fractal and fractional

MDPI

Article

Parameter Identification in the Two-Dimensional Riesz Space Fractional Diffusion Equation

Rafał Brociek *,†, Agata Chmielowska †and Damian Słota †

Department of Mathematics Applications and Methods for Artificial Intelligence, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland; agata.chmielowska@polsl.pl (A.C.); damian.slota@polsl.pl (D.S.)

* Correspondence: rafal.brociek@polsl.pl

† These authors contributed equally to this work.

Received: 26 June 2020; Accepted: 3 August 2020; Published: 6 August 2020

check for updates

Abstract: This paper presents the application of the swarm intelligence algorithm for solving the inverse problem concerning the parameter identification. The paper examines the two-dimensional Riesz space fractional diffusion equation. Based on the values of the function (for the fixed points of the domain) which is the solution of the described differential equation, the order of the Riesz derivative and the diffusion coefficient are identified. The paper includes numerical examples illustrating the algorithm's accuracy.

Keywords: fractional derivative; fractional differential equation; inverse problem; parameter identification; two-dimensional differential equation

1. Introduction

Models with fractional derivatives have found applications in many fields of science and engineering, such as control theory [1], mechanics [2], image processing [3] and heat conduction. The use of the fractional derivatives in modeling the heat conduction phenomena has been presented in the papers [4,5], wherein the models with the fractional derivative of Riemann–Liouville and Caputo type was considered. In case of the heat conduction problems the equations with the fractional derivatives are particularly useful for modeling the phenomena that take place in the porous materials and composites due to the fact that the anomalous diffusion process occurs there. For such materials the models with the fractional derivatives give better results than the models based on the classic derivatives, as has been shown, for example, in papers [6,7]. In the articles [7,8] the heat conduction process was considered in composites and porous aluminum. More about the applications of the fractional derivatives can be found in [9–12].

In this paper the model described by the differential equation with the fractional derivative of the Riesz type [13] is being considered. The definition of this derivative is based on the left and right-hand sided Riemann–Liouville derivatives. The algorithm for solving the inverse problem for model with the partial Riesz derivative with respect to the spacial variable is being presented. The inverse problems in general are a class of problems related to the proper selection of the input parameters to obtain the expected values as an output. To solve the inverse problem unambiguously, additional information is required. In this article the problem of reconstructing the diffusion coefficient and the order of the fractional derivative is being solved and the additional information is a set of values of a function which is a solution of the considered differential equation for the chosen points of the domain.

Solving the inverse problem often requires solving the direct problem multiple times for the chosen values of the reconstructed parameters; in our case these are the diffusion coefficient and the order of the fractional derivative. For solving the direct problem, the alternating direction implicit

method [13], which allows one to solve the multidimensional problem by iteratively solving a number of one-dimensional problems, was used. The functional representing the error of the approximate result was constructed using the obtained solution of the direct problem and then it was minimized by application of the real ant colony optimization [14] algorithm. That is the probabilistic artificial intelligence algorithm inspired by the behavior of the swarm of ants. In this paper, a numerical example presenting the accuracy of the method for various numbers of measurement points and various input data disturbance is also presented.

2. Formulation of the Problem

We consider a two-dimensional differential equation with the fractional derivative of a Riesz type with respect to the spacial variable:

$$c\varrho\frac{\partial u(x,y,t)}{\partial t} = \lambda_x\frac{\partial^{\alpha_1}u(x,y,t)}{\partial|x|^{\alpha_1}} + \lambda_y\frac{\partial^{\alpha_2}u(x,y,t)}{\partial|y|^{\alpha_2}} + d(u,x,y,t), \tag{1}$$

defined on the domain $D = \{(x,y,t) : x \in [0,L_x], y \in [0,L_y], t \in [0,T], L_x, L_y, T \in \mathbb{R}_+\}$. Such an equation is called a two-dimensional Riesz space fractional diffusion equation [13], so the λ_x and λ_y coefficients are called diffusion coefficients from now on. In this equation the fractional derivatives of order $\alpha_1, \alpha_2 \in (1,2)$ are the derivatives of a Riesz type defined as follows:

$$\begin{aligned}\frac{\partial^{\alpha_1}u(x,y,t)}{\partial|x|^{\alpha_1}} &= -c_{\alpha_1}\left(\frac{\partial^{\alpha_1}u(x,y,t)}{\partial x^{\alpha_1}} + \frac{\partial^{\alpha_1}u(x,y,t)}{\partial(-x)^{\alpha_1}}\right),\\ \frac{\partial^{\alpha_2}u(x,y,t)}{\partial|y|^{\alpha_2}} &= -c_{\alpha_2}\left(\frac{\partial^{\alpha_2}u(x,y,t)}{\partial y^{\alpha_2}} + \frac{\partial^{\alpha_2}u(x,y,t)}{\partial(-y)^{\alpha_2}}\right),\end{aligned} \tag{2}$$

where $c_\alpha = \frac{1}{2\cos(\frac{\pi\alpha}{2})}$ and:

$$\begin{aligned}\frac{\partial^{\alpha_1}u(x,y,t)}{\partial x^{\alpha_1}} &= \frac{1}{\Gamma(2-\alpha_1)}\frac{\partial^2}{\partial x^2}\int_0^x \frac{u(s,y,t)}{(x-s)^{\alpha_1-1}}\,ds,\\ \frac{\partial^{\alpha_1}u(x,y,t)}{\partial(-x)^{\alpha_1}} &= \frac{(-1)^2}{\Gamma(2-\alpha_1)}\frac{\partial^2}{\partial x^2}\int_x^{L_x} \frac{u(s,y,t)}{(s-x)^{\alpha_1-1}}\,ds,\\ \frac{\partial^{\alpha_2}u(x,y,t)}{\partial y^{\alpha_2}} &= \frac{1}{\Gamma(2-\alpha_2)}\frac{\partial^2}{\partial y^2}\int_0^y \frac{u(x,s,t)}{(y-s)^{\alpha_2-1}}\,ds,\\ \frac{\partial^{\alpha_2}u(x,y,t)}{\partial(-y)^{\alpha_2}} &= \frac{(-1)^2}{\Gamma(2-\alpha_2)}\frac{\partial^2}{\partial y^2}\int_y^{L_y} \frac{u(x,s,t)}{(s-y)^{\alpha_2-1}}\,ds,\end{aligned} \tag{3}$$

where Γ is the Gamma function. To the Equation (1) we add the initial condition:

$$u(x,y,0) = f(x,y), \quad x \in [0,L_x],\ y \in [0,L_y], \tag{4}$$

and homogeneous Dirichlet boundary conditions:

$$\begin{aligned}u(0,y,t) = u(L_x,y,t) = 0,\\ u(x,0,t) = u(x,L_y,t) = 0.\end{aligned} \tag{5}$$

The differential equations of the fractional order have found applications in modeling various kinds of phenomena, such as the diffusion or heat conduction in porous materials [4,7]. In the model presented above we assume that $\alpha_1 = \alpha_2 = \alpha$ and $\lambda_x = \lambda_y = \lambda$. The considered inverse problem is to

reconstruct a value of an order of the derivative α and the diffusion coefficient λ using the values of the u function for the fixed points of the domain. We call these values the input data for the inverse problem and denote them as:

$$\hat{U}_{i,j}^k = u(x_i, y_j, t_k), \qquad i,j = 1,2,\ldots,N_1, \quad k = 1,2,\ldots,N_2,$$

where N_1 is the number of measurement points and N_2 is the number of measurements taken for each point. By solving the differential Equation (1) (numerically) for the fixed values of the α and λ parameters we can get the values of the function $U_{i,j}^k(\alpha,\lambda)$ at the measurement points. By taking these values, in each point of the mesh, and comparing them with the input data, we can build a function that defines the error of the approximate solution:

$$J(\alpha,\lambda) = \sum_{i,j=1}^{N_1} \sum_{k=1}^{N_2} \left(U_{i,j}^k(\alpha,\lambda) - \hat{U}_{i,j}^k \right)^2. \tag{6}$$

By minimizing the above function, we get the approximate values of the α and λ parameters.

3. Methods of Solution

The algorithm for solving the considered inverse problem consists of two major parts: solving the direct problem—determining the solution of the differential Equation (1) with the given conditions; and finding the minimum of the objective function (6).

3.1. Solution of the Direct Problem

In order to solve the differential Equation (1) in the considered area we introduce a grid:

$$S = \{(i\Delta x, j\Delta y, k\Delta t) : \Delta x = \frac{L_x}{N}, \Delta y = \frac{L_y}{M}, \Delta t = \frac{T}{K}, \ i = 0,1,\ldots N, j = 0,1,\ldots,M, k = 0,1,\ldots,K\}.$$

Let us take a following notation: $U_{i,j}^k \approx u(x_i, y_j, t_k) = u(i\Delta x, j\Delta y, k\Delta t)$. The initial condition may be denoted as $U_{i,j}^0 = f(x_i, y_j) = f_{i,j}$, while for the source component d we use notation $d_{i,j}^k = d(u_{i,j}^k, x_i, y_j, t_k)$. We approximate the first derivative with respect to time in the following way:

$$\frac{\partial u(x_i, y_j, t_k)}{\partial t} \approx \frac{U_{i,j}^k - U_{i,j}^{k-1}}{\Delta t}.$$

The fractional derivative (3) with respect to the x variable is approximated by the shifted Grünwald-Letnikov formula [15,16]:

$$\frac{\partial^\alpha u(x_i, y_j, t_k)}{\partial x^\alpha} \approx \frac{1}{(\Delta x)^\alpha} \sum_{r=0}^{i+1} g_\alpha^{(r)} U_{i-r+1,j}^k, \tag{7}$$

$$\frac{\partial^\alpha u(x_i, y_j, t_k)}{\partial (-x)^\alpha} \approx \frac{1}{(\Delta x)^\alpha} \sum_{r=0}^{N-i+1} g_\alpha^{(r)} U_{i+r-1,j}^k, \tag{8}$$

where:

$$g_\alpha^{(r)} = (-1)^r \frac{\Gamma(\alpha+1)}{\Gamma(\alpha-r+1)\Gamma(r+1)}.$$

We use similar method to approximate the derivative with respect to the y variable. By taking into account all of the above dependencies, Equation (1) may be written in the form:

$$\begin{aligned}\frac{U_{i,j}^k - U_{i,j}^{k-1}}{\Delta t} = &-\frac{\lambda c_\alpha}{(\Delta x)^\alpha}\left[\sum_{r=0}^{i+1} g_\alpha^{(r)} U_{i-r+1,j}^k + \sum_{r=0}^{N-i+1} g_\alpha^{(r)} U_{i+r-1,j}^k\right] \\ &-\frac{\lambda c_\alpha}{(\Delta y)^\alpha}\left[\sum_{r=0}^{j+1} g_\alpha^{(r)} U_{i,j-r+1}^k + \sum_{r=0}^{M-j+1} g_\alpha^{(r)} U_{i,j+r-1}^k\right] + f_{i,j}^{k-1}.\end{aligned} \tag{9}$$

Next we introduce the operator δ_x defined as follows:

$$\delta_x u_{i,j}^k = -\frac{\lambda c_\alpha}{(\Delta x)^\alpha}\left[\sum_{r=0}^{i+1} g_\alpha^{(r)} U_{i-r+1,j}^k + \sum_{r=0}^{N-i+1} g_\alpha^{(r)} U_{i+r-1,j}^k\right]$$

We also define the operator δ_y similarly. By substituting into (9), we get the following equation:

$$(1 - \Delta t\delta_x - \Delta t\delta_y) U_{i,j}^k = U_{i,j}^{k-1} + \Delta t f_{i,j}^{k-1}. \tag{10}$$

To solve the direct problem we use the alternating direction implicit method (ADIM). This method reduces the multidimensional problem to a several independent one-dimensional problems to solve. We write the Equation (10) in a following form:

$$(1 - \Delta t\delta_x)(1 - \Delta t\delta_y) U_{i,j}^k = U_{i,j}^{k-1} + \Delta t f_{i,j}^{k-1}. \tag{11}$$

ADIM is a iterative method and in the considered case we may define it in two steps:

1. First we solve the problem in the direction of the Ox axis (for the fixed y_j):

$$(1 - \Delta t\delta_x) U_{i,j}^* = U_{i,j}^{k-1} + \Delta t f_{i,j}^{k-1}. \tag{12}$$

   This way we get the intermediate solution $U_{i,j}^*$.
2. In the next step we solve the problem in the direction of Oy axis (for the fixed x_i):

$$(1 - \Delta t\delta_y) U_{i,j}^k = U_{i,j}^*. \tag{13}$$

In the considered problem the Dirichlet boundary conditions are zeros. Due to that, we may easily determine the boundary values for the solution U^* which are:

$$U_{0,j}^* = (1 - \Delta t\delta_y) U_{0,j}^k = 0, \qquad U_{N,j}^* = (1 - \Delta t\delta_y) U_{N,j}^k = 0.$$

We must write the differential schemes (12)–(13) in the matrix form, thereby obtaining the system of equations, solutions of which are the values of the function u at the grid points. More about the ADIM method, its stability and convergence may be found in the [13].

3.2. Minimalization of the Objective Function

The second necessary step to solving the considered inverse problem is the minimization of the functional (6). For this we use the real ant colony optimization (RealACO) algorithm [14], which is the swarm artificial intelligence algorithm. We describe it briefly below, and for that, use the following notation:

J—minimalized function (objective function), n—size of the problem,

nT—number of threads, $M = nT \cdot p$—number of ants in the population,

I—number of iterations, L—number of pheromone spots, q, ξ—algorithm parameters.

The pheromone spots mean the potential solutions. The ants modify these solutions while looking for better results. We now present the consecutive steps of the algorithm.

Start of the algorithm

1. Set the algorithm's input parameters: L, M, I, nT, q, ξ.
2. Generate L initial solutions serving as pheromone spots. Assign them to the T_0 set (which is the initial archive).
3. Compute the values of the objective function for all of the pheromone spots (parallel computing) and sort the archive T0 from the best solution to the worst.

Iteration process

4. Assign the probabilities to the pheromone spots according to the formula:

$$p_l = \frac{\omega_l}{\sum_{l=1}^{L} \omega_l} \quad l = 1, 2, \ldots, L,$$

where weights ω_l are connected to the l-th solution and are expressed by the formula:

$$\omega_l = \frac{1}{qL\sqrt{2\pi}} \cdot e^{\frac{-(l-1)^2}{2q^2L^2}}.$$

5. The ant must randomly chose the l-te solution with the probability p_l.
6. The ant must transform the j-th coordinate ($j = 1, 2, \ldots, n$) of the l-th solution s_j^l by sampling the neighborhood using the probability density function (in this case the Gauss function):

$$g(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{\frac{-(x-\mu)^2}{2\sigma^2}},$$

where $\mu = s_j^l$, $\sigma = \frac{\xi}{L-1} \sum_{p=1}^{L} |s_j^p - s_j^l|$.

7. The steps 5 and 6 must be repeated by every ant. This way, we get M new solutions (pheromone spots).
8. Partition the new solutions into nT groups. Compute the value of the objective function for each of the new solutions (parallel computing).
9. Add the new solutions to the archive T_i; sort them by the quality and reject the M worst solutions.
10. The steps 3–9 must be repeated I times.

Knowing the values of the parameters L, M and I, we can determine the number of the objective function calls during the algorithm execution. This number is $L + M \cdot I$. More about the method and its application for the inverse problems (particularly in the field of heat conductivity) can be found in [5,17–19].

4. Numerical Example

We now present the method described in the previous section by using it in a numerical example. We consider the Equation (1) with the homogenous Dirichlet boundary conditions. We also take under consideration the following data:

$$c = 900.0,\ \varrho = 2700.0,\ L_x = L_y = 1.0,\ T = 300.0,\ f(x, y) = 150{,}000(x-1)^4x^2(y-1)^4y^2,$$

$$
\begin{aligned}
d(u,x,y,t) = {} & -1{,}215{,}000{,}000(x-1)^4x^2(y-1)^4y^2 \\
& + 191{,}492(t-300)(x^{0.5} - 8x^{1.5} + 19.2x^{2.5} - 18.2857x^{3.5} + 6.09524x^{4.5})(y-1)^4y^2 \\
& + (t-300)\big(65{,}297.4(1-x)^{4.5} - 587{,}677(1-x)^{3.5}(0.0833333 + x) \\
& + 514{,}217(1-x)^{2.5}(0.025 + 0.166667x + x^2)\big)(y-1)^4y^2 + 191{,}492(t-300)(x-1)^4 \\
& x^2(1.y^{0.5} - 8.y^{1.5} + 19.2y^{2.5} - 18.2857y^{3.5} + 6.09524y^{4.5}) + 32{,}648.7(-300+t)(x-1)^4x^2 \\
& \big(2(1-y)^{4.5} - 18(1-y)^{3.5}(0.0833333 + y) + 15.75(1-y)^{2.5}(0.025 + 0.166667y + y^2)\big).
\end{aligned}
$$

The exact solution of such a problem for $\alpha = 1.5$ and $\lambda = 240$ is a function:

$$u(x,y,t) = 500(300-t)x^2y^2(x-1)^4(y-1)^4.$$

As was described in the Section 2 the reconstructed parameters of the considered model are order of the derivative α and the diffusion coefficient λ. The aim of this numerical example is to illustrate the method's accuracy. Therefore, the input data for the inverse problem were generated for the exact values $\alpha = 1.5$, $\lambda = 240.0$ using the grid of size $150 \times 150 \times 600$. For the algorithm for solving the inverse problem, the grid of size $100 \times 100 \times 300$ was used. Using a different grid size for generating the input data for the inverse problem and a different size for solving the inverse problem was necessary to avoid the occurrence of the so-called inverse crimes [20,21]. Additionally, in order to examine the stability of the algorithm, the input data were disturbed by the pseudo-random normal distribution errors of size 1% and 3%. Running time of the algorithm on a standard PC was about 14 h.

In this example we considered two cases of measurement point locations:

- Four points placed in the "corners" of the domain: $A(0.2, 0.8), B(0.8, 0.8), C(0.2, 0.2), D(0.8, 0.2)$; then we have $N_1 = 4$.
- Two points placed in the opposite "corners" of the domain: $B(0.8, 0.8), C(0.2, 0.2)$; then we have $N_1 = 2$.

The locations of the measurement points are presented in the Figure 1. For each configuration of points, we simulated the measurements every 1 s, 2 s and 4 s ($N_2 = 300, N_2 = 150, N_2 = 75$).

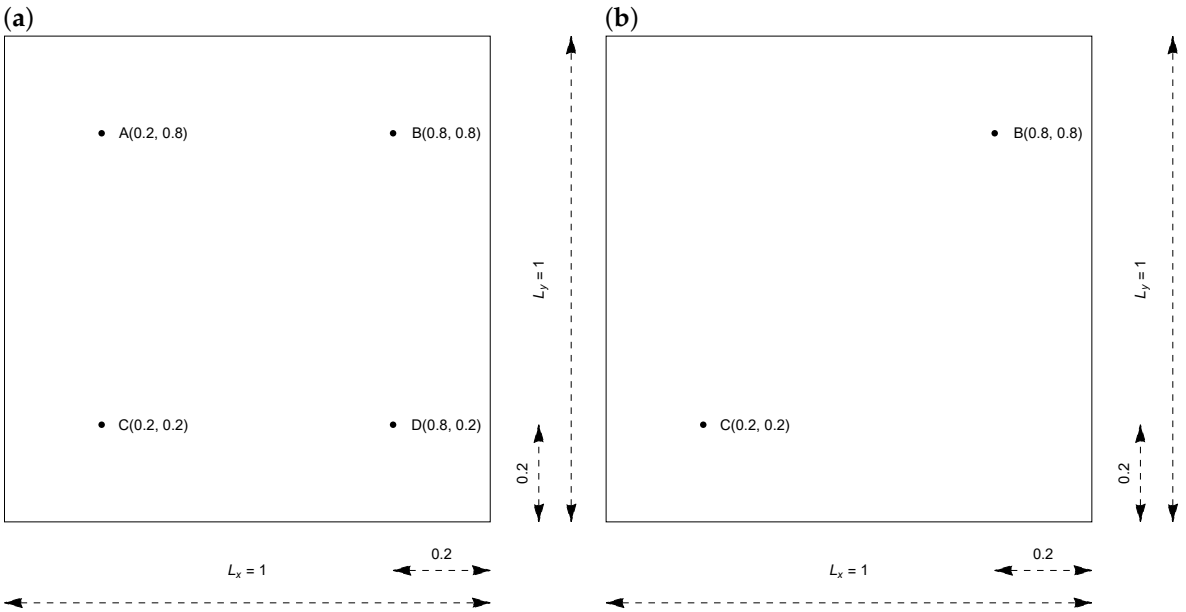

Figure 1. Locations of the measurement points in cases of (**a**) four measurement points; (**b**) two measurement points.

Tables 1 and 2 present the obtained results of the α and λ parameter reconstructions for various sizes of disturbance of the input data and different numbers of measurements taken. The presented algorithm is probabilistic, and therefore for each case it was executed five times. The tables include the values of standard deviation calculated for these executions.

Table 1. Results of calculations in the case of four measurement points ($\overline{\alpha}_i$—reconstructed value of order derivative, $\overline{\lambda}_i$—reconstructed value of diffusion coefficient, δ—the relative error of reconstruction, σ—standard deviation, J—the value of functional).

Measurements Intervals	Noise	$\overline{\alpha}$	$\delta_{\overline{\alpha}}$[%]	$\sigma_{\overline{\alpha}}$	$\overline{\lambda}$	$\delta_{\overline{\lambda}}$[%]	$\sigma_{\overline{\lambda}}$	J
1 s	0%	1.4944	0.38	0.0677	241.66	0.69	33.19	0.0135
	1%	1.5114	0.76	0.0502	233.87	2.55	21.77	1.7948
	3%	1.5083	0.56	0.0967	235.54	1.86	46.92	16.2992
2 s	0%	1.5181	1.21	0.0474	231.30	3.62	21.56	0.0243
	1%	1.5129	0.86	0.0588	232.90	2.95	27.43	0.8427
	3%	1.5094	0.63	0.0246	234.68	2.21	11.53	7.9230
4 s	0%	1.4937	0.41	0.0444	241.84	0.76	21.25	0.0021
	1%	1.4959	0.27	0.0626	240.91	0.38	29.82	0.4088
	3%	1.5480	3.21	0.0511	219.71	8.45	21.82	3.5613

Table 2. Results of calculations in the case of two measurement points ($\overline{\alpha}_i$—reconstructed value of order derivative, $\overline{\lambda}_i$—reconstructed value of diffusion coefficient, δ—the relative error of reconstruction, σ—standard deviation, J—the value of functional).

Measurements Intervals	Noise	$\overline{\alpha}$	$\delta_{\overline{\alpha}}$[%]	$\sigma_{\overline{\alpha}}$	$\overline{\lambda}$	$\delta_{\overline{\lambda}}$[%]	$\sigma_{\overline{\lambda}}$	J
1 s	0%	1.5115	0.77	0.1666	233.47	2.71	72.28	0.0974
	1%	1.4901	0.65	0.0793	243.52	1.46	38.56	1.7712
	3%	1.5084	0.56	0.1738	235.38	1.92	50.94	16.1719
2 s	0%	1.4492	3.38	0.1637	261.82	9.09	21.76	0.0191
	1%	1.4922	0.51	0.0502	242.45	1.02	21.35	0.8085
	3%	1.5066	0.44	0.0813	235.71	1.78	35.82	7.8638
4 s	0%	1.4773	1.51	0.0666	249.11	3.79	32.52	0.0007
	1%	1.4664	2.23	0.1769	254.01	5.83	43.95	0.4032
	3%	1.5796	5.31	0.0821	207.71	13.45	35.06	3.4968

In the case of four measurement points, the best solutions (the smallest errors) were received for measurements taken every 1 s. The errors of the order of the derivative α were no more than 1% and the biggest value of the error of reconstruction of the λ coefficient was gotten for the input data disturbed by 1% error. In each case the lower the value of the functional, the lower the disturbance of the input data. In the case of four measurement points, the worst solution we got was for the input data disturbed by the 3% error and measurements taken every 4 s. That was the consequence of, first of all, the biggest error of the input data, and second of all, the smallest number of measurements taken ($N_2 = 75$). The values of the standard deviation of the obtained results are more or less at a similar level. The value of the standard deviation was affected by the RealACO algorithm parameters and number of the algorithm calls. During some of the executions (when the measurements were taken every 4 s and the disturbance was 3%) the solution ended falling into a local minimum, which significantly affected the value of the standard deviation. The proper selection of the RealACO parameters and increasing the number of algorithm calls caused a decrease of the value of the standard deviation. In case of the two measurement points, we can observe that in most cases (but not in all of them) the obtained errors of the solutions are bigger than in the case of the four measurement points. The values of the functional are, however, at a similar level in both cases. By comparing the two tables, we can also observe that the values of the standard deviations are bigger in most cases for the four measurement points. In this case, increasing the number of measurement points should positively affect the obtained

results. For the obtained approximate values of the α and λ parameters, the absolute errors of the function u reconstruction were calculated for the measurement points and are presented in the Tables 3 and 4. In each case the errors were very small. The average values of the relative error in case of four measurement points for the 3% disturbance and measurements intervals 1 s, 2 s and 4 s were 0.31%, 0.51% and 6.03% accordingly. In case of two measurement points the errors were 0.34%, 1.77% and 10.14%, accordingly, for those measurements intervals. The higher values of the error were the consequence of the fact that the solution on the boundary of the considered area was equal to zero. The absolute errors in those cases were small and for the four measurement points did not exceed the values of 0.0141 on average and 0.0485 at maximum. For the two measurement points, in turn, these values do not exceed the 0.0245 average and 0.0759 maximum.

Table 3. The errors of the function u reconstruction for the measurement points in the case of four measurement points (Δ_{avg}—average absolute error, Δ_{max}—maximum average error).

Noise	**0%**	**1%**	**3%**	**0%**	**1%**	**3%**	**0%**	**1%**	**3%**
		co 1 s			**co 2 s**			**co 4 s**	
Δ_{avg}	0.0021	0.0024	0.0026	0.0042	0.0038	0.0022	0.0021	0.0018	0.0141
Δ_{max}	0.0091	0.0269	0.0148	0.0223	0.0367	0.0258	0.0065	0.0074	0.0485

Table 4. The errors of the function u reconstruction for the measurement points in the case of two measurement points (Δ_{avg}—average absolute value, Δ_{max}—maximum absolute value).

Noise	**0%**	**1%**	**3%**	**0%**	**1%**	**3%**	**0%**	**1%**	**3%**
		co 1 s			**co 2 s**			**co 4 s**	
Δ_{avg}	0.0071	0.0027	0.0037	0.0064	0.0018	0.0053	0.0019	0.0035	0.0245
Δ_{max}	0.0365	0.0073	0.0176	0.0163	0.0071	0.0299	0.0041	0.0087	0.0759

Figure 2 presents the absolute errors of the solution u reconstruction at the measurement points B and C in cases of two measurement points, the smallest number of measurements taken and various sizes of the input data disturbance. In case of the accurate input data and 1% disturbance, the errors of the solution reconstruction for the measurement points were minimal. The errors were slightly higher with 3% disturbance.

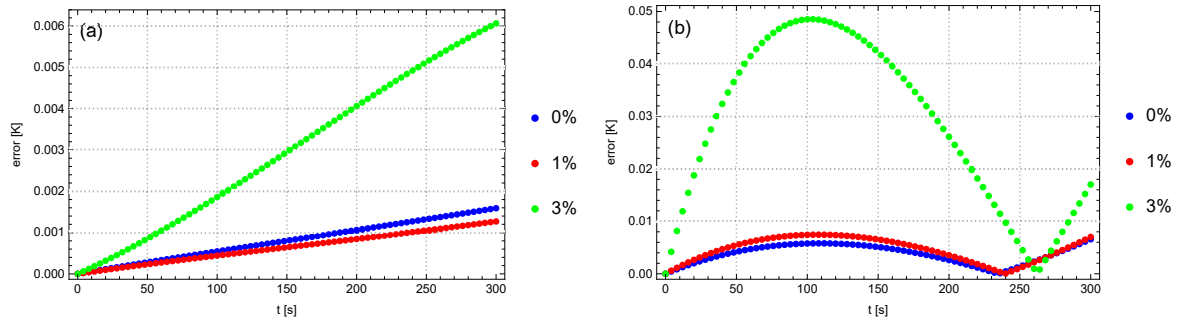

Figure 2. The absolute errors of the solution u reconstruction for the measurement points B (**a**) and C (**b**) in the case of the measurements taken every 4 s.

5. Conclusions

The aim of the paper was to reconstruct the order of the derivative and the diffusion coefficient while the values of the solution in the selected points of the domain were known. The obtained results show that for the proper number of the measurement points and a reasonable size of the input data disturbance, the presented algorithm allows one to reconstruct the parameters of interest well. Moreover, the absolute values of the considered solution reconstruction are small. Consecutive

executions of the algorithm gave similar results, which was proven by the values of the standard deviation obtained.

In the future we plan to try to improve the algorithm by combining the heuristic algorithm used with the deterministic algorithm in order to speed up the procedure. The second research area will be the addition of the regulatory element, the purpose of which will be the improvement of the algorithm's operation in cases of larger input data disturbances and smaller numbers of measurements taken.

Author Contributions: Conceptualization, D.S., R.B. and A.C.; methodology, D.S.; software, R.B.; validation, A.C.; investigation, R.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tenreiro Machado, J.A.; Silva, M.F.; Barbosa, R.S.; Jesus, I.S.; Reis, C.M.; Marcos, M.G.; Galhano, A.F. Some applications of fractional calculus in engineering. *Math. Probl. Eng.* **2010**, *2010*, 639801. [CrossRef]
2. Carpinteri, A.; Mainardi, F. *Fractal and Fractional Calculus in Continuum Mechanics*; Springer: New York, NY, USA, 1997.
3. Mathieu, B.; Melchior, B.; Oustaloup, A.; Ceyral, C. Fractional differentiation for edge detection. *Signal Process.* **2003**, *83*, 2421–2432. [CrossRef]
4. Voller, V.R. Anomalous heat transfer: Examples, fundamentals, and fractional calculus models. *Adv. Heat Transf.* **2018**, *50*, 338–380.
5. Brociek, R.; Słota, D. Application of Real Ant Colony Optimization algorithm to solve space and time fractional heat conduction inverse problem. In *International Conference on Information and Software Technologies*; Springer: Cham, Switzerland, 2017; Volume 46, pp. 171–182.
6. Obrączka, A.; Kowalski, J. Modeling the distribution of heat in the ceramic materials using fractional differential equations. In *Materiały XV Jubileuszowego Sympozjum "Podstawowe Problemy Energoelektroniki, Elektromechaniki i Mechatroniki", PPEEm*; Szczygieł, M., Ed.; Archiwum Konferencji PTETiS; Komitet Organizacyjny Sympozjum PPEE i Seminarium BSE: Gliwice, Poland, 2012; Volume 32, pp. 132–133. (In Polish)
7. Brociek, R.; Słota, D.; Król, M.; Matula, G.; Kwaśny, W. Comparison of mathematical models with fractional derivative for the heat conduction inverse problem based on the measurements of temperature in porous aluminum. *Int. J. Heat Mass Transf.* **2019**, *143*, 118440. [CrossRef]
8. Zhuag, Q.; Yu, B.; Jiang, X. An inverse problem of parameter estimation for time-fractional heat conduction in a composite medium using carbon–carbon experimental data. *Physical B* **2015**, *456*, 9–15. [CrossRef]
9. Majka, Ł. Fractional derivative approach in modeling of a nonlinear coil for ferroresonance analyses. In *Non-Integer Order Calculus and Its Applications*; Ostalczyk, P., Sankowski, D., Nowakowski, J., Eds.; Lecture Notes in Electrical Engineering; Springer: Cham, Switzerland, 2019; Volume 496, pp. 135–147.
10. Sowa, M. A harmonic balance methodology for circuits with fractional and nonlinear elements. *Circuits Syst. Signal Process.* **2018**, *37*, 4695–4727. [CrossRef]
11. Jday, F.; Mdimagh, R. Uniqueness result for a fractional diffusion coeffcient identification problem. *Bound. Value Probl.* **2019**, *2019*, 170. [CrossRef]
12. Ragusa, M.A.; Scapellato, A. Mixed Morrey spaces and their applications to partial differential equations. *Nonlinear Anal. Theory Methods Appl.* **2017**, *151*, 51–65. [CrossRef]
13. Liu, F.; Chen, S.; Turner, I.; Burrage, K.; Anh, V. Numerical simulation for two-dimensional Riesz space fractional diffusion equations with a nonlinear reaction term. *Cent. Eur. J. Phys.* **2013**, *11*, 1221–1232. [CrossRef]
14. Socha, K.; Dorigo, M. Ant colony optimization for continuous domains. *Eur. J. Oper. Res.* **2008**, *185*, 1155–1173. [CrossRef]
15. Meerschaert, M.; Tadjeran, C. Finite difference approximations for fractional advection–dispersion flow equations. *J. Comput. Appl. Math.* **2004**, *172*, 65–77. [CrossRef]
16. Oldham, K.B.; Spanier, J. *The Fractional Calculus*; Academic Press: New York, NY, USA; London, UK, 1974.

17. Brociek, R.; Słota, D.; Król, M.; Matula, G.; Kwaśny, W. Modeling of heat distribution in porous aluminum using fractional differential equation. *Fractal Fract.* **2017**, *1*, 17. [CrossRef]
18. Tam, J.H.; Ong, Z.C.; Ismail, Z.; Ang, B.C.; Khoo, S.Y.; Li, W.L. Inverse identification of elastic properties of composite materials using hybrid GA-ACO-PSO algorithm. *Inverse Probl. Sci. Eng.* **2018**, *26*, 1432–1463. [CrossRef]
19. Hetmaniok, E. Solution of the inverse problem in solidification of binary alloy by applying the ACO algorithm. *Inverse Probl. Sci. Eng.* **2016**, *24*, 889–900. [CrossRef]
20. Kaipio, J.; Somersalo, E. *Statistical and Computational Inverse Problems*; Springer: New York, NY, USA, 2005.
21. Kaipio, J.; Somersalo, E. Statistical inverse problems: Discretization, model reduction and inverse crimes. *J. Comput. Appl. Math.* **2007**, *198*, 493–504. [CrossRef]