NESUS
cost IC1305
Network for Sustainable Ultrascale Computing

Proceedings of the Third International Workshop on Sustainable Ultrascale Computing Systems (NESUS 2016) Sofia, Bulgaria

Jesus Carretero, Javier Garcia Blas, Svetozar Margenov
(Editors)

October, 6-7, 2016

Ciegis, R., Starikovicius, V., & Margenov, S. (2016). On Parallel Numerical Algorithms for Fractional Diffusion Problems. En *Proceedings of the Third International Workshop on Sustainable Ultrascale Computing Systems* (NESUS 2016) Sofia, Bulgaria (pp. 85-90). Madrid: Universidad Carlos III de Madrid. Computer Architecture, Communications, and Systems Group (ARCOS)

On Parallel Numerical Algorithms for Fractional Diffusion Problems

Raimondas Čiegis[†], Vadimas Starikovičius[†], Svetozar Margenov[*]
[†]Vilnius Gediminas Technical University, Lithuania
raimondas.ciegis@vgtu.lt,
vadimas.starikovicius@vgtu.lt
[*]IICT-BAS, Bulgaria
margenov@parallel.bas.bg

Abstract

In this work, we consider a parallel numerical solution of problems depending on fractional powers of elliptic operator. Three different state of the art approaches are used to transform the original non-local problem into well-known local PDE problems. Parallel numerical algorithms for all three approaches are developed and discussed. Results of their parallel performance tests are presented and analysed.

Keywords Fractional Diffusion, Parallel Numerical Algorithms, Multigrid, Strong Scalability

I. Introduction

The permanent development of the large scale computational systems with all their diversity requires a constant attention, which must be paid to a development and selection of proper parallel algorithms for the solution of various problems. In this paper, we consider a parallel numerical solution of problems involving fractional powers of elliptic operators. Such problems arise in a wide range of areas, including image processing, porous media flow, material sciences (see, e.g., [1] and the references therein.

In this paper, we investigate the scalability and efficiency of parallelization of three state of the art discrete algorithms used for numerical solution of fractional power elliptic problems. These algorithms are reducing the given non-local diffusion problem to some local classical differential problems formulated in spaces of higher dimension. It is important to note, that despite using a common embedding technique, these three approaches lead to very different challenges in construction of efficient parallel algorithms.

The rest of this paper is organized as follows. In Section II, we describe the problem under consideration with fractional power of elliptic diffusion operator. In Section III, three partial differential equations (PDEs) models are formulated and applied to construct efficient numerical solution techniques for considered problems. They are based on different types of PDEs and all of them define local operators but embedded into higher dimension space. The finite volume method is used to approximate the formulated PDEs by the discrete schemes. The parallelization issues of these three numerical solution algorithms are discussed in Section IV. Parallel performance results of the developed parallel algorithms are presented and analysed. The strong scalability is considered. Some final conclusions are given in Section V.

II. Problem Formulation

Let Ω be a bounded domain in $\mathbb{R}^n$, $n \geq 2$ with boundary $\partial\Omega$. Given a function f, we seek u such that

$$L^{\beta} u = f, \quad X \in \Omega \tag{1}$$

with some boundary conditions on $\partial\Omega$, $0 < \beta < 1$ and

$$Lu = -\sum_{j=1}^{n} \frac{\partial}{\partial x_j}\left(k(X)\frac{\partial u}{\partial x_j}\right).$$

Let us denote by $\{\phi_k\}$, $k = 1, 2, \ldots, N$ the orthonormal basis (for convenience, here we restrict to the case of finite number of modes typical for discrete approximations)

$$L\phi_k = \lambda_k \phi_k.$$

Then the fractional powers of the diffusion operator are defined by

$$L^\beta u = \sum_{k=1}^{N} \lambda_k^\beta w_k \phi_k, \tag{2}$$

where $w_k = (u, \phi_k)$.

Note, that the direct implementation of this approach is very expensive. It requires the computation of all eigenvectors and eigenvalues of large matrices. This algorithm can be used for practical computations if the fractional power of Laplace operator is solved in rectangular domain, when FFT techniques can be applied.

III. PDE approach for the fractional non-local model

In this section we formulate three PDE models to approximate problems involving fractional powers of elliptic operators. These approximations allow us to construct efficient solution techniques for the original problem. The formulated PDEs are approximated by the finite volume schemes.

III.1 Extension to the mixed boundary value problem in the semi-infinite cylinder $C = \Omega \times (0, \infty) \subset \mathbb{R}^{n+1}$

Non-local problem (1) is equivalent to the following classical local linear problem in the extended space $\mathbb{R}^{n+1}$ [2, 3]:

$$-\frac{\partial}{\partial y}\left(y^\alpha \frac{\partial V}{\partial y}\right) + y^\alpha L V = 0, \quad (X, y) \in C,\ \alpha = 1 - 2\beta, \tag{3}$$

$$-y^\alpha \frac{\partial V}{\partial y} = d_\beta f, \quad X \in \bar{\Omega} \times \{0\},$$
$$V = 0, \quad (X, y) \in C_B = \partial C \setminus \bar{\Omega} \times \{0\},$$

where d_β is a positive normalization constant that depends only on β. Then $u(X) = V(X, 0)$.

In order to construct a finite volume approximation of (3), the semi-infinite cylinder is approximated by the truncated cylinder $C_Y = \Omega \times \{0, Y\}$ with a sufficiently large Y. A uniform mesh Ω_h is introduced in Ω and anisotropic mesh $\omega_h = \{y_j = (j/M)^\gamma Y, j = 0, \dots, M\}$ is used to compensate the singular behaviour of the solution as $y \to 0$, where $\gamma > 3/(2\beta)$ [2, 3].

By using the finite volume method and standard notations of the finite differences we define the discrete problem, which approximates (3):

$$-\left(y^\alpha_{j+1/2}\frac{V_{h,j+1} - V_{h,j}}{H_{j+1/2}} - y^\alpha_{j-1/2}\frac{V_{h,j} - V_{h,j-1}}{H_{j-1/2}}\right) + \frac{y^{\alpha+1}_{j+1/2} - y^{\alpha+1}_{j-1/2}}{\alpha+1} L_h V_h = 0, \quad (X_h, y_j) \in C_{Y_h}, \tag{4}$$

$$-y^\alpha_{1/2}\frac{V_{h,1} - V_{h,0}}{H_{1/2}} + \frac{y^{\alpha+1}_{1/2}}{\alpha+1} L_h V_h = d_\beta f_h,$$
$$X_h \in \bar{\Omega}_h \times \{0\},$$
$$V_h = 0, \quad (X_h, y_j) \in \partial C_{Yh} \setminus \bar{\Omega}_h \times \{0\},$$

where

$$L_h V_h = -\sum_{k=1}^{n} \partial_{x_k}\big(k(X_h)\partial_{\bar{x}_k} V_h\big),$$
$$y_{j+1/2} = \frac{y_j + y_{j+1}}{2},\ H_{j+1/2} = y_{j+1} - y_j.$$

III.2 Integral representation of the solution of initial problem (1)

The algorithm is based on the integral representation of the non-local operator using the classical local operators [4]:

$$L^{-\beta} = \frac{2\sin(\pi\beta)}{\pi}\Big[\int_0^1 y^{2\beta-1}(I + y^2 L)^{-1} dy + \int_0^1 y^{1-2\beta}(y^2 I + L)^{-1} dy\Big]. \tag{5}$$

Different quadrature schemes can be used to approximate these singular integrals. In this paper, we have applied a graded partition of integration interval $[0, 1]$ to resolve the singular behaviour of $y^{2\beta-1}$:

$$y_{1,j} = \begin{cases} (j/M)^{\frac{1}{2\beta}} & \text{if } 2\beta - 1 < 0, \\ j/M & \text{if } 2\beta - 1 \ge 0, \end{cases}, \quad j = 0, \dots, M.$$

A similar partition is used to resolve the singularity of $y^{1-2\beta}$. Then the following approximation of integrals (5) is applied

$$L_h^{-\beta} f_h = \frac{2\sin(\pi\beta)}{\pi} \times \Bigg[\sum_{j=1}^{M} \frac{y_{1,j}^{2\beta} - y_{1,j-1}^{2\beta}}{2\beta} \big(I_h + y_{1,j-1/2}^2 L_h\big)^{-1} f_h + \sum_{j=1}^{M} \frac{y_{2,j}^{2-2\beta} - y_{2,j-1}^{2-2\beta}}{2-2\beta} \big(y_{2,j-1/2}^2 I_h + L_h\big)^{-1} f_h \Bigg]. \quad (6)$$

One or two level parallelization strategies can be applied to solve the multiple independent local linear sub-problems $(I_h + y_j^2 L_h)^{-1} f$ and $(y_j^2 I_h + L_h)^{-1} f$.

III.3 Reduction to a pseudo-parabolic PDE problem

The solution of non-local problem (1) is sought as a mapping [5]:

$$V(X,t) = \big(t(L-\delta I) + \delta I\big)^{-\beta} f,$$

where $L \geq \delta_0 I$, $\delta = \gamma\delta_0$, $0 < \gamma < 1$.

Thus it follows that $V(X,1) = L^{-\beta} f$. The function V satisfies the evolutionary pseudo-parabolic problem

$$(tG + \delta I)\frac{\partial V}{\partial t} + \beta G V = 0, \quad 0 < t \leq 1, \qquad (7)$$
$$V(0) = \delta^{-\beta} f, \quad t = 0,$$

where $G = L - \delta I$.

Again, instead of the non-local problem (1) we solve a non-stationary local pseudo-parabolic problem (formally in $\mathbb{R}^{n+1}$ space). In order to solve (7), we use the following finite volume scheme [6]:

$$(t^{n-1/2} G_h + \delta I_h)\frac{V_h^n - V_h^{n-1}}{\tau} + \beta G_h V_h^{n-1/2} = 0, \quad (8)$$
$$0 < n \leq M,$$
$$V_h^0 = \delta^{-\beta} f_h,$$

where $G_h = L_h - \delta I_h$, $V_h^{n-1/2} = (V_h^n + V_h^{n-1})/2$ and $t^{n-1/2} = (t^{n-1} + t^n)/2$.

IV. Parallel algorithms

In this section we are considering and discussing the parallelization of all three numerical solution algorithms presented in Section III. Our analysis is restricted to the strong scalability, when the size of discrete problems is fixed and different numbers of processors are used in the computations. Such an information is very important when a medium size problem should be solved as fast as possible (consider optimization algorithms when computation of the value of the objective function reduces to numerical solution of fractional power of elliptic problem).

All parallel numerical tests in this work were performed on the computer cluster "HPC Sauletekis" (`http://www.supercomputing.ff.vu.lt`) at the High Performance Computing Centre of Vilnius University, Faculty of Physics. We have used up to 10 nodes with Intel® Xeon® processors E5-2670 with 16 cores (2.60 GHz) and 128 GB of RAM per node. Computational nodes are interconnected via the InfiniBand network.

IV.1 Discrete elliptic problem

The approximate PDE model (3) transforms the non-local fractional diffusion problem (1) into well-studied case of PDEs problems with elliptic operators. The selected finite volume scheme (4) means that our first numerical algorithm essentially deals with a solution of one large system of linear equations. In case, when the problem domain Ω is two-dimensional, one needs to solve a system with 7 point stencil of size $N = N_{x_1} \times N_{x_2} \times M$.

A standard approach for the parallel solution of such problems is the domain decomposition method [7]. The discrete mesh of the problem domain and its associated fields are partitioned into sub-domains, which are allocated to different processes. Note that in our case, the discrete mesh C_{Y_h} of the truncated cylinder $C_Y = \Omega \times \{0, Y\}$ needs to be partitioned. In this work, we use a simple one-dimensional partitioning in y direction.

It is well known that the parallel performance of PDE problem solver essentially depends on the quality of the parallel linear solver. In this work, we have used the parallel multigrid solver from AGMG package [8, 9].

To test the parallel performance of the developed algorithm, we have considered the problem (3) in the 2D unit square domain Ω using the discrete mesh of the size $N_{x_1} = N_{x_2} = 1000$ and $M = 250$. The tolerance of multigrid solver was set to 10^{-6} in all tests. Obviously, the computational complexity of this problem also depends on the fractional power β. The available numerical tests in the literature mostly concern the cases $\beta \in \{0.25, 0.5, 0.75\}$. In this article, we present numerical tests only for the most complicated case $\beta = 0.75$. Here we restrict to the analysis of 1D domain decomposition and the y coordinate is divided into M/p size blocs and distributed among p processes. From a scalability analysis it is known that for the larger problems and larger number of processors the 2D and 3D partitionings are more efficient decomposition strategies and this topic will be investigated in a separate paper.

Parallel performance results are presented in Table 1. The total wall time T_p is given in seconds. Here $p = n_d \times n_c$ is the number of used parallel processes computing with n_d nodes and n_c cores per node. In Table 1, we present the obtained values of parallel algorithmic speed-up $S_p = T_1/T_p$ and efficiency $E_p = S_p/p$.

p	1=1x1	2=1x2	4=1x4	8=1x8
T_p	1020	575.6	308.6	170.4
S_p	1	1.77	3.31	5.99
E_p	1	0.89	0.83	0.75
p	16=1x16	32=2x16	32=8x4	48=3x16
T_p	127.4	94.3	75.6	158.8
S_p	8.01	10.82	13.50	6.43
E_p	0.50	0.34	0.42	0.13

Table 1: The total wall time T_p, speed-up S_p and efficiency E_p solving problem (3) with $N_{x_1} = N_{x_2} = 1000$, $M = 250$, $\beta = 0.75$.

The obtained speed-up and efficiency values are not very good. The efficiency of the parallel algorithm is much better when a weak scalability analysis is done and the size of the discrete problem is increased proportionally to the increased number of processes. However, the presented results of strong scalability analysis show potential drawbacks of the first approach for the parallel solution with a larger number of processors.

IV.2 Integral evaluation problem

Using the second approach described in Section III.2, the non-local fractional diffusion problem (1) is transformed into a computation of two integrals (5). Each term in both sums of numerical approximation (6) can be computed independently, what is very convenient for the parallelization.

In our second parallel solver, we employ the well-known Master-Slave parallel model [10, 11]. Master process generates and distributes tasks (a block of consecutive y_j values) between the slave processes. For each received y_j value a slave process solves the local elliptic problem $(I_h + y_j^2 L_h)^{-1} f$ or $(y_j^2 I_h + L_h)^{-1} f$ in domain Ω.

Differently from the usual Master-Slave model, in our solver, slave processes do not return to the master results of each task immediately after its solution. The slave processes accumulate the obtained results - compute partial sums of the solution u for each mesh point. These big data vectors of the size $N_{x_1} \times N_{x_2}$ are sent only once, after the solution of the last task. The problem solution u is collected from the partial sums at the master process by MPI reduction operation [12].

To test the parallel performance of the developed algorithm, we have considered the problem (5) in the 2D unit square domain Ω using the discrete mesh of the size $N_{x_1} = N_{x_2} = 1000$ and $M = 3000$ in (6). A single task was defined as a block of 10 consecutive y_j values. For the local elliptic problems the tolerance of multigrid solver was set to 10^{-6}. The fractional power β was set to 0.75.

Parallel performance results of our second parallel solver are presented in Table 2. The total wall time $T_{s,n_d \times n_c}$ is given in seconds. Here $p = n_d \times n_c$ is the total number of used parallel processes computing with n_d nodes and n_c cores per node, $s = p - 1$ is the number of slave processes, which are solving computational tasks. In Table 2, we also present the obtained values of parallel algorithmic speed-up $S_s = T_1/T_{s,n_d \times n_c}$ and efficiency $E_s = S_s/s$.

	1, 1x2	2, 1x3	4, 1x5	8, 1x9	15, 1x16
T_s	11862	6192	3098	1605	1047
S_s	1	1.92	3.83	7.39	11.33
E_s	1	0.96	0.96	0.92	0.76
	31, 2x16	47, 3x16	63, 4x16	127, 8x16	159, 10x16
T_s	521.5	354.0	268.0	140.2	113.6
S_s	22.75	33.51	44.26	84.6	104.4
E_s	0.73	0.71	0.70	0.67	0.66

Table 2: The total wall time $T_{s,n_d \times n_c}$, speed-up S_s and efficiency E_s solving problem (5) with $N_{x_1} = N_{x_2} = 1000$, $M = 3000$, block size - 10, $\beta = 0.25$.

A slight degradation of the performance of our second parallel solver is caused by the load imbalance of the slave processes. The computational complexity of the local elliptic problems is different for the different y_j values. The number of tasks assigned to the single slave process is decreasing as the number of processes increases. This causes an increasing influence of the load imbalance on the total solution time.

The reduction of the single task (i.e. y_j block size) should reduce this drawback. However, this will cause an increasing communication between the master and slave processes. At some point, this can cause an idling of slave processes, waiting for the tasks from busy master.

IV.3 Discrete pseudo-parabolic problem

Using the third approach described in Section III.3, the non-local fractional diffusion problem (1) is transformed into another well-studied case of pseudo-parabolic PDE problem (7).

The constructed finite volume scheme (8) implies that our third numerical algorithm will advance in pseudo-time solving one system of linear equations at each of M iterations. In case, when the problem domain Ω is two-dimensional, the linear system will have 5 point stencil matrix of size $N = N_{x_1} \times N_{x_2}$.

One can easily see the similarities and differences with the first approach. One of the important practical implications is the significantly smaller amount of memory required to fit the system matrix, solution, and other data.

Again, a standard domain decomposition method is used for the parallel solution of pseudo-parabolic PDE problem. The discrete mesh of problem domain Ω and its associated fields are partitioned into subdomains, which are allocated to different processes. As in the previous tests, a simple one-dimensional block partitioning is used.

To test the parallel performance of the developed algorithm, we have considered the problem (7) in the 2D unit square domain Ω using the discrete mesh of the size $N_{x_1} = N_{x_2} = 1000$ and $M = 1000$. The tolerance of AGMG multigrid solver was set to 10^{-6} in all tests. Obviously, the computational complexity of problem (7) also depends on the fractional power β and parameter δ. In this case, we have performed numerical tests for $\beta = 0.25$ and $\delta = 10$.

Parallel performance results are presented in Table 3. The total wall time T_p is given in seconds. Here $p = n_d \times n_c$ is the number of used parallel processes computing with n_d nodes and n_c cores per node. In Table 3, we present the obtained values of parallel algorithmic speed-up $S_p = T_1/T_p$ and efficiency $E_p = S_p/p$.

p	1=1x1	2=1x2	4=1x4	8=1x8
T_p	2481.1	1562.7	813.6	421.7
S_p	1	1.59	3.05	5.88
E_p	1	0.79	0.76	0.74
p	16=1x16	32=2x16	32=8x4	48=3x16
T_p	320.9	376.6	345.3	610.3
S_p	7.73	6.59	7.18	4.07
E_p	0.48	0.21	0.22	0.08

Table 3: The total wall time T_p, speed-up S_p and efficiency E_p solving problem (7) with $N_{x_1} = N_{x_2} = 1000$, $M = 1000$, $\beta = 0.25$, $\delta = 10$.

Again, as it was with the first solver, the obtained speed-up and efficiency values are not very good. Since the size of 2D problem is even smaller, in this case the parallel scalability of AGMG multigrid solver is even more critical, than in the case of the first solver.

V. Conclusions

Three different parallel numerical algorithms were developed for fractional diffusion problems. All of them rely on transformations of the original non-local problem to well-known local PDE problems.

The advantage of this approach is that due to the common use of these PDEs models their numerical solution methods are well developed. The software packages for their numerical solution (including parallel) are subject to a long-time development and permanent improvements.

The first and third algorithms strongly depend on the parallel scalability of the available multigrid solvers. The third algorithm has the significantly smaller demand on the amount of the required memory compared to the first one.

The performance results of second parallel algorithm are very promising. The issue of load balancing needs a special attention and further research. Possibility of employing a multilevel parallelism makes this approach even more attractive.

The weak scalability of these parallel algorithms will be studied in a following paper.

Acknowledgment

The work presented in this paper has been partially supported by EU under the COST programme Action IC1305, 'Network for Sustainable Ultrascale Computing (NESUS)'. This research was also funded by a grant (No. MIP-074/2015) from the Research Council of Lithuania.

Computations were performed on resources at the High Performance Computing Centre "HPC Sauletekis" in Vilnius University Faculty of Physics.

References

[1] S. Harizanov, S. Margenov, P. Marinov and Y. Vutov, "Volume constrained 2-phase segmentation method utilizing a linear system solver based on the best uniform polynomial approximation of $x^{1/2}$ ", *Journal of Computational and Applied Mathematics*, vol. 310, no. 1, 115–28, 2017.

[2] R. H. Nochetto, E. Otarola, and A. J. Salgado, "A PDE approach to fractional diffusion in general domains: a priori error analysis", *Foundations of Computational mathematics*, vol. 15, no. 3, pp. 733-791, 2015.

[3] R. H. Nochetto, E. Otarola, and A. J. Salgado, "A PDE approach to numerical fractional diffusion", in *Proceedings of the 8th ICIAM*, Beijing, China, 2015, pp. 211-236.

[4] A. Bonitto and J. E. Pasciak, "Numerical approximation of fractional powers of elliptic operators", *Math. Comp.*, vol. 84, pp. 2083-2110, 2015.

[5] P. N. Vabishchevich, "Numerical solving unsteady space-fractional problems with the square root of an elliptic operator", *Math. Model. and Anal.*, vol. 21, no.2, pp. 220-238, 2016.

[6] R. Čiegis and N. Tumanova, "Stability analysis of finite difference schemes for pseudo-parabolic problems with non-local boundary conditions", *Math. Model. and Anal.*, vol. 19, no.2, pp. 285-297, 2014.

[7] A. Quarteroni and A. Valli, *A Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, 1999.

[8] Y. Notay, "AGMG software and documentation", `http://homepages.ulb.ac.be/~ynotay/AGMG`. Last access: September 2016.

[9] A. Napov and Y. Notay, "An algebraic multigrid method with guaranteed convergence rate", *SIAM J. Sci. Comput.*, vol. 34, pp. 1079-1109, 2012.

[10] V. Kumar, A. Grama, A. Gupta, and G. Karypis, *Introduction to Parallel Computing: Design and Analysis of Algorithms*, Benjamin/Cummings, 1994.

[11] R. Čiegis, V. Starikovičius, N. Tumanova, and M. Ragulskis, "Application of distributed parallel computing for dynamic visual cryptography", *Journal of Supercomputing*, doi:10.1007/s11227-016-1733-8.

[12] MPI Forum, "MPI: A Message Passing Interface Standard", `www.mpi-forum.org`, 1995.