Applied Mathematics Letters 95 (2019) 92–97

Contents lists available at ScienceDirect

Applied Mathematics Letters

www.elsevier.com/locate/aml

Block structured preconditioners in tensor form for the all-at-once solution of a finite volume fractional diffusion equation[☆]

D. Bertaccini[a,b,*], F. Durastante[c]

[a] *Università di Roma "Tor Vergata", Dipartimento di Matematica, viale della Ricerca Scientifica 1, Roma, Italy*
[b] *Istituto per le Applicazioni del Calcolo (IAC) "M. Picone", National Research Council (CNR), Roma, Italy*
[c] *Università di Pisa, Dipartimento di Informatica, largo Bruno Pontecorvo 3, Pisa, Italy*

ARTICLE INFO

Article history:
Received 28 January 2019
Received in revised form 20 March 2019
Accepted 20 March 2019
Available online 29 March 2019

Keywords:
Structured preconditioners in tensor form
Tensor trains
Fractional diffusion equations
Finite volumes

ABSTRACT

We propose a tensor structured preconditioner for the tensor train GMRES algorithm (or TT-GMRES for short) to approximate the solution of the all-at-once formulation of time-dependent fractional partial differential equations discretized in time by linear multistep formulas used in boundary value form and in space by finite volumes.

Numerical experiments show that the proposed preconditioner is efficient for very large problems and is competitive, in particular with respect to the AMEn algorithm.

1. Introduction

Solution of partial differential equations of fractional order, or FDEs for short, is an emerging crucial topic since FDEs naturally occurs in many applications in a variety of fields, e.g., physics, population dynamics, viscoelastic and viscoplastic phenomena, electrical networks, and many others, see, e.g., [1]. Here we deal with the solution of the following boundary value conservative fractional diffusion equation with Riesz derivatives of order $2-\alpha_\ell$ with $\alpha_\ell \in (0,1)$ [2]

$$\frac{\partial u}{\partial t} - \sum_{j=1}^{\ell} \frac{\partial}{\partial x_j}\left[\kappa_j(\mathbf{x})\frac{\partial^{\alpha_\ell} u}{\partial |x_j|}\right] = z(\mathbf{x},t), \quad \begin{array}{l} x_j \in (L_j,R_j)^\ell, \\ t \in (t_0,T], \end{array} \qquad \ell = 1,2,\ldots \tag{1}$$

[☆] This work was partially supported by INdAM-GNCS project "Tecniche innovative per problemi di algebra lineare" (2018); by the MIUR Excellence Department Project of the Department of Mathematics, University of Rome Tor Vergata, CUP E83C18000100006; and by the "MISSION: SUSTAINABILITY" project "NUMnoSIDS", CUP E86C18000530005.

[*] Corresponding author at: Università di Roma "Tor Vergata", Dipartimento di Matematica, viale della Ricerca Scientifica 1, Roma, Italy.

E-mail addresses: bertaccini@mat.uniroma2.it (D. Bertaccini), fabio.durastante@di.unipi.it (F. Durastante).

https://doi.org/10.1016/j.aml.2019.03.028
0893-9659/© 2019 Elsevier Ltd. All rights reserved.

where

$$\frac{\partial^{\alpha_\ell} u}{\partial |x_j|} = \eta(\alpha_\ell)\left(\frac{\partial^{1-\alpha_\ell} u}{\partial_+ |x_j|^{1-\alpha_\ell}} + \frac{\partial^{1-\alpha_\ell} u}{\partial_- |x_j|^{1-\alpha_\ell}}\right), \quad \eta(\alpha_\ell) = -\frac{1}{2\cos\left(\frac{(1-\alpha_\ell)\pi}{2}\right)},$$

for $\kappa_j(\mathbf{x}) \geq 0$ the diffusion coefficients, $z(\mathbf{x},t)$ the source term, $\frac{\partial^{1-\alpha_\ell} u}{\partial_\pm |x_j|^{1-\alpha_\ell}}$ the left- and right-sided Riemann–Liouville derivatives, coupled with zero Dirichlet boundary conditions, and an initial condition $u(x,0) = u_0(x)$. We propose here a structured preconditioner to be applied in tensor train form [3] (TT for short) for the all-at-once formulation [4,5] of the underlying problem to be used with the TT-GMRES algorithm [6,7].

2. Discrete boundary value problem formulation

For a fixed integer $m > 0$ we consider the partition of (L_1, R_1) with interfaces $\{x_i = L_1 + i\Delta x; \Delta x = (R_1 - L_1)/(m+1)\}_{i=0}^{m+1}$, and denote their centers as $x_{i-1/2}$, for $i = 1,\ldots,m+1$. The finite volume discretization in space of (1) for $\ell = 1$ reads

$$\mathbf{u}_t(t) = J_m^{(1)}\mathbf{u}(t) + \mathbf{z}(t),\ J_m^{(1)} = K_{m,1}^{-} G_m^{(\alpha_1)} + K_{m,1}^{+} {G_m^{(\alpha_1)}}^T,\ (\mathbf{u}(t))_{i=1}^m = u(x_i,t), \tag{2}$$

where $K_{m,1}^{\pm} = \mathrm{diag}(\{\kappa_1(x_{i\pm 1/2})\}_{i=1}^m)$, $G_m^{(\alpha_1)}$ is the Toeplitz matrix $(G_m^{(\alpha_\ell)})_{r,s} = g_{r-s}^{(\alpha_\ell)}$

$$g_k^{(\alpha_\ell)} = \begin{cases} 2(-k+\frac{1}{2})^{\alpha_\ell} - (-k-\frac{1}{2})^{\alpha_\ell} - (-k+\frac{3}{2})^{\alpha_\ell}, & 1-m \leq k \leq -1, \\ \frac{3}{2^{\alpha_\ell}} - \frac{3^{\alpha_\ell}}{2^{\alpha_\ell}}, & k = 0, \\ \frac{3^{\alpha_\ell}}{2^{\alpha_\ell}} - \frac{3}{2^{\alpha_\ell}}, & k = 1, \\ -2(k-\frac{1}{2})^{\alpha_\ell} + (k+\frac{1}{2})^{\alpha_\ell} + (k-\frac{3}{2})^{\alpha_\ell} & 2 \leq k \leq m-1, \end{cases}$$

and

$$(\mathbf{z}(t))_i = z(t)_i = \frac{\Gamma(\alpha_1+1)\Delta x^{1-\alpha_1}}{\eta(\alpha_1)} \int_{x_{i-1/2}}^{x_{i+1/2}} s(x,t)dx,\ i = 1,\ldots,m.$$

The fully discrete system, in the form of a discrete boundary value problem, is then obtained by applying to (2) a fully implicit method for differential equations based on linear multistep formulas in boundary value form [8–10]. We consider the k-step Generalized Backward Differentiation Formula, or GBDF for short, because of the stiffness of the underlying problem (see, e.g., [4] for more details) over a uniform mesh $t_j = t_0 + jh$, for $j = 0,\ldots,s$, and $h = (T-t_0)/s$:

$$\sum_{i=-\nu}^{k-\nu} \alpha_{i+\nu}\mathbf{u}_{n+i} = h\mathbf{f}_{n+i},\ \begin{array}{l} n = \nu,\ldots,s-k+\nu, \\ \mathbf{f}_n = J_m\mathbf{u}_n + \mathbf{z}_n, \end{array} \quad \nu = \begin{cases} \frac{k+2}{2}, & k \text{ even}, \\ \frac{k+2}{1}, & k \text{ odd}, \end{cases} \tag{3}$$

complemented by additional $(k-1)$ equations for the auxiliary initial and final values. Thus, the required k-order method is determined by taking the coefficients $\{\alpha_i\}_{i=0}^k$ as the solution of the Vandermonde linear system

$$W_\nu^{(k)}\boldsymbol{\alpha} \equiv \begin{bmatrix} 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ -\nu & \cdots & -1 & 0 & 1 & \cdots & k-\nu \\ \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ (-\nu)^k & \cdots & (-1)^k & 0 & 1 & \cdots & (k-\nu)^k \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \mathbf{e}_2 \equiv \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

By assembling together the equations for all the time steps we find

$$\mathcal{M}\mathbf{u} \equiv (A \otimes I_m - hI_{s+1} \otimes J_m^{(\ell)})\mathbf{u} = \mathbf{e}_1 \otimes \mathbf{u}_0 + h(I_{s+1} \otimes I_m)\mathbf{z} \equiv \mathbf{b}, \tag{4}$$

where $\mathbf{e}_1 = [1, 0, \dots, 0]^T \in \mathbb{R}^{s+1}$, $\mathbf{u} = [\mathbf{u}_0^T, \dots, \mathbf{u}_s^T]^T \in \mathbb{R}^{(s+1)m}$, $\mathbf{g} = [\mathbf{g}_0^T, \dots, \mathbf{g}_s^T]^T \in \mathbb{R}^{(s+1)m}$ while the matrix $A \in \mathbb{R}^{(s+1)\times(s+1)}$ is generated by the coefficients of the formula (3) and of the auxiliary linear multistep formulas producing the GBDF's $(\nu, k-\nu)$-boundary conditions

$$\sum_{i=0}^{k} \alpha_{i,j}\mathbf{u}_i = h\mathbf{f}_j,\ j = 1, \dots, \nu - 1, \quad \sum_{i=0}^{k} \alpha_{i,j}\mathbf{u}_{s-\nu-1+i} = h\mathbf{f}_{m-\nu-1+j},\ j = \nu + 1, \dots, k,$$

where the $\{\alpha_{i,j}\}_j$ are computed to get an overall $\mathcal{O}(h^{k+1})$ truncation error, see [10, Chapter 5.4] for details. In general, for $\ell > 1$, the Jacobian $J_m^{(\ell)}$ is then expressed as the following sum of Kronecker products in "Laplace-like" form [3] discretizing the fractional derivatives

$$J_m^{(\ell)} = \sum_{i=1}^{\ell} \left(K_{m,\ell}^{-} \bigotimes_{p=1}^{i-1} I \otimes G_{m^{1/\ell}}^{(\ell)} \otimes \bigotimes_{p=1}^{\ell-1} I + K_{m,\ell}^{+} \bigotimes_{p=1}^{i-1} I \otimes {G_{m^{1/\ell}}^{(\ell)}}^T \otimes \bigotimes_{p=1}^{\ell-1} I \right),$$

in which the $K_{m,\ell}^{\pm}$ have also a Kronecker tensor structure whenever the functions $\{\kappa_j\}_{j=1}^{\ell}$ are separable in the x_j variables.

2.1. Tensor representation and matrix–vector product in tensor format

To efficiently solve the linear system (4), we need to exploit its tensorial structure, see [5]. Recall that a tensor $\mathcal{X}$ is a multidimensional array $\mathcal{X} = [X(i_1, \dots, i_d)]$, $i_k \in \{1, \dots, n_k\}$, in which d is the tensor dimensionality, i.e., the number of indices, and $n_1 \times \cdots \times n_d$ is the size of the tensor, i.e., n_i is the number of nodes along each index. The exponential growth of the number of elements (n^d) makes impossible using component-wise storage and operations, thus several compression (representation) of such objects have been developed in the literature [11]. We focus here on the TT-tensor representation [3,12]. Any tensor with dimensionality d has $d - 1$ unfoldings of the form

$$X_k = [X(i_1, \dots, i_k; i_{k+1}, \dots, i_d)],\ \text{where } X(i_1, \dots, i_k; i_{k+1}, \dots, i_d) = X(i_1, \dots, i_d),$$

$i_1, \dots, i_k$, $i_{k+1}, \dots, i_d$ are row and column multi-indices, and the X_k are matrices of size $Q_k \times R_k$ with $Q_k = n_1 \times \cdots \times n_k$, and $R_k = n_{k+1} \times \cdots n_d$. Then, a d-dimensional tensor $\mathcal{X}$ with size $n_1 \times \cdots \times n_d$ is said to be in the TT-format with cores X_k of size $r_{k-1} \times n_k \times r_k$, for $k = 1, \dots, d$, $r_0 = r_d = 1$, if its elements are defined by

$$\mathcal{X} = X(i_1, \dots, i_d) = \sum_{\alpha_0, \dots, \alpha_d} X_1(\alpha_0, i_1, \alpha_1) X_2(\alpha_1, i_2, \alpha_2) \dots X_d(\alpha_{d-1}, i_d, \alpha_d), \tag{5}$$

and the r_k are called TT-ranks. Any dense tensor can be converted into the TT-format by the TT-SVD algorithm, and also the right-hand side of (4) can be restated in this format straightforwardly. Indeed, if a vector of length $N = n_1 \times \cdots \times n_d$ is treated as a d-dimensional tensor with mode sizes n_k, and represented in TT-format, the matrices acting on it have the form

$$\mathcal{M}(i_1, \dots, i_d, j_1, \dots, j_d) = M_1(i_1, j_1) \dots M(i_d, j_d), \qquad M_k(i_k, j_k) \in \mathbb{R}^{r_{k-1}\times r_k},$$

where $(i_1, \dots, i_d)$ enumerates the row of $\mathcal{M}$, and $(j_1, \dots, j_d)$ enumerates the columns. If we have a matrix $\mathcal{M}$ in TT-format, and a vector $\mathcal{X}$ in TT-format with cores X_k, and entries $X(j_1, \dots, j_d)$ then the matrix–vector multiplication amounts to the following sum

$$\mathcal{Y} = Y(i_1, \dots, i_d) = \sum_{j_1, \dots, j_d} \mathcal{M}(i_1, \dots, i_d, j_1, \dots, j_d)\mathcal{X}(j_1, \dots, j_d) = Y_1(i_1) \dots Y_d(i_d),$$

where $Y_k(i_k) = \sum_{j_k} M_k(i_k, j_k) \otimes X_k(j_k)$, from which we see that the ranks of $\mathcal{Y}$ are the product of the ranks of the matrix and of the vector. Thus, as expected (see, e.g., [3]), we experience that often the result should be approximated during the computations to avoid a rapid rank explosion.

3. The solution strategy

By exploiting the techniques we developed in [4,13], we propose a preconditioner based on the structure of (4) that can be written in the form

$$\begin{aligned} P =&\breve{A} \otimes I - h\breve{B} \otimes \tilde{J}_m^{(\ell)} = U\Lambda_A U^H \otimes I - hU\Lambda_B U^H \otimes \tilde{J}_m^{(\ell)} \\ =&(U \otimes I)(\Lambda_A \otimes I_m - h\Lambda_B \otimes \tilde{J}_m^{(\ell)})(U^H \otimes I), \qquad \text{with } U^H U = I, \Lambda_A, \Lambda_B \text{ diagonal,} \end{aligned} \tag{6}$$

where $\breve{A}$ and $\breve{B}$ are the projection of the Toeplitz-like matrices A and $B = I_{s+1}$ on the $\{-1\}$-circulant algebra [4,9], and $\{\tilde{J}_m^{(\ell)}\}_m$ is an approximation of the Jacobian matrix. For the considered model problem, it is the scaled Laplacian [14]

$$\tilde{J}_m^{(\ell)} = \sum_{i=1}^{\ell} \left(K_{m,\ell}^{-} + K_{m,\ell}^{+} \right) \bigotimes_{p=1}^{i-1} I \otimes \Delta_{m^{1/\ell}} \otimes \bigotimes_{p=1}^{\ell-1} I, \quad \Delta_{m^{1/\ell}} = \operatorname{tridiag}(-1, 2, -1) \in \mathbb{R}^{m^{1/\ell} \times m^{1/\ell}}. \tag{7}$$

To apply the preconditioner in the TT-GMRES algorithm [7], we need a suitable approximation of the inverse of (6) in TT-format [15]. First, the orthogonal matrix $(U \otimes I)$ in (6) is easy to represent, invert and use in tensor format. Then, consider the inverse of the block-diagonal part $(\Lambda_A \otimes I_m - h\Lambda_B \otimes \tilde{J}_m^{(\ell)}) = \Lambda(A, B, \tilde{J}_m^{(\ell)})$. We look for X such that $X\Lambda(A, B, \tilde{J}_m^{(\ell)}) = I$, i.e., such that $(I \otimes \Lambda(A, B, J_m)^T)\mathbf{x} = \operatorname{vec}(I)$, where $\mathbf{x} = \operatorname{vec}(X)$ is a vectorized form of X. Since we use the TT-format for $\Lambda(A, B, \tilde{J}_m^{(\ell)})$, we apply an approximate inversion method for tensors proposed in [6,15] to keep bounded the solution rank r, getting a candidate approximation X_r of X, and then the overall preconditioner P is necessarily approximated by P_r as well:

$$P^{-1} \approx P_r^{-1} = (U \otimes I)X_r(U^H \otimes I). \tag{8}$$

By using arguments similar to those in [9, Theorem 4] and considering negligible the effect of the approximate inversion that produces P_r above, we can prove the clustering result below.

Theorem 1. *Let us consider the limited memory block ω-circulant preconditioner (6) such that $\omega = \exp(i\omega\theta)$, $\theta = \pi$ and $\tilde{J}_m$ as in (7). If there exists $\epsilon \in (0, 1)$ not depending on the mesh, provided fine enough, and $\|P_r^{-1} - P^{-1}\| \leq \epsilon\|P^{-1}\|$, then the eigenvalues of the preconditioned matrix $P_r^{-1} M$ are equal to $1 \in \mathbb{C}$ except for $\mathcal{O}(m^{\frac{\alpha}{2}})$ outliers.*

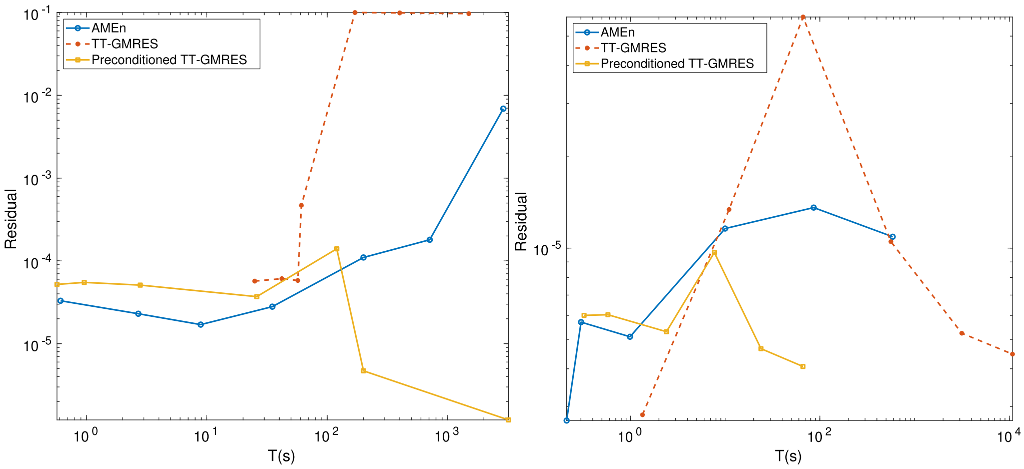
Note that in general the deterioration needed to keep an upper bound to the tensorial rank can destroy the underlying cluster.

4. Numerical examples

The numerical tests are performed with Matlab 7.13.0.564 (R2011b), and the TT-Toolbox [3] Version 2.2.2, on a Intel® Xeon® CPU X5680 3.33 GHz, with 24 Gb of RAM. We solve (1) on the domain $(L_i, R_i) = (0, 1)$, $(t_0, T] = (0, 5]$, for the following choice of the coefficients

$$\begin{array}{lll} \ell = 1, \quad \kappa_1 = \Gamma(1+\alpha)(1+x_1) & z = x_1(1+x_1)^{1+\alpha}e^t, & u_0 = \sin(\pi x_1), \\ \ell = 2, \quad \begin{array}{l} \kappa_1 = \Gamma(1+\alpha_1)(1+x_1)(1+x_2) \\ \kappa_2 = \Gamma(1+\alpha_2)(1-x_1)^2(1-x_2)^2. \end{array} & z = x_1 x_2 (1+x_2)(1+x_1)^{1+\alpha_1}e^t, & u_0 = \sin(\pi x_1)\sin(\pi x_2) \end{array}$$

To advance in time with an error that is comparable with the one produced in space, the 2 steps GBDF formula is used. To benchmark our preconditioner (7), we compare it against the AMEn solver [5,16] that is set to achieve a tolerance on the residual of $\tau = 1e-6$, with a maximum rank for the residuals of 10, a maximum number of $20\log_2(m^{1/\ell})$ sweeps, and the truncation computed with respect to the Frobenius norm.



(a) 1D – $m = s = 2^k$,$k = 6, 7, \dots, 12$, $\alpha = 0.2$.

(b) 2D – $s = 2^k$, $m = 2^k \times 2^k$,$k = 3, 4, \dots, 8$, $\alpha_1 = \alpha_2 = 0.2$.

Fig. 1. Time–Residual graphs for the test problem for $\ell = 1, 2$, every marker corresponds to a problem of different (growing) size.

The TT-GMRES [7] is set to achieve the same tolerance τ, the method is restarted every 20 iterations and the maximum number of iteration is $20 \log_2(m^{1/\ell})$. The compression/truncation tolerances for the method are $\varepsilon_x = 1e-3$, $1e-6$ for the compression of the solution, and $\varepsilon_z = 1e-4$, $1e-7$ for the compression of the Krylov basis vectors for $\ell = 1$, 2 respectively. The maximum allowed rank r for the construction of (8) is $r = 4$, corresponding to a residual on the inverse of approximately 10^{-2}. Details on the parameter used above are in [7]. The results are given in Fig. 1. We can observe that the TT-GMRES without preconditioner either fails to achieve convergence or reaches convergence after many iterations, as it happens also for standard nonpreconditioned restarted GMRES [4]. This is observed for more general cases in [16]. On the other hand, our preconditioner delivers a reasonable reduction of the timings with respect to the AMEn solver, whose convergence also deteriorates increasing the size of the problem.

5. Conclusions and perspectives

We extended and specialized an existing preconditioning technique for structured linear systems by exploiting suitable tensor representations. This permits to use the preconditioner in a tensor setting. To improve the performances of the AMEn solver, the solution of the auxiliary linear systems arising in each optimization step of the underlying outer algorithm should be further investigated by focusing on their structure. On the other hand, the experiments we performed (see, e.g., Fig. 1) show that the application of our structured TT-preconditioner to the TT-GMRES algorithm performs reasonably better.

We plan to investigate the use of QTT-tensor representation [17] for the whole linear system and carefully analyze the nontrivial deterioration of the cluster of the eigenvalues of the preconditioned matrices due to the upper bound on the tensorial rank. The implementation on GPGPUs, taking advantage of the underlying shared memory framework will be also considered.

Acknowledgments

The authors would like to thank the referees for their valuable comments.

References

[1] A.A. Kilbas, H.M. Srivastava, J.J. Trujillo, Theory and Applications of Fractional Differential Equations, North-Holland Mathematics Studies, vol. 204, Elsevier Science B.V., Amsterdam, 2006, p. xvi+523, ISBN: 978-0-444-51832-3; ISBN: 0-444-51832-0.

[2] H. Wang, N. Du, A superfast-preconditioned iterative method for steady-state space-fractional diffusion equations, J. Comput. Phys. (ISSN: 0021-9991) 240 (2013) 49–57, http://dx.doi.org/10.1016/j.jcp.2012.07.045.

[3] I.V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. (ISSN: 1064-8275) 33 (5) (2011) 2295–2317, http://dx.doi.org/10.1137/090752286.

[4] D. Bertaccini, F. Durastante, Limited memory block preconditioners for fast solution of fractional partial differential equations, J. Sci. Comput. (ISSN: 0885-7474) 77 (2) (2018) 950–970, http://dx.doi.org/10.1007/s10915-018-0729-3.

[5] T. Breiten, V. Simoncini, M. Stoll, Low-rank solvers for fractional differential equations, Electron. Trans. Numer. Anal. (ISSN: 1068-9613) 45 (2016) 107–132.

[6] I.V. Oseledets, S.V. Dolgov, Solution of linear systems and matrix inversion in the TT-format, SIAM J. Sci. Comput. (ISSN: 1064-8275) 34 (5) (2012) A2718–A2739, http://dx.doi.org/10.1137/110833142.

[7] S.V. Dolgov, TT-GMRES: solution to a linear system in the structured tensor format, Russian J. Numer. Anal. Math. Modelling (ISSN: 0927-6467) 28 (2) (2013) 149–172, http://dx.doi.org/10.1515/rnam-2013-0009.

[8] A.O.H. Axelsson, J.G. Verwer, Boundary value techniques for initial value problems in ordinary differential equations, Math. Comp. (ISSN: 0025-5718) 45 (171) (1985) 153–171, S1–S4, http://dx.doi.org/10.2307/2008054.

[9] D. Bertaccini, M.K. Ng, Block $\{\omega\}$-circulant preconditioners for the systems of differential equations, Calcolo (ISSN: 0008-0624) 40 (2) (2003) 71–90.

[10] L. Brugnano, D. Trigiante, Boundary value methods: the third way between linear multistep and Runge-Kutta methods, Comput. Math. Appl. (ISSN: 0898-1221) 36 (10–12) (1998) 269–284, http://dx.doi.org/10.1016/S0898-1221(98)80028-X, Advances in difference equations, II.

[11] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. (ISSN: 0036-1445) 51 (3) (2009) 455–500, http://dx.doi.org/10.1137/07070111X.

[12] I.V. Oseledets, E.E. Tyrtyshnikov, Breaking the curse of dimensionality, or how to use SVD in many dimensions, SIAM J. Sci. Comput. (ISSN: 1064-8275) 31 (5) (2009) 3744–3759, http://dx.doi.org/10.1137/090748330.

[13] D. Bertaccini, F. Durastante, Solving mixed classical and fractional partial differential equations using short-memory principle and approximate inverses, Numer. Algorithms (ISSN: 1017-1398) 74 (4) (2017) 1061–1082, http://dx.doi.org/10.1007/s11075-016-0186-8.

[14] M. Donatelli, M. Mazza, S. Serra-Capizzano, Spectral analysis and multigrid methods for finite volume approximations of space-fractional diffusion equations, SIAM J. Sci. Comput. (ISSN: 1064-8275) 40 (6) (2018) A4007–A4039, http://dx.doi.org/10.1137/17M115164X.

[15] I. Oseledets, E. Tyrtyshnikov, N. Zamarashkin, Tensor–train ranks for matrices and their inverses, Comput. Methods Appl. Math. (ISSN: 1609-4840) 11 (3) (2011) 394–403.

[16] S.V. Dolgov, D.V. Savostyanov, Alternating minimal energy methods for linear systems in higher dimensions, SIAM J. Sci. Comput. (ISSN: 1064-8275) 36 (5) (2014) A2248–A2271, http://dx.doi.org/10.1137/140953289.

[17] B.N. Khoromskij, $O(d\log N)$-quantics approximation of N-d tensors in high-dimensional numerical modeling, Constr. Approx. (ISSN: 0176-4276) 34 (2) (2011) 257–280, http://dx.doi.org/10.1007/s00365-011-9131-1.