Local R-Functional Modelling (LRFM)

Alexey Tolok [1] and Nataliya Tolok [1]

[1] *V.A. Trapeznikov Institute of Control Science of Russian Academy of Sciences, 65 Profsoyuznaya street, Moscow, 117997, Russia*

Abstract

A new class of functions - "FLOZ- functions" (Functions of LOcal Zeroing out), which makes it possible to form the zero domain of a scalar-valued multidimensional function of complex configuration by means of R-functional modelling is considered. We represent the solution of the inverse problem of analytical geometry for a non-convex contour construction obtained by V.L. Rvachev's mathematical apparatus of R-functions. The problems of constructing an algorithm for automation the proposed by V.L. Rvachev solutions are described. Presented arguments show the complexity of constructing an algorithm based on recursive attachment. The functional voxel model was created in the RANOK 2D system. An approach to the function of local zeroing out (FLOZ-function) construction for the general (multidimensional) case is described. A two-dimensional function of local zeroing out is selected for solving the problem of a non-convex contour constructing. It is shown that the function of local zeroing out allows to create the sequential algorithm of automation the non-convex contour construction. Examples of automation the considered problems of V.L. Rvachev to the non-convex contour construction are given. The function of local zeroing out for three-dimensional space (3D FLOZ-function) is considered. An example of functional voxel modelling of a 3D sphere model based on a triangulated network consisted of 80 triangles is given.

Keywords

R-functional modelling (RFM), function of local zeroing out (FLOZ-function), functional voxel modelling (FVM), M-image

1. Introduction

The modeling of geometric objects with scalar-valued functions initially attracted specialists primarily by the fact that such an object chiefly provides the maximum computational accuracy of scalar values and the completeness of its differential characteristics' representation at each point of a given space. Secondly, the spatial constraints imposed on a geometric object in the case of modeling by other approaches (such as, for example, parametric) can be removed now. In the 60s of the last century, V.L. Rvachev proposed a new class of functions (R-functions) [1] that allowed to carry out basic set-theoretic operations (intersection and union) over the domain of two scalar functions, and this opens up new possibilities for improving such approach to modeling.

However, nowadays the construction of scalar-valued functions remains not an easy task that requires a decent mathematical background and spatial intuition, allowing you to predict the formulation of the expected result. In 1982, in work [2] V.L.Rvachev defines the problem of automating the scalar-valued function construction by the example of describing a non-convex complex contour. For that purpose, the R-functional modeling algorithm must provide recursion depth when identifying convex domains that differ in their predicate statement (positive or negative region). Let us consider the principle of operation of such an algorithm using a specific example analyzed in the works of V.L. Rvachev.

GraphiCon 2021: 31st International Conference on Computer Graphics and Vision, September 27-30, 2021, Nizhny Novgorod, Russia
EMAIL: tolok_61@mail.com (A. Tolok); nat_tolok@mail.com (N. Tolok);
ORCID: 0000-0002-7257-9029 (A. Tolok); 0000-0002-5511-4852 (N. Tolok);

Figure 1 shows the circuit shown in [2] as an example. It contains fourteen vertices, where the inner space of the function for describing the contour must have the positive sign of scalar values. Let us simulate such an area according to the principle given in [2].

The half-plane defined by two consecutive points (x^i, y^i) and (x^{i+1}, y^{i+1}) in [2] is formulated as

$$\Sigma_i^2 = -x(y^{i+1} - y^i) + y(x^{i+1} - x^i) - y^i x^{i+1} + x^i y^{i+1}.$$

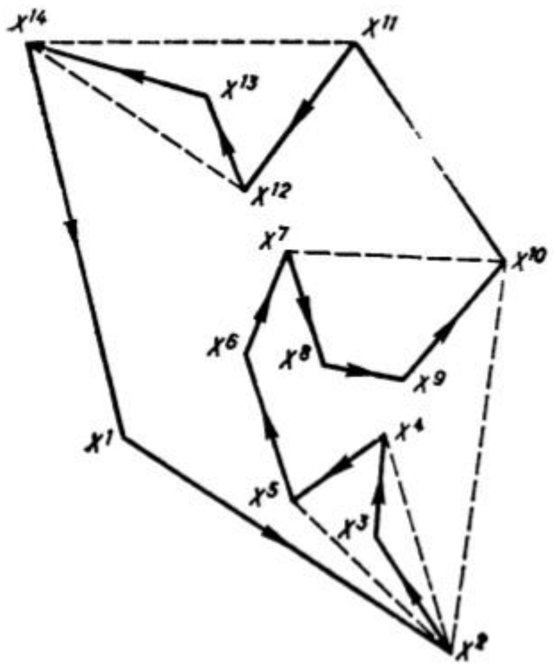

Figure 1: An example of a non-convex region, considered in [2]

The dashed lines complement the polygon $x^1x^2 \dots x^{14}x^1$ to a convex one with segments x^2x^{10} и $x^{11}x^{14}$. The positive half-planes of these segments Σ_{210}^2 and Σ_{1114}^2 (notation adopted by the author) are located to the left of the straight lines x^2x^{10} and $x^{11}x^{14}$. In this case, the convex polygon $x^1x^2x^{10}x^{11}x^{14}x^1$ is defined by the conjunction formula:

$$\Sigma_1^2 \cap \Sigma_{210}^2 \cap \Sigma_{10}^2 \cap \Sigma_{1114}^2 \cap \Sigma_{14}^2. \tag{1}$$

Embedding a convex polygonal region $x^{10}x^7x^6x^5x^2$ into the resulting region leads to the replacement of the symbol Σ_{210}^2 in the formula (1) with the disjunction:

$$\Sigma_{25}^2 \cup \Sigma_5^2 \cup \Sigma_6^2 \cup \Sigma_{710}^2. \tag{2}$$

So, the polygon area $x^1x^2x^5x^6x^7x^{10}x^{11}x^{14}x^1$ is defined by the formula

$$\Sigma_1^2 \cap (\Sigma_{25}^2 \cup \Sigma_5^2 \cup \Sigma_6^2 \cup \Sigma_{710}^2) \cap \Sigma_{10}^2 \cap \Sigma_{1114}^2 \cap \Sigma_{14}^2.$$

The process of sequentially cutting out the remaining convex polygonal areas will result in the following:

$$\Sigma_1^2 \cap \Big(\big((\Sigma_2^2 \cup \Sigma_3^2) \cap \Sigma_4^2 \big) \cup \Sigma_5^2 \cup \Sigma_6^2 \cup (\Sigma_7^2 \cap \Sigma_8^2 \cap \Sigma_9^2) \Big) \cap \Sigma_{10}^2 \cap \big(\Sigma_{11}^2 \cup (\Sigma_{12}^2 \cap \Sigma_{13}^2) \big) \cap \Sigma_{14}^2. \tag{3}$$

We will model the resulting expression in a specialized RANOK 2D system, which is provided by the language-oriented programming compiler for automated calculation of scalar values of the described function [3]. Below is the source code for describing a non-convex contour:

```
RECTANGLE(0,0,25,25)
RECTBMP(200,200)
ARGUMENT x,y
CONSTANT x1=4, y1=7, x2=16, y2=1,
            x3=14, y3=4, x4=15, y4=7,
            x5=12, y5=6, x6=11, y6=10,
            x7=13, y7=13, x8=14, y8=10,
            x9=17, y9=9, x10=20, y10=13,
            x11=14, y11=21, x12=10, y12=17,
                    x13=8, y13=20, x14=2, y14=22
    FUNCTION w1=-x*(y2-y1)+y*(x2-x1)-y1*x2+x1*y2
                    w2=-x*(y3-y2)+y*(x3-x2)-y2*x3+x2*y3
                    w3=-x*(y4-y3)+y*(x4-x3)-y3*x4+x3*y4
                    w4=-x*(y5-y4)+y*(x5-x4)-y4*x5+x4*y5
                    w5=-x*(y6-y5)+y*(x6-x5)-y5*x6+x5*y6
                    w6=-x*(y7-y6)+y*(x7-x6)-y6*x7+x6*y7
                    w7=-x*(y8-y7)+y*(x8-x7)-y7*x8+x7*y8
                    w8=-x*(y9-y8)+y*(x9-x8)-y8*x9+x8*y9
                    w9=-x*(y10-y9)+y*(x10-x9)-y9*x10+x9*y10
                    w10=-x*(y11-y10)+y*(x11-x10)-y10*x11+x10*y11
                    w11=-x*(y12-y11)+y*(x12-x11)-y11*x12+x11*y12
                    w12=-x*(y13-y12)+y*(x13-x12)-y12*x13+x12*y13
                    w13=-x*(y14-y13)+y*(x14-x13)-y13*x14+x13*y14
                    w14=-x*(y1-y14)+y*(x1-x14)-y14*x1+x14*y1
    w=w1&(((w2|w3)&w4)|w5|w6|(w7&w8&w9))&w10&(w11|(w12&w13))&w14
    RETURN w
```

Figure 2 shows the result of calculating the surface of a scalar function that provides a null boundary, marked in red.

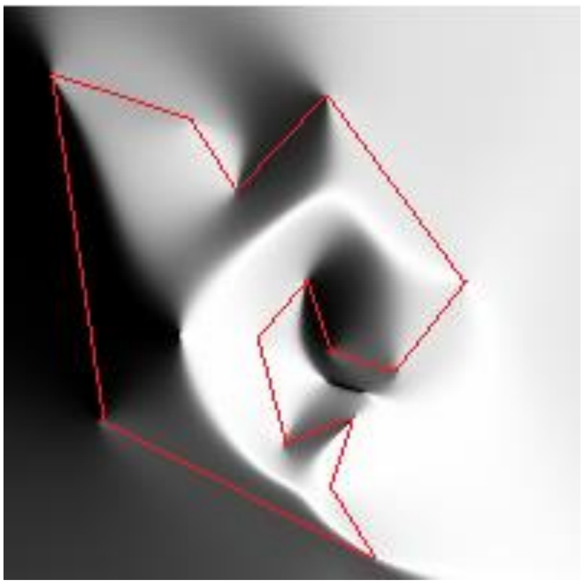

Figure 2: Construction of the surface of a scalar function by the method of V.L. Rvachev

The presented surface tends to monotonicity and smoothness throughout the entire interval of the range of values, despite the sharp corners of the given contour.

The disadvantage of this approach is the complex algorithmization of recursive data processing of a set of contour points when automating such an algorithm. It is required to implement some "simulator" of nested brackets of the obtained formula, which provides recursive immersion with dynamically increasing depth and constantly changing number of nodes of the next nested contour. This often leads to the organization of additional scripts, in fact, substituting for the work of the compiler, and therefore does not give application efficiency.

To get rid of the recursive construction of a complex non-convex contour, it is necessary to transform the algorithm into a simple sequence of computing a single logical operation (for example, an intersection). To do this, it is enough to construct a scalar function that calculates a zero value in a given limited area, and fills the rest of the area with positive values. Such a function should provide generality with the multidimensional space, forming its own class - the Function of Local Zeroing out (FLOZ-function).

2. Function of Local Zeroing out (FLOZ-function)

FLOZ-function – a multidimensional function that acquires zero for a local neighborhood defined by vertices on the range of positive values.

A local neighborhood of a FLOZ-function should be understood as a simple geometric figure, the number of vertices which corresponds to the dimension of the FLOZ space (Fig. 3). For example, in xOy space it is a segment defined by two points, in $Oxyz$ space it is a triangle, etc.

The neighborhood is defined by a local function:

$$L(X_m) = n_1x_1 + \cdots n_{m-1}x_{m-1} + n_m = 0. \tag{4}$$

A local function is understood as a linear polynomial whose coefficients are local geometric characteristics n_i, obtained by normalizing the coefficients a_i from the equation:

$$\begin{vmatrix} x_1 & \dots & x_m & 1 \\ x_1^1 & \dots & x_m^1 & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_1^n & \dots & x_m^n & 1 \end{vmatrix} = a_1x_1 + \cdots + a_mx_m + a_{m+1} = 0, \text{ so}$$

$$n_i = \frac{a_i}{\sqrt{a_1^2 + \cdots + a_{m+1}^2}}. \tag{5}$$

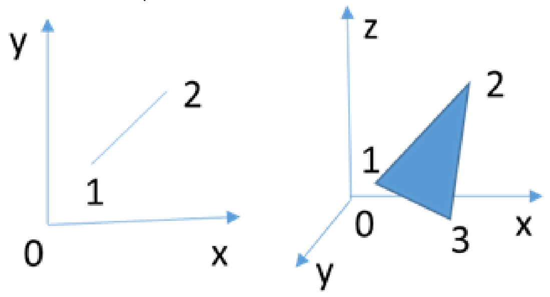

Figure 3: Local neighborhood of a FLOZ-function

The area (volume) of an n-dimensional neighborhood of FLOZ-function can be formulated in terms of the norm $\sqrt{a_1^2 + \cdots + a_m^2}$ as:

$$V_{fl}^n = \frac{\sqrt{a_1^2 + \cdots + a_m^2}}{(m-1)!}. \tag{6}$$

Successively replacing the coordinates of one of the control points x_i^j of the determinant with the coordinates of the considered point of the space $P_k(X_m), x_i \in X_m$, we obtain n terms of the neighborhood..

<u>Statement:</u> The sum of the folded neighborhoods $\sum_{i=0}^{m} V_{P_kX_m}^m$, obtained in turn by successive replacement of control points with a point P_k is equal to V_{fl}^m if P_k belongs to the local neighborhood of FLOZ. Then the function of local zeroing out (FLOZ-function) in general form can be written as:

$$Floz(X_m) = \sum_{i=0}^{m} V_{P_i(X_m)}^m - V_{fl}^m = 0. \tag{7}$$

We will show how FLOZ-functions work using the examples of the implementation of simple spaces E^2 and E^3.

1. The space E^2, that is, the two vertices define a line segment. The equation of a straight line to which the segment belongs can be written in such form:

$$\begin{vmatrix} x & y & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} = (y_1 - y_2)x - (x_1 - x_2)y + (x_1y_2 - y_1x_2) = a_1x + a_2y + a_3 = 0. (8)$$

In turn, the length of the segment according to the general formulation V_{fl}^{n} can be written as

$$V_{fl}^{2} = \frac{\sqrt{a_1^2 + a_2^2}}{(2-1)!} = \sqrt{(y_1 - y_2)^2 + (x_1 - x_2)^2} = d. \qquad (9)$$

In this case

$V_{P_1}^{2} = \sqrt{(y - y_2)^2 + (x - x_2)^2} = d_1$ и $V_{P_2}^{2} = \sqrt{(y_1 - y)^2 + (x_1 - x)^2} = d_2$, a

$$\boldsymbol{Floz}(\boldsymbol{X_2}) = d_1 + d_2 - d = 0. \qquad (10)$$

Figure 4 clearly demonstrates the principle of operation of a two-dimensional FLOZ-function.

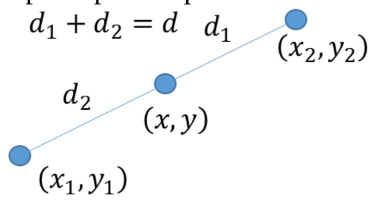

Figure 4: Two-dimensional FLOZ

Let us show that the α-system of V.L. Rvachev's function [2] allows two FLOZ-functions to be connected correctly.

The intersection function in complete form is written as

$$R_{\alpha}^{\cap} = \frac{1}{1-\alpha}\left(x + y - \sqrt{x^2 + y^2 - 2\alpha xy}\right). \qquad (11)$$

Since the coefficient α does not affect the zero boundary and the sign of the function value in any way, we will remove it from the reasoning by assigning a zero value to α. We will obtain

$$R_{0}^{\cap} = x + y - \sqrt{x^2 + y^2}. \qquad (12)$$

It is necessary to make sure that there is at least one zero value of the function-arguments $R_{0}^{\cap} = 0$. If we substitute $y = 0$, into the expression, then we get

$$R_{0}^{\cap} = x - \sqrt{x^2} = 0. \qquad (13)$$

A similar result is expected for $x = 0$. This means that when the function $R_{0}^{\cap}$ works with floz-functions, the zero values on both neighborhoods are preserved. For positive FLOZes $x + y > \sqrt{x^2 + y^2}$. This follows from the right-angled triangle rule. Hence, we can conclude that it is the R-intersection function $R_{\alpha}^{\cap}$ that can correctly connect two FLOZes on the positive range of values while preserving their basic property of zeroing a given local neighborhood.

Figure 5 shows the procedure for the intersection of two given floz-functions on the considered area.

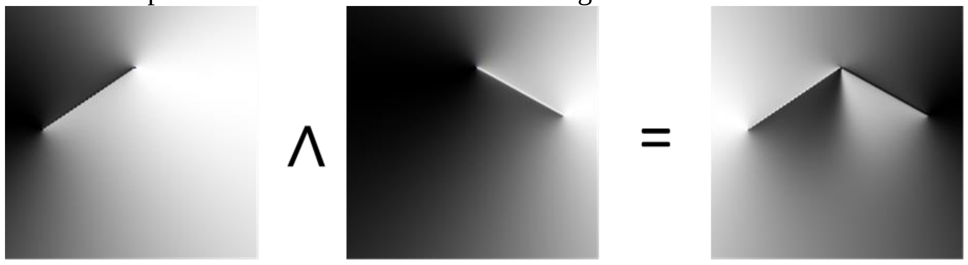

Figure 5: The procedure of the intersection of two FLOZ-functions.

2. The space E^3, i.e. three vertices define a triangle. The equation of the plane for the given vertices will be written

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = a_1^S x + a_2^S y + a_3^S z + a_4^S = 0, \qquad (14)$$

Hence the area of the triangular element from the given vertices

$$V_{fl}^3 = \frac{\sqrt{{a_1^S}^2 + {a_2^S}^2 + {a_3^S}^2}}{(3-1)!} = \frac{\sqrt{{a_1^S}^2 + {a_2^S}^2 + {a_3^S}^2}}{2} = S_\Delta. \tag{15}$$

Three equations of the plane, specified by brute-force search or exhaustive search through pairs of vertices with one current point $P_k(X_3)$ to determine the coefficients and areas, can be written as:

$$\begin{vmatrix} x & y & z & 1 \\ x & y & z & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = a_1^{S_1}x + a_2^{S_1}y + a_3^{S_1}z + a_4^{S_1} = 0, \tag{16}$$

$$V_{P_1}^3 = \frac{\sqrt{{a_1^{S_1}}^2 + {a_2^{S_1}}^2 + {a_3^{S_1}}^2}}{(3-1)!} = \frac{\sqrt{{a_1^{S_1}}^2 + {a_2^{S_1}}^2 + {a_3^{S_1}}^2}}{2} = S_1, \tag{17}$$

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x & y & z & 1 \\ x_3 & y_3 & z_3 & 1 \end{vmatrix} = a_1^{S_2}x + a_2^{S_2}y + a_3^{S_2}z + a_4^{S_2} = 0, \tag{18}$$

$$V_{P_1}^3 = \frac{\sqrt{{a_1^{S_2}}^2 + {a_2^{S_2}}^2 + {a_3^{S_2}}^2}}{(3-1)!} = \frac{\sqrt{{a_1^{S_2}}^2 + {a_2^{S_2}}^2 + {a_3^{S_2}}^2}}{2} = S_2, \tag{19}$$

$$\begin{vmatrix} x & y & z & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x & y & z & 1 \end{vmatrix} = a_1^{S_3}x + a_2^{S_3}y + a_3^{S_3}z + a_4^{S_3} = 0, \tag{20}$$

$$V_{P_1}^3 = \frac{\sqrt{{a_1^{S_3}}^2 + {a_2^{S_3}}^2 + {a_3^{S_3}}^2}}{(3-1)!} = \frac{\sqrt{{a_1^{S_3}}^2 + {a_2^{S_3}}^2 + {a_3^{S_3}}^2}}{2} = S_3, \tag{21}$$

$$\boldsymbol{Floz}(\boldsymbol{X_3}) = S_1 + S_2 + S_3 - S_\Delta = 0. \tag{22}$$

Figure 6 demonstrates the principle of FLOZ-function modeling by three vertices in the space E^3. By analogy with a two-dimensional space, a triangular FLOZ is modeled in the RANOK 3D system, described in a specialized FORTU language [3]:

```
ARGUMENT X,Y,Z;
CONSTANT X1=1., Y1=0., Z1=1.;
         X2=-1., Y2=1., Z2=-1.;
         X3=-1., Y3=-1., Z3=-1.;
          A=Y1*(Z2-Z3)-Y2*(Z1-Z3)+Y3*(Z1-Z2);
          B=-(X1*(Z2-Z3)-X2*(Z1-Z3)+X3*(Z1-Z2));
          C=X1*(Y2-Y3)-X2*(Y1-Y3)+X3*(Y1-Y2);
          S=1./2.*SQRT(A*A+B*B+C*C);
VARIABLE AA=Y*(Z2-Z3)-Y2*(Z-Z3)+Y3*(Z-Z2);
        BB=-(X*(Z2-Z3)-X2*(Z-Z3)+X3*(Z-Z2));
        CC=X*(Y2-Y3)-X2*(Y-Y3)+X3*(Y-Y2);
        S1=1./2.*SQRT(AA*AA+BB*BB+CC*CC);
        AA=Y1*(Z-Z3)-Y*(Z1-Z3)+Y3*(Z1-Z);
        BB=-(X1*(Z-Z3)-X*(Z1-Z3)+X3*(Z1-Z));
        CC=X1*(Y-Y3)-X*(Y1-Y3)+X3*(Y1-Y);
VARIABLE S2=1./2.*SQRT(AA*AA+BB*BB+CC*CC);
```

```
                AA=Y1*(Z2-Z)-Y2*(Z1-Z)+Y*(Z1-Z2);
                BB=-(X1*(Z2-Z)-X2*(Z1-Z)+X*(Z1-Z2));
                CC=X1*(Y2-Y)-X2*(Y1-Y)+X*(Y1-Y2);
    VARIABLE S3=1./2.*SQRT(AA*AA+BB*BB+CC*CC);
    VARIABLE SS=S1+S2+S3-S;
      RETURN SS;
```

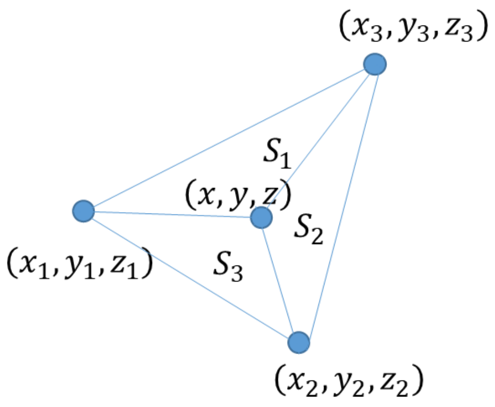

Figure 6: FLOZ in the space E^3.

Figure 7 displays the result of visualization of zero values of one FLOZ in a given 3D space (obtained in the RANOK 3D system).

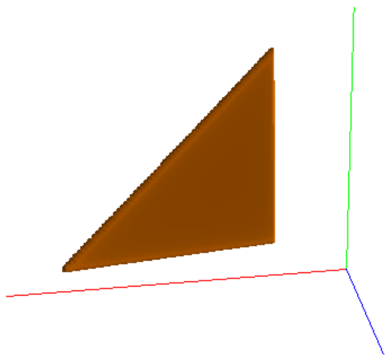

Figure 7: Three-dimensional FLOZ made in the RANOK 3D system

Figure 8 shows an example of a "*flozation*" of a triangulated mesh describing a three-dimensional object "sphere", consisting of 80 triangles.

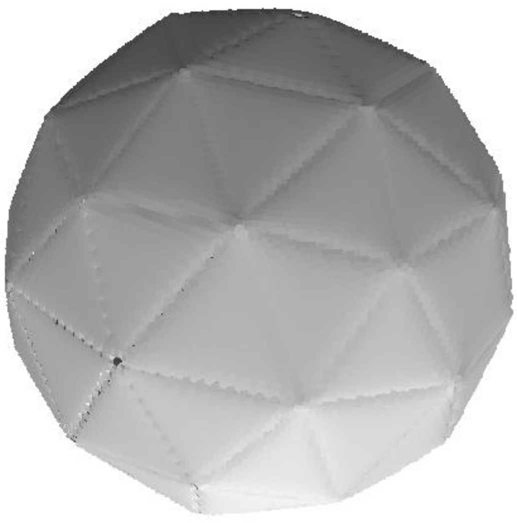

Figure 8: Flozation with 80 triangles on the surface of a sphere

3. Flozation of a complex non-convex contour

Let's return to the problem of a non-convex contour constructing. Taking into account the obtained properties of the local intersection of FLOZes, we can talk about the possibility of organizing a sequential algorithm for the "flozation" of the contour. As a result of the formulated R-functional construction, we obtain a surface of positive scalar values, where a non-convex contour defined by a set of two-dimensional FLOZes belongs to zero values. Figure 9 shows the order of forming the surface of the function in the process of step-by-step adding of the missing FLOZes. Below there is an example of the implementation of the R-functional structure in the RANOK 2D system:

```
RECTANGLE(0,0,25,25)
RECTBMP(200,200)
ARGUMENT x,y
CONSTANT x1=4, y1=7, x2=16, y2=1
         x3=14, y3=4, x4=15, y4=7
         x5=12, y5=6, x6=11, y6=10
         x7=13, y7=13, x8=14, y8=10
         x9=17, y9=9, x10=20, y10=13
         x11=14, y11=21, x12=10, y12=17
         x13=8, y13=20, x14=2, y14=22
FUNCTION s1=((x-x1)^2+(y-y1)^2)^0.5
         s2=((x2-x)^2+(y2-y)^2)^0.5
      w1=s1+s2-((x2-x1)^2+(y2-y1)^2)^0.5
         s1=((x-x2)^2+(y-y2)^2)^0.5
         s2=((x3-x)^2+(y3-y)^2)^0.5
      w2=s1+s2-((x3-x2)^2+(y3-y2)^2)^0.5
         s1=((x-x3)^2+(y-y3)^2)^0.5
         s2=((x4-x)^2+(y4-y)^2)^0.5
      w3=s1+s2-((x4-x3)^2+(y4-y3)^2)^0.5
         s1=((x-x4)^2+(y-y4)^2)^0.5
         s2=((x5-x)^2+(y5-y)^2)^0.5
      w4=s1+s2-((x5-x4)^2+(y5-y4)^2)^0.5
         s1=((x-x5)^2+(y-y5)^2)^0.5
         s2=((x6-x)^2+(y6-y)^2)^0.5
      w5=s1+s2-((x6-x5)^2+(y6-y5)^2)^0.5
         s1=((x-x6)^2+(y-y6)^2)^0.5
         s2=((x7-x)^2+(y7-y)^2)^0.5
      w6=s1+s2-((x7-x6)^2+(y7-y6)^2)^0.5
         s1=((x-x7)^2+(y-y7)^2)^0.5
         s2=((x8-x)^2+(y8-y)^2)^0.5
      w7=s1+s2-((x8-x7)^2+(y8-y7)^2)^0.5
         s1=((x-x8)^2+(y-y8)^2)^0.5
         s2=((x9-x)^2+(y9-y)^2)^0.5
      w8=s1+s2-((x9-x8)^2+(y9-y8)^2)^0.5
         s1=((x-x9)^2+(y-y9)^2)^0.5
         s2=((x10-x)^2+(y10-y)^2)^0.5
      w9=s1+s2-((x10-x9)^2+(y10-y9)^2)^0.5
         s1=((x-x10)^2+(y-y10)^2)^0.5
         s2=((x11-x)^2+(y11-y)^2)^0.5
      w10=s1+s2-((x11-x10)^2+(y11-y10)^2)^0.5
         s1=((x-x11)^2+(y-y11)^2)^0.5
         s2=((x12-x)^2+(y12-y)^2)^0.5
      w11=s1+s2-((x12-x11)^2+(y12-y11)^2)^0.5
         s1=((x-x12)^2+(y-y12)^2)^0.5
         s2=((x13-x)^2+(y13-y)^2)^0.5
```

```
                    w12=s1+s2-((x13-x12)^2+(y13-y12)^2)^0.5
                        s1=((x-x13)^2+(y-y13)^2)^0.5
                        s2=((x14-x)^2+(y14-y)^2)^0.5
                    w13=s1+s2-((x14-x13)^2+(y14-y13)^2)^0.5
                        s1=((x-x14)^2+(y-y14)^2)^0.5
                        s2=((x1-x)^2+(y1-y)^2)^0.5
                    w14=s1+s2-((x1-x14)^2+(y1-y14)^2)^0.5
        w=w1&w2&w3&w4&w5&w6&w7&w8&w9&w10&w11&w12&w13&w14
        RETURN w.
```

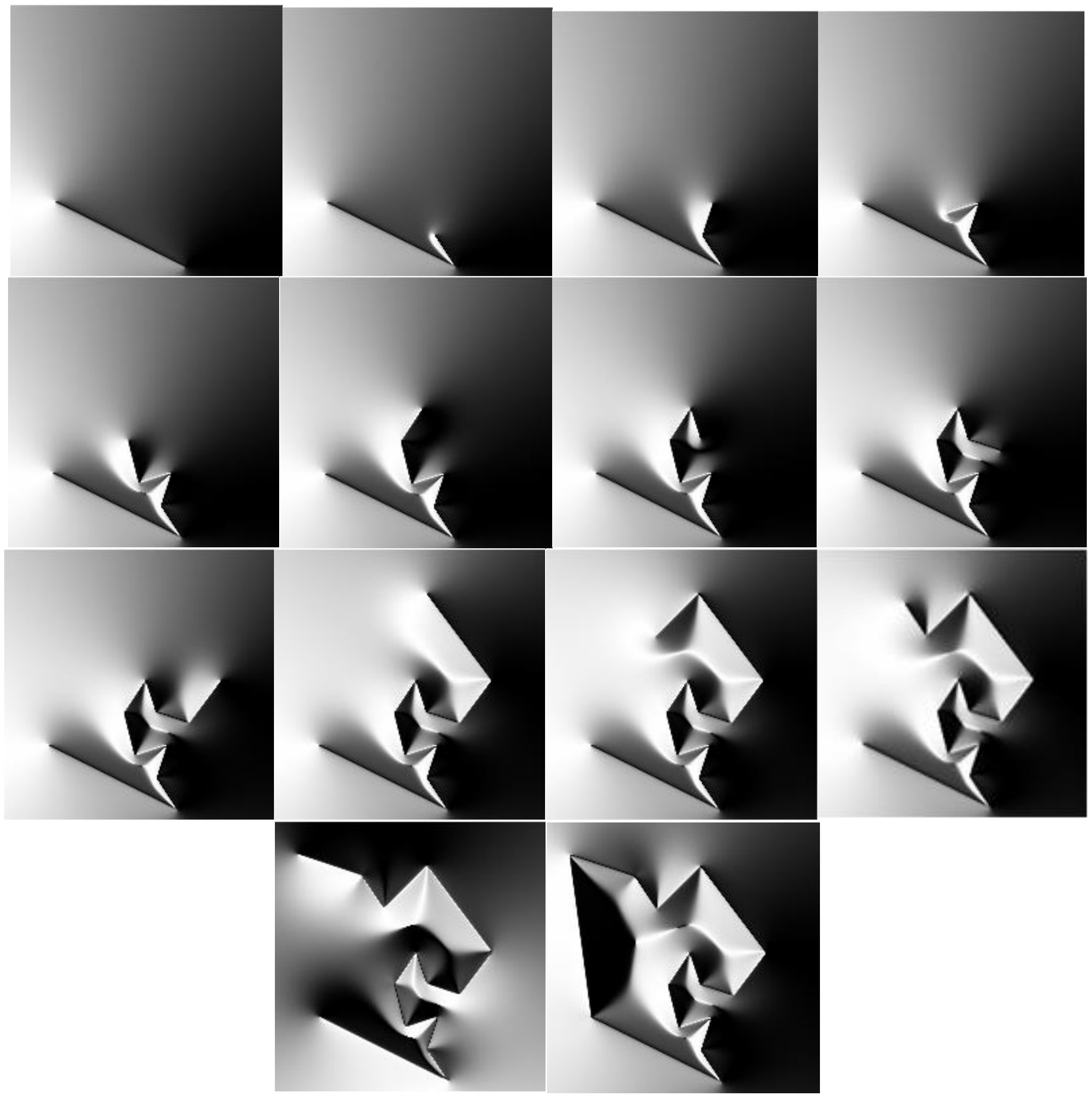

Figure 9: Implementation of the non-convex contour flozation sequence

There is a significant increase in computational operations, which negatively affects the running time of the algorithm. The time period in comparison with the recursive algorithm increases by two orders of magnitude. Thus, avoiding recursive complexity, we get exponentially increasing time for each added flozation step. This problem is typical for R-functional modeling in general, since the design of calculations is based on the structural attachment of functions. This problem can be avoided only by sequentially saving intermediate calculated values in a given function area at each step of the R-functional set-theoretic operation. In the process of flozation, this is expressed by the construction of the next FLOZ.

Let us consider one of the methods of computer modeling of the analytical function area, which allows us to fulfill the task.

4. The method of the functional voxel modeling

The essence of the method is described extensively in the works [3-7]. The idea is that the domain of the m-dimensional function $f(X_m)$ can be represented on a computer by the domain of local functions at each point of such a domain

$$n_1x_1 + \cdots n_{m+1}x_{m+1} + n_{m+2} = 0 \tag{23}$$

In this case, the local geometric characteristics n_i are represented by a set of $m+2$ voxel images of dimension m. This allows you to replace a complex mathematical expression with such a graphical representation. In this case, the local function at a single point is capable of combining the main part of the properties of the replaced complex mathematical expression, greatly simplifying the calculations. For example, we represent a function of the form

$$z = (x-1)e^{-[x^2+(y+1)^2]} + 10(0{,}2x - x^3 - y^5)e^{-(x^2+y^2)} + e^{-\frac{(x^2+y^2)}{3}}; \tag{24}$$

on a computer by images (Fig. 10).

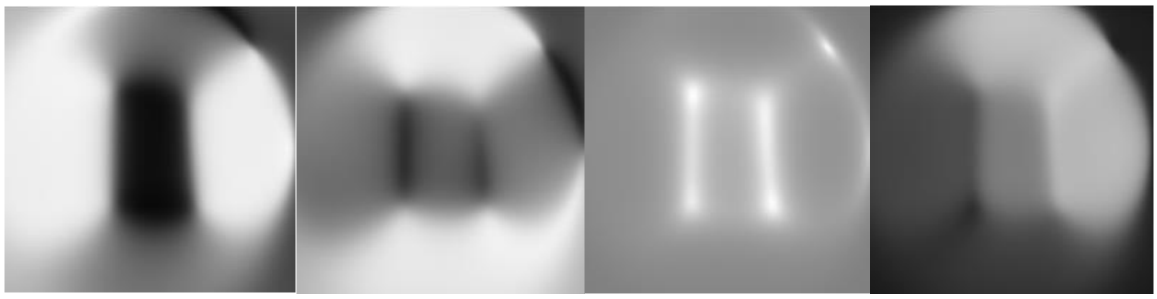

Figure 10: An example of a computer representation of a function z

The local function takes the form:

$$n_1x + n_2y + n_3z - n_4 = 0, \tag{25}$$

where

$$n_i = \frac{2C_i - P}{P}, \tag{26}$$

$P = 256$ – the number of the palette intensity gradation. This means that the function z takes the form

$$z = \frac{n_4}{n_3} - \frac{n_1}{n_3}x - \frac{n_2}{n_3}y. \tag{27}$$

To implement the phased flozation shown in Figure 9, it is sufficient to calculate by expression (27) the value of z for the intersected regions z^1 and z^2. Then, to carry out the R-intersection: $z = z^1 + z^2 - \sqrt{(z^1)^2 + (z^2)^2}$. Repeating this calculation for two more neighbour points of the M-image, we obtain the minimal spatial triangle, which plane can also be expressed by equation (25).

Expressing the numerical information through the intensity gradation of the monochrome palette

$$C_i = \frac{(n_i+1)P}{2}, \text{ где } P = 256 \tag{28}$$

we will obtain four M-images, fixing a new stage of flozation. Figure 11 shows an example of the proposed FV-approach implementation to stepwise flozation. Unlike FVR-modeling, where the calculation is performed for one point of the image, here the R-function is recalculated three times and an additional calculations generate new components n_i.

At the same time, I would like to note a significant reduction in the operating time of the algorithm with the same technical characteristics of the computer. The time spent on calculating the contour is now no more than 30 seconds, which is approximately 2 seconds for each performed flozation step for an image resolution of 400x400 pixel. In this case, the main advantage is achieved: the running time of the algorithm increases linearly and depends on the number of FLOZes, as well as the resolution of M-images used for the computer representation of the functional-voxel model.

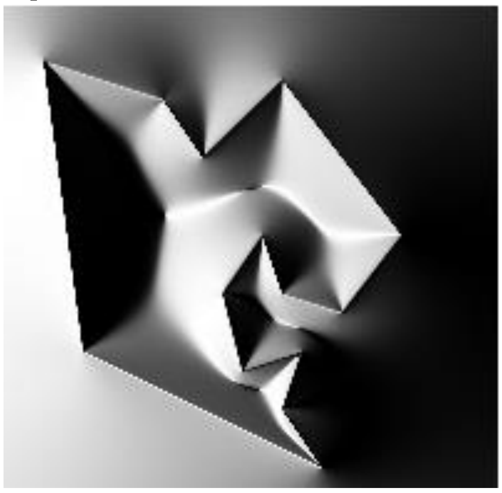

Figure 11: An example of nonconvex contour flozation using the FV-approach.

5. References

[1] V. L. Rvachev, Geometrical applications of Boolean algebra, Kiev, Technika, 1967. in Russian.

[2] V. L. Rvachev, Theory of R-functions and Some Applications, Kiev, Naukova Dumka, 1982. in Russian.

[3] A. V. Tolok, Functional voxel method in computer modeling, Moscow, Fizmatlit, 2016. in Russian.

[4] M. A. Loktev, Peculiarities of Functional-Voxel Modeling Application in Problems of Pathfinding with Obstacles, Information technologies in design and production 1 (2016) 45-49. in Russian.

[5] A. M. Plaksin, S. A. Pushkarev, Geometric modeling of objects thermal characteristics by the functional-voxel method, Geometry and Graphics 1 (2020) 25-32. In Russian.

[6] E. A. Lotorevich, Principles of spatial visual layout of analytical models displayed in voxel graphical space, Mechanical Engineering Technology 11 (2013) 59-63. In Russian.

[7] A. V. Tolok, N. B. Tolok, Mathematical Programming Problems Solving by Functional Voxel Method, Automation and Remote Control 9 (2018) 1703-1712.