DOI: http://doi.org/10.25728/cs.2022.5.5

DIFFERENTIATION AND INTEGRATION IN FUNCTIONAL VOXEL MODELING

A.V. Tolok[1] and N.B. Tolok[2]

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

[1]✉ tolok_61@mail.ru, [2]✉ nat_tolok@mail.ru

Abstract. This paper presents a simple method for generating the partial derivatives of a multidimensional function using functional voxel models (FV-models). The general principle of constructing, differentiating, and integrating an FV-model is considered for two-dimensional functions. Integration is understood as obtaining local geometrical characteristics for the antiderivative of a local function with solving the Cauchy problem when finally constructing the FV-model. The direct and inverse differentiation algorithm involves the basic properties of the local geometrical characteristics of functional voxel modeling and the inherent linear approximation principle of the codomain of the algebraic function. Simple computer calculations of this algorithm yield an FV-model suitable for any further algebraic operations. An illustrative example of constructing a functional voxel model of a complex two-dimensional algebraic function is provided. Functional voxel models of partial derivatives are obtained based on this model. These models and the boundary condition at a given point are used to obtain an initial FV-model of a complex algebraic function. The approach is applicable to algebraic functions defined on the domain of various dimensions.

Keywords: functional voxel model, local geometrical characteristics, local function, partial derivative, antiderivative.

INTRODUCTION

Differential calculus is still topical: it underlies almost all theoretical mechanics and mathematical physics as well as control theory. Nowadays, there exists a developed mathematical apparatus based on formal partial differentiation and derivation of integrand expressions to solve the inverse problem. Rather complex solutions are formed using the tables of known antiderivatives for various-type simple expressions and integration rules. Many attempts were undertaken to automate this process and obtain equations for further calculations [1–6]. In this case, the computer acts as a calculator without acquiring any "intelligent skills." The main problem is the inapplicability of such approaches to complex differentiable functions with peaks and discontinuities. Such functions arise in *R*-functional modeling and actively participate in the analytical modeling of geometric models to describe different objects and continuous processes. Among some examples, we mention a function describing a rectangular or polygonal zero contour, etc. For example, the following expression describes the positive domain of a rectangle with sides *a* and *b*:

$$a^2 + b^2 - x^2 - y^2 - \sqrt{\left(a^2 - x^2\right)^2 + \left(b^2 - y^2\right)^2} \geq 0. \quad (1)$$

Numerical methods based on discrete calculus have much contributed to automating the process of differentiation (the method of differences) and integration (the method of trapezoids, etc.). The problem grows sharply when increasing the dimensionality, especially with respect to the automation of expressions. In numerical methods, all arguments of a function become constants, except for the argument of differentiation, and the required order of the derivative is achieved by successive differentiation. However, numerical methods have an obvious disadvantage: their result is the value of the derivative at a point, not its algebraic function, which is required for solving the inverse integration problem [7–9].

Thus, the approaches discussed above cannot generally provide an automated solution of the direct and inverse differentiation problems.

We consider a developing computer method called functional voxel modeling (FVM). This method is intended for the discrete computer representation of continuous functions on a given multidimensional domain. It involves local geometrical characteristics (LGCs) on a given domain of an algebraic function. FVM was described in detail in [10, 11]. This method is based on the computer representation of the domain of local functions that replace the given domain of an algebraic function at each point. In contrast to numerical methods, the result at a point is not a numerical value but a function of simple linear form. For a two-dimensional complex algebraic function as one example, we consider the fundamental principle of obtaining the domain of local linear functions in a functional voxel computer model (FV-model) and the main differential operations performed to obtain the derivatives and the antiderivative.

1. THE FUNCTIONAL VOXEL MODELING OF AN ALGEBRAIC FUNCTION

As an illustrative example of obtaining an FV model, we consider the smooth function

$$u = x\sin\left(\pi\frac{y}{k}\right) + y^2\cos\left(\pi\frac{x}{k}\right), \tag{2}$$

on the domain [–1, 1] × [–1, 1] in the space xOy, where the coefficient k takes any value, e.g., 0.5.

This example of a continuous and smooth function provides a mathematical solution of partial derivatives for comparing FV-models.

We apply a regular rectangular grid with a cell spacing of 0.02 to the domain of the function. Let the nodes be numbered as in Fig. 1 to form a group of nodes of the triangular grid segment.

For the given coordinates of the three points, the determinant-based equation of the plane has the form

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = ax + by + cz + d = 0,$$

where

$$a = y_1(z_2 - z_3) - y_2(z_1 - z_3) + y_3(z_1 - z_2),$$

$$b = -(x_1(z_2 - z_3) - x_2(z_1 - z_3) + x_3(z_1 - z_2)),$$

$$c = x_1(y_2 - y_3) - x_2(y_1 - y_3) + x_3(y_1 - y_2),$$

$$d = -(x_1(y_2z_3 - y_3z_2) - x_2(y_1z_3 - y_3z_1) + x_3(y_1z_2 - y_2z_1)). \tag{3}$$

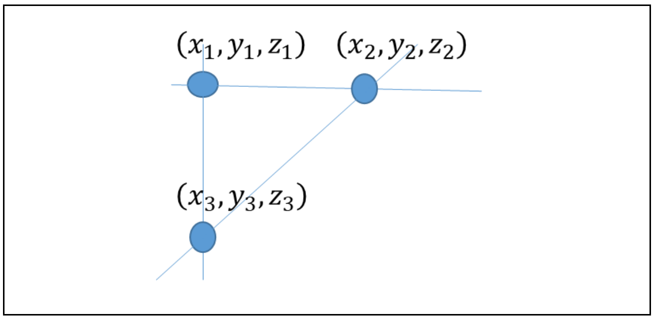


Fig. 1. The nodes of an approximation grid.

Next, we normalize the coefficients by the length of the four-dimensional gradient vector:

$$n_1 = \frac{a}{\sqrt{a^2+b^2+c^2+d^2}},$$

$$n_2 = \frac{b}{\sqrt{a^2+b^2+c^2+d^2}},$$

$$n_3 = \frac{c}{\sqrt{a^2+b^2+c^2+d^2}},$$

$$n_4 = \frac{d}{\sqrt{a^2+b^2+c^2+d^2}}.$$

Let the color gradation values of the monochrome palette P be associated with the values of the normal components (LGCs) as follows:

$$M_i = \frac{P(1+n_i)}{2}, \ \left(P = 256,\ i = \overline{1,4}\right).$$

Figure 2 shows the M-images (image-models) of the FV-model color mapping for the corresponding domain of the local geometrical characteristics of function (2).

At each point of the domain, this data representation allows automatically producing a local function that duplicates function (1) but has the simplest possible form:

$$n_1x + n_2y + n_3z + n_4 = 0. \tag{4}$$

To illustrate the next steps of differentiation, we model the M-images for the partial derivatives of function (2) expressed traditionally:

$$\frac{\partial u}{\partial x} = \sin\left(\pi\frac{y}{a}\right) - y^2\frac{\pi}{a}\sin\left(\pi\frac{x}{a}\right), \tag{5}$$

$$\frac{\partial u}{\partial y} = x\frac{\pi}{a}\cos\left(\pi\frac{y}{a}\right) + 2y\cos\left(\pi\frac{x}{a}\right). \tag{6}$$

Figures 3 and 4 demonstrate the FV-models for equations (5) and (6), respectively. Each M-image visualizes the changes in the local geometrical characteristics forming the local function for each point.

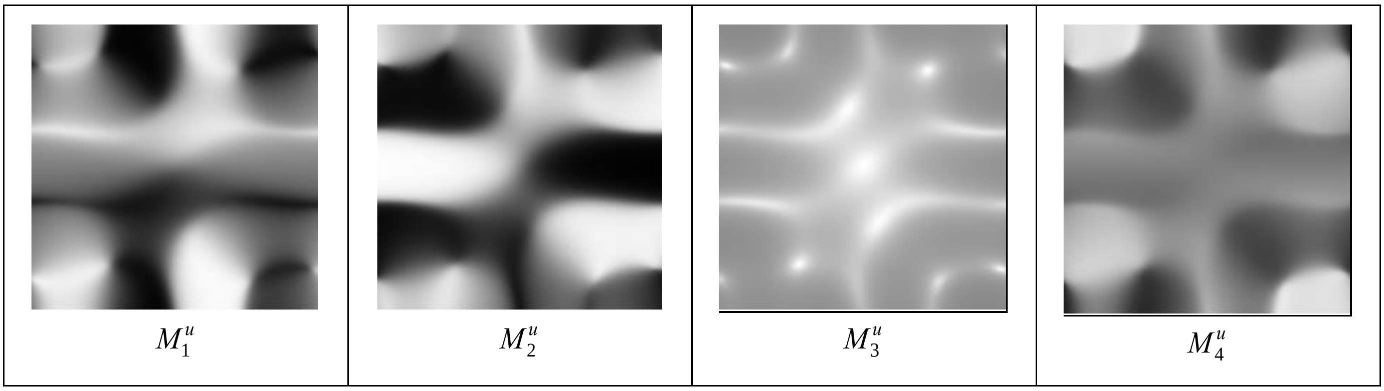


Fig. 2. The basic *M*-images of the function *u*.

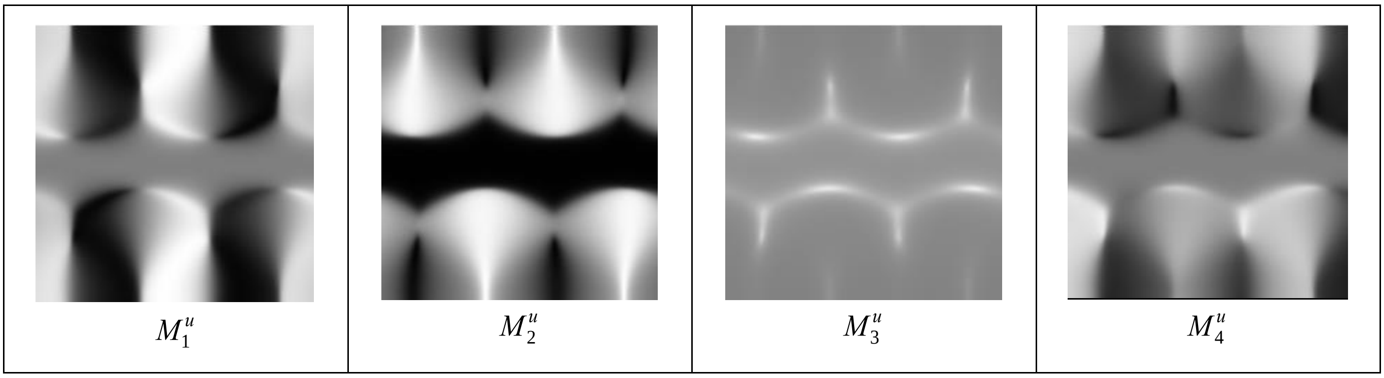


Fig. 3. The basic *M*-images of the function $\partial u / \partial x$.

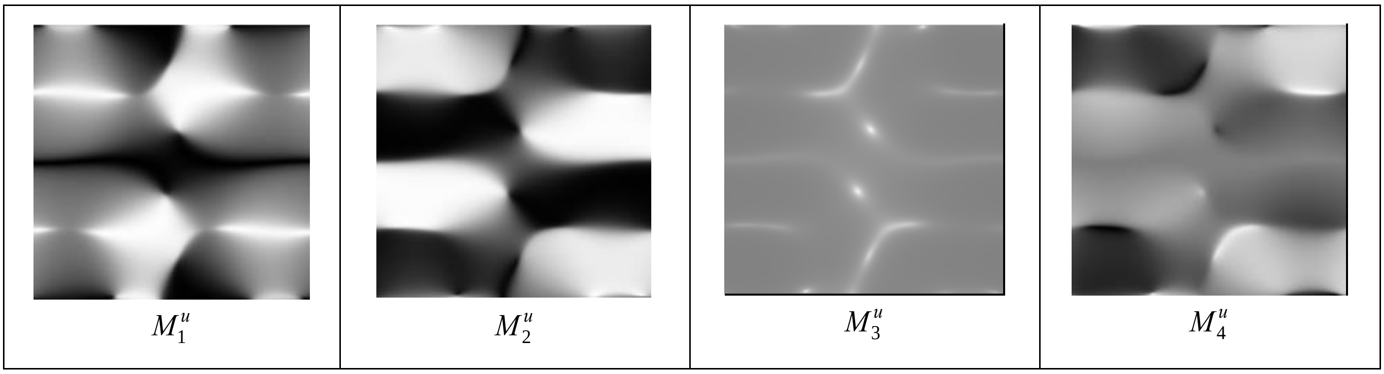


Fig. 4. The basic *M*-images of the function $\partial u / \partial y$.

2. PARTIAL DIFFERENTIATION OF FV-MODELS

We model the partial derivative along the axis Ox using the above algorithm and the relation

$$\frac{\partial u}{\partial x} = \frac{a}{c} = \frac{n_1}{n_3}, \tag{7}$$

where n_1 and n_3 are the coefficients of equation (4).

Having such values for each point of the domain with the *M*-images M_1^u and M_3^u (Fig. 2), we obtain a similar approximation scheme (Fig. 5).

For the three points, the determinant-based equation of the plane has the form

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & \left(\frac{n_1}{n_3}\right)_1 & 1 \\ x_2 & y_2 & \left(\frac{n_1}{n_3}\right)_2 & 1 \\ x_3 & y_3 & \left(\frac{n_1}{n_3}\right)_3 & 1 \end{vmatrix} = ax + by + cz + d = 0.$$

Figure 6 shows the *M*-images of the color mapping for the corresponding domain of the local geometrical characteristics of (7). Obviously, these *M*-images visually coincide with the ones in Fig. 3.

By analogy, we can obtain *M*-images for the derivative along the axis Oy (Fig. 7), where

$$\frac{\partial u}{\partial y} = \frac{b}{c} = \frac{n_2}{n_3}. \tag{8}$$

Consider an example of modeling the second derivative along the axis Ox. For this purpose, we differentiate function (3) using its FV-model (Fig. 6) and the local differentiation algorithm proposed above.

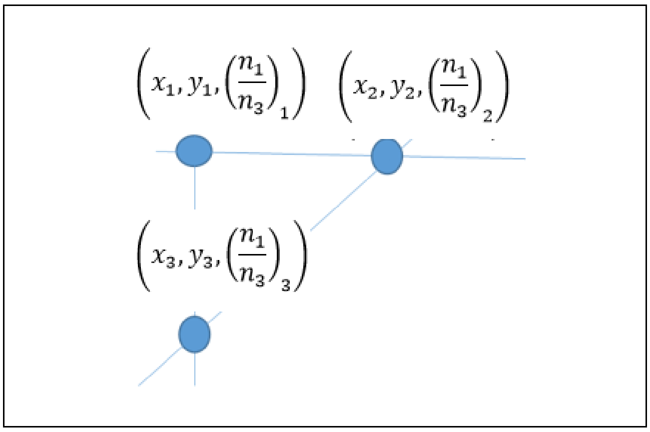


Fig. 5. The nodes of an approximation grid.

The resulting FV-models of the derivatives

$$\frac{\partial^2 u}{\partial x^2} \text{ and } \frac{\partial^2 u}{\partial x \partial y}$$

are presented in Fig. 8.

Thus, the FVM approach to differential images allows obtaining derivatives of different order without much difficulty.

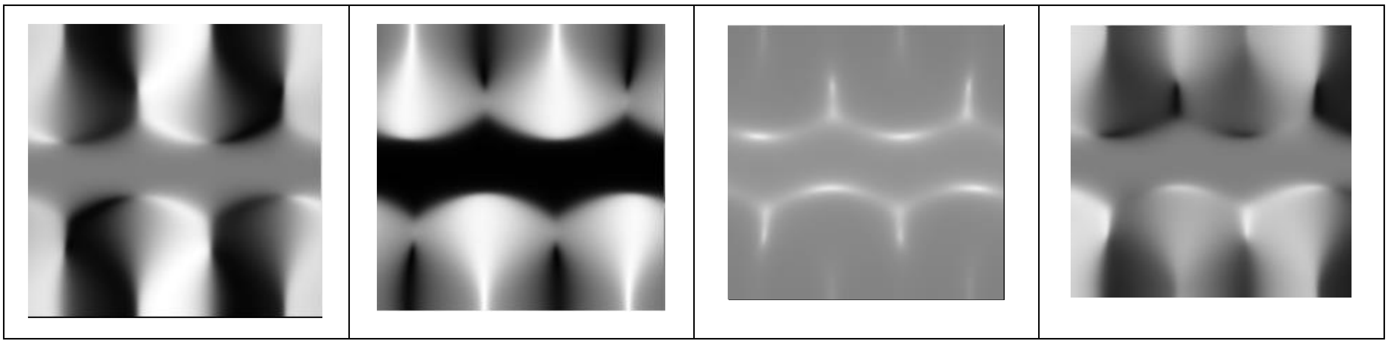


Fig. 6. The basic *M*-images of the function n_1 / n_3.

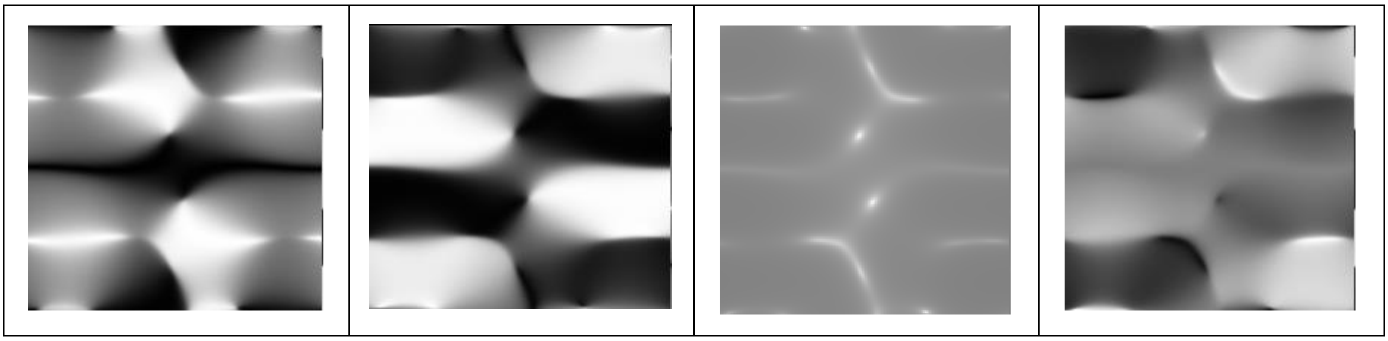
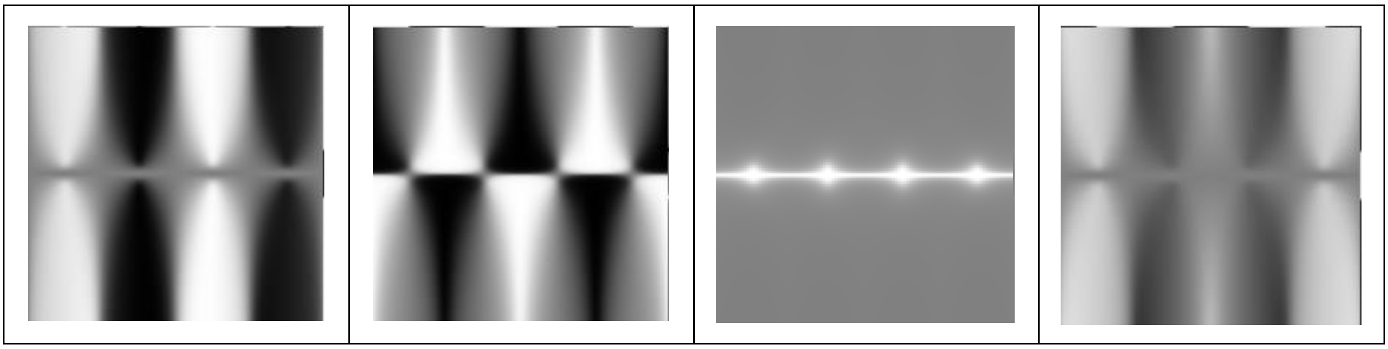
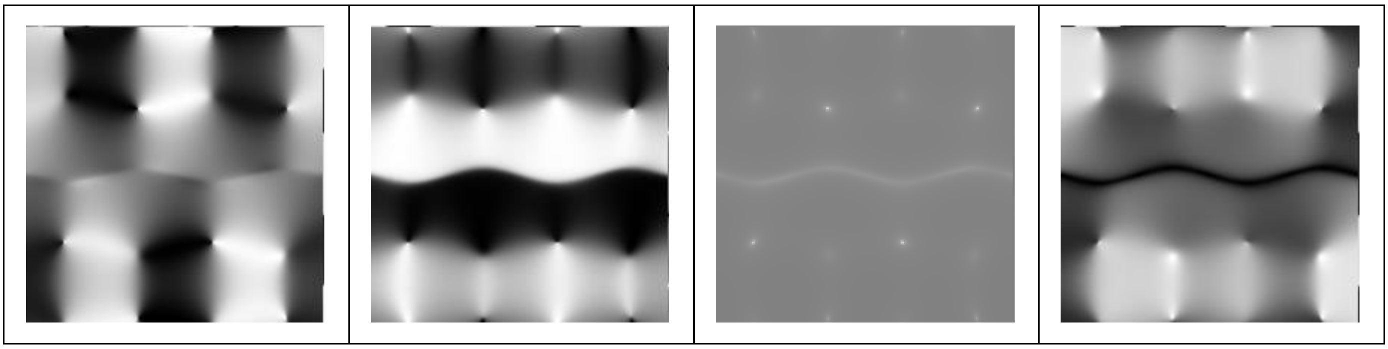


Fig. 7. The basic *M*-images of the function n_2 / n_3.



(a)



(b)

Fig. 8. The basic *M*-images of functions: (a) $\partial^2 u / \partial x^2$ and (b) $\partial^2 u / \partial x \partial y$.

3. INTEGRATION OF FV-MODELS

Consider the inverse differentiation problem: finding the antiderivative (integration). Let us refer to formulas (7) and (8), i.e., the equations

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

Note that the coefficient c in the denominators is the doubled area of the triangle with the vertices $(x_1, y_1), (x_2, y_2)$, and (x_3, y_3) in the plane xOy, i.e., is calculated by formula (3).

The coordinates of the nodes of the approximation grid are known or can be easily determined for the given domain of the function and the sizes of the *M*-images. Hence, we can calculate the coefficient c and then the coefficients a and b by the formulas

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

This leads to an indefinite local integral at the point (x_i, y_j):

$$ax + by + cz = 0.$$

To find the antiderivative, assume that $z_1 = f(x_1, \ y_1)$ is known; in other words, we calculate

$$z_1 = x_1 \sin\left(\pi \frac{y_1}{0.5}\right) + y_1^2 \cos\left(\pi \frac{x_1}{0.5}\right).$$

Then

$$d = -ax_1 - by_1 - cz_1.$$

Completing the definition of the coefficients of the local equation, we obtain the corresponding values for the other nodes of the approximation grid segment (Fig. 9):

$$z_2 = -\frac{a}{c} x_2 - \frac{b}{c} y_2 - \frac{d}{c},$$

$$z_3 = -\frac{a}{c} x_3 - \frac{b}{c} y_3 - \frac{d}{c}.$$

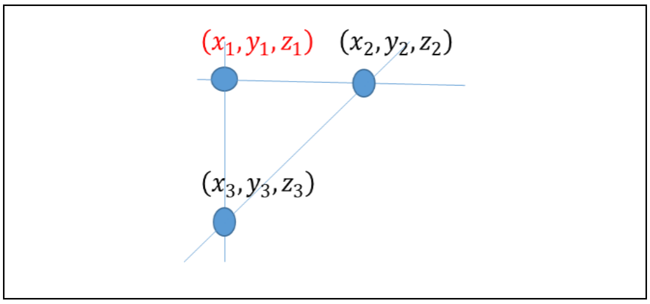


Fig. 9. The nodes of an approximation grid.

Now we apply the local integration algorithm to the second derivative. As expected, the resulting *M*-images should be as similar as possible to the *M*-images of the FV-model of the first derivative (Fig. 6). The initial *M*-images are the *M*-images obtained for the derivatives $\frac{\partial^2 u}{\partial x^2}$ and $\frac{\partial^2 u}{\partial x \partial y}$, presented in Figs. 7 and 8, respectively. The result of the local integration algorithm is shown in Fig. 10. These *M*-images visually coincide with the ones in Fig. 6.

The error in the resulting images is due to the loss of accuracy when passing to integer values of the palette. In many cases, this error is insignificant since the values differ by the third decimal place.

Applying the local integration algorithm to the *M*-images of the first derivative yields the result in Fig. 11. It is quite comparable to the *M*-images of the original function u; see the FV-model in Fig. 2.

Consider an example of differentiating function (1) using the proposed approach. Figure 12 demonstrates the *M*-images of the FV-model for the expression

$$u = a^2 + b^2 - x^2 - y^2 - \sqrt{\left(a^2 - x^2\right)^2 + \left(b^2 - y^2\right)^2}$$

with $a = 0.5$ and $b = 1$ on the domain $[-1, 1] \times [-1, 1]$.

The *M*-images of the partial derivative along the axis Ox are shown in Fig. 13.

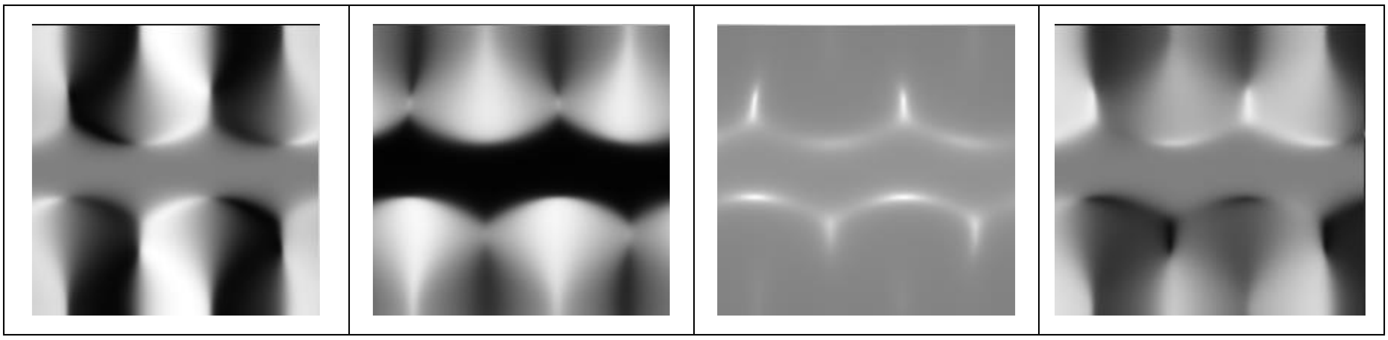


Fig. 10. The basic *M*-images of the second derivative of the function *u*.

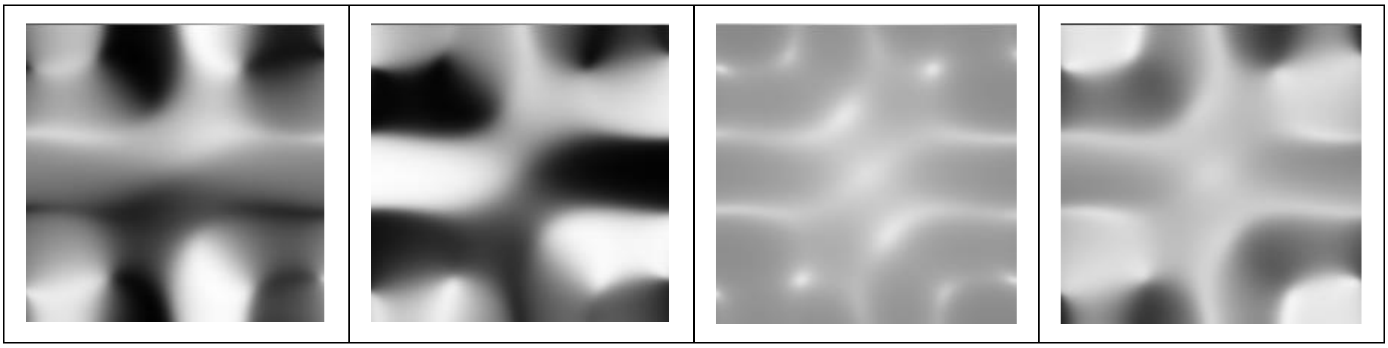


Fig. 11. The basic *M*-images of the integral of the second derivative of the function *u*.

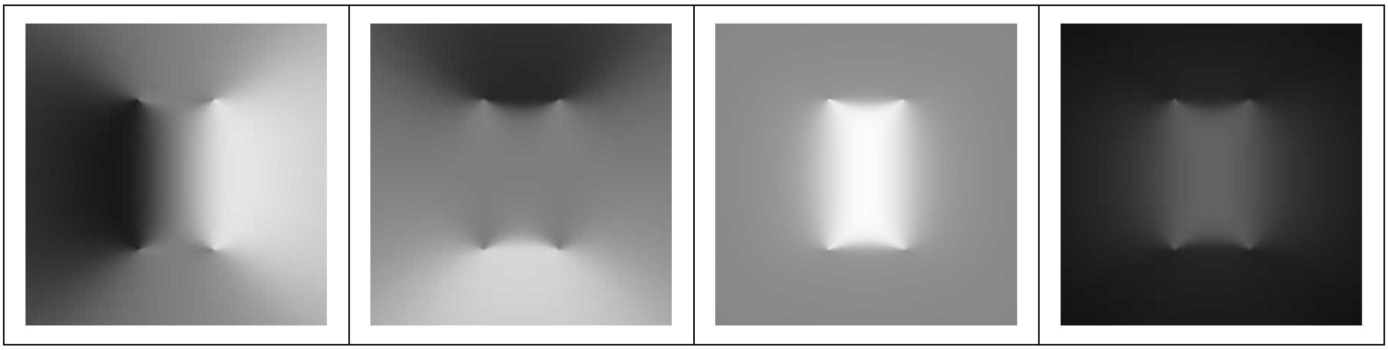


Fig. 12. The basic *M*-images of the integral of the second derivative of the function *u*.

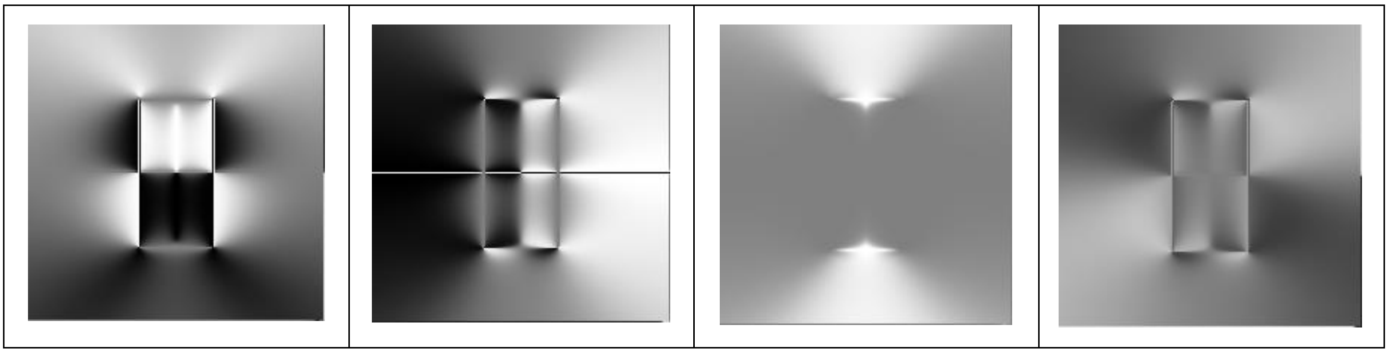


Fig. 13. The basic *M*-images of the first derivative of the function n_1/n_3.

CONCLUSIONS

This paper has presented a tool for automating the differentiation and integration of wide-range complex algebraic functions based on functional voxel modeling. Due to linear approximation, the proposed approach allows differentiating and integrating a wide class of undifferentiated functions that arise in *R*-functional modeling. Despite a visual error in the result, this approach is robust and ensures a solution even if the function has no mathematical form. An example of two-dimensional functions has been provided to demonstrate and visually compare the results. Note that the algorithm is easily transferable to any dimension.

REFERENCES

1. Rystsov, D., Automatic Differentiation, *Abnormal Programming, Mathematics*, Habr, 2009. http://habrahabr.ru/post/63055/ (In Russian.)
2. Neidinger, R.D., Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming, *SIAM Review*, 2010, vol. 52, no. 3, pp. 545–563. DOI:10.1137/080743627.
3. Baydin, A.G., Pearlmutter, B.A., Radul, A.A., and Siskind, J., Automatic Differentiation in Machine Learning: a Survey, *Journal of Machine Learning Research*, 2018, vol. 18, art. no. 153, pp. 1–43.
4. Rall, L.B., *Automatic Differentiation: Techniques and Applications*, Lecture Notes in Computer Science, vol. 120, Berlin–Heidelberg–New York, Springer-Verlag, 1981. ISBN 3-540-10861-0.
5. Conal, E., The Simple Essence of Automatic Differentiation, *Proc. ACM Program. Lang.*, 2018, vol. 2, iss. ICFP, art. no. 70, pp. 1–29. DOI: https://doi.org/10.1145/3236765.
6. Liao, H., Pearlmutter, B.A., Potluru, V.K., and Woodruff, D.P., Automatic Differentiation of Sketched Regression, *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics* (*AISTATS*), Palermo, Italy, 2020, PMLR, vol. 108, pp. 4367–4376.
7. Bakhvalov, N.S., Zhidkov, N.P., and Kobel'kov, G.M., *Chislennye metody* (Numerical Methods), 3rd ed., Moscow: BINOM. Laboratoriya znanii, 2004. (In Russian.)
8. Berezin, I.S. and Zhidkov, N.P., *Metody vychislenii. T. I* (Computing Methods. Vol. I), 2nd ed. (In Russian.)
9. Mysovskikh, I.P., *Lektsii po metodam vychislenii* (Lectures on Computing Methods), Moscow: Nauka, 1982. (In Russian.)
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.)

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

Received July 18, 2022, and revised September 15, 2022.
Accepted October 3, 2022.

Author information

Tolok, Alexey Vyacheslavovich. Dr. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ tolok_61@mail.ru

Tolok, Nataliya Borisovna. Cand. Sci. (Eng.), Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia
✉ nat_tolok@mail.ru

Cite this paper

Tolok, A.V. and Tolok, N.B., Differentiation and Integration in Functional Voxel Modeling. *Control Sciences* **5,** 51–57 (2022). http://doi.org/10.25728/cs.2022.5.5

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