J. COMBIN. MATH. COMBIN. COMPUT. 127a (2025) 2899--2912

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

A new algorithm for polynomial interpolation in complex surface modeling

Hao Li[1,✉]

[1] *Baoding Open University, Baoding, Hebei, 071000, China*

ABSTRACT
High-fidelity modeling of complex surfaces is the basis for accurate characterization of surface quality and realistic analysis of performance in the fields of digital process design of products and digital twin. This paper proposes to improve the new polynomial interpolation algorithm to improve the effect of the polynomial interpolation algorithm fitting in complex surface modeling through the center variable, and combines the moving least squares approximation function with the new polynomial interpolation algorithm to further optimize the effect of the complex surface modeling through the regular moving construction of the fitting surface by the local approximation method. It is found that the overall average error and standard deviation between the turbine blade surface roughness modeled based on the new polynomial interpolation algorithm and the roughness meter measurements are within 1.7 μm (0.7580-1.6715 μm), and the error is within the acceptable range. It is also found that using the method of this paper can save a lot of time and realize the rapid modeling of complex surfaces of the body. It also has good smoothness, which provides convenience for the subsequent processing of complex surface modeling. The new polynomial interpolation algorithm proposed in this paper provides a new idea for the research in the field of complex surface modeling, and can be applied to the actual production to assist the design and production of related products.

Keywords: polynomial interpolation, central variable, least squares approximation function, local approximation, surface modeling

1. Introduction

In curved surface modelling, Bernstein polynomial composites are widely used in geometric operations such as evaluation, dissection, free deformation, Trimmed curved surface representation, interconversion between Bézier monomorphic forms and tensor product Bézier forms, ascending order, and natural extension of Bézier surfaces, etc. Therefore, the study of composite algorithms of Bernstein polynomials is of important theoretical and practical significance [1-4].

✉ Corresponding author.
E-mail address: Liyouxiang2024@126.com (H. Li).
Received 18 February 2024; Revised 10 April 2024; Accepted 25 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-164

Surface modelling is widely used in shape design, manufacturing, CAD, computer graphics, meteorology and exploration and other types of scientific research and engineering design, especially in the measurement of complex surfaces modelling, the use of physical models to measure the data, the construction of a model with high accuracy, so as to carry out the design and manufacture of the physical is an effective means of new product development [5-7]. The given data points are generally divided into two categories: ordered and disordered. If a given data point is a series of scattered and disordered point sets, the reconstruction of such data points cannot be solved by conventional geometric modeling techniques, and the surface needs to be constructed by using scattered data interpolation technology, and the constructed surface is required to have a certain degree of smoothness in practical applications, so the scattered data interpolation technology and the surface smoothness problem have a wide range of research background and application value in CAD and CAGD systems [8-10].

Generally speaking, the goodness of an interpolated surface is measured from two aspects: the interpolation accuracy of the surface and the smoothness of the surface [11-12]. The higher the interpolation accuracy, the better the constructed surface is, and the smoothness of the surface is related to the size of the surface strain energy. There are many methods to construct polynomial interpolation surfaces for scattered spatial data points. One of the most common methods is to triangulate the scattered data and construct the surface on a triangular mesh. In the literature, each triangle mesh is divided into four sub-triangles, and each sub-triangle is constructed with five Bézier surface slices, which satisfy the C1 continuity between the sub-surface slices, but the overall surface can only reach the G1 continuity [13-14]. Although the literature only needs function values at the vertices of the triangles as interpolation conditions, but in the local area of each data point, a whole surface is formed, which sometimes is difficult to have the surface shape suggested by the data points. In the literature, the boundary continuity condition is used to establish a system of equations, and the C1 continuous surface is obtained by solving the equations, but the number of given data points should not be too many, and this method is not suitable if the number of data points is large, and the local adjustability of the constructed whole surface is poor. In addition, there are polynomial surface construction methods that interpolate boundary curves and cross-boundary derivatives in the triangular domain [15-17].

In this paper, the deficiencies of polynomial interpolation in complex surface modeling are improved, and a new algorithm for polynomial interpolation of special quadratic polynomials is proposed to improve the efficiency of the algorithm. The center variable in the algorithm is used to construct the polynomial interpolation function in the process of complex surface modeling, and the matrix form of polynomial interpolation is obtained by inverse deduction to improve the speed of polynomial interpolation and the degree of surface fitting. Subsequently, the polynomial interpolation algorithm is further optimized by proposing the moving least squares approximation function, and the local approximation method is used to reduce the fitting error of the new polynomial interpolation algorithm in the modeling of complex surfaces. The subdomain and basis function parameters of the new polynomial interpolation algorithm for complex surface modeling are then determined according to the selection principle. In this study, the feasibility of the new polynomial interpolation algorithm in complex surface modeling is verified through simulations and case studies of turbine blade and car body surface modeling.

2. Method

2.1. Principles of Polynomial Interpolation Application

Mathematics often uses polynomials as functions for interpolation, called polynomial interpolation. Common interpolation methods include Lagrange interpolation, Newtonian interpolation, and Ermitian interpolation. When polynomial interpolation is used, the more interpolation conditions that need to be met, the higher the number of polynomially interpolated curves will generally result. The higher the number of times, the higher the possibility that the curve will have too many wiggles or waves, so polynomial interpolation is seldom used in engineering practice.

Higher-order polynomial interpolation curves are not suitable for interpolation due to their drawbacks, and a single lower-order polynomial curve is difficult to use to characterize curves of complex shapes [18]. The only alternative is to piece together a segment of low order curves segment by segment under certain connectivity conditions, which is known as segmental interpolation. Curves defined by segmented interpolation are collectively called combinatorial curves. The corresponding surfaces defined by segmentation are called combinatorial surfaces.

In the process of constructing a spline curve, the spline interpolation algorithm is divided into a global algorithm and a local algorithm according to whether or not the change of the data points will cause the change of the whole curve. The so-called overall algorithm (also called global algorithm) means that the spline interpolation curve depends on all data points in the spline interpolation process. If a data point is changed, added or deleted, the resulting system of linear equations must be solved again. For localized algorithms, on the other hand, the change, addition or deletion of data points only affects the interpolation curve in a certain local area. Generally speaking, localized algorithms are relatively simple, less computationally intensive, faster and easier to implement.

Interpolation is an ancient and practical numerical method. In production practice, the function $f(x)$ sometimes can not directly write the expression, but can only give the function in a number of points of the function value or derivative value. According to this group of values, the construction of a simpler function $\theta(x)$, when x to take the value of this group of points of the variable, $\theta(x)$ should be equal to the corresponding $f(x)$ function value or derivative value, such a function of the approximation problem known as the interpolation problem. The interpolation problem of curved surfaces is one of the most important problems in curved surface modeling technology, which is expressed as the construction of an approximation function that can represent the whole curve or surface according to the known value points on the curve or surface. The interpolation method of curved surfaces has a wide range of applications in many fields, such as mechanical engineering, aerospace, ships, architecture, mapping, industrial modeling, etc. The interpolation method of curved surfaces has a wide range of applications in many fields, such as mechanical engineering, aerospace, ships, architecture, mapping, and industrial modeling.

2.2. Construction of a new algorithm for polynomial interpolation

2.2.1. Improved polynomial interpolation form. In order to balance the computational effort and the fitting effect in complex curve modeling, this paper aims to propose a new class of improved polynomial interpolation algorithms, which hopefully can reflect the advantages of polynomial interpolation. Improved polynomial interpolation is a special class of quadratic polynomials, in improved polynomial interpolation, the quadratic term coefficient matrix A changes its previous fixed form, and the quadratic term coefficient matrix A changes according to a certain rule at each iteration step, in order to achieve the improvement of algorithmic efficiency of the quadratic polynomial interpolation algorithm.

$P(X)$ is a quadratic polynomial interpolation, $X=(x_1,x_2,\ldots,x_n)^T$ is a n-dimensional variable, x_j is a component in the jth position of X, and x_j is said to be the center variable if a total of $n-1$ quadratic terms shaped like $x_jx_i\,(i=1,\ldots,j-1,j+1,\ldots,n)$ all appear in the functional expression of $P(X)$.

The existence of the center variable allows polynomial interpolation to improve the fit of the objective function in the process of modeling complex surfaces, and the relationship between the variables can be better considered. How to use the center variable and control the number of center variables becomes the key to designing an improved quadratic polynomial interpolation function.

Taking x_j as the center variable as an example, the polynomial interpolation function is constructed in the following form:

$$P(X)=\sum_{i=1}^{n}\left(a_i x_i^2+\sigma_{ij}\lambda_i x_j x_i+b_i x_i\right)+c \tag{1}$$

λ_i is the coefficient of the cross term $x_j x_i$, and $x_j x_i$ is the inner product of the component x_i with the central variable x_j, which can also be interpreted as the projection of the component x_i in the direction of the central variable x_j. σ_{ij} is a sign function when $i=j$, $\sigma_{ij}=0$, and when $i\neq j$, $\sigma_{ij}=1$. This is done in the sense of avoiding repetitive construction of the squared and quadratic terms repetitions of x_j.

The following inverts the matrix form of polynomial interpolation based on the polynomial interpolation function form (1), i.e:

$$\sum_{i=1}^{n}\left(a_i x_i^2+\sigma_{ij}\lambda_i x_j x_i\right)=\frac{1}{2}X^T AX \tag{2}$$

Polynomial interpolation yields the specific shape of the quadratic coefficient matrix A in its matrix form (3) based on its functional form of (1) and the equational relationship of (2):

$$\begin{pmatrix} 2a_1 & 0 & 0 & \lambda_1 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ddots & \lambda_{j-1} & 0 & \cdots & 0 \\ \lambda_1 & \cdots & \lambda_{j-1} & 2a_j & \lambda_{j+1} & \cdots & \lambda_n \\ 0 & \cdots & 0 & \lambda_{j+1} & \ddots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n & 0 & 0 & 2a_n \end{pmatrix} \tag{3}$$

Matrix (3) is a symmetric matrix of n row and n columns, where each element on the diagonal corresponds to the coefficient of the squared term in the polynomial interpolating functional form and is numerically doubly related. This is due to the fact that in the quadratic polynomial interpolation matrix form, all quadratic terms are artificially multiplied by 1/2 in order to give their matrices the same form as the quadratic coefficient matrices in the algebraic operations (both A) The only elements in matrix (3) other than the diagonal positions are the non-zero elements on the jth row and jth column, whose values correspond to the coefficients of the quadratic terms of the first type of polynomial interpolation, except for the squared terms.

If the coefficients of all quadratic terms in (1) are not 0, i.e., $\forall i\in\{1,2,\ldots,n\}$, $a_i,\lambda_i\neq 0$, the number of nonzero elements in matrix (3) is $3n-2$, and the number of zero elements is n^2-3n+2. When n is larger, the number of zero elements is much larger than the number of nonzero elements, for example, at $n=10$, the difference between the number of zero elements and the number of nonzero elements is 44, and when increasing to $n=50$, the difference between the number of zero elements and the number of nonzero elements will increase to 2204, so matrix (3) satisfies a sparse matrix definition of a sparse matrix. The advantage of sparse matrices is that when using mathematical software to perform calculations, sparse matrices are faster because commonly used mathematical software such as Mathematica and MATLAB are optimized for matrix operations and only operate on the non-zero elements of the matrix, which is one of the greater advantages of sparse matrices compared to dense matrices.

2.2.2. Moving Least Squares Based Optimization. In region Ω, the moving least squares approximation function can be defined by the following equation [19]:

$$u^h(x)=p^T(x)a(x),\forall x\in\Omega \tag{4}$$

where $p^T(x)=\left[p_1(x),p_2(x),\ldots,p_m(x)\right]$ is the basis function and $a(x)$ is a vector containing coefficients $a_j(x)$, $j=1,2,\cdots,m$ which are functions of spatial coordinates.

For a one-dimensional problem there are the following functions, the

Linear basis:

$$p^T(x)=[1,x]m=2 \tag{5}$$

Square base:

$$p^T(x)=\left[1,x,x^2\right]m=3 \tag{6}$$

The following functions are available for the two-dimensional problem, the

Linear basis:

$$p^r(x)=\left[1 \quad x \quad y\right]m=3 \tag{7}$$

Secondary base:

$$p^T(x)=\left[1 \quad x \quad y \quad x^2 \quad xy \quad y^2\right]m=6 \tag{8}$$

Three times base:

$$p^T(x)=\left[1 \quad x \quad y \quad x^2 \quad xy \quad y^2 \quad x^3 \quad x^2y \quad xy^2 \quad y^3\right]m=10 \tag{9}$$

Define the weighted discrete L_2-mode as:

$$\begin{aligned} J(x) &= \sum_{i=1}^{n} w_i(x)\left[p^T(x)a(x)-\hat{u}_i\right]^2 \\ &= \left[P\cdot a(x)-\hat{u}_i\right]^T\cdot W\cdot\left[P\cdot a(x)-\hat{u}_i\right] \end{aligned} \tag{10}$$

where $W_i(x)$ is the weight function of node i and for all x within the support domain of $W_i(x)$, there are $W_i(x)>0$, x_i denote the x values of node i, n is the number of nodes with weight function $W_i(x)>0$ in domain Ω, and matrices P and W are respectively:

$$P=\begin{bmatrix} p^T(x_1) \\ p^T(x_2) \\ L \\ p^T(x_n) \end{bmatrix}_{n\times m} = \begin{bmatrix} p_1(x_1) & p_2(x_1) & \cdots & p_m(x_1) \\ p_1(x_2) & p_2(x_2) & \cdots & p_m(x_2) \\ \vdots & \vdots & & \vdots \\ p_1(x_n) & p_2(x_n) & \cdots & p_m(x_n) \end{bmatrix}_{n\times m} \tag{11}$$

$$W=\begin{bmatrix} w_1(x) & L & 0 \\ L & L & L \\ 0 & L & w_n(x) \end{bmatrix} \tag{12}$$

And:

$$\hat{u}^T=\left[\hat{u}_1 \quad \hat{u}_2 \quad \ldots. \quad \hat{u}_n\right] \tag{13}$$

Special attention should be paid here to the fact that $\hat{u}_i, i=1,2,\ldots,n$ in Eqs. (10) and (13) is the nominal node value, which is usually not the node value of the unknown trial function $u^h(x)$.

From the condition that $J(x)$ obtains a minimal value, $\dfrac{\partial J(x)}{\partial a(x)}=0$ can be obtained, i.e:

$$A(x)a(x)=B(x)\hat{u} \tag{14}$$

Among them:

$$A(x)=P^{T}WP=B(x)P=\sum_{i=1}^{n}w_{i}(x)p\left(x_{i}\right)p^{T}\left(x_{i}\right) \tag{15}$$

$$B(x)=P^{T}W=\left[w_{1}(x)p\left(x_{1}\right)w_{2}(x)p\left(x_{2}\right)\cdots w_{n}(x)p\left(x_{n}\right)\right] \tag{16}$$

The moving least squares approximation function can be defined only if the matrix A in equation (14) is non-singular, and for A to be non-singular, it must be such that the rank of P is equal to m. It is therefore necessary to define the moving least squares approximation function if, for each sample point $x \in \Omega$ at least m of the weight functions are non-zero (i.e., $n \geq m$), and if the nodes in the Ω cannot have a particular distribution, e.g., distributed on a straight line, where by the The sample points can be either the nodes under consideration or Gaussian integration points.

Solve for $a(x)$ from Eq. (14) and substitute into Eq. (4) to obtain:

$$u^{h}(x)=\Psi(x)\hat{u}=\sum\psi_{i}(x)\hat{u}_{i},u^{h}(x)\equiv u_{i}\neq\hat{u}_{i},x\in\Omega \tag{17}$$

Style:

$$\Psi(x)=p^{T}(x)A^{-1}(x)B(x) \tag{18}$$

Or:

$$\psi_{i}(x)=\sum_{j=1}^{m}p_{j}(x)\left[A^{-1}(x)B(x)\right]_{ji} \tag{19}$$

A new polynomial interpolation algorithm based on least squares improvement is a local approximation method in complex surface modeling [20]. The basic idea is to use the values of its surrounding data points at each to-be-determined point to build multiple modeling surfaces moving according to certain rules, and the weighted average of the function values of this point on these multiple modeling surfaces is the function value of the desired to-be-determined point. The process is as follows.

1) Determine the size S of the locally fitted subdomains, which are generally chosen to be square or circular.

2) At each to-be-determined point P, determine all local fitting subdomains $\Omega_{i}\left(i=1,2,\cdots,N\right)$ containing points P of size S, and the data points $x_{j}\left(j=1,2,\cdots,n\right)$ in each subdomain.

3) At each local fitting subdomain Ω_{i}, select the appropriate modeling surface:

$$f_{i}(x,y)=p^{T}(x,y)A_{i} \tag{20}$$

Among them:

$$\begin{aligned}&p^{T}(x,y)=\left(p_{1}(x,y)\quad p_{2}(x,y)\quad\cdots\quad p_{m}(x,y)\right)\\&A_{i}=\left(a_{1}\quad a_{2}\quad\cdots\quad a_{m}\right)^{T}\end{aligned} \tag{21}$$

a_{i}, $i=1,2,\cdots,m$ are the coefficients to be determined, and note that $p^{T}(x,y)$ here is not required to be a complete polynomial basis function. The error equation corresponding to the n data points is then $\left(j=1,2,\cdots,n\right)$:

$$v_{j}=p^{T}\left(x_{j},y_{j}\right)A_{i}-z_{j} \tag{22}$$

is written in matrix form as:

$$v=PA_{i}-z \tag{23}$$

Eqs. $v^{T}=\left[v_{1}\quad v_{2}\quad\cdots\quad v_{n}\right]$, $z^{T}=\left[z_{1}\quad z_{2}\quad\cdots\quad z_{n}\right]$, then:

$$P=\begin{bmatrix} p_1(x_1,y_1) & p_2(x_1,y_1) & \cdots & p_m(x_1,y_1) \\ p_1(x_2,y_2) & p_2(x_2,y_2) & \cdots & p_m(x_2,y_2) \\ & \cdots & & \\ p_1(x_n,y_n) & p_2(x_n,y_n) & \cdots & p_m(x_n,y_n) \end{bmatrix} \tag{24}$$

According to the theory of least squares, the solution of the fitting coefficients of the modeled complex surface is:

$$A_i=\left(P^T P\right)^{-1} P^r z \tag{25}$$

4) Combine the information of the N complex surfaces at the pending point P and optimize it by taking the weighted average of the function values of point P on the N complex surfaces as the function value of point P, i.e.:

$$z\left(x_p,y_p\right)=\frac{\sum_{i=1}^{N} w_i f_i\left(x_p,y_p\right)}{\sum_{i=1}^{N} w_i} \tag{26}$$

where w_i is the weight assigned to each fitted subdomain.

2.3. Selection principles of relevant parameters in the improved algorithm

1) Selection of modeling subdomain. The fitting subdomain in complex surface modeling is generally square or circular. The fitting subdomain of each point to be determined can be the same, or it can be adjusted according to the distribution of the surrounding data points, but the following conditions should be met in general.

(1) It should be ensured that all fitting subdomains can cover the area to be fitted.

(2) out of the above derivation process can be seen, in order to find the coefficients to be determined a_i, $i=1,2,\cdots,m$, should be ensured that each fitting sub-domain, including the number of data points must be greater than the number of coefficients to be determined m.

(3) In order to maintain the characteristics of local approximation, the range of fitting sub-domain is not easy to be too large, so as not to reduce the accuracy and increase the calculation of large.

2) Selection of basis functions. The basis function $p^T(x)=\left[p_1(x),p_2(x),\cdots,p_m(x)\right]$ here is not limited to the selection of complete polynomials, it can be in various forms as follows:

$$\begin{aligned} p^T(x,y)&=\begin{bmatrix}1 & x & y\end{bmatrix} \\ p^T(x,y)&=\begin{bmatrix}1 & x & y & xy\end{bmatrix} \\ p^T(x,y)&=\begin{bmatrix}1 & x & y & xy & x^2\end{bmatrix} \\ p^T(x,y)&=\begin{bmatrix}1 & x & y & xy & x^2 & y^2\end{bmatrix} \\ p^T(x,y)&=\begin{bmatrix}1 & x & y & xy & x^2 & y^2 & x^2y\end{bmatrix} \end{aligned} \tag{27}$$

Basis functions of third or higher order are generally not used, as they tend to cause errors with fewer data points and do not improve the fitting results significantly.

3. Results and discussion

3.1. *Turbine blade surface modeling simulation experiment*

3.1.1. Development environment and tools. In this section, we take the turbine engine blade samples milled by 5-axis machine tool as an example to simulate and analyze the complex surface modeling, and verify the application of the new improved polynomial interpolation algorithm, and the development environment and tool description of the complex curve modeling system are shown in Table 1. The data collected by the acquisition equipment are imported into the development system, and the blade point cloud is firstly calculated based on PCA, the system interface will display the coordinates of the center of gravity of the point cloud and the direction vectors of the three main directions, and the blade point cloud will be automatically stratified according to the three direction vectors calculated, and the number of the calculated layer data groups and the curve shapes of the first group of layer data will be given. The next step is the new polynomial interpolation algorithm for complex curve modeling calculation, first for the base shape function interpolation generation, you can freely drag and drop to set the target resolution according to the need to set the current need for interpolation between the 15th and 16th groups of base shape function, after the start of the calculation, the system can give the interpolation results between the two groups of the measured base shape function, and give the morphology of the first five modes. After the calculation, the system can give the interpolation results between the two groups of measured base shape functions, and the fitting parameters of the first five modes, and you can choose to export the results.

Table 1. System development environment and tool specification table

Name	Explanation
Development machine	PC: Intel core i5, 2.5g CPG, 4g memory
Operating system	Windows 7, 64 bits
Development language	C++
System framework	MFC
Graphic API	OpenGL
Development tool	Visual Studio 2015

3.1.2. Simulation experiment results:

1) Complex surface modeling surface integrity analysis. In order to verify the integrity of the proposed polynomial interpolation algorithm in complex surface modeling, the reconstructed surface is compared with the original surface, and is analyzed mainly in terms of magnitude parameters. The commonly used magnitude parameters are the average surface roughness Sa, the root-mean-square roughness Sq, and the maximum height of the contour Sz. Among them, Sa mainly shows the deviation of the surface from the average value of the filtered surface, which is an important characterization method for the surface processing of the parts, Sq mainly uses the standard deviation to characterize the roughness of the surface of the parts, and Sz reflects the degree of undulation of the surface of the parts, and the larger the value is, the more obvious the surface undulation is. The main calculation methods of the three parameters are shown in equations (28), (29) and (30):

$$Sa = \frac{1}{A}\iint_A |L(x,y)|\,dxdy \tag{28}$$

$$Sq = \sqrt{\frac{1}{A}\iint_A L^2(x,y)\,dxdy} \tag{29}$$

$$Sz = |\max(L(x,y))| + |\min(L(x,y))| \tag{30}$$

Three different locations of the hyperbolic surfaces were selected, and the Sa, Sq, and Sz parameters of the original surfaces were calculated first, and then the Sa, Sq, and Sz parameters of the surfaces were also calculated by the polynomial interpolation algorithm, and the results of the two calculations are shown in Table 2. The results are shown in Table 2, where Sa_0, Sq_0, and Sz_0 denote the values of the original surfaces, and Sa_F, Sq_F, and Sz_F denote the values of the surfaces modeled by the new polynomial interpolation algorithm, which are differentiated by the subscripts in the experiments. After calculation and comparison, the average Sa error of the surface modeled by the new polynomial interpolation algorithm is 1.90%, followed by Sz at 13.67%, and the error of Sq is the largest at 19.00%, which is mainly due to the fact that there are more abnormal and mutated data in the measured data, which may be caused by large peaks and valleys, or irregular pits or bumps on the surface of the part. They will cause the increase of Sq and Sz error, while in the new polynomial interpolation algorithm proposed in this paper, the effect of mutation data and anomaly data is almost not taken into account, which leads to a larger error of Sq and Sz.

Table 2. The comparison of amplitude parameters

Unit(μm)	Position 1	Position 2	Position 3	Average
Sa_0	0.7854	0.6982	0.6529	0.7122
Sa_F	0.7952	0.6782	0.6425	0.7053
Error	1.25%	2.86%	1.59%	1.90%
Sq_0	0.0826	0.0254	0.0752	0.0611
Sq_F	0.0745	0.0152	0.0699	0.0532
Error	9.81%	40.16%	7.05%	19.00%
Sz_0	7.5286	4.5289	6.9985	6.3520
Sz_F	8.5296	5.2485	7.8259	7.2013
Error	13.30%	15.89%	11.82%	13.67%

2) Surface modeling roughness analysis results. In order to further verify the accuracy of the new polynomial interpolation algorithm in complex surface modeling, the roughness of the original surface and the fitted reconstructed surface of the turbine blade are compared with the measured roughness of the roughness meter respectively. The brand of the roughness meter used in the experiment is Mahr, and the parameters of the roughness meter are shown in Table 3.The measuring principle of the Mahr roughness meter is stylus scribing measurement, and correspondingly, the results of the original surface data and the fitted results of the new algorithm of polynomial interpolation are also computed by the layer data.

Table 3. Mahr roughness meter parameter

Project	Parameter
Principle of measurement	stylus
Measuring range	350μm, 180μm, 90μm (automatic switching)
Contour resolution	32nm, 16nm, 8nm (automatic switching)
Lc sample length	0.25mm, 0.8mm, 2.5mm
Lt sample length	1.5mm, 4.8mm, 15mm
Measuring force	About 0.7mN
Sampling number	1-16

The results of the roughness experiments were taken from 10 representative groups of data and the experimental results are shown in Table 4. In each experimental group, the overall average error is the mean value of the residuals, the standard deviation is the standard deviation calculation of the residual values, the original surface roughness is the three-dimensional roughness of the selected small surfaces, and the generated surface roughness is the roughness of the surfaces constructed according to the method, all in μm, and at the end of the calculation of the error of the two. From the calculation result data, it can be seen that the overall average error and standard deviation are within 1.7 μm (0.7580-1.6715 μm), which indicates that the error of the turbine blade generated based on the new polynomial interpolation algorithm proposed in this paper indicates that the error of the complex surfaces basically meets the requirements of the accuracy, and the validity of the new polynomial interpolation algorithm can be verified. The roughness errors of the complex surface modeling vary greatly, with the roughness errors of the 4th (26.44%), 5th (34.03%), 9th (25.99%) and 10th (20.29%) data sets being higher than 20%. However, as mentioned above, the main reason for this is that there will be more anomalies and mutations in the measured data, which may be caused by larger peaks and valleys. In summary, the errors are within the acceptable range and the accuracy of the new algorithm proposed in this paper can be verified.

Table 4. The surface result error is compared with roughness

Group number	Overall mean error (μm)	Standard deviation (μm)	Original surface roughness Sa(μm)	Surface roughness Sa(μm)	Roughness error
1	0.7580	1.3609	1.2248	1.1581	5.45%
2	1.1472	1.4312	1.3544	1.2635	6.71%
3	0.6956	1.1126	1.2948	1.5248	17.76%
4	0.4623	1.4477	1.2779	0.9400	26.44%
5	0.3215	1.4599	0.7368	0.9875	34.03%
6	0.7741	1.5965	1.4887	1.2875	13.52%
7	1.1954	1.1424	0.8461	0.9834	16.23%
8	0.7398	1.6715	1.1211	1.2245	9.22%
9	0.7583	1.3759	1.4736	1.8566	25.99%
10	0.5176	1.0078	0.9265	1.1145	20.29%

3.2. Complex surface modeling examples

3.2.1. Experimental Objects. In this paper, a model of Audi is chosen as a numerical example for 3D body complex surface modeling, and the feasibility of the new polynomial interpolation algorithm proposed in this paper is demonstrated through practical examples. By comparing the generation time of the surface model with other surface modeling methods, the efficiency of the new algorithm is demonstrated. The reliability of the proposed method in complex surface modeling is demonstrated through quality analysis and comparison test of the complex surfaces generated by the method.

3.2.2. Comparative analysis of time to generate surface models. In order to test the processing efficiency of the new polynomial interpolation algorithm proposed in this paper for modeling complex surfaces in vehicles, some methods for surface modeling are listed in this paper. And the time for generating surfaces of some methods is tested in detail. However, due to the lack of clear standards, the test data obtained are not objective and real standard data, and only have symbolic reference significance.

There are three common methods for modeling car body surfaces. The first is the traditional manual method of forward modeling along the “points, lines, and surfaces” in three-dimensional commercial software. The second method is to reverse the body surface modeling in 3D commercial software by processing the point cloud data. The third method is to take the four front, rear, side and top views of the car body as the background, import the unified 3D body curve grid template, generate the curve grid of the car model by adjusting the control points of the 3D body curve, and then generate the complex surface model based on the new polynomial interpolation algorithm proposed in this paper. Both Method I and Method II can generate relatively accurate car body models, but the workload is relatively large, and in general, it takes many working days to complete, so the time test is also difficult. Compared with the first two methods, Method III is relatively simple and generates body surfaces through a 3D curvilinear grid of the body. Therefore, this paper focuses on a detailed test of the time required for surface modeling in Method I and Method III.

The time taken by the two complex surface modeling methods is shown in Table 5, which provides a visual comparison of the time taken by the two modeling methods. The traditional manual method was tested in four steps: background image import and adjustment, model import, curve adjustment, and surface generation. The time used for each step was carefully organized and counted. After finishing and analyzing, the time of the four parts is 20-40min, 4.5-7.0min, 2.6-3.5h, and 36-50min, respectively. The modeling method applying the new polynomial interpolation algorithm of this paper is also divided into four steps, which are extraction of feature lines, modeling of body curves, addition and optimization of additional lines, and modeling of body surfaces. After finishing and analyzing, the time of the four parts is 35-45 min, 1.5-2.41 min, 12.52-25 min, and 2.5-4.8 min respectively, which shows that the method of this paper can save a lot of time and realize the rapid modeling of complex surfaces of the body. At the same time, it can be seen that the extraction of feature lines and the addition and optimization of additional lines in the body curve mesh generation take up most of the time in the process of modeling the complex surface of the body, which affects the efficiency of the whole body complex surface modeling.

Table 5. The different surface generation method time comparison

Modeling method			
Manual method		This method	
Step	Time	Step	Time
Background drawing import and adjustment	20-40min	2D feature line extraction	35-45min
Template import	4.5-7.0min	3D modeling	1.5-2.41min
Curve adjustment	2.6-3.5h	Additional line	12.52-25min
Surface modeling	36-50min	Surface modeling	2.5-4.8min
Total time	216.5-307min	Total time	51.5-77.2min

3.2.3. Quality analysis of body surface modeling. The results of the error analysis of the complex surfaces obtained by applying the new polynomial interpolation algorithm proposed in this paper for modeling are shown in Table 6. The distance error generated in the modeling of the complex surface of the car body based on the new polynomial interpolation algorithm is between 0.031-0.053mm on average. It can be seen that the fitting curve given by the algorithm in this paper has a good smoothness under the premise of a small error, which provides a convenient subsequent processing for the modeling of complex surfaces, and is a more ideal method.

Table 6. Surface modeling error analysis

Body parts	Data point number	Curve control vertex	Maximum error	Mean error
Bumper	18	10	0.148	0.031
Automobile A	20	20	0.101	0.035
Front bracket	15	11	0.199	0.047
Mean error	21	19	0.099	0.053
Mean error	18	8	0.092	0.048

4. Conclusion

This paper studies the optimization problem of polynomial interpolation in the process of modeling complex surfaces, and optimizes the new algorithm of polynomial interpolation by constructing polynomial interpolation function and with the help of moving least squares approximation algorithm. The application is verified by modeling the surfaces of turbine blades and car body surfaces milled by 5-axis machine tools as an example, and the results show that:

1) The average surface roughness error of the turbine blade surface modeled by the new polynomial interpolation algorithm is only 1.90%, and the overall average error and standard deviation between the results and those measured by the roughness meter are within 1.7 μm (0.7580-1.6715 μm), which is within the acceptable range of error, so that the accuracy of the new algorithm proposed in this paper can be verified.

2) The test of the car body surface modeling method applying the new polynomial interpolation algorithm of this paper is also divided into four steps, and the time for the four parts of feature line extraction, body curve modeling, additional line addition and optimization, and car body surface modeling are 35-45 min, 1.5-2.41 min, 12.52-25 min, and 2.5-4.8 min, respectively, which indicates that the method of this paper can save a lot of time and realize the car body surface modeling. Can save a lot of time and realize the rapid modeling of complex surfaces of the body. In addition, the distance error generated by the new polynomial interpolation algorithm for modeling complex surfaces of the car body is between 0.031-0.053mm on average.

3) The comprehensive results show that the new polynomial interpolation algorithm proposed in this

paper has great advantages in solving the complex surface modeling technology, and it is feasible to model a variety of complex surfaces.

Funding

1) Hebei Open University project Exploration of Network Teaching Team Operation Mechanism Based on System Cooperation Mode (Project approval number: WT202002).
2) Project: Scientific Research Fund project of Hebei Radio and Television University Study on the Strategy of Improving the Opening quality of Provincial Courses from the perspective of Open University (No.: YB201606).

References

[1] Gold, C. M. (2020). Surface interpolation, spatial adjacency and GIS. In Three dimensional applications in GIS (pp. 21-35). CRC Press.

[2] Konopatskiy, E., & Bezditnyi, A. (2020). Geometric modeling of multifactor processes and phenomena by the multidimensional parabolic interpolation method. In Journal of Physics: Conference Series (Vol. 1441, No. 1, p. 012063). IOP Publishing.

[3] Shankar, V., Narayan, A., & Kirby, R. M. (2018). RBF-LOI: Augmenting radial basis functions (RBFs) with least orthogonal interpolation (LOI) for solving PDEs on surfaces. Journal of Computational Physics, 373, 722-735.

[4] Kedward, L., Allen, C. B., & Rendall, T. C. (2017). Efficient and exact mesh deformation using multiscale RBF interpolation. Journal of Computational Physics, 345, 732-751.

[5] Bayona, V. (2019). Comparison of moving least squares and RBF+ poly for interpolation and derivative approximation. Journal of Scientific Computing, 81, 486-512.

[6] Khan, K., Lobiyal, D. K., & Kilicman, A. (2019). Bézier Curves and Surfaces Based on Modified Bernstein Polynomials. Azerbaijan Journal of Mathematics, 9(1).

[7] Li, X., & Li, S. (2020). A complex variable boundary point interpolation method for the nonlinear Signorini problem. Computers & Mathematics with Applications, 79(12), 3297-3309.

[8] Klingenberger, M., Hirsch, O., & Votsmeier, M. (2017). Efficient interpolation of precomputed kinetic data employing reduced multivariate Hermite Splines. Computers & Chemical Engineering, 98, 21-30.

[9] Alarcón, A., & Castro-Infantes, I. (2018). Interpolation by conformal minimal surfaces and directed holomorphic curves. Analysis & PDE, 12(2), 561-604.

[10] Yuksel, C. (2020). A class of C 2 interpolating splines. ACM Transactions on Graphics (TOG), 39(5), 1-14.

[11] Huang, Z., Carr, N., & Ju, T. (2019). Variational implicit point set surfaces. ACM Transactions on Graphics (TOG), 38(4), 1-13.

[12] Liu, T., Chen, M., & Lu, B. (2018). Efficient and qualified mesh generation for Gaussian molecular surface using adaptive partition and piecewise polynomial approximation. SIAM Journal on Scientific Computing, 40(2), B507-B527.

[13] Dai, J., & Krems, R. V. (2020). Interpolation and extrapolation of global potential energy surfaces for polyatomic systems by Gaussian processes with composite kernels. Journal of Chemical Theory and Computation, 16(3), 1386-1395.

[14] Tong, Y., Yu, L., Li, S., Liu, J., Qin, H., & Li, W. (2021). Polynomial fitting algorithm based on neural network.

ASP Transactions on Pattern Recognition and Intelligent Systems, 1(1), 32-39.

[15] Long, J., Liu, Y., Xing, S., Zhang, L., Qu, M., Qiu, L., ... & Shen, J. (2020). Optimal interpolation methods for farmland soil organic matter in various landforms of a complex topography. Ecological Indicators, 110, 105926.

[16] Lehto, E., Shankar, V., & Wright, G. B. (2017). A radial basis function (RBF) compact finite difference (FD) scheme for reaction-diffusion equations on surfaces. SIAM Journal on Scientific Computing, 39(5), A2129-A2151.

[17] Saye, R. I. (2022). High-order quadrature on multi-component domains implicitly defined by multivariate polynomials. Journal of Computational Physics, 448, 110720.

[18] Svetina Andrej. (2024). Approximation of holomorphic Legendrian curves with jet-interpolation. Journal of Mathematical Analysis and Applications(2P1).

[19] Saeb Faraji Gargari,Ziyang Huang & Sadegh Dabiri. (2024). An upwind moving least squares approximation to solve convection-dominated problems: An application in mixed discrete least squares meshfree method. Journal of Computational Physics112931-.

[20] Samardhi Yadav & Vishal Vaibhav. (2024). Transparent boundary condition and its effectively local approximation for the Schrödinger equation on a rectangular computational domain. Journal of Computational Physics113243-113243.