*Mathematical and Computational Applications*

MDPI

*Article*

# Application of Generalized Finite Difference Method and Radial Basis Function Neural Networks in Solving Inverse Problems of Surface Anomalous Diffusion

**Luchuan Shi * and Qiang Xi ***

College of Mechanics and Engineering Science, Hohai University, Nanjing 211110, China
* Correspondence: 15295779559@163.com (L.S.); xiqiang1994@126.com (Q.X.)

**Abstract:** In this study, a new hybrid method based on the generalized finite difference method (GFDM) and radial basis function (RBF) neural network technologies is developed to solve the inverse problems of surface anomalous diffusion. Specifically, the GFDM is utilized to compute the time-fractional derivative model on the surface, whereas RBF neural networks are employed to invert the diffusion coefficient, source term coefficient, and the fractional order within the anomalous diffusion equation governing the surface. The results of four examples show that for the three parameters of diffusion coefficient, source term coefficient, and fractional order, the errors of inversion results are in the order of $10^{-2}$ under different conditions. Therefore, this method can obtain the required parameters quickly and accurately under different conditions.

**Keywords:** anomalous diffusion; time-fractional derivative; generalized finite difference method; extrinsic; radial basis function neural networks

check for updates

Academic Editor: Gianluigi Rozza

Received: 23 November 2024
Revised: 30 December 2024
Accepted: 7 January 2025
Published: 9 January 2025

**Citation:** Shi, L.; Xi, Q. Application of Generalized Finite Difference Method and Radial Basis Function Neural Networks in Solving Inverse Problems of Surface Anomalous Diffusion. *Math. Comput. Appl.* **2025**, *30*, 7. https://doi.org/10.3390/mca30010007

## 1. Introduction

Since the 1980s, advancements in science and technology have led to the gradual revelation of a series of physical phenomena known as anomalous diffusion [1–4]. These phenomena are complex and difficult to fully explain or accurately model using traditional integer-order diffusion systems. Anomalous diffusion has been demonstrated in various fields, including porous media mechanics [5], non-Newtonian fluid mechanics [6], viscoelastic mechanics [7], and soft matter mechanics [8,9]. The fractional diffusion model [10] is one of the most common models to describe anomalous diffusion behavior. Its advantage lies in its simpler parameters and more convenient numerical calculations. The non-local properties of fractional derivatives [11] enable them to describe complex non-Markov systems [12].

Due to the non-local characteristics and complexity of fractional differential equations [11], obtaining analytical solutions for these equations is extremely challenging. Therefore, developing efficient and accurate numerical algorithms to solve fractional differential equations is crucial. In recent years, numerous meshless methods have been proposed. These methods replace the traditional grid with nodes, thereby avoiding the complicated grid generation process. Meshless methods include the fundamental solution method [13], Trefftz method [14], singular boundary method [15], generalized finite difference method [16,17], etc. The GFDM is a relatively new localized meshless method. First proposed by T. Liszka and J. Orkisz in the 1980s [18], it has been continuously improved and refined until a relatively complete version was proposed by Benito et al. in 2001 [19]. The GFDM has been successfully applied in many fields due to its ability to overcome the constraints of traditional regular grids and its capacity to handle complex boundary conditions without the need for extensive preprocessing. Moreover, its localized configuration makes it particularly suitable for dealing with large-scale problems, including obstacle problems [20], inverse problems [21], slogging phenomena [22], etc.

In recent years, with the rapid development of neural network technology, using neural networks to solve inverse problems of anomalous diffusion equations has become increasingly prevalent [23–26]. The RBF neural network is a relatively advanced type of three-layer feedforward neural network. Compared with back-propagation (BP) neural networks, RBF neural networks exhibit superior performance in classification accuracy, learning speed, and approximation ability [27]. An RBF neural network is a kind of neural network with local approximation performance, so an RBF neural network is faster in inversion than other neural networks. In addition, an RBF neural network can accurately invert the results when dealing with the problems in this paper, so an RBF neural network is adopted to invert the parameters.

Compared with other work, this is the first time the inverse surface diffusion problem is solved by combining GFDM and RBF neural networks. In this paper, the time-fractional derivative model is introduced to define the anomalous diffusion process on the surface. In the numerical implementation, the GFDM and the extrinsic processing technique of the surface partial differential equation are employed for the discrete solution. Subsequently, RBF neural networks are introduced into the inverse problem of inverting diffusion equation parameters to obtain the required equation parameters. Specifically, a training database is established by solving the proposed time-fractional diffusion equation using the GFDM.

## 2. The GFDM in Solving Anomalous Diffusion on the Surface

### *2.1. Time-Fractional Derivative Model on the Surface*

To address the intricate issue of anomalous diffusion on a surface, the time-fractional derivative model is introduced. Given that $S$ represents a closed and smooth surface within a three-dimensional space, the fractional diffusion equation formulated on this surface can be articulated as follows:

$$\frac{\partial^{\alpha} u(\mathbf{x},t)}{\partial t^{\alpha}} = \left(D\Delta_s + \vec{v}\cdot\nabla_s - \lambda\right)u(\mathbf{x},t) + Q(\mathbf{x},t), 0<\alpha<1, \mathbf{x}\in S, t\in(0,T), \tag{1}$$

and the following initial condition is given as

$$u(\mathbf{x},0) = u_0(\mathbf{x}), \mathbf{x}\in S, \tag{2}$$

where $D$, $\lambda$, and $\vec{v}$ represent diffusion coefficient, reaction coefficient, and velocity vector, respectively. $Q(\mathbf{x},t)$ and $u_0(\mathbf{x})$ are known functions, $T$ is the total time, $\Delta_s$ and $\nabla_s$ represent the Laplace–Beltrami and surface gradient operators, respectively [28]. $\partial^{\alpha}/\partial t^{\alpha}$ is the Caputo time-fractional of order $\alpha$ with respect to $t$ [29], which is defined by the following equation:

$$\frac{\partial^{\alpha} u(\mathbf{x},t)}{\partial t^{\alpha}} = \frac{1}{\Gamma(1-\alpha)}\int_0^t \frac{\partial u(\mathbf{x},\eta)}{\partial \eta}\frac{d\eta}{(t-\eta)^{\alpha}}, 0<\alpha<1, \tag{3}$$

where $\Gamma$ is the Gamma function. By using the standard finite difference method, the time-fractional derivative model can be discretized as [30]

$$\begin{cases} \begin{array}{c} D\theta\Delta_s u^{k+1} + \theta\vec{v}\cdot\nabla_s u^{k+1} - (\lambda\theta+\alpha_0)u^{k+1} = \\ -\theta Q^{k+1} - (1-\theta)\left(D\Delta_s u^k + \vec{v}\cdot\nabla_s u^k + Q^k - \lambda u^k\right) \\ -\alpha_0 u^k + \sum_{j=1}^{k}\alpha_j b_j\left(u^{k-j+1} - u^{k-j}\right), k\geq 1, \end{array} \\ -\theta Q^1 - (1-\theta)\left(D\Delta_s u^0 + \vec{v}\cdot\nabla_s u^0 + Q^0 - \lambda u^0\right) - \alpha_0 u^0, k=0. \end{cases} \tag{4}$$

where $\alpha_j$ and $b_j$ are known coefficients, and $\theta$ is the time integral constant. $a_0 = \frac{dt^{-\alpha}}{\Gamma(1+\alpha)}$, where $dt$ is the time step.

### 2.2. Surface Differential Operators

Suppose $S$ is a closed and smooth surface defined in three-dimensional space, and for an arbitrary point $\mathbf{x}_i = [x_i, y_i, z_i]^T$ on this surface, the normal vector at this point is expressed as $\mathbf{n} = \mathbf{n}(\mathbf{x}) = \left[n_{x_i}, n_{y_i}, n_{z_i}\right]^T$. Then, the surface gradient operator at x can be formulated as follows:

$$\nabla_s = \mathbf{P}\nabla = \left(\mathbf{I}_3 - \mathbf{n}\mathbf{n}^T\right)\nabla, \tag{5}$$

where $\mathbf{I}_d$ is a $d \times d$ identity matrix and $\mathbf{P}$ represents the projection matrix. The Laplace–Beltrami operator at $\mathbf{x}$ can be expressed as

$$\Delta_s = \nabla_s \cdot \nabla_s. \tag{6}$$

By introducing the extrinsic technique, the analytical relationship between the operators defined in tangent space and the operators defined in Euclidean space can be derived from the normal vector $\mathbf{n}$ corresponding to all points $\mathbf{x}_i$ selected on the surface [26]:

$$\nabla_s u = \nabla u - \mathbf{n}\partial_{\mathbf{n}} u, \Delta_s u = \Delta u - H_s \partial_{\mathbf{n}} u - \partial_{\mathbf{n}}^{(2)} u. \tag{7}$$

where $\partial_{\mathbf{n}} u = \mathbf{n}^T \nabla u$, $\partial_{\mathbf{n}}^{(2)} u = \mathbf{n}^T J(\nabla u)\mathbf{n}$, and $H_s(x) = trace\left(J(\mathbf{n})\left(\mathbf{I}_d - \mathbf{n}\mathbf{n}^T\right)\right)$, and $J$ represents the jacobian operators in Euclidean space.

### 2.3. GFDM for Time-Fractional Derivative Model on the Surface

Similar to traditional finite difference methods, the generalized finite difference method constructs numerical differential formulas to approximate partial derivative terms of partial differential equations. The general formula can be expressed as follows:

$$D_u = WU, \tag{8}$$

where the partial derivative vector $D_u$, the weighting matrix $W$, and the solution vector $U$ are, respectively, defined as

$$D_u = \left[\frac{\partial u_i}{\partial x_k}, \quad \ldots\ldots \quad , \frac{\partial^2 u_i}{\partial x_k \partial x_{k''}}\right]_{1\times n_v}^T, k = 1, 2, \ldots\ldots, d; k'' = k, k+1, \ldots\ldots, d, \tag{9}$$

$$U = [u_1, u_2, \ldots\ldots, u_N]^T, \tag{10}$$

$$W = \begin{bmatrix} \omega_1^{x_k,i} & \cdots & \omega_N^{x_k,i} \\ \vdots & \ddots & \vdots \\ \omega_1^{x_k x_{k''},i} & \cdots & \omega_N^{x_k x_{k''},i} \end{bmatrix}_{n_v \times N}, k = 1, 2, \ldots, d; k'' = k, k+1, \ldots, d. \tag{11}$$

where $d$ represents the dimension, $u_i (i = 1, 2, \ldots, N)$ is the solution of the discretized node $x_i$, $x_i = (x_{i1}, x_{i2}, \ldots, x_{id}) \in \Xi \subset R^d$, and $n_v = 2d + C_d^2$.

The GFDM, which is defined in Euclidean space, can be directly employed for approximating surface differential operators with extrinsic technology. Figure 1 shows a sketch of discrete points of the computational domain and neighborhood selection using the meshless generalized finite difference method, and the equation for the GFDM is provided as follows:

$$\left\{\begin{array}{l}
\dfrac{\partial u_{x^{\xi}}}{\partial x} = W_{x_i}u_i + \sum_{j=1}^{N} W_{x_j}u_j, \\
\dfrac{\partial u_{y^{\xi}}}{\partial y} = W_{y_i}u_i + \sum_{j=1}^{N} W_{y_j}u_j, \\
\dfrac{\partial u_{z^{\xi}}}{\partial z} = W_{z_i}u_i + \sum_{j=1}^{N} W_{z_j}u_j, \\
\dfrac{\partial u_{x^{\xi}}}{\partial x^2} = W_{x_ix_i}u_i + \sum_{j=1}^{N} W_{x_jx_j}u_j, \\
\dfrac{\partial u_{y^{\xi}}}{\partial y^2} = W_{y_iy_i}u_i + \sum_{j=1}^{N} W_{y_jy_j}u_j, \\
\dfrac{\partial u_{z^{\xi}}}{\partial z^2} = W_{z_iz_i}u_i + \sum_{j=1}^{N} W_{z_jz_j}u_j, \\
\dfrac{\partial u_{x^{\xi}y^{\xi}}}{\partial x\partial y} = W_{x_iy_i}u_i + \sum_{j=1}^{N} W_{x_jy_j}u_j, \\
\dfrac{\partial u_{x^{\xi}z^{\xi}}}{\partial x\partial z} = W_{x_iz_i}u_i + \sum_{j=1}^{N} W_{x_jz_j}u_j, \\
\dfrac{\partial u_{y^{\xi}z^{\xi}}}{\partial y\partial z} = W_{y_iz_i}u_i + \sum_{j=1}^{N} W_{y_jz_j}u_j.
\end{array}\right. \tag{12}$$

where $x,\ y,\ z$ are the Euclidian coordinates. Thus, the unified equation can be derived as

$$Lu|_{x^{\xi}} = W^{\xi}U^{\xi}, \tag{13}$$

where $L$ represents any differential operator, $W^{\xi}$ represents the row vector of $1 \times N_s$, and $U^{\xi}$ represents the column vector of the local solution of $N_s \times 1$. If $S$ is a smooth closed surface, then $\nabla_s u$ can be expressed as

$$\nabla_s u = \mathbf{P} \cdot \nabla u = \mathbf{P}\begin{bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial u}{\partial y} \\ \dfrac{\partial u}{\partial z} \end{bmatrix} = \begin{bmatrix} p_{11}\dfrac{\partial u}{\partial x} + p_{12}\dfrac{\partial u}{\partial y} + p_{13}\dfrac{\partial u}{\partial z} \\ p_{21}\dfrac{\partial u}{\partial x} + p_{22}\dfrac{\partial u}{\partial y} + p_{23}\dfrac{\partial u}{\partial z} \\ p_{31}\dfrac{\partial u}{\partial x} + p_{32}\dfrac{\partial u}{\partial y} + p_{33}\dfrac{\partial u}{\partial z} \end{bmatrix}. \tag{14}$$

$\Delta_s u$ can be expressed as

$$\begin{aligned}
\Delta_s u|_{x^{\xi}} = {} & \left(W^{\xi}_{x_1x_1} + W^{\xi}_{x_2x_2} + W^{\xi}_{x_3x_3}\right)U^{\xi} - H_S\left(x^{\xi}\right)\left[n^{\xi}_{x_1}n^{\xi}_{x_2}n^{\xi}_{x_3}\right]\begin{bmatrix} W^{\xi}_{x_1} \\ W^{\xi}_{x_2} \\ W^{\xi}_{x_3} \end{bmatrix} U^{\xi} \\
& -\left[n^{\xi}_{x_1}n^{\xi}_{x_2}n^{\xi}_{x_3}\right]\begin{bmatrix} W^{\xi}_{x_1x_1} & W^{\xi}_{x_1x_2} & W^{\xi}_{x_1x_3} \\ W^{\xi}_{x_2x_1} & W^{\xi}_{x_2x_2} & W^{\xi}_{x_2x_3} \\ W^{\xi}_{x_3x_1} & W^{\xi}_{x_3x_2} & W^{\xi}_{x_3x_3} \end{bmatrix}\begin{bmatrix} n^{\xi}_{x_1} \times I_N \\ n^{\xi}_{x_2} \times I_N \\ n^{\xi}_{x_3} \times I_N \end{bmatrix} U^{\xi}.
\end{aligned} \tag{15}$$

By using the GFDM and substituting Equations (14) and (15) into Equation (4), the time-fractional derivative model can be discretized as

$$\begin{aligned}
& D\theta W_{\Delta s}u^{k+1} + \theta\vec{v}\cdot\begin{bmatrix} W^{x}_{\nabla s}u^{k+1} \\ W^{y}_{\nabla s}u^{k+1} \\ W^{z}_{\nabla s}u^{k+1} \end{bmatrix} - (\lambda\theta + \alpha_0)u^{k+1} = \\
& \begin{cases}
-\theta Q^{k+1} - (1-\theta)\left(DW_{\Delta s}u^{k} + \vec{v}\cdot\begin{bmatrix} W^{x}_{\nabla s}u^{k} \\ W^{y}_{\nabla s}u^{k} \\ W^{z}_{\nabla s}u^{k} \end{bmatrix} + Q^{k} - \lambda u^{k}\right) \\
\qquad -\alpha_0 u^{k} + \sum_{j=1}^{k}\alpha_j b_j\left(u^{k-j+1} - u^{k-j}\right), k \geq 1, \\
-\theta Q^{1} - (1-\theta)\left(DW_{\Delta s}u^{0} + \vec{v}\cdot\begin{bmatrix} W^{x}_{\nabla s}u^{0} \\ W^{y}_{\nabla s}u^{0} \\ W^{z}_{\nabla s}u^{0} \end{bmatrix} + Q^{0} - \lambda u^{0}\right) - \alpha_0 u^{0}, k = 0.
\end{cases}
\end{aligned} \tag{16}$$

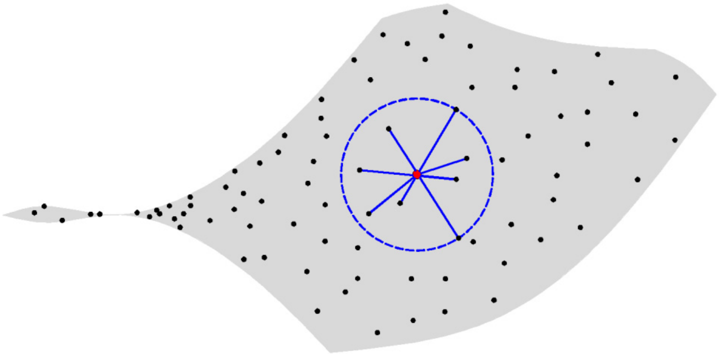


**Figure 1.** A sketch of discrete points of the computational domain and neighborhood selection using the meshless generalized finite difference method.

## 3. RBF Neural Networks

As shown in Figure 2, RBF neural networks, consisting of an input layer, one hidden layer, and an output layer, are one of the current famous artificial neural networks.

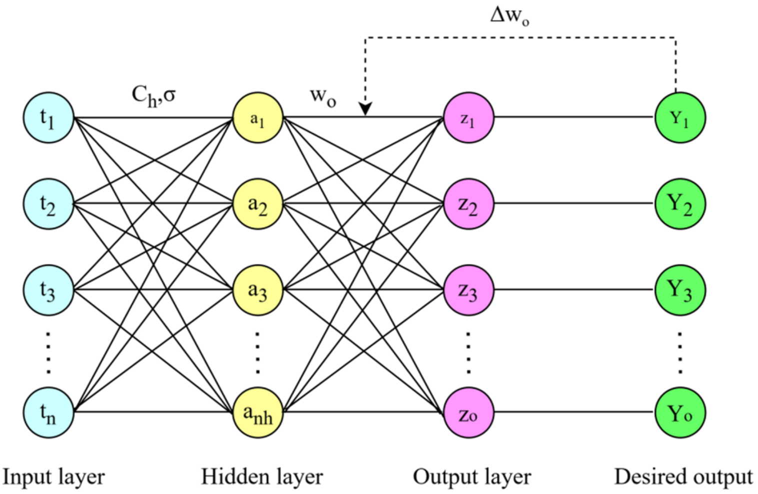


**Figure 2.** Structure of RBF neural networks.

The training process comprises two pivotal steps: initially, employing the K-means clustering algorithm to determine the centers of the basis functions from the training dataset; subsequently, optimizing the weights connecting the hidden layer to the output layer by minimizing the objective function. This training methodology enables the RBF neural networks to adapt more rapidly and accurately to diverse datasets and task requirements.

The Gaussian radial basis function is selected as the activation function, and the equation is given as follows:

$$\varphi(r) = e^{-\frac{r^2}{2\sigma^2}}, \tag{17}$$

The output of the $h$-th hidden neuron is given as

$$a_h = exp\left(-\frac{1}{2\sigma^2}\|\mathbf{C}_h - \mathbf{t}\|^2\right), \tag{18}$$

where $\mathbf{t} = (t_1, t_2, \cdots, t_n)^T$ is the input vector of the coefficients of the diffusion system equation from each detection point. The detection point will be selected from the GFDM nodes, and the input vector will be the spread value at that point in time. $\mathbf{C}_h$ and $\sigma$ are the neuron's center and the center spread parameter, respectively.

For the $o$-th neuron, the output $z_o$ is

$$z_o = \sum_{h=1}^{nh} w_o a_h + \theta_o, \tag{19}$$

where $nh$ is the number of neurons in the hidden layer. $w_o$ and $\theta_o$ are the weight and threshold connecting the neurons in the hidden layer to the neurons in the output layer.

The objective function is given as

$$J = \sum_{o=1}^{no} \|z_o - Y_o\|^2, \tag{20}$$

The weights are modified by moving in the opposite direction to the gradient of the target function.

The adjustment formula is given as

$$\Delta w_o = -\mu \frac{\partial J}{\partial w} = \mu(z_o - Y_o) exp\left(-\frac{1}{2\sigma^2}\|\mathbf{C}_h - \mathbf{t}\|^2\right). \tag{21}$$

where $\Delta w_o$ represents the weight increments, and the constant $\mu$ represents the learning rate.

## 4. Numerical Results and Discussion

In this section, we combine the GFDM with the RBF neural networks to establish a parameter inversion model for the diffusion equation, based on the results of surface anomalous diffusion. Given that the RBF neural networks demonstrate higher accuracy than other artificial neural networks in parameter inversion, we verify the effectiveness and precision of the RBF neural networks in inverting diffusion coefficients based on surface anomalous diffusion results through the following examples.

In the numerical implementation, the neural network program is executed by MATLAB software R2016b through the 'newrb' function in the neural networks' toolbox. The RBF neural networks iteratively create an RBF network using the function 'newrb', the target mean square error is set to 0.01, the radial basis function distribution coefficient is set to 1, the maximum number of neurons is set to 5000, and the number of neurons to be added between each display is set to 25. The optimization method is based on gradient descent.

In the parameter identification stage of the diffusion equation, the simulated diffusion results with an unknown diffusion coefficient at the observation point are input into the trained regression neural networks, and the corresponding parameters can be obtained. Different parameters are randomly generated in the detection domain, and the parameters are brought into the fractional diffusion equation on the surface after the GFDM. A large number of corresponding diffusion results and parameters are calculated as training samples for neural networks to learn. By constantly training and optimizing the weight parameters of the neural networks, a complex nonlinear mapping relationship between the features and labels of the training samples is constructed, and the trained neural networks can be used as a detection tool for the corresponding diffusion results and parameters.

The inversion error of diffusion equation parameters is given as

$$E = \frac{\left|X_{true} - X_{pred}\right|}{X_{true}}\%. \tag{22}$$

where $E$ is the error value, $X$ is the target parameter, $X_{true}$ is the true value of the target parameter, and $X_{pred}$ is the inversion value of the target parameter.

**Example 1.**

Firstly, the inversion of the diffusion coefficient in the diffusion equation in the presence of analytical solutions is considered, and the GFDM approximation is used to invert the diffusion coefficient D. The fractional diffusion equation on a closed smooth surface in three-dimensional space is defined as Equation (1), where $D$ is the diffusion coefficient, $\alpha = \frac{1}{2}$, $Q(\boldsymbol{x}, t) = 0$, and the analytical solution is given as

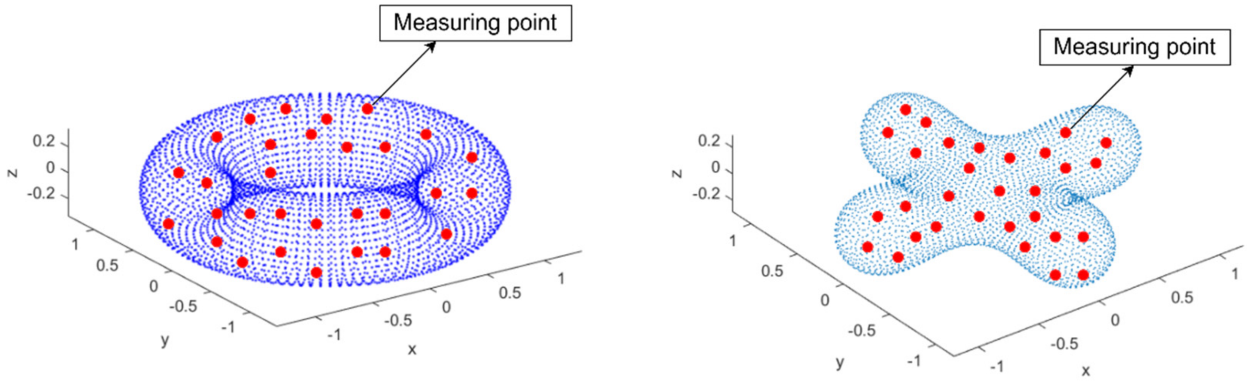
$$u(t, \mathbf{x}) = sinx \cdot cosy \cdot tanhz \cdot t, \tag{23}$$

In order to verify the accuracy, convergence, and stability of the GFDM for precisely capturing the anomalous diffusion phenomenon on the surface, Root Mean Square Error (RMSE) is introduced to measure it, and its equation is given as follows

$$RMSE(u_i(\mathbf{x}, t)) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left[u\left(\mathbf{x}_j, t\right) - \widetilde{u}\left(\mathbf{x}_j, t\right)\right]^2}. \tag{24}$$

where $N$ represents the total number of discrete points of the surface, $\widetilde{u}\left(\boldsymbol{x}_j, t\right)$ represents the numerical solution, and $u\left(\boldsymbol{x}_j, t\right)$ represents the analytical solution.

At the same time, RBF neural networks and BP neural networks are introduced to invert the parameters, so as to compare the advantages and disadvantages of the two methods in dealing with such problems. The hyperparameter settings are the same for both methods. There is no diffusion at the initial moment. From the initial moment, all the cloth points on the surface of the entire surface begin to diffuse at the same time. The time step is 0.01 s. Different diffusion coefficients are randomly generated in the detection domain, a point is randomly selected among the distribution points of the surface, and 30 points are randomly selected on the surface as observation points. The simulated diffusion results corresponding to diffusion coefficient $D$ at the observation point of 0.1 s are taken as the input feature $X$ of the neural networks and diffusion coefficient $D$ is taken as the output $Y$, so that a sample $(X, Y)$ can be constructed. For two different shapes, torus and cross, the number of samples for each shape training is 1000 sets of data. Each set of data contains the diffusion values of thirty detection points at different times. Then, 70% of the samples are divided into a study set and 30% of the samples are divided into verification sets to evaluate the generalization ability of the neural networks. The number of samples for testing is four sets of data, and the test set is not involved in the training of the neural networks. Figure 3 shows the distribution of distribution points and measuring points when the surface is torus and ellipsoid. Table 1 shows the RMSE between the GFDM approximation and the analytic solution on different surfaces at $t = 0.1$. Tables 2 and 3, respectively, show the inversion results and errors of diffusion coefficients when the surface is torus and cross.



(**a**) Torus (**b**) Cross

**Figure 3.** The distribution of distribution and measurement points of surfaces: (**a**) torus and (**b**) cross.

**Table 1.** RMSE between numerical solution and analytical solution on different surfaces.

| Shape | Sphere | Torus | Cross | RBC |
|---|---|---|---|---|
| N | 2500 | 4000 | 3996 | 4096 |
| RMSE | $2.73 \times 10^{-3}$ | $1.16 \times 10^{-3}$ | $2.50 \times 10^{-2}$ | $3.46 \times 10^{-3}$ |

**Table 2.** Inversion results, errors, and CPU time of RBFNN and BPNN with torus surface.

| $D_{true}$ | $D_{pred}$ | | $E$ (%) | | CPU Time (s) | |
|---|---|---|---|---|---|---|
| | RBFNN | BPNN | RBFNN | BPNN | RBFNN | BPNN |
| 0.2550 | 0.2552 | 0.2553 | 0.08 | 0.12 | 0.1 | 300 |
| 0.5050 | 0.5053 | 0.5054 | 0.06 | 0.08 | 0.1 | 305 |
| 0.7550 | 0.7526 | 0.7556 | 0.31 | 0.08 | 0.1 | 310 |
| 1.0050 | 1.0060 | 1.0061 | 0.10 | 0.11 | 0.1 | 306 |

**Table 3.** Inversion results, errors, and CPU time of RBFNN and BPNN with cross surface.

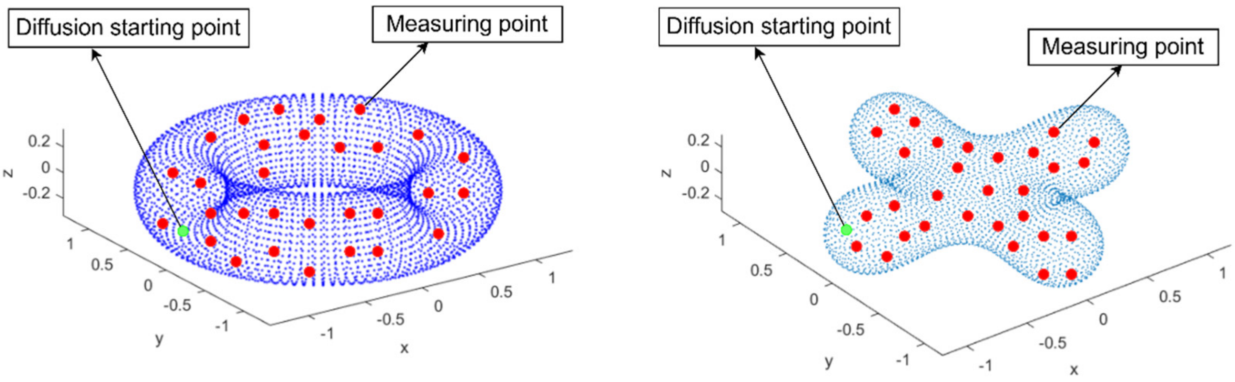
| $D_{true}$ | $D_{pred}$ | | $E$ (%) | | CPU Time (s) | |
|---|---|---|---|---|---|---|
| | RBFNN | BPNN | RBFNN | BPNN | RBFNN | BPNN |
| 0.2550 | 0.2538 | 0.2512 | 0.47 | 1.49 | 0.1 | 420 |
| 0.5050 | 0.5084 | 0.0535 | 0.67 | 0.51 | 0.1 | 415 |
| 0.7550 | 0.7536 | 0.7513 | 0.19 | 0.92 | 0.1 | 455 |
| 1.0050 | 1.0044 | 1.0094 | 0.06 | 0.35 | 0.1 | 435 |

From Table 1, it can be observed that for different models, the RMSE is in the order of $10^{-2}$. The numerical results clearly demonstrate that the GFDM can precisely capture the anomalous diffusion phenomenon on the surface and accurately approximate analytical solutions. However, the limitation of GFDM is that it is difficult to obtain accurate results when it is used to deal with non-smooth, non-closed, and asymmetrical surface models. For future research, more widely used numerical methods may be needed to solve such problems, such as finite element methods.

From Tables 2 and 3, it can be observed that for both methods, the predicted value of diffusion coefficient D is almost consistent with the actual value on different models. The error is in the order of $10^{-3}$. However, the inversion speed of an RBF neural network is significantly faster than that of a BP neural network. Therefore, the proposed model can achieve the fast and accurate inversion of diffusion coefficient D in the case of analytic solutions.

**Example 2.**

In this case, there is no analytical solution to the equation, and the GFDM approximation is used to invert diffusion coefficient D. A total of 30 points on the surface are ramdonly selected as observation points. The time step is 0.01 s. The simulated diffusion results corresponding to diffusion coefficient D at the observation point of 0.1 s is taken as the input feature X of the neural networks. Diffusion coefficient D is taken as output Y, so that a sample $(X, Y)$ can be constructed. For two different shapes, torus and cross, the number of samples for each shape training is 1000 sets of data. Each set of data contains the diffusion values of thirty detection points at different times. Then, 70% of the samples are divided into a study set and 30% of the samples are divided into verification sets to evaluate the generalization ability of the neural networks. The number of samples for testing is four sets of data, and the test set is not involved in the training of the neural networks. Figure 4 shows a schematic diagram of point sources and measuring points when the surface is torus and ellipsoid. Tables 4 and 5 show the inversion results and errors of diffusion coefficients when the surface is torus and cross, respectively.



(**a**) Torus (**b**) Cross

**Figure 4.** The distribution of distribution and measurement points of surfaces: (**a**) torus and (**b**) cross.

**Table 4.** Inversion results and errors of diffusion coefficients with torus surface.

| $D_{true}$ | $D_{pred}$ | *E* (%) |
|---|---|---|
| 1.5050 | 1.5042 | 0.05 |
| 2.5050 | 2.5181 | 0.66 |
| 3.5050 | 3.4826 | 0.50 |
| 4.5050 | 4.4961 | 0.20 |

**Table 5.** Inversion results and errors of diffusion coefficients with cross surface.

| $D_{true}$ | $D_{pred}$ | *E* (%) |
|---|---|---|
| 1.5050 | 1.5012 | 0.25 |
| 2.5050 | 2.4925 | 0.50 |
| 3.5050 | 3.4941 | 0.34 |
| 4.5050 | 4.5179 | 0.29 |

From Tables 4 and 5, it can be observed that for different models, the predicted value of diffusion coefficient D is almost consistent with the actual value, and the error is in the order of $10^{-3}$. Therefore, the proposed model can achieve accurate inversion of diffusion coefficient D in the absence of an analytic solution.

**Example 3.**

Based on the inversion of the diffusion coefficient and source term coefficient in the previous example, the diffusion coefficient and source term coefficient will be further inverted at the same time. Since the observed data in practical applications often contain noise (especially Gaussian noise), to verify the numerical performance of RBF neural networks under different noise levels, this paper adds Gaussian noise at different levels to the data, and the noise is added in the following way

$$u_{noise} = u + noise \cdot std(u) \cdot \varepsilon, \varepsilon \sim N(0,1), \tag{25}$$

where *noise* is noise level, $std(u)$ is the standard deviation of the whole observed dataset, $\varepsilon$ is the probability density function, and $N(0,1)$ is the standard normal distribution.

The fractional diffusion equation on a closed smooth surface in three dimensions is given as

$$\frac{\partial^{\alpha} u(\mathbf{x},t)}{\partial t^{\alpha}} = D\Delta_s u(\mathbf{x},t) + Q(\mathbf{x},t), 0 < \alpha < 1, x \in S, t \in (0,T), \tag{26}$$

where $D$ is the diffusion coefficient, $\alpha = \frac{1}{2}$, source term coefficient $Q(\mathbf{x}, t) = d$, and the analytical solution is given as

$$u(t, \mathbf{x}) = x \cdot \left(x^4 - 10x^2y^2 + 5y^4\right) \cdot \left(x^2 + y^2 - 60z^2\right) \cdot t. \tag{27}$$

There is no diffusion at the initial moment, and from the initial moment, all the cloth points on the surface of the entire surface begin to diffuse at the same time and the time step is 0.01 s. In the detection domain, different diffusion coefficients D and source term coefficients d are randomly generated. Then, 30 points on the surface are randomly selected as observation points. The simulated diffusion results corresponding to diffusion coefficient D and source term coefficient d at 0.1 s of the observation point are taken as the input feature X of the neural networks, and diffusion coefficient D and source term coefficient d are taken as the output Y. This allows you to construct a sample $(X, Y)$. For three different shapes, spherical surface, torus, and cross, the number of samples for each shape training is 1000 sets of data. Each set of data contains the diffusion values of thirty detection points at different times. Then, 70% of the samples are divided into a study set and 30% of the samples are divided into verification sets to evaluate the generalization ability of the neural networks. The number of samples for testing is six sets of data, and the test set is not involved in the training of the neural networks. Tables 6–8 show the inversion results and errors of diffusion coefficient D and source coefficient d when the surface is spherical, torus, and cross, respectively. The noise level considers interference with the inversion error at 0%, 1%, and 5%.

**Table 6.** Parameter coupling inversion errors when the surface is a sphere.

| | | **Error Values at Different Noise Levels** | | |
|---|---|---|---|---|
| | | **0%** | **1%** | **5%** |
| Case 1 | diffusion coefficient D | 0.0123 | 0.0134 | 0.0144 |
| | source term coefficient d | 0.0207 | 0.0216 | 0.0225 |
| Case 2 | diffusion coefficient D | 0.0231 | 0.0242 | 0.0250 |
| | source term coefficient d | 0.0247 | 0.0256 | 0.0265 |
| Case 3 | diffusion coefficient D | 0.0244 | 0.0253 | 0.0260 |
| | source term coefficient d | 0.0157 | 0.0166 | 0.0175 |

**Table 7.** Parameter coupling inversion errors when the surface is torus.

| | | **Error Values at Different Noise Levels** | | |
|---|---|---|---|---|
| | | **0%** | **1%** | **5%** |
| Case 1 | diffusion coefficient D | 0.0221 | 0.0230 | 0.0239 |
| | source term coefficient d | 0.0222 | 0.0231 | 0.0240 |
| Case 2 | diffusion coefficient D | 0.0155 | 0.0164 | 0.0173 |
| | source term coefficient d | 0.0102 | 0.0111 | 0.0120 |
| Case 3 | diffusion coefficient D | 0.0109 | 0.0118 | 0.0127 |
| | source term coefficient d | 0.0298 | 0.0307 | 0.0316 |

From Tables 6–8, it can be observed that for different models, the predicted values of the diffusion coefficient and source coefficient are almost consistent with the actual values, and the error is in the order of $10^{-2}$. Therefore, the proposed model can achieve joint inversion of the diffusion coefficient and source coefficient more accurately. At the same time, it can be noted that when the noise level increases, the inversion error will increase, but in an acceptable range.

**Table 8.** Parameter coupling inversion errors when the surface is cross.

| | | Error Values at Different Noise Levels | | |
|---|---|---|---|---|
| | | **0%** | **1%** | **5%** |
| Case 1 | diffusion coefficient D | 0.0179 | 0.0188 | 0.0197 |
| | source term coefficient d | 0.0182 | 0.0191 | 0.0200 |
| Case 2 | diffusion coefficient D | 0.0125 | 0.0134 | 0.0143 |
| | source term coefficient d | 0.0251 | 0.0260 | 0.0269 |
| Case 3 | diffusion coefficient D | 0.0262 | 0.0271 | 0.0280 |
| | source term coefficient d | 0.0287 | 0.0296 | 0.0305 |

**Example 4.**

In this example, the diffusion coefficient and the fractional order will be inverted at the same time. The fractional diffusion equation on a closed smooth surface in three dimensions is defined as Equation (26), where $D$ is the diffusion coefficient, $\alpha = \frac{1}{2}$, source term coefficient $Q(x,t) = 0$, and the analytical solution is given as Equation (27).

The time step is 0.01 s. In the detection domain, different diffusion coefficients D and fractional orders $\alpha$ are randomly generated. Then, 30 points on the surface are randomly selected as observation points. The simulated diffusion results corresponding to diffusion coefficient D and fractional order $\alpha$ at 0.1 s of the observation point are taken as the input feature X of the neural networks, and diffusion coefficient D and fractional order $\alpha$ are taken as the output Y. This allows you to construct a sample $(X,Y)$. For two different shapes, spherical surface and torus, the number of samples for each shape training is 1000 sets of data. Each set of data contains the diffusion values of thirty detection points at different times. Then, 70% of the samples are divided into a study set and 30% of the samples are divided into verification sets to evaluate the generalization ability of the neural networks. The number of samples for testing is six sets of data, and the test set is not involved in the training of the neural networks. Tables 9 and 10 show the inversion results and errors of diffusion coefficient D and fractional order $\alpha$ when the surface is spherical and torus, respectively.

**Table 9.** Parameter coupling inversion errors when the surface is a sphere.

| | | True Value | Predicted Value | Error Value (%) |
|---|---|---|---|---|
| Case 1 | diffusion coefficient D | 0.6500 | 0.6534 | 0.5231 |
| | fractional order $\alpha$ | 0.1500 | 0.1528 | 1.8667 |
| Case 2 | diffusion coefficient D | 0.7500 | 0.7542 | 0.5600 |
| | fractional order $\alpha$ | 0.2500 | 0.2568 | 2.7200 |
| Case 3 | diffusion coefficient D | 0.8500 | 0.8553 | 0.6235 |
| | fractional order $\alpha$ | 0.3500 | 0.3438 | 1.7714 |

**Table 10.** Parameter coupling inversion errors when the surface is a torus.

| | | True Value | Predicted Value | Error Value (%) |
|---|---|---|---|---|
| Case 1 | diffusion coefficient D | 0.6500 | 0.6513 | 0.2000 |
| | fractional order $\alpha$ | 0.1500 | 0.1571 | 4.7333 |
| Case 2 | diffusion coefficient D | 0.7500 | 0.7553 | 0.7067 |
| | fractional order $\alpha$ | 0.2500 | 0.2543 | 1.7200 |
| Case 3 | diffusion coefficient D | 0.8500 | 0.8535 | 0.4118 |
| | fractional order $\alpha$ | 0.3500 | 0.3340 | 4.5714 |

From Tables 9 and 10, it can be observed that for different models, the predicted values of the diffusion coefficient and fractional order are almost consistent with the actual values, and the error is in the order of $10^{-2}$. Therefore, the proposed model can achieve joint inversion of the diffusion coefficient and fractional order more accurately. Furthermore, the numerical results obtained in these examples clearly demonstrate that RBF neural networks can invert the parameters of the diffusion equation with high accuracy. The hybrid method based on the GFDM and RBF neural networks can accurately invert the parameters even if there is an analytic solution of the equation.

In fact, during the input of training samples, the inversion result is relatively accurate only when the diffusion coefficient and the order increase or decrease concurrently. If the diffusion coefficient and the order change irregularly, it is very difficult to accurately invert the sought-after result. In order to obtain the result accurately when the desired parameter changes irregularly, it may be necessary to introduce a more advanced neural grid, such as PINNs.

## 5. Conclusions

This paper delves into the inverse problems of anomalous diffusion on the surface. For the inverse problems, RBF neural networks are primarily utilized to invert the relevant parameters of anomalous diffusion on the surface. The GFDM is primarily employed to discretely solve the fractional derivative model on the surface. The training database for RBF neural networks to inverse the desired parameters is established by solving the proposed time-fractional diffusion equation using the GFDM.

The results of the four examples show that the hybrid method can obtain the required parameters quickly and accurately under different conditions. For GFDM, RMSE results show that GFDM has the advantage of high calculation accuracy. And, because of the factors of GFDM itself, it is more simplified and the calculation time is shorter than other methods. For RBF neural networks, the inversion results show that it has high precision in the inversion of different problems. And, because of the factor of RBF neural networks itself, the inversion time of RBF neural networks is shorter than that of other methods.

For future research, there is a lack of studies on non-smooth, non-closed, and asymmetrical surface models in this field, which warrants further investigation. In this paper, we introduced the time-fractional model to define the anomalous diffusion problem on the surface. However, other models, such as the space fractional derivative model or the space-time fractional derivative model, can also be used to describe anomalous diffusion on the surface.

**Author Contributions:** Conceptualization, L.S. and Q.X.; methodology, L.S.; software, L.S.; validation, L.S. and Q.X.; formal analysis, L.S.; investigation, L.S.; resources, L.S.; data curation, L.S.; writing—original draft preparation, L.S.; writing—review and editing, L.S. and Q.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Dataset available on request from the authors, the raw data supporting the conclusions of this article will be made available by the authors on request.

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

## References

1. Einstein, A. On the movement of small particles suspended in stationary liquids required by the molecular-kinetic theory of heat. *Ann. Der Phys.* **1905**, *17*, 549–560. [CrossRef]
2. Montroll, E.W.; Weiss, G.H. Random walks on lattices. II. *J. Math. Phys.* **1965**, *6*, 167–181. [CrossRef]
3. Montroll, E.W.; Scher, H. Random walks on lattices. IV. Continuous time random walks and influence of absorbing boundaries. *J. Stat. Phys.* **1973**, *9*, 101–135.
4. Mandelbrot, B.B. *The Fractal Geometry of Nature*; WH Freeman: New York, NY, USA, 1982.

5. Koch, D.L.; Brady, J.F. Anomalous diffusion in heterogeneous porous media. *Phys. Fluids* **1988**, *31*, 965–973. [CrossRef]
6. Ezzat, M.A. Thermoelectric MHD non-Newtonian fluid with fractional derivative heat transfer. *Phys. B Condens. Matter* **2010**, *405*, 4188–4194. [CrossRef]
7. Welch, S.; Rorrer, R.; Duren, R.G. Application of Time-Based Fractional Calculus Methods to Viscoelastic Creep and Stress Relaxation of Materials. *Mech. Time-Depend. Mater.* **1999**, *3*, 279–303. [CrossRef]
8. Sun, H.G.; Chen, W.; Li, C. Fractional differential models for anomalous diffusion. *Phys. A Stat. Mech. Its Appl.* **2010**, *389*, 2719–2724. [CrossRef]
9. Chen, W.; Sun, H.G.; Zhang, X.D. Anomalous diffusion modeling by fractal and fractional derivatives. *Comput. Math. Appl.* **2010**, *59*, 1754–1758. [CrossRef]
10. Liu, F.; Meerschaert, M.M.; McGough, R.J.; Zhuang, P.; Liu, Q. Numerical methods for solving the multiterm time-fractional wave-diffusion equation. *Fract. Calc. Appl. Anal.* **2013**, *16*, 9–25. [CrossRef]
11. Atangana, A.; Baleanu, D. New Fractional Derivatives with Nonlocal and Non-Singular Kernel: Theory and Application to Heat Transfer Model. *Therm. Sci.* **2016**, *20*, 763–769. [CrossRef]
12. Malaiya, Y.K.; Su, S. Analysis of an Important Class of Non-Markov Systems. *IEEE Trans. Reliab.* **2009**, *R-31*, 64–68. [CrossRef]
13. Rek, Z.; Sarler, B. The method of fundamental solutions for the Stokes flow with the subdomain technique. *Eng. Anal. Bound. Elem.* **2021**, *128*, 80–89. [CrossRef]
14. Liu, Y.C.; Fan, C.M.; Wei, Z.Y. Numerical solutions of two-dimensional Laplace and biharmonic equations by the localized Trefftz method. *Comput. Math. Appl.* **2020**, *88*, 120–134. [CrossRef]
15. Fu, Z.J.; Chen, W.; Gu, Y. Burton–Miller-type singular boundary method for acoustic radiation and scattering. *J. Sound Vib.* **2014**, *333*, 3776–3793. [CrossRef]
16. Li, P.W.; Fan, C.M. Generalized finite difference method for two-dimensional shallow water equations. *Eng. Anal. Bound. Elem.* **2017**, *80*, 58–71. [CrossRef]
17. Fu, Z.J.; Tang, Z.C.; Zhao, H.T. Numerical solutions of the coupled unsteady nonlinear convection-diffusion equations based on generalized finite difference method. *Eur. Phys. J. Plus* **2019**, *134*, 272. [CrossRef]
18. Liszka, T.; Orkisz, J. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Comput. Struct.* **1980**, *11*, 83–95. [CrossRef]
19. Benito, J.J.; Urena, F.; Gavete, L. Solving parabolic and hyperbolic equations by the generalized finite difference method. *J. Comput. Appl. Math.* **2006**, *209*, 208–233. [CrossRef]
20. Chan, H.F.; Fan, C.M.; Kuo, C.W. Generalized finite difference method for solving two-dimensional non-linear obstacle problems. *Eng. Anal. Bound. Elem.* **2013**, *37*, 1189–1196. [CrossRef]
21. Li, P.W.; Fan, C.M.; Chen, C.Y. Generalized Finite Difference Method for Numerical Solutions of Density-driven Groundwater Flow. *Comput. Model. Eng. Sci.* **2014**, *101*, 319–350.
22. Zhang, T.; Ren, Y.F.; Fan, C.M. Simulation of two-dimensional sloshing phenomenon by generalized finite difference method. *Eng. Anal. Bound. Elem.* **2016**, *63*, 82–91. [CrossRef]
23. Chanda, S.; Balaji, C.; Venkateshan, S.P. Non-intrusive measurement of thermal contact conductance at polymer-metal two-dimensional annular interface. *Heat Mass Transf.* **2019**, *55*, 327–340. [CrossRef]
24. Harsha, K.; Gnanasekaran, N. A combined ANN-GA and experimental based technique for the estimation of the unknown heat flux for a conjugate heat transfer problem. *Heat Mass Transf.* **2018**, *54*, 3185–3197.
25. Fabio, V.D.; Luciano, L.; Sabrina, F.P. Physics informed neural networks for an inverse problem in peridynamic models. *Eng. Comput.* **2024**. [CrossRef]
26. Ku, C.Y.; Liu, C.Y.; Chiu, Y.J.; Chen, W.D. Deep Neural Networks with Spacetime RBF for Solving Forward and Inverse Problems in the Diffusion Process. *Mathematics* **2024**, *12*, 1407. [CrossRef]
27. Lin, W.M.; Yang, C.D.; Lin, J.H. A fault classification method by RBF neural networks with OLSlearning procedure. *IEEE Trans. Power Deliv.* **2001**, *16*, 473–477. [CrossRef]
28. Cheung, K.C.; Ling, L. A kernel-based embedding method and convergence analysis for surfaces pdes. *Siam J. Sci. Comput.* **2018**, *40*, A266–A287. [CrossRef]
29. Podlubny, I. *Fractional Differential Equations*; Academic Press: San Diego, CA, USA, 1999.
30. Fu, Z.J.; Chen, W.; Ling, L. Method of approximate particular solutions for constant and variable-order fractional diffusion models. *Eng. Anal. Bound. Elem.* **2015**, *57*, 37–46. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.