DOI: http://doi.org/10.25728/cs.2023.6.7

THE FUNCTIONAL VOXEL METHOD APPLIED TO SOLVING A LINEAR FIRST-ORDER PARTIAL DIFFERENTIAL EQUATION WITH GIVEN INITIAL CONDITIONS[1]

A.V. Tolok and N.B. Tolok

Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia

✉ tolok_61@mail.ru, ✉ nat_tolok@mail.ru

Abstract. This paper considers an approach to solving the Cauchy problem for a linear first-order partial differential equation by the functional voxel (FV) method. The approach is based on the principles of differentiation and integration developed for functional voxel modeling (FVM) and yields local geometrical characteristics of the resulting function at linear approximation nodes. A classical approach to solving the Cauchy problem for a partial differential equation is presented on an example, and an FV-model is built as a reference for further comparison with the FVM results. An algorithm for solving differential equations by FVM means is described. The FVM results are visually and numerically compared with the accepted reference. Unlike numerical methods for solving such problems, which give the values of a function at approximation nodes, the FV-model contains local geometrical characteristics at the nodes (i.e., gradient components in the space increased by one dimension). This approach allows obtaining an implicit-form nodal local function as well as an explicit-form differential local function.

Keywords: functional voxel modeling, partial differential equation, Cauchy problem, local function, local geometrical characteristics.

INTRODUCTION

Continuous processes in control systems can be often described by differential equations with initial conditions. For example, under a known input signal, the output signal is determined by the solution of the Cauchy problem for an ordinary differential equation.

The resulting function for a partial differential equation is not difficult to obtain and has long been provided by both analytical and numerical computer methods. However, the resulting function obtained manually or by means of an analytical computer-based calculator is a formulaic expression [1–5], whereas numerical methods produce numerical values at approximation grid nodes [6–9]. In this case, due to the absence of an analytical expression, the researcher cannot obtain, e.g., functions of partial derivatives for the available solution function, etc. The functional voxel (FV) method [10, 11] fills a given area of an analytical function with local functions describing a linear law for each minimal neighborhood of the area obtained during linear discretization. Hence, it becomes possible to apply not just the value at a point but the corresponding analytical expression in further calculations, with all the ensuing advantages.

The paper [12] considered the principles of differentiation and integration by functional voxel modeling (FVM) means. The transition to the FV-model of partial derivatives and back to the FV-model of the antiderivative is quite simple: an infinitesimal neighborhood of a point in a given (m – 1)-dimensional domain is described by the linear equation $n_1x_1 + n_2x_2 + \ldots + n_mx_m + n_{m+1} = 0$, where the coefficients are the components of the unit vector of the gradient with dimension (m + 1) increased by one. For computer representation, each component is encoded by a numerical value of the color palette, forming a sepa-

[1] This study was carried out within the scientific program of the National Center for Physics and Mathematics, direction No. 9 “Artificial Intelligence and Big Data in Technical, Industrial, Natural, and Social Systems.”

rate (m – 1)-dimensional image M_i. As a result, to describe and store a given area of the space E^3 (i.e., $u = f(x, y)$) on the computer, one needs to form four 2D bitmap images (M_1, M_2, M_3, M_4).

In this case, it suffices to set initial conditions (formulate the Cauchy problem) to construct the FV-model of an antiderivative. Based on the results of [12], we apply the FV method and implement an FV-model for solving a partial differential equation with given initial conditions.

1. PROBLEM STATEMENT

To demonstrate the algorithm, let us consider an example of solving a homogeneous partial differential equation of the form [13]

$$\frac{\partial u}{\partial x} + \left(e^{-x} - y\right)\frac{\partial u}{\partial y} = 0 \qquad (1)$$

with the initial condition

$$u(0, y) = 3y + 2. \qquad (2)$$

The differential equation (1) has the analytical solution

$$u(x, y) = 3(e^x y - x) + 2. \qquad (3)$$

Here,

$$\frac{\partial u}{\partial x} = 3(e^x y - 1), \qquad (4)$$

$$\frac{\partial u}{\partial y} = 3e^x. \qquad (5)$$

Figure 1 shows the graph of function (3) on the domain $x \in [0, 1]$, $y \in [0, 1]$, obtained by conventional visualization in MathCAD with a sampling step of 1/30.

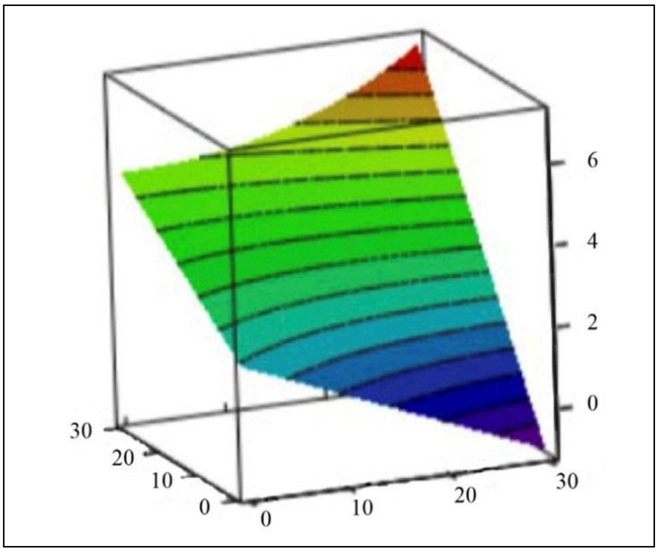


Fig. 1. The graph of function (3) in MathCAD.

By visual analysis of Fig. 1, it is possible to determine approximate values at the corner points of the surface segment under consideration. Their precise values are (0; 0; 2), (1; 0; –1), (0; 1; 5), and (1; 1; 7.1584).

Figures 2a and 2b present the graphs of functions (4) and (5), respectively, as the partial derivatives of function (3). We calculate their values at the corner points of the surface segments:

– for function (4), (0; 0; –3), (1; 0; –3), (0; 1; 0), and (1; 1; 5.1584);

– for function (5): (0; 0; 3), (0; 1; 3), (1; 0; 8.15484), and (1; 1; 8.15484).

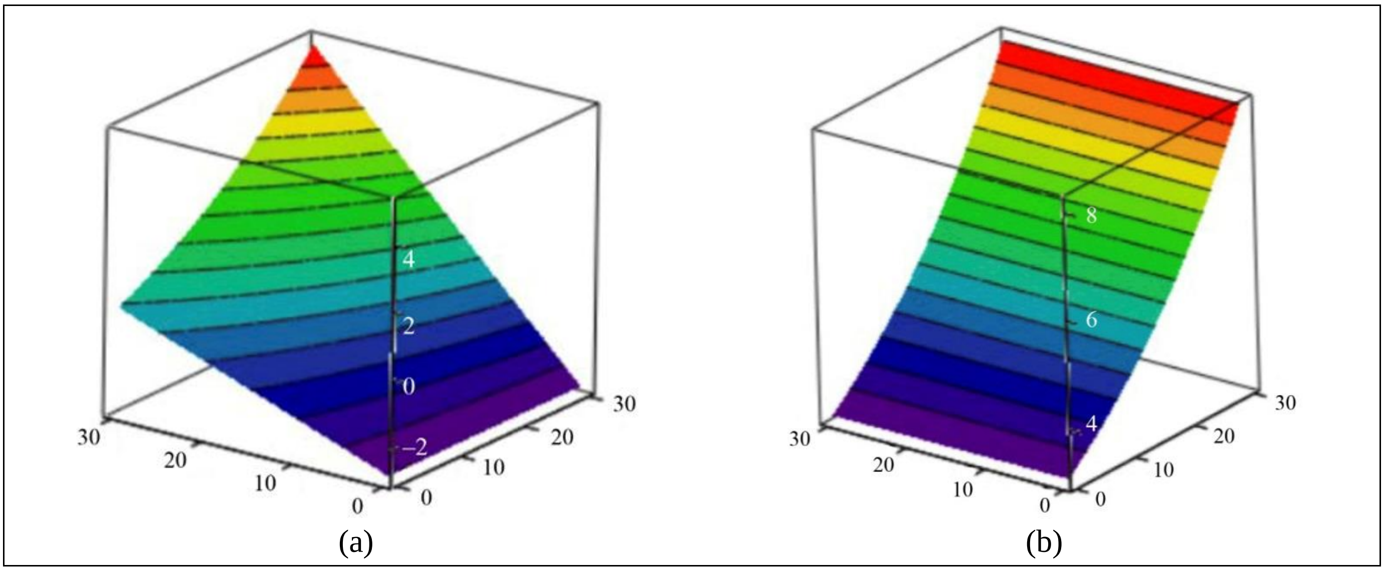


Fig. 2. The graphs of functions (4) and (5) in MathCAD.

The computer algorithm for obtaining the domains of local functions can be described as follows.

Step 1. A rectangular grid is applied to a given domain of the function for further linear approximation. The dimension of the grid space coincides with that of the function domain. The grid step is determined by the ratio of the size of the function domain to the size of the corresponding sides of the graphical image.

Step 2. During linear approximation on the function domain, the current grid element (the node and its nearest neighbors) is sequentially determined, i.e. the simplest element of the corresponding dimension is formed. For example, for a function of two variables $u = f(x, y)$, we have a triangular approximation element in which the neighbors are the nearest grid nodes shifted parallel to the axes Ox and Oy. For a function of three variables $u = f(x, y, z)$, the approximation element is a tetrahedron with a node and neighbors shifted parallel to the axes Ox, Oy, and Oz, respectively, etc.

Step 3. The local equation for the selected approximation element is obtained using the determinant of the matrix consisting of the homogeneous coordinates of the triangle nodes and the variable point on the domain. For example, in the case $u = f(x, y)$, the determinant of the matrix of dimensions 4 × 4 has the form

$$\begin{vmatrix} x & y & u & 1 \\ x_1 & y_1 & u_1 & 1 \\ x_2 & y_2 & u_2 & 1 \\ x_3 & y_3 & u_3 & 1 \end{vmatrix} = ax + by + cu + d = 0. \qquad (6)$$

Note that the matrix determinant allows obtaining such an equation for any matrix dimensions. The coefficients a, b, c, and d represent the components of the gradient vector increased by one dimension; the original function $u = f(x, y)$ can be replaced by a local function of the form

$$u = -\frac{a}{c}x - \frac{b}{c}y - \frac{d}{c}$$

since the plane given by equation (6) passes through the node under consideration.

Step 4. There is no sense in storing the local function for each point of the domain on the computer. It suffices to store the coefficients a, b, c, and d in the form of four bitmap images. This representation provides visual clarity of the data, which will be further used to assess the solution, and their compact storage.

To proceed, we normalize each coefficient by the length $N = \sqrt{a^2 + b^2 + c^2 + d^2}$ of the gradient vector, obtaining the components of the unit normal:

$$\vec{n} = (n_1, n_2, n_3, n_4),$$

where $n_1 = a / N$, $n_2 = b / N$, $n_3 = c / N$, and $n_4 = d / N$.

The color at the image point is defined as

$$M_i = \frac{P(1 + n_i)}{2},$$

where $P = 256$ and $i = 1, ..., 4$.

The inverse transition from the color value M_i to the component n_i is performed by the formula

$$n_i = \frac{2M_i - P}{P}.$$

Further, applying the FV method to function (3), we obtain a computer FV representation, i.e., the domain of local functions of the form $n_1x + n_2y + n_3u + n_4 = 0$, where n_1, n_2, n_3, and n_4 are the coefficients of the local function (local geometrical characteristics). They are displayed on the computer by *M*-images M_1, M_2, M_3, and M_4 (Fig. 3) with a resolution of 400 × 400. In [10], *M*-images were understood as image models displaying in tone or color one of the local geometrical characteristics of the FV-model. The accuracy of representing a numerical value by half-tint is provided in the RGB format (256 color grades). To increase clarity, we demonstrate the *M*-images for the color palette *P* = 16 777 214 (256 × 256 × 256) color grades in Fig. 4. The resulting patterns characterize the transition from red color grades through green color ones to blue color grades, providing higher visibility due to the resulting patterns for comparing the result with the reference. In our case, the reference is the *M*-images in Figs. 3 и 4.

At this stage, we assume that there is sufficient initial information for the numerical and visual experiment.

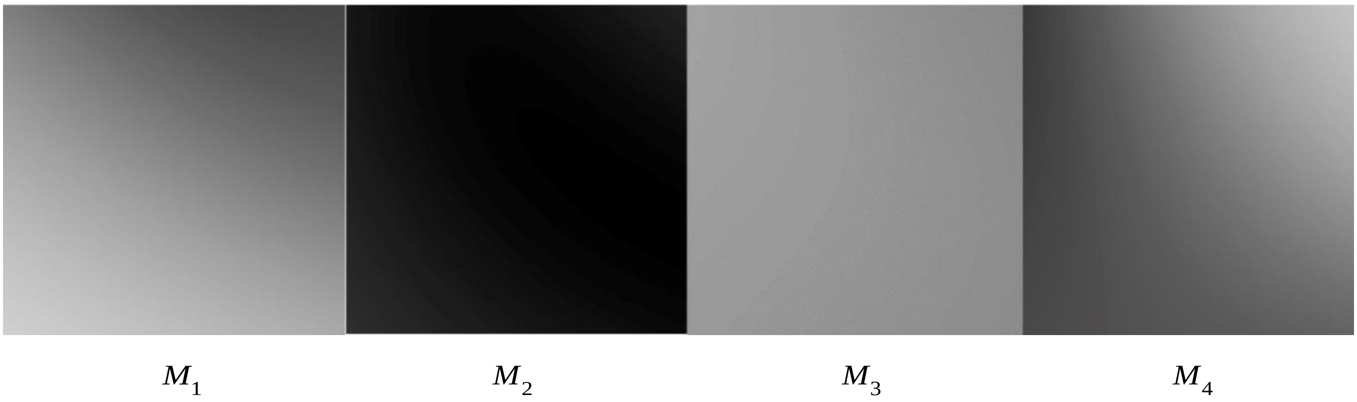


Fig. 3. The graphical representation of the local geometrical characteristics of function (3) (256 greyscale grades).

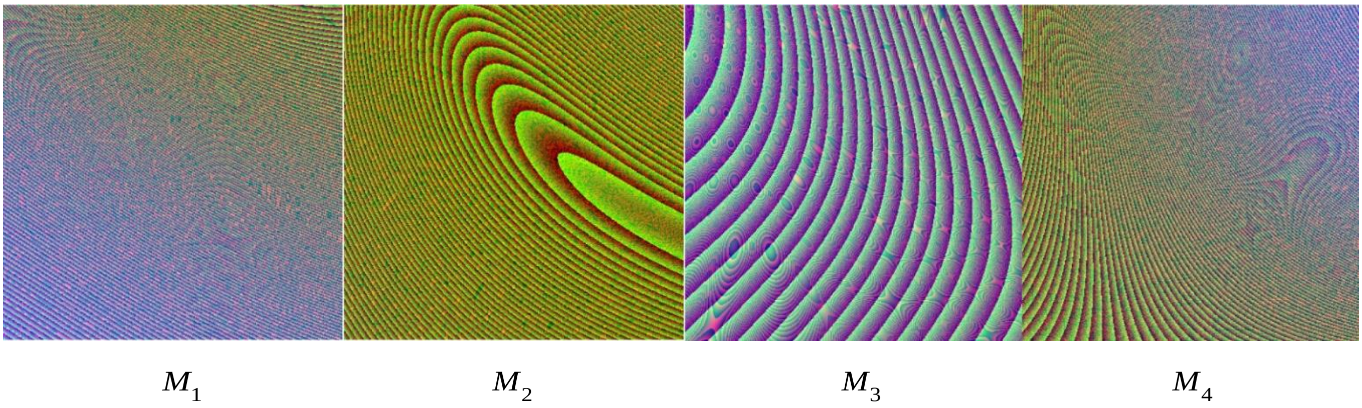


Fig. 4. The graphical representation of the local geometrical characteristics of function (3) (16 777 214 color grades).

2. THE ALGORITHM FOR BUILDING AN FV-MODEL TO SOLVE DIFFERENTIAL EQUATIONS

The paper [8] presented an algorithm for obtaining the FV-model of the antiderivative of a function by given FV-models of its partial derivatives. Note that it suffices to determine the local geometrical characteristics at one approximation point to calculate the function values at the other points of the triangular element of the approximation grid. Hence, local geometrical characteristics can be further found in the entire solution domain.

To apply this algorithm, we express the partial derivatives of a given function to obtain their exact values at the point under consideration.

In the example above, the initial condition is function (2). It represents the cross-section of the desired surface of function (3) for $x = 0$.

Therefore,

$$\frac{\partial u}{\partial y} \approx \frac{\Delta u}{\Delta y}, \ \Delta y = h,$$

$$\Delta u = u_{i+1}(0,\ (i+1)h) - u_{i+1}(0,\ ih),$$

$$i = [0 \ ... \ n],$$

where h is the approximation step.

Figure 2b shows the numerical data confirming the validity of this solution. Clearly, $\partial u / \partial y = 3$ at the point (0, 0) and $\partial u / \partial y = 8.15484$ at the point (49, 0). Along the axis Oy, the value of the derivative exponentially increases.

For $x = 0$, the partial derivatives can be defined as follows:

$$\frac{\partial u}{\partial x} \approx \frac{(e^{-x} - y)\Delta y}{\Delta u}.$$

With the local function of the FV-model written as $ax + by + cu + d = 0$, we obtain

$$\frac{\partial u}{\partial x} = -\frac{a}{c}, \ \frac{\partial u}{\partial y} = -\frac{b}{c},$$

where the coefficient c can be replaced by the approximation value C (Fig. 5):

$$C = x_1(y_2 - y_3) - x_2(y_1 - y_3) + x_3(y_1 - y_2).$$

Performing the transition to the components of the gradient vector, we have

$$A = -\frac{a}{c}C, \quad B = -\frac{b}{c}C,$$
$$D = -Ax_1 - By_1 - Cu_1.$$

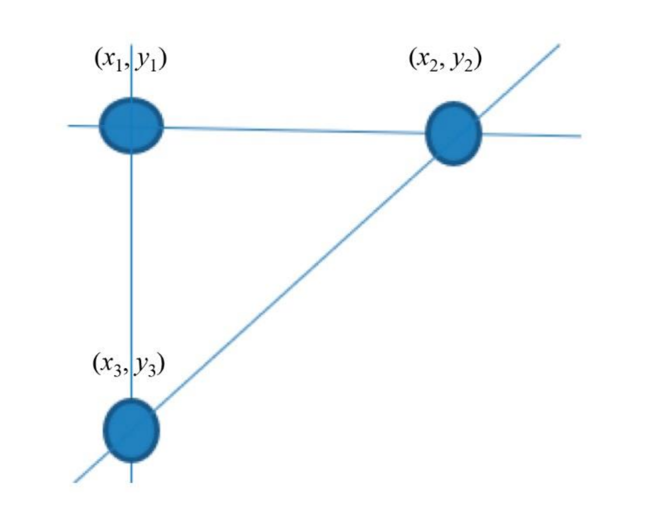


Fig. 5. Approximation nodes.

At the first calculation step, the value u_1 is given by formula (3); in other cases, the values are obtained through the next coefficients for the local function:

$$u_i = -\frac{A}{C}x_i - \frac{B}{C}y_i - \frac{D}{C}.$$

The algorithm for solving the differential equation includes the following basic steps:

1. Select the next triangular approximation element $(x_0, y_0, x_1, y_1, x_2, y_2)$.

2. Calculate the coefficient

$$C = x_0(y_1 - y_2) - x_1(y_0 - y_2) + x_2(y_0 - y_1).$$

3. For $x = 0$, calculate one of the derivatives according to given conditions (in the example above, $\partial u / \partial x = ((3y_2 + 2) - (3y_0 + 2)) / \Delta y$). For other values x, the derivative is determined by the relation $\partial u / \partial x = \Delta u / \Delta y$, where $\Delta u = (u_1 - u_0)$ and

$$u_0 = -\frac{A_0}{C_0}x_0 - \frac{B_0}{C_0}y_0 - \frac{D_0}{C_0},$$

$$u_1 = -\frac{A_1}{C_1}x_1 - \frac{B_1}{C_1}y_1 - \frac{D_1}{C_1}.$$

4. Calculate the second derivative based on the first derivative (in the example, $\partial u / \partial x = -(e^{-x_0} - y_0)(\Delta u / \Delta y)$).

5. Calculate the coefficients A_i, B_i, and D_i.

6. Pass to the (i + 1)th triangular element.

For each node of the approximation triangular grid, the local geometrical characteristics are successively calculated by FVM [6, 7], and the solution domain of the desired differential equation is filled with the local functions $n_1x + n_2y + n_3u + n_4 = 0$. On the computer, such a domain is represented by the corresponding images M_1, M_2, M_3, and M_4; see Fig. 6 (256 greyscale grades) and Fig. 7 (16 777 215 RGB color grades).

The result in Figs. 6 and 7 is visually comparable with that in Figs. 3 и 4. This confirms the adequacy of the algorithm. The numerical estimates of the nodal values of the function and its partial derivatives at the corner points of the domain $x \in [0, 1]$, $y \in [0, 1]$ are presented in the table.

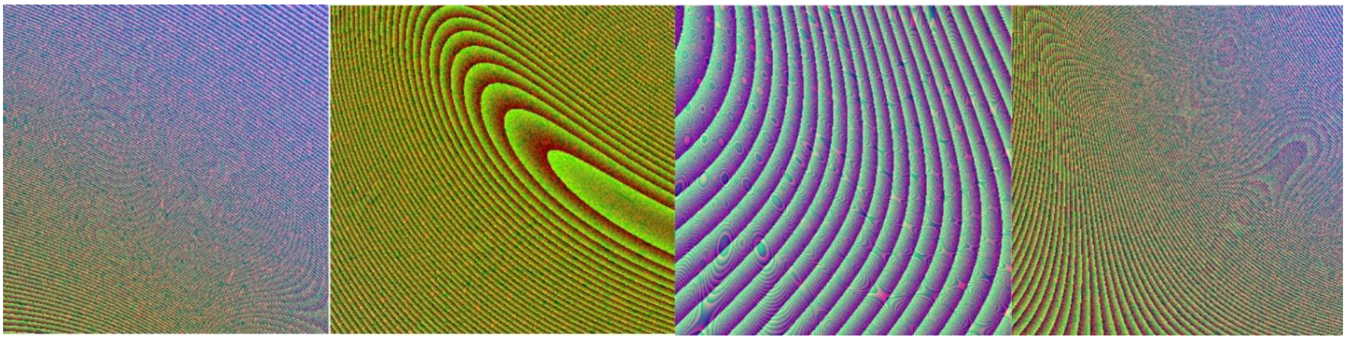
The nodal values of function (3) and its partial derivatives (numerical estimates)

x	y	u	$\frac{\partial u}{\partial x}$	$\frac{\partial u}{\partial y}$
0	0	2.0000	–3.0000	3.0000
0	399	5.0000	0.0000	3.0000
49	0	0.0235	–2.0279	7.9164
49	399	7.1486	4.9453	7.9164

Let us compare the points for the corresponding M-images with the accepted references. According to the comparison results, among 640 054 points of the image, the number of points with a value differing at most by unity is, respectively, $M_1 = 9515$, $M_2 = 3254$, $M_3 = 2116$, and $M_4 = 6086$ (not more than 1.5%).

M_1 M_2 M_3 M_4

Fig. 6. The graphical representation of the local geometrical characteristics of the differential equation solution (256 greyscale grades).



M_1 M_2 M_3 M_4

Fig. 7. The graphical representation of the local geometrical characteristics of the differential equation solution (16 777 215 color grades).

The resulting solution is a linear local function represented by the local geometrical characteristics for the points of the selected domain:

$$n_1 x + n_2 y + n_3 u + n_4 = 0.$$

Expressing $u\ (x,\ y)$, we obtain the local differential equation

$$u = -\frac{n_1}{n_3}x - \frac{n_2}{n_3}y - \frac{n_4}{n_3} \text{ or } u = \frac{\partial u}{\partial x}x + \frac{\partial u}{\partial y}y - \frac{n_4}{n_3}.$$

CONCLUSIONS

This paper has considered an approach to solving a linear first-order partial differential equation using the functional voxel method. An algorithm for solving such differential equations based on the proposed approach has been presented. The numerical simulation results have confirmed the adequacy of this algorithm.

In future studies, this algorithm will be compared with well-known numerical methods in terms of the growing error of the function value at the approximation grid nodes with different steps. Also, the accuracy of local geometrical characteristics will be compared with the approximation accuracy of the analytical solution.

REFERENCES

1. Egorov, A.I., *Obyknovennye differentsial'nye uravneniya s prilozheniyami* (Ordinary Differential Equations with Applications), Moscow: FIZMATLIT, 2003. (In Russian.)
2. Romanko, V.K., *Kurs differentsial'nykh uravnenii i variatsionnogo ischisleniya* (A Course in Differential Equations and Calculus of Variations), Moscow: Laboratoriya bazovykh znanii, 2000. (In Russian.)
3. Stepanov, V.V., *Kurs differentsial'nykh uravnenii* (A Course in Differential Equations), Moscow: Editorial URSS, 2004. (In Russian.)

4. Mathematical Analysis. Online Calculator. URL: https://allcalc.ru/node/863. (Accessed June 27, 2023.)
5. Microsoft Math Solver. URL: https://mathsolver.microsoft.com. (Accessed June 27, 2023.)
6. Krainov, A.Yu. and Moiseeva, K.M., *Chislennye metody resheniya kraevykh zadach dlya obyknovennykh differentsial'nykh uravnenii* (Numerical Methods for Solving Boundary-Value Problems for Ordinary Differential Equations), Tomsk: STT, 2016. (In Russian.)
7. Myshenkov, V.I. and Myshenkov, E.V., *Chislennye metody. Ch. 2. Chislennoe reshenie obyknovennykh differentsial'nykh uravnenii* (Numerical Methods. Part 2: Numerical Solution of Ordinary Differential Equations), Moscow: Moscow State University of Forestry, 2005. (In Russian.)
8. Soundararajan, R., Subburayan, V., and Wong, P.J.Y., Streamline Diffusion Finite Element Method for Singularly Perturbed 1D-Parabolic Convection Diffusion Differential Equations with Line Discontinuous Source, *Mathematics*, 2023, vol. 11, art. no. 2034. DOI: https://doi.org/ 10.3390/math11092034.
9. Mohammed, M., Well-Posedness for Nonlinear Parabolic Stochastic Differential Equations with Nonlinear Robin Conditions, *Symmetry*, 2022, vol. 18, art. no. 1722.
10. Tolok, A.V., *Funktsional'no-voksel'nyi metod v komp'yuternom modelirovanii* (The Functional Voxel Method in Computer Modeling), Moscow: Fizmatlit, 2016. (In Russian.)
11. Tolok, A.V., *Lokal'naya komp'yuternaya geometriya* (Local Computer Geometry), Moscow: IPR-Media, 2022. (In Russian.)
12. Tolok, A.V. and Tolok, N.B., Differentiation and Integration in Functional Voxel Modeling, *Control Sciences*, 2022, no. 5, pp. 51–57.
13. Konev, V.V., *Uravneniya v chastnykh proizvodnykh* (Partial Differential Equations), Tomsk: Tomsk Polytechnic University, 2011. URL: https://portal.tpu.ru/SHARED/k/KONVAL/notes/Partial.pdf. (In Russian.)

This paper was recommended for publication by V.G. Lebedev, a member of the Editorial Board.

Received July 28, 2023,
and revised October 3, 2023.
Accepted October 26, 2023.

Author information

Tolok, Alexey Vyacheslavovich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ tolok_61@mail.ru
ORCID iD: https://orcid.org/0000-0002-7257-9029

Tolok, Nataliya Borisovna. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ nat_tolok@mail.ru
ORCID iD: https://orcid.org/0000-0002-5511-4852

Cite this paper

Tolok, A.V. and Tolok, N.B., The Functional Voxel Method Applied to Solving a Linear First-Order Partial Differential Equation with Given Initial Conditions. *Control Sciences* **6,** 65–71 (2023). http://doi.org/10.25728/cs.2023.6.7

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com