Parameters identification for fractional-fractal model of filtration-consolidation using GPU

Vsevolod Bohaienko[a], Anatolij Gladky[a]

[a] *V.M. Glushkov Institute of Cybernetics of NAS of Ukraine, Glushkov ave., 40, Kyiv, 03187, Ukraine*

Abstract

The paper considers some computational problems arising in the important practical field of the determination of safe operation conditions of engineering facilities that pollute soils and groundwater. In the case of complex geological and hydrological conditions, such problems are widely considered using mathematical modeling of deformation and consolidation processes in water-saturated soils, particularly, in the foundations of hydraulic structures. To simulate the dynamics of such processes, we use a fractional-fractal approach that allows considering temporal non-locality of transfer processes in media of fractal structure. The used one-dimensional differential model contains a non-local Caputo derivative with respect to the time variable and a local fractal derivative with respect to the space variable. Some of model's parameters, namely the orders of fractional derivatives, can only be determined fitting them to the measured data related to the state of a process. We propose to use particle swarm optimization algorithm to perform an identification of fractional derivatives' orders and present the results of its testing on noised subsets of direct problem solutions. In this context, we have determined that the order of space-fractal derivative is restored with a relative error of not more than 1% while the order of time-fractional derivative is restored with higher errors of not more than 10%. The lowest number of observation points that ensures stable restoration of the orders was equal to 25. As high computational complexity is combined with highly independent computational blocks while applying evolutionary optimization algorithms to the problems of differential models' parameters identification, we implemented the proposed algorithm on graphical processing units (GPU) using OpenCL framework and on multi-threaded systems using OpenMP. The results of performance testing showed up to 4-times lower GPU execution time compared to the case of multi-threaded execution on 6 cores of central processing unit (CPU).

Keywords

filtration-consolidation processes, fractional-fractal model, parameters identification, particle swarm optimization, GPU, OpenCL.

1. Introduction

The determination of the conditions for safe functioning of numerous engineering facilities that pollute soils and groundwater are among the most important and relevant in the connection with environment protection issues. This makes urgent the development of effective and reliable methods for mathematical modeling of deformation and consolidation dynamics in saturated soils, particularly, in the foundations of hydraulic structures. Such methods are usually based on the solution of initial-boundary value problems for the corresponding systems of partial differential equations (e.g. [1-3]). Recently, in order to take into account the effects of temporal and spatial correlations, a series of models have been developed by introducing into them local and non-local operators of fractional-order differentiation (e.g. [4-6]).

CMIS-2021: The Fourth International Workshop on Computer Modeling and Intelligent Systems, April 27, 2021, Zaporizhzhia, Ukraine
EMAIL: sevab@ukr.net (V. Bohaienko); gladky@ukr.net (A. Gladky)
ORCID: 0000-0002-3317-9022 (V. Bohaienko); 0000-0002-3839-9550 (A. Gladky)

In this context, we continue the studies presented in [6] on the simulation of anomalous processes of filtration-consolidation in water-saturated soils under the influence of salt transfer. The model developed in [6] uses the fractional-fractal approach [7-10] that enables modeling of temporal non-locality of processes in soils of fractal structure. Comparing to the previously developed mass transfer models within such an approach this model widens its application taking into consideration soil compaction and chemical osmosis.

Practical usage of fractional-order differential models to predict the dynamics of mass transfer and soil compaction requires identification of their parameters. However, the orders of fractional derivatives usually do not have technical means of measurement. The only approach that can be used to determine their values is to select them the way to make the model best describe available measurements.

In general, two approaches used to identify the parameters of fractional-differential models can be singled out. Within the first one, corresponding inverse problems are solved analytically or using numerical-analytical methods (e.g. [11, 12]). Such an approach is usually applied to relatively simple, mostly one-dimensional models. Despite some problems of restoration of fractional derivative orders are solved by least-squares techniques [13] or the Tikhonov method [14], the second approach that is based on metaheuristic optimization techniques (e.g. [15-17]) have less limitation on its usage. One of such algorithms that were efficiently applied [17] to some models of mass transfer processes in soils is the particle swarm optimization (PSO) algorithm [18]. Regarding most metaheuristic optimization algorithms it should also be noted that data parallelism can be easily exploited in them.

Hence, in this paper we study the accuracy and performance of the PSO algorithm applying it to solve parameters identification problem for the fractal-fractional model of filtration-consolidation proposed recently in [6]. We also consider PSO algorithm's implementation on graphical processors (GPU) and study its performance comparing it with the case of multi-threaded implementation on CPU.

2. Direct problem

Considering the non-local in time isothermal filtration-consolidation process in a soil of fractal structure saturated with a salt solution, we study the following mathematical model [6]:

$$D_t^{(\beta)} H = \frac{\partial}{\partial x^\alpha}\left(\frac{\partial}{\partial x^\alpha}\left(C_\upsilon H - \mu C \right) \right) \tag{1}$$

$$\sigma\, D_t^{(\beta)} C = d_* \frac{\partial}{\partial x^\alpha}\left(\frac{\partial C}{\partial x^\alpha} \right) + \frac{\partial}{\partial x^\alpha}\left(kH - \nu C \right) \cdot \frac{\partial C}{\partial x^\alpha} \tag{2}$$

where $H(x,t)$ is the water head (m), $C(x,t)$ is the concentration of salts in the liquid phase (kg/m^3), k is the filtration coefficient in fractal dimension ($m^{2\alpha-1}/s^\beta$), ν is the coefficient of chemical osmosis in fractal dimension ($\frac{m^{3+2\alpha}}{kg \cdot s^\beta}$), d_* is the coefficient of convective diffusion in fractal dimension ($m^{2\alpha}/s^\beta$), C_υ is the consolidation coefficient in fractal dimension ($m^{2\alpha}/s^\beta$), $\mu = \frac{\nu C_\upsilon}{k}$, σ is the porosity of a medium, $D_t^{(\beta)} f(t) = \frac{1}{\Gamma(1-\beta)} \frac{\partial}{\partial t}\left(\int_0^t (t-\tau)^{-\beta} f(\tau) d\tau - \tau^{-\beta} f(0) \right)$ is the regularized fractional Caputo derivative of the order β, $0 < \beta \le 1$ with respect to the variable t [19-21], $\frac{\partial}{\partial x^\alpha}$ is the operator of fractal derivative [7-9], which can be represented under the known conditions as $\frac{\partial}{\partial x^\alpha} f(x) = \frac{d}{dx} f(x) \frac{1}{\alpha x^{\alpha-1}}$, $\alpha > 0.5$ is the fractal dimension.

The equations (1),(2) contains coefficients in fractal dimension in space and time. To make them consistent with measurable values, we, following [22], introduce parameters σ_t (s), σ_x (m) of time

and space dimensions and represent fractal dimensional filtration coefficient as $k = \bar{k}\dfrac{\sigma_x^{2-2\alpha}}{\sigma_t^{1-\beta}}$ where $\bar{k}$ is the measurable value of the coefficient. The same is performed for the other coefficients in fractal dimension. While the choice of the values of dimensional parameters σ_t, σ_x strongly influence simulation results, there are significant difficulties of their direct determination. Thus, we consider the model (1),(2) as semi-empirical and set $\sigma_t = \sigma_x = 1$ (one of the cases studied in [22]) further focusing on the identification of fractional derivatives' orders only.

Let us note, that from equations (1), (2) when $\alpha \to 1$ we obtain a system of equations [23] of the corresponding fractional-differential model that does not consider fractal properties of a soil. When $\alpha, \beta \to 1$ we obtain the integer-order classical model [1, 2].

Using in (1), (2) the representation of the fractal derivative operator on the base of integer-order derivative [7-9] we obtain the system of equations in the following form [6]:

$$D_t^{(\beta)} H = C_\upsilon \left[s_\alpha(x) \frac{\partial^2 H}{\partial x^2} + r_\alpha(x) \frac{\partial H}{\partial x} \right] - \mu \left[s_\alpha(x) \frac{\partial^2 C}{\partial x^2} + r_\alpha(x) \frac{\partial C}{\partial x} \right] \tag{3}$$

$$\sigma\, D_t^{(\beta)} C = d_* \left[s_\alpha(x) \frac{\partial^2 C}{\partial x^2} + r_\alpha(x) \frac{\partial C}{\partial x} \right] + s_\alpha(x) \frac{\partial \upsilon}{\partial x} \cdot \frac{\partial C}{\partial x} \tag{4}$$

where

$$\upsilon(x,t) = kH(x,t) - \nu C(x,t),\ r_\alpha(x) = \frac{1-\alpha}{\alpha^2} x^{1-2\alpha},\ s_\alpha(x) = \frac{1}{\alpha^2} x^{2(1-\alpha)}.$$

To describe the dynamics of the considered process in the domain $\Omega = \{(x,t):\ 0 < x < l, \quad t > 0\}$ with permeable boundaries we complement the equations (3), (4) with boundary conditions

$$H(0,t) = 0, \quad H(l,t) = 0, \quad H(x,0) = H_0 \tag{5}$$

$$C(0,t) = C_0, \quad C(l,t) = 0, \quad C(x,0) = 0 \tag{6}$$

where H_0 is the initial value of water head, C_0 is the specific value of salts concentration at the inlet of filtration flow.

3. Numerical method for the direct problem

We numerically solve the boundary value problem (3)-(6) using a finite-difference technique briefly described below.

In the grid domain $\omega_{h\tau} = \{(x_i, t_j): x_i = ih \quad (i = \overline{0, m+1}),\ t_j = j\tau \quad (j = \overline{0, n})\}$ where h, τ are the grid steps with respect to the spatial variable and time the considered problem can be discretized using the linearized Crank–Nicholson scheme written using the notations from [24] as [6]

$$\sigma\, \Delta_t^{(\beta)} C = 0.5 d_* \left[s_\alpha \left(\hat{C}_{\bar{x}x} + C_{\bar{x}x} \right) + r_\alpha \left(\hat{C}_{\overset{0}{x}} + C_{\overset{0}{x}} \right) \right] + 0.5 s_\alpha \upsilon_{\overset{0}{x}} \left(\hat{C}_{\overset{0}{x}} + C_{\overset{0}{x}} \right) \tag{7}$$

$$\Delta_t^{(\beta)} H = 0.5 C_\upsilon \left[s_\alpha \left(\hat{H}_{\bar{x}x} + H_{\bar{x}x} \right) + r_\alpha \left(\hat{H}_{\overset{0}{x}} + H_{\overset{0}{x}} \right) \right] - 0.5 \mu \left[s_\alpha \left(\hat{C}_{\bar{x}x} + C_{\bar{x}x} \right) + r_\alpha \left(\hat{C}_{\overset{0}{x}} + C_{\overset{0}{x}} \right) \right] \tag{8}$$

where $\Delta_t^{(\beta)} u$ is the discrete analogue of the fractional derivative $D_t^{(\beta)} u$ defined as

$$\Delta_{t_{j+1}}^{(\beta)} u \approx \frac{u^{j+1} - u^j}{\tau^\beta \Gamma(2-\beta)} + \sum_{\nu=0}^{j-1} \omega_\nu^{(j)} \frac{u^{\nu+1} - u^\nu}{\tau},\ \omega_\nu^{(j)} = \frac{\tau^{1-\beta}}{\Gamma(2-\beta)} \left[(j-\nu+1)^{1-\beta} - (j-\nu)^{1-\beta} \right], \tag{9}$$

and $\Gamma(z)$ is the Euler's gamma function.

Substituting (9) into (7), (8) we reduce the solution of the problem (3)-(6) on the $(j+1)$-th time step to the solution of the systems of linear algebraic equations

$$A_i^j C_{i-1}^{j+1} - S_i^j C_i^{j+1} + B_i^j C_{i+1}^{j+1} = F_i^j \qquad (i = \overline{1, m};\ j = \overline{0, n}), \tag{10}$$

$$\tilde{A}_i^j H_{i-1}^{j+1} - \tilde{S}_i^j H_i^{j+1} + \tilde{B}_i^j H_{i+1}^{j+1} = \tilde{F}_i^j \qquad (i = \overline{1, m};\ j = \overline{0, n}), \tag{11}$$

$$C_0^{j+1} = C_0, \quad C_{m+1}^{j+1} = 0, \quad C_i^0 = 0 \qquad (i = \overline{0, m+1};\ j = \overline{0, n}), \tag{12}$$

$$H_0^{j+1}=0,\quad H_{m+1}^{j+1}=0,\quad H_i^0=H_0 \qquad (i=\overline{0,m+1};\ j=\overline{0,n}) \tag{13}$$

where

$$A_i^j=\frac{0.5}{h}\left[d_*\left(\frac{s_\alpha^i}{h}-\frac{r_\alpha^i}{2}\right)-\frac{s_\alpha^i}{4h}\left(\upsilon_{i+1}^j-\upsilon_{i-1}^j\right)\right],\ B_i^j=\frac{0.5}{h}\left[d_*\left(\frac{s_\alpha^i}{h}+\frac{r_\alpha^i}{2}\right)+\frac{s_\alpha^i}{4h}\left(\upsilon_{i+1}^j-\upsilon_{i-1}^j\right)\right],$$

$$S_i^j=A_i^j+B_i^j+\frac{\sigma}{\tau^\beta\Gamma(2-\beta)},$$

$$F_i^j=\sigma\left[\sum_{v=0}^{j-1}\omega_v^{(j)}\frac{C_i^{v+1}-C_i^v}{\tau}-\frac{C_i^j}{\tau^\beta\Gamma(2-\beta)}\right]-\frac{0.5d_*}{h}\left[\frac{s_\alpha^i}{h}\left(C_{i-1}^j-2C_i^j+C_{i+1}^j\right)+\frac{r_\alpha^i}{2}\left(C_{i+1}^j-C_{i-1}^j\right)\right]-$$

$$-\frac{0.5s_\alpha^i}{4h^2}\left(\upsilon_{i+1}^j-\upsilon_{i-1}^j\right)\left(C_{i+1}^j-C_{i-1}^j\right),$$

$$\tilde{A}_i^j=\frac{0.5C_\upsilon}{h}\left(\frac{s_\alpha^i}{h}-\frac{r_\alpha^i}{2}\right),\ \tilde{B}_i^j=\frac{0.5C_\upsilon}{h}\left(\frac{s_\alpha^i}{h}+\frac{r_\alpha^i}{2}\right),\ \tilde{S}_i^j=\tilde{A}_i^j+\tilde{B}_i^j+\frac{1}{\tau^\beta\Gamma(2-\beta)},$$

$$\tilde{F}_i^j=\sum_{v=0}^{j-1}\omega_v^{(j)}\frac{H_i^{v+1}-H_i^v}{\tau}-\frac{H_i^j}{\tau^\beta\Gamma(2-\beta)}-\frac{0.5C_\upsilon}{h}\left[\frac{s_\alpha^i}{h}\left(H_{i-1}^j-2H_i^j+H_{i+1}^j\right)+\frac{r_\alpha^i}{2}\left(H_{i+1}^j-H_{i-1}^j\right)\right]+$$

$$+\frac{0.5\mu}{h}\left[\frac{s_\alpha^i}{h}\left(C_{i-1}^{j+1}-2C_i^{j+1}+C_{i+1}^{j+1}+C_{i-1}^j-2C_i^j+C_{i+1}^j\right)+\frac{r_\alpha^i}{2}\left(C_{i+1}^{j+1}-C_{i-1}^{j+1}+C_{i+1}^j-C_{i-1}^j\right)\right],$$

$$s_\alpha^i=s_\alpha(x_i),\ r_\alpha^i=r_\alpha(x_i),\ \upsilon_i^j=kH_i^j-\nu C_i^j.$$

Equations (10)-(13) can be effectively solved by the Thomas algorithm [24].

4. Inverse problem and particle swarm optimization algorithm

Having the direct problem and numerical method stated, we further consider the inverse problem of identification of derivatives' orders α,β in the model (3)-(6) on the base of some subset of possibly noised measurements. The inverse problem is posed the following way:

- assume there are N known values C_i, $H_i, i=1,...,N$ of concentration and water head measured in the moments of time T_i in the depths x_i;
- assuming that the values of other model's parameters are known, find such values $\bar{\alpha},\bar{\beta}$ of orders α,β that minimize the following goal function:

$$F(\hat{\alpha},\hat{\beta})=\sum_{i=1}^{N}\left[\left(H(x_i,T_i,\hat{\alpha},\hat{\beta})-H_i\right)^2+\left(C(x_i,T_i,\hat{\alpha},\hat{\beta})-C_i\right)^2\right] \tag{14}$$

where $C(x,t,\hat{\alpha},\hat{\beta})$, $H(x,t,\hat{\alpha},\hat{\beta})$ are the solutions of the direct problem (3)-(6) with $\alpha=\hat{\alpha}$, $\beta=\hat{\beta}$ in the point x at the time t.

Taking into account the complexity of the inverse problem and the fact that the parameters to be identified are floating-point numbers, we propose to solve it by the particle swarm optimization (PSO) algorithm [18], which can be briefly described as follows:

- denote S as the number of particles in the swarm; $\bar{x}_i=(\hat{\alpha},\hat{\beta})$, v_i as the coordinates and velocity of the particle i; p_i as the coordinates of the particle i that corresponds to the best goal function (14) value obtained by it; g as the coordinates that corresponds to the best goal function value obtained by the swarm;
- introduce the parameters of the algorithm $\omega, \varphi_p, \varphi_g$;
- at the initialization stage, the coordinates of the particles are generated randomly and velocities are set to zero. The values of the goal function are calculated for each particle along with the initial values of p_i and g.
- on the iteration j for the particle i

- generate random numbers $r_p, r_g \in [0,1]$;
- modify the velocity: $v_i^{(j+1)} = \omega \cdot v_i^{(j)} + \varphi_p r_p (p_i^{(j)} - \overline{x}_i^{(j)}) + \varphi_g r_g (g^{(j)} - \overline{x}_i^{(j)})$;
- modify particle's coordinates: $\overline{x}_i^{(j+1)} = \overline{x}_i^{(j)} + v_i^{(j)}$;
- calculate goal function value and modify p_i and g.

- Iteration process is finished when the given maximal number of iterations $I_{\max}$ is reached, the zero best value of the goal function is achieved, or the difference between the best and the worst values of the goal function for the particles in the swarm becomes lower than the given constant.

In our case, the computation of direct problem's solutions is the most time-consuming part when identifying parameters values. As the changes in particles' states are independent, the corresponding direct problems can be efficiently solved on shared-memory parallel systems.

5. Parallel implementation

The following scheme is used to implement the solution of parameters' identification problem on GPU.

As the solution of direct problems while calculating goal function values forms the main part of algorithm's computational complexity, we choose it as the only part that is executed on GPU. We allocate a group of threads for each particle and, correspondingly, for a set of derivatives' orders for which the problem (3)-(6) has to be solved.

The main share of time while solving the direct problem is formed by the calculation of sums derived from the discretization (9) of the Caputo fractional derivative because the number of terms in them grows linearly with the increase of time step number. These sums must be calculated on each time step for each grid node, thus, within a thread block we use thread to grid node mapping. So, on each time step, at the first stage all threads compute the values of sums in the right parts of (10),(11) and then the threads with local ids 0 perform the solution of (10),(11). Solution on GPU is performed up to the moment of time $\max\limits_i T_i$ without interaction with CPU. Then the solutions for the time steps $j: t_j \le T_i < t_{j+1}$ are transferred into the memory of CPU and the values of $H(x_i, T_i, \hat{\alpha}, \hat{\beta})$ and $C(x_i, T_i, \hat{\alpha}, \hat{\beta})$ are determined using linear interpolation both in space and time.

In such a computational scheme, the amount of used GPU memory is $S \cdot (m+2) \cdot \left\lceil \max\limits_i T_i / \tau \right\rceil$ and the number of running GPU threads is $S \cdot (m+2)$. Assuming that the time spent on calculations on CPU can be neglected comparing with the time spent on GPU and recalling that the Thomas algorithm has the complexity order $O(m)$, the execution time of the presented algorithm is $T(S,m,n) = k \cdot m \cdot \lfloor S \cdot (m+2) / N_c \rfloor \cdot n \cdot (n-1)/2$ where k is the parameter of system's performance, N_c is the number of GPU cores, n is the number of time steps.

Further, we compare GPU implementation's performance with the performance of multi-threaded implementation in which for the same procedure CPU core to particle mapping is used.

6. Accuracy testing results

The described solution schemes for direct and inverse problems were implemented in C++ language. The source codes are publicly available through https://github.com/sevaboh/cons_chem_osm. To experimentally study the accuracy of the inverse problem's solution algorithm, we generate testing datasets as described below.

We fix the following values of physical parameters [6]: $C_v = 0.34$, $l = 25$, $\mu = 0.00095$, $\sigma = 0.38$, $d = 0.02$, $k = 0.01$, $\nu = 2.8 \cdot 10^{-5}$, $C_0 = 200$, $H_0 = 10$. With $\tau = 0.01$ and fixed m, α, β, the problem (3)-(6) is solved up to the time step for which $T_e = 1$. Then, the given per cent P of grid nodes that simulate the number of observation points are randomly selected as x_i. Setting all $T_i = T_e$,

we further set $H_i = H(x_i, T_e, \hat{\alpha}, \hat{\beta}) + r$, $C_i = C(x_i, T_e, \hat{\alpha}, \hat{\beta}) + r$ where r is the random variable that is uniform in $[-0.5R, 0.5R]$, R is the given noise level. The solution to the problem (3)-(6) obtained for $T_e = 2$ and noised values in observation points for $R = 2$, $P = 0.5$, $\beta = 0.6, \alpha = 0.8$ are depicted in Figure 1.

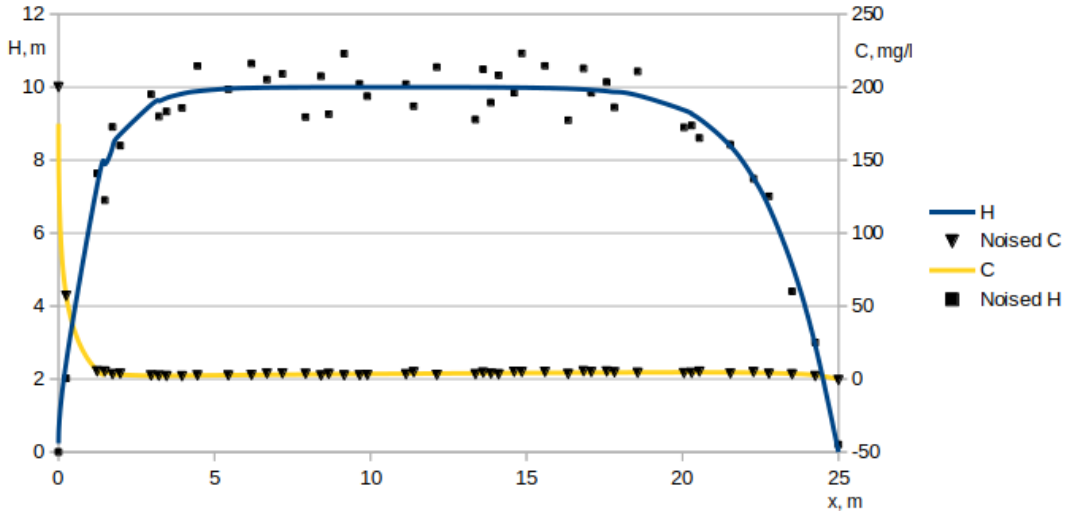

Figure 1: Solutions and noised values in observation points

Several series of computational experiments were performed for $m = 100$ and different values of α, β, P, R. In the first one with fixed $\beta = 0.6, \alpha = 0.8$ and $S = 10, \omega = \varphi_p = \varphi_g = 0.5$, we tested the influence of noise level and number of observation points on the accuracy of fractional derivatives orders restoration. The values of α, β were restricted to the range $[0.5,1]$; P was equal to 0.75, 0.5, 0.25, and 0.1; R was equal to 1, 2, and 3. We performed 10 runs of the algorithm with 30 iterations in each run.

The value of α was in all conducted experiments restored with the maximal relative error of 0.65%. Standard deviation within the performed 10 runs was not more than 0.29% of the obtained average values of α.

The relative errors for the case of β are given in Table. 1. Here the errors increase with the increase of noise level R. For $P \geq 0.25$ (more than 25 observation points) changes in the errors are small but its decrease to 0.1 (10 observation points) up to 3 times decrease the accuracy. Standard deviation behave the similar way being 4% of the average values for $P \geq 0.25$ and becoming up to 8% for $P = 0.1$.

Table 1
Relative errors of the restoration of β

R	$P = 0.75$	*0.5*	*0.25*	*0.1*
1	4.98%	1.85%	4.36%	16.66%
2	6.78%	6.86%	7.10%	26.13%
3	9.13%	8.74%	10.16%	34.16%

To determine the influence of initial values of α and β on the accuracy of their restoration we change α in the range [0.55, 0.95] for the fixed $\beta = 0.75$ and vice versa change β in the same range for $\alpha = 0.75$. The obtained relative errors are given for $P = 0.5, R = 2$ in Tables 2 and 3. In these computational experiments, α was as in the previous one restored with rather high accuracy (relative error not more than 0.6% in all cases). The error of β restoration being not more than 8.43% was higher when β tends to the edges of the range.

The last series of experiments with variable T_e showed that better accuracy of parameters identification can be achieved for higher values of T_e: relative error of β restoration with $R = 2$, $P = 0.5$ lowered from ~7% for $0.1 \le T_e \le 1$ to ~1.5% for $T_e = 2$.

Table 2

Relative errors e_α, e_β of α and β restoration for the fixed $\alpha = 0.75$

β	e_β	e_α
0.55	6.96%	0.30%
0.6	6.67%	0.28%
0.7	3.26%	0.25%
0.8	0.94%	0.32%
0.95	8.43%	0.60%

Table 3

Relative errors e_α, e_β of α and β restoration for the fixed $\beta = 0.75$

α	e_β	e_α
0.55	0.48%	0.18%
0.6	0.23%	0.16%
0.7	4.73%	0.22%
0.8	0.96%	0.43%
0.95	5.52%	0.55%

7. Solution time testing results

GPU implementation of the PSO algorithm was performed using the OpenCL framework while multi-threaded implementation used OpenMP.

Time spent on parameters' identification using GPU and multi-threaded implementations was measured solving the problem with $\beta = 0.6, \alpha = 0.8, P = 0.5, R = 2$ on one node of SCIT-4 cluster of VM Glushkov Institute of cybernetics of NAS of Ukraine (NVidia RTX 2080 Ti GPU, 2 Intel(R) Xeon(R) Bronze 3104 CPUs, 1 CPU with 6 cores was allocated for OpenMP).

During the tests we changed the number m of nodes in the finite-difference grid and the number S of particles in PSO swarm. Average solution time among 10 runs is given in Table 4. Regarding the accuracy, it increased with the increase of the number of observation points as m increases with fixed P reaching the maximal relative error of 0.24% for $m = 200$.

Table 4

Average solution time, ms

	$m = 50$		$m = 100$		$m = 200$	
S	GPU	CPU	GPU	CPU	GPU	CPU
10	10524	11819	19944	23586	38791	46907
20	10539	23162	20010	46407	38930	92550
30	10590	29163	20082	58477	39008	116521
40	10652	40834	20089	80818	39160	162622

As can be seen from the data in Table 4, execution time of GPU algorithm does not depend on the number of particles and linearly increases with the increase of grid size. This can be due to the non-complete allocation of GPU's computational resources. On the other hand, CPU execution time linearly increases both with the increase of m and S that proves the efficiency of GPU implementation, which in the conducted experiments gave up to 4-times decrease of solution time.

The efficiency of GPU implementation slightly increased with the increase of m: average solution time per one grid node per particle decreased from 21.1 ms for $m=50$ to 19.4 ms for $m=200$. The corresponding indicator for CPU implementation did not change significantly with the change of m but decreased from 23.6 ms for $S=10$ to 20.4 ms for $S=40$. Such behavior reflects the general heuristic that parallel implementation's efficiency increase with the increase of the volume of computations.

The series of computational experiments conducted for $S=10$, $N=100$ with variable $T_e=0.5,1,2,3,4$ confirmed quadratic dependency of execution time on the number of time steps (Figure 2). GPU implementation here allowed obtaining slower increase of solution time.

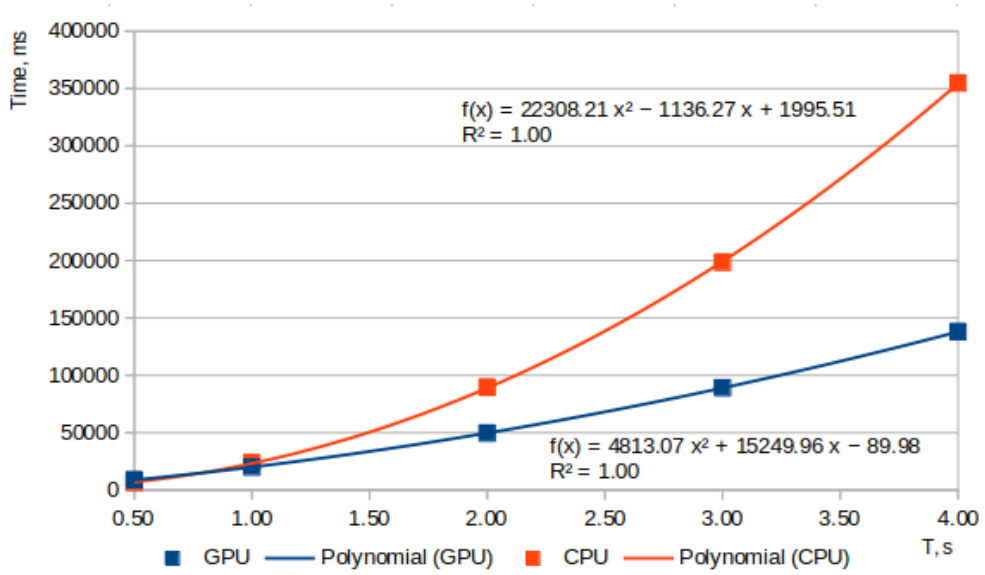

Figure 2: Execution time of GPU and CPU implementations subject to the ending time of simulation

8. Conclusions

In the paper we presented the computational scheme for identifying the values of fractional derivatives orders in the fractional-fractal model of filtration-consolidation based on the available measurements. The scheme is based on the particle swarm optimization technique and is implemented in multi-threaded mode and for the execution on GPU.

The results of computational experiments show that the order of the local fractal derivative with respect to the spatial variable can be restored with significantly higher accuracy than the order of non-local time-fractional derivative. Expectedly, relative errors increase with the increase of noise level. Considering the number of observation points, errors less than ~10% were obtained in all cases for their number more or equal to 25.

OpenCL GPU implementation of the inverse problem's solution algorithm performed closely to the OpenMP CPU implementation that used 6 CPU cores on the minimal tested volume of computations ($m=50$, $S=10$). With the increase of the volume, GPU overperformed CPU by up to 4 times.

Thus, the proposed computational scheme can be effectively applied with regard to both accuracy and speed to adapt the fractional-fractal model to the observed dynamics of salt transfer and compaction of water-saturated soils in complex geological and hydrological conditions.

9. References

[1] V.M. Bulavatsky, Iu.H. Kryvonos, V.V. Skopetsky, Non-classical mathematical models of heat and mass transfer (in Ukrainian), Naukova Dumka, Kyiv, Ukraine, 2005.

[2] A.P. Vlasiuk, P.M. Martyniuk, Mathematical modeling of consolidation in soils in the conditions of salt solution filtration (in Ukrainian), Publishing house of UDUVHP, Rivne, Ukraine, 2004.

[3] V.A. Florin, Fundamentals of Soil Mechanics, National Technical Information Service, Moscow, USSR, 1961.

[4] V.M. Bulavatsky, Some modelling problems of fractional-differential geofiltrational dynamics within the framework of generalized mathematical models. Journal of Automation and Information Science 48(5) (2016) 27–41. doi:10.1615/JAutomatInfScien.v48.i5.30.

[5] V.M. Bulavatsky, Bohaienko V.O., Numerical simulation of fractional-differential filtration-consolidation dynamics within the framework of models with non-singular kernel. Cybernetics and Systems Analysis 54(2) (2018) 193–204. doi:10.1007/s10559-018-0020-5.

[6] V. Bohaienko, V. Bulavatsky, Fractional-Fractal Modeling of Filtration-Consolidation Processes in Saline Saturated Soils. Fractal and Fractional 4 (4) 2020. doi:10.3390/fractalfract4040059.

[7] A. Allwright, A. Atangana, Fractal advection-dispersion equation for groundwater transport in fractured aquifers with self-similarities. The European Physical Journal Plus 133(2) (2018) 1–14. doi:10.1140/epjp/i2018-11885-3.

[8] W. Chen, Time-space fabric underlying anomalous diffusion. Chaos, Soliton. Fract. 28(4) (2006) 923–929. doi:10.1016/j.chaos.2005.08.199.

[9] W. Cai, W. Chen, F. Wang, Three-dimensional Hausdorff derivative diffusion model for isotropic/anisotropic fractal porous media. Thermal Science 22(1) (2018) S1-S6.

[10] A. Atangana, A. Akgül, K.M. Owolabi, Analysis of fractal fractional differential equations. Alexandria Engineering Journal 59(3) 2020 1117-1134. doi:10.1016/j.aej.2020.01.005.

[11] L.D. Long, N.H. Luc, Y. Zhou, C. Nguyen, Identification of Source Term for the Time-Fractional Diffusion-Wave Equation by Fractional Tikhonov Method. Mathematics 7 (2019). doi:10.3390/math7100934.

[12] V.M. Bulavatsky, V.O. Bohaienko, Some Consolidation Dynamics Problems within the Framework of the Biparabolic Mathematical Model and its Fractional-Differential Analog. Cybernetics and Systems Analysis 56(5) (2020) 770-783. doi:10.1007/s10559-020-00298-7.

[13] K. Liao, T. Wei, Identifying a fractional order and a space source term in a time-fractional diffusion-wave equation simultaneously. Inverse Problems 35(11) (2019). doi:10.1088/1361-6420/ab383f.

[14] M. Krasnoschok, S. Pereverzyev, S.V. Siryk, N. Vasylyeva, Determination of the fractional order in semilinear subdiffusion equations. Fractional Calculus and Applied Analysis 23(3) (2020) 694-722. doi:10.1515/fca-2020-0035.

[15] F. Gao, X.J. Lee, H.Q. Tong, F.X. Fei, H.L. Zhao, Identification of Unknown Parameters and Orders via Cuckoo Search Oriented Statistically by Differential Evolution for Noncommensurate Fractional-Order Chaotic Systems. Abstract and Applied Analysis 2013 (2013). doi:10.1155/2013/382834.

[16] L.G. Yuan, Q.G. Yang, Parameter identification and synchronization of fractional-order chaotic systems. Commun. Nonlinear Sci. Numer. Simul. 17(1) (2012) 305–316. doi:10.1016/j.cnsns.2011.04.005.

[17] V. Bohaienko, A. Gladky, M. Romashchenko, T. Matiash, Identification of fractional water transport model with ψ-Caputo derivatives using particle swarm optimization algorithm. Applied Mathematics and Computation 390 (2021). doi:10.1016/j.amc.2020.125665.

[18] Y. Zhang, Sh. Wang, G. Ji, A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. Mathematical Problems in Engineering. 2015 (2015). doi:10.1155/2015/931256.

[19] I. Podlubny, Fractional differential equations, Academic Press, New York, 1999.

[20] A.A. Kilbas, H.M. Srivastava, J.J. Trujillo, Theory and applications of fractional differential equations, Elsevier, Amsterdam, The Netherlands, 2006.

[21] T. Sandev, Z. Tomovsky, Fractional equations and models. Theory and applications, Springer Nature Switzerland AG, Cham, Switzerland, 2019.

[22] J.F. Gómez-Aguilar, M. Miranda-Hernández, M.G. López-López, V.M. Alvarado-Martínez, D. Baleanu, Modeling and simulation of the fractional space-time diffusion equation. Communications in Nonlinear Science and Numerical Simulation 30 (2016) 115–127. doi:10.1016/j.cnsns.2015.06.014.
[23] V.M. Bulavatsky, Mathematical Model of Geoinformatics for Investigation of Dynamics for Locally Nonequlibrium Geofiltration Processes. Journal of Automation and Information Sciences 43(12) (2011) 12–20. doi:10.1615/JAutomatInfScien.v43.i12.20.
[24] A. Samarskii, The Theory of Difference Schemes, CRC Press, New York, 2001.