Vol. 4, No. 1, 2022

https://doi.org....

UDC 004.4`2

Ya. I. Sokolovskyy[1], V.I.Yarkun[2], M.V. Levkovych[1], D.B. Ratynchuk[2]
[1] Lviv Polytechnic National University, Lviv, Ukraine
[2] Ukrainian National Forestry University, Lviv, Ukraine

SOFTWARE AND ALGORITHMIC PROVISION OF PARALLEL CALCULATION OF NON-ISOTHERMAL MOISTURE TRANSFER BASED ON THE APPARATUS OF FRACTIONAL DERIVATIVES

A new mathematical model of the nonstationary process of heat and moisture transfer in the two-dimensional region is constructed on the basis of the use of Caputo and Grunwald-Letnikov derivatives. An implicit finite-difference scheme for approximation of a mathematical model of noisothermal moisture transfer taking into account the fractional integro-differential apparatus is developed. The given algorithm of numerical realization of model allows to receive values of function of temperature and humidity for all points of area of partition. The method of fractional steps is adapted for numerical realization of mathematical model. It allowed performing parallel calculations of two difference half-step taking into account the corresponding spatial coordinate. The implemented algorithm of parallel calculation of non-isothermal moisture transfer in materials of fractal structure makes it possible to obtain a reliable result without the need to conduct complex and expensive practical experiments. It is proposed to use the Model-View-Presenter design pattern for software development. The software a parallel algorithm using .Net threads and algorithmic features of the TPL library has developed. It has been tested on multi-core computer systems with CPUs of different capacities. The .NET Stopwatch class was used to measure the execution time of sequential and parallel algorithms. A two-dimensional case with a mesh partition is studied for three cases: a square shape, a wide rectangular shape, and a tall rectangular shape. Graphs of dynamics of acceleration and efficiency of algorithms are given, and their analysis is also carried out. To smooth the graphs of acceleration and efficiency of algorithms, we apply approximation of experimental data based on the obtained results.

Key words: Caputo derivatives, Core, Efficiency, Fractal materials, Grunwald-Letnikov derivatives, Heat-and-moisture transfer, Parallelization, Thread.

Introduction

The relevance of the study. Non-equilibrium processes of heat and mass transfer in capillary-porous anisotropic media are characterized by fractality of spatial and temporal characteristics. Therefore, mathematical modeling of such processes requires the development of new non-traditional methods and means of modeling, which are based on the mathematical apparatus of integro-differential equations of fractional order. Unlike traditional mathematical models based on local partial differential equations, they will be able to describe the fractal nature of nonlinear relationships in time and space, the self-similarity of fractal systems, the "memory" effect, and spatial correlations. Such mathematical models of fractional order describe the evolution of physical systems with residual memory, which occupy an intermediate place between Markov systems and systems characterized by full memory. In particular, the fractionalization indicator show the fate of system states that are preserved throughout the entire process of its functioning. However, taking into account all components of the model, in particular the effects of memory and spatial correlation on a real time scale, leads to a significant increase in the computational complexity of implementation algorithms. And this leads to significant costs of computing resources. In addition, most modern technical means of parallelization are expensive. Therefore, to parallelize processes, it is advisable to use publicly available hardware that supports parallel computing. The relevance of this research is determined by the construction of such mathematical models that would be amenable to parallel implementation on available computing equipment and at the same time would provide the necessary speed and high efficiency of the computing process.

The purpose of the study – to investigate the efficiency and acceleration of parallel algorithms that implement the implicit finite-difference scheme of the mathematical model of the non-stationary process of moisture transfer taking into account the fractional integro-differential apparatus.

The main tasks of the research – build a mathematical model of the non-stationary process of heat and moisture transfer in the two-dimensional domain based on the use of Caputo and Grunwald-Letnikov derivatives. Develop an implicit finite-difference scheme for approximating the specified mathematical model;

– to develop an algorithm for parallelizing the process of heat and moisture transfer in a two-dimensional domain.

– to develop software that implements a parallel algorithm using the TPL library in .Net;

– test the software on available hardware with processors of different power.

The object of the study is heat-mass transfer processes in capillary-porous materials with a fractal structure during drying.

The subject of research is mathematical models, methods of analysis and parallel algorithms implementing heat-mass transfer processes in capillary-porous materials based on the use of Caputo and Grunwald-Letnikov derivatives.

The research methods. The work used finite-difference methods using the fractional integro-differentiation apparatus for numerical implementation of the mathematical model of non-isothermal wet transfer and methods of parallel implementation of the proposed algorithm.

The scientific novelty consists in building a mathematical model of the non-stationary process of heat and moisture exchange in capillary-porous materials of the fractal structure and increasing the efficiency of the parallel algorithm for implementing the model.

Practical significance. Based on the proposed mathematical models, an algorithm and software were developed for the parallelization of the computational process of heat-mass transfer in capillary-porous materials with a fractal structure.

Analysis of literary sources

Non-traditional approaches should be used to model complex systems characterized by structural inhomogeneity, self-organization and biological variability of rheological properties, deterministic chaos, and "memory" effects. This approach can be based on the use of fractional integro-differentiation [3]. In addition, the mathematical apparatus of non-integer differentiation and integration is ambiguous. The derivative or integral can be used to model systems in different senses: Wright, Weyl, Riemann-Liouville, Marshaw, Grunwald-Letnikov, or Caputo. From the analysis of existing works, the most used fractional derivatives are Riemann-Liouville derivatives [1] - [2] and Caputo [4] - [5]. In particular, in [5] a numerical approach to the Caputo derivative is given and the stability of derivative numerical algorithms is proved using the Fourier method. In other articles [5] propose algorithms for numerical calculations of integrals that arise during time sampling in diffusion equations with the Caputo derivative are developed. The speed of these algorithms was tested. There are works where the non-integer derivative in spatial coordinates in the boundary value problem of heat transfer is defined in the Grunwald-Letnikov sense, and a method for solving the problem using fractional steps is proposed.

It is expedient to use distributed and parallel computing technologies for research this will reduce time and reduce the use of natural resources. A works [1], [9-12], are devoted to this problem. They present parallel multistage methods for solving ordinary differential equations.

Thus, in this work a two-dimensional mathematical model of heat and humidity transfer in the material is built, it is described by a system of fractional-differential equations in partial derivatives. An algorithm for numerical implementation of a mathematical model based on the splitting method is developed and presented. Parallel software for analysis of acceleration and efficiency of this algorithm has been created.

The results of the study and their discussion

Mathematical model.

The mathematical model of the nonstationary process of heat and moisture transfer in the two-dimensional region is described by an interconnected system of differential equations in partial derivatives with fractional order in time t and spatial coordinates x_1 and x_2:

$$c\rho\,{}^{C}D_t^{\gamma}T=\sum_{i=1}^{2}\lambda_i\,{}^{GL}D_{x_i}^{\upsilon}T+\varepsilon\rho_0 r\,{}^{C}D_t^{\gamma}U, \tag{1}$$

$${}^{C}D_t^{\gamma}U=\sum_{i=1}^{2}a_i\,{}^{GL}D_{x_i}^{\upsilon}U+\sum_{i=1}^{2}\delta a_i\,{}^{GL}D_{x_i}^{\upsilon}T, \tag{2}$$

with the initial conditions:

$$T\big|_{t=0}=T_0(x_1,x_2),\ U\big|_{t=0}=U_0(x_1,x_2), \tag{3}$$

and the following boundary conditions of the third kind:

$$\lambda_i D_{x_i}T\big|_{x_i=0,l_i'}+\rho_0(1-\varepsilon)\beta_i\left(U\big|_{x_i=0,l_i'}-U_p\right)=\alpha_i\left(T\big|_{x_i=0,l_i'}-t_c\right), \tag{4}$$

$$a_i\delta D_{x_i}T\big|_{x_i=0,l_i'}+a_i D_{x_i}U\big|_{x_i=0,l_i'}=\beta_i\left(U_p-U\big|_{x_i=0,l_i'}\right), \tag{5}$$

where $T=T(t,x_1,x_2)$ is temperature, $U=U(t,x_1,x_2)$ is moisture content, $(t,x_1,x_2)\in G$ $G=[0,T_{\max}]\times[0,l_1']\times[0,l_2']$, ε is phase transition coefficient, ρ_0 is basic density, $\lambda_i\,(T,U)\,(i=1,2)$ are coefficients of thermal conductivity, $c(T,U)$ is specific heat capacity, $a_i\,(T,U)\,(i=1,2)$ are coefficients of water conductivity, $\rho\,(U)$ is density, r is specific heat of vapour generation, $\delta\,(T,U)$ is thermogradient coefficient, $a_i\,(T,U)\,(i=1,2)$ are coefficients of water conductivity, $U_p\,(t_c,\varphi)$ is equilibrium humidity, t_c is the ambient temperature value, φ is relative moisture content of the drying agent α_i are coefficients of heat exchange, β_i are coefficients of moisture exchange, D_{x_i} is derivative of the first order in spatial coordinates, ${}^{C}D_t^{\gamma}$, ${}^{C}D_{x_i}^{\gamma}$ are Caputo derivatives of fractional order $\gamma,(0<\gamma<1)$ in time and spatial coordinates, ${}^{GL}D_{x_i}^{\upsilon}$ is the Grunwald-Letnikov derivative of fractional order $\upsilon,(1<\upsilon<2)$ in spatial coordinates.

Derived Caputo γ, where $\gamma\in(0,1)$ is determined by the following relationship:

$${}^{C}D_t^{\gamma}f(t)=\frac{1}{\Gamma(1-\gamma)}\int_0^t(t-\xi)^{-\gamma}f'(\xi)\,d\xi, \tag{6}$$

where: $\Gamma(\cdot)$ – Gamma function.

The Grunwald-Letnikov derivative is described by the following relation:

$${}_{a}^{GL}D_x^{\upsilon}f(x)=\lim_{h\to 0}\frac{1}{h^{\upsilon}}\sum_{j=0}^{\frac{(x-a)}{h}}(-1)^j\frac{\Gamma(\upsilon+1)}{j!\Gamma(\upsilon-j+1)}f(x-jh). \tag{7}$$

Given the ratio for the Caputo derivative (6) and the results of studies [5], we can write its numerical approximation in the form:

$$\begin{aligned}{}^{C}D_{t^k}^{\gamma}f(t^k)=&\frac{1}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\sum_{j=0}^{k-1}[w_{1,k-j}(f^{j+1}-f^{j-1})+\\&w_{2,k-j}(f^{j+1}-2f^{j}+f^{j-1})]+\mathrm{O}(\Delta\tau^{3-\gamma}),\end{aligned} \tag{8}$$

where: $w_{1,k-j}=\frac{2-\gamma}{2}\left[(k-j)^{1-\gamma}-(k-j-1)^{1-\gamma}\right]$, $w_{2,k-j}=(k-j)^{2-\gamma}-(k-j-1)^{2-\gamma}-(2-\gamma)(k-j-1)^{1-\gamma}$, $j=0,\ldots,k-1$; $k=1,\ldots,\mathrm{K}$; $t^k=k\Delta\tau$.

The numerical approximation of the fractional Grunwald-Letnikov derivative is recorded taking into account (7):

$${}^{GL}D_{x_n}^{\upsilon}f(x_n)\approx\frac{1}{h^{\upsilon}}\sum_{j=0}^{n}g_j f_{n-j+1}, \tag{9}$$

where $g_0=1$, $g_j=(-1)^j\upsilon(\upsilon-1)\ldots(\upsilon-j+1)/j!$.

Thus, taking into account the approximation of the fractional derivatives Kaputo (8) and Grunwald-Letnikov (9), we obtained a numerical approximation of the mathematical model of no isothermal moisture transfer (1-2):

$$\frac{c\rho}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\sum_{j=0}^{k-1}\left[w_{1,k-j}\left(T_{n,m}^{j+1}-T_{n,m}^{j-1}\right)+w_{2,k-j}\tilde{T}\right]=\frac{\lambda_1}{h_1^{\upsilon}}\sum_{j=0}^{n}g_j T_{n-j+1,m}^{k}+$$
$$+\frac{\lambda_2}{h_2^{\upsilon}}\sum_{j=0}^{m}g_j T_{n,m-j+1}^{k}+\frac{\varepsilon\rho_0 r}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\sum_{j=0}^{k-1}\left[w_{1,k-j}\left(U_{n,m}^{j+1}-U_{n,m}^{j-1}\right)+w_{2,k-j}\tilde{U}\right], \quad (10)$$

$$\frac{\Delta\tau^{-\gamma}}{\Gamma(3-\gamma)}\sum_{j=0}^{k-1}\left[w_{1,k-j}\left(U_{n,m}^{j+1}-U_{n,m}^{j-1}\right)+w_{2,k-j}\tilde{U}\right]=\frac{a_1\delta}{h_1^{\upsilon}}\sum_{j=0}^{n}g_j T_{n-j+1,m}^{k}+$$
$$\frac{a_1}{h_1^{\upsilon}}\sum_{j=0}^{n}g_j U_{n-j+1,m}^{k}+\frac{a_2}{h_2^{\upsilon}}\left(\delta\sum_{j=0}^{m}g_j T_{n,m-j+1}^{k}+\sum_{j=0}^{m}g_j U_{n,m-j+1}^{k}\right), \quad (11)$$

where: $T_{n,m}^{k}=T\left(t^{k},x_{1(n)},x_{2(m)}\right), U_{n,m}^{k}=U\left(t^{k},x_{1(n)},x_{2(m)}\right)$ is the value of the function of temperature and moisture content, respectively, at points t^{k}, $x_{1(n)}$, $x_{2(m)}$, $\tilde{T}=T_{n,m}^{j+1}-2T_{n,m}^{j}+T_{n,m}^{j-1}$, $\tilde{U}=U_{n,m}^{j+1}-2U_{n,m}^{j}+U_{n,m}^{j-1}$,

$$\varpi_{\Delta\tau,h_1,h_2}=\{\left(t^{k},x_{1n},x_{2m}\right): x_{1(n)}=(n-1)h_1,\ x_{2(m)}=(m-1)h_2,$$
$$t^{k}=k\Delta\tau,\ n=1,\ldots,N;\ m=1,\ldots,M;\ k=0,1,\ldots,K;$$
$$h_1=\frac{l_1'}{N-1};h_2=\frac{l_2'}{M-1};\Delta\tau=\frac{\mathrm{T}_{\max}}{K}\}.$$

Algorithm of numerical implementation of the model

The mathematical model in the finite-difference form (10) - (11) is depicted in the form of an implicit scheme and for its numerical implementation it is necessary to use the splitting method [13-15].

In this work, the method of fractional steps is used, which allows to calculate in parallel two difference half-step taking into account a certain spatial coordinate and to find the value of the function of temperature and moisture content for all points of the $\varpi_{\Delta\tau,h_1,h_2}$ partition area.

We present the algorithm of numerical implementation:

1. In the time step $k=0$ implement a cycle by $n=1,\ldots,N$ and a cycle by $m=1,\ldots,M$ from the initial conditions:

$$T_{n,m}^{0}=T_0\left(x_{1(n)},x_{2(m)}\right),\ U_{n,m}^{0}=U_0\left(x_{1(n)},x_{2(m)}\right), \quad (12)$$

2. At the time step $k=1$ we take into account the property $f_{-1}=f_0+\mathrm{O}\left(\Delta\tau^{3}\right)$ and the value of the coefficients $w_{1,1}=(2-\gamma)/2,\ w_{2,1}=1$. Get the difference scheme:

$$\frac{c\rho}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(T_{n,m}^{1}-T_{n,m}^{0}\right)=\frac{\lambda_1}{h_1^{\upsilon}}\sum_{j=0}^{n}g_j T_{n-j+1,m}^{1}+$$
$$+\frac{\lambda_2}{h_2^{\upsilon}}\sum_{j=0}^{m}g_j T_{n,m-j+1}^{1}+\frac{\varepsilon\rho_0 r}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(U_{n,m}^{1}-U_{n,m}^{0}\right), \quad (13)$$

$$\frac{\Delta\tau^{-\gamma}}{\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(U_{n,m}^{1}-U_{n,m}^{0}\right)=\frac{a_1\delta}{h_1^{\upsilon}}\sum_{j=0}^{n}g_j T_{n-j+1,m}^{1}+$$
$$+\frac{a_1\delta}{h_1^{\upsilon}}\sum_{j=0}^{n}g_j U_{n-j+1,m}^{1}+\frac{a_2}{h_2^{\upsilon}}\left(\delta\sum_{j=0}^{m}g_j T_{n,m-j+1}^{1}+\sum_{j=0}^{m}g_j U_{n,m-j+1}^{1}\right), \quad (14)$$

The time interval $\Delta\tau$ between the points t^{0} and t^{1} on the difference grid is split in half. In the general case between the points of t^{k} and t^{k+1} splitting is shown in Fig.1.

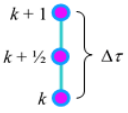

Fig.1. Splitting the time interval

2.1. In the first half-step of the interval $\Delta\tau$ we will consider only the Grunwald-Letnikov υ derivative in the spatial coordinate x_1. Thus, the first half-step will look like:

$$\frac{c\rho}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(T_{n,m}^{1/2}-T_{n,m}^{0}\right)=\frac{\lambda_1}{h_1^{\upsilon}}\sum_{j=0}^{n}g_j T_{n-j+1,m}^{1/2}+$$
$$+\frac{\varepsilon\rho_0 r}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(U_{n,m}^{1/2}-U_{n,m}^{0}\right), \quad (15)$$

$$\frac{\Delta\tau^{-\gamma}}{\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(U_{n,m}^{1/2}-U_{n,m}^{0}\right)=\frac{a_1}{h_1^{\upsilon}}\left(\delta\sum_{j=0}^{n}g_j T_{n-j+1,m}^{1/2}+\sum_{j=0}^{n}g_j U_{n-j+1,m}^{1/2}\right). \tag{16}$$

2.2. In the second half of the $\Delta\tau$ interval, we will take into account only the Grunwald-Letnikov υ derivative in the spatial coordinate of the x_2. Thus, the second half-step will look like:

$$\frac{c\rho}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(T_{n,m}^{1}-T_{n,m}^{1/2}\right)=\frac{\lambda_2}{h_2^{\upsilon}}\sum_{j=0}^{m}g_j T_{n,m-j+1}^{1}+ \\ +\frac{\varepsilon\rho_0 r}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(U_{n,m}^{1}-U_{n,m}^{1/2}\right), \tag{17}$$

$$\frac{\Delta\tau^{-\gamma}}{\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(U_{n,m}^{1}-U_{n,m}^{1/2}\right)=\frac{a_2}{h_2^{\upsilon}}\left(\delta\sum_{j=0}^{m}g_j T_{n,m-j+1}^{1}+\sum_{j=0}^{m}g_j U_{n,m-j+1}^{1}\right). \tag{18}$$

In both half-steps we obtain a matrix of non-traditional form $A=\left(a_{ij}\right)$ $Z=const$:

$$a_{ij}=\begin{cases}0,\ j\geq i+2;\\ 0,i=M,1\leq j\leq M-2;\\ Zg_1\delta,i=j\neq 1\neq M;\\ a_2\delta\gamma,i=j=1;i=M,\ j=M-1;\\ -a_2\delta,i=j=M;i=1,\ j=2;\\ Zg_{i-j+1}\delta,other.\end{cases} \tag{19}$$

the algorithm for solving half-step (15) - (18) is given in [1,].

2.3. Combining two half-step: relations (15) and (17), (16) and (18), respectively, we obtain:

$$\frac{c\rho}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(T_{n,m}^{1}-T_{n,m}^{0}\right)=\frac{\lambda_2}{h_2^{\upsilon}}\sum_{j=0}^{m}g_j T_{n,m-j+1}^{1}+ \\ +\frac{\lambda_1}{h_1^{\upsilon}}\sum_{j=0}^{n}g_j T_{n-j+1,m}^{1/2}+\frac{\varepsilon\rho_0 r}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(U_{n,m}^{1}-U_{n,m}^{0}\right), \tag{20}$$

$$\frac{\Delta\tau^{-\gamma}}{\Gamma(3-\gamma)}\frac{4-\gamma}{2}\left(U_{n,m}^{1}-U_{n,m}^{0}\right)=\frac{a_1\delta}{h_1^{\upsilon}}\sum_{j=0}^{n}g_j T_{n-j+1,m}^{1/2}+ \\ +\frac{a_1}{h_1^{\upsilon}}\sum_{j=0}^{n}g_j U_{n-j+1,m}^{1/2}+\frac{a_2}{h_2^{\upsilon}}\left(\delta\sum_{j=0}^{m}g_j T_{n,m-j+1}^{1}+\sum_{j=0}^{m}g_j U_{n,m-j+1}^{1}\right). \tag{21}$$

3. In time step $k=2$, the values of the $T_{n,m}^{0}$, $U_{n,m}^{0}$, $T_{n,m}^{1}$, $U_{n,m}^{1}$ functions are known from the initial conditions and are found in paragraph 2 of this algorithm. Taking into account half-step 2.1-2.3, we will similarly find the values of the functions $T_{n,m}^{2}$, $U_{n,m}^{2}$.

4. Thus, each subsequent time step contains a number of known values of the temperature and moisture content functions, which were found from the previous steps. For $k=K$ we get:

$$\frac{c\rho}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\Big(\frac{4-\gamma}{2}\left(T_{n,m}^{1}-T_{n,m}^{0}\right)+\frac{2-\gamma}{2}\left(T_{n,m}^{2}-T_{n,m}^{0}\right)+ \\ +T_{n,m}^{2}-2T_{n,m}^{1}+T_{n,m}^{0}+\ldots+\frac{2-\gamma}{2}\left(T_{n,m}^{K'}-T_{n,m}^{K-2}\right)+T_{n,m}^{K'}- \\ -2T_{n,m}^{K''}+T_{n,m}^{K-2}\Big)=\frac{\lambda_i}{h_i^{\upsilon}}\sum_{j=0}^{n(m)}g_j T_{n-j+1(m-j+1)}^{K-1/2}+\frac{\varepsilon\rho_0 r}{\Delta\tau^{\gamma}\Gamma(3-\gamma)}\times \\ \Big(\frac{4-\gamma}{2}\left(U_{n,m}^{1}-U_{n,m}^{0}\right)+\frac{2-\gamma}{2}\left(U_{n,m}^{2}-U_{n,m}^{0}\right)+U_{n,m}^{2}-2U_{n,m}^{1}+ \\ +U_{n,m}^{0}+\ldots+\frac{2-\gamma}{2}\left(U_{n,m}^{K'}-U_{n,m}^{K-2}\right)+U_{n,m}^{K'}-2U_{n,m}^{K''}+U_{n,m}^{K-2}\Big), \\ \frac{\Delta\tau^{-\gamma}}{\Gamma(3-\gamma)}\Big(\frac{4-\gamma}{2}\left(U_{n,m}^{1}-U_{n,m}^{0}\right)+\frac{2-\gamma}{2}\left(U_{n,m}^{2}-U_{n,m}^{0}\right)+U_{n,m}^{2}- \\ -2U_{n,m}^{1}+U_{n,m}^{0}+\ldots+\frac{2-\gamma}{2}\left(U_{n,m}^{K'}-U_{n,m}^{K-2}\right)+U_{n,m}^{K'}-2U_{n,m}^{K''}+U_{n,m}^{K-2}\Big)= \\ =\frac{a_i}{h_i^{\upsilon}}\left(\delta\sum_{j=0}^{n(m)}g_j T_{n-j+1(m-j+1)}^{K-1/2}+\sum_{j=0}^{n(m)}g_j U_{n-j+1(m-j+1)}^{K-1/2}\right).$$

In the case of $K' = K - 1/2, K'' = K - 1$, we obtain the first half-step (taking into account the derived υ , in the spatial coordinate x_1). In the case of $K' = K, K'' = K - 1/2,$ we obtain the second half-step (taking into account the derivative of υ in the spatial coordinate of x_2).

Given half-step 2.1-2.3 in the same way we find the values of the functions of $T^K_{n,m}$, $U^K_{n,m}$.

5. Given the boundary conditions of the third kind and their classical approximations of derivatives, we find $T^k_{N,m}$, $T^k_{1,m}$, $U^k_{N,m}$, $U^k_{1,m}$, $T^k_{n,M}$, $T^k_{n,1}$, $U^k_{n,M}$, $U^k_{n,1}$.

Software implementation.

A distributed software complex was developed for this work *Fig.2*. It consists of a cloud database and three interconnected software products: module of mathematical model; module analysis of algorithms; graph approximation web-system. All three products work according to a clearly defined procedure. They have implemented various functionalities for the study of heat and mass transfer. Connections for data exchange are organized between these modules. In the future, some calculations will be transferred to cloud technology. This will increase the productivity of the study several times.

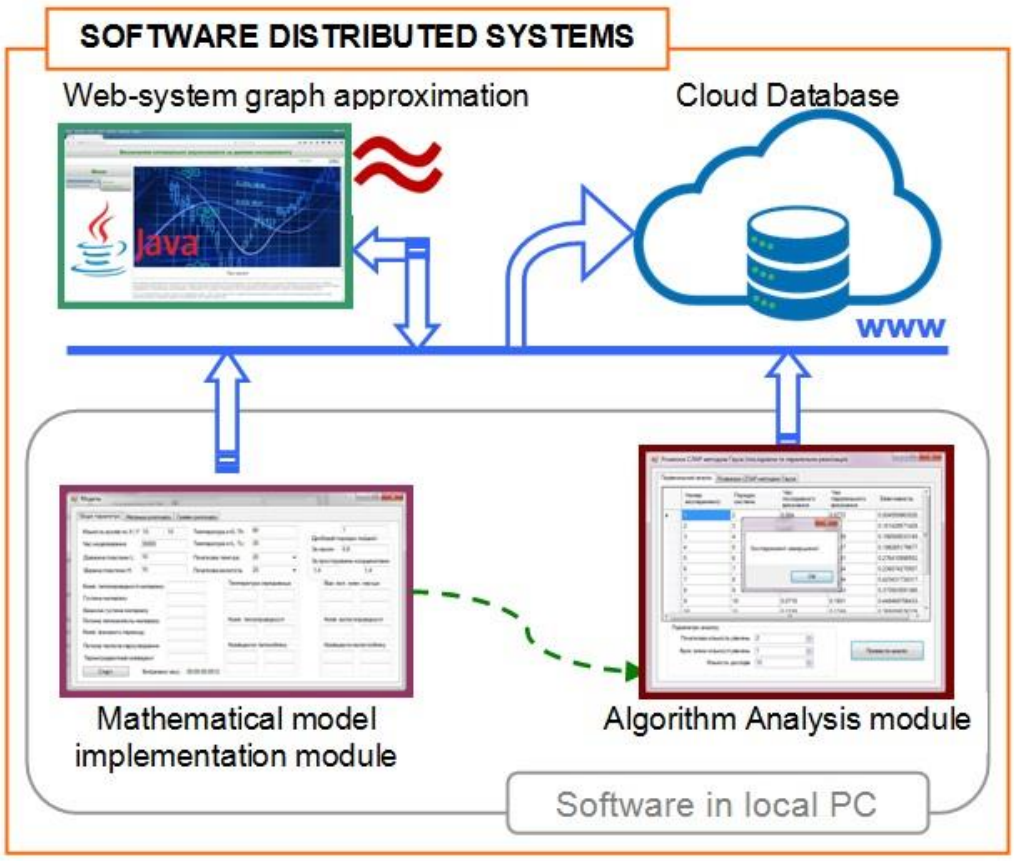

Fig. 2. Software distributed systems

Mathematical model module

Software for the study of non-isothermal moisture transfer in fractal structure materials [11]. It implements a mathematical model for the distribution of heat and moisture in a material using a parallel algorithm *Fig.3*.

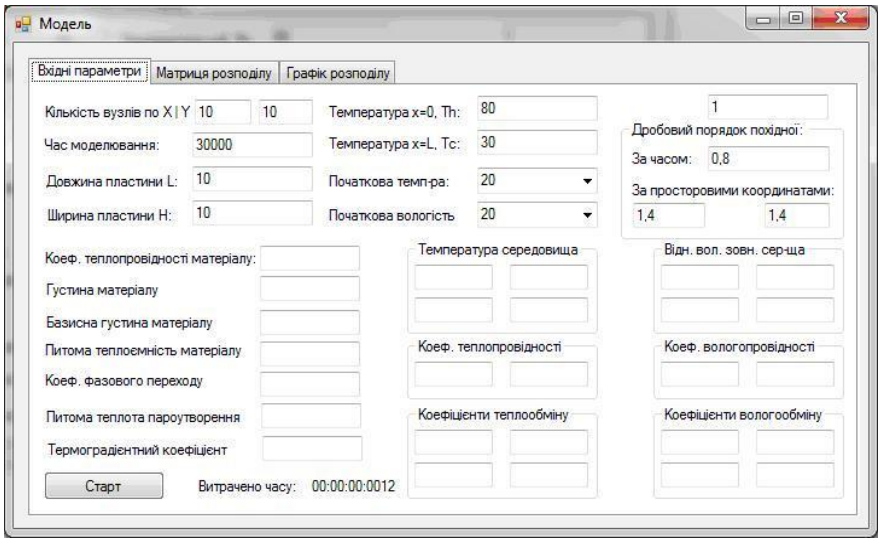

Fig. 3. Mathematical model implementation module

Module analysis of algorithms

It is software for performing comparative analysis of serial and parallel algorithms. This is the main unit of the distributed system so let's look at it in more detail.

We need to measure the time between two events in the program. This is the time of execution of the serial algorithm and the parallel algorithm. The .NET Stopwatch class allows you to accurately measure elapsed time. A Stopwatch instance can measure elapsed time at several intervals with the total elapsed time calculated by adding all the intervals. Stopwatch is good when we need to watch the time and get some additional information about how much elapsed processor ticks does the method take etc.

There are times when classic performance measurement techniques produce conflicting results, even with repeated launches of the same application. The most common sources of error are the compiler; inability to predict the required number of background threads for training ("warm-up") on a particular application; total OS load [8].

The software is implemented in object-oriented analysis notations by .Net mechanisms. We focus on large-scale parallelism throughout the system rather than small-scale parallelism in the method. The Producer/Consumer Parallel Programming Pattern were used to design the software [7].

The Model-View-Presenter pattern is used to design the software *Fig.4*. It separates components and allows you to improve code effectively. It is used when we need to maximize the amount of code to parallelize. Our software is divided into 3 parts. View - intended for dialogue with the user. It collects input data. Model - provides functionality and work with data that will be obtained from the user. Presenter - contains software logic. Manage experiments to implement parallelization algorithms.

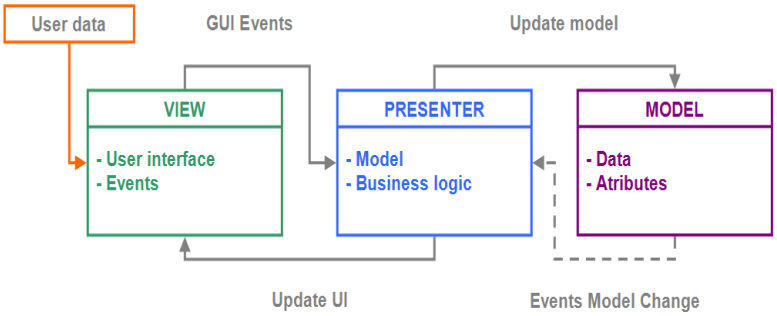

Fig. 4. Software architecture.

The developed architecture in the software system provides not only scaling, but also easy extension of parallel implementation, which will allow to implementing similar more complex models.

Graph approximation web-system

It is intended to select and display the optimal approximants for the one- and two-dimensional dependencies obtained experimentally. This system smoothes out the acceleration and efficiency graphs of the algorithms. We plot and approximate the experimental data based on the results obtained in tabular form.

Determination of acceleration and efficiency of the algorithm

Acceleration of the Parallel Algorithm - the ratio of the execution time of the serial algorithm to the execution time of the parallel algorithm: $S = T_1 / T_p$. Where T_1 – is the time of the sequential algorithm, T_p - is the time of the parallel algorithm on p processors.

A parallel algorithm can give a lot of acceleration, but using many processes is inefficient. The concept of efficiency is used to evaluate the scalability of a parallel algorithm $E = T_1 / pT_p = S / p$ [6].

Experiment settings

To perform the simulation, you must first make the necessary input parameters in the software “Algorithm Analysis”. Clicking the "Conduct Analysis" button and set the input parameters to begin the modeling process. To conduct experiments and study the acceleration and efficiency of the proposed algorithm, we introduce the necessary input parameters. We introduce the dimension of the partition of the grid for three cases. For a square shape, the initial size of the grid is 1000x1000 nodes. For a wide rectangular shape, the initial size of the grid is 500x5000 nodes. For a high rectangular shape, the initial size of the grid is 5000x500 nodes. The grid change step is 1000 nodes; the number of measurements for each case is 100. For both serial and parallel modes, the size of the area division grid used varied from 1000 to 104000 nodes.

The results presented in this paper are based on experiments performed on computers with such processors: Intel Core i3-2350m - 2.30GHz (2c/4th); Intel Core i5-3427U - 2,80GHz (2c/4th); Intel Core i7-3770K - 3.50GHz (4c/4th).

Considering that the implementation of the algorithm is a time-consuming process and given the size of the partition grid and the simulation time, one should expect long simulations. Simulation time is also affected by the CPU specifications on which the program will run.

Let's consider the Task Manager report Fig. 5. on a computer with a processor with Specifications: Number of cores 2, Number of threads four, CPU base clock 2.30GHz.

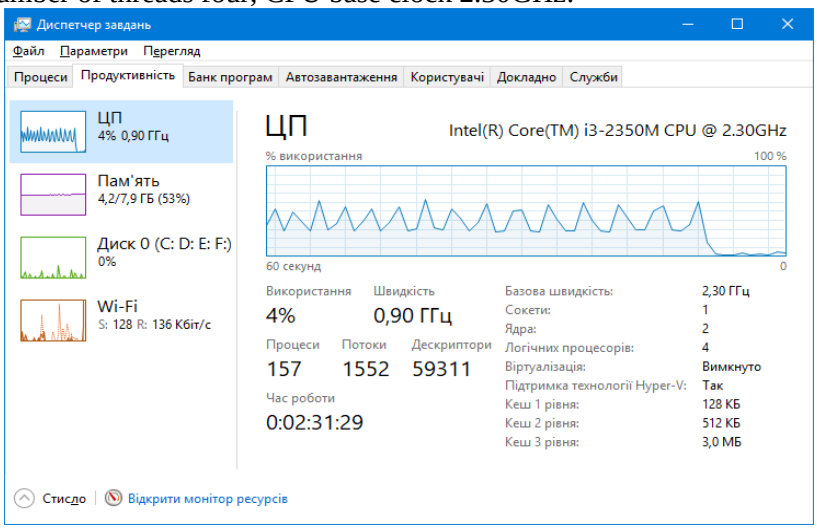

Fig.5. Task Manager

It indicates the change in CPU load during the execution of serial and parallel algorithms. Here we see the maximum load on the CPU cores during the execution of the parallel algorithm. After the completion of parallel calculations, the processor cores are unloaded to a minimum value. This kind of processor core management is provided by Task Parallel Library included in Framework .Net 4.0 [10], [6].

Analysis of parallel solutions.

Consider the acceleration graphs of the algorithm Fig.6. The approbation of the parallel algorithm was performed by dividing the grid into a square shape by 100K x 100K node. Can see how fast the acceleration of all three experiments is growing. We have approximated the experimental data and see the best result of the algorithm on the Core i7-3770K processor. As the number of experiments increases, the acceleration ranges from 2 to 4 on a dual-core machine and 4.8 to 5.2 on computers with 4-core processors. The efficiency demonstrated by the Core i3-2350m processor fluctuates within 1, which indicates low parallelization performance.

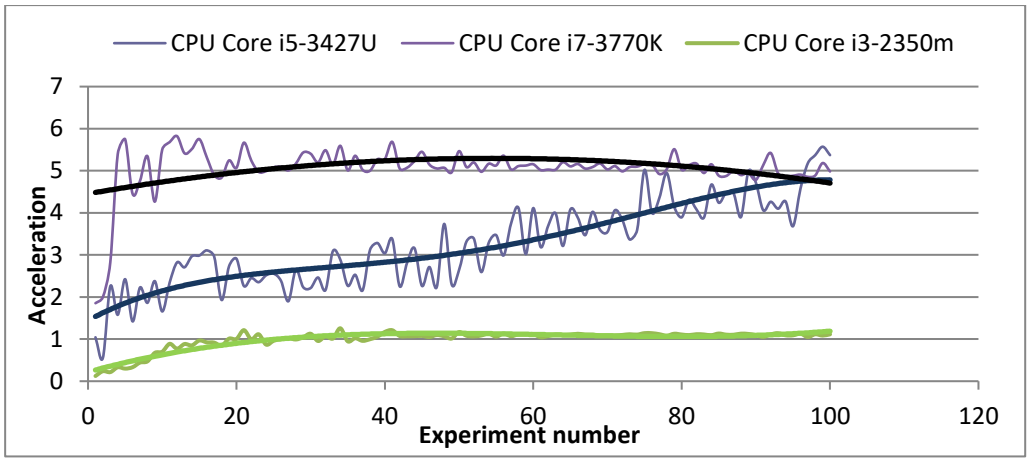

Fig.6. Algorithm acceleration change. Max grid nodes: 100K x 100K

Graphs of the acceleration algorithm tested when breaking the grid in the form of a wide rectangular shape at 500 x 100K node are presented in Fig.7. The acceleration of the parallel algorithm increases, and reaches a constant value after 20 experiments. On a dual-core computer, the average acceleration is 2.3. The algorithm has the best performance on a quad-core Core i7-3770K. As the number of experiments increases, the acceleration ranges from 4 to 5.2 on processors with a higher frequency.

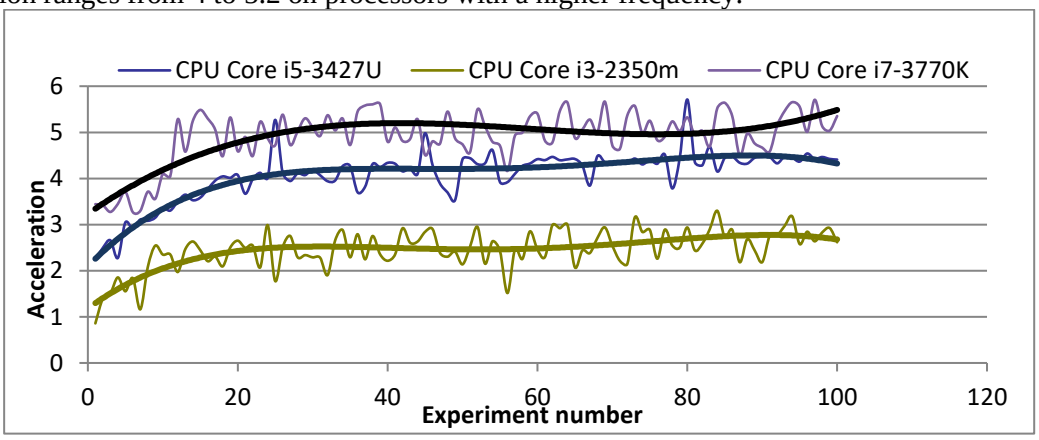

Fig. 7. Algorithm acceleration change. Max grid nodes: 500 x 100K

The result of approbation of the parallel algorithm was carried out by dividing the grid in the form of a high rectangular shape by 100K x 500 node shown in Fig. 8. The best indicator of the algorithm on the Core i7-3770K processor. The average acceleration of the parallel algorithm on different processors differs. The highest acceleration rate was obtained on a quad-core computer - 2.7. A dual-core computer shows a low acceleration of 0.6, but a dual-core with a high processor frequency is effective at 1.6.

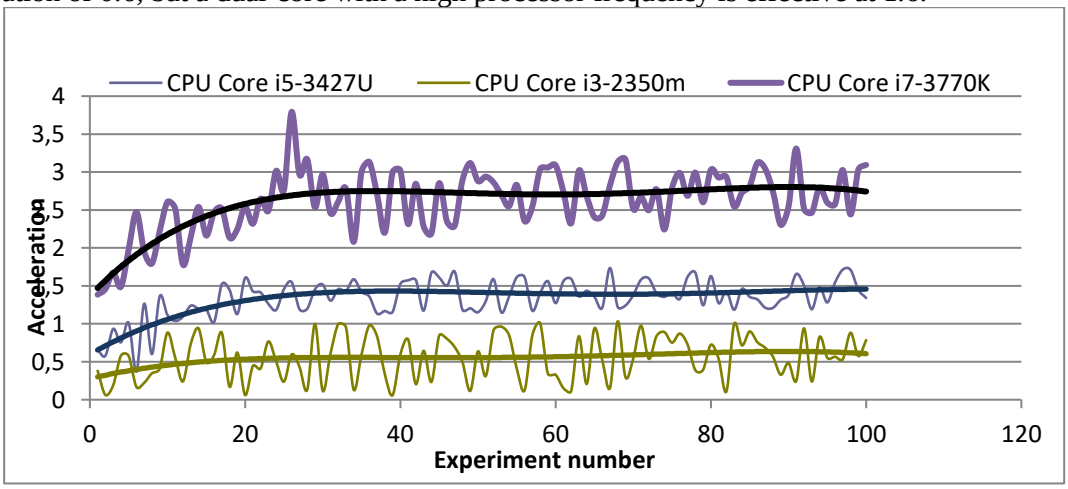

Fig. 8. Algorithm acceleration change. Max grid nodes: 100K x 500

It is possible to conduct many experiments and get good acceleration and efficiency, even at very large mesh sizes. However, we reduced the number of measurement to 100.

Conclusions

Fractal geometry is the optimal method for studying materials with a fractal structure while minimizing internal stress and maximizing the ratio of objects. Taking into account the numerical approximations of the fractional derivatives of Grunwald-Letnikov and Caputo, an implicit finite-difference scheme of the mathematical model is developed. An algorithm for numerical implementation of the model is presented, which uses the method of fractional steps to split a complex mathematical model. As a result of this method, the process of numerical implementation of the model is reduced to the parallel calculation of temperature and humidity of the material.

Most modern devices are equipped with multi-core a processor, which encourages the development of parallel algorithms and the creation of parallel software.

As a result of the study, it was established that the parallel implementation of the proposed algorithm on available hardware is faster than the sequential one:

- When breaking the grid in the form of a square shape in 2, 3.2, 5.2 times;
- When breaking the grid in the form of a wide rectangular shape -in 2.6, 4.1, 5 times;
- When breaking the grid in the form of a high rectangular shape in 0.6, 1.4, 2.6 times.

Thus, the algorithm should be used to study two-dimensional heat and mass transfer on the basis of fractions of fractional order when breaking the grid in the form of a square or wide rectangular shape. Given the obtained indicators, the algorithm can be attributed to algorithms that have high acceleration and efficiency. Analyzing the obtained estimates of acceleration and efficiency, we can conclude that the parallel implementation of the algorithm is more effective for cases where the division of the area exceeds 500x500 nodes.

We were also able to run individual operations with extremely large partitions and received accelerations within these limits. Smaller data sets resulted in very short runtimes that did not yield important results. With a little effort it is possible to create parallel software that can save a lot of resources on field experiments.

List of sources used

1 Y. Sokolovskyy, M. Levkovych, V. Yarkun, Y. Protsyk, A. Nechepurenko (2020). Parallel Realization of the Problem of Heat and Moisture Transfer in Fractal-structure Materials. IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2020 - Proceedings, pp. 86–90. https://doi.org/10.1109/CSIT49958.2020.9322021

2 Y. Sokolovskyy, A. Nechepurenko, T. Samotii, S. Yatsyshyn, O. Mokrytska, V. Yarkun (2020) Software and algorithmic support for finite element analysis of spatial heat-and-moisture transfer in anisotropic capillary-porous materials. IEEE 3rd International Conference on Data Stream Mining and Processing, DSMP 2020, 2020, pp. 316–320. https://doi.org/10.1109/DSMP47368.2020.9204175

3 Maduna, Lebo & Patnaik, Asis. (2020). Heat, moisture and air transport through clothing textiles. Textile Progress. 52. 129-166. http://dx.doi.org/10.1080/00405167.2021.1955524

4 M. Zayernouri, G. E. Karniadakis. (2014). Exponentially accurate spectral and spectral element methods for fractional ODEs, J. Comput. Phys., vol. 257, pp. 460-480. https://doi.org/10.1016/j.jcp.2013.09.039

5 C.P. Li, R.F. Wu, H.F. Ding. (2015). High-order approximation to Caputo derivatives and Caputo-type advection-diffusion equations. Communications in Applied and Industrial Mathematics, 6(2), e-536: 1-32. https://doi.org/10.1016/j.cam.2015.11.037

6 Gaston C. Hillar. (2012). “Professional Parallel Programming with C#: Master Parallel Extensions with .NET 4”, Willey Publishing Inc., p.576

7 F. Zhu, Y. Zhou, S. Liu. (2013) “Analysis of the moisture diffusion transfer through fibrous porous membrane used for waterproof breathable fabrics”, Heat and Mass Transfer vol.49, pp.1503–1508. http://dx.doi.org/10.1007/s00231-013-1191-2

8 S. Toub. (2010). Patterns of Parallel Programming, Microsoft Corporation, p.118.

9 Y. Sokolovskyy, V. Shymanskyi, M. Levkovych, V. Yarkun (2016) “Mathematical modeling of heat and moisture transfer and reological behavior in materials with fractal structure using the

parallelization of predictor-corrector numerical method", 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), Lviv, pp. 108-111. https://doi.org/10.1109/DSMP.2016.7583518

10 A. Baban, C. Bonchis, A. Fikl, F. Rosu. (2016). Parallel simulations for fractional-order systems. 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), pp. 141-144. https://doi.org/10.1109/SYNASC.2016.033

11 D. Tavares, R. Almeida, D. Torres. (2016). "Caputo derivatives of fractional variable order: Numerical approximations", Communications in Nonlinear Science and Numerical Simulation, Volume 35, Pages 69-87. http://dx.doi.org/10.1016/j.cnsns.2015.10.027

12 N. Kalinnik, T. Rauber. (2018). "Execution Behavior Analysis of Parallel Schemes for Implicit Solution Methods for ODEs". 17th International Symposium on Parallel and Distributed Computing (ISPDC), 2018, pp. 1-8. https://doi.org/10.1109/ISPDC2018.2018.00010

13 Y. Sokolovskyy, M. Levkovych, O. Mokrytska, V. Atamanvuk. (2018). "Mathematical Modeling of Two-Dimensional Deformation-Relaxation Processes in Environments with Fractal Structure", IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018, pp. 375–380. https://doi.org/10.1109/DSMP.2018.8478569

14 C. Bonchis, E. Kaslik, F. Rosu. (2019). "HPC optimal parallel communication algorithm for the simulation of fractional-order systems", Journal of Supercomputing, 2019, 75(3), pp. 1014–1025. https://link.springer.com/article/10.1007/s11227-018-2267-z

15 Y. Sokolovskyy, M. Levkovych, I. Sokolovskyy. (2020). The study of heat transfer and stress-strain state of a material, taking into account its fractal structure, Mathematical Modeling and Computing, pp. 400–409. https://doi.org/10.23939/mmc2020.02.400

УДК 004.4`2

Я.І. Соколовський[1], В.І. Яркун[2], М.В. Левкович[1], Д.Б. Ратинчук[2]
[1] НУ «Львівська політехніка», Львів, Україна
[2] Національний лісотехнічний університет України, Львів, Україна

ПРОГРАМНО-АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ ПАРАЛЕЛЬНОГО РОЗРАХУНКУ НЕІЗОТЕРМІЧНОГО ВОЛОГОПЕРЕНЕСЕННЯ НА ПІДСТАВІ АПАРАТУ ПОХІДНИХ ДРОБОВОГО ПОРЯДКУ

Побудовано нову математичну модель нестаціонарного процесу тепло- і вологообміну в двовимірній області на основі використання похідних Капуто та Грюнвальда-Летникова. Розроблено неявну скінченно-різницеву схему для апроксимації математичної моделі ноізотермічного вологоперенесення з урахуванням дробового інтегро-диференціального апарату. Наведений алгоритм чисельної реалізації моделі дозволяє отримати значення функції температури і вологості для всіх точок області розбиття. Для чисельної реалізації математичної моделі адаптовано метод дробових кроків. Це дозволило провести паралельні обчислення двох різницевих півкроків з урахуванням відповідної просторової координати. Реалізований алгоритм паралельного розрахунку неізотермічного волого-перенесення в матеріалах фрактальної структури дає можливість отримати достовірний результат без необхідності проведення складних і дороговартісних практичних експериментів. Для розроблення програмного забезпечення пропонується використовувати шаблон проектування Model-View-Presenter. У програмному забезпеченні розроблено паралельний алгоритм із використанням потоків .Net і алгоритмічних особливостей бібліотеки TPL. Для вимірювання часу виконання послідовних і паралельних алгоритмів використовувався клас .NET Stopwatch. Тестування програмного забезпечення здійснено на багатоядерних комп'ютерних системах з центральним процесором різної потужності. Досліджується двовимірний випадок для області розбиття квадратної форми, широкої прямокутної форми та високої прямокутної форми. Наведено графіки динаміки прискорення та ефективності алгоритмів, а також проведено їх аналіз. Для згладжування графіків прискорення та ефективності алгоритмів застосовуємо апроксимацію експериментальних даних.

Ключові слова: Похідні Капуто, Ядро, Ефективність, Фрактальні середовища, Похідні Грюнвальда-Летнікова, Тепло- і вологоперенесення, Паралелізація, Потік.