Performance Evaluation of Space Fractional FitzHugh-Nagumo Model: an Implementation with PETSc Library

Nikita Markov[1], Konstantin Ushenin[1,2], and Ahmed Hendy[1]

[1] Ural Federal University, Yekaterinburg, Russia
[2] Institute of Immunology and Physiology of the Ural Branch of the Russian Academy of Sciences, Yekaterinburg, Russia

Abstract. Space fractional derivatives have been used instead of integer operators with success in a variety of practical applications to study and describe transport processes in media characterized by spatial connectivity properties and high structural heterogeneity altering the classical laws of diffusion. This study provides an implementation of space fractional FitzHugh-Nagumo model of excitable medium in two-dimensional space. This implementation is based on shifted Grunwald-Letnikov computational scheme which transforms the space fractional diffusion equations to a system of algebraic equations. After that, the conjugated gradient method is used to solve the sparse matrix system. MPI and PETSc library is used for implementation. We investigate the influence of some factors to our implementation performance and scalability such as core numbers, size of computation mesh, and fractional derivative orders. All tests are done on ccNUMA architecture with two CPUs.

Keywords: space fractional reaction-diffusion equations · fractional Laplacian · FitzHugh-Nagumo model · spiral waves · conjugated gradient method · distributed memory system · PETSc library

1 Introduction

Living system simulations make great impact on the understanding of biological processes. They help in checking the side effects of drugs and allow to create a personalized model of human organs which help in planning of surgery and medication. Recently, some researchers have been looking for a way to apply fractional derivatives [1] to living system models. The main advantage of these approaches is to describe sub-diffusion processes. The sub-diffusion is a diffusion process which gives a squared displacement of particles which are non-linearly dependent on time. Especially, this variant of diffusion exists in tissue and subcellular space. The model of visco-elastic property of soft tissue which can explain oscillation damping more precisely is one of models which contains fractional derivatives. The review of these sorts of models are described in the introduction of Ph.D. thesis [2]. In addition, there are several other applications of models with fractional derivatives to life science: cancer growth [3], initial stage of HIV [4], and population growth [5].

Individual attention is required for the excitable medium models. These models describe neurons and muscle tissue (myocardium). The special problem is a more realistic description of diffusion in consideration of subdiffusion transport in subcellural space and tissue structure. In [6], the authors proposed a replacment of integer Laplacian in excitable medium model by fractional Laplacian. This replacement gives a precise reproduction of several physiological experiments in computer simulations. An important feature of fractional derivative in a new model behavior, which can not be repeated with changing diffusion coefficient or introduction of nonlinear parts.

Unfortunately, the replacement of integer derivatives by fractional ones leads to a significant increase in the computational time. This fact motivates us to introduce a high performance implementation for fractional FitzHugh-Nagumo model [7] based on some computational methods [8]. A fractional FitzHughNagumo model is a generalizing of the standard model that describes the propagation of the electrical potential in heterogeneous cardiac tissue. The fractional model consists of a couple fractional Riesz space nonlinear reaction-diffusion model and a system of ordinary differential equations. This implementation is required for physiological experiments in the future.

We use PETSc library [9–11], which provides iterative solvers for sparse matrix. This library is used in more than 14 software packages for scientific computing [9] and support multicore architectures, computation accelerators and systems with distributed memory. The implementation based on this popular and well-known library is more suitable for our goals.

Besides that, we investigate the influence of some factors in our implementation performance and scalability such as core numbers, size of computation mesh, and fractional derivative orders. All tests were conducted on Cache coherent non-uniform memory access (ccNUMA) architecture with two six core central processing unit (CPU). The rest of this paper is as follows: the next section is devoted to describe FitzHugh-Nagumo model and fractional Laplalacian operator, then we proceed to describe conducted experiments.

2 Numerical Implementation of FitzHugh-Nagumo Model

The FitzHugh-Nagumo model is a reaction-diffusion equation which describes waves propagation in an excitable medium. Initially, it was created as a mathematical description of the action potential propagation on an axon membrane.

$$\frac{\partial u}{\partial t} = \nabla \cdot (K \nabla u) + u(1-u)(u-a) - v, \tag{1}$$

$$\frac{\partial v}{\partial t} = \varepsilon(\beta u - \gamma v - \sigma), \tag{2}$$

where u is a “fast” variable which describes membrane potential of a cell; v is a “slow” variable which connects with the medium conductivity by inverse ratio;

K is a diffusion tensor; ε, β, γ, σ are the constants which describe the model behavior.

During a long period of time, the FitzHugh-Nagumo model was being used for research of auto-oscillatory process in the excitable media. For example, the spiral waves in a two dimensional domain and scroll waves in a three dimensional domain. These structures are models of arrhythmic activity in mammalian hearts.

The model is suitable for this study, because it is simple, contains only two phase variables and converges on big time steps. In addition, spiral waves provide test example for a performance analysis which is close to realistic problems.

2.1 Space Fractional Laplacian Operators

Usually, researchers and engineers use derivatives and integrals with integer order of degree in applications. However, there are several definitions of derivatives and integrals with real and complex numbers of degree. They are named fractional derivatives and fractional integrals. Some of these definitions are given by Grunwald-Letnikov, Riemann-Liouville, Caputo, Marchaud, Hadamard, and more [12]. In some cases, many of these definitions are equivalent to each other.

In this paper we use a fractional derivative with order α_1 $(1 < \alpha_1 \leq 2)$ which is defined by Riesz [12].

$$\frac{\partial^{\alpha_1} v}{\partial |x|^{\alpha_1}} = -\frac{1}{2\cos(\pi\alpha_1/2)}\Big(\frac{\partial^{\alpha_1} v}{\partial x^{\alpha_1}} + \frac{\partial^{\alpha_1} v}{\partial (-x)^{\alpha_1}}\Big), \tag{3}$$

where

$$\frac{\partial^{\alpha_1} v(x,y,t)}{\partial x^{\alpha_1}} = \frac{1}{\Gamma(2-\alpha_1)}\frac{\partial^2}{\partial x^2}\int_{-\infty}^{x} \frac{v(\zeta,y,t)}{(x-\zeta)^{\alpha_1-1}}d\zeta, \tag{4}$$

$$\frac{\partial^{\alpha_1} v(x,y,t)}{\partial (-x)^{\alpha_1}} = \frac{1}{\Gamma(2-\alpha_1)}\frac{\partial^2}{\partial x^2}\int_{x}^{\infty} \frac{v(\zeta,y,t)}{(\zeta-x)^{\alpha_1-1}}d\zeta. \tag{5}$$

Similar expressions are defined for the space Riesz fractional derivative of order α_2 $(1 < \alpha_2 \leq 2)$ with respect to y.

After that, we can introduce a fractional Laplacian operator.

$$\nabla^{\beta} = \Big(\frac{\partial^{\beta}}{\partial |x|^{\beta}}, \frac{\partial^{\beta}}{\partial |y|^{\beta}}, \frac{\partial^{\beta}}{\partial |z|^{\beta}}\Big)^T \tag{6}$$

In [1, 13, 12], the authors describe the theory of fractional derivatives in detail.

After the replacement of integer Laplacian operator with the fractional one, the fractional FitzHugh-Nagumo model appears:

$$\frac{\partial u}{\partial t} = K_x \frac{\partial^{\alpha_1} u}{\partial |x|^{\alpha_1}} + K_y \frac{\partial^{\alpha_2} u}{\partial |y|^{\alpha_2}} + u(1-u)(u-a) - v, \tag{7}$$

$$\frac{\partial v}{\partial t} = \varepsilon(\beta u - \gamma v - \sigma), \tag{8}$$

where K_x and K_y are conductivity coefficients along axes. For further explanation, we denote non-linear part of first equation as $f = u(1-u)(u-a) - v$, and left side of second equations as $s = \varepsilon(\beta u - \gamma v - \sigma)$.

2.2 Numerical Method

In our research, we use the numerical method which is described in [14, 8]. This numerical method transforms (7) to a system of linear algebraic equations which can be solved by conjugated gradient (CG) method.

The numerical simulation is formulated by considering positive integers m_1, m_2 and N such that $h_x = D_x/m_1, \quad h_y = D_y/m_2, \quad \tau = T/N$ be characteristics of space and time grids. The orthogonal uniform computation mesh includes $m \times m$ nodes with step by space h. τ is a time step and n is a number of conducted time steps. $u^n_{i,j}$ is the value of unknown function $u(x_i, y_j, t_n)$.

The GrunwaldLetnikov approximations for l.h.s and r.h.s fractional derivatives are given by:

$$\frac{\partial^\alpha u}{\partial x^\alpha}\Big|_{(x_i,y_i,t_n)} = \frac{1}{(h_x)^a}\sum_{l=0}^{i+1} g_\alpha^{(l)} u(x_{i-l+1}, y_j, t_n) + O(h_x), \tag{9}$$

$$\frac{\partial^\alpha u}{\partial (-x)^\alpha}\Big|_{(x_i,y_i,t_n)} = \frac{1}{(h_x)^a}\sum_{l=0}^{m-i+1} g_\alpha^{(l)} u(x_{i+l-1}, y_j, t_n) + O(h_x), \tag{10}$$

where coefficients g^l_α are defined by recurrent formula:

$$g_\alpha^{(0)} = 1, \quad g_\alpha^{(l)} = (-1)^l \binom{\alpha}{l} = -g_\alpha^{l-1}\frac{\alpha - l + 1}{l}. \tag{11}$$

Similar expressions are defined for y axis direction. The Order of fractional derivative for y direction is denoted as α_2.

Substitution of expressions (9), (10) in (7) leads to the following numerical scheme [14]:

$$\begin{aligned} u^n_{i,j} &+ r^{(1)}\Big[\sum_{l=0}^{i+1} g_{\alpha_1}^{(l)} u^n_{i-l+1,j} + \sum_{l=0}^{m-i+1} g_{\alpha_1}^{(l)} u^n_{1+l-1,j}\Big] \\ &+ r^{(2)}\Big[\sum_{l=0}^{j+1} g_{\alpha_2}^{(l)} u^n_{i,j-l+1} + \sum_{l=0}^{m-j+1} g_{\alpha_2}^{(l)} u^n_{i,j+l-1}\Big] \quad = u^{n-1}_{i,j} + \tau f^{n-1}_{i,j}, \end{aligned} \tag{12}$$

$$v^n_{i,j} = v^{n-1}_{i,j} + \tau s^{n-1}_{i,j}, \tag{13}$$

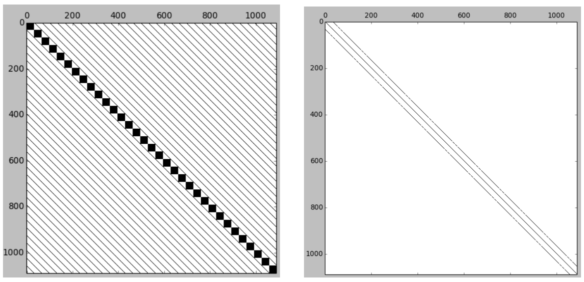
$$r^{(1)} = \frac{\tau K_x}{2\cos(\frac{\pi\alpha_1}{2})h^{\alpha_1}}, \; r^{(2)} = \frac{\tau K_y}{2\cos(\frac{\pi\alpha_2}{2})h^{\alpha_2}}, \tag{14}$$

where $f^n_{i,j} = u^n_{i,j}(1 - u^n_{i,j})(u^n_{i,j} - a) - v^n_{i,j}$ is nonlinear part in the first model equation, and $s^n_{i,j} = \varepsilon(\beta u^n_{i,j} - \gamma v^n_{i,j} - \sigma)$ is right hand part of second equations.

Boundary conditions in Dirichlet form are described in the following form:

$$u^n_{0,j} = u^n_{m,j} = u^n_{i,0} = u^n_{i,m} = 0. \tag{15}$$

The linear system $Au^{(n)} = b^{(n-1)}$ is constructed based on these expressions. A is matrix of the system with size $(m-1)(m-1) \times (m-1)(m-1)$ in left hand



(a) $\alpha_1 = \alpha_2 = 1.9$ (b) $\alpha_1 = \alpha_2 = 2.0$

Fig. 1: Structure of the matrix A for different values α_1, α_2

side of equations (12). $u^n = (u^n_{1,1}, u^n_{1,2}, ..., u^n_{1,m-1}, u^n_{2,1}, ..., u^n_{m-1,m-1})$. $b^{(n-1)}$ is a right hand side vector of system (12) with length $(m-1)(m-1)$.

Figure 1 presents matrix A with different values of $\alpha_1 = \alpha_2$. The problem became identical to usual FitzHugh-Nagumo model if $\alpha_1 = \alpha_2 = 2.0$.

In work [14] authors use Gauss-Seidel method to solve the equations. In our research we use conjugated gradient method without precondition.

2.3 Spiral Wave Initiation Protocol

In this research, we use initial conditions which lead to a stable spiral wave in excitable medium. These structures are models of arrhythmic activity in mammalian hearts. Spiral waves are auto oscilatory process. This allows to use them for long time simulations without additional activation of excitable medium.

Initial conditions correspond to ones in [14]. These are square with $u^0_{i,j} = 1$ and rectangle with $v^0_{i,j} = 0.1$ which are not intersect with each other.

For better understanding, we rewrite expressions from [14] in terms of percent from square side:

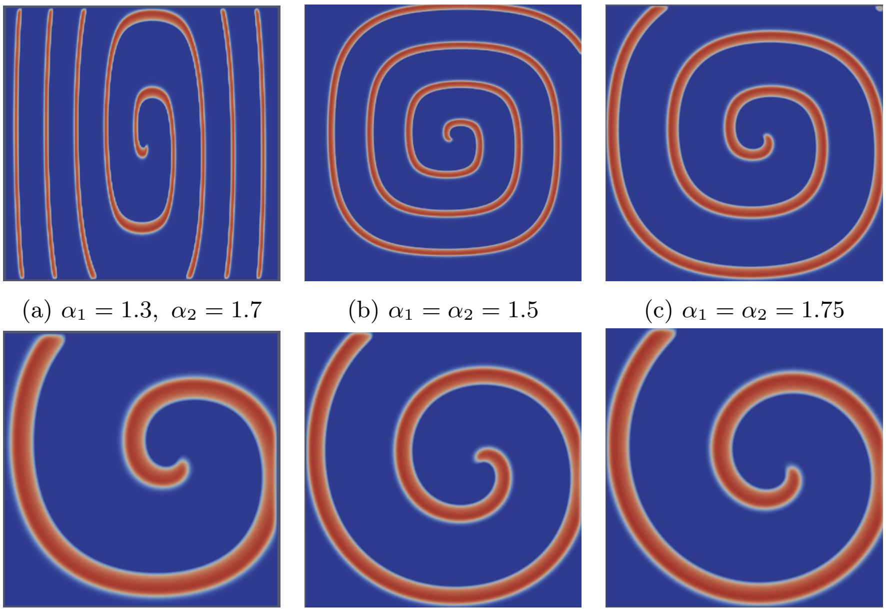
$$u(x, y, 0) = \begin{cases} 1.0, & \text{if } 0\% < x \leq 50\%,\ 0\% < y < 50\%, \\ 0.0, & \text{otherwise} \end{cases} \tag{16}$$

$$v(x, y, 0) = \begin{cases} 0.1, & \text{if } 0\% < x \leq 50\%,\ 50\% < y < 100\%, \\ 0.0, & \text{otherwise} \end{cases} \tag{17}$$

Spiral waves for different fractional derivative orders are shown in Figure 2.

2.4 Implementation

At the first stage of the algorithm, arrays and the matrix are initialized by PETSc library. PETSc library does a partition of arrays and the matrix across different processes. The matrix and arrays are constructed in a parallel way. Each process keeps in memory a single part of every structure. The formulas of matrix values



(a) $\alpha_1 = 1.3,\ \alpha_2 = 1.7$ (b) $\alpha_1 = \alpha_2 = 1.5$ (c) $\alpha_1 = \alpha_2 = 1.75$

(d) $\alpha_1 = \alpha_2 = 1.8$ (e) $\alpha_1 = \alpha_2 = 1.9$ (f) $\alpha_1 = \alpha_2 = 2.0$

Fig. 2: Spiral waves for different values of $\alpha_1,\ \alpha_2$ (curvature of the waves is due to boundary conditions)

are described in numerical scheme (12). After initialization and construction of the matrix, the program sets u^0 and v^0 in order to initiate the spiral wave in the simulation (16)-(17). It should be noted that in the main computational cycle each operation is conducted in a parallel way and the linear system is solved by conjugate gradient method.

3 Numerical Experiments and Results

We consider three series of experiments to study performance of the simulation. In the first series, we establish the effect of order of fractional derivative to the performance. We take different values for $\alpha_1,\ \alpha_2$ and fix the number of cores to one. The second series is concerned with the establishment of scalability for different numbers of threads (from 1 to 12). The fractional orders $\alpha_1 = \alpha_2$ are fixed. The third series discusses the relation between number of mesh nodes and scalability. The order of fractional derivatives $\alpha_1 = \alpha_2$ chosen to be 1.9 and 2.0. All experiments were conducted on one node of Ural Federal University's cluster with two CPU Intel Xeon E5-2620 v2 @ 2.10GHz, which are joined in ccNUMA architecture.

In [6], the authors compare features of excitation wave behavior when the order of fractional derivative is equal to 1.9, 1.75, and 1.5. In this research, we

Algorithm 1 The algorithm of initialization and main computational cycle

```
procedure Compute($t, s, f, T$)
    Initiate($A_{i,j}$, $u^n_{i,j}$, $v^n_{i,j}$)
    ConstructTheMatrix($A$)
    SetInitialValues($u^0$, $v^0$)
    while $t \cdot n < T$ do
        $b \leftarrow u^{n-1} + t \cdot f(u^{n-1}, v^{n-1})$
        Solve($Au^n = b$)
        $v^n \leftarrow v^{n-1} + t \cdot b$
        Print($t$, $u^n$, $v^n$)
        $n \leftarrow n + 1$
    end while
    return $u^n$, $v^n$
end procedure
```

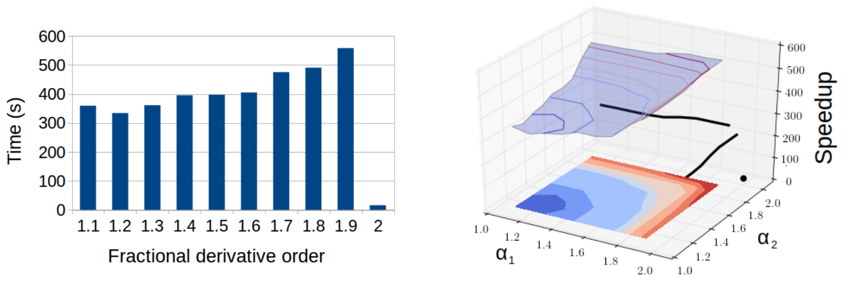


Fig. 3: Histogram (a) shows computation time for different $\alpha_1,\ \alpha_2 \in [1.1, 2.0]$, $\alpha_1 = \alpha_2$; plot (b) shows computation time for several possible combinations of $\alpha_1,\ \alpha_2 \in [1.1, 2.0]$

use the same parameters. Other variants and combinations are used to complete the study. If $\alpha_1 = \alpha_2 = 2.0$, fractional FitsHugh-Nagumo model is reduced to standard FitsHugh-Nagumo model with integer order of Laplacian. If $\alpha_1 = 1.9$ or $\alpha_2 = 1.9$, the simulation behavior is similar to the case of ordinary Laplacian, but performance drops. This happens because the matrix of the linear system is more complex for fractional derivative, see Fig. 1. This leads to a high number of steps in iterative CG solver.

The first series of experiments are done on a square area with size 2.5×2.5 and 256×256 mesh nodes, step-by-time $\tau = 0.1$. The parameters α_1, α_2 are chosen in range from 1.1 to 2.0 with step 0.1. The simulation takes 1400 time steps for each parameter combination. A spiral wave initiation protocol (16,17) is used as initial condition.The computational power is restricted by one core.

When $\alpha_1 = \alpha_2$, $\alpha_1 \in [1.1, 2.0]$, the results are shown in Figure 3(a). For all possible combinations of parameters $\alpha_1 \in [1.1, 2.0]$, $\alpha_2 \in [1.1, 2.0]$, the results are shown in Figure 3(b). The computational time grows with increasing α_1, α_2

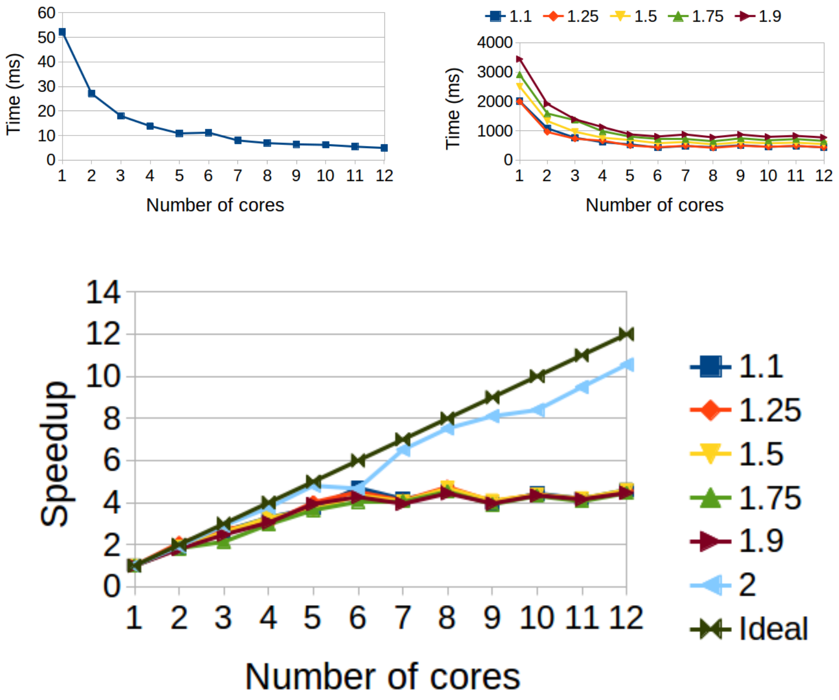


Fig. 4: Diagram (a) shows performance for integer Laplacian order $\alpha_1 = \alpha_2 = 2.0$. This is a case when fractional derivatives are reduced to ordinary derivatives. Diagram (b) shows performance for other order of fractional Laplacian when $\alpha_1 = \alpha_2$. Please, notice the scale. Diagram (c) shows scalability of the implementation for different orders of fractional derivatives

and dramatically decreases when $\alpha_1 = 2.0$ or $\alpha_2 = 2.0$. The computational time for $\alpha_1 = \alpha_2 = 2.0$ is 35 times lower than in case of $\alpha_1 = \alpha_2 = 1.9$.

We can explain this result by reduction of the fractional problem to the problem with integer Laplacian when $\alpha_1 = \alpha_2 = 2.0$. This reduction leads to a simplification of the matrix (Fig. 1). The multi-diagonal matrix of fractional problem is reduced to banded matrix with five diagonals. This simple matrix structure requires less iterations of CG solvers. The difference in computational time for different derivative orders can be explained by features of the matrix or features of spiral wave behavior.

In the second series, we fixed values of $\alpha_1 = \alpha_2$; $\alpha_1, \alpha_2 \in [1.1, 2.0]$ and studied the scalability for number of cores from 1 to 12. Fractional derivative orders are set to be 1.1, 1.25, 1.5, 1.75, 1.9, 2.0. Simulation takes 1000 time steps on a 5.0×5.0 mesh with 512×512 mesh nodes.

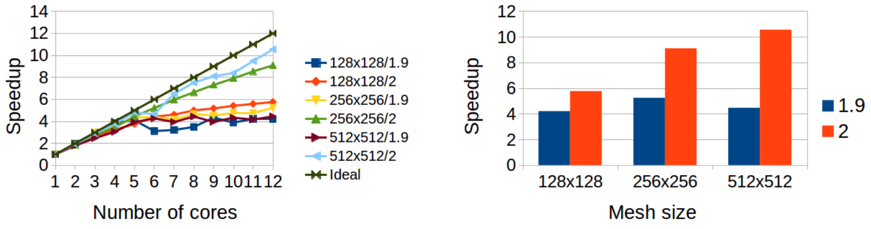


Fig. 5: A Plot (a) shows the implementation speedup for fractional Laplacian with integer and not-integer orders on computation mesh with different size. A chart (b) shows a dependency between scalability and computational mesh size when number of cores equal to 12 and parameters $\alpha_1 = \alpha_2$ are set to 1.9 or 2.0

The time of computations is shown on a Diagrams 4(a), 4(b) and scalability is shown on a Plot 4(c). The scalability is close to linear for $\alpha_1 = \alpha_2 = 2.0$, but in case $\alpha_1 = \alpha_2 < 2.0$ it increases up to a sixth core and stays on plate. Similar result is shown on two and three nodes of cluster. In our opinion, these results may be explained by counts of slow non-local memory readings from different cores in ccNUMA architecture.

In the third series, we study a dependency between scalability and mesh size. The computation is done on a different square meshes with sizes 1.25×1.25, 2.5×2.5, 5.0×5.0 and 128×128 , 256×256 , 512×512. This formulation of the problem is important, because the computational scheme 12 establishes dependency of each value to the values from adjoined row and column way to the mesh border. Expansion of the computational mesh leads to increasing of the number of nonzero elements in the matrix. Parameters of experiment are taken as: $h = 0.009767$ is step by space, $\tau = 0.1$ is step by time, 1000 time steps were computed. The order of fractional derivatives $\alpha_1 = \alpha_2$ are set to be 1.9 and 2.0.

Figure 5(b) shows scalability of the problem with fractional and ordinary Laplacians. Scalability is close to linear for a “big” mesh, which employs near or more than 256×256 mesh nodes. For fractional model, linear scalability is bounded by six cores.

For problem with integer order Laplacian, we may explain the results by decreasing of the ratio between useful computation and utility operations.

After this experiment, we suppose that our implementation may be used for various electrophysiological computational study, but the application cases are restricted by mesh size and one cluster node.

4 Comparison with Related Work

In this section we compare our implementation with some existed works in literature. In [15], the authors proposed some numerical methods for efficient computation of a class of matrix functions on GPU. This method allows to improve performance of exponential integrator and computation of fractional Laplacian. The authors describe computationally fractional-by-space reaction-diffusion equations and fractional-by-space FitzHugh-Nagumo model on CPU and GPU. Libraries cuBLAS and cuSPARCE are used as parallelization technology. MATLAB is used for implementation.

In their study, 20000 time steps are computed during 175 369 seconds with CPU and 14 579 seconds with GPU. Approximately, it is about 8.768 seconds for one time step on CPU and 0.729 seconds for GPU. Our implementation computes 1000 time steps in 772 sec. with 12 CPU cores. It is about 0.771 steps in one second. Thus, performance of our implementation is faster than CPU implementation from the paper [15] and close in performance to GPU implementation.

In [16], the authors described parallel algorithm for solution fractional-by-space reaction-diffusion equations in one dimensional case. Finite difference method is used for fractional partial derivative discretization. Performance study was achieved with one core, one CPU (Intel Xeon X5540) and 64 cores. In the last case, MPI technology is used. Programming language was Fortran 90. Remarkably, computational scheme is based on Grunwald approximations. This is similar to our study, but we solve the problem in two dimensions.

In their study, performance increased by 3.38-3.58 times with 4 cores and 4.55–5.23 times with 8 cores. We computed minimal and maximal speedup achieved in our simulation for all variant of parameters excluding case of Laplacian with integer order. In our implementation, performance increased by 2.29 – 3.85 times on 4 cores, 3.5 – 5.0 times on 8 cores and 4.06 – 5.28 times on 12 cores. Based on these metrics, we can say that our implementation scalability is close to scalability of the implementation [16].

Besides that, the authors in [17] described program to solve Maxwell's fractional equation which uses PETSc, but performance and scalability tests are not described. In [18] and [19], a study of performance for several numerical methods without parallelization was performed. In [2] and [20, 21], a study of effective parallelization techniques was done. The authors used PETSc. However, these works are not compatible with our work directly, because the authors solved the problem with another conditions and other computational methods. In addition, our study is more detailed. We study performance from hardware and model parameters together in one research.

5 Conclusion

In this paper, we describe a high performance implementation of the fractional FitzHugh-Nagumo model. It is based on numerical method [8] to approximate fractional reaction diffusion equation and conjugate gradient method to solve the sparse linear system. Our solution is done with the aid of PETSc library which is used a lot for scientific computing [9]. The simulation with fractional derivatives shows 17%-65% less performance than ones with integer order derivatives. We study the dependencies of simulation performance such as size of mesh and order of fractional derivative. In addition, we establish simulation scalability in ccNUMA architecture with two six-core CPU. The proposed implementation shows a linear scalability when a fractional problem is reduced to a problem with an ordinary Laplacian ($\alpha_1 = \alpha_2 = 2.0$). However, the implementation speedup reaches plateau from six cores and more with other values $\alpha_1,\ \alpha_2$.

The proposed implementation can be used for research of physiological features of excitable medium model with fractional derivative in the future.

Acknowledgements. The study was supported by the Russian Science Foundation (no. 14-35-00005). We used the computational cluster of Ural Federal University for this research.

References

1. Podlubny, I.: Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications. Volume 198. Academic press (1998)
2. Zhang, W.: High Performance Computing for Solving Fractional Differential Equations with Applications. PhD thesis, University of Oslo (2014)
3. Ahmed, E., Hashish, A., Rihan, F.: On fractional order cancer model. Journal of Fractional Calculus and Applied Analysis **3**(2) (2012) 1–6
4. Arafa, A., Rida, S., Khalil, M.: Fractional modeling dynamics of hiv and cd4+ t-cells during primary infection. Nonlinear Biomedical Physics **6**(1) (2012) 1
5. Xu, H.: Analytical approximations for a population growth model with fractional order. Communications in Nonlinear Science and Numerical Simulation **14**(5) (2009) 1978–1983
6. Bueno-Orovio, A., Kay, D., Grau, V., Rodriguez, B., Burrage, K.: Fractional diffusion models of cardiac electrical propagation: role of structural heterogeneity in dispersion of repolarization. Journal of The Royal Society Interface **11**(97) (2014) 20140352
7. FitzHugh, R.: Impulses and physiological states in theoretical models of nerve membrane. Biophysical journal **1**(6) (1961) 445
8. Liu, F., Zhuang, P., Turner, I., Anh, V., Burrage, K.: A semi-alternating direction method for a 2-d fractional fitzhugh–nagumo monodomain model on an approximate irregular domain. Journal of Computational Physics **293** (2015) 252–263
9. Balay, S., Abhyankar, S., et al.: PETSc Web page. http://www.mcs.anl.gov/petsc (2016)
10. Balay, S., Abhyankar, S., et al.: PETSc users manual. Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory (2016)
11. Balay, S., Gropp, W., McInnes, L., Smith, B.: Efficient management of parallelism in object oriented numerical software libraries. In Arge, E., Bruaset, A.M., Langtangen, H.P., eds.: Modern Software Tools in Scientific Computing, Birkhäuser Press (1997) 163–202
12. Samko, S., Kilbas, A., Marichev, O., et al.: Fractional integrals and derivatives. Theory and Applications, Gordon and Breach, Yverdon **1993** (1993)
13. Baleanu, D., Diethelm, K., Scalas, E., Trujillo, J.: Models and numerical methods. World Scientific **3** (2012) 10–16

14. Liu, F., Turner, I., Anh, V., Yang, Q., Burrage, K.: A numerical method for the fractional fitzhugh–nagumo monodomain model. ANZIAM Journal **54** (2013) 608–629
15. Farquhar, M., Moroney, T., Yang, Q., Turner, I.: Gpu accelerated algorithms for computing matrix function vector products with applications to exponential integrators and fractional diffusion. SIAM Journal on Scientific Computing **38**(3) (2016) C127–C149
16. Gong, C., Bao, W., Tang, G.: A parallel algorithm for the riesz fractional reaction-diffusion equation with explicit finite difference method. Fractional Calculus and Applied Analysis **16**(3) (2013) 654–669
17. Ge, J.: Fractional Diffusion Modeling of Electromagnetic Induction in Fractured Rocks. PhD thesis, Texas A&M University (2014)
18. Burrage, K., Hale, N., Kay, D.: An efficient implicit fem scheme for fractional-in-space reaction-diffusion equations. SIAM Journal on Scientific Computing **34**(4) (2012) A2145–A2172
19. Yang, Q., Turner, I., Moroney, T., Liu, F.: A finite volume scheme with preconditioned lanczos method for two-dimensional space-fractional reaction–diffusion equations. Applied Mathematical Modelling **38**(15) (2014) 3755–3762
20. Alyoubi, A., Ganesh, M.: An efficient hpc framework for parallel long-time and large-scale simulation of a class of anomalous single-phase models. In: High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on, IEEE (2015) 1552–1557
21. Alyoubi, A., Ganesh, M.: Parallel mixed fem simulation of a class of single-phase models with non-local operators. Journal of Computational and Applied Mathematics **307** (2016) 106–118