

Article

Orebody Modeling Method Based on the Coons Surface Interpolation

Zhaohao Wu ^{1,2}, Lin Bi ^{1,2,3}, Deyun Zhong ^{1,2,3,*}, Ju Zhang ^{1,2} , Qiwang Tang ³ and Mingtao Jia ^{1,2,3}¹ School of Resources and Safety Engineering, Central South University, Changsha 410083, China² Research Center of Digital Mine, Central South University, Changsha 410083, China³ Changsha Digital Mine Co., Ltd., Changsha 410221, China

* Correspondence: deyun_zhong@csu.edu.cn

Abstract: This paper presents a surface modeling method for interpolating orebody models based on a set of cross-contour polylines (geological polylines interpreted from the raw geological sampling data) using the bi-Coons surface interpolation method. The method is particularly applicable to geological data with cross-contour polylines acquired during the geological and exploration processes. The innovation of this paper is that the proposed method can automatically divide the closed loops and automatically combine the sub-meshes. The method solves the problem that it is difficult to divide closed loops from the cross-contour polylines with complex shapes, and it greatly improves the efficiency of modeling based on complex cross-contour polylines. It consists of three stages: (1) Divide closed loops using approximate planes of contour polylines; each loop is viewed as a polygon combined with several polylines, that is the n-sided region. (2) After processing the formed n-sided regions, Coons surface interpolation is improved to complete the modeling of every single loop (3) Combine all sub-meshes to form a complete orebody model. The corresponding algorithm was implemented using the C++ programming language on 3D modeling software. Experimental results show that the proposed orebody modeling method is useful for efficiently recovering complex orebody models from a set of cross-contour polylines.



Citation: Wu, Z.; Bi, L.; Zhong, D.; Zhang, J.; Tang, Q.; Jia, M. Orebody Modeling Method Based on the Coons Surface Interpolation. *Minerals* **2022**, *12*, 997. <https://doi.org/10.3390/min12080997>

Academic Editor: Behnam Sadeghi

Received: 1 July 2022

Accepted: 3 August 2022

Published: 6 August 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: geological modeling; orebody modeling; coons surface interpolation; contour interpolation; polygon triangulation

1. Introduction

Three-dimensional geological modeling and visualization is one of the hot topics in the field of geosciences [1]. As a geological body with high economic value, orebody modeling is also one of the current research hot topics and is of great significance for promoting mine digitization and improving mining efficiency [2]. In 2002, Mallet [3] proposed a new method called discrete smooth interpolation (DSI), based on which a number of orebody modeling works have been published [4–7]. Additionally, some commercial software [8] has been developed using different technologies including triangulate surfaces, radial functions, and parametric surfaces [9]. However, due to the complex shape of the orebody, much research work still needs to be done in orebody modeling. Our research is dedicated to developing the software for modeling orebody using small triangles based on the Coons surface [10].

We can often obtain the cross-contour polylines of planes and sections of orebodies through geological logging in the geological exploration and production exploration of mines, especially of some precious metal mines with flat orebody shapes. These contour polylines are interlaced with complex shapes. Therefore, if the explicit modeling method based on the contour stitching method is used, the model will be constructed with poor quality and low efficiency [11]. Modeling orebodies with high speed and great quality is of great significance to improving the production efficiency of mines [12]. The orebody

modeling method we propose based on Coons surface interpolation can automatically divide the contour polylines into closed loops and interpolate and model the formed n-sided regions; then, the constructed sub-meshes will be combined without manual intervention. Based on the above steps, the complete orebody model can be quickly constructed.

1.1. Related Research Work

Based on the above analysis, we transform the 3D modeling of the orebody into the Coons surface interpolation of the interpreted contour polylines. Therefore, we will briefly summarize the research work on free-form surface reconstruction, contour interpolation modeling, and orebody modeling.

1.1.1. Free-Form Surface Reconstruction

Free-form surface reconstruction is a key technology of orebody 3D modeling [13–15]. In practical application, the commonly used surface reconstruction methods include implicit function surface, triangular Bezier surface, B-spline surface, NURBS surface, Coons surface, polygon model, and so on.

The implicit function surface [16–18] refers to the surface represented by equation $F(x, y, z) = 0$, which can represent quadric surfaces such as spherical surfaces, cylindrical surfaces, and more complex surfaces. When reconstructing the surface, the implicit function equations are constructed using the existing data, and then the surface model can be generated by obtaining isosurfaces through the Marching Cube method or other methods. Though using the implicit function to reconstruct a surface has the advantages of small calculation and easy solution, there are also some problems such as unclear geometric meaning. In the future, a great amount of research work needs to be carried out to make extensive use of implicit modeling.

In 1982, Farin [19] proposed the method of constructing the triangular Bezier surface based on the triangulation of scattered data. This surface is explicit and C^1 continuous with flexible construction. Using the triangular Bezier surface for surface reconstruction requires less manual intervention and is easy to automate. However, it also has some disadvantages, such as the insufficient ability for surface modification and poor controllability.

The B-spline curve and surface [20–22] have many excellent properties, such as geometric invariance, convex hull, convexity preservation, and local support, which lead to the powerful function of representing and designing free-form curves and surfaces. It is a commonly used free-form surface suitable for engineering shape design, but it cannot be well applied to represent and design elementary curves and surfaces. Therefore, researchers proposed the NURBS surface method [23–25] for surface reconstruction, which both has the powerful function of representing and designing free-form curves and surfaces like B-spline and can accurately represent quadric arcs and quadric surfaces. The unification of analytical geometry and free-form curves and surfaces is well realized by this method. Due to the introduction of a weight factor and a manipulation control vertex, it provides sufficient flexibility for various shape designs. However, the parameterization effect will become very poor with an inappropriate weight factor. In addition, there are still many problems to be further solved in the calculation of NURBS surface intersection.

In 1964, Professor S. A. Coons of the Massachusetts Institute of Technology (MIT) proposed a general method to describe surfaces, the Coons surfaces method [10], which has unique advantages in engineering applications. Given the boundary conditions, Coons patches of the corresponding degree can be constructed. Therefore, in theory, the Coons method can be used to construct arbitrary order patches through boundary constraints [26,27], which makes Coons surfaces show strong advantages in representing and designing surfaces. However, in the Coons surface method, it is difficult to control the surface shape. When the boundary curve is determined, the shape can only be modified by changing the torsion vector [28].

The most commonly used polygon model method is the triangle model method based on triangulation. The triangulation technology in the plane has matured [29], and great research progress has been made in the triangulation of surface data and tetrahedrons of spatial scattered data [30,31]. The characteristics of the polygon model are: (1) adjacent surface patches can only be C^0 continuous and (2) data amounts are large. When there needs high accuracy of a model, a large number of polygons need to be used to represent the surface approximately, which has a huge amount of data. As a result, this method is not suitable for constructing complex models with high accuracy. Therefore, in this paper, the polygon model method is used only in the modeling of simple single-sided regions.

1.1.2. Contour Interpolation Modeling

Visualization based on contour interpolation refers to the modeling of three-dimensional entities using a sequence of two-dimensional contours, which is an important research direction in scientific computing visualization. According to the number of contour polylines of two adjacent layers, it can be divided into single contour modeling and multi-contour modeling.

Single contour modeling refers to the three-dimensional model reconstruction of the contour polylines when there is only one contour polyline in each layer on two adjacent planes, which is relatively simple. Researchers have proposed many optimization methods for single contour modeling, such as the minimum surface product method [32], the maximum volume method [33] based on the global search strategy, the shortest diagonal method [34], and the synchronous advance method of adjacent contours [35] based on local calculation and decision.

Multi-contour modeling refers to 3D model reconstruction when there are multiple contour polylines on two adjacent planes, which difficulty is the correspondence and branch processing between contour polylines. In terms of the correspondence of contour polylines, Meyers et al. [36] proposed the minimum spanning tree method, but this method still has some limitations. In some special cases, the contour polylines cannot correctly correspond, and subsequent manual processing is required. For the branching problem between contour lines, Ekoule et al. [37] proposed a method for constructing the middle contour polyline in the middle of the two adjacent planes of contour polylines. This method transforms the branching problem into the connection problem between a series of single contour polylines. However, in some cases, this method will leave holes on the constructed model surface.

Jones et al. [38] proposed an isosurface construction method using volume data, which requires that contour polylines have simple shapes and be completely closed. There is no need to judge the corresponding relationship and branch relationship when using this method for surface reconstruction, and it is suitable for both convex and non-convex contour polylines. However, it also has the disadvantage of a large amount of calculation. Additionally, Zhong et al. [39] proposed a method of reconstructing the 3D orebody by cross-section contour polylines, which allows for adding geometric constraints and can easily control the shape of the model, but many discontinuous parts may be generated after reconstruction if the data are relatively sparse. Moreover, Wu et al. [40] proposed a 3D orebody modeling method based on the normal estimation of cross-contour polylines. First, the normals of the cross-contour polylines will be estimated, based on which the radial basis function will be used for interpolating and modeling. This method improves the modeling automation of contour polylines, but it requires strict intersection between contour polylines, and they can be completely broken at intersections.

1.1.3. Orebody Modeling

The 3D modeling of the orebody is to interpret the topological information and determine the geometric shape of the orebody using mine geological exploration data and production exploration data [41]. According to the different principles of model construction, it can be divided into explicit modeling and implicit modeling [42]. In explicit modeling, first, the contour polylines of the orebody should be delineated manually, and then the 3D reconstruction of two-dimensional contour polylines can be realized by the contour stitching method [37]. Because explicit modeling requires a great deal of manual interaction, the limitation of low efficiency of this method is gradually highlighted for orebodies with a large number of sections and complex shapes. In addition, when the modeling data change locally, the modelers need to reinterpret the data, delineate the orebody, and model through contour stitching. This process is complex, which is not conducive to the dynamic updating of the model [43]. In contrast with explicit modeling, implicit modeling is based on the implicit function. The relationships of the spatial sampling data are used to construct an implicit function, and the 3D model can be constructed by the spatial interpolation algorithm [18]. This method not only greatly reduces the manual interaction but also realizes the dynamic editing of the model, which also improves the automation and efficiency of modeling [44].

1.2. Solution Strategy

In this paper, we consider transforming the problem of orebody modeling into a modeling problem based on the Coons surface interpolation of n -sided regions. The proposed method requires the contour polylines to be in an approximate plane within the tolerance range. Firstly, we creatively propose a closed-loop division method for contour polylines interpreted from data obtained in mine production and exploration. In this method, the approximate plane of each contour polyline is used to cut all polylines, and the polylines in the final subspace are connected and grouped within the tolerance range to form closed loops. Secondly, we preprocess the formed n -sided region. For the simple single-sided region, the method based on constrained Delaunay triangulation is used for modeling. In terms of complex single-sided regions and non-quadrilateral regions, they are transformed into four-sided regions by different methods. Finally, the Coons surface is used for interpolation and modeling based on the processed four-sided regions to realize the 3D visualization of the orebody. In summary, we creatively propose a closed-loop division and modeling method of orebody contour polylines.

2. Overview of the Method

The basic idea of this modeling method using Coons surface interpolation is as follows. Through the steps of n -sided region processing and function solving, the Coons surface interpolation is used to construct sub-meshes through the formed closed loops, which are obtained by interpreted cross-contour polylines. Finally, the sub-meshes will be combined to realize the modeling of the complete orebody.

The proposed method mainly consists of the following three steps. Figure 1 shows the overall process of the method.

1. Use approximate planes of each contour polyline to cut all polylines for the closed-loop division.
2. Preprocess the formed loops and separately model the processed four-sided regions through Coons surface interpolation.
3. Combine all the sub-meshes to construct a complete orebody model.

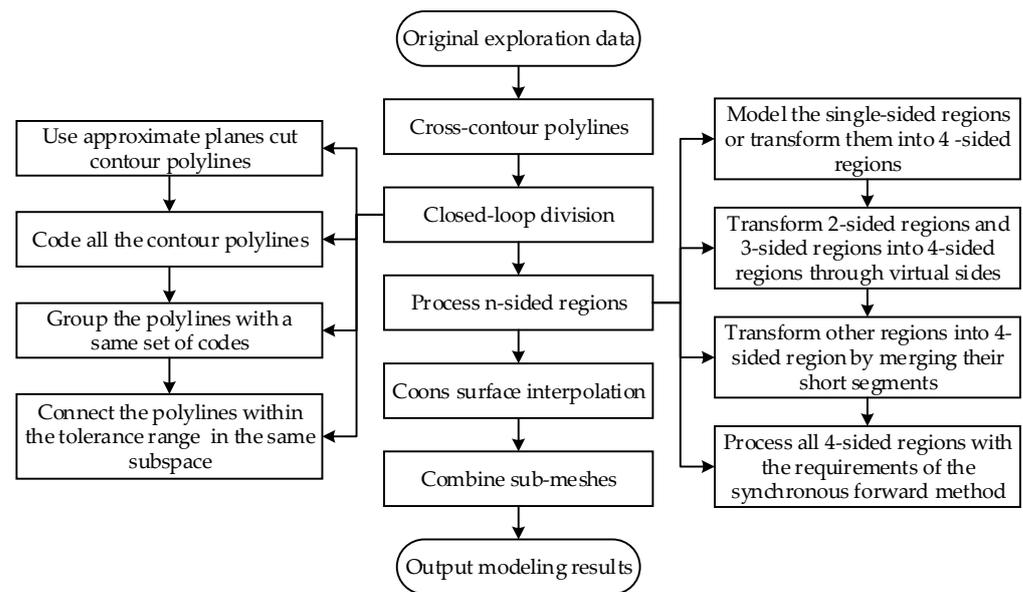


Figure 1. Overall flow chart.

3. Method

3.1. Coons Surface

With the continuous development of computer technology and modern industry, the application scope of Coons surfaces has gradually expanded from the shape design of cars, ships, and aircraft to various fields such as architectural design, medical research, and geological modeling. Coons surfaces can be divided into three categories according to boundary control conditions. The boundary control conditions of the first kind of Coons surface only contain four boundary curves or part of them. The boundary control conditions of the second kind of Coons surface include both boundary curves and boundary tangent vectors. For the third kind of Coons surface, the boundary control conditions include boundary curves and boundary tangent vectors, as well as boundary second-order derivative vectors. The bicubic Coons surface used in interpolation in this paper belongs to the second kind of Coons surface, which can ensure the interpolated patches are continuous both in position and slope.

To satisfy the slope continuity and the position continuity at the four boundaries, the four boundaries $P(u, j)$ and $P(i, v)$ and their cross-border tangent vectors $P_v(u, j)$ and $P_u(i, v)$ must be given at the same time, as shown in Figure 2.

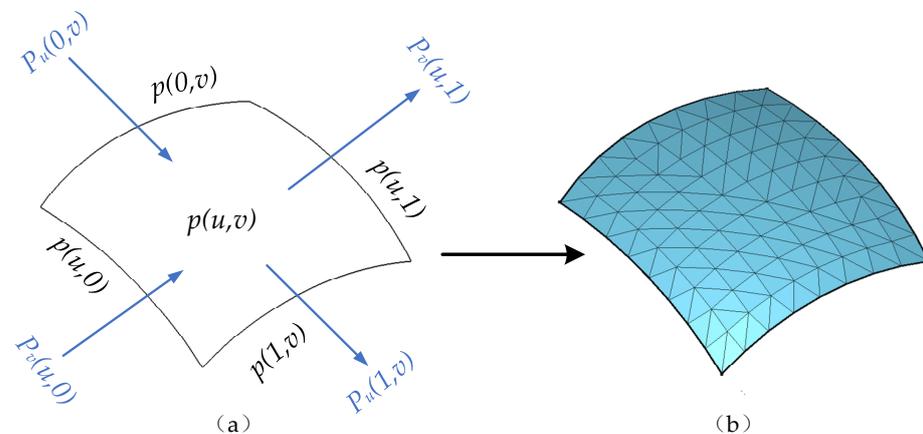


Figure 2. Schematic diagram of the interpolation of the second kind Coons surface: (a) boundaries and their cross-border tangent vectors of the surface patch; (b) the interpolated surface.

Under these constraints, the second kind of Coons surface equation [10] can be derived:

$$Q(u, v) = - \begin{bmatrix} -1 & F_0(u)F_1(u)G_0(u)G_1(u) \end{bmatrix} \times \begin{bmatrix} 0 & P(u,0) & P(u,1) & P_v(u,0) & P_v(u,1) \\ P(0,v) & P(0,0) & P(0,1) & P_v(0,0) & P_v(0,1) \\ P(1,v) & P(1,0) & P(1,1) & P_v(1,0) & P_v(1,1) \\ P_u(0,v) & P_u(0,0) & P_u(0,1) & P_{uv}(0,0) & P_{uv}(0,1) \\ P_u(1,v) & P_u(1,0) & P_u(1,1) & P_{uv}(1,0) & P_{uv}(1,1) \end{bmatrix} \begin{bmatrix} -1 \\ F_0(v) \\ F_1(v) \\ G_0(v) \\ G_1(v) \end{bmatrix}, u, v \in [0, 1] \quad (1)$$

This fifth-order matrix is the boundary information matrix of the second kind of Coons patches (Coons patches with given boundaries and cross-border tangent vectors). F_0, F_1, G_0 and G_1 are mixed functions, which meet the following conditions [10]:

$$\begin{aligned} F_i(j) &= G'_i(j) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \\ F'_i(j) &= G_i(j) = 0, i, j = 0, 1 \end{aligned} \quad (2)$$

Based on Equation (2), we can solve that [10]:

$$\begin{cases} F_0(u) = 2u^3 - 3u^2 + 1 \\ F_1(u) = -2u^3 + 3u^2 \\ G_0(u) = u^3 - 2u^2 + u \\ G_1(u) = u^3 - u^2 \end{cases} \quad (3)$$

The above methods and formulas are extremely rigorous in theory, but too much boundary information is needed, which is not convenient for calculation and application. To meet its application requirements and simplify the calculation process as much as possible, Professor Coons proposed using corner information and mixed functions to define the boundary curves and their cross-border tangent vectors.

If the point vectors, tangent vectors of u direction, tangent vectors of v direction, and mixed partial derivative vectors of the four corners are known, as shown in Figure 3, by applying the mixed function F_0, F_1, G_0 and G_1 , we can express the four boundaries of the surface patch and their cross-border tangent vectors as follows [10].

$$\begin{cases} P(0, v) = F_0(v)P(0, 0) + F_1(v)P(0, 1) + G_0(v)P_v(0, 0) + G_1(v)P_v(0, 1) \\ P(1, v) = F_0(v)P(1, 0) + F_1(v)P(1, 1) + G_0(v)P_v(1, 0) + G_1(v)P_v(1, 1) \\ P(u, 0) = F_0(u)P(0, 0) + F_1(u)P(0, 1) + G_0(u)P_u(0, 0) + G_1(u)P_u(0, 1) \\ P(u, 1) = F_0(u)P(1, 0) + F_1(u)P(1, 1) + G_0(u)P_u(1, 0) + G_1(u)P_u(1, 1) \\ P_u(u, 0) = F_0(u)P_u(0, 0) + F_1(u)P_u(0, 1) + G_0(u)P_{uv}(0, 0) + G_1(u)P_{uv}(0, 1) \\ P_u(u, 1) = F_0(u)P_u(1, 0) + F_1(u)P_u(1, 1) + G_0(u)P_{uv}(1, 0) + G_1(u)P_{uv}(1, 1) \\ P_v(u, 0) = F_0(u)P_v(0, 0) + F_1(u)P_v(0, 1) + G_0(u)P_{uv}(0, 0) + G_1(u)P_{uv}(0, 1) \\ P_v(u, 1) = F_0(u)P_v(1, 0) + F_1(u)P_v(1, 1) + G_0(u)P_{uv}(1, 0) + G_1(u)P_{uv}(1, 1) \end{cases} \quad (4)$$

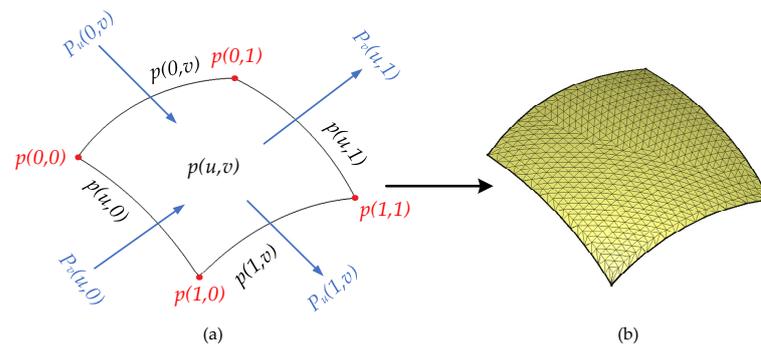


Figure 3. Schematic diagram of the interpolation of the bicubic Coons surface: (a) corner information, boundaries, and their cross-border tangent vectors of the surface patch; (b) the interpolated surface.

After bringing Equation (2) into Equation (1) and simplifying it, we can obtain the simplified surface equation [10]:

$$\left\{ \begin{aligned} Q(u,v) = [F_0(u)F_1(u)G_0(u)G_1(u)] \times \begin{bmatrix} P(0,0) & P(0,1) & P_v(0,0) & P_v(0,1) \\ P(1,0) & P(1,1) & P_v(1,0) & P_v(1,1) \\ P_u(0,0) & P_u(0,1) & P_{uv}(0,0) & P_{uv}(0,1) \\ P_u(1,0) & P_u(1,1) & P_{uv}(1,0) & P_{uv}(1,1) \end{bmatrix} \begin{bmatrix} F_0(v) \\ F_1(v) \\ G_0(v) \\ G_1(v) \end{bmatrix}, u,v \in [0,1] \end{aligned} \right. \quad (5)$$

where F_0, F_1, G_0 and G_1 are the same as Equation (3).

The cubic mixed function and Equation (3) can be written as [10]:

$$Q(u,v) = UMBM^T V^T \quad (6)$$

where:

$$U = [u^3 \quad u^2 \quad u \quad 1], \quad V = [v^3 \quad v^2 \quad v \quad 1], \quad u,v \in [0,1] \quad (7)$$

$$M = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (8)$$

$$B = \begin{bmatrix} P(0,0) & P(0,1) & P_v(0,0) & P_v(0,1) \\ P(1,0) & P(1,1) & P_v(1,0) & P_v(1,1) \\ P_u(0,0) & P_u(0,1) & P_{uv}(0,0) & P_{uv}(0,1) \\ P_u(1,0) & P_u(1,1) & P_{uv}(1,0) & P_{uv}(1,1) \end{bmatrix} \quad (9)$$

The parametric equation of the Coons surface is [10]:

$$\begin{cases} x(u,v) = UMB_x M^T V^T \\ y(u,v) = UMB_y M^T V^T \\ z(u,v) = UMB_z M^T V^T \end{cases} \quad (10)$$

where $B_x, B_y,$ and B_z are the coordinate components of boundary information matrix B .

3.2. Closed-Loop Division

Cut all contour polylines using the approximate planes where each contour polyline is located. The contour polyline on the positive side (the positive and negative sides can be specified arbitrarily without affecting the final dividing result) is coded as 1 in this plane bit. The contour polyline on the negative side is coded as 2 in this plane bit. The contour polyline on the approximate plane is coded as 0 because it belongs to both positive and negative sides, and it will be expanded in the subsequent steps. Each time an approximate plane is added for cutting, the length of each polyline mark string is increased by 1, whose value is 0, 1, or 2.

As shown in Figure 4, there are six contour polylines in the figure, of which the bottom two are coplanar. Cut the contour polylines with the five approximate planes $\alpha, \beta, \gamma, \delta, \epsilon$ where the contour polylines are located and divide them into four subspaces ①, ②, ③, ④. The direction indicated by the arrow is the positive direction of the plane, and the tag array length of each contour polyline is 5.

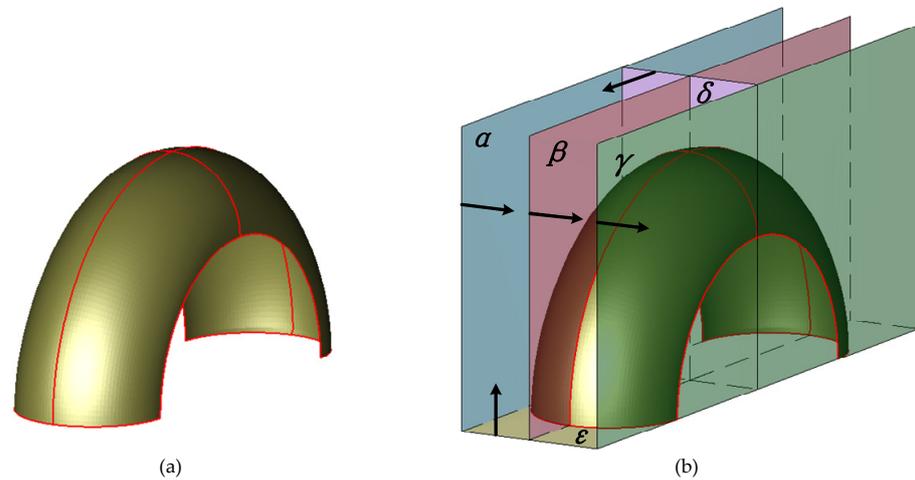


Figure 4. Schematic diagram of the closed-loop division: (a) the example model and the cross-contour polylines of it; (b) the formed subspace after using approximate planes to cut polylines.

As shown in Figures 4 and 5, the contour polylines a, b, c, d are cut by the approximate planes and coded according to the above rules. The initial coding result is shown in Table 1.

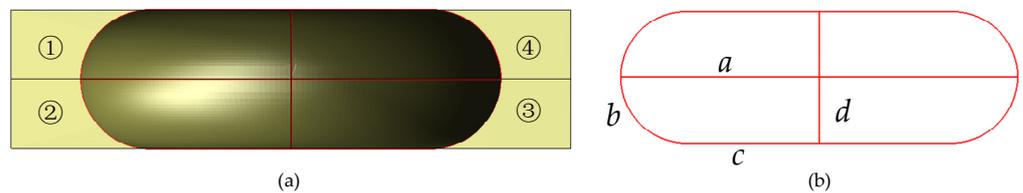


Figure 5. The top view of formed subspaces: (a) the top view of the complete subspaces; (b) the top view of the divided contour polylines.

Table 1. Initial coding table after cutting polylines with the approximate planes.

Plane Bit	α	β	γ	δ	ϵ	
Segments	a	1	0	2	1	1
b	1	1	2	1	0	
c	1	1	0	1	1	
d	1	1	2	0	1	

Copy the contour polyline containing 0 in the code array. On the plane bit where 0 is located, two contour polylines are coded as 1 and 2, and the other codes remain unchanged. For example, expand $a: 10,211$ to $a_1: 11,211$ and $a_2: 12,211$. After processing all the contour polylines, the final coding result is obtained as shown in Table 2.

The contour polylines with the same code array are the contour polylines in the same subspace. As shown in Table 2, a_1, b_1, c_2, d_1 marked by red have the same code arrays, which means that they are in the same subspace. Then, in the same subspace, the contour polylines within the specified tolerance are connected to form the closed loops.

Table 2. Final coding table after extending the side whose code array contains 0.

Plane Bit	α	β	γ	δ	ϵ
Segments					
a_1	1	1	2	1	1
a_2	1	2	2	1	1
b_1	1	1	2	1	1
b_2	1	1	2	1	2
c_1	1	1	1	1	1
c_2	1	1	2	1	1
d_1	1	1	2	1	1
d_2	1	1	2	2	1

It is worth noting that if the topological adjacency relationship between some contour polylines is too complex, the above automatic closed-loop division algorithm may not be able to obtain the required results. At this time, it is necessary to manually process and group the contour polylines to complete the closed-loop division.

3.3. Process *n*-Sided Regions

Using Coons surface for interpolation calculation needs the boundary information of four sides. However, many of the closed loops divided before are non-quadrilateral. Therefore, we need to preprocess the *n*-sided regions before interpolation calculation.

3.3.1. Process the Single-Sided Region

For the simple single-sided region, the polygon triangulation method based on constrained Delaunay triangulation is used to directly model it, as shown in Figure 6. For the complex single-sided region, we consider transforming it into a four-sided region by analyzing its shape characteristics and then performing the Coons surface interpolation.

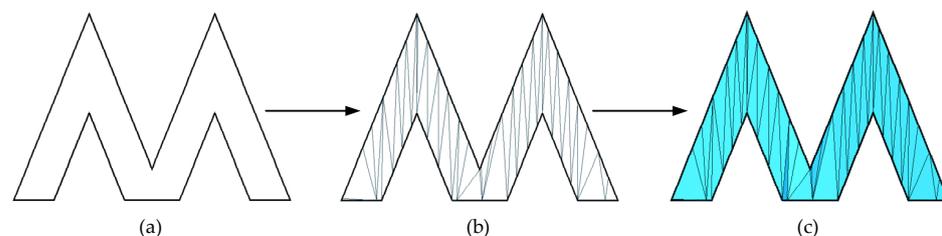


Figure 6. Schematic diagram of polygon triangulation: (a) the original polygon; (b,c) the process and the result of polygon triangulation.

3.3.2. Process the Two-Sided and the Three-Sided Regions

For the two-sided and the three-sided regions, we consider generating the virtual sides to transform them into four-sided regions.

As shown in Figure 7, for the two-sided region, two virtual edges are generated at two common intersections *A* and *B* on both sides. Then, *n* transition lines are generated between *A* and *B*, which means *n* virtual points are inserted on the two virtual sides. When performing interpolation calculation, the coordinate values of *n* points on the same virtual side are the same. The coordinates of $A_i, i = 1, 2, \dots, n$ are the same as *A*. The coordinates of $B_i, i = 1, 2, \dots, n$ are the same as *B*.

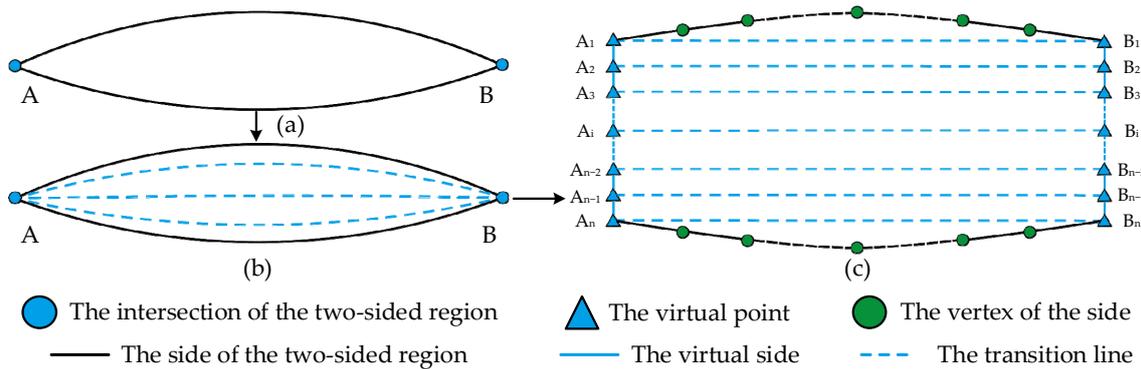


Figure 7. Schematic diagram of the two-sided region processing: (a) the original two-sided region; (b,c) the process and result of adding virtual sides and virtual points. The black dotted lines are omitted lines.

For the three-sided region, a virtual side is constructed at the intersection of any two sides. The transition lines are generated, whose number is the same as the number of vertices on the third side. Then, the same number of virtual points are inserted on the virtual side, and the coordinates of the virtual points are the same as that of the intersection where the virtual side is inserted.

3.3.3. Process Other Regions

For the five-sided region and regions that have more than five sides, first, search for the shortest side among all sides, and then search for the shorter side adjacent to this side for merging and connection. Repeat the above operation until this region is merged into a four-sided region.

3.3.4. Process All the Four-Sided Regions

After the above preprocessing, all the regions are transformed into four-sided regions. Since the synchronous forward method is used for interpolation, the number of vertices on the opposite side is required to be the same. However, the vertices cannot be directly added to the sides because of the requirement of forming manifold graphics. In this paper, we consider translating the sides of the four-sided region to the middle according to the specified distance, which will generate four new sides. As shown in Figure 8b, the same numbers of vertices are inserted on the opposite sides, which makes the same t of the vertices at the same position on the opposite sides. The t is the ratio of the distance from the vertex to the starting point to the side length. At last, the polygon triangulation method is used to model the newly formed sides and old sides, as shown in Figure 8c.

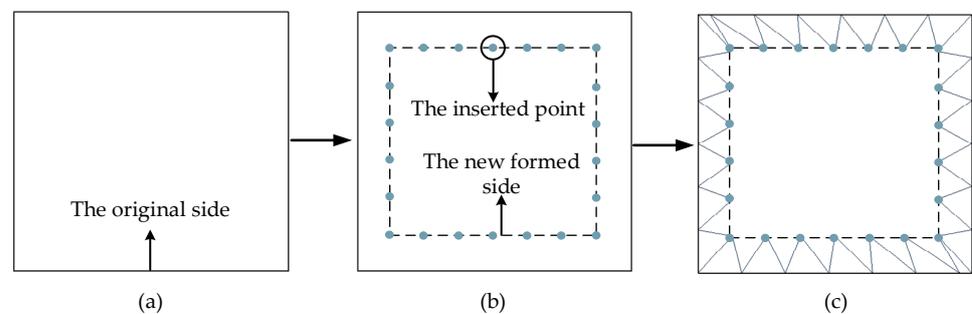


Figure 8. Schematic diagram of the side processing of the four-sided region: (a) the original four-sided region; (b) the formed sides and the points inserted on them; (c) the result of the polygon triangulation of the new and old sides.

3.4. Sided Region Modeling

After preprocessing, the four-sided regions can be modeled using bicubic Coons surface interpolation. Since the synchronous forward method is used to perform the Coons surface interpolation, the numbers of the vertices on the opposite sides are the same, which are m and n . This means that the number of the formed interpolation points in the grid will be $m \times n$. As shown in Figure 9, set the first group of opposite sides respectively as u_0 and u_1 , the second group of opposite sides respectively as v_0 and v_1 .

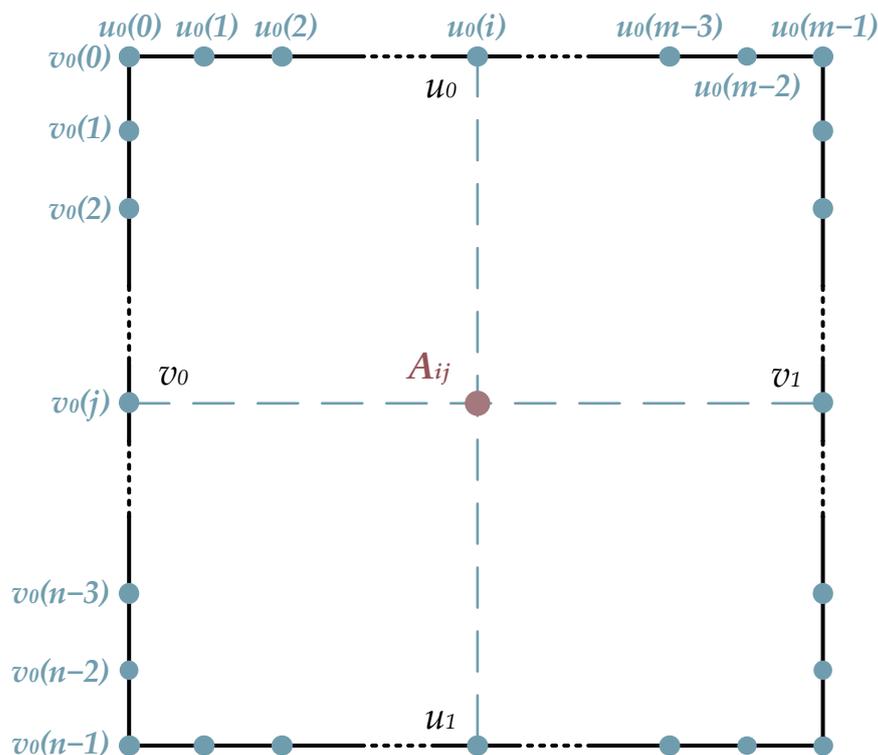


Figure 9. Schematic diagram of the Coons surface interpolation.

Based on Equation (9), the x , y , and z of each vertex can be calculated respectively by harmonic function, according to the coordinates of four corners and four relevant points, as well as the t of the four relevant points. Because the synchronous forward method is adopted, the t values of the opposite sides are the same.

Set the t of the first group sides u_0 and u_1 as $\{T_i | 0 \leq i < m, 0 \leq T_i \leq 1\}$, and the t of the second group sides v_0 and v_1 as $\{t_j | 0 \leq j < n, 0 \leq t_j \leq 1\}$. Then, set the point interpolated by the i point on the u_0 and the j point on the v_0 as A_{ij} , whose calculation formulas of coordinates are:

$$\begin{aligned}
 x = & u_0(i)_x \times F_0(t_j) + v_0(j)_x \times F_0(T_i) + u_1(i)_x \times F_1(t_j) + v_1(j)_x \times F_1(T_i) \\
 & - (u_0(0)_x \times F_0(T_i) + u_0(m-1)_x \times F_1(T_i)) \times F_0(t_j) \\
 & + (u_1(0)_x \times F_0(T_i) + u_1(m-1)_x \times F_1(T_i)) \times F_1(t_j)
 \end{aligned} \tag{11}$$

$$\begin{aligned}
 y = & u_0(i)_y \times F_0(t_j) + v_0(j)_y \times F_0(T_i) + u_1(i)_y \times F_1(t_j) + v_1(j)_y \times F_1(T_i) \\
 & - (u_0(0)_y \times F_0(T_i) + u_0(m-1)_y \times F_1(T_i)) \times F_0(t_j) \\
 & + (u_1(0)_y \times F_0(T_i) + u_1(m-1)_y \times F_1(T_i)) \times F_1(t_j)
 \end{aligned} \tag{12}$$

$$\begin{aligned}
 z = & u_0(i)_z \times F_0(t_j) + v_0(j)_z \times F_0(T_i) + u_1(i)_z \times F_1(t_j) + v_1(j)_z \times F_1(T_i) \\
 & - (u_0(0)_z \times F_0(T_i) + u_0(m-1)_z \times F_1(T_i)) \times F_0(t_j) \\
 & + (u_1(0)_z \times F_0(T_i) + u_1(m-1)_z \times F_1(T_i)) \times F_1(t_j)
 \end{aligned} \tag{13}$$

where the harmonic function is:

$$\begin{cases} F_0(t) = 2t^3 - 3t^2 + 1 \\ F_1(t) = -2t^3 + 3t^2 \end{cases} \quad (14)$$

$u_0(0)_x, u_0(0)_y, u_0(0)_z$ refers to respectively x, y, z coordinates of the first point on the u_0 ; $u_0(m-1)_x, u_0(m-1)_y, u_0(m-1)_z$ refer to, respectively, x, y, z coordinates of the last point on the u_0 ; $u_1(0)_x, u_1(0)_y, u_1(0)_z$ refers to respectively x, y, z coordinates of the first point on the u_1 ; $u_1(m-1)_x, u_1(m-1)_y, u_1(m-1)_z$ refer to, respectively, x, y, z coordinates of the last point on the u_1 ; $u_0(i)_x, u_0(i)_y, u_0(i)_z$ refer to, respectively, x, y, z coordinates of the point on the u_0 , whose t is T_i ; $v_0(j)_x, v_0(j)_y, v_0(j)_z$ refer to, respectively, x, y, z coordinates of the point on the v_0 , whose t is t_j ; $u_1(i)_x, u_1(i)_y, u_1(i)_z$ refer to, respectively, x, y, z coordinates of the point on the u_1 , whose t is T_i ; $v_1(j)_x, v_1(j)_y, v_1(j)_z$ refer to, respectively, x, y, z coordinates of the point on the v_1 , whose t is t_j .

Performing the interpolation calculation according to Equations (10)–(13), the modeling of this four-sided region can be completed, as shown in Figure 10.

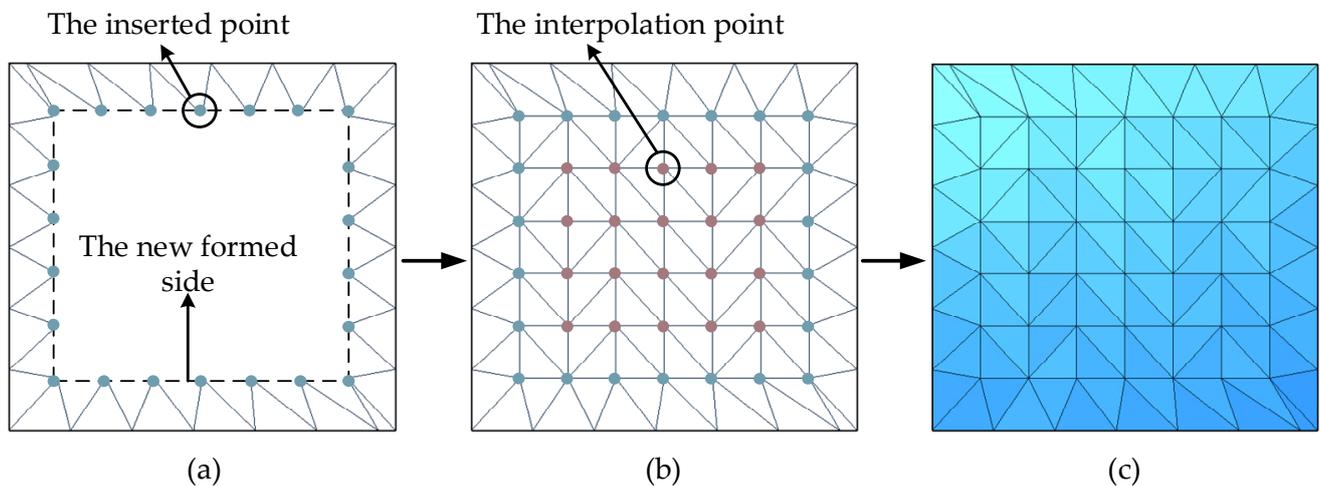


Figure 10. Schematic diagram of the four-sided region modeling:(a) the preprocessed four-sided region; (b,c) the process and the result of the Coons surface interpolation and modeling.

3.5. Combine Sub-Meshes

Multiple polygon data sets are set into one complete data set using the mature polygon data merging technology, which means all the sub-meshes are merged into a complete triangular mesh model. In the process of merging, all the polygon data will be extracted, but the extraction and expansion of point and cell attributes (scalar, vector, and normal) can only be carried out when multiple data sets contain them. As shown in Figure 11, we can obtain a complete triangular mesh model.

We use C++ to implement the relevant algorithms of the modeling method based on the Coons surface interpolation and test them on a 64-bit computer with Windows 10 professional edition. This computer has 3.70 Ghz Intel (R) Core (TM) i7-8700u, 32GB ram, and NVIDIA GeForce RTX 2080 GPU.

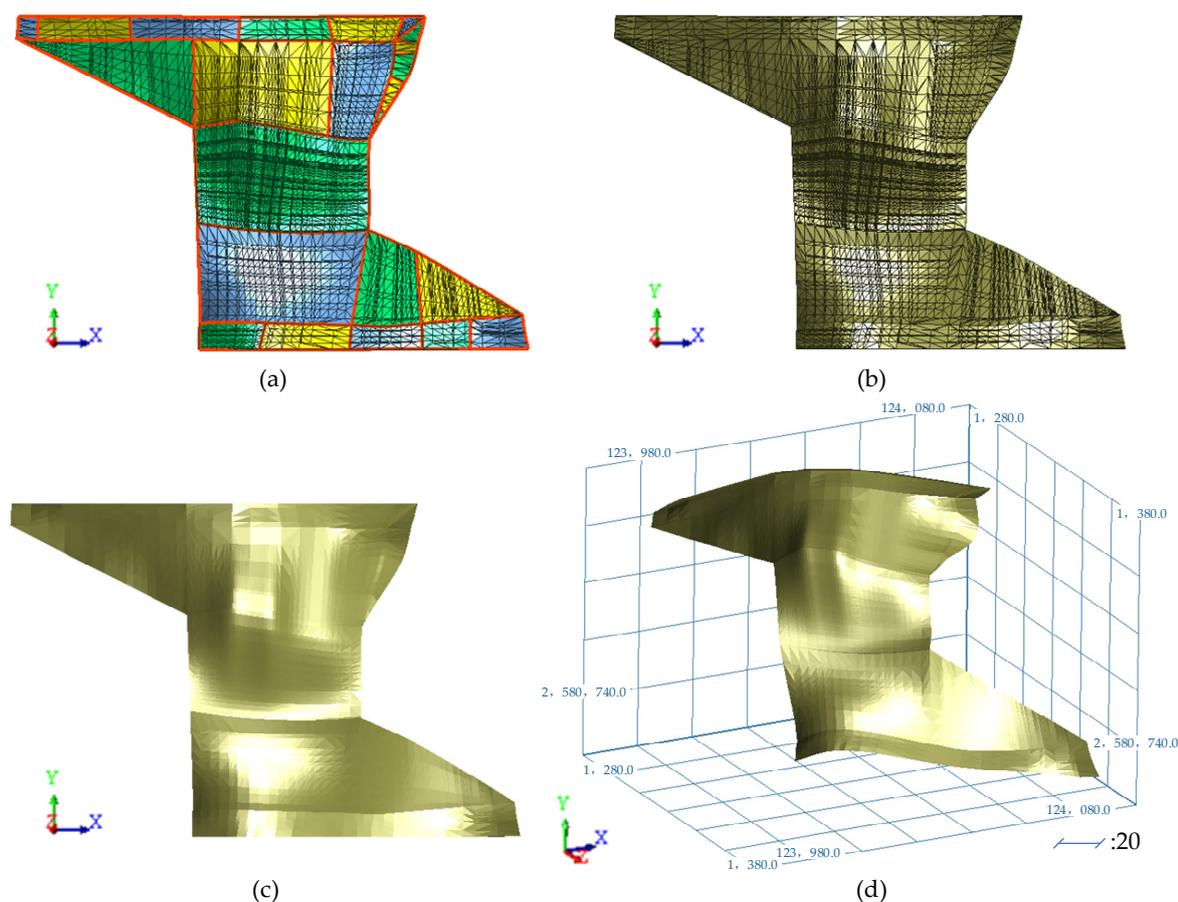


Figure 11. Schematic diagram of the sub-mesh combination: (a) the sub-meshes; (b–d) the complete model.

4. Results and Discussion

In this paper, five groups of different data are used to test the modeling method proposed in this work. These data are from the orebody of a tin mine in Yunnan, China, and are obtained in production and exploration. Based on these data, the original contour polylines can be produced through geological logging. The following experimental figures clearly show the process and the result of single closed-loop modeling and sub-mesh combination.

4.1. Examples

Figure 12 shows the processes of closed-loop division, single closed-loop modeling, and sub-mesh combination of the first group of cross-contour polylines. Figure 12a shows the original cross contour polylines of the orebody, each of which is in the same plane within the tolerance range. Then, the contour polylines are cut with the approximate planes. As shown in Figure 12b, each contour polyline is divided and grouped. After that, single closed-loop modeling is carried out on the contour polylines through the Coons surface interpolation, and the results are shown in Figure 12c. It can be seen that every closed loop has been modeled, but the sub-meshes are separate from each other. Finally, the sub-meshes are combined into a complete orebody model using polygon data merging technology, as shown in Figure 12d.

Figures 13 and 14 show the modeling process on contour polylines with different shapes. It can be seen that the algorithm implemented here can still carry out closed-loop division, single closed-loop modeling, and sub-mesh combination on more complex contour lines with good effects.

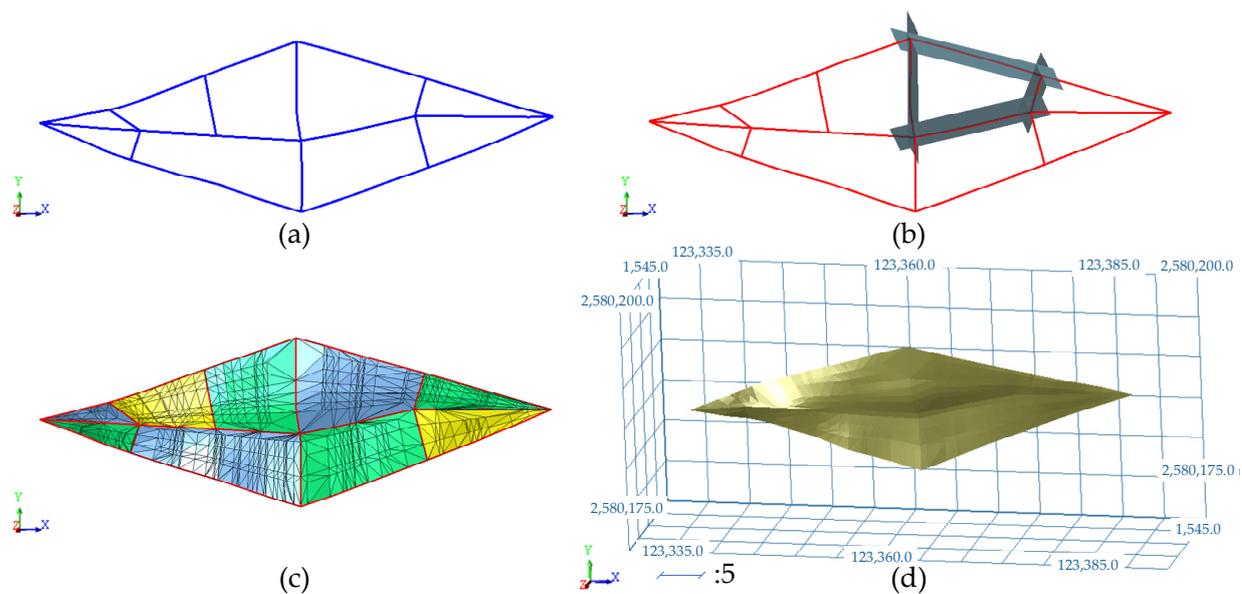


Figure 12. Experimental results of the first set of contour polylines: (a) the original contour polylines; (b) closed-loop division; (c) single closed-loop modeling; (d) sub-mesh combination.

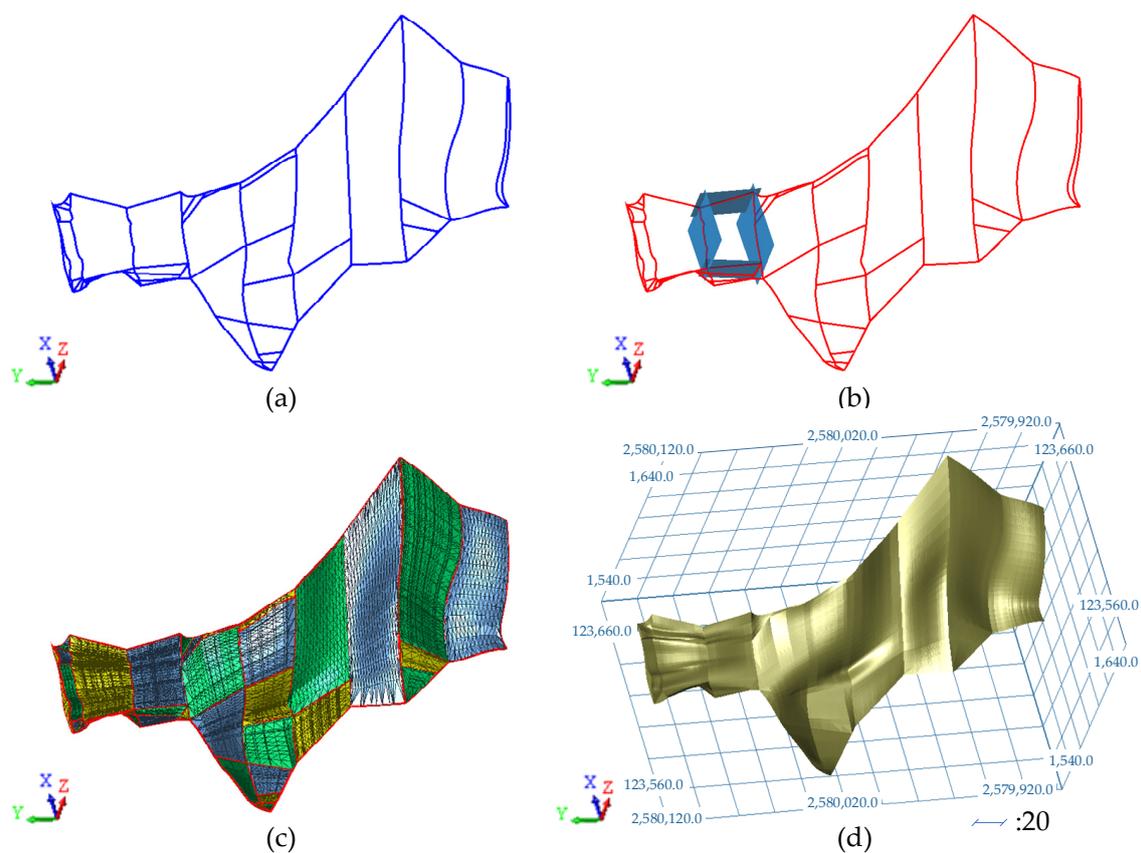


Figure 13. Experimental results of the second set of contour polylines: (a) the original contour polylines; (b) closed-loop division; (c) single closed-loop modeling; (d) sub-mesh combination.

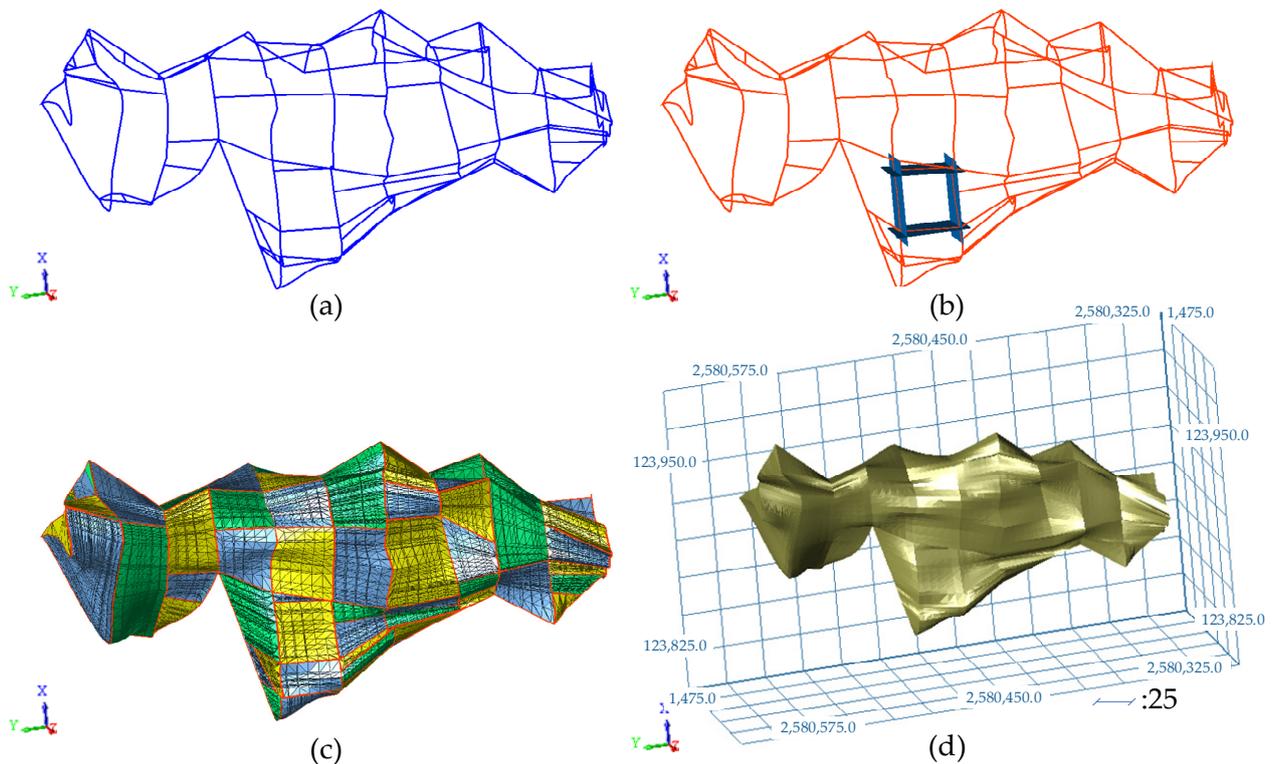


Figure 14. Experimental results of the third set of contour polylines: (a) the original contour polylines; (b) closed-loop division; (c) single closed-loop modeling; (d) sub-mesh combination.

Figure 15 shows the modeling process of more complex contour polylines, in which most of the contour polylines do not strictly intersect. Figure 15a shows the original contour polylines. First, search the original intersections of the contour polylines, and the result is shown in Figure 15b. The green points are the original intersections that need no further processing. It can be seen that only a few polylines possess the original intersections. Red points are isolated points that have no other contour polylines within the specified scope. Then, the contour polylines are moved to intersect within the tolerance range. Figure 15c shows the intersections after processing, and most of the contour polylines have intersected with each other. After that, the closed-loop division is carried out based on the intersected contour polylines, and the result is shown in Figure 15d. Based on the closed loops, sub-meshes can be constructed through the Coons surface interpolation. Figure 15e shows the modeling result of every closed loop. The sub-meshes are smooth but split from each other. Finally, the sub-meshes are combined to obtain a complete orebody model, as shown in Figure 15f.

As shown in Figure 16, the method is applied to complex orebodies with different shapes, and good orebody models are obtained, showing that this method has strong adaptability.

4.2. Discussion

The above experimental results show that the proposed method can deal with the contour polylines obtained by geological logging in mine production with good effects. Firstly, the constructed model conforms to the geological trend and is smooth in every mesh. Secondly, the closed-loop division can be carried out automatically, which will greatly improve the efficiency of modeling. Thirdly, since modeling between different regions will not affect each other, it can be carried out at the same time, which will greatly reduce the overall modeling time. However, this method still has some limitations that need to be further studied and solved. Firstly, due to the inherent limitations of the Coons technology, the model constructed by Coons surface interpolation is difficult to update,

and it is difficult to take faulting crosscutting into account, which will affect the accuracy of the model. In mine production, many orebodies are cross-cut by faults. For these orebodies, we should choose other modeling methods. Secondly, due to the complexity of the data in actual production, when this method is applied to some complex contour polylines, as shown in Figure 15, it can not complete all closed-loop divisions well, which needs to be specified manually. Finally, before Coons surface interpolation, all regions except simple single-sided regions need to be transformed into four-sided regions, which may affect the final modeling results by introducing numerical instability issues when adding virtual edges and vertices. Therefore, to further improve the efficiency and accuracy of the modeling of cross-contour polylines, it is necessary to develop a closed-loop division method with higher precision and an n-sided regions expansion method with less effect on interpolation modeling in the future.

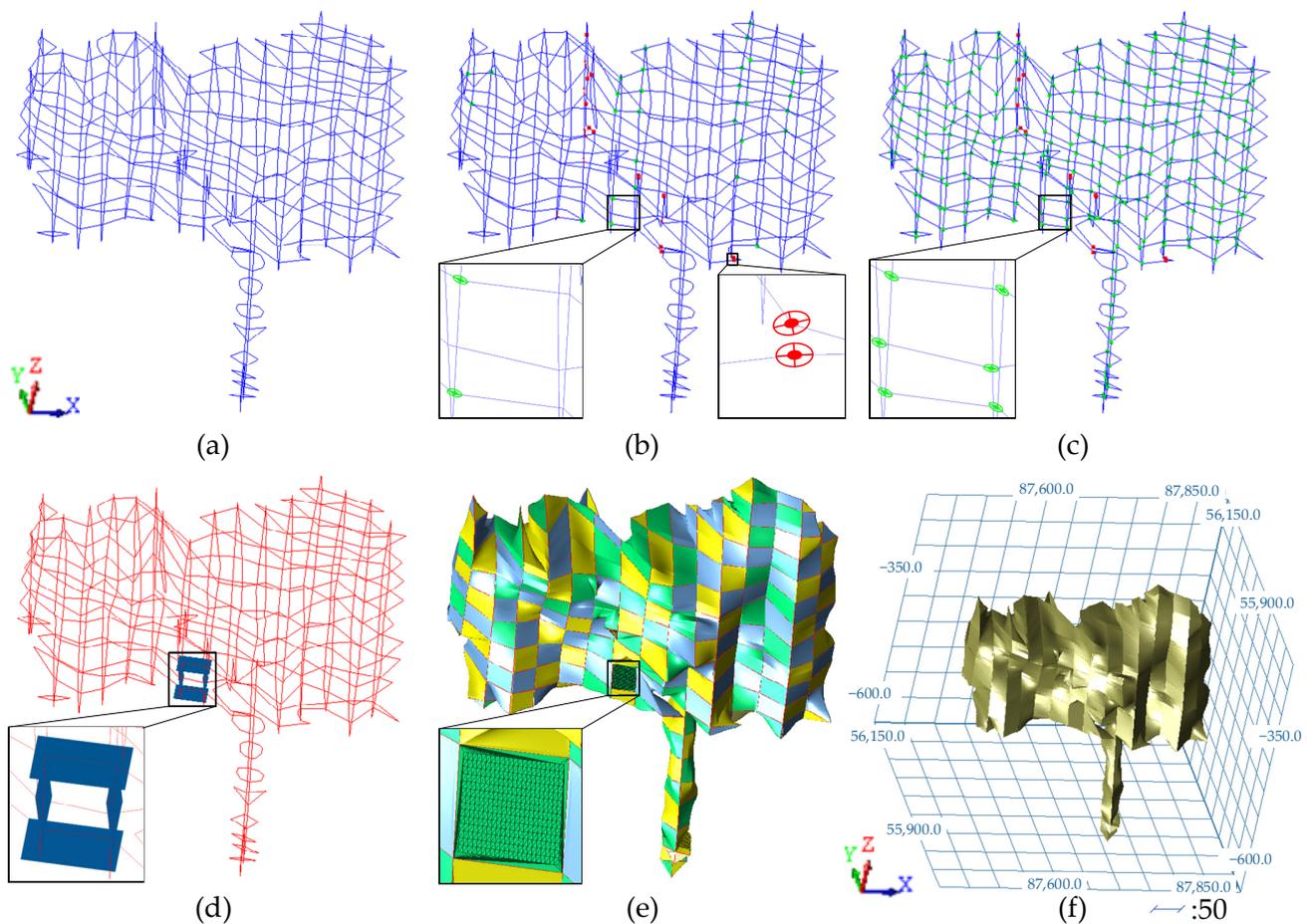


Figure 15. Experimental results of the fourth set of contour polylines: (a) the original contour polylines; (b) original intersections; (c) processed intersections; (d) closed-loop division; (e) single closed-loop modeling; (f) sub-mesh combination.

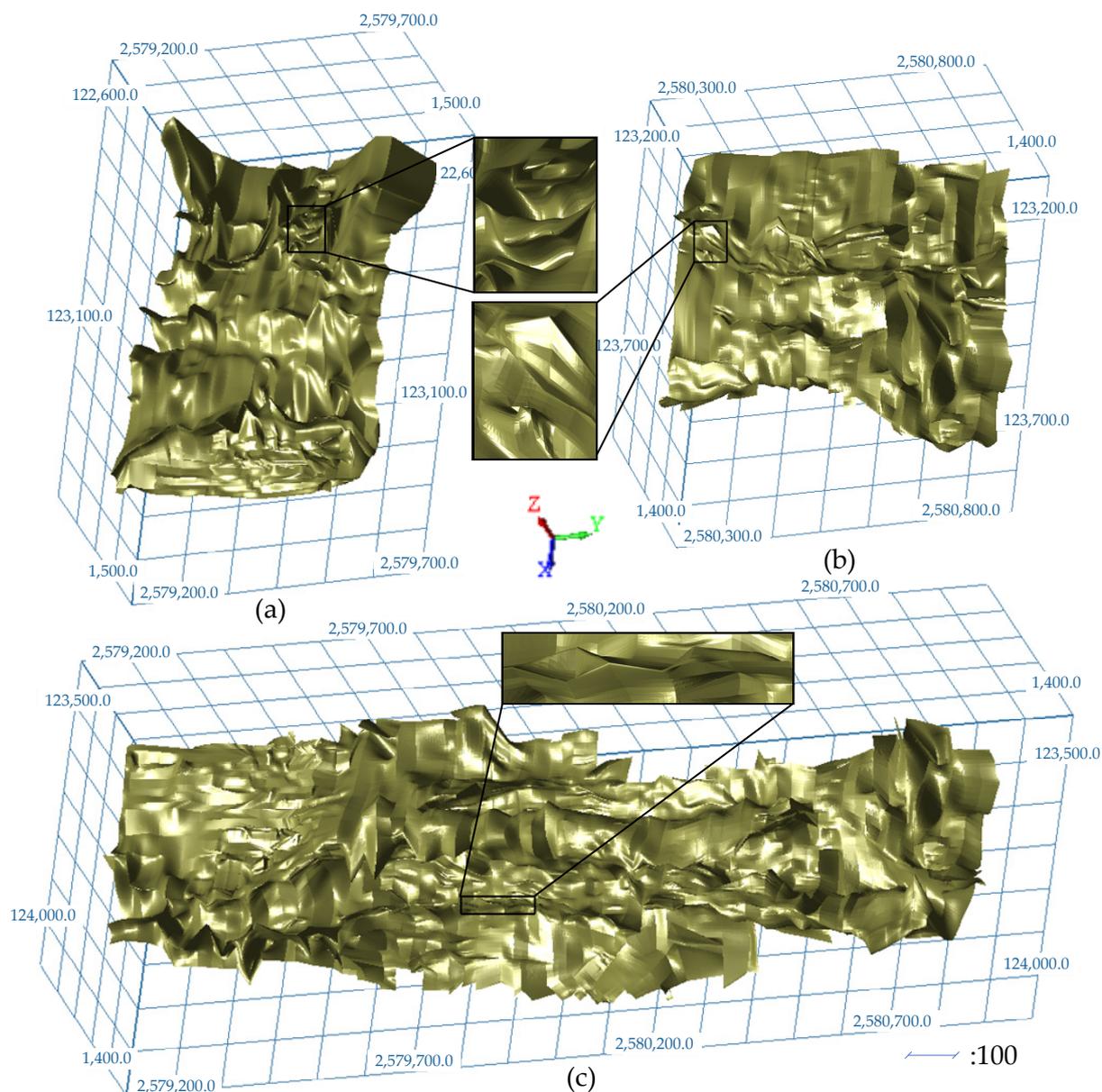


Figure 16. Experimental results of the fifth set of contour polylines: (a–c) the orebody models with different complex shapes.

5. Conclusions

We propose a modeling method based on bicubic Coons surface interpolation, which needs the contour polyline to be in an approximate plane within the tolerance range. The innovation of this paper is that we proposed a creative method of automatically dividing the closed loops, which will greatly reduce the modeling time. Additionally, the constructed sub-meshes can be combined without manual intervention. This method solves the problem that it is difficult to divide closed loops based on contour polylines with complex shapes, and it greatly improves the efficiency of modeling based on complex cross-contour polylines. Firstly, the contour polylines are cut by the approximate planes to divide the subspaces. Then, the divided polylines are grouped and connected according to the specified tolerance to form closed loops. Secondly, the Coons surface interpolation is used for modeling based on the closed-loop information. Finally, the constructed sub-meshes are combined to form a complete orebody model. Through experiments with multiple groups of cross-contour

polylines, the results show that the proposed method can accurately divide closed loops and model the contour polylines with good effects.

Author Contributions: Conceptualization, Z.W., L.B., D.Z. and Q.T.; methodology, Z.W., L.B., D.Z., J.Z. and Q.T.; software, Z.W., L.B., D.Z., J.Z. and Q.T.; formal analysis, Z.W., L.B., and D.Z.; data curation, Z.W., and D.Z.; writing—original draft preparation, Z.W.; writing—review and editing, Z.W., L.B. and D.Z.; visualization, Z.W., L.B., D.Z., and J.Z.; project administration, L.B., and M.J.; funding acquisition, D.Z. and M.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (52104171), the National Key R&D Program of China (2019YFC0605304), and the China Postdoctoral Science Foundation (2022T150740).

Acknowledgments: We thank the reviewers for their comments and suggestions to improve the quality of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, B.; Shi, B.; Song, Z. A simple approach to 3D geological modelling and visualization. *Bull. Eng. Geol. Environ.* **2009**, *68*, 559–565.
2. Jin, B.-X.; Fang, Y.-M.; Song, W.-W. 3D visualization model and key techniques for digital mine. *Trans. Nonferrous Met. Soc. China* **2011**, *21*, s748–s752. [[CrossRef](#)]
3. Mallet, J.-L. *Geomodeling*; Oxford University Press: New York, NY, USA, 2002; p. 624.
4. Jessell, M. Three-dimensional geological modelling of potential-field data. *Comput. Geosci.* **2001**, *27*, 455–465. [[CrossRef](#)]
5. Feltrin, L.; Mclellan, J.G.; Oliver, N. Modelling the giant, Zn–Pb–Ag Century deposit, Queensland, Australia. *Comput. Geosci.* **2009**, *35*, 108–133. [[CrossRef](#)]
6. Vollgger, S.A.; Cruden, A.R.; Ailleres, L.; Cowan, E.J. Regional dome evolution and its control on ore-grade distribution: Insights from 3D implicit modelling of the Navachab gold deposit, Namibia. *Ore Geol. Rev.* **2015**, *69*, 268–284. [[CrossRef](#)]
7. Tungyshbayeva, Z.; Royer, J.J.; Zhautikov, T.M. 3D modeling and resources estimation of a gold deposit, Zhungarie province, Kazakhstan. In Proceedings of the 13th SGA Biennial Meeting on Mineral Resources in a Sustainable World, Nancy, France, 24–27 August 2015; pp. 1759–1761.
8. Leapfrog. 2013. Available online: <http://www.leapfrog3d.com> (accessed on 1 February 2022).
9. Royer, J.J.; Mejia, P.; Caumon, G.; Collon-Drouaillet, P. 3&4D geomodeling applied to mineral resources exploration—A new tool for targeting deposits. In Proceedings of the 12th SGA Biennial Meeting, Uppsala, Sweden, 12–15 August 2013.
10. Coons, S.A. *Surfaces for Computer-Aided Design of Space Forms*; Technical Report MAC-TR-41; Massachusetts Institute of Technology: Cambridge, MA, USA, 1967.
11. Zhong, D.-Y.; Wang, L.-G.; Bi, L.; Jia, M.-T. Implicit modeling of complex orebody with constraints of geological rules. *Trans. Nonferrous Met. Soc. China* **2019**, *29*, 2392–2399. [[CrossRef](#)]
12. Hodgkinson, J.H.; Elmoultie, M. Cousins, siblings and twins: A review of the geological model’s place in the digital mine. *Resources* **2020**, *9*, 24. [[CrossRef](#)]
13. Kong, T.; Zhang, Y.; Fu, X. The model of feature extraction for free-form surface based on topological transformation. *Appl. Math Model* **2018**, *64*, 386–397. [[CrossRef](#)]
14. Yamany, S.; Farag, A. Surface signatures: An orientation independent free-form surface representation scheme for the purpose of objects registration and matching. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 1105–1120. [[CrossRef](#)]
15. Saini, D.; Kumar, S. Free-form surface reconstruction from arbitrary perspective images. In Proceedings of the IEEE International Advance Computing Conference, ITM, Gurgaon, India, 21–22 February 2014; pp. 1054–1059. [[CrossRef](#)]
16. Turk, G.; O’Brien, J.F. Modelling with implicit surfaces that interpolate. *ACM Trans. Graph.* **2002**, *21*, 855–873. [[CrossRef](#)]
17. Yuan, Z.; Yu, Y.; Wang, W. Object-space multiphase implicit functions. *ACM Trans. Graph.* **2012**, *31*, 1–10. [[CrossRef](#)]
18. Barthe, L.; Dodgson, N.A.; Sabin, M.A.; Wyvill, B.; Gaildrat, V. Two-dimensional Potential Fields for Advanced Implicit Modeling Operators. *Comput. Graph. Forum* **2003**, *22*, 23–33. [[CrossRef](#)]
19. Farin, G. Smooth interpolation to scattered 3D data. In *Surfaces in Computer Aided Geometric Design (Oberwolfach, 1982)*; Barnhill, R.E., Boehm, W., Eds.; North-Holland: Amsterdam, The Netherlands, 1983; pp. 43–63.
20. Cheng, F.H.; Wang, X.F.; Barsky, B.A. Quadratic B-spline curve interpolation. *Comput. Math. Appl.* **2001**, *41*, 39–50. [[CrossRef](#)]
21. Hu, S.-M.; Tai, C.-L.; Zhang, S.-H. An extension algorithm for B-splines by curve unclamping. *Comput. Des.* **2002**, *34*, 415–419. [[CrossRef](#)]
22. Park, H. B-spline surface fitting based on adaptive knot placement using dominant columns. *Comput. Des.* **2011**, *43*, 258–264. [[CrossRef](#)]

23. Krishnamurthy, A.; Khardekar, R.; McMains, S.; Haller, K.; Elber, G. Performing efficient NURBS modeling operations on the GPU. *IEEE Trans. Vis. Comput. Graph.* **2009**, *15*, 530–543. [[CrossRef](#)]
24. Selimovic, I. Improved algorithms for the projection of points on NURBS curves and surfaces. *Comput. Aided Geom. Des.* **2006**, *23*, 439–445. [[CrossRef](#)]
25. Wang, Q.; Hua, W.; Li, G.Q.; Bao, H.J. Generalized NURBS curves and surfaces. In Proceedings of the International Conference on Geometric Modeling and Processing, Beijing, China, 13–15 April 2004; pp. 365–368.
26. Randrianarivony, M. On global continuity of Coons mappings in patching CAD surfaces. *Comput. Des.* **2009**, *41*, 782–791. [[CrossRef](#)]
27. Farin, G.; Hansford, D. Discrete coons patches. *Comput. Aided Geom. Des.* **1999**, *16*, 691–700. [[CrossRef](#)]
28. Hugentobler, M.; Schneider, B. Breaklines in Coons surfaces over triangles for the use in terrain modelling. *Comput. Geosci.* **2005**, *31*, 45–54. [[CrossRef](#)]
29. Sapidis, N.S.; Besl, P.J. Direct construction of polynomial surfaces from dense range images through region growing. *ACM Trans. Graph.* **1995**, *14*, 171–200. [[CrossRef](#)]
30. Hoppe, H.; Deroose, T.; Duchamp, T.; McDonald, J.; Stuetzle, W. Surface reconstruction from unorganized points. *ACM SIGGRAPH Comput. Graph.* **1992**, *26*, 71–78. [[CrossRef](#)]
31. Edelsbrunner, H.; Mücke, E. Three-dimensional alpha shapes. *ACM Trans. Graph.* **1994**, *13*, 75–82. [[CrossRef](#)]
32. Fuchs, H.; Kedem, Z.M.; Uselton, S.P. Optimal surface reconstruction from planar contours. *Commun. ACM* **1977**, *20*, 693–702. [[CrossRef](#)]
33. Keppel, E. Approximating complex surfaces by triangulation of contour lines. *IBM J. Res. Dev.* **1975**, *19*, 2–11. [[CrossRef](#)]
34. Floater, M.S.; Reimers, M. Meshless parameterization and surface reconstruction. *Comput. Aided Geom. Des.* **2001**, *18*, 77–92. [[CrossRef](#)]
35. Macedonio, G.; Pareschi, M. An algorithm for the triangulation of arbitrarily distributed points: Applications to volume estimate and terrain fitting. *Comput. Geosci.* **1991**, *17*, 859–874. [[CrossRef](#)]
36. Meyers, D.; Skinner, S.; Sloan, K. Surfaces from contours. *ACM Trans. Graph. (TOG)* **1992**, *11*, 228–258. [[CrossRef](#)]
37. Ekoule, A.B.; Peyrin, F.; Odet, C.L. A triangulation algorithm from arbitrary shaped multiple planar contours. *ACM Trans. Graph.* **1991**, *10*, 182–199. [[CrossRef](#)]
38. Jones, M.W.; Min, C. A new approach to the construction of surfaces from contour data. *Comput. Graph. Forum* **2010**, *13*, 75–84. [[CrossRef](#)]
39. Zhong, D.-Y.; Wang, L.-G.; Jia, M.-T.; Bi, L.; Zhang, J. Orebody modeling from non-parallel cross sections with geometry constraints. *Minerals* **2019**, *9*, 229. [[CrossRef](#)]
40. Wu, Z.; Zhong, D.; Li, Z.; Wang, L.; Bi, L. Orebody modeling method based on the normal estimation of cross-contour polylines. *Mathematics* **2022**, *10*, 473. [[CrossRef](#)]
41. Song, R.; Nan, J. The design and implementation of a 3D orebody wire-frame modeling prototype system. In Proceedings of the Second International Conference on Information and Computing Science, Manchester, UK, 21–22 May 2009; Volume 2, pp. 357–360. [[CrossRef](#)]
42. Jessell, M.; Ailleres, L.; De Kemp, E.; Lindsay, M.; Wellmann, F.; Hillier, M.; Laurent, G.; Carmichael, T.; Martin, R. Next generation three-dimensional geologic modeling and inversion. In *Building Exploration Capability for the 21st Century*; Kelley, K.D., Golden, H.C., Eds.; Soc Economic Geologists, Inc.: Littleton, CO, USA, 2014; pp. 261–272.
43. Knight, R.H. Orebody solid modelling accuracy—A comparison of explicit and implicit modelling techniques using a practical example from the Hope Bay District, Nunavut, Canada. In Proceedings of the 6th International Mining Geology Conference, Darwin, Australia, 21–23 August 2006; pp. 175–183.
44. Lipuš, B.; Guid, N. A new implicit blending technique for volumetric modelling. *Vis. Comput.* **2005**, *21*, 83–91. [[CrossRef](#)]