Solid Modeling of Geometric Objects in Point Calculus

Evgeniy Konopatskiy [1] and Andrey Bezditnyi [2]

[1] *Donbas National Academy of Civil Engineering and Architecture, Derzhavina Street, 2, Makeevka, 286123, Ukraine*
[2] *Sevastopol branch of «Plekhanov Russian University of Economics», Vakulenchuk Street, 29, Sevastopol, 299053, Russian Federation*

Abstract

The paper describes an approach to solid modeling of geometric objects in the form of an organized three-parameter set of points in three-dimensional space. The relevance of the research topic is due to the widespread use of solid-state models in various branches of science and technology, mechanical engineering, construction and medicine. Solid-state computer models are currently one of the basic computer graphics tools and an integral part of computer-aided design and calculation systems. It is widely used as one of the control elements of CNC machines and 3D printing, the development of information systems in the design and construction of buildings and structures, finite element calculations of deformed states in aircraft and mechanical engineering, their manufacture in medicine, etc. The choice of point calculus as a mathematical apparatus for the analytical description of solid models of geometric objects is substantiated. Examples of modeling sets of elliptical bodies and toroidal bodies in a simplex of three-dimensional space are given.

Keywords

Solid modeling, point calculus, geometric modeling, geometric scheme, elliptical body, toroidal body.

1. Introduction

Solid-state computer models are currently an integral part of computer-aided design and calculation systems. In addition, solid models are one of the basic tools for computer graphics and virtual reality. They have found wide application as one of the control elements of clock-controlled machine tools and printing on 3D printers, the development of information systems in construction and the calculation of the stress-strain state of building structures, the calculation of parts and assemblies for strength and stability in mechanical engineering, etc. These software products include a number of foreign (AutoCAD, Mechanical Desktop, Inventor, SolidWorks, Solid Edge, 3DS Max и др.), and a number of domestic (Compass-3D, nanoCAD, etc.) developments. You can also distinguish a number of programs that have grown from computer graphics to information modeling, while retaining the functionality of geometric modeling and shaping of geometric bodies and surfaces (Revit, Archicad, etc.). And in mathematical programs and computer algebra systems (Maple, MathCAD, Mathlab, etc.), the possibility of building and graphical visualization of solid models is not provided at all. Even if the mathematical apparatus itself allows you to implement a solid model, the interpreter simply lacks a set of necessary commands for displaying it and then exporting it to computer-aided design systems. In addition, the overwhelming majority of these software products are paid, and therefore the computational algorithms and mathematical apparatus embedded in them are a trade secret, which the developers carefully guard from competitors. This significantly complicates the assessment of the effectiveness of the mathematical component embedded in them. There are a number of works on solid modeling and computer graphics containing algorithms and mathematical apparatus of computational

and computer geometry [1-4]. However, it is difficult to establish a specific relationship between these works and the mathematical apparatus embedded in specific commercial software products.

The disadvantages of existing solid-state modeling systems include the fact that at the stage of model development, all systems use only one of the cores of a multicore processor. For example, the official website Autodesk has the following information: "AutoCAD and AutoCAD for Mac products support multi-core technology only in certain applications (2D regeneration). To take full advantage of multi-core processors, you need to use multi-threaded software. AutoCAD is a single threaded application". At the stage of calculation, for example, using the finite element method, the multicore capabilities of modern processors are already being used. But it is at the stage of development of the model of using just one core may not be enough. And when the model turns out to be large enough, as a result of the implementation of a large project with a large number of elements, it is extremely difficult to work with it even with a powerful modern computer. Now this problem is being solved through the use of additional software, which assembles one large project from different models using a limited display of models. For example, in Autodesk Revit, several models with calculations are developed, which are then assembled together using the Navisworks software package. However, this disadvantage can be avoided by using the mathematical apparatus of parallel computations embedded in computational algorithms for solid modeling.

Another drawback of the existing solid modeling systems is the scarce tooling of shape-generating elements and geometric primitives. For example, even such simple shape-generating elements as curves of the 2nd order, which have already been studied quite deeply, both analytically and geometrically [5-10], are not fully implemented in any of the above systems of solid modeling. In the best case, it is possible to construct an ellipse in various parameterizations. The parabola is more complicated. And the construction of an arc of a hyperbola is possible exclusively using approximation (for example, by splines). Accordingly, more complex shaping elements are implemented in a similar way, and some of them, for example, a body with a generator in the form of a cycloid, does not have the ability to be realized at all. After all, this curve has unique properties, thanks to which it can be used both as a tautochrone and as a brachistochrone. And if we approximate the cycloid line with other lines, then, of course, it will look the same, but it will lose its unique properties. In [11] another example is given, which describes that in order to build a simple and familiar onion dome that adorns St. Basil's Cathedral with curves of the required curvature, it was necessary to develop a whole language of geometric constructions for the computer implementation of geometric models. But then another problem arises. Even if we use additional program code written, for example, in VBA or AutoLISP for AutoCAD, we will certainly face the problem of graphical output of the results, which will again be reduced to approximation, because visualization of other geometric primitives is simply not provided for the functionality of software products. If we consider the problem from the point of view of further use of the finite element method for calculating solid models, then this approach is quite justified and, in some cases, even more convenient. On the other hand, it is completely inapplicable if it is necessary to use the geometric properties of the model. And here it is appropriate to remember that the simplest of the geometric primitives is the point. The point itself does not even have a size. This is a geometric analogue of an infinitesimal value. However, using many points, you can get geometric objects of any complexity. As our organism consists of atoms, so any geometric object, any space, can be described by an organized set of points.

2. Justification of the choice of point calculus for the analytical description of solid geometric models

The mathematical apparatus "Point calculus" (other names are BN-calculus, Balyuba-Naydysh's point calculus) was developed by a team of scientists from the Melitopol School of Applied Geometry under the leadership of Academician V.M. Naidysh [12-13]. At its core, it can be attributed to computational geometry, since for any graphical operation of constructing a geometric object, the point calculus associates an analytical operation, which makes it possible to represent the entire cycle of geometric constructions in the form of point equations and computational algorithms based on them. In other words, point calculus allows you to translate geometric algorithms for modeling lines, surfaces and bodies into a digital language understandable for a computer, while maintaining the geometric

meaning of all graphic constructions through a number of parameters that make up point equations. It should also be taken into account that in many cases geometric algorithms can provide a simpler solution, and tools for geometric solution of various problems have accumulated and improved over many centuries.

In point calculus, any geometric object is represented by an organized set of points. A similar approach to the definition of geometric objects uses the Wurf calculus proposed by H. Staudt. In total, the point calculus in an affine space is a special case of the Wurf calculus in a projective space.

Traditionally, to determine the relative position of geometric bodies, it is necessary to coordinate each body separately, and then recalculate one coordinate system into another using the rotation and translation matrices. Point calculus uses a different approach based on the definition of geometric objects in a simplex, where a simplex is understood as an n-dimensional generalization of a tetrahedron. In a particular case, a simplex can be represented as a Cartesian coordinate system. The advantage of this approach is that all geometric objects are defined in a local simplex, and the result of their joint interaction in the form of a final solid geometric model is defined in a global coordinate system. Moreover, the transition from local simplices to the global one is carried out automatically, which is a consequence of the special invariant properties used in the point calculus. This feature of the point calculus makes it possible to develop computational algorithms for solid-state modeling using parallel computations on a computer with a modern multi-core processor.

Point calculus is based on invariants of affine geometry. On the one hand, this puts forward additional requirements for the choice of parameters of point equations, and on the other hand, it provides their coordinate-wise calculation. Based on this, the parameters of the point calculus can be invariants of parallel projection, presented in an explicit or implicit form. In addition, all geometric operations on geometric objects, taking into account their mutual position, also use the invariant properties of affine geometry. These include a simple ratio of three points of a straight line, parallelism of straight lines, intersection of straight lines, construction of a tangent, etc. All these geometric operations have their computational analogs in the point calculus and therefore can serve as an effective tool for computer modeling of complex geometric bodies consisting of several simpler bodies.

Parameters in point calculus are of two types: fixed and movable (current). A fixed parameter defines a point in space that has specific constant coordinates. The current point calculation parameter defines the moving points, which are called the current points. Such points move in space due to a change in the current parameter and fill the space with their movement. Thus, any continuous geometric objects are represented as a set of points. To organize this set-in point calculus, a special method of a moving simplex has been developed [14, 15], which is a generalization of the kinematic method for modeling geometric objects to a multidimensional space in point calculus. Then the problem of solid modeling is reduced to determining such a current point, which would completely fill the space bounded by some planes and surfaces in 3-dimensional space with its motion.

The work [16] outlined the principles of solid modeling in point calculus and the general concept, which consists in defining a geometric body as a geometric set of points, in which the number of current parameters defining it is equal to the dimension of space. This concept is a consequence of the further development of the geometric theory of multidimensional interpolation [17-20]. Based on the proposed concept in [16], examples were given of modeling simple geometric bodies in point calculus, which include a variety of pyramids, prisms, cones and cylinders. In continuation of this work, we will consider other examples of building solid models in point calculus.

3. Geometric modeling of elliptical bodies in point calculus

By an elliptical body we mean a geometric object filled with points with an elliptical generatrix. In the general case, this is the body of a triaxial ellipsoid, with the help of which, for example, a geoid is described [21]. One of the special cases of such a body will be a sphere. In accordance with the general approach to modeling geometric objects in point calculus, it is necessary to first develop a geometric scheme, which is a graphical algorithm for constructing a geometric object (in our case, an elliptical body).

In the simplex of 3-dimensional space $ABCD$ (fig. 1) we define 3 semiaxes: DA, DB and DC, specifying the coordinates of the points of the simplex. The axes can be positioned in 3-dimensional

space arbitrarily, in a particular case, you can use a rectangular Cartesian simplex with right angles at the vertex D. Note that when $|DA| = |DB| = |DC|$ we get the surface of a sphere centered at a point D, when $|DA| = |DB| \neq |DC|$ (or $|DA| \neq |DB| = |DC|$, or $|DC| = |DA| \neq |DB|$) we get an ellipsoid of revolution, and when $|DA| \neq |DB| \neq |DC|$ we get a triaxial ellipsoid.
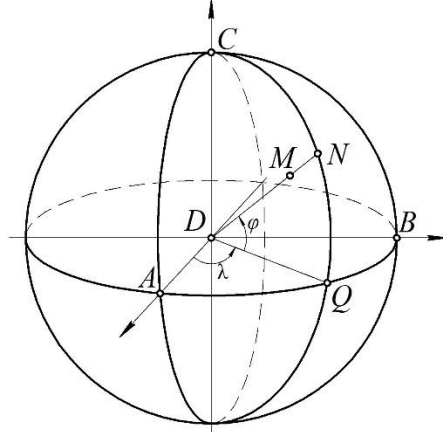

Figure 1: Geometric scheme for modeling elliptical bodies

Let us define a guiding elliptical line AQB by the following point equation [13]:

$$Q = (A - D)\cos\lambda + (B - D)\sin\lambda + D,$$

where λ is the current angular parameter of the point equation, which ranges from 0 to 2π.

The generating line of the ellipsoid surface QNC is defined by a similar point equation with the current parameter φ:

$$N = (Q - D)\cos\varphi + (C - D)\sin\varphi + D =$$
$$= (A - D)\cos\lambda\cos\varphi + (B - D)\sin\lambda\cos\varphi + (C - D)\sin\varphi + D,$$

where φ is the current angular parameter, which varies from $-\dfrac{\pi}{2}$ to $\dfrac{\pi}{2}$.

With the stated limits of change of the current parameters φ and λ the entire surface of the ellipsoid will be built. If necessary, changing the intervals of the current parameters, you can build the desired surface fragment.

Fill the internal space of the elliptical body with points by rotating the segment DN around the point D, where N is the current point of the ellipsoid surface:

$$M = D\bar{u} + Nu = (A - D)u\cos\lambda\cos\varphi + (B - D)u\sin\lambda\cos\varphi + (C - D)u\sin\varphi + D, \qquad (1)$$

where u is the current linear parameter, which ranges from 0 to 1;

$\bar{u} = 1 - u$ is the complement of the current parameter u to 1.

Note that the point equation (1) defines the entire variety of elliptic bodies, which are determined exclusively by the points of the simplex. In this case, the coordinates of the points of the simplex immediately determine the position of the elliptical body in space, its size and variety (ball, body of an ellipsoid of revolution, body of a triaxial ellipsoid).

After performing the coordinate-wise calculation, we obtain a system of parametric equations for the set of elliptic bodies in general form:

$$\begin{cases} x = (x_A - x_D)u\cos\lambda\cos\varphi + (x_B - x_D)u\sin\lambda\cos\varphi + (x_C - x_D)u\sin\varphi + x_D \\ y = (y_A - y_D)u\cos\lambda\cos\varphi + (y_B - y_D)u\sin\lambda\cos\varphi + (y_C - y_D)u\sin\varphi + y_D \\ z = (z_A - z_D)u\cos\lambda\cos\varphi + (z_B - z_D)u\sin\lambda\cos\varphi + (z_C - z_D)u\sin\varphi + z_D \end{cases}.$$

4. Geometric modeling of toroidal bodies in point calculus

Let's consider another example of modeling bodies of revolution, which also includes a toroidal body. By a toroidal body we mean a geometric object, by analogy with an elliptical body, filled with points and formed by the rotation of an elliptical line along an elliptical trajectory (one might say, an orbit). Then a special case of a toroidal body will be a toroid (a circle rotating around a circle).

Let us consider in more detail a graphical algorithm for constructing a toroidal body in the general case (Fig. 2). In a simplex $ABCD$ an elliptical guide line is given by the current point and is determined by two semiaxes DA and DB. The generating elliptical line is given by the current point N and is determined by two semiaxes PQ and PR, which form a movable simplex of the plane PQR. In this case, the length of one semiaxis of the generating line $|PR| = |CD|$ and is determined by the coordinates of the corresponding points of the simplex, and the length of the second semiaxis is given by a separate value $|PQ| = \Delta a$. Fill the inner part of the elliptical torus due to the movement of the current point M, which defines a segment NP that rotates around the point P due to the movement of the point N along an elliptical trajectory. Thus, for an unambiguous definition of an elliptical body in accordance with the geometric scheme of its graphic construction (Fig. 2), it is necessary to know the coordinates of the points of the simplex $ABCD$ and the length of the horizontal semiaxis of the generating elliptical line Δa.
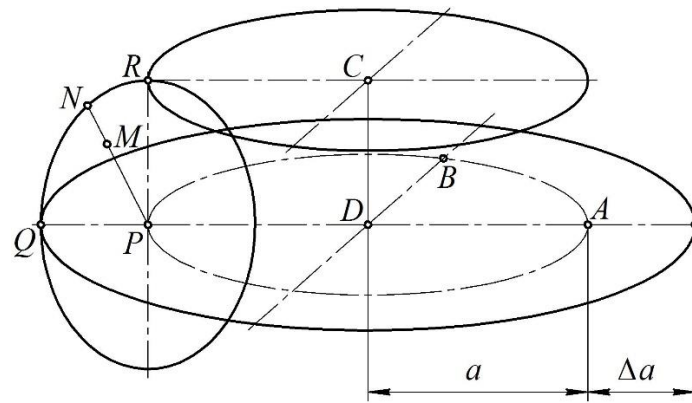

Figure 2: Geometric scheme for modeling toroidal bodies

Define a guide line using the point P equation of an ellipse with conjugate axes:
$$P = (A - D)\cos\varphi + (B - D)\sin\varphi + D,$$
where φ is the current angular parameter, which varies from 0 to 2π.

The current point R is determined using the length of the horizontal semiaxis of the generating elliptical line Δa:
$$\frac{QP}{PD} = \frac{\Delta a}{a} \Rightarrow Q = (P - D)\frac{\Delta a}{a} + P = (A - D)\left(1 + \frac{\Delta a}{a}\right)\cos\varphi + (B - D)\left(1 + \frac{\Delta a}{a}\right)\sin\varphi + D,$$
where is $a = |AD| = \sqrt{(x_A - x_D)^2 + (y_A - y_D)^2 + (z_A - z_D)^2}$.

The current point R of the moving simplex PQR is determined using the pointwise parallel transfer formula [7]:
$$R = P + C - D = (A - D)\cos\varphi + (B - D)\sin\varphi + C.$$

The forming line of the surface of the elliptic torus is determined using a parameter λ that varies from 0 to 2π:
$$N = (Q - P)\cos\lambda + (R - P)\sin\lambda + P =$$
$$= (A - D)\cos\varphi\left(1 + \frac{\Delta a}{a}\cos\lambda\right) + (B - D)\sin\varphi\left(1 + \frac{\Delta a}{a}\cos\lambda\right) + (C - D)\sin\lambda + D.$$

The following figure shows visualizations of geometric models of elliptical toroidal surfaces: Fig. 3a is the toroidal surface with elliptical guide and generatrix, Fig. 3b is the toroidal surface with a circular guide and generatrix (toroid).
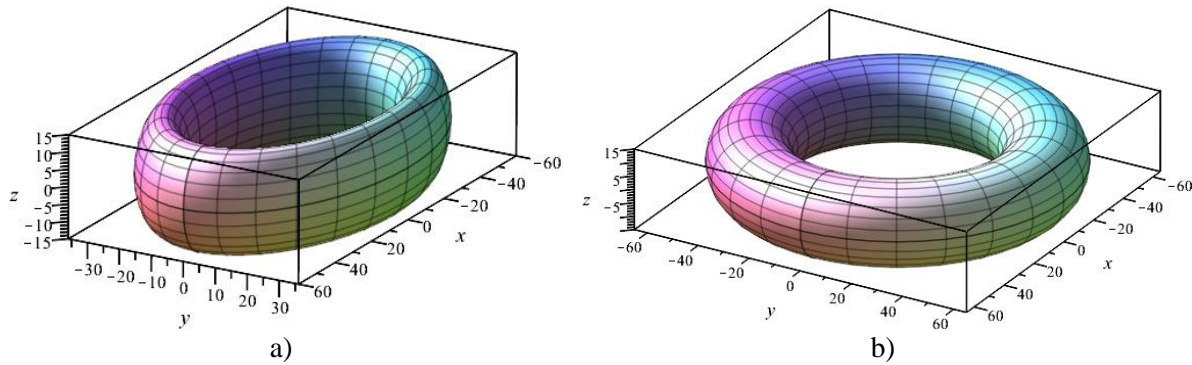

Figure 3: Visualization of elliptical toroidal surfaces

Fill the inner part of the toroidal body with the set of the current point M, which is determined by the parameter u:

$$M = P\bar{u} + Nu = (A - D)\left(1 + \frac{\Delta a}{a} u \cos \lambda\right)\cos \varphi + (B - D)\left(1 + \frac{\Delta a}{a} u \cos \lambda\right)\sin \varphi + (C - D)u \sin \lambda + D.$$

As a result, we get an organized three-parameter set of points, which in three-dimensional space is determined by the following system of the same type of parametric equations:

$$\begin{cases} x = (x_A - x_D)\left(1 + \dfrac{\Delta a}{a} u \cos \lambda\right)\cos \varphi + (x_B - x_D)\left(1 + \dfrac{\Delta a}{a} u \cos \lambda\right)\sin \varphi + (x_C - x_D)u \sin \lambda + x_D \\[2mm] y = (y_A - y_D)\left(1 + \dfrac{\Delta a}{a} u \cos \lambda\right)\cos \varphi + (y_B - y_D)\left(1 + \dfrac{\Delta a}{a} u \cos \lambda\right)\sin \varphi + (y_C - y_D)u \sin \lambda + y_D \\[2mm] z = (z_A - z_D)\left(1 + \dfrac{\Delta a}{a} u \cos \lambda\right)\cos \varphi + (z_B - z_D)\left(1 + \dfrac{\Delta a}{a} u \cos \lambda\right)\sin \varphi + (z_C - z_D)u \sin \lambda + z_D \end{cases}$$

5. Conclusion

The prospect for further research is the development of the mathematical apparatus of solid modeling of geometric objects in point calculus, the development of mathematical foundations of geometric operations of solid modeling in point calculus (such as: extrusion, rotation, kinematic operation, section operation, etc.) or their analogs, development computational algorithms for modeling simple and complex bodies in the form of a sequence of point equations, development of computational algorithms for Boolean operations on bodies in point calculus (such as: union, intersection and subtraction of bodies).

6. References

[1] M. De Berg, O. Cheong, M. Van Kreveld, M. Overmars, Computational Geometry: Algorithms and Applications, Springer-Verlag Berlin Heidelberg, 2008. doi: 10.1007/978-3-540-77974-2.
[2] S.L. Devadoss, J. O'Rourke, Discrete and Computational Geometry, Princeton University Press, 2011.
[3] D. Um, Solid Modeling and Application: Rapid Prototyping, CAD and CAE Theory, Springer Cham Heidelberg New York Dordrecht London, 2016. doi: 10.1007/978-3-319-21822-9.
[4] N.N. Golovanov, Geometric modeling, INFRA-M, Moscow, 2019.
[5] N.D. Vertinskaya, On some features of the behavior of second-order curves on the projective plane, Modern science-intensive technologies, No. 10, 2014, pp. 124-127.

[6] N.A. Sal´kov, Dupin cyclide and second-order curves. Part 1, Geometry and graphics, Vol. 4, No. 2, 2016, pp. 19-28.

[7] D.V. Voloshinov, A unified constructive algorithm for second-order curves' foci creation, Geometry and graphics, Vol. 6, No. 2, 2018, pp. 47-54.

[8] V.A. Korotkiy, Computer visualization of conic curve passing through the imaginary point and the imaginary concerning direct, Scientific visualization, Vol. 10, No. 1, 2018, pp. 56-68. doi: 10.26583/sv.10.1.04.

[9] V.A. Korotkiy, E.A. Usmanova, Quadratic birational transformations in computer simulation of building structures, IoP conference series: Materials Science and Engineering: Conf. Series 451 (2018), 012116. doi: 10.1088/1757-899X/451/1/012116.

[10] K.L. Panchuk, Second-order elliptic plane curves: monograph, Publishing house of OmSTU, Omsk, 2015.

[11] A.A. Boykov, Development and application of the geometry constructions language to building computer geometric models, IoP conference series: Journal of Physics: Conf. Series 1901 (2021), 012058. doi:10.1088/1742-6596/1901/1/012058.

[12] I.G. Balyuba, Constructive geometry of manifolds in point calculus, Kiev, 1995.

[13] I.G. Balyuba, E.V. Konopatskiy, A.I. Bumaga, Point calculus, Makeevka, 2020.

[14] I.G. Balyuba, The movable simplex method for constructing 2-surfaces of multidimensional space, Model and information technologies: Collection of Science Works, Volume 1, 2010, pp. 310–318.

[15] I.P. Davydenko, Designing of surfaces of spatial forms by the method of a movable simplex, Makeevka, 2012.

[16] E.V. Konopatskiy, A.A. Bezditnyi, M.V. Lagunova, A.V. Naidysh, Principles of solid modelling in point calculus, IoP conference series: Journal of Physics: Conf. Series 1901 (2021), 012063. doi: 10.1088/1742-6596/1901/1/012063.

[17] E.V. Konopatskiy, A.A. Bezditnyi, Geometric modeling of multifactor processes and phenomena by the multidimensional parabolic interpolation method, IoP conference series: Journal of Physics: Conf. Series 1441 (2020), 012063. doi: 10.1088/1742-6596/1441/1/012063.

[18] E.V. Konopatskiy, A.A. Bezditnyi, Application of mixed geometric interpolants for modeling the strength characteristics of steel fiber concrete, IoP conference series: Journal of Physics: Conf. Series 1546 (2020), 012037. doi: 10.1088/1742-6596/1546/1/012037.

[19] E.V. Konopatskiy, A.I. Bumaga, A.A. Bezditnyi, Geometric approach to finding the best possible solutions based on composition optimization of the mixed aggregate of fine-grained concrete, IoP conference series: Materials Science and Engineering: Conf. Series 962 (2020), 032031. doi: 10.1088/1757-899X/962/3/032031.

[20] E.V. Konopatskiy, O.S. Voronova, O.A. Shevchuk, A.A. Bezditnyi, About one method of numeral decision of differential equalizations in partials using geometric interpolants, CEUR Workshop Proceedings, Volume 2763, 2020, pp. 213–219. doi: 10.30987/conferencearticle_5fce27708eb353.92843700.

[21] O.A. Chernysheva, Krasovsky's ellipsoid in point calculus of Baluba-Naydysh, Bulletin of the Donbass National Academy of Civil Engineering and Architecture, Makeevka, Issue 2014-4(108), 2014, pp. 6–9.