 0$ and $a < b \in \mathbb{R} \cup \{\pm\infty\}$,

$$J^{\alpha}_{a,x} y(x) = \frac{1}{\Gamma(\alpha)} \int_a^x (x-\xi)^{\alpha-1} y(\xi) d\xi; \tag{2}$$

Riemann-Liouville given $\alpha > 0$ and $m \in \mathbb{Z}^+$ such that $m-1 < \alpha \leq m$, the left-side Riemann-Liouville fractional derivative reads as

$$_{\mathrm{RL}}D^{\alpha}_{a,x} y(x) = \frac{1}{\Gamma(m-\alpha)} \left(\frac{d}{dx}\right)^m \int_a^x \frac{y(\xi)d\xi}{(x-\xi)^{\alpha-m+1}}, \tag{3}$$

while the right-side Riemann-Liouville fractional derivative

$$_{\mathrm{RL}}D^{\alpha}_{x,b} y(x) = \frac{1}{\Gamma(m-\alpha)} \left(-\frac{d}{dx}\right)^m \int_x^b \frac{y(\xi)d\xi}{(\xi-x)^{\alpha-m+1}}; \tag{4}$$

Symmetric Riesz given $\alpha > 0$ and $m \in \mathbb{Z}^+$ such that $m-1 < \alpha \leq m$, the symmetric Riesz derivative reads as

$$\frac{d^{\alpha} y(x)}{d|x|^{\alpha}} = \frac{1}{2}\left({}_{\mathrm{RL}}D^{\alpha}_{a,x} + {}_{\mathrm{RL}}D^{\alpha}_{x,b}\right); \tag{5}$$

Caputo given $\alpha > 0$ and $m \in \mathbb{Z}^+$ such that $m-1 < \alpha \leq m$, the left-side Caputo fractional derivative reads as

$$_{\mathrm{C}}D^{\alpha}_{a,x} y(x) = \frac{1}{\Gamma(m-\alpha)} \int_a^x \frac{y^{(m)}(\xi)d\xi}{(x-\xi)^{\alpha-m+1}}, \tag{6}$$

while the right-side

$$_{\mathrm{C}}D^{\alpha}_{x,b} y(x) = \frac{(-1)^m}{\Gamma(m-\alpha)} \int_x^b \frac{y^{(m)}(\xi)d\xi}{(\xi-x)^{\alpha-m+1}}, \tag{7}$$

Grünwald-Letnikov given $\alpha > 0$ and $m \in \mathbb{Z}^+$ such that $m-1 < \alpha \leq m$, the left-side Grünwald-Letnikov fractional derivative reads as

$$_{\mathrm{GL}}D^{\alpha}_{a,x} y(x) = \lim_{\substack{h\to 0\\ Nh=t-a}} \frac{1}{h^{\alpha}} \sum_{j=0}^{N} (-1)^j \binom{\alpha}{j} y(x-jh), \tag{8}$$

while the right-side

$$_{\mathrm{GL}}D^{\alpha}_{x,b} y(x) = \lim_{\substack{h\to 0\\ Nh=b-t}} \frac{1}{h^{\alpha}} \sum_{j=0}^{N} (-1)^j \binom{\alpha}{j} y(x+jh). \tag{9}$$

Generally speaking, the definitions given above are equivalent only for functions that are suitably smooth. Nevertheless, in some case, relationships can be established between the fractional derivatives written in the above forms (see [30]).

To treat fractional differential equations, i.e., dealing with equations written in terms of the operators in Definition 1, we recall some matrix-based approaches. We can consider the one introduced in [31] and further generalized in [33, 34]. This method is based on a suitable matrix representation of discretized fractional operators in a way that is the analogues of the numerical differentiation for standard integer order differential equations.

Author's personal copy

Let us fix an interval $[a, b] \subseteq \mathbb{R}$, an order of fractional derivative α and consider the equidistant nodes of step size $h = (b-a)/N$, $\{x_k = a+kh\}_{k=0}^N$, with $x_0 = a$ and $x_N = b$. Then, for functions $y(x) \in \mathcal{C}^r([a, b])$ with $r = \lceil\alpha\rceil$ and such that $y(x) \equiv 0$ for $x < a$, we have that

$$ {}_{\mathrm{RL}}D^{\alpha}_{a,x}y(x) = {}_{\mathrm{GL}}D^{\alpha}_{a,x}y(x), \quad {}_{\mathrm{RL}}D^{\alpha}_{x,b}y(x) = {}_{\mathrm{GL}}D^{\alpha}_{x,b}y(x). \tag{10} $$

Therefore, we can approximate both the left- and right-side Riemann-Liouville derivatives with the truncated Grünwald-Letnikov expansion

$$ {}_{\mathrm{RL}}D^{\alpha}_{a,x_k}y(x) \approx \tfrac{1}{h^\alpha}\textstyle\sum_{j=0}^{k}(-1)^j\binom{\alpha}{j}y((k-j)x), \quad k = 0, \ldots, N, \tag{11} $$

$$ {}_{\mathrm{RL}}D^{\alpha}_{x_k,b}y(x) \approx \tfrac{1}{h^\alpha}\textstyle\sum_{j=0}^{k}(-1)^j\binom{\alpha}{j}y((k+j)x), \quad k = 0, \ldots, N. \tag{12} $$

In this way, we can define the lower and upper Toeplitz triangular matrices, respectively

$$ B_L^{(\alpha)} = \frac{1}{h^\alpha}\begin{bmatrix} \omega_0^{(\alpha)} & 0 & 0 & \ldots & 0 \\ \omega_1^{(\alpha)} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \omega_N^{(\alpha)} & \omega_{N-1}^{(\alpha)} & \cdots & \omega_1^{(\alpha)} & \omega_0^{(\alpha)} \end{bmatrix}, \quad B_U^{(\alpha)} = \left(B_L^{(\alpha)}\right)^T, \tag{13} $$

where the coefficients $\omega_j^{(\alpha)}$ are defined as

$$ \omega_j^{(\alpha)} = (-1)^j\binom{\alpha}{j}, \quad j = 0, 1, \ldots, N. \tag{14} $$

To satisfy the semi-group property of the left-right Riemann-Liouville fractional derivatives is also needed that $y^{(k)}(a) = 0$ for the left, respectively $y^{(k)}(b) = 0$ for the right, for each $k = 1, 2, \ldots, r-1$. It is crucial to observe that there is a decay of the coefficients along the diagonals of the discretization matrices.

Proposition 1 *Given the discretization formula in* (13)*, the following decay rate for the coefficients holds*

$$ |\omega_j^{(\alpha)}| = O(j^{-\alpha-1}), \quad \textit{for } j \to +\infty. \tag{15} $$

This is a well-known consequence of the asymptotic relation for the Gamma function in [38]:

$$ \lim_{x\to+\infty}\frac{\Gamma(x+\alpha)}{x^\alpha\Gamma(x)} = 1, \quad \forall\alpha \in \mathbb{R}. \tag{16} $$

If we are interested in obtaining an even sharper bound for the constant, the estimate for the sequence of real binomial coefficients in [22, Theorem 4.2] can be applied. The strategy we have described is a numerical scheme with accuracy $O(h)$.

Author's personal copy

With the same strategy, a discretization, showing the same decaying property, for the Symmetric Riesz fractional derivative is obtained.

Schemes with higher accuracy can also be derived by observing that

$$(1-z)^{\alpha} = \sum_{j=0}^{+\infty} \omega_j^{(\alpha)} z^j, \; z \in \mathbb{C}. \tag{17}$$

Therefore, as have been done in [25], by substituting the generating function of the first order one-side differences with the one of the desired order and posing

$$z = \exp(-\imath\theta),$$

we get the coefficients for the scheme of higher accuracy. We stress that all the procedure can be done automatically by using Fornberg algorithm [18] and FFTs. What we need to observe is that also in this case we can state the following proposition.

Proposition 2 *Given a one-side finite difference discretization formula represented by the polynomial* $p_q(z)$ *of degree* q*, with an associated Fourier symbol* $f(\theta) \triangleq p_q(\exp(-i\theta))$*, and* $\alpha \in (0,1)\cup(1,2)$*, we have that the coefficients for the discretization formula of* ${}_{RL}D^{\alpha}_{x,a}$ *are given by the Fourier coefficients* $\{\omega_j^{(\alpha,q)}\}_j$ *of the function* $f(\theta)^{\alpha}$ *and*

$$\begin{aligned} 0 < \alpha < 1, &\quad \omega_j^{(\alpha,q)} = O(1/|j|^{\alpha}), \quad j \to +\infty, \\ 1 < \alpha < 2, &\quad \omega_j^{(\alpha,q)} = O(1/|j|^{1+\alpha}), \quad j \to +\infty. \end{aligned}$$

Proof The results follows by standard relations between Hölder continuity, regularity, and Fourier coefficients (see [30, Section 7.6] for details). □

For our purposes, it is enough to observe that also in this case the matrix shows a polynomial decay of the coefficients, as can be seen in Fig. 1.

Another strategy to obtain methods with higher order of accuracy is using the *shifted Grünwald-Letnikov* approximation from [26]. For our purpose, it is enough to say that it consists in building matrices that are no more only lower triangular, but with coefficients on the other diagonals obtained with the same approximations. Therefore, the decay of the entries is preserved with the same behavior in Propositions 1 and 2.

To discretize symmetric Riesz fractional derivatives, other approaches can be also taken into account. We recall here only the so-called central-fractional-difference approach from [28], with its further generalizations in [10]. By observing that for $\alpha \in (1,2]$, the Riesz fractional derivative operator of Definition 1 can be rewritten as

$$\frac{\partial^{\alpha} u(x)}{\partial |x|^{\alpha}} = -\frac{1}{2\cos(\alpha\frac{\pi}{2})\Gamma(2-\alpha)} \frac{d^2}{dx^2} \int_{\mathbb{R}} \frac{u(\xi)d\xi}{|x-\xi|^{\alpha-1}}, \tag{18}$$

Author's personal copy

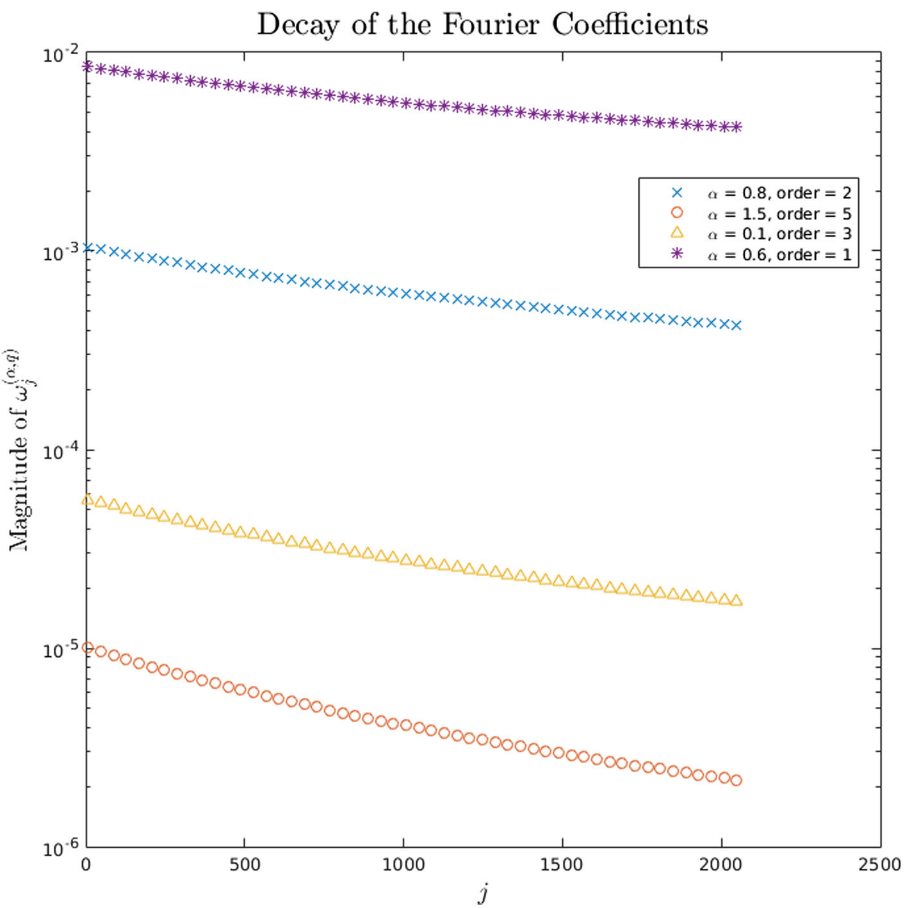

Fig. 1 Decay of the Fourier coefficients as in Propositions 1 and 2

the following $O(h^2)$ scheme, given by [28], can be obtained

$$\frac{\partial^\alpha u(x)}{\partial |x|^\alpha} = -\frac{1}{h^\alpha} \sum_{k=-\frac{b-x}{h}}^{\frac{x-a}{h}} \varsigma_k u(x-kh) + O(h^2),$$
$$\varsigma_k = \frac{(-1)^k \Gamma(\alpha+1)}{\Gamma(\alpha/2-k+1)\Gamma(\alpha/2+k+1)}. \tag{19}$$

The decay of the coefficients of the scheme from [28] have been proved with the same techniques of Proposition 1.

Corollary 1 *For large values of k for the coefficients* ς_k *of* (19)*, we have*

$$\varsigma_k = O(k^{-\alpha-1}) \tag{20}$$

Author's personal copy

By discretizing the differential operator over the same uniform grid over $[a, b] \subseteq \mathbb{R}$, it is possible to restate the problem in symmetric Toeplitz matrix form:

$$O_N^{(\alpha)} = -\frac{1}{h^\alpha}\begin{bmatrix} \varsigma_0 & \varsigma_1 & \cdots & \varsigma_N \\ \varsigma_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \varsigma_1 \\ \varsigma_N & \cdots & \varsigma_1 & \varsigma_0 \end{bmatrix}, \quad \frac{\partial^\alpha u(x)}{\partial |x|^\alpha} = O_N^{(\alpha)} u(x_k) + O(h^2). \tag{21}$$

We remark also that discretizations of higher order for the symmetric Riesz derivative were introduced and effectively applied in [15]. For our purposes, it is enough to note that the entries of the matrix form are weighted sums of the coefficients ς_k of (19). Therefore, they show the same decay properties, although with coefficients of different magnitude.

The matrices generated by the Grünwald-Letnikov approximation and the central-fractional-differences share the same decay property along the diagonals. This feature depends on a structural property of the fractional derivatives. While the classical derivatives are local operators, the fractional derivatives and integral operator of definition (1) are non-local. Again, as we have just observed, the role of the *history* of the behavior of the $y(x)$ function, when we go near to the starting or ending point, has less importance: the *short–memory* principle.

2.1 The short–memory principle

To introduce the *short–memory* principle, we can follow the approach in [30, Section 7.3] defining a *memory length* L and then imposing the approximation

$$_{\mathrm{RL}}D^\alpha_{a,x} y(x) \approx {}_{\mathrm{RL}}D^\alpha_{x-L,x} y(x), \quad x > a + L. \tag{22}$$

Therefore, the error produced by zeroing out the entries of the matrix representing the operator is given by

$$E(x) = |{}_{\mathrm{RL}}D^\alpha_{a,x} y(x) - {}_{\mathrm{RL}}D^\alpha_{x-L,x} y(x)| \leq \frac{\sup_{x\in[a,b]} y(x)}{L^\alpha |\Gamma(1-\alpha)|}, \; a + L \leq x \leq b. \tag{23}$$

Thus, fixed an admissible error ε, we get that

$$L \geq \left(\frac{\sup_{x\in[a,b]} f(x)}{\varepsilon |\Gamma(1-\alpha)|} \right)^{\frac{1}{\alpha}} \Rightarrow \begin{array}{l} E(x) \leq \varepsilon, \\ a + L \leq x \leq b. \end{array} \tag{24}$$

This underlying properties can be used to build a predictor-corrector approach for FDEs as in [14], or to apply truncation to reduce the computational cost for the exponential of the discretization matrix as in [35].

Differently from these approaches, we want to preserve all the information obtained by the discretization and use the short–memory principle, i.e., the decaying of the entries, to gain information on the inverse of the discretization matrix.

Classical results on the decays of the inverse of a matrix A, as in [13], have been proven to be useful in different frameworks. They have been generalized also to other

Author's personal copy

matrix function than the simple $f(z) = z^{-1}$. For our needs, we are going use the results in [20] that require that A is not sparse or banded but has entries that show polynomial or exponential decay. This is exactly the case of our discretizations (see Fig. 1 for the coefficients and Fig. 2 for the decay of the inverse).

We can now state the result from [20, 21] to check that the decay really exists.

Theorem 1 *Given an invertible matrix* $(A)_{h,k}$ *such that*

$$|a_{h,k}| \leq C(1 + |h - k|)^{-s}, \tag{25}$$

then its inverse A^{-1} *shares the same property. Moreover, the class* $\mathcal{Q}_s$ *of such matrices is an algebra.*

Proposition 3 *Given the discretizations in Propositions 1, 2 or in Corollary 1, we have that if* $\alpha \in (1, 2)$ *the following relation holds*

$$\exists\, C > 0 \;:\; |(A^{-1})_{h,k}| = |\theta_{h,k}| \leq C(1 + |h - k|)^{-\alpha-1}, \tag{26}$$

while for $\alpha \in (0, 1)$ *we get*

$$\exists\, C > 0 \;:\; |(A^{-1})_{h,k}| = |\theta_{h,k}| \leq C(1 + |h - k|)^{-\alpha}. \tag{27}$$

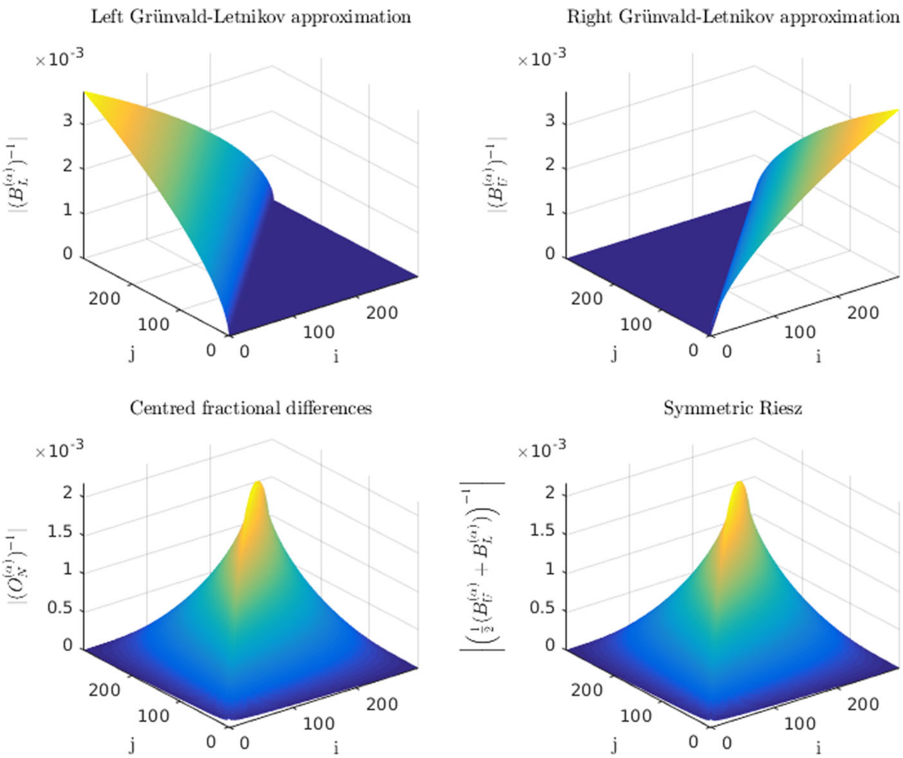

Fig. 2 Decay of the inverse matrix relative to the various discretizations, $n = 300$ and $\alpha = 1.7$

Author's personal copy

Proof To prove the results, it is enough to observe that by (15) of Proposition 1, we get

$$\exists C > 0 \ : \ |(A)_{h,k}| = |a_{h,k}| = |\omega^{(\alpha)}_{|h-k|}| \leq C(1+|h-k|)^{-\alpha-1}. \tag{28}$$

Therefore, by Theorem 1, the results hold. The bounds for the other discretizations are obtained in the same way. □

2.2 Multidimensional FPDEs

The other case that need consideration is the one of multidimensional FPDEs. In the linear constant coefficients case, the matrices of discretization can be written as a sum of Kronecker products of matrices that discretize the equation in one dimension.

Given two matrices $A, B \in \mathbb{R}^{n\times n}$, we have that their Kronecker product $(C_{\alpha,\beta})_{\alpha,\beta} \triangleq (A_{i,j})_{i,j} \otimes (B_{k,l})_{k,l}$ is defined elementwise as

$$c_{\alpha,\beta} = a_{i,j}b_{k,l}, \quad \alpha = n(i-1)+k, \ \beta = n(j-1)+l, \quad 1 \leq i,j,k,l \leq n. \tag{29}$$

As a corollary of the previous, we can state the following result.

Proposition 4 *Given $A, B \in \mathbb{R}^{n\times n}$, $A = (a_{i,j})$, $B = (b_{i,j})$,*

$$|a_{i,j}| \leq C_1(1+|i-j|)^{-s_1}, \ |b_{i,j}| \leq C_2(1+|i-j|)^{-s_2}, \tag{30}$$

and I the identity matrix of order n, we have that

$$A \oplus B \triangleq A \otimes I + I \otimes B, \tag{31}$$

and there exist $C, s > 0$ such that

$$|(A \oplus B)_{\alpha,\beta}| \leq C(1+|\alpha-\beta|)^{-s}. \tag{32}$$

Proof By Theorem 1, we know that the set of matrices whose entries show a polynomial decay is an algebra. Therefore, we only need to prove that $A\otimes I$ and $I\otimes B$ show the decaying property too. For $(I \otimes B)_{\alpha,\beta}$, we get simply that $(I \otimes B)_{\alpha,\beta} = \delta_{i,j}b_{k,l}$, where $\delta_{i,j}$ is the usual Kronecker delta. Then, we get the block matrix with n copies of the B matrix on the main diagonal. This implies that we have preserved the same decay property of the B matrix having added only zeros entries. On the other hand, for $(A \otimes I)$ we get

$$(A \otimes I)_{\alpha,\beta} = a_{i,j}\delta_{k,l},$$

that is the following block matrix

$$A \otimes I = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \dots & A_{n,n} \end{bmatrix}, \quad A_{i,j} = a_{i,j}I. \tag{33}$$

Again, we can use the same decay bound for the matrix A, even if it is no more sharp because the values are now interspersed by diagonals of zero. □

Author's personal copy

Sharper bounds for this kind of structures have been obtained in [9]. Nevertheless, they refer to the case of banded matrices A, B and can become of interest when we consider equations with both fractional and classical derivatives.

Remark 1 The decay of the entries and the strategy of dropping entries of prescribed small modulus in the inverse matrices can be applied also when some form of the short–memory principle has been used to approximate directly the system matrix.

3 Solution strategies

The observed decay of the entries for the inverse of the matrices that discretize the FPDEs allows us to devise an effective algorithm. All is based on the possibility of discarding elements of prescribed small modulus in the calculation of an approximate inverse of the matrix of interest. For the case of non-fractional PDEs, i.e., with non-fractional partial derivatives, this technique was often used to produce explicit preconditioners for Krylov subspace methods: the approximate inverse preconditioners (see, e.g., [1, 3, 4, 6]).

On this basis, our proposal is to solve the discretized differential equations in matrix form, written in terms of the operators we introduced in Section 2. To this end, we use an appropriate Krylov subspace method (see, e.g., [36]) with an approximate inverse preconditioner in factored form discussed in Section 3.1. *Experiments 1, 2,* and *3* in Section 4 illustrate the application of our approach to some test problems.

On the other hand, if we have a good approximation of the inverse of the discretization matrix, we can use it as a direct method for the solution of the given FPDE. In this way, the solution procedure is reduced to compute an appropriate approximate inverse and do some matrix-vector products.

For this second approach, we consider the solution of a pure fractional partial differential equation, i.e., without derivatives of integer order. Having discretized it in terms of the formulas in Section 2, we get sequences of matrices, which, together with their inverses, share the decay property called *short–memory principle*. In practice, we approximate the underlying inverses with the approximate inverses in Section 3.1. Similarly to what was done in [35], we are going to trade off accuracy and performance. Nevertheless, instead of discarding elements of the discretized operator and then solving the associated linear systems, we are going to act directly on the inverse of the operator, building a sort of direct method (see *Experiment 4* in Section 4).

Before trying both strategies on some test problems, we recall a class of algorithms from [7] to compute the approximate inverse through the use of a *biorthogonalization* procedure (*conjugation* for the Hermitian case). Moreover, we will consider a strategy from [3–5, 16] to update these approximations in the case where the discretization of the equation depends on the time step, giving rise to a sequence of algebraic linear systems with variable coefficient matrices.

We stress that also other strategies, based on the use of approximate inverses architecture, have been used in literature. The one considering the solutions of Hermitian positive definite Toeplitz-plus-diagonal systems from [27], which is based on the exponential decay of the entries of the matrix combined with the use of approximate

Author's personal copy

inverse circulant-plus-diagonal preconditioners. A similar idea, although specialized for the case of discretization of fractional diffusion equations, has been developed in [29]. In the latter, the construction of an approximate inverse preconditioner is obtained by the circulant approximation of the inverses of a scaled Toeplitz matrix that is then combined together in a row-by-row fashion.

3.1 Approximate inverse preconditioners

Given a matrix A symmetric and positive definite, we are interested in an appropriate sparse approximation of the inverse of A in factorized form

$$A^{-1} \approx ZD^{-1}Z^T, \tag{34}$$

where the matrix Z is lower triangular. If the process is carried out without dropping and in exact arithmetic, we get $Z = L^{-T}$, where L is the unit Cholesky factor of A. Moreover, we consider sparse approximations for unsymmetric matrices

$$A^{-1} \approx WD^{-1}Z^T. \tag{35}$$

We recall that there exist stabilized algorithms for computing approximate inverses for nonsymmetric matrices that are less prone to breakdowns than others (see [2]). Indeed, we stress that in general, incomplete factorizations (ILUs; see [36]) are not a guarantee to be nonsingular also for positive definite matrices. This issue holds also for efficient inverse ILU techniques, i.e., approximations of the inverse matrix generated by sparsification and inversion of an ILU algorithm. In particular, we will not use inverse ILU techniques for our test problems in Section 4 because of the frequent breakdowns.

To show that the decay allows to build approximate inverses significantly more sparse, we report in Table 1 the fill-in percentage of the approximate factorization for the test problem

$$\begin{cases} u_t = d_+(x)\, {}_{\text{RL}}D^{\alpha}_{a,x}u + d_-(x)\, {}_{\text{RL}}D^{\alpha}_{x,b}u + f(x,t). & x \in (a,b), \\ u(a,t) = u(b,t) = 0, & t \in [0,T], \\ u(x,0) = u_0(x). & x \in [a,b]. \end{cases} \tag{36}$$

where the fractional derivative order is $\alpha \in (1,2)$, $f(x,t)$ is the forcing term, and $d_\pm(x)$ are two non-negative functions representing the variable diffusion coefficients.

Table 1 Ratio of fill-in for various drop tolerances for the problem (36) with matrix size $n = 2048$ and $\alpha = 1.8$

Tolerance	nnz(W)	nnz(Z)	Fill-in (%)
5.00e-01	4094	4095	0.20
1.00e-01	52253	51867	2.48
1.00e-02	117294	116453	5.57
1.00e-03	214242	208970	10.09
1.00e-04	332837	319870	15.56
1.00e-05	466732	443718	21.71
1.00e-06	630214	594081	29.19

Author's personal copy

The convergence properties of this approach will be discussed in the numerical experiments section.

Among reliable methods for computing directly an approximate factorization of the inverse of the underlying matrices, we mention the Bridson's *outer product* formulation in [7, 8] and its implementation in [11]. The algorithm is based on a *biconjugate* Gram–Schmidt process (i.e., conjugate with respect to the bilinear form associated with A). Sparsity in the inverse factors is obtained by carrying out biconjugation process incompletely. A left-looking reformulation of the algorithm with the use of outer products permits to exploit even more zeros under suitable conditions (see [6]).

We do not discuss the techniques for generating approximate inverse in factorized form any further because it is out of the scope of this paper (see, e.g., [2, 6–8] for details).

3.2 Updating factorizations for the approximate inverses

The discretization of time-dependent fractional PDEs produces a sequence of linear systems whose matrices are dependent on the time step. Usually facing a sequence of linear systems with different matrices makes expensive to rebuild a new preconditioner each time. On the other hand, reusing the same preconditioner can be inappropriate; see [3, 4].

The update strategy considered here is the one developed in [3, 4] and further in [1]. We consider having a sequence of nonsymmetric matrices $\{A_k\}_{k=0}^{t}$, where A_0 is called the *reference matrix*.

Consider an initial approximation (35) for the inverse of A_0 in factorized form:

$$P_0^{-1} = ZD^{-1}W^T, \tag{37}$$

then by writing each A_k, for $k \geq 1$ as

$$A_k = A_k - A_0 + A_0 = A_0 + \Delta_k, \ \ \Delta_k \triangleq A_k - A_0, \tag{38}$$

we build the updated preconditioner as

$$A_k^{-1} \approx P_k^{-1} = Z(D + E_k)^{-1}W^T, \ \ E_k \triangleq g(W^T \Delta_k Z), \tag{39}$$

where g is a sparsification function, e.g., a function that extracts some banded approximation of its matrix argument.

Techniques with more than one *reference matrix* have been devised in [5, 16] by means of matrix interpolation. The formula (39) is modified to change at each update the Z and W factors, i.e.,

$$A_k^{-1} \approx P_k^{-1} = Z_k(D_k + E_k)^{-1}W_k^T, \ \ E_k \triangleq g(W_k^T \Delta_k Z_k). \tag{40}$$

In order to use the inverse approximated in a direct manner, rather than as a preconditioner for an iterative method, we considered to use the update rule as a way to get an approximation for the inverse of the matrix of the other steps.

Author's personal copy

4 Numerical experiments

The numerical experiments are performed on a laptop running Linux with 8 Gb memory and CPU Intel(R) Core(TM) i7-4710HQ CPU with clock 2.50 GHz, while the GPU is an NVIDIA GeForce GTX 860M. The scalar code is written and executed in MATLAB R2015a, while for the GPU we use C++ with Cuda compilation tools, release 6.5, V6.5.12 and the CUSP library [11].

We build our approximate inverses with the CUSP library [11] that implements the standard scaled Bridson algorithm for approximate inverses in factorized form. This choice was made to exploit highly parallel computer architectures such as GPUs. In this kind of setting, it is possible to implement efficiently matrix-vector multiplications that are the numerical kernel of both iterative Krylov subspace and approximate inverse preconditioners considered here. Moreover, the implementation of the underlying approximate inverses relays on a left-looking variant of the *biorthogonalization* procedure that, as have been observed in [6], sometimes suffers less from pivot breakdown. In the proposed experiments, we use GPU only to build the preconditioners in order to emphasize the related performances. However, it is possible to implement also the solving phase on GPUs, i.e., the iterative methods for the underlying algebraic linear systems. This will be the subject of a future work. Indeed, we are actually studying the migration to the *Parallel Sparse BLAS* (PSBLAS) library [17] that contains some Krylov subspace iterative methods and the basic kernels for computing matrix-vector products, in conjunction with the MLD2P4 framework [12], containing some parts of the preconditioners. The former is a package of multilevel preconditioners that can work with the PSBLAS library (see [6] and references therein).

The proposed strategies are scalable also on architectures that use more than one degree of parallelism, i.e., using more than one GPU to further improve the speedups.

We start considering the following problem containing both integer and fractional derivatives

$$
\begin{cases}
\dfrac{\partial u(x,y,t)}{\partial t} = \nabla \cdot (< ((a(x), b(y)), \nabla u >) & (x,y) \in \Omega, t > 0, \\
\qquad\qquad\qquad +\mathcal{L}^{(\alpha,\beta)} u, & \\
u(x,y,0) = u_0(x,y), & \\
u(x,y,t) = 0 & (x,y) \in \partial\Omega, t > 0.
\end{cases}
\tag{41}
$$

where the fractional operator is given by

$$
\begin{aligned}
\mathcal{L}^{(\alpha,\beta)} \cdot \triangleq\ & d_{+}^{(\alpha)}(x,y)\, {}_{\mathrm{RL}}D_{a,x}^{\alpha} \cdot + d_{-}^{(\alpha)}(x,y)\, {}_{\mathrm{RL}}D_{x,b}^{\alpha} \cdot \\
& + d_{+}^{(\beta)}(x,y)\, {}_{\mathrm{RL}}D_{a,y}^{\beta} \cdot + d_{-}^{(\beta)}(x,y)\, {}_{\mathrm{RL}}D_{y,b}^{\beta} \cdot
\end{aligned}
\tag{42}
$$

To discretize the problem, we consider the five points discretization of the Laplace operator combined with the shifted Grünwald-Letnikov approximation for the fractional one. As a time integrator, we choose the Backward-Euler method, considering the behavior of the error of approximation and the overall stability of the method (see [26]).

Author's personal copy

Experiment 1 As a first choice for the solution of problem (41), we consider the coefficients

$$\begin{aligned} a(x) &= 1 + 0.5\sin(3\pi x),\ b(y) = 1 + 0.7\sin(4\pi y) \\ d_+^{\alpha}(x, y, t) &= d_-^{\alpha}(x, y, t) = e^{4t} x^{4\alpha} y^{4\beta}, \\ d_+^{\beta}(x, y, t) &= d_-^{\beta}(x, y, t) = e^{4t}(2 - x)^{4\alpha}(2 - y)^{4\beta} \\ u_0(x, y) &= x^2 y^2 (2 - x)^2 (2 - y)^2. \end{aligned} \tag{43}$$

over the domain $[0, 2]^2$ and the time interval $t \in [0, 2]$. In Table 2 ,we consider the use of the `GMRES(50)` algorithm with a tolerance of $\varepsilon = 1e{-}6$ and a maximum number of admissible iterations `MAXIT` $= 1000$. The approximate inverses are computed with a drop tolerance $\delta = 0.1$. The update formula (39) for the preconditioner is used with only a diagonal update, i.e., the function $g(\cdot)$ extracts only the main diagonal. In Table 2 ,we report the average timings and the number of external/internal iterations. When a "†" is reported, at least one of the iterative solvers does not reach the prescribed tolerance in `MAXIT` iterations. We report in Fig. 3 the solution obtained at different time steps. We consider the solution of this problem also with `BiCGSTAB` and `GMRES` algorithms with the same settings. The results for this case are reported in Tables 3 and 4, respectively.

Experiment 2 We now consider a slightly different model in which the coefficient function is used in divergence form

$$\begin{cases} \dfrac{\partial u(x, y, t)}{\partial t} = \nabla \cdot (a(x, y)\nabla u) & (x, y) \in \Omega, t > 0, \\ \qquad\qquad\quad +\mathcal{L}^{(\alpha,\beta)} u, & \\ u(x, y, 0) = u_0(x, y), & \\ u(x, y, t) = 0 & (x, y) \in \partial\Omega, t > 0. \end{cases} \tag{44}$$

The fractional operator $\mathcal{L}^{(\alpha,\beta)}$ is defined as in (42). We set $N_x = N_y = 80$ and $N_t = 70$ over the domain $\Omega = [0, 2]^2$ and $T = [0, 2]$. The same choice of the previous example have been done for the fractional diffusion coefficients, while various function have been tested for variable diffusion. We start with the `GMRES(50)` algorithm

Table 2 Test problem in (41)

			GMRES(50)		Fixed PREC.		Updated PREC.	
α	β	Fill-in	IT	T(s)	IT	T(s)	IT	T(s)
1.6	1.2	5.38 %	†	†	4.45 26.24	7.68e-01	2.04 26.08	**2.95e-01**
1.3	1.8	6.51 %	†	†	4.55 27.57	7.79e-01	2.20 23.71	**3.09e-01**
1.5	1.5	4.41 %	†	†	4.43 28.04	7.61e-01	2.04 26.08	**2.93e-01**

$Nx = Ny = 80$, $Nt = 50$

Author's personal copy

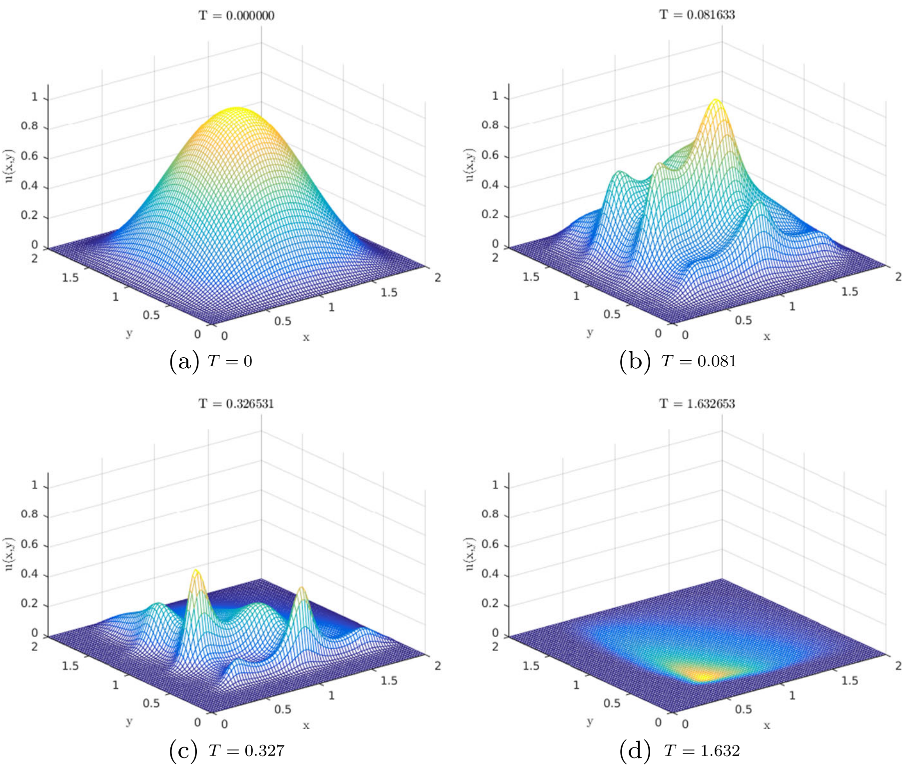

Fig. 3 Plot of the solution of the problem (41) for $\alpha = 1.6$, $\beta = 1.2$

with a tolerance of $\varepsilon = 1e - 6$. The drop tolerance for the approximate inverse preconditioners is set again to $\delta = 0.1$. Results for this case are collected in Tables 5, 6 and 7.

Experiment 3 We now consider a different but related model problem: a time-dependent 2D mixed fractional convection-diffusion equation, where fractional

Table 3 Test problem in (41)

			BiCGSTAB		Fixed PREC.		Updated PREC.	
α	β	Fill-in	IT	T(s)	IT	T(s)	IT	T(s)
1.6	1.2	5.38 %	†	†	134.78	5.74e-01	69.08	**2.94e-01**
1.3	1.8	6.51 %	†	†	127.18	5.38e-01	75.05	**3.21e-01**
1.5	1.5	4.41 %	†	†	123.46	5.08e-01	74.43	**3.09e-01**

$Nx = Ny = 80$, $Nt = 50$

Author's personal copy

Table 4 Test problem in (41)

			GMRES		Fixed PREC.		Updated PREC.	
α	β	Fill-in	IT	T(s)	IT	T(s)	IT	T(s)
1.6	1.2	5.38 %	†	†	1.00 159.55	1.89e+00	1.00 70.12	**3.30e-01**
1.3	1.8	6.51 %	†	†	1.00 159.86	1.85e+00	1.00 71.12	**3.37e-01**
1.5	1.5	4.41 %	†	†	1.00 161.12	1.91e+00	1.00 70.08	**3.28e-01**

$Nx = Ny = 80$, $Nt = 50$

diffusion is combined with classical transport. By using the same notation of the previous cases, we write

$$\begin{cases} u_t(x, y, t) = \mathcal{L}^{(\alpha,\beta)}u+ < (t_1(x, y), t_2(x, y)), \nabla u >, & (x, y) \in [0, 2]^2,\ t \geq 0 \\ u(x, y, 0) = u_0(x, y), & \\ u(x, y, t) = 0, & (x, y) \in \partial[0, 2]^2\ \forall t \geq 0. \end{cases} \tag{45}$$

Regarding the space variables, we discretize the fractional operator with shifted Grünwald-Letnikov approximation and the transport term with standard first order centered finite differences. For the time approximation, we use the backward Euler method to be consistent with the approximation error and for the dominant diffusion behavior of the equation.

We choose as a first set of coefficients for this experiment the following variable (in space) coefficients for the fractional operator $\mathcal{L}^{(\alpha,\beta)}$

$$\begin{aligned} d_+^\alpha(x, y, t) &= d_-^\alpha(x, y, t) = x^{4\alpha}y^{4\beta}, \\ d_+^\beta(x, y, t) &= d_-^\beta(x, y, t) = (2-x)^{4\alpha}(2-y)^{4\beta} \end{aligned} \tag{46}$$

while for the transport part we choose

$$t_1(x, y) = \frac{y+\beta}{x+y+\alpha}, \quad t_2(x, y) = \frac{x+\alpha}{x+y+\beta}, \tag{47}$$

Table 5 Test problem in (44)

			GMRES(50)		Fixed PREC.		Updated PREC.	
α	β	Fill-in	IT	T(s)	IT	T(s)	IT	T(s)
1.2	1.8	5.96 %	†	†	2.93 31.77	4.89e-01	1.52 27.48	**1.94e-01**
1.5	1.5	2.74 %	†	†	2.90 30.36	4.55e-01	1.55 24.33	**1.88e-01**
1.8	1.8	4.27 %	†	†	3.65 23.93	5.82e-01	2.19 20.54	**2.90e-01**
1.8	1.3	5.21 %	†	†	3.10 30.87	5.05e-01	1.67 28.26	**2.25e-01**

$Nx = Ny = 80$, $Nt = 70$ and $a(x, y) = \exp(-x^3 - y^3)$

Author's personal copy

Table 6 Test problem in (44)

α	β	Fill-in	GMRES IT	GMRES T(s)	Fixed PREC. IT	Fixed PREC. T(s)	Updated PREC. IT	Updated PREC. T(s)
1.8	1.8	4.27 %	†	†	1.00 130.94	1.18e+00	1.00 69.14	**3.16e-01**
1.8	1.3	5.21 %	†	†	1.00 114.45	9.56e-01	1.00 59.19	**2.55e-01**
1.5	1.5	2.74 %	†	†	1.00 107.43	9.30e-01	1.00 51.35	**2.24e-01**
1.2	1.8	5.96 %	†	†	1.00 111.30	9.95e-01	1.00 52.01	**2.22e-01**

$Nx = Ny = 80$, $Nt = 70$ and $a(x, y) = \exp(-x^3 - y^3)$

with the usual initial condition $u_0(x, y) = x^2 y^2 (2 - x)^2 (2 - y)^2$. The `GMRES(50)` algorithm is used with a fixed approximate inverse preconditioner with drop tolerance $\delta = 0.1$. The results are in Table 8.

A similar behavior is obtained also with the `GMRES` algorithm, with the same overall settings and the same drop tolerance for the approximate inverse preconditioner. Results are reported in Table 9.

Finally, we consider the solution with the `BiCGSTAB(2)` algorithm [19], instead of the classical `BiCGSTAB`, to deal with the possibility of having a discretization matrix with non-real eigenvalues, eigenvalues that are not approximated well by the first order factors of the polynomials built by the standard `BiCGSTAB`. Tolerance for the method and drop tolerance for the approximate inverse preconditioner are set to be the same of the ones for the other algorithms. Results are collected in Table 10.

Experiment 4 We consider the following constant coefficients fractional diffusion equation from [39]

$$\begin{cases} u_t(x,t) = K \frac{\partial^\alpha}{\partial |x|^\alpha} u(x,t), \ t \in [0,T], \ x \in [0,\pi], \ \alpha \in (1,2), \\ u(x,0) = x^2(\pi - x), \\ u(0,t) = u(\pi,t) = 0. \end{cases} \tag{48}$$

Table 7 Test problem in (44)

α	β	Fill-in	BiCGSTAB IT	BiCGSTAB T(s)	Fixed PREC. IT	Fixed PREC. T(s)	Updated PREC. IT	Updated PREC. T(s)
1.8	1.8	4.27 %	†	†	†	†	77.67	**3.34e-01**
1.8	1.3	5.21 %	†	†	116.08	4.97e-01	60.93	**2.63e-01**
1.5	1.5	2.74 %	†	†	117.43	4.96e-01	55.17	**2.32e-01**
1.2	1.8	5.96 %	†	†	107.55	4.65e-01	54.17	**2.36e-01**

$Nx = Ny = 80$, $Nt = 70$ and $a(x, y) = \exp(-x^3 - y^3)$

Author's personal copy

Table 8 Test problem in (45)

α	β	Fill-in	GMRES(50) IT	GMRES(50) T(s)	Preconditioned IT	Preconditioned T(s)
1.2	1.2	5.54 %	18.39 25.53	3.24e+00	2.37 24.39	**3.59e-01**
1.2	1.3	5.89 %	26.34 27.49	4.70e+00	2.34 25.95	**3.47e-01**
1.2	1.5	6.38 %	47.61 28.58	8.54e+00	2.34 22.10	**3.44e-01**
1.2	1.8	9.33 %	256.00 25.78	4.68e+01	2.00 18.20	**2.60e-01**
1.3	1.2	6.07 %	25.93 29.05	4.69e+00	2.00 22.27	**2.79e-01**
1.3	1.3	5.51 %	31.63 28.86	5.62e+00	2.47 27.49	**3.86e-01**
1.3	1.5	6.14 %	56.44 24.20	1.04e+01	2.39 31.37	**3.95e-01**
1.3	1.8	8.53 %	387.71 23.53	7.04e+01	2.22 21.69	**3.15e-01**
1.8	1.2	8.05 %	294.10 29.90	5.34e+01	1.56 26.95	**2.14e-01**
1.8	1.3	7.39 %	477.75 28.00	8.80e+01	1.61 29.17	**2.32e-01**
1.8	1.5	7.05 %	†	†	2.25 21.75	**3.20e-01**
1.8	1.8	7.18 %	†	†	3.71 28.61	**6.33e-01**

$Nx = Ny = 80$, $Nt = 60$

whose analytical solution, on an infinite domain, is given by

$$u(x,t) = \sum_{n=1}^{+\infty} \left(\frac{8}{n^3}(-1)^{n+1} - \frac{4}{n^3} \right) \sin(nx) \exp\left(-n^{\alpha} K t\right). \tag{49}$$

To solve numerically the problem in (48), we consider both a direct application of the short–memory principle, i.e., we extract a banded approximation of the discretization matrix, and the use of approximate inverses instead of the true inverses. Both the results will be compared with the solution(s) obtained by solving the underlying sequence of linear systems.

As a first test, we consider the discretization of the symmetric Riesz FDE as a half-sum of left- and right-sided Caputo derivatives, using the backward Euler scheme for advancing in time. We set the number of diagonals to be extracted as $d = 150$, and, to obtain a similar bandwidth in the inverse, drop tolerances of $\delta = 5e-7$ and $\delta = 1e-7$. The averages of 2-norm of the errors are reported in Table 11. For the choice of $\delta = 5e-7$, the bandwidth of the approximate inverse is 334 (instead of the 300 given by the direct application of the short–memory principle) and an error that is of comparable modulus. The time needed for the solution is $T = 0.93s$ with the band-approximation while we get $T = 0.08s$ by using the approximate inverses. On the other hand, if we decrease the drop tolerances, we can obtain a solution with

Author's personal copy

Table 9 Test problem in (45)

			GMRES		Preconditioned	
α	β	Fill-in	IT	T(s)	IT	T(s)
1.2	1.2	5.54 %	1.00 365.86	5.51e+00	1.00 79.22	**4.00e-01**
1.2	1.3	5.89 %	1.00 443.95	7.97e+00	1.00 79.92	**4.07e-01**
1.2	1.5	6.38 %	1.00 660.56	1.69e+01	1.00 78.15	**4.06e-01**
1.2	1.8	9.33 %	†	†	1.00 64.12	**2.87e-01**
1.3	1.2	6.07 %	1.00 436.73	7.72e+00	1.00 66.49	**3.08e-01**
1.3	1.3	5.51 %	1.00 514.36	1.06e+01	1.00 83.31	**4.37e-01**
1.3	1.5	6.14 %	†	†	1.00 82.80	**4.28e-01**
1.3	1.8	8.53 %	†	†	1.00 74.12	**3.64e-01**
1.5	1.2	6.06 %	1.00 637.86	1.58e+01	1.00 59.08	**2.59e-01**
1.5	1.3	6.16 %	†	†	1.00 66.36	**3.03e-01**
1.5	1.5	5.78 %	†	†	1.00 95.56	**5.45e-01**
1.5	1.8	7.35 %	†	†	1.00 92.97	**5.06e-01**
1.8	1.2	8.05 %	†	†	1.00 53.88	**2.34e-01**
1.8	1.3	7.39 %	†	†	1.00 57.36	**2.52e-01**
1.8	1.5	7.05 %	†	†	1.00 73.69	**3.67e-01**
1.8	1.8	7.18 %	†	†	1.00 115.46	**7.25e-01**

$Nx = Ny = 80$, $Nt = 60$

also a smaller error than the one obtained by solving the sequence of linear system with Gaussian elimination implemented in MATLAB, i.e., the well-known \, because the good information are already completely included in the underlying reduced model.

We also consider discretizing the symmetric Riesz derivatives with Ortigueira's centered fractional differences scheme, again by using backward Euler for advancing in time. The averages of the 2-norm of the errors are reported in Table 11. In this case, we obtain a bandwidth of 330 for the approximate inverses that can be compared with 300 of the direct approximation. The timings are $T = 0.77s$ with the direct application of the short–memory principle and $T = 0.09s$ by using the approximate inverses. The profile of the relative error has the same behavior of the former discretization.

We experimented also the possibility outlined in Remark 1, i.e., the use of a banded approximation of the discretization matrix and of the approximate inverse as the true inverses. We observe that in these cases, the effect of the terms of small norm is adding noise, i.e., ill-conditioning of the matrices, that reduces the overall accuracy obtained by the discretization methods.

Author's personal copy

Table 10 Test problem in (45)

			GMRES		Preconditioned	
α	β	Fill-in	IT	T(s)	IT	T(s)
1.2	1.2	5.54 %	220.75	1.43e+00	44.88	**3.45e-01**
1.2	1.3	5.89 %	303.59	1.95e+00	44.69	**3.43e-01**
1.2	1.5	6.38 %	†	†	46.27	**3.48e-01**
1.2	1.8	9.33 %	†	†	39.61	**3.07e-01**
1.3	1.2	6.07 %	297.31	1.94e+00	36.27	**2.73e-01**
1.3	1.3	5.51 %	384.61	2.49e+00	49.42	**3.70e-01**
1.3	1.5	6.14 %	†	†	52.27	**3.93e-01**
1.3	1.8	8.53 %	†	†	46.22	**3.67e-01**
1.5	1.2	6.06 %	†	†	32.20	**2.42e-01**
1.5	1.3	6.16 %	†	†	37.92	**2.86e-01**
1.5	1.5	5.78 %	†	†	60.53	**4.54e-01**
1.5	1.8	7.35 %	†	†	64.41	**4.91e-01**
1.8	1.2	8.05 %	†	†	31.63	**2.42e-01**
1.8	1.3	7.39 %	†	†	34.68	**2.67e-01**
1.8	1.5	7.05 %	†	†	46.03	**3.50e-01**
1.8	1.8	7.18 %	†	†	80.80	**6.14e-01**

$Nx = Ny = 80$, $Nt = 60$

Table 11 Test problem in (48)

			Complete		Banded		Direct	
Type	α	K	$T(s)$	err.	$T(s)$	err.	$T(s)$	err.
C	1.8	0.25	9.39	2.42e-02	0.93	4.29e-02	**0.08**	4.24e-02
							0.16	1.64e-02
O	1.8	0.25	9.27	3.03e-02	0.77	4.29e-02	**0.09**	4.43e-02
							0.11	3.33e-02
O*	1.5	0.75	8.64	7.43e-02	0.58	2.62e-01	**0.10**	7.66e-02
O*	1.2	1.25	9.43	5.27e-01	0.63	2.78e-01	**0.06**	2.73e-01
C*	1.8	1.50	8.66	1.15e-01	1.05	1.53e-01	**0.16**	1.83e-01

Timings and averages of the 2-norm of the errors over all time steps. C discretization using the half-sum of Caputo derivatives, O Ortigueira discretization, C^* and O^* our method applied to the banded approximation of the discretization matrix (see Remark 1). Column *Complete*: the reference solution is used with the standard discretization matrix. Column *Banded*: the *Short–memory principle* is applied to the discretization of the operator. Column *Direct*: our approach using the approximate inverses. All the discretizations use $N = 2^{10}$

Author's personal copy

5 Conclusions and possible extensions

The use of a structural property of the differential operator, i.e., the *short–memory principle*, and the fact that it is inherited by the discretization, gave us a chance for the efficient use of sparse approximate inverses to solve FDEs. The former have been used both for preconditioning Krylov iterative solvers and as direct methods through the use of matrix-vector products. We stress that, differently from the approach relying only on the structure of the matrix, ours permits to consider cases in which fractional and classical derivatives appear together. We observed that our proposal can exploit the high-performance capabilities of GPGPU computing, providing also a fast build of the preconditioners.

We plan to increase the use and the degree of GPU parallelism in a future work by implementing PSBLAS and using more than one GPU in order to increase the speedups.

Acknowledgments The authors would like to thank one of the referees for helpful comments which have improved our presentation.

References

1. Bellavia, S., Bertaccini, D., Morini, B.: Nonsymmetric preconditioner updates in Newton-Krylov methods for nonlinear systems. SIAM J. Sci. Comput. **33**(5), 2595–2619 (2011)
2. Benzi, M.: Preconditioning techniques for large linear systems: a survey. J. Comp. Phys. **182**(2), 418–477 (2002)
3. Benzi, M., Bertaccini, D.: Approximate inverse preconditioning for shifted linear systems. BIT **43**(2), 231–244 (2003)
4. Bertaccini, D.: Efficient preconditioning for sequences of parametric complex symmetric linear systems. ETNA **18**, 49–64 (2004)
5. Bertaccini, D., Durastante, F.: Interpolating preconditioners for the solution of sequence of linear systems. Comp. Math. Appl. **72**, 1118–1130 (2016)
6. Bertaccini, D., Filippone, S.: Sparse approximate inverse preconditioners on high performance gpu platforms. Comp. Math. Appl. **71**(3), 693–711 (2016)
7. Bridson, R., Tang, W.-P.: Refining an approximate inverse. J. Comput. Appl. Math. **123**(1), 293–306 (2000)
8. Bridson, R., Tang, W.-P.: Multiresolution approximate inverse preconditioners. SIAM J. Sci. Comput. **23**(2), 463–479 (2001)
9. Canuto, C., Simoncini, V., Verani, M.: On the decay of the inverse of matrices that are sum of kronecker products. Lin. Alg. Appl. **452**, 21–39 (2014)
10. Ċelik, C., Duman, M.: Crank–Nicolson method for the fractional diffusion equation with the Riesz fractional derivative. J. Comput Phys. **231**(4), 1743–1750 (2012)
11. Dalton, S., Bell, N., Olson, L., Garland, M.: Cusp: generic parallel algorithms for sparse matrix and graph computations, version 0.5.0 (2014). http://cusplibrary.github.io/
12. D'Ambra, P., Serafino, D.D., Filippone, S.: Mld2p4: a package of parallel algebraic multilevel domain decomposition preconditioners in fortran 95. ACM T. Math. Software **37**(3), 30 (2010)
13. Demko, S., Moss, W.F., Smith, P.W.: Decay rates for inverses of band matrices. Math. Comput. **43**(168), 491–499 (1984)
14. Deng, W.: Short memory principle and a predictor–corrector approach for fractional differential equations. J. Comput. Appl Math. **206**(1), 174–188 (2007)

Author's personal copy

15. Ding, H., Li, C., Chen, Y.: High–order algorithms for Riesz derivative and their applications (II). J. Comput. Phys. **293**, 218–237 (2015)
16. Durastante, F.: Interpolant update of preconditioners for sequences of large linear systems. In: Mathematical Methods, Computational Techniques and Intelligent Systems (MAMECTIS '15), vol. 41, pp. 40–47. WSEAS Press (2015)
17. Filippone, S., Colajanni, M.: Psblas: a library for parallel linear algebra computation on sparse matrices. ACM T. Math. Software **26**(4), 527–550 (2000)
18. Fornberg, B.: Classroom note: calculation of weights in finite difference formulas. SIAM Rev. **40**(3), 685–691 (1998)
19. Gutknecht, M.H.: Variants of BICGSTAB for matrices with complex spectrum. SIAM J. Sci. Comput. **14**(5), 1020–1033 (1993)
20. Jaffard, S.: Propriétés Des Matrices ≪Bien Localisées≫ Près De Leur Diagonale Et Quelques Applications. In: Ann. I. H. Poincare-An, vol. 7, pp. 461–476 (1990)
21. Krishtal, I., Strohmer, T., Wertz, T.: Localization of matrix factorizations. Found. Comput. Math. **15**(4), 931–951 (2015)
22. Lampret, V.: Estimating the sequence of real binomial coefficients. J. Ineq. Pure and Appl. Math. **7**(5)
23. Li, C., Chen, A., Ye, J.: Numerical approaches to fractional calculus and fractional ordinary differential equation. J. Comput. Phys. **230**(9), 3352–3368 (2011)
24. Li, C., Zeng, F.: Numerical methods for fractional calculus, vol. 24. CRC Press (2015)
25. Lubich, C.: Discretized fractional calculus. SIAM J. Math. Anal. **17**(3), 704–719 (1986)
26. Meerschaert, M.M., Tadjeran, C.: Finite difference approximations for fractional advection–dispersion flow equations. J. Comput. Appl. Math. **172**(1), 65–77 (2004)
27. Ng, M.K., Pan, J.: Approximate inverse circulant–plus–diagonal preconditioners for Toeplitz–plus–diagonal matrices. SIAM J. Sci. Comput. **32**(3), 1442–1464 (2010)
28. Ortigueira, M.D.: Riesz potential operators and inverses via fractional centred derivatives. Int. J. Math. Math. Sci (2006)
29. Pan, J., Ke, R., Ng, M.K., Sun, H.-W.: Preconditioning techniques for diagonal–times–Toeplitz matrices in fractional diffusion equations. SIAM J. Sci. Comput. **36**(6), A2698–A2719 (2014)
30. Podlubny, I.: Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications, vol. 198. Academic Press (1998)
31. Podlubny, I.: Matrix approach to discrete fractional calculus. Fract. Calc. Appl. Anal. **3**(4), 359–386 (2000)
32. Podlubny, I.: Geometric and physical interpretation of fractional integration and fractional differentiation. Fract. Calc. Appl. Anal. **5**(4), 367–386 (2002)
33. Podlubny, I., Chechkin, A., Skovranek, T., Chen, Y., Jara, B.M.V.: Matrix approach to discrete fractional calculus II: partial fractional differential equations. J. Comput. Phys. **228**(8), 3137–3153 (2009)
34. Podlubny, I., Skovranek, T., Jara, B.M.V., Petras, I., Verbitsky, V., Chen, Y.: Matrix approach to discrete fractional calculus III: non-equidistant grids, variable step length and distributed orders. Philos. T. R. Soc. A **371**(1990), 20120153 (2013)
35. Popolizio, M.: A matrix approach for partial differential equations with Riesz space fractional derivatives. Eur. Phys. J.-Spec. Top. **222**(8), 1975–1985 (2013)
36. Saad, Y.: Iterative methods for sparse linear systems. SIAM (2003)
37. Srivastava, H.M., Trujillo, J.J.: Theory and applications of fractional differential equations. Elsevier, Amsterdam (2006)
38. Wendel, J.: Note on the gamma function. Am. Math. Mon. **55**(9), 563–564 (1948)
39. Yang, Q., Liu, F., Turner, I.: Numerical methods for fractional partial differential equations with Riesz space fractional derivatives. Appl. Math. Model. **34**(1), 200–218 (2010)