

Partial Differential Equation-based Surface Modelling and Applications

SHUANGBU WANG

A thesis submitted in partial fulfilment of the requirements
of Bournemouth University for the degree of

Doctor of Philosophy



October 27, 2021

Copyright statement

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognize that its copyright rests with its author and due acknowledgment must always be made of any material contained in, or derived from, this thesis.

Abstract

Due to its advantages in creating complicated 3D models with small data, good continuities, and physics-based deformations for better realism etc., partial differential equation (PDE)-based modelling provides a powerful technique of creating, manipulating, and animating 3D models, and has been attracting considerable attention in the community of computer graphics in the last three decades.

Various PDE-based approaches have been proposed for surface modelling. However, the following challenges have not been addressed. First, since PDE is a non-industry standard in CAD, CAM and CAE systems and lacks effective boundary control methods, the numerical solution for PDE-based surface modelling has few engineering applications. Second, there is no unified framework for solving different modelling problems. Existing research studies use special PDE-based mathematical models for specific applications, which cannot be applied on other occasions. Third, previous surface manipulation methods cannot deform shapes within arbitrary boundaries and usually involve heavy numerical calculations due to the use of numerical solutions. This thesis aims to address these challenges.

For the first challenge, a numerical solution using the finite difference method to a fourth-order PDE was presented. Based on this solution, an optimal conversion of PDE surfaces representing high-speed train heads into NURBS surfaces was developed, and a novel multi-objective aerodynamic optimization method of high-speed train heads was proposed, which is the first pipeline of using the PDE-based approach to optimize shapes in the CFD simulation. For the second challenge, a unified PDE mathematical model for surface modelling using analytical 2-, 3- and 4- sided PDE patches with C^n continuity was proposed. Based on the analytical solution, a PDE-based surface reconstruction method was proposed to generate optimal surfaces under the constraints of the feature curves in automotive styling design, and a PDE-based 3D modelling plugin was developed for Blender. For the third challenge, a physics-based method was presented to interactively manipulate surface shapes of 3D

models using the approximate analytical solution of a fourth-order PDE with C^1 continuity, and an interactive user interface was developed as a plug-in of Maya to facilitate surface manipulation.

Acknowledgements

I am deeply grateful to my supervisors Prof. Lihua You and Prof. Jian J. Zhang for their consistent support and guidance during my research work. I have benefited greatly from their unassuming approaches to research and science. I would never forget the countless meetings and discussions with Prof. You, which helped me undertake this research. I am very lucky to have met such a good supervisor.

I would like to thank Dr. Xiaosong Yang, who is enthusiastic and always ready to help with any questions that I have. I would also like to thank Nan Xiang and Ruibin Wang for their great help to this research work. Thanks to all colleagues in NCCA for the memorable time.

This research is supported by the PDE-GIR project which has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 778035. I also wish to thank the China Scholarship Council for the financial funding which supports my four-year PhD journey.

Special thanks to my parents and wife. Love you today, tomorrow and forever.

Declaration

This thesis has been created by myself and has not been submitted in any previous application for any degree. The work in this thesis has been undertaken by myself except where otherwise stated.

Contents

Abstract	ii
Acknowledgements	iv
Declaration	v
Table of contents	vi
List of figures	ix
List of tables	xv
1 Introduction	1
1.1 Background	1
1.2 Main challenges	3
1.3 Research aims and objectives	5
1.4 Contributions	7
1.5 List of publications	8
1.6 Thesis structure	9
2 Literature Review	11
2.1 PDE-based modelling	11
2.1.1 Numerical methods	12
2.1.2 Accurate analytical methods	13
2.1.3 Approximate analytical methods	14
2.2 Surface manipulation	15
2.2.1 Purely geometric surface manipulation	15
2.2.2 Physics-based surface manipulation	17
2.3 Surface representation conversion	18
2.4 Aerodynamic optimization design of the high-speed train	20
2.4.1 Parametric modelling	20
2.4.2 Aerodynamic analysis	21

2.5	Surface reconstruction	23
2.5.1	Parametric surfaces	23
2.5.2	CAD-generated surfaces	24
3	Numerical PDE surface patches with engineering applications	26
3.1	Numerical solution of PDE	27
3.2	PDE-based surface conversion	30
3.2.1	Overview for PDE-based surface conversion . . .	31
3.2.2	NURBS method	33
3.2.3	Least squares fitting	35
3.2.4	Genetic algorithm	35
3.2.5	Applications	40
3.2.6	Summary	48
3.3	PDE-based multi-objective aerodynamic optimization . .	48
3.3.1	Overview for aerodynamic optimization	49
3.3.2	Parametric modelling method	51
3.3.3	Optimization algorithm	56
3.3.4	CFD simulation method	58
3.3.5	Mesh generation	63
3.3.6	Results and discussion	66
3.3.7	Summary	75
3.4	Summary	75
4	Unified PDE-based surface modelling framework	77
4.1	Unified PDE mathematical model	78
4.1.1	The boundary conditions	79
4.1.2	Solutions	80
4.1.3	C^n continuity	82
4.2	PDE-based automotive styling design	83
4.2.1	Optimization	85
4.2.2	Experiments	86
4.2.3	Comparison with existing methods	91
4.3	User interface	94
4.3.1	Single Patch Creator	95

4.3.2	Multi-Patch Creator	97
4.3.3	Surface Modification	98
4.4	Summary	99
5	Physics-based surface manipulation using PDE	102
5.1	Method	103
5.1.1	Theory of plate bending	103
5.1.2	Mathematical model	104
5.1.3	Solution	105
5.2	Interaction design	107
5.2.1	Projective transformation	107
5.2.2	Boundary extraction	109
5.2.3	Mapping relationship	110
5.2.4	User interface	112
5.3	Results and comparison	115
5.4	Summary	118
6	Conclusion and future work	120
6.1	Conclusion	120
6.2	Future work	122
	Bibliography	124

List of Figures

3.1	The $I \times J$ finite difference grid	28
3.2	Framework of the proposed method	33
3.3	The basic structure of the genetic algorithm	37
3.4	Effects of the weight deformation on a NURBS surface	38
3.5	Process of conversion	41
3.6	Flowchart of the optimal conversion	42
3.7	Visualization of the errors between the PDE surface patches and NURBS surface patches	44
3.8	Computational complexity	45
3.9	Error comparison between the original PDE surface-represented (a) and NURBS surface-represented (b) high-speed train head where (c) uses different colors to visualize the errors between them	46
3.10	Effects of the required conversion accuracy and weight deformation	47
3.11	Overall design flow.	49
3.12	The half of the high-speed train head model. (a) Contour lines. (b) PDE surface patches	52
3.13	The effect of the contour line deformation on the shape of the half train head. (a) The deformations of the contour lines $\widetilde{A_1A_6}$ and $\widetilde{A_1E_1C_1}$ and their connected contour lines; (b) The PDE surface models before and after adjusting the two contour lines.	53

3.14	The influence of the three shape control parameters on the shape deformation of the PDE surface patch $A_3A_4B_4B_3$. (a) The PDE surface patch obtained from $a_1 = -2.49$, $a_2 = -2.49$, $a_3 = -1.68$; (b) The PDE surface patch obtained from $a_1 = -0.13$, $a_2 = -0.13$, $a_3 = -1.68$; (c) Shape comparison between (a) and (b).	55
3.15	The implementation process of NSGA-II.	56
3.16	LHS for three shape control parameters a_1 , a_2 and a_3 . . .	58
3.17	The simplified model and the computational domain of the high-speed train. (a) The simplified model of the whole train. (b) The computational domain.	60
3.18	The meshes for CFD simulation. (a) The surface meshes of the train head. (b) The longitudinal section of the train head in the computational domain. (c) The longitudinal section of the whole train in the computational domain. (d) The closer view of the nose cone and front spoiler of the train head.	65
3.19	The Pareto-optimal front in the global optimization stage.	68
3.20	The optimization results in the local optimization stage.	69
3.21	Comparison of the high-speed train head models. (a) The original model. (b) The globally optimized model. (c) The locally optimized model. (d) The cross-sections of the three models (the blue, grey and red lines represent original, globally optimized and locally optimized models, respectively).	70
3.22	Comparison of the pressure distributions of the original (a), globally optimized (b) and locally optimized (c) high-speed train heads (the left column) and tails (the right column).	71
3.23	Comparison of the streamlines around the original (a), globally optimized (b) and locally optimized (c) train heads (the left column), whole trains (the middle column) and train tails (the right column).	73
4.1	2-, 3- and 4-sided PDE surface patches.	79
4.2	Two adjacent PDE surface patches.	82

4.3	The feature curves in the styling design. The images taken from [Ford Media 2013; Jaguar 2017] show automotive clay models with black tapes, which represent feature curves.	84
4.4	Reconstructed surface and surface analysis for Example 1. (a) Input mesh of a car hood. (b) The part used for reconstruction from (a) with two feature curves (red curve). (c) Reconstructed surface. (d) Color map of error. (e) The zebra map of (b). (f) The zebra map of (c).	88
4.5	Reconstructed surface and surface analysis for Example 2. (a) Input mesh of a car hood. (b) The part used for reconstruction from (a) with three feature curves (red curve). (c) Reconstructed surface. (d) Color map of error. (e) The zebra map of (b). (f) The zebra map of (c). . . .	89
4.6	Reconstructed surfaces with C^0 , C^1 , C^2 and C^3 continuities and their surface analysis for Example 3. (a) Input mesh data, two parts for reconstruction and original zebra map. (b) Reconstructed C^0 surfaces with curvature combs, color map of error (the error range: $0 - 7.59 \times 10^{-3}$) and zebra map. (c) Reconstructed C^1 surfaces with curvature combs, color map of error (the error range: $0 - 7.56 \times 10^{-3}$) and zebra map. (d) Reconstructed C^2 surfaces with curvature combs, color map of error (the error range: $0 - 1.02 \times 10^{-2}$) and zebra map. (e) Reconstructed C^3 surfaces with curvature combs, color map of error (the error range: $0 - 1.35 \times 10^{-3}$) and zebra map. (The detail of curvature combs is shown in the red circle in the first row, and the blue and green colors in the second row represent the minimum and maximum errors respectively).	91

4.7	Reconstructed surfaces with C^0 , C^1 , C^2 and C^3 continuities and their surface analysis for Example 4. (a) Input mesh data, two parts for reconstruction and original zebra map. (b) Reconstructed C^0 surfaces with curvature combs, color map of error (the error range: $0 - 2.15 \times 10^{-2}$) and zebra map. (c) Reconstructed C^1 surfaces with curvature combs, color map of error (the error range: $0 - 2.32 \times 10^{-2}$) and zebra map. (d) Reconstructed C^2 surfaces with curvature combs, color map of error (the error range: $0 - 1.91 \times 10^{-2}$) and zebra map. (e) Reconstructed C^3 surfaces with curvature combs, color map of error (the error range: $0 - 3.21 \times 10^{-2}$) and zebra map. (The detail of curvature combs is shown in the red circle in the first row, and the blue and green colors in the second row represent the minimum and maximum errors respectively).	92
4.8	Reconstructed loft, sweep and PDE surfaces with error analysis. (a) Input mesh data. (b) The loft surface with two feature curves (red curve) and color map of error (the error range: $0 - 2.42 \times 10^{-2}$). (c) The sweep surface with one feature curve (red curve) and a backbone curve (purple curve), and color map of error (the error range: $0 - 2.74 \times 10^{-2}$). (d) The PDE surface with two feature curves (red curve) and color map of error (the error range: $0 - 8.21 \times 10^{-3}$). (The blue and green colors in the second row represent the minimum and maximum errors respectively)	93
4.9	PDE Panel in Blender.	95
4.10	2- 3- and 4-sided PDE surface patches.	96
4.11	Bézier curves (The left has 2 control points and the right has 5 control points).	96
4.12	The process of creating a 4-sided surface patch. (a)-(d) indicate the first, second, third and fourth selected curves. (e) is the generated PDE surface patch.	97
4.13	The curve network and the generated adjacent surface patches.	97

4.14	The imported curve network of a hat.	98
4.15	Two examples of imported curve networks and created surface models (The top is a hat model, and the bottom is body model).	99
4.16	The process of surface modification.	100
4.17	The modified surface patch.	100
5.1	The boundary of a deformation region. (a) the xyz geometric coordinate system. (b) the uv parametric coordinate system. (Point C is the centroid of the deformation region)	106
5.2	Projective transformation of the deformation region. (a) the global xyz coordinate system. (b) the local $x'y'z'$ coordinate system.	108
5.3	Extraction of boundary curve. (a) Voronoi diagram for a set of vertices. (b) Delaunay triangulation. (c) The boundary curve of α -shape.	109
5.4	The mapping relationship. (a) The $x'y'$ plane. (b) The uv plane.	111
5.5	The user interface.	112
5.6	The surface manipulation process of the user interface.	113
5.7	The influence of the moving direction and sensitivity of the cursor on the surface shape. (a) The deformation region and the initial position of the cursor. (b)-(c) The deformed shapes with the same sensitivity and different moving directions (the red dotted arrow) of the cursor (the red dotted square is the initial position of the cursor). (d)-(f) The deformed shapes with the same moving direction and different sensitivities of the cursor. (The sensitivities are 40, 70 and 110 in (d), (e) and (f), respectively)	114
5.8	Test results of four basic 3D models with different deformation regions surrounded by red curves.	115
5.9	Test results of three scanned 3D models with different deformation regions surrounded by red curves.	116

5.10	The comparison between different deformation methods. (a) The proposed method. (b) Delta Mush [Mancewicz et al. 2014]. (c) Laplacian Deformer [Sorkine et al. 2004]. (The first and third rows are deformed shapes, and the second and fourth rows are contour maps)	117
5.11	The realism comparison of deformed shapes in the side view. (a) Ground truth. (b) The proposed method. (c) Delta Mush [Mancewicz et al. 2014]. (d) Laplacian Deformer [Sorkine et al. 2004]. (The left and right columns are deformed shapes of a triangle and a rectangle, respectively)	118

List of Tables

3.1	Optimal number of control points and weight deformation ($\varepsilon = 1\%$)	43
3.2	Computation time of each surface patch	45
3.3	The design parameters and deformation directions of ten contour lines.	54
3.4	Comparison of CFD setup strategies of verified simulation models	64
3.5	Computational results of the four sets of meshes.	66
3.6	The ranges of the ten design variables.	67
3.7	Aerodynamic drag and lift coefficients of the original, globally optimized (GO) and locally optimized (LO) trains.	70
3.8	Comparison of the proposed method with other optimization methods.	74
4.1	Statistical data of C^0 , C^1 , C^2 and C^3 surfaces of Examples 3 and 4	90
4.2	Statistical data of loft, sweep and PDE surfaces	91
4.3	Comparison of the proposed method with another surface reconstruction method.	94

Chapter 1

Introduction

1.1 Background

Surface modelling is to apply mathematical methods to represent solid-appearing objects for computer graphics and computer aided design. It is widely applied in creative and industrial sectors, especially in the computer animation and game industry, to produce 3D geometric models. Up to the date of July 4, 2019, the output value of the global animation industry has reached approximately USD 250 billion and animation-related derivatives have as much as USD 500 billion [PRNewswire 2019b]. In addition, the global 3D mapping and modelling market is expected to grow at a compound average growth rate of around 13% during 2019-2024 [PRNewswire 2019a]. Surface modelling is one of the core techniques not only in the animation industry but also in the manufacturing and medical sectors, such as automobile design, simulation and virtual surgical guidance.

Partial differential equations (PDEs) were introduced into surface modelling about three decades ago to develop PDE-based surface modelling [Bloor & Wilson 1990]. Before that, they had already been applied in engineering and scientific calculations to describe the underlying physics. For example, a fourth-order PDE can be used to describe the underlying physics of bending deformations of a thin elastic plate subjected to lateral loads, and a wave equation, which is a second-order PDE, can be used

to describe the underlying physics of transverse vibrations of a tensed string [Epstein 2017]. Due to this nature, PDE-based surface modelling is physics-based and has the potential to create more realistic appearances of 3D models and 3D deformations.

The basic mechanism of PDE-based surface modelling is to create surfaces by the solution to a PDE with shape control parameters subjected to exact satisfaction of boundary conditions. It mainly has three advantages [Castro et al. 2008]. (1) Small data: complicated and detailed shapes can be described with few PDE surface patches. (2) Efficient shape deformation: applying shape control parameters can accurately control surface shapes with more degrees of freedom. (3) Good continuities: any high-order continuity between two adjacent PDE surface patches is readily achieved and naturally maintained. Due to these virtues, PDE-based surface modelling has been attracting considerable attention in the community of computer graphics in the last three decades.

Various PDE-based approaches have been proposed for surface modelling. According to different expression forms of PDE, there are fourth-order PDE with one shape control parameter [Ugail et al. 1999; Athanasopoulos et al. 2009; Sheng et al. 2010], fourth-order PDE with three shape control parameters [You et al. 2004b; Zhang & You 2002] and sixth-order PDE [Kubiesa et al. 2004; Zhang & You 2004]. Higher-order PDE is easier to generate surfaces with different shapes and satisfy higher continuous boundary conditions, but it is usually more difficult to find a solution. According to different boundary conditions, there are position (C^0) continuity [Sheng et al. 2010; Bloor & Wilson 1989b], position and tangent (C^1) continuities [Ugail et al. 1999; Athanasopoulos et al. 2009; Zhang & You 2002], and position, tangent and curvature (C^2) continuities [Kubiesa et al. 2004; Zhang & You 2004; You et al. 2004a]. Current PDE-based approaches investigate C^0 , C^1 , and C^2 continuities with second-order, fourth-order, and sixth-order PDEs, respectively. However, C^n ($n > 2$) continuities have not been investigated. The existing PDE approaches have found a lot of applications including pharmaceutical tablet design [Ahmat et al. 2014], facial geometry parameterisation [Sheng et al. 2011], cyclic animation [Castro et al. 2010], aircraft design [Athanasopoulos

et al. 2009], mesh compression [Shen et al. 2018], and CT data visualization [Chen et al. 2017]. Although these research studies make great contributions to the development of PDE-based surface modelling in the community of computer graphics, the challenges highlighted in Section 1.2 have not been addressed.

1.2 Main challenges

The main challenges for PDE-based surface modelling are numerical PDE surface patches with engineering applications, a unified framework able to deal with 2-, 3- and 4-sided surface patches with C^n continuities and interactive surface manipulation as discussed below.

Numerical PDE surface patches with engineering applications

Since it is difficult to analytically solve complex boundary conditions of PDE, most existing studies only developed analytical methods of creating 2-sided PDE surface patches. Numerical methods are more powerful in dealing with PDE with complex boundary conditions [Brown et al. 1998; Du & Qin 2005a]. However, numerical PDE surface patches are rarely used in engineering applications due to two reasons. First, PDE-based surface modelling has not become an industrial standard in computer-aided design (CAD), computer-aided manufacturing (CAM) and computer-aided engineering (CAE), which limits its engineering applications. In contrast, NURBS is an industry standard for the representation, design, and data exchange of geometric information processed by computers [Piegl & Tiller 2012]. Automatically converting PDE-based 3D models into NURBS representations will greatly promote engineering applications of PDE-based modelling. Second, a complex PDE 3D model usually consists of several PDE surface patches, and how to control the boundaries of all patches and then control the whole shape of the 3D model is a challenging issue. Unlike the shape of a 2-sided PDE surface patch that is controlled by two curves, the numerical method, which is used to deal with the PDE surface patches with complex boundary conditions, such as 4-sided PDE surface patches, is more difficult to control their boundaries.

Unified PDE-based surface modelling framework In the last three decades, most research studies of PDE-based modelling used specific PDE-based mathematical models for specific applications, which cannot be applied on other occasions. For example, Zhang & You [2002] constructed specific boundary conditions consisting of sine and cosine functions for the vase design, and Ahmat et al. [2014] developed a higher dimension version of a particular Biharmonic equation [Bloor & Wilson 1989b] for pharmaceutical tablet design. It is very challenging to build a unified PDE-based surface modelling framework. Here, ‘unified’ means the PDE and boundary conditions used to define the surface modelling framework are applicable to all the situations including different numbers of boundaries and/or different orders of continuities. There are two main reasons. First, boundary conditions for different occasions are quite different. For example, the boundary conditions of vase design are two closed curves [Zhang & You 2002], while those of face modelling are two open curves [Sheng et al. 2011]. Since two boundary curves can only generate a 2-sided surface patch and have less power to create various surface shapes, more complex boundary conditions are required to create these surface shapes such as 3- and 4-sided PDE surface patches. Unfortunately, most existing research studies can only generate 2-sided surface patches, and it is challenging to unify various boundary conditions such as 2-, 3- and 4-sided PDE surface patches. In addition, how to achieve 3- and 4-sided boundary constraints is an unsolved problem. Second, different surface continuities are required in different applications. For example, C^1 continuity is adequate for 3D character modelling [You et al. 2020] but an automobile surface usually requires a continuous curvature, i. e., C^3 or G^3 continuity [Hosaka 2012]. It is difficult to construct a unified continuity condition that satisfies not only position, tangent and curvature continuities but also higher-order continuities, i. e. C^n continuity. To sum up, how to analytically represent 2-, 3- and 4-sided PDE surface patches with high-order continuities for creating complicated PDE-based 3D models is a challenging task.

Interactive PDE-based surface manipulation Since PDE surfaces are controlled by underlying physical laws, they are natural and close

to the real world and have a potential to make the created shapes more realistic. In order to generate realistic shapes, some research studies try to manipulate surface shapes combined with PDE-based technologies [Du & Qin 2005a; You et al. 2006; Haixia et al. 2008]. However, current PDE-based surface manipulation methods have two limitations. First, they cannot deform a surface shape within a region with arbitrary boundary shapes. Second, since physics-based methods are difficult to find analytical solutions, numerical methods such as the finite element method, are commonly used to resolve their mathematical models. As a result, they usually involve heavy numerical calculations and slow responding time, which are not suitable for real-time and interactive applications. Therefore, it is challenging to develop interactive PDE-based surface manipulation with analytical deformation representations and tackle complex deformation regions with arbitrary boundary shapes.

1.3 Research aims and objectives

The aim of this thesis is to develop PDE-based surface modelling techniques for not only releasing its potential in engineering applications but also proposing new mathematical methods in both the creation and manipulation of surfaces. According to the aims and existing challenges, there are four main research questions that need to be addressed.

- **How to convert PDE surfaces into NURBS surfaces?** Since a PDE-based surface model usually consists of several PDE surface patches, the main difficulty of the conversion is to keep the continuities between different patches as well as reduce data size.
- **How to control PDE surface patches?** The shape of a surface patch is determined by the boundary curves and control parameters in PDE. In order to find an optimal shape, it is necessary to control the boundaries and parameters of all patches.
- **How to construct the unified framework of PDE?** According to different numbers of the boundary curves or different degrees of continuities, the PDE surface patch can be 2-sided, 3-sided and

4-sided or C^0 , C^1 , C^2 , \dots , C^n continuities. It is challenging to unify different forms of PDE surface patches.

- **How to analytically deform shapes within arbitrary boundaries?** In order to find the analytical solution of PDE for deforming shapes, the boundary conditions should be carefully defined at first. The condition of arbitrary boundaries increases the difficulty of solving PDE.

Based on the above questions, this thesis aims to achieve four main objectives:

- The first objective is to convert PDE surfaces into NURBS surfaces as well as keep the continuities and reduce the data size. The key is to develop an optimization method to minimize the error between PDE surfaces and NURBS surfaces and reduce the design variables to easily obtain the optimal results, which is discussed in Section 3.2 in Chapter 3.
- The second objective is to effectively control PDE surface patches. Since the shape of a single patch is defined by control parameters in PDE and boundary curves, the key is to explore a control method for both of them so that the PDE surface model can be applied to engineering applications such as the aerodynamic optimization of the shape, which is discussed in Section 3.3 in Chapter 3.
- The third objective is to develop a unified PDE-based surface modelling framework. Since PDE and boundary conditions can have different forms, it is unattainable to unify all of them. This objective assumes the PDE form is fixed and focuses on achieving unified boundary conditions including different numbers of boundaries and different orders of continuities, which is discussed in Chapter 4.
- The fourth objective is to manipulate surface shapes within arbitrary boundaries. In order to satisfy arbitrary boundaries, this objective concentrates on mapping arbitrary boundaries into an invariable boundary form and explores an effective mathematical model to generate deformations, which is discussed in Chapter 5.

1.4 Contributions

To fulfil the research aims, this thesis has made several major contributions which are listed as follows.

Numerical PDE surface patches with engineering applications:

- A numerical solution for a fourth-order PDE with three control parameters is presented, and the finite difference method is applied to solve 4-sided boundary conditions of a PDE surface patch.
- Based on the obtained numerical solution, an optimal conversion of PDE surfaces representing high-speed train heads into NURBS surfaces is developed. The least squares fitting and the genetic algorithm are combined to obtain the optimal conversion between PDE surfaces and NURBS surfaces.
- Based on the same numerical solution, a multi-objective aerodynamic optimization method of high-speed train heads is proposed to optimize the head shape. The high-speed train head is described by the developed numerical PDE surface patches with more details and fewer design variables, and the global and local optimization methods are developed to optimize both global and local shapes.

Unified PDE-based surface modelling framework:

- A unified PDE mathematical model for surface modelling with analytical 2-, 3- and 4-sided PDE surface patches is proposed, and boundary conditions defining C^n continuity are formulated.
- Based on the unified model, PDE-based surface reconstruction is proposed to generate optimal surfaces with C^n continuity under the constraints of two feature curves in automotive styling design.
- A user interface of PDE-based surface modelling is developed as a plug-in in the 3D animation and modelling software Blender. It can create 2-, 3- and 4-sided PDE surface patches from two, three and four 3D curves, respectively, and generate complete 3D models from input curve networks.

Interactive PDE-based surface manipulation:

- A physics-based surface manipulation method with a fourth-order PDE involving a sculpting force is proposed to achieve surface deformations and C^1 continuity.
- A mapping method, which maps a deformation region from a 3D space to a 2D parametric plane, is developed to simplify the resolution of the PDE originating from the bending deformation of a thin elastic plate.
- An interactive user interface of the proposed surface manipulation method is developed, which has been integrated into the software package Maya as a plug-in and can be used to achieve physics-based surface manipulation in deformation regions with arbitrarily complicated boundary shapes.

1.5 List of publications

Relevant publications

- [1] **Wang, S.**, Xiang, N., Xia, Y., You, L. and Zhang, J., 2021. Real-time surface manipulation with C^1 continuity through simple and efficient physics-based deformations. *The Visual Computer*, pp.1-13.
- [2] **Wang, S.**, Wang, R., Xia, Y., Sun, Z., You, L. and Zhang, J., 2021. Multi-objective aerodynamic optimization of high-speed train heads based on the PDE parametric modeling. *Structural and Multidisciplinary Optimization*, pp.1-20.
- [3] **Wang, S.**, Xia, Y., Wang, R., You, L. and Zhang, J., 2019. Optimal NURBS conversion of PDE surface-represented high-speed train heads. *Optimization and Engineering*, 20(3), pp.907-928.

Other irrelevant publications

- [4] **Wang, S.**, Xia, Y., You, L. and Zhang, J., 2020. Reconstruction of Curve Networks from Unorganized Spatial Points. *Journal of Universal Computer Science (J. UCS)*, 26(9), pp.1265-1280.

- [5] Zhu, Z., Chaudhry, E., **Wang, S.**, Xia, Y., Iglesias, A., You, L. and Zhang, J.J., 2021, June. Shape Reconstruction from Point Clouds Using Closed Form Solution of a Fourth-Order Partial Differential Equation. In International Conference on Computational Science (pp. 207-220). Springer, Cham.
- [6] Xia, Y., **Wang, S.**, You, L. and Zhang, J., 2021, June. Semantic Similarity Metric Learning for Sketch-Based 3D Shape Retrieval. In International Conference on Computational Science (pp. 59-69). Springer, Cham.
- [7] Xia, Y., **Wang, S.**, Li, Y., You, L., Yang, X. and Zhang, J.J., 2020. Single Color Sketch-Based Image Retrieval in HSV Color Space. In Transactions on Computational Science XXXVII (pp. 77-90). Springer, Berlin, Heidelberg.
- [8] Xia, Y., **Wang, S.**, Li, Y., You, L., Yang, X. and Zhang, J.J., 2019, June. Fine-grained color sketch-based image retrieval. In Computer Graphics International Conference (pp. 424-430). Springer, Cham.

1.6 Thesis structure

The following part of this thesis contains five more chapters:

- Chapter 2 presents a literature review on the related research topics, including PDE-based modelling, surface manipulation, surface representation conversion, aerodynamic optimization and surface reconstruction.
- Chapter 3 presents a numerical solution of a fourth-order PDE to create 4-sided surface patches, and applies it in optimal conversion of PDE surfaces representing high-speed train heads into NURBS surfaces and multi-objective aerodynamic optimization of high-speed train heads.
- Chapter 4 presents a novel unified PDE mathematical model defining analytical 2-, 3- and 4-sided surface patches with C^n continuity, and applies it in automotive styling design and development of a

user interface as a plug-in in Blender.

- Chapter 5 presents a physics-based method using PDE to interactively manipulate surface shapes of 3D models with C^1 continuity and a user interface as a plug-in in Maya
- Chapter 6 concludes the thesis and discusses the future work.

Chapter 2

Literature Review

In this chapter, the related works on the key techniques of the thesis including PDE-based modelling and surface manipulation are first discussed. Then, other related works of some applications of the proposed methods such as surface representation conversion, aerodynamic optimization and surface reconstruction are also reviewed.

2.1 PDE-based modelling

Physics-based surface modelling can create more realistic 3D models. There are various physics-based surface modelling methods such as the finite element method, finite difference method and mass-spring systems. However, these physics-based surface modelling methods involve heavy numerical calculations and are not applicable to the situations where real-time performance is required. PDE-based modelling is physics-based and can be solved analytically. This thesis focuses on PDE-based surface modelling. In what follows, only the references related to PDE-based surface modelling will be reviewed.

PDEs were introduced in surface modelling by Bloor and Wilson about three decades ago [Bloor & Wilson 1990]. After that, PDE-based surface modelling attracts a lot of research attention. Various numerical, accurate analytical and approximate analytical solutions have been developed to promote their applications. The biggest problem for PDE-based

surface modelling is how to solve PDEs. In the main challenges of the thesis, the key is to create the surface by solving PDEs, and different solutions produce different modelling processes, which impact the surface precision and computational cost. Existing solution methods of PDEs, i. e., numerical methods, accurate analytical methods and approximate analytical methods, will be briefly reviewed below.

2.1.1 Numerical methods

Numerical methods are most effective in solving PDEs for surface modelling applications. Popular numerical methods are the finite element method, finite difference method, and direct discretization of polygonal models using a discrete Laplace-Beltrami operator. Since there are many publications on numerical PDE-based surface modelling, it is impossible to review all of them. In what follows, only some of them are reviewed.

A B-spline finite element method was proposed in [Brown et al. 1998] and used to approximate PDE surfaces. A bivariate B-spline finite element method was developed in [Du & Qin 2005a] and applied to tackle dynamic PDE surface modelling. By considering Laplacian and Laplacian gradient energies leading to biharmonic and triharmonic equations, respectively, and viewing the discretization of biharmonic and triharmonic equations as transformation of a mixed element discretization, a mixed finite element method was investigated in [Jacobson et al. 2010] to solve the biharmonic and triharmonic equations and address variational surface modelling. Through presenting a novel technique to evaluate the finite element basis functions and coupling the finite element method with a hybrid loop and Catmull-Clark subdivision algorithm, a numerical simulation method was examined in [Pan et al. 2014] for hybrid subdivision surface design using geometric PDEs.

By solving a Euler-Lagrange equation with the finite difference method, the smoothing properties were achieved by reducing the amplitude of various high frequency Fourier harmonics in surfaces [Bloor et al. 1995]. Using the finite difference method to discretize a general sixth-order geometric PDE, the problems of surface processing and modelling including

creation of high-order continuous surfaces are efficiently solved in [Liu & Xu 2007]. Employing the finite difference discretization and variational interpolating approach with the localized iterative solver, an implicit fourth-order PDE and an implicit second-order PDE were numerically integrated in [Du & Qin 2005b] to achieve shape design of solid models. Except for using the finite element method, Du & Qin [2005a] also applied the finite difference method to the framework of dynamic PDE-based surface design and achieved real-time performance. In order to develop facial animation, You et al. [2009] presented an adaptive finite difference method to solve a fourth-order PDE and limit the surface regions where significant deformations exist.

Using the umbrella operator to linearly approximate the Laplacian operator at each vertex of polygon models and numerically integrating a diffusion equation involving the Laplacian and the second Laplacian, rough features from irregular data are rapidly removed to portray a smooth surface [Desbrun et al. 1999]. In order to smooth arbitrary triangle meshes while satisfying G^1 boundary conditions, a new algorithm was presented in [Schneider & Kobbelt 2001] to numerically solve a PDE with the discrete Laplace-Beltrami operator and 1-neighborhood based discretization of the mean curvature normal at a vertex. The discrete Laplace-Beltrami operator was also used to obtain variational minimization for surface deformations in [Botsch & Sorkine 2008], estimate surface normal in [Park et al. 2015], and perform surface reconstruction in [Stanko et al. 2016].

2.1.2 Accurate analytical methods

Accurate analytical methods exactly satisfy both PDEs and boundary conditions. It is very difficult or even impossible to obtain accurate analytical solutions of PDEs when boundary conditions are complicated. Therefore, the existing work only developed accurate analytical solutions for 2-sided PDE patches subjected to some simple boundary conditions. Treating blending design as a boundary value problem and considering some simple boundary functions, the accurate analytical solutions for x and y components were obtained from a second-order PDE and the

accurate analytical solution for z component was obtained from a fourth-order PDE to create parametric surfaces [Bloor & Wilson 1989a]. For primary surfaces represented with a combination of hyperbolic sine and cosine functions, some accurate analytical solutions were presented to blend the primary surfaces together in [Bloor & Wilson 1989b]. By mapping a unit cube in (u, v, w) parameter space to a hexahedral solid in physical space, the accurate analytical solutions to a second-order PDE with three parametric variables were obtained in [Bloor & Wilson 1993] to investigate the functionality in solids. The accurate closed form solutions for the boundary conditions represented with triangular functions are investigated in [Zhang & You 2002] for vase design.

2.1.3 Approximate analytical methods

Approximate analytical solutions cannot exactly satisfy both PDEs and boundary conditions. Many existing approximate analytical solutions first exactly satisfy boundary conditions and then minimize the error of PDEs to find an optimal surface. Fourier series-based approximate analytical solution was proposed in [Bloor & Wilson 1990] to generate free-form surfaces for the designs of hull of yacht-like boat, propeller blade, phone handset, and ship hull. The Fourier series-based approximate analytical solution was used to solve a fourth-order PDE for the interactive surface design in [Ugail et al. 1999] and parametric design of aircraft geometry in [Athanasopoulos et al. 2009]. It was also extended to obtain an approximate analytical solution to a sixth-order PDE for surface modelling in [Kubiesa et al. 2004]. A weighted residual method was presented to approximately solve a time-dependent fourth-order PDE for creation of deformable moving surfaces in [You & Zhang 2003], a time-independent fourth-order PDE in [You et al. 2004b] for C^1 continuous surface blending, and a time-independent sixth-order PDE for surface modelling in [Zhang & You 2004] and C^2 continuous surface blending in [You et al. 2004a].

To sum up, numerical methods are most effective in solving various PDEs but with discrete representations, a large amount of data, and high computational costs. Accurate analytical solutions are only applicable to

some simple surface modelling tasks. And existing approximate analytical solutions can deal with more complicated surface modelling than accurate analytical solutions, but most studies focus on 2-sided PDE patches. How to develop explicit representations of analytical 3- and 4-sided PDE patches has not been well investigated. In addition, there is no unified PDE-based modelling framework to deal with 2-, 3- and 4-sided PDE patches and satisfy arbitrary continuity requirements, which limits the application of PDE-based surface modelling on different occasions. The work described in this thesis will present a numerical solution for 4-sided PDE patches, apply it to solve practical engineering problems, and propose a unified PDE mathematical model of analytical 2-, 3- and 4-sided PDE patches with C^m continuity to make PDE-based surface modelling technique applicable to different occasions.

2.2 Surface manipulation

Surface manipulation, also known as surface or mesh editing, is a fundamental research topic in geometric modelling and computer-aided design. Depending on whether physics of the model deformation is introduced or not, surface manipulation can be roughly divided into two categories: purely geometric and physics-based [Zheng et al. 2003].

2.2.1 Purely geometric surface manipulation

There is rich literature on the topic of purely geometric surface manipulation. Here, some studies with mature practical applications will be briefly introduced.

One of the most representative purely geometric surface manipulation methods is free-form deformation (FFD), the idea of which was first introduced by [Barr 1987] and then further developed by [Sederberg & Parry 1986]. This method embeds an object in a lattice and achieves the deformations of the object by deforming the lattice. By using the initial lattice points to define an arbitrary trivariate Bézier volume, and allowing the combination of many lattices to form arbitrarily shaped spaces,

Coquillart [1990] introduced extended FFD. In order to provide a better control of the deformation and a more intuitive interface, Hsu et al. [1992] proposed a direct manipulation method of FFD. Based on FFD technique, several space deformation models such as rational FFD [Kalra et al. 1992], NURBS-based FFD [Lamousin & Waggenspack 1994], volume-preserving FFD [Hirota et al. 2000], T-spline FFD [Song & Yang 2005] and Proxy-driven FFD [Zhang et al. 2020] have been developed. Although FFD is very popular and supported in many 3D modelling software packages such as 3DS Max and Maya, it has several drawbacks. For example, in order to achieve a small and local deformation, the lattice grids need to be subdivided so that one lattice point can control the target deformation region without impact other regions. However, the subdivision will result in crowded or even messy lattice grids which not only block the view of the deformed shape but also cause inconvenience for interactive manipulation [Zhang et al. 2020]. In addition, the FFD method is difficult to achieve exact shape deformation because the deformed shape does not follow the lattice points exactly, so that it is unclear which lattice points should be moved and how transformation will affect the deformation of the model [Zheng et al. 2003].

The Laplacian coordinate is another successful surface deformation technique and a variant of [Sorkine et al. 2004]. It has been integrated into the software Blender. The potential of the Laplacian coordinate for local mesh morphing and deformation is introduced by [Alexa 2003]. By solving a linear least squares system, Lipman et al. [2004] reconstructed the surface from discrete Laplacians of the mesh functions and spatial boundary conditions. In order to make Laplacian coordinates invariant to rotation and isotropic scaling, Sorkine et al. [2004] proposed a Laplacian surface editing method which implicitly transforms the differential coordinates. Based on the idea of the Laplacian coordinate, Zhou et al. [2005] used the volumetric graph Laplacian to solve the problem of large deformations. Since the Laplacian method needs to define anchor vertices first and then move some of them to achieve the deformation of non-anchor vertices, it is not intuitive and convenient to deform complex shapes, e. g., multiple deformations within a surface.

Unlike FFD and Laplacian methods that deform the surface by adding extra lattice points or anchors, the Delta Mush method [Mancewicz et al. 2014] is to directly manipulate the polygonal meshes by moving mesh vertices. Its basic idea is to smooth the deformed shapes of the polygonal meshes. Delta Mush is also regarded as a surface deformation method and widely applied in 3D software packages such as Maya and Houdini.

To sum up, purely geometric surface manipulation methods do not consider any underlying physical laws so that the quality and aesthetics of deformed shapes mainly depend on the users' perception and skills.

2.2.2 Physics-based surface manipulation

Physics-based methods are to deform surface shapes by incorporating physical characteristics such as forces, torques and strain energies. They have been widely embraced by the computer graphics community [Nealen et al. 2006]. Terzopoulos et al. [1987] and Terzopoulos & Fleischer [1988a] introduced dynamic differential equations for flexible materials such as rubber, cloth and paper by employing elasticity theory. After that, they extended their work from elasticity to viscoelasticity, plasticity and fracture [Terzopoulos & Fleischer 1988b]. By minimizing an energy function with user controlled geometric constraints and loads, Celniker & Gossard [1991] developed a curve and surface finite element method for free-form shape design. Gdkbay & zg [1994] described a physically based modelling system based on a primal formulation and a hybrid formulation derived from elasticity theory. By using the theory of plate bending in elasticity, You et al. [2006] developed a mathematical model of physically based elastic deformations. Considering non-homogeneous material properties and conducting finite element simulations of deformable objects in local frames, McDonnell & Qin [2007] presented a modelling technique for physically based deformation.

Involving physics laws in traditional control point-based deformation methods is also a popular research topic. Terzopoulos & Qin [1994] developed a dynamic NURBS to deal with mass distributions, internal deformation energies, and other physical quantities of shape manipulation

of NURBS. After that, they further investigated the surfaces with symmetries and topological variability and developed a dynamic NURBS swung surface [Qin & Terzopoulos 1995]. By enhancing the power of triangular spline models and using Lagrangian mechanics, Qin & Terzopoulos [1997] developed the dynamic triangular NURBS and manipulated the surfaces defined over arbitrary, nonrectangular domains through the finite element solution of its mathematical model. Applying sculpting forces on a surface and formulating and minimizing the energy functional of the surface, Vassilev [1997] proposed a method to manipulate deformable B-spline surfaces.

Since physics-based methods are difficult to find analytical solutions, numerical methods such as the finite element method, are commonly used to resolve the mathematical models in the above studies. Therefore, physics-based methods usually require a high computational cost and long computing time, which do not meet the demand of interactive surface manipulation. In this thesis, an interactive PDE-based surface manipulation method will be developed with analytical solutions.

2.3 Surface representation conversion

In this thesis, one of the developed applications is optimal conversion from PDE surfaces to NURBS surfaces. Therefore, the review focuses on the main related work of NURBS surface representation conversion.

NURBS originates from B-spline technology and plays an important role in the CAD, CAM and CAE world. The description of NURBS was first given by [Versprille 1975], which extended B-splines to rational B-splines. By understanding the advantages of NURBS for geometry representation and design, Boeing proposed it as part of the standard to the 1981 International Graphics Exchange Standard meeting, and many companies, such as Structural Dynamics Research Corporation and Intergraph Corporation, started to develop modellers and systems based on NURBS in the 1980s [Piegl 1991]. Because of the useful geometric properties, NURBS become part of many national and international standards such as IGES [Kennicott 1996], PHIGS [ISO 1997] and STEP

[ISO 2003] for the representation, design, and data exchange of geometric information.

Converting other types of surfaces into NURBS surfaces is most often required since NURBS has become an industrial standard in CAD, CAM and CAE, especially in the field of reverse engineering. The key issue in the conversion is to solve NURBS fitting. In general, fitting is usually achieved with polynomial approximation, which involves the minimization of an error at discrete data points. Depending on the application domain and the expected type of error, different norms can be selected for the minimization process, such as l_1 , l_2 and l_3 norms [Heidrich et al. 1996]. The least-square (l_2 norm) is usually applied in NURBS fitting. Ma & Kruth [1998] presented an algorithm for NURBS curves and surfaces fitting from free-form objects based on least squares fitting. The basic idea is to identify weights of the control points by applying symmetric eigenvalue decomposition techniques and then establish control points in a similar way. Based on this identified method of weights, Saini & Kumar [2014] proposed a method to reconstruct surfaces from arbitrary perspective images using a NURBS model, and Kruth & Kerstens [1998] described NURBS surface fitting from a cloud of points subject to incorporation of sufficient boundary conditions. Through determining the knot vectors, selecting the degrees, calculating the weights, and constructing an initial NURBS surface, Yin [2004] provided a new algorithm for fitting NURBS surfaces to scattered points using minimization of deviation under boundary constraints. By letting ordered measured points be control points and using least squares minimization, Dan & Lancheng [2006] developed a new conversion method which modified a constructed surface to obtain a desired fitted surface.

NURBS consists of multi-parameters: control points, knots and weights, which make the conversion task become a multi-variable nonlinear optimization problem involving large amount of data. In order to find a good NURBS model from large amount of data, Ulker [2012] applied the heuristic of artificial immune system for global optimization to find a smooth curve and the optimization of NURBS weights and knot vectors. Jing et al. [2009] used a simulated annealing method to optimize weights

and knot parameters of NURBS for curve and surface fitting. Genetic algorithm (GA) is a common multi-variables optimization method. Li-maiem et al. [1996] applied GA to obtain control point and knot value optimization, and proposed a new method for curve and surface approximation from scanned data points. Similarly, Yoshimoto et al. [1999] and Sarfraz [2004] applied GA to optimize both knots and weights of control points for curves and surfaces.

To sum up, although there has been much work investigating conversion methods from different surface types to NURBS surfaces, no reports which convert PDE surfaces into NURBS surfaces were found. In addition, most prior conversion methods focus on a single surface patch or an object with a simple shape. Thus, the data sizes of origin surfaces and converted surfaces are small and there is no need to minimize the data sizes in the process of conversion and further applications such as aerodynamic simulation-based optimization. However, for a complicated and large-scale object such as high-speed train heads investigated in this thesis, it involves a large amount of data. This thesis will present an application of optimal conversion of PDE surfaces representing high-speed train heads into NURBS surfaces with required accuracy and small data amount by using numerical 4-sided PDE surface patches.

2.4 Aerodynamic optimization design of the high-speed train

Aerodynamic optimization design is to find a set of parameters which can construct the optimal shape of the parametric model based on the required aerodynamic characteristics. It mainly includes parametric modelling and aerodynamic analysis processes.

2.4.1 Parametric modelling

Parametric modeling methods of describing the head shape of high-speed trains can be roughly grouped into two categories, i.e., the framework modeling and shape deformation methods. The framework modeling

method is to directly construct the whole framework of high-speed train heads and obtain the surface model by filling surface patches into the framework [Suzuki & Nakade 2013; Yao et al. 2016]. However, this method is not ideal in accurately describing train heads because the local shape in surface patches is uncontrollable and has no ability to deform if the framework remains unchanged. Moreover, the tangential continuity at the joint of two adjacent surface patches cannot be guaranteed because the two adjacent patches are only constrained by the position coordinates of the framework. In order to optimize local shapes instead of the whole shape of a train head and deform the surface smoothly, various shape deformation methods have attracted considerable attention such as the free-form deformation [Li et al. 2016; Zhang et al. 2018] and the arbitrary shape deformation [Sun et al. 2010; Yao et al. 2014]. The shape deformation method can focus on improving specified local shapes, but its global shape is non-optimized. Besides, these shape deformation methods usually use one parameter to control the shape change of a local region which limits the diversity of the deformation shape.

2.4.2 Aerodynamic analysis

In early stages, researchers usually focused on studying the aerodynamic performance of a given high-speed train using experimental methods such as the wind tunnel test and moving model test [Schetz 2001]. Maeda et al. [1989] presented a method to estimate the aerodynamic drag of trains and evaluated the accuracy of the results by measuring the total resistances in the open air and in a tunnel. De Wolf & Demmenie [1997] developed a train tunnel test facility which can launch models up to 500 km/h to measure the interacting pressure waves and their reduction in tunnels for high-speed trains. Cheli et al. [2010] presented an aerodynamic analysis on two different versions of the high-speed train EMUV250 to study the cross wind behavior by a combined use of wind tunnel investigations and numerical CFD simulations. However, the experiment method, greatly depending on engineering experience of researchers to seek the optimal solution, is limited by some disadvantages such as expensive testbed and high cost of time.

With the development of computer technology and increase of the computational power, numerical simulations have been widely applied in the optimization design of high-speed train heads and usually combined with surrogate models to accelerate optimization process. Kwon et al. [2001] used the response surface methodology and the axisymmetric compressible Euler equations to optimize the nose shape and introduced the Hicks-Henne shape functions to define the design space. Ku et al. [2010] employed the vehicle modeling function for multi-objective optimization of the high-speed train nose and performed a multi-step design optimization using the BFGS algorithm with a response surface model. Lee & Kim [2008] adopted the Hicks-Henne shape functions to parameterize the high-speed train nose and presented an approximate optimization method to reduce the micro-pressure wave by using an SVM-based metamodel and sequential quadratic programming. Sun et al. [2010] proposed an optimization approach to improve the aerodynamic drag of a CRH3 high-speed train head by combining CFD analysis with the genetic algorithm and introduced an arbitrary shape deformation technology. These methods, however, focus on studying single-objective optimization problems which are inadequate to find optimal solutions.

The multi-objective optimization of high-speed train heads has been much investigated in recent decade. The task of multi-objective optimization is to find a set of solutions, i. e., Pareto-optimal solutions, which represent a trade-off among objective functions [Deb 1999]. Suzuki & Nakade [2013] developed a multi-objective optimization method of high-speed train heads using an evolutionary algorithm to estimate the aerodynamic drag and pressure variation, and parameterized the train head shape by B-spline curves and Coons patches. Li et al. [2016] optimized the aerodynamic drag and lift forces of a CRH2 high-speed train head using NSGA-II based on the Kriging model and applied five design variables to control the local shapes of the train head with a free-form deformation method. By using the similar modeling and analysis methods in [Li et al. 2016], Zhang et al. [2018] studied the aerodynamic drag, lift and side forces of a high-speed train running on an embankment under crosswinds. Muñoz-Paniagua et al. [2014] defined the geometrical parame-

terization of the nose shape of a high-speed train by three design variables, and adopted the genetic algorithm with a radial basis function network to minimize the maximum pressure gradient and aerodynamic drag of the high-speed train entering a tunnel. Yao et al. [2014] constructed a multi-objective optimization process of a high-speed train head using the modeling method of arbitrary shape deformation and the optimization method of NSGA-II based on a Kriging model. They also studied a multi-objective particle swarm algorithm with a SVM regression model for the aerodynamic optimization of high-speed train heads [Yao et al. 2016].

To sum up, the existing optimization methods of high-speed train heads perform unsatisfactorily in optimizing both global and local shapes. Especially, the parametric modelling algorithms used in the existing optimization methods cannot accurately and completely describe the global shape and local details of high-speed train heads. In order to address this problem, a multi-objective aerodynamic optimization design method of high-speed train heads is developed by using the numerical solution of 4-sided PDE surface patches.

2.5 Surface reconstruction

As an application of the proposed unified PDE-based modelling framework in Chapter 4, a surface reconstruct method from 3D data points using 2-sided PDE surface patches is developed. The focus is on the surface reconstruction under the constraint of the feature curves in automotive style design. The automotive surface reconstruction methods mainly include parametric surfaces and CAD-generated surfaces, which will be reviewed below.

2.5.1 Parametric surfaces

Surface reconstruction from 3D data points has been investigated intensively. Parametric surfaces [Piegl & Tiller 2012], such as Bézier, B-splines and NURBS surfaces, are widely applied in CAD systems. Gálvez et al.

[2008] proposed a particle swarm optimization method to reconstruct a Bézier surface from a set of 3D data. With the similar scheme, they further proposed an evolutionary-based NURBS surface reconstruction method [Gálvez & Iglesias 2012] and a GA-based B-spline surface reconstruction method [Gálvez et al. 2012] from clouds of 3D data points. Ma & Kruth [1995] presented a parameterization method to randomly distribute points for performing least squares fitting of B-spline curves and surfaces. He & Qin [2004] reconstructed a triangular B-spline surface with the user-specified n degree from a set of scanned 3D points. Park et al. [1999] proposed a NURBS surface fitting technique from scattered and unorganized range data using hierarchical graph representation. These parametric surfaces, however, are free-form and their boundary curves cannot fully represent feature curves in the automotive styling design.

2.5.2 CAD-generated surfaces

The CAD-generated surfaces [Chang 2014], such as loft and sweep surfaces, are popularly used in surface reconstruction. Lin et al. [1997] proposed a surface lofting method for the reverse engineering of complex shapes from the measurement data. Ueng et al. [1998] developed a sweep-surface reconstruction method from 3D measured data by using nonlinear least-squares minimization. Since the sweep surface shape depends on the profile curve and path curve, the sweeping method is similar to the practical claying process which is applied in automotive styling design [Hosaka 2012]. Tsuchie [2017] proposed a sweep-based method for reconstructing an underlying surface from scanned data of styling design objects with less torsion on the line of curvature. With the same scheme, the intersecting underlying surfaces with C^0 continuity were further developed by applying sweep-based method [Tsuchie 2019]. In automotive styling design, the feature curves are used as profile curves to reconstruct sweep or loft surfaces, which preserve feature curves and guarantee aesthetics. These methods, however, are essentially control point-based, which involve excessive control points, long computing time in the reconstruction process, and low order of continuity between adjacent surfaces.

To sum up, existing surface reconstruction methods have several limitations in automotive styling design, such as non-preservation of feature curves, long computing time, excessive control points and poor continuity between adjacent surfaces. To tackle this problem, the developed analytical solution of the unified framework of PDE-based modelling will be adopted to achieve C^n continuous surface reconstruction with high efficiency and accuracy in automotive styling design.

Chapter 3

Numerical PDE surface patches with engineering applications

Numerical methods are effective in generating PDE surface patches with complex boundary conditions [Brown et al. 1998; Du & Qin 2005a]. Compared with an analytical 2-sided surface patch defined by two boundary curves, a numerical 4-sided PDE surface patch is more powerful in describing the details of surfaces. Moreover, by using the boundary conditions and the control parameters in PDE, the shape of a numerical 4-sided PDE surface patch is easier to be modified. With these advantages, the PDE-based modelling can be adopted in engineering applications such as shape optimization of high-speed train heads [Wang et al. 2018]. The reasons why they are rarely used in engineering applications mainly include non-industrial standards in CAD, CAM and CAE systems and the difficulty of controlling the boundaries to modify the shape.

In this chapter, a numerical solution to a fourth-order PDE with three control parameters will be presented. Specifically, a finite difference method will be applied to solve 4-sided boundary conditions of a PDE surface patch in Section 3.1. Based on the obtained solution, a new method will be developed to achieve optimal conversion of PDE surfaces representing high-speed train heads into NURBS surfaces in Section

3.2 and a novel multi-objective aerodynamic optimization method of high-speed train heads will be proposed in Section 3.3, which is the first pipeline of using the PDE-based approach to optimize shapes in CFD simulation.

3.1 Numerical solution of PDE

The mathematical model of PDE-based parametric modelling can take several forms due to different orders, i.e., the second, fourth and sixth orders [Castro et al. 2008]. Since a second-order PDE cannot guarantee the tangent continuity between two adjacent PDE surface patches, and a sixth-order PDE is difficult to be solved and has low calculation efficiency [Zhang & You 2004], a fourth-order PDE is selected as the mathematical model of surface patches of the train head which well balances performance and efficiency. The vector-valued fourth-order PDE is defined as

$$(a_1 \frac{\partial^4}{\partial u^4} + a_2 \frac{\partial^4}{\partial u^2 \partial v^2} + a_3 \frac{\partial^4}{\partial v^4}) \mathbf{f}(u, v) = 0 \quad (3.1)$$

where $\mathbf{f}(u, v) = [x(u, v), y(u, v), z(u, v)]^T$ is a vector-valued position function which represents the generated parametric surface, a_1 , a_2 and a_3 are three shape control parameters, and u and v are the parametric variables of describing the parametric location of a point (x, y, z) on a PDE surface patch which are defined by $u \in [0, 1]$ and $v \in [0, 1]$.

The methods of solving a PDE can be analytical or numerical. Analytical methods represent a PDE surface patch by a vector-valued continuous function which can be calculated rapidly and precisely. However, analytical methods are usually applicable to a low-order or simple PDE and difficult to solve the PDE for the surface patches defined by four boundaries. Compared with analytical methods, numerical methods are more powerful in solving a high-order PDE and can deal with various complicated surface modelling problems although numerical methods are computationally more complicated and expensive. Therefore, numerical methods are more suitable in solving the PDE shown in Eq. (3.1). One popular numerical method is the finite difference method due to its

simplicity, intuitiveness, and high efficiency in solving a vector-valued fourth-order PDE to create surface patches with four boundaries.

The finite difference method is to discretize the parametric region of a surface into $I \times J$ regularly and uniformly distributed nodes as shown in Fig. 3.1. The small dots, squares and triangles represent the unknown inner nodes, the known boundary nodes and the virtual nodes beyond the boundaries of a PDE surface patch, respectively. The virtual nodes will be involved in the finite difference equations for the inner nodes next to boundary curves and will be determined with the first partial derivatives on the boundaries, which are used to guarantee the boundary tangent continuity.

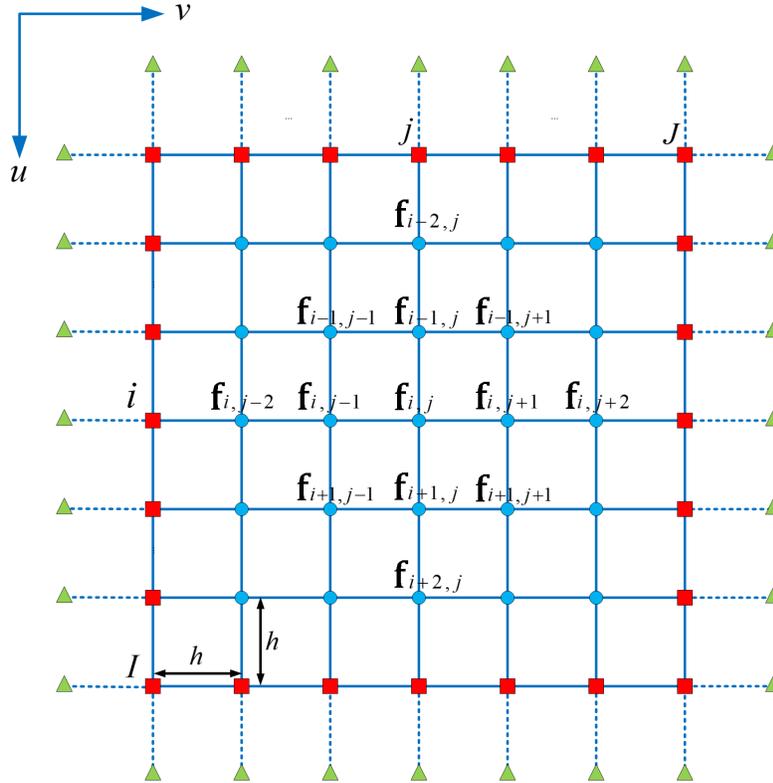


Figure 3.1: *The $I \times J$ finite difference grid*

In Fig. 3.1, $\mathbf{f}_{i,j} = \mathbf{f}(u_i, v_j)$ ($i = 2, 3, \dots, I - 1$, $j = 2, 3, \dots, J - 1$) represents an arbitrary inner node (i, j) on the finite difference grid. Based on the Taylor series expansion of the function $\mathbf{f}(u, v)$, the central

difference approximation of $\frac{\partial \mathbf{f}_{i,j}}{\partial u}$ and $\frac{\partial \mathbf{f}_{i,j}}{\partial v}$ can be derived as follow

$$\frac{\partial \mathbf{f}_{i,j}}{\partial u} = \frac{1}{2h}(\mathbf{f}_{i+1,j} - \mathbf{f}_{i-1,j}) \quad (3.2)$$

$$\frac{\partial \mathbf{f}_{i,j}}{\partial v} = \frac{1}{2h}(\mathbf{f}_{i,j+1} - \mathbf{f}_{i,j-1}) \quad (3.3)$$

where h denotes the grid interval.

The central difference approximations of the fourth partial derivatives derived from Eqs. (3.2) and (3.3) can be expressed as:

$$\frac{\partial^4 \mathbf{f}_{i,j}}{\partial u^4} = \frac{1}{h^4}[6\mathbf{f}_{i,j} - 4(\mathbf{f}_{i-1,j} + \mathbf{f}_{i+1,j}) + (\mathbf{f}_{i-2,j} + \mathbf{f}_{i+2,j})] \quad (3.4)$$

$$\frac{\partial^4 \mathbf{f}_{i,j}}{\partial v^4} = \frac{1}{h^4}[6\mathbf{f}_{i,j} - 4(\mathbf{f}_{i,j-1} + \mathbf{f}_{i,j+1}) + (\mathbf{f}_{i,j-2} + \mathbf{f}_{i,j+2})] \quad (3.5)$$

$$\begin{aligned} \frac{\partial^4 \mathbf{f}_{i,j}}{\partial u^2 \partial v^2} = & \frac{1}{h^4}[4\mathbf{f}_{i,j} - 2(\mathbf{f}_{i-1,j} + \mathbf{f}_{i+1,j} + \mathbf{f}_{i,j-1} + \mathbf{f}_{i,j+1}) \\ & + (\mathbf{f}_{i-1,j+1} + \mathbf{f}_{i-1,j-1} + \mathbf{f}_{i+1,j+1} + \mathbf{f}_{i+1,j-1})] \end{aligned} \quad (3.6)$$

After substituting Eqs. (3.4), (3.5) and (3.6) into (3.1), the following linear algebra equation at the inner node (i, j) is obtained.

$$\begin{aligned} & (6a_1 + 4a_2 + 6a_3)\mathbf{f}_{i,j} - (4a_3 + 2a_2)\mathbf{f}_{i,j+1} - (2a_2 + 4a_1)\mathbf{f}_{i+1,j} \\ & - (4a_3 + 2a_2)\mathbf{f}_{i,j-1} - (2a_2 + 4a_1)\mathbf{f}_{i-1,j} + a_2\mathbf{f}_{i-1,j+1} + a_2\mathbf{f}_{i+1,j+1} \\ & + a_2\mathbf{f}_{i+1,j-1} + a_2\mathbf{f}_{i-1,j-1} + a_3\mathbf{f}_{i,j+2} + a_1\mathbf{f}_{i+2,j} + a_3\mathbf{f}_{i,j-2} + a_1\mathbf{f}_{i-2,j} = 0 \end{aligned} \quad (3.7)$$

According to the central difference approximations defined by Eqs. (3.4), (3.5) and (3.6), the finite difference equations at the inner nodes next to the four boundaries of the grid involve the boundary nodes and the virtual nodes. All the boundary nodes are known. The virtual nodes can be determined from the known boundary tangents, i. e., the first partial derivatives of the function $\mathbf{f}(u, v)$ with respect to the parametric variable u or v on the four boundaries of the PDE surface patch. In Fig. 3.1, assuming $i = 2$ and the tangent at the boundary node $\mathbf{f}_{1,j}$ is $\mathbf{T}_{1,j}$, the virtual node $\mathbf{f}_{0,j}$ next to the boundary node $\mathbf{f}_{1,j}$ can be represented by

$\mathbf{f}_{2,j} - \mathbf{T}_{1,j}$. Therefore, the errors brought in by the virtual nodes depend on directions and sizes of given tangents. After merging these boundary conditions into the finite difference equations, Eq. (3.7) can be written in a following matrix form

$$[A] \{Q\} = \{E\} \quad (3.8)$$

where $[A]$ is an $I \times J$ by $I \times J$ square and nonsingular coefficient matrix. $\{Q\}$ is a column vector of the discrete points of the PDE surface patch. $\{E\}$ is a known column vector involving boundary nodes and boundary tangents. When $i = 2$, $i = I - 1$, $j = 2$ or $j = J - 1$, the corresponding element in $\{E\}$ consists of a boundary node and a tangent. When $i = 3$, $i = I - 2$, $j = 3$ or $j = J - 2$, the corresponding element in $\{E\}$ is a boundary node. When $i = 4, 5, \dots, I - 3$ or $j = 4, 5, \dots, J - 3$, the corresponding element in $\{E\}$ is 0.

The PDE surface patch can be obtained directly by matrix inversion.

$$\{Q\} = [A]^{-1} \{E\} \quad (3.9)$$

3.2 PDE-based surface conversion

PDE-based surface modelling has not become an industrial standard for the applications in CAD, CAM and CAE. In contrast, NURBS is an industry standard for the representation, design, and data exchange of geometric information processed by computers [Piegl & Tiller 2012]. Automatically converting PDE-based 3D models into NURBS representations will greatly promote engineering applications of PDE-based modelling. However, for a complicated and large-scale object such as high-speed train heads, it involves a large amount of data which causes two main problems. First, the conversion process will cost a lot of computational time and raise the computational cost. Second, after achieving the conversion, the NURBS representation involves plenty of control points, which is difficult to achieve subsequent engineering applications such as shape redesign and aerodynamic simulation. In order to tackle the two problems, the

conversion method should not only achieve good conversion accuracy, but also minimize the amount of the data representing the converted NURBS surfaces.

In this section, 4-sided PDE surface patches obtained with the above finite difference method are used to represent a high-speed train head model. The continuity between adjacent PDE surface patches is C^1 continuity. Based on the 4-sided PDE surface patches, a new method is developed to achieve optimal conversion of PDE surfaces representing high-speed train heads into NURBS surfaces.

3.2.1 Overview for PDE-based surface conversion

For complicated and large-scale objects such as high-speed train heads to be considered in this work, plenty of control points are required to describe its shape. This will introduce many design variables and increase the need of storage capacity. If the weights of control points are also involved in the optimization calculations, independent design variables will be significantly increased. Too many control points and weights will lead to a large search space, greatly reduce computational efficiency, increase the difficulty in finding optimal converted NURBS surfaces. In the following applications of aerodynamic simulation-based optimization, they will also cause heavy calculations and make the optimal shape hard to obtain. Therefore, the aim of the optimal conversion from PDE surface-represented train heads to NURBS surface-represented train heads should look for the minimum design variables and weights while satisfying the required conversion accuracy ε . Two new ideas are introduced to achieve this aim. First, a lot of weights are approximated with a weight deformation discussed in Subsection 3.2.4 to noticeably reduce the number of weights. Second, the number of control points is minimized to decrease total control points while still satisfying the required conversion accuracy ε .

Including the weight deformation and the number and positions of control points in the same optimization objective function will greatly increase the computational complexity of the optimal conversion. When

the total number of control points is known, the errors between PDE surfaces and NURBS surfaces can be minimized with the least square method to obtain the optimal positions of control points. Therefore, the complicated optimal conversion problem can be converted into two simple interlinking sub-problems: 1) obtaining the minimum number of control points and optimal weight deformation, and 2) determining the optimal positions of control points. The genetic algorithm (GA) is commonly used to generate high-quality solutions to optimization and search problems. In this section, GA is employed to determine the minimum number of control points and the optimal weight deformation and combined with the least squares method to obtain the optimal conversion from PDE surface-represented train heads to NURBS surface-represented train heads.

As shown in Fig. 3.2, the proposed method can be divided into three steps: 1) PDE surface-based train head modelling, 2) NURBS surface formulation, and 3) GA-based optimal conversion. In 1) PDE surface-based train head modelling, a complicated train head (Fig. 3.2: the left image of the top row) is first decomposed into a number of simple parts (the second image from the left in top row), each part is described with a PDE surface patch (the third image from the left in top row) obtained from the finite difference solution of a vector-valued PDE (3.1), and all the PDE surface patches are automatically and smoothly stitched together to represent the whole train head model (the right image in top row). In 2) NURBS surface formulation, the discrete vertices of each PDE surface patch are extracted, the number of control points and weight deformation obtained from the GA are input to define a NURBS surface with unknown control points, and the least square method is introduced to minimize the errors between the PDE surface and NURBS surface and determine the optimal positions of control points. In 3) GA-based optimal conversion, the maximum error between the PDE surface and the corresponding NURBS surface obtained from the least squares method is first calculated. If the maximum error satisfies the required conversion accuracy ε , i. e., $\text{maximum error} \leq \varepsilon$, the optimal NURBS-represented train head is obtained for further applications in CAD, CAM and CAE

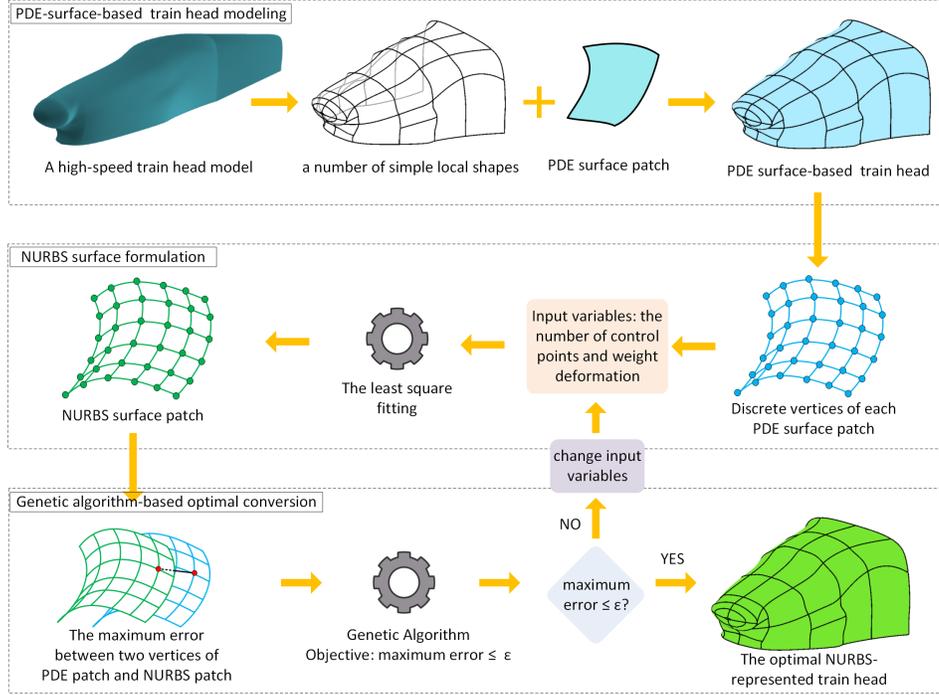


Figure 3.2: *Framework of the proposed method*

and the optimization calculations terminate. If the maximum error is larger than ε , the GA will find a new number of control points and weight deformation and replace the previous input variables with them in the second step. The new variables are input into the least squares minimization to determine the new positions of control points.

3.2.2 NURBS method

Non-uniform rational B-splines (NURBS) are commonly supported by CAD, CAM, and CAE systems and have already become part of numerous industry wide standards. A NURBS surface of degree p in the u direction and degree q in the v direction can be expressed as [Piegl & Tiller 2012]:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u, v) P_{i,j} \quad (0 \leq u \leq 1, 0 \leq v \leq 1) \quad (3.10)$$

where $P_{i,j}$ are $(m+1) \times (n+1)$ control points of the NURBS surface, and $R_{i,j}(u, v)$ are piecewise rational basis functions, which are defined as:

$$R_{i,j}(u, v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{\bar{i}=0}^n \sum_{\bar{j}=0}^m N_{\bar{i},p}(u)N_{\bar{j},q}(v)w_{\bar{i},\bar{j}}} \quad (3.11)$$

In the above equation, $w_{i,j}$ and $w_{\bar{i},\bar{j}}$ are the weights, and $N_{i,p}(u)$, $N_{j,q}(v)$, $N_{\bar{i},p}(u)$ and $N_{\bar{j},q}(v)$ are the non-rational B-spline functions defined on the knot vector

$$U = \{\underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_{p+1}\} \quad (3.12)$$

$$V = \{\underbrace{0, \dots, 0}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{1, \dots, 1}_{q+1}\} \quad (3.13)$$

where $r = n + p + 1$ and $s = m + q + 1$.

Taking the parametric variables u and v to be the same values as those at the finite difference nodes, i. e., $u = u_k$ and $v = v_k$ with $k = (i-1)J + j$ where i and j are shown in Fig. 3.1, R_{kl} ($k = 1, 2, \dots, K; l = 1, 2, \dots, L$) is used to represent $R_{i,j}(u_k, v_k)$ with $K = I \times J$ to be the total number of the finite difference nodes and L to be the total number of control points, i. e., $L = (m+1) \times (n+1)$. After replacing the control point $P_{i,j}$ ($i = 0, 1, 2, \dots, n; j = 0, 1, 2, \dots, m$) in Eq. (3.10) with the symbol P_l , Eq. (3.10) can be written in a matrix form as follows:

$$\begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_K \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1L} \\ R_{21} & R_{22} & \cdots & R_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ R_{K1} & R_{K2} & \cdots & R_{KL} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_L \end{bmatrix} \quad (3.14)$$

or

$$\{S\} = [R]\{P\} \quad (3.15)$$

3.2.3 Least squares fitting

The least squares method is a standard approach in regression analysis which is used to find the best fit to a dataset. It is widely applied in reverse engineering. With the least squares method, the sum of the squared errors between a PDE surface patch and its corresponding NURBS surface patch at the finite difference nodes can be written as

$$f = \sum_{k=1}^K |Q_k - S_k|^2 \quad (3.16)$$

where Q_k and S_k are k^{th} element of the column vector $\{Q\}$ and $\{S\}$ which represents the k^{th} discrete vertex of a PDE surface patch and a NURBS surface patch, respectively.

Introducing Eq. (3.9) and Eq. (3.15) into the above equation, Eq. (3.16) is transformed into the following form:

$$f = \sum_{k=1}^K |[A]_k^{-1}\{E\} - [R]_k\{P\}|^2 \quad (3.17)$$

where $[A]_k^{-1}$ is the k^{th} row of the matrix $[A]^{-1}$, $\{E\}$ is the column vector given in Eq. (3.9), $[R]_k$ is the k^{th} row of the matrix $[R]$, and $\{P\}$ is the column vector given in Eq. (3.15).

According to the least squares method, $\frac{\partial f}{\partial P_l} = 0$ ($l = 1, 2, \dots, L$), which converts Eq. (3.17) into the following normal equation.

$$[R]^T[A]^{-1}\{E\} = [R]^T[R]\{P\} \quad (3.18)$$

Solving the above equation, the positions of all the unknown control points of the NURBS surface patch are obtained.

3.2.4 Genetic algorithm

The accuracy of the converted NURBS surface depends on the number of control points: more control points will lead to more accurate calculations which makes the converted NURBS surface closer to the PDE surface.

Therefore, it is required to find enough control points, which make the converted NURBS surface satisfy the required conversion accuracy. Since increasing control points will introduce more design variables, cause more storage capacity, and raise the computational cost in both conversion process and following applications, it is important to find the least control points which not only minimize the number of design variables but also satisfy the required conversion accuracy ε .

In addition to control points, weights of a NURBS surface also affect the conversion accuracy. As shown in Eq. (3.10) and Eq. (3.11), one control point has one weight. If all the weights are taken to be design variables, the optimization calculation cost will greatly increase and the optimal conversion will be more difficult to obtain. To tackle this problem, a curve deformation algorithm is proposed to change all the weights into a single weight deformation.

GA is very efficient in random search to solve unclear and complex problems. This thesis uses it to find the minimum number of control points and the optimal weight deformation and combines it with the least squares method to determine the most suitable positions of control points for the optimally converted NURBS surface. The basic structure of the GA is shown in Fig. 3.3.

Design variables

As discussed above, the number of control points and weight deformation are chosen to be design variables. Inspired by the algorithm of deforming a curve discussed in [Yu et al. 2013], a new weight deformation algorithm is proposed to deform a surface. It transforms multiple weight variables into a single weight deformation, makes the deformation bigger and bigger when moving from the boundaries to the centre of a NURBS surface, and has no effects on the four boundaries of a NURBS surface patch to ensure the continuity between adjacent NURBS surface patches. The deformation algorithm is given by

$$w_{i,j} = \bar{w}_{i,j} \left(1 + dw \frac{(i-1)(n-i)}{(i-1)^2 + (n-i)^2} \times \frac{(j-1)(m-j)}{(j-1)^2 + (m-j)^2} \right) \quad (3.19)$$

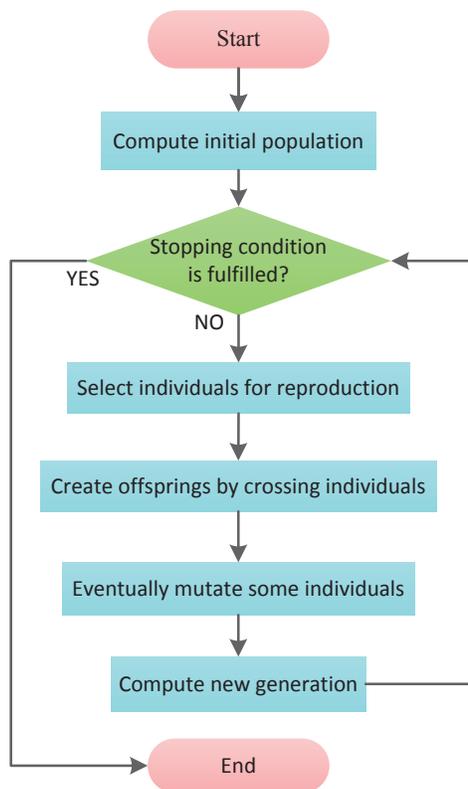


Figure 3.3: *The basic structure of the genetic algorithm*

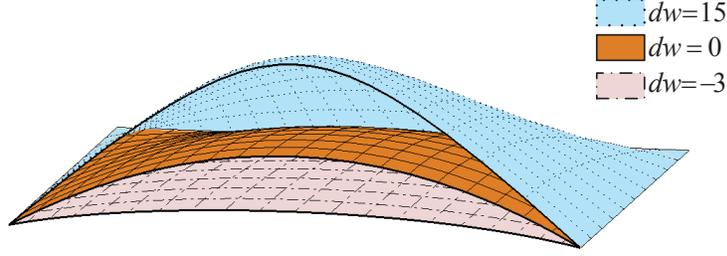


Figure 3.4: *Effects of the weight deformation on a NURBS surface*

where $\bar{w}_{i,j}$ and $w_{i,j}$ are original and new weights, respectively, $n + 1$ and $m + 1$ are the number of weights in the u direction and v direction respectively, and dw is the weight deformation.

Figure 3.4 shows an example how the weight deformation changes the shape of a NURBS surface. When dw is 0, the weights of all control points are equal to 1 ($w_{i,j} = 1$). It means the NURBS surface becomes a B-spline surface. Therefore, the geometric meaning of weight deformation is to adjust the convexity of a B-spline surface.

The least number of control points of a NURBS surface is related to the degree p in the u direction and degree q in the v direction. For a NURBS surface of the degree p with $n + 1$ control points in the u direction, its knot vector has $n + p + 2$ knots. Since the knot vector is non-periodic and the first and last knots have multiplicity $p + 1$, the number of control points must satisfy $n + 1 \geq p + 1$. Similarly, in the v direction, the number of control points must satisfy $m + 1 \geq q + 1$. Thus, for a NURBS surface, the least number of control points is $(p + 1) \times (q + 1)$. In addition, the maximum number of control points is equal to K , which is the total number of discrete vertices of NURBS surface. Therefore, the input range of the number of control points is between $(p + 1) \times (q + 1)$ and K .

There is no strict limitation for the range of weight deformation, except that it needs to include the value of zero. A wider range of weight deformation produces more precise results but causes more computing time. On the other side, when the value of weight deformation is very large or small, the shape of surface is nearly unchanged. Therefore, after a dozen experiments, a suitable range of weight deformation is found to

be $[-4, 30]$.

After the ranges of the number of control points and weight deformation have been specified, they are represented by a set of strings coded in binary.

Objective function

Since Eq. (3.9) determines a PDE surface patch represented by the column vector $\{Q\}$ and Eq. (3.15) describes the corresponding NURBS surface patch represented by the column vector $\{S\}$, the difference between the two surface patches is obtained by calculating the errors of the two surface patches at the finite difference nodes i ($i = 1, 2, 3, \dots, K$) which can be defined by the Euclidean distances $d(Q_i, S_i)$ where Q_i is the i^{th} element of the column vector $\{Q\}$ and S_i is the i^{th} element of the column vector $\{S\}$. The optimization objective function is to find the minimum number of control points and the optimal weight deformation, which minimizes the maximum error between the PDE surface patch and its corresponding NURBS surface patch. Once the maximum error satisfies the required conversion accuracy ε , the GA terminates and the optimal NURBS surface patch is obtained. Therefore, the optimization objective function can be formulated as

$$\begin{aligned} & \arg \min_{cp, dw} |\varepsilon_{max}(cp, dw) - \varepsilon| \\ & \text{subject to } \varepsilon_{max}(cp, dw) \leq \varepsilon \end{aligned} \quad (3.20)$$

where

$$\varepsilon_{max}(cp, dw) = \max_{1 \leq i \leq K} \left\{ \frac{1}{L_s} d(Q_i, S_i) \right\} \quad (3.21)$$

where cp and dw are design variables that are the number of control points and the weight deformation, respectively, and L_s is the grid interval of the PDE surface. The reason why choosing the shortest distance is to measure the distance between the PDE surface patch and the NURBS surface patch under the same order of magnitude of the PDE surface.

Genetic operators

The basic structure of the GA is shown in Fig. 3.3. The transition from one generation to the next one consists of four basic components [Bodenhofer 2003]:

- 1) Selection: The selection process selects individuals for reproduction according to their fitness. In this work, the individuals are the binary strings of the number of control points and the weight deformation, and the fitness is the objective function value.
- 2) Crossover: Crossover is a probabilistic process that merges the genetic information of two selected individuals and produces next generation.
- 3) Mutation: Mutation is a random deformation of the strings with a certain probability. It mutates a bit position (genes) of binary representation of chromosomes by simply flipping its value at random. The positive effect is preservation of genetic diversity and avoidance of local maximum.
- 4) Sampling: The sampling process computes a new generation from the previous one and its offspring.

3.2.5 Applications

As a demonstration of the proposed method, an example of converting the PDE surface patches of a simplified high-speed train head into optimal NURBS surface patches is presented here. To obtain the NURBS-represented train head, the first step is to represent a high-speed train head with a number of PDE-based surface patches. The PDE surface patches are reconstructed according to the scanned data of an actual high-speed train head by ignoring bogies and other auxiliary structures. The reconstruction from the scanned data is not the research objective of this chapter. Hence the reconstruction process will not be discussed here. Since the high-speed train head is a symmetrical structure, there is no need to represent the shape of whole train head using PDE-based surface patches. Thus, the half of train head is created by PDE-based

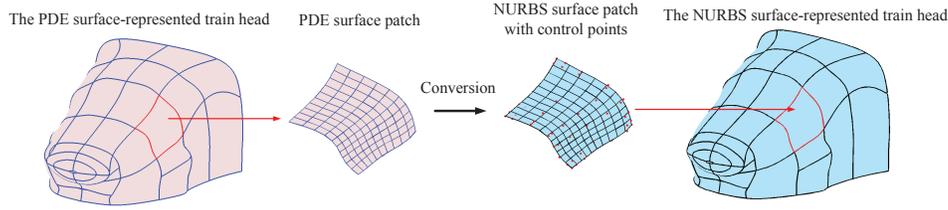


Figure 3.5: *Process of conversion*

surface patches, and then the whole shape of the PDE-based train head is obtained by symmetry. The patches of the half of the train head are then converted to optimal NURBS surface patches, which are assembled to represent a complete NURBS surface-represented train head. The conversion process is shown in Fig. 3.5.

For illustrative purposes, the half of a high-speed train head is divided into 22 surface patches according to shape changes. The whole train head includes 44 surface patches, which are shown in Fig. 3.5. The 22 surface patches are converted to optimal NURBS surface patches by the proposed method. The flowchart of the optimal conversion is shown in Fig. 3.6.

The flowchart includes three steps. First, the coordinate values and u and v values of the 22 PDE surface patches of the train head at the finite difference nodes together with a required conversion accuracy and GA variables (the number of control points and weight deformation) are input to the algorithm of least squares surface fitting to determine the optimal positions of the converted NURBS surface patches. Second, the GA is applied to minimize the objective function and obtain the new number of control points and weight deformation. Third, whether the maximum error is smaller than the required conversion accuracy is checked. If yes, the optimal NURBS surface patches are obtained and the GA terminates. Otherwise, the new number of control points and weight deformation are input to the algorithm of least squares surface fitting to determine new optimal positions of control points.

In this example, the required conversion accuracy ε is taken to be 1% ($\varepsilon = 1\%$). Note that the value of ε is user-defined and can be any other

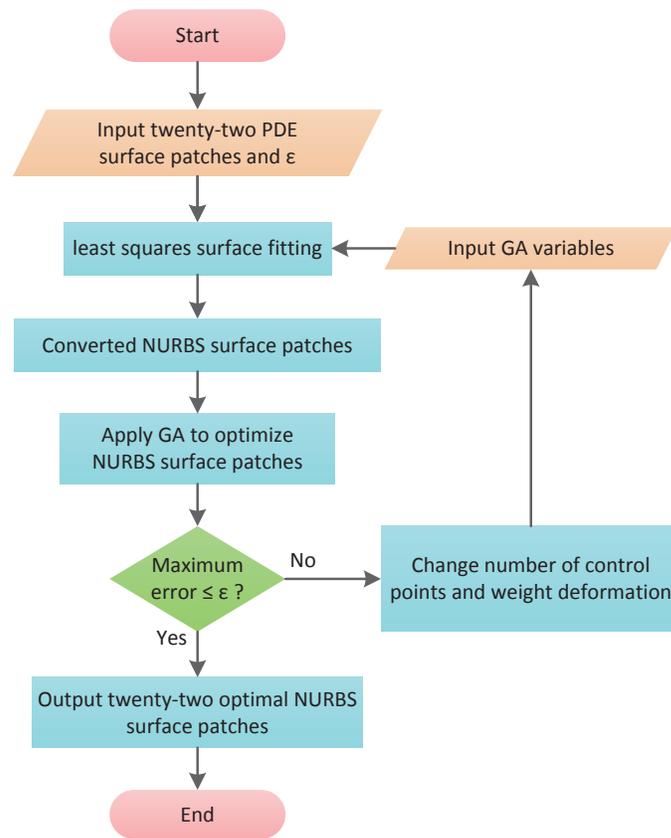


Figure 3.6: *Flowchart of the optimal conversion*

Table 3.1: *Optimal number of control points and weight deformation ($\varepsilon = 1\%$)*

Patch index	Number of control points	Weight deformation
1	121	-3.9969
2	121	0.7452
3	110	-3.9913
4	100	-3.2886
5	90	2.9158
6	72	1.7053
7	77	2.8173
8	48	0.7253
9	96	2.6475
10	99	3.8784
11	80	-3.2496
12	80	8.3377
13	48	2.8943
14	64	9.5815
15	56	0.3618
16	110	16.3041
17	70	-3.2187
18	90	30.0000
19	99	18.3193
20	120	3.2596
21	88	-3.2854
22	100	14.9994

value. It depends on user's acceptable maximum error. The optimal number of control points and weight deformation for the 22 NURBS surface patches of the high-speed train head are given in Table 3.1. In order to visualize the errors between the PDE surface patches and the obtained optimal NURBS surface patches, the errors are represented with different colors in Fig. 3.7. In the figure, the lighter colors mean smaller errors whereas the darker colors mean bigger errors.

Since ε is set to 1%, the minimized maximum errors should be between 0 and 1%, i. e., $0\% \leq \varepsilon_{max}(cp, dw) \leq 1\%$. Fig. 3.7 indicates that all the errors are in the range. Among them, the maximum errors of patches 4, 8, 11, 12, 17, and 21 reach the upper limit of the range which is 1%, and the maximum errors of patches 13 and 20 are 0.38% and 0.32%, respectively,

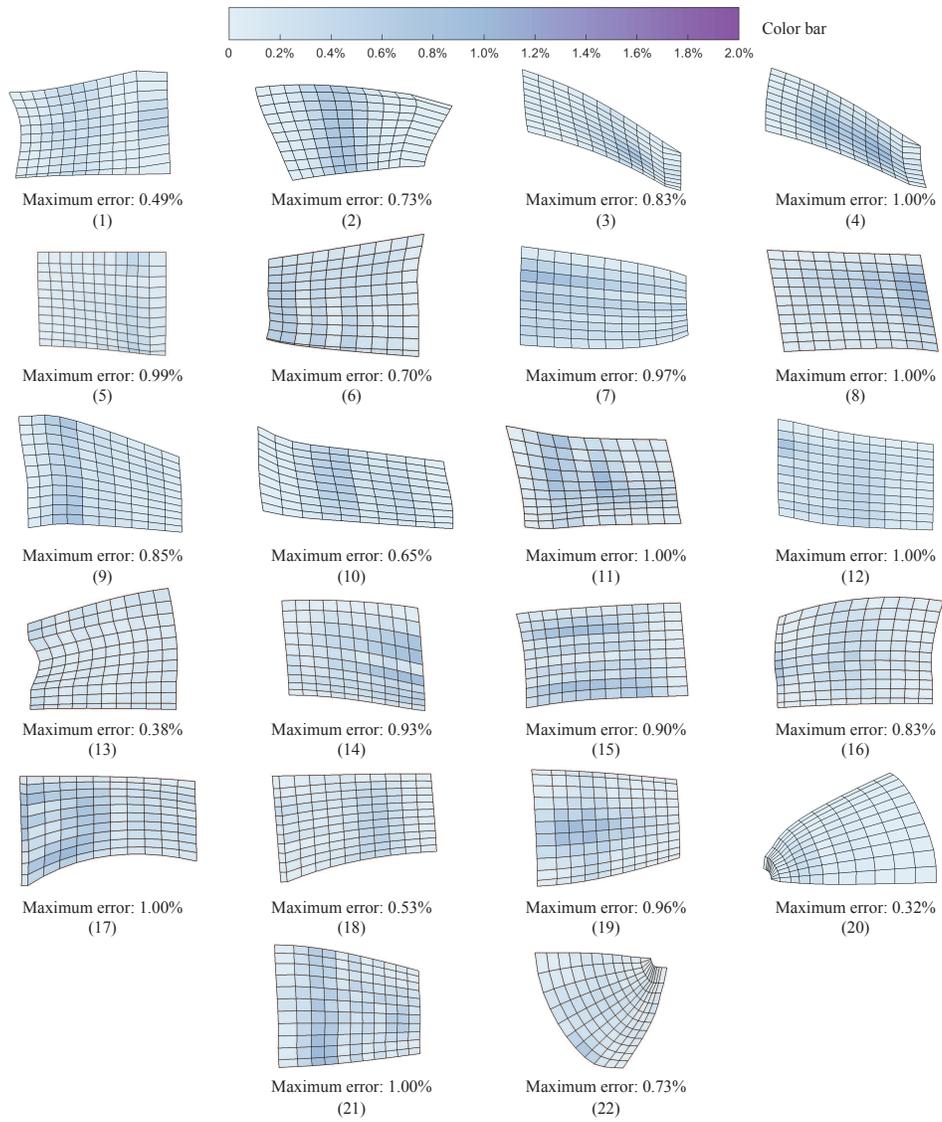
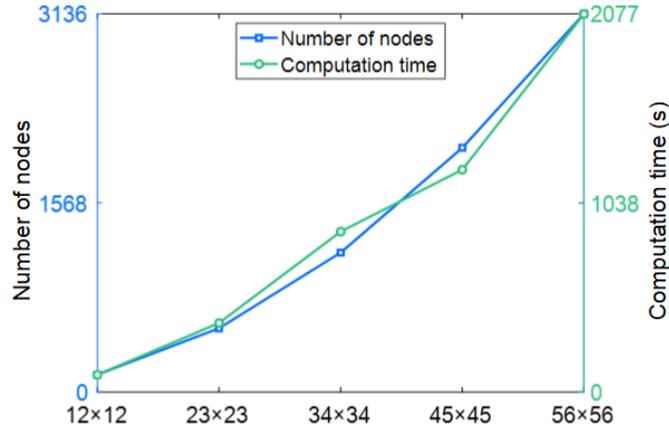


Figure 3.7: Visualization of the errors between the PDE surface patches and NURBS surface patches

Table 3.2: *Computation time of each surface patch*

Patch index	Time (s)	Patch index	Time (s)
1	154	2	176
3	151	4	118
5	135	6	101
7	65	8	90
9	101	10	85
11	73	12	80
13	59	14	60
15	64	16	112
17	78	18	95
19	102	20	117
21	104	22	106

**Figure 3.8:** *Computational complexity*

which are minimal.

The computation time of each surface patch is shown in Table 3.2. The maximum and minimum computation time are patch 1 and patch 13, respectively, and the average time is 101 seconds. In order to explore the computational complexity of the proposed method, patch 18 is used as an example to calculate the computation time of different numbers of nodes, i. e., 12×12 , 23×23 , 34×34 , 45×45 and 56×56 , and evaluate the relationship between the computation time and the number of nodes, as shown in Fig. 3.8. The result indicates the computation time is approximately linear with respect to the number of nodes.

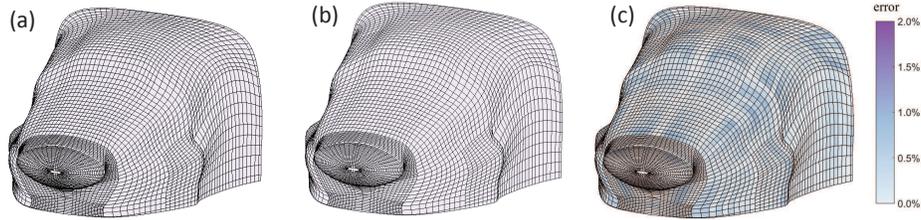


Figure 3.9: *Error comparison between the original PDE surface-represented (a) and NURBS surface-represented (b) high-speed train head where (c) uses different colors to visualize the errors between them*

For the whole train head, the errors between the PDE representation and NURBS representation are visualised in Fig. 3.9. In this figure, (a) is the original PDE surface-represented train head, (b) is optimally converted NURBS surface-represented train head, and (c) uses different colors to visualize the errors between the original PDE surface-represented train head and the optimally converted NURBS surface-represented train head. The images given in the figure clearly show the errors for all the surface patches are not more than 1%.

The above discussions indicate that the developed method is very effective in obtaining the optimal conversion from the PDE surface-represented train head to the NURBS surface-represented train head.

In order to investigate the effects of the required conversion accuracy ε and weight deformation, ε is set to 0%, 1%, 2%, 5% and 10%, respectively, and two cases are considered: one with fixed weights $w_{i,j} = 1$ [Brujic et al. 2002; Xiao & Li 2005], and another with the weight deformation. The obtained results are depicted in Fig. 3.10 where the blue curve is from the optimal conversion with the fixed weights, and the pink curve is from the optimal conversion with the weight deformation.

Figure 3.10 shows that the total number of control points for the 44 NURBS patches of the half of the train head linearly decreases with the increase of the required conversion accuracy for both cases. The linear decrease when the required conversion accuracy is less than 1% is larger than the linear decrease when the required conversion accuracy is more than 1%. When the required conversion accuracy ε is 0%, 1%, 2%, 5% and 10%, the minimum control points for the optimal conversion

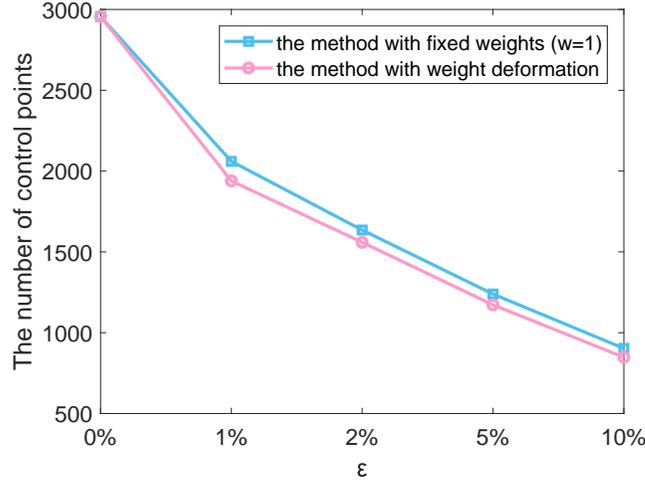


Figure 3.10: *Effects of the required conversion accuracy and weight deformation*

with fixed weights are 2956, 2060, 1635, 1239 and 903, respectively, and the minimum control points for the optimal conversion with weight deformation are 2956, 1939, 1559, 1172 and 848, respectively, indicating the application of the weight deformation can decrease the number of control points to some extent. Compared to the minimum control points obtained by the optimal conversion with the fixed weights, the optimal conversion with the optimal weight deformation reduces the minimum control points by 0%, 5.87%, 4.65%, 5.41% and 6.09%, respectively. When $\varepsilon = 0\%$, the errors at the finite difference nodes between the PDE surface patches and the corresponding NURBS surface patches are zero, and Eq. (3.17) becomes $[A]_k^{-1}\{E\} - [R]_k\{P\} = 0$ ($k = 1, 2, 3, \dots, K$). That is, the K equations are solved to determine K control points. Therefore, the number of control points is always K whether the fixed weights or the weight deformation are considered. When the required conversion accuracy increases from 0% to 1%, 2%, 5% and 10%, the total control points are reduced by 34.40%, 47.26%, 60.35% and 71.31% for the optimal conversion with the optimal weight deformation, which indicates setting the required conversion accuracy to a high value will greatly decrease the number of control points. Therefore, proper selection of the required conversion accuracy is very important for reducing the number of control points.

3.2.6 Summary

With the numerical solution of the 4-sided PDE surface patch, a novel method of converting a PDE surface-represented high-speed train head into optimal NURBS surfaces is developed, and a weight deformation method is proposed to transform many weights into a single weight deformation for reduction of design variables. The least squares fitting method and GA are combined to optimally convert PDE surface patches to NURBS surface patches, which satisfies the required conversion accuracy with minimum control points. The proposed method is demonstrated by presenting an example of converting PDE surfaces of a high-speed train head into NURBS surfaces. The example indicates that the developed method is able to obtain the optimal NURBS surface-represented high-speed train head with high accuracy and minimum control points.

The influences of the required conversion accuracy and weight deformation on the optimal conversion are also investigated. Since a high value of the required conversion accuracy can greatly reduce control points, it is very important to properly specify the required conversion accuracy to minimize control points. By comparing the optimal conversion with fixed weights and that with the optimal weight deformation, it can be concluded that introduction of the weight deformation can further lower the number of control points.

3.3 PDE-based multi-objective aerodynamic optimization

With the increasing running speed, the aerodynamic issues of high-speed trains are being raised and impact the running stability and energy efficiency. The optimization design of the head shape is significantly important in improving the aerodynamic performance of high-speed trains. Existing aerodynamic optimization methods are limited by the parametric surface modelling methods of train heads which are unable to accurately and completely parameterize both global shapes and local details [Li et al. 2016; Zhang et al. 2018; Muñoz-Paniagua et al. 2014;

Yao et al. 2016, 2014]. Due to this reason, they cannot optimize both global and local shapes of train heads.

In this work, the developed numerical 4-sided PDE surface patches are used to develop a novel multi-objective aerodynamic optimization method of high-speed train heads. With this method, the half of a train head is parameterized with 17 PDE surface patches, which describe global shapes and local details and keep the surface smooth. The aerodynamic drag and lift are adopted as optimization objectives, and the optimization design process is divided into two stages: global optimization and local optimization. In the first stage, the non-dominated sorting genetic algorithm (NSGA-II) is adopted to obtain the framework of the train head with an optimized global shape. In the second stage, Latin hypercube sampling (LHS) is applied in the local shape optimization of the PDE surface patches determined by the optimized framework to improve local details.

3.3.1 Overview for aerodynamic optimization

As shown in Fig. 3.11., the design flow of the proposed method can be basically divided into three parts: 1) PDE-based parametric modeling, 2) global optimization, and 3) local optimization.

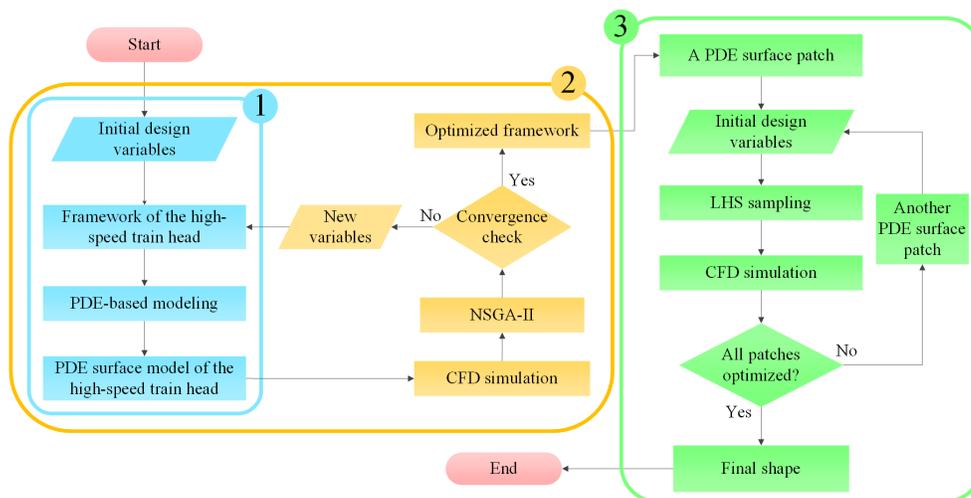


Figure 3.11: Overall design flow.

In the first part, i. e., PDE-based parametric modeling, the half of

a high-speed train head is first decomposed into 17 simple parts and the boundaries of each part are represented by 10 contour lines. After initializing the design variables, which control the shape of contour lines, the framework of the train head is obtained as shown in Fig. 3.12(a). Then, each part is described by a PDE surface patch generated from the finite difference solution of a vector-valued fourth-order PDE given in Eq. (3.1). Since any two adjacent PDE surface patches share the same boundary conditions, all the PDE surface patches are automatically and smoothly stitched together to represent the PDE surface model of the high-speed train head.

The second part is the global optimization process, which contains the first part. By inputting the PDE surface model of the train head into the CFD simulation, the aerodynamic drag coefficient of the whole train (C_d) and lift coefficient of the train tail (C_l) are obtained, which are two optimization objectives in NSGA-II. The objective function of NSGA-II is to minimize the C_d and $|C_l|$. There are 10 design variables which control the shapes of contour lines of the train head. The criterion of the global optimization convergence is that the mean error of the non-dominated front F_1 between two successive generations is less than or equal to a threshold value ε . If the global optimization does not converge, the values of the design variables will be updated and inputted into the first part to create a new framework of the train head. Otherwise, the globally optimized solution, i. e., the optimized framework of the train head, is obtained.

Based on the optimized framework of the train head from the second part, the third part is to optimize local shapes of all PDE surface patches in the CFD simulation. In the local optimization process, only one PDE surface patch is selected and optimized each time. Therefore, only three shape control parameters of the PDE surface patch are taken to be the design variables and set to some initial values. By using the LHS method, 125 sample sets are created to generate various shapes of the PDE surface patch. Then, the corresponding train heads with these shapes of the PDE surface patch are inputted into the CFD simulation. If the aerodynamic performance of C_d and C_l are improved, the PDE surface patch will be

replaced by the new shape. Otherwise, the original shape of the PDE surface patch will be kept.

Since there are 17 PDE surface patches of the train head, the above calculations are repeated 17 times until all the PDE surface patches have been optimized, and then the locally optimized solution, i. e., the final optimized shape of the train head, is obtained.

The global and local optimizations can be conducted simultaneously. However, the design variables of the simultaneous optimization will include 10 parameters of the contour lines and 3 control parameters of 17 PDE surface patches. The total number of design variables is 61, which cannot be simultaneously computed due to the limited computational resources, and it is also more difficult to find optimal results. Due to this reason, the global and local optimizations are conducted respectively in this work.

3.3.2 Parametric modelling method

By using the numerical PDE-based surface modelling method in Section 3.1, an original simplified high-speed train head model is built according to some practical constraints, such as the appropriate slenderness ratio, the appropriate space in driving cab, the good driver's perspective, etc. According to the shape changes, half of the high-speed train head is divided into 17 parts as shown in Fig. 3.12(a). Each part is represented by one PDE surface patch as shown in Fig. 3.12(b). All PDE surface patches are generated by Eq. (3.9) and controlled by boundary curves, boundary tangents and three shape control parameters while the numerical solution (3.9) of Eq. (3.1) gives the approximated values of the grid nodes for every PDE surface patch.

Global deformation controlled by ten design parameters

First, the whole shape of the high-speed train head is optimized by proposing a global deformation method. In order to achieve high efficiency of shape optimization of the high-speed train head, ten contour lines, i. e., $\widetilde{A_1A_6}$, $\widetilde{B_1B_6}$, $\widetilde{C_1C_6}$, $\widetilde{D_1D_6}$, $\widetilde{A_2D_2}$, $\widetilde{A_3D_3}$, $\widetilde{A_4D_4}$, $\widetilde{A_5D_5}$, $\widetilde{C_1D_1}$ and $\widetilde{A_1E_1C_1}$, are chosen to define half of the high speed train head as shown

in Fig. 3.12(a) and reduce the design variables. Each of the contour lines consists of some line segments and each line segment is between two adjacent joint vertices. For example, the contour line $\widetilde{A_4D_4}$ consists of three line segments $\widetilde{A_4B_4}$, $\widetilde{B_4C_4}$ and $\widetilde{C_4D_4}$. Since the two contour lines $\widetilde{A_1C_1}$ and $\widetilde{A_6D_6}$ decide the key position information of the train nose cone and coach, respectively, their shapes are fixed to ensure a smooth transition between different train parts.

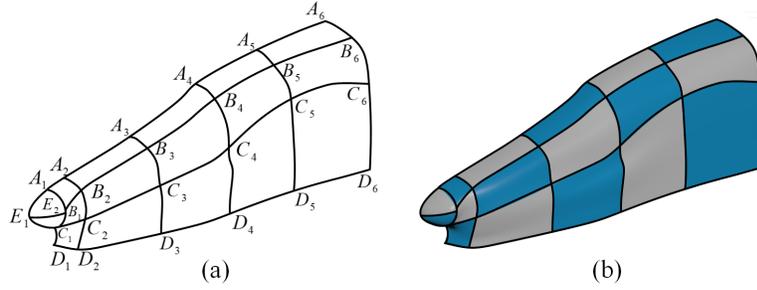


Figure 3.12: *The half of the high-speed train head model. (a) Contour lines. (b) PDE surface patches*

In order to further reduce the number of the total design variables in the optimization process, a design parameter is used to describe the shape of one contour line through the following equation:

$$\mathbf{L}_{in} = \bar{\mathbf{L}}_{in} \left[1 + \frac{\delta_i(n-1)(N_i-n)}{(n-1)^2 + (N_i-n)^2} \right] \quad (n = 1, 2, \dots, N_i; i = 1, 2, \dots, 10) \quad (3.22)$$

where the subscript i indicates the i^{th} contour line, N_i indicates the number of the total points on the i^{th} contour line, $\bar{\mathbf{L}}_{in}$ and \mathbf{L}_{in} are the vector-valued positions of the n^{th} point on the i^{th} contour line before and after the deformation, and δ_i is the design parameter, which controls the shape of the i^{th} contour line.

By applying Eq. (3.22) to deform a contour line, the deformation becomes bigger and bigger when moving from two end vertices to the center of the contour line, such as the red contour line $\widetilde{A_1A_6}$ shown in Fig. 3.13(a). Since each of the contour lines of the train head model is connected to some other line segments, the deformation of each of the contour lines will impact the shapes of these line segments. For example, when $\widetilde{A_1A_6}$ in Fig. 3.13(a) is deformed, the line segments $\widetilde{A_2B_2}$, $\widetilde{A_3B_3}$,

$\widetilde{A_4B_4}$ and $\widetilde{A_5B_5}$, which are connected to $\widetilde{A_1A_6}$, are deformed accordingly. The following deformation algorithm is used to describe the deformation of each connected line segment

$$\mathbf{Lc}_{jm} = \bar{\mathbf{Lc}}_{jm} + \Delta\mathbf{L}_j \cos\left(\frac{\pi(m-1)}{2(M_j-1)}\right) \quad (m = 1, 2, \dots, M_j) \quad (3.23)$$

where the subscript j indicates any one line segment connected to a deformed contour line, m and M_j represent the m^{th} point and the number of the total points on the j^{th} line segment, respectively, $m = 1$ is the joint vertex where the j^{th} line segment is connected to a deformed contour line, $m = M_j$ is another joint vertex on the j^{th} line segment, $\bar{\mathbf{Lc}}_{jm}$ and \mathbf{Lc}_{jm} are the vector-valued positions of the m^{th} point on the j^{th} line segment before and after the deformation, and $\Delta\mathbf{L}_j = \mathbf{L}_{in} - \bar{\mathbf{L}}_{in}$ is the vector-valued deformation at the joint vertex between the deformed j^{th} contour line and the connected j^{th} line segment.

Here, the contour lines $\widetilde{A_1A_6}$ and $\widetilde{A_1E_1C_1}$ are taken to be an example to show the effect of the contour line deformation on the shape of the high-speed train head depicted in Fig. 3.13. In Fig. 3.13(a), the contour lines in black and red present the original and deformed shapes, respectively. In Fig. 3.13(b), the blue meshes represent the PDE surface model of the original half train head and the transparent surface represents the deformed shape after adjusting $\widetilde{A_1A_6}$ and $\widetilde{A_1E_1C_1}$ by using Eqs. (3.22) and (3.23).

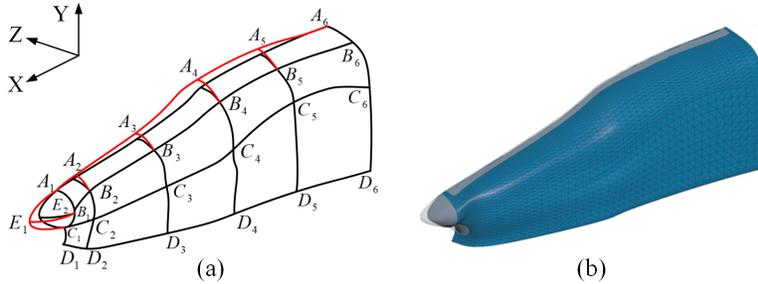


Figure 3.13: The effect of the contour line deformation on the shape of the half train head. (a) The deformations of the contour lines $\widetilde{A_1A_6}$ and $\widetilde{A_1E_1C_1}$ and their connected contour lines; (b) The PDE surface models before and after adjusting the two contour lines.

Table 3.3 shows ten design parameters, the corresponding contour lines and deformation directions. By adjusting the values of ten design parameters, the purpose of controlling the global deformation of the whole high-speed train head model with fewer design parameters is achieved.

Table 3.3: *The design parameters and deformation directions of ten contour lines.*

Design parameter	Contour line	Deformation direction
δ_1	$\widetilde{A_1A_6}$	Y axis
δ_2	$\widetilde{B_1B_6}$	Y axis
δ_3	$\widetilde{C_1C_6}$	Z axis
δ_4	$\widetilde{D_1D_6}$	Z axis
δ_5	$\widetilde{A_2D_2}$	Y axis
δ_6	$\widetilde{A_3D_3}$	Y axis
δ_7	$\widetilde{A_4D_4}$	Y axis
δ_8	$\widetilde{A_5D_5}$	Y axis
δ_9	$\widetilde{A_1E_1C_1}$	X axis
δ_{10}	$\widetilde{C_1D_1}$	Y axis

The process of the global deformation consists of five steps. First, ten design parameters $\delta_1, \delta_2, \dots, \delta_{10}$ are generated randomly within their design spaces. Second, one contour line is randomly selected and its deformation is determined by introducing the design parameter of the selected contour line into Eq. (3.22). Third, the deformations of all the line segments connected to the deformed contour line are determined by Eq. (3.23). Fourth, the shapes of all contour lines are updated with the deformations obtained in the second and third steps. Fifth, another contour line is selected and the second, third and fourth steps are repeated until all the ten contour lines are deformed.

Using the ten design parameters to control the global deformation of the high-speed train head greatly improves the efficiency in the first stage of the multi-objective optimization process of the high-speed train head. After optimizing the two optimization objectives, i. e., aerodynamic drag and aerodynamic lift, the optimized values of the ten design parameters are found and the globally optimized framework of the high-speed train

head is obtained.

Local deformation controlled by three shape control parameters

After obtaining the optimized framework of the high-speed train head, its surface model can be generated using Eq. (3.9) by filling 17 PDE surface patches into the framework as shown in Fig. 3.12(b). For acquiring a better hydro-mechanical property, the shape of the high-speed train head will be further adjusted by deforming the local shapes of PDE surface patches in the second stage. As explained previously, the shape of each PDE surface patch is controlled by the three shape control parameters in PDE (3.1), i. e., a_1 , a_2 and a_3 . With different values of the three shape control parameters, different surface shapes are obtained. For example, setting the three shape control parameters a_1 , a_2 and a_3 for the PDE surface patch $A_3A_4B_4B_3$ to two groups of different values: $a_1 = -2.49$, $a_2 = -2.49$, $a_3 = -1.68$ and $a_1 = -0.13$, $a_2 = -0.13$, $a_3 = -1.68$ gives different surface shapes shown in Fig. 3.14.

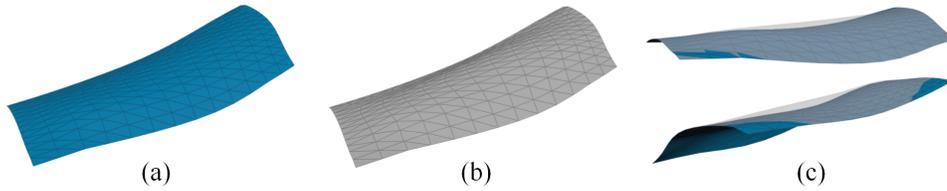


Figure 3.14: *The influence of the three shape control parameters on the shape deformation of the PDE surface patch $A_3A_4B_4B_3$. (a) The PDE surface patch obtained from $a_1 = -2.49$, $a_2 = -2.49$, $a_3 = -1.68$; (b) The PDE surface patch obtained from $a_1 = -0.13$, $a_2 = -0.13$, $a_3 = -1.68$; (c) Shape comparison between (a) and (b).*

Using the three shape control parameters in PDE (3.1) to control the deformation of all PDE surface patches, the local shape of the high-speed train head can be further adjusted, which improves the surface quality and optimization results in the second stage. After the local shape optimization, the ultimate optimized shape of the high-speed train head is obtained.

3.3.3 Optimization algorithm

NSGA-II

In the first stage of the multi-objective optimization design process, NSGA-II is applied to obtain the optimized framework of the high-speed train head. NSGA-II is an evolutionary multi-objective optimization algorithm and it is proposed by [Deb et al. 2002]. This algorithm is suitable for solving complex multi-objective optimization problems and has fast and accurate search performance. Fast non-dominated sorting approach with elite strategy is used in NSGA-II, which greatly improves the sorting speed. Moreover, the use of the elite strategy ensures that the good solution will not be discarded. The implementation process of NSGA-II is shown in Fig. 3.15.

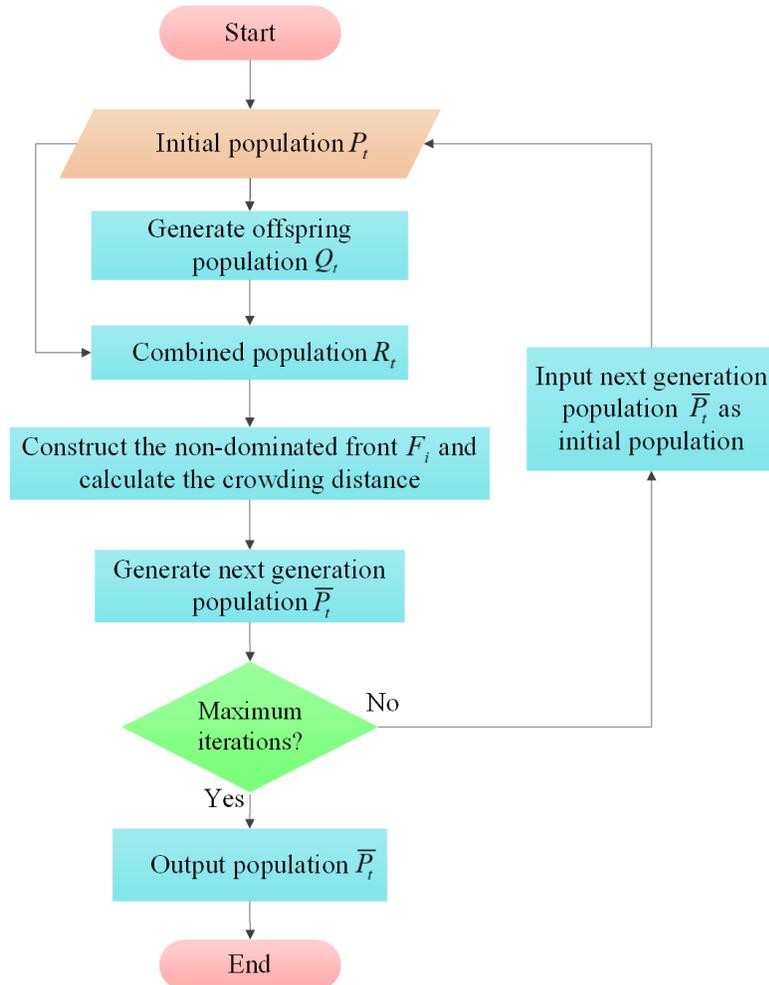


Figure 3.15: The implementation process of NSGA-II.

The NSGA-II algorithm includes five steps. First, the initial population P_t with a size N is randomly generated and non-dominated sorting is performed. Second, the selection, crossover, and mutation operations are applied on population P_t to generate an offspring population Q_t with the same size N . Third, the two populations P_t and Q_t are combined to form a new population R_t with a size $2N$. Then, the combined population R_t is sorted based on the non-dominated sorting approach to get non-dominated front F_i ($i = 1, 2, 3, \dots$). Meanwhile, the crowding distance of each individual in F_i is calculated. Fourth, according to the order of i from small to large, F_i is added into the next generation population \bar{P}_t . When the addition of a certain F_i causes the size of \bar{P}_t to exceed the population size N , individuals in the F_i will be added into \bar{P}_t according to the crowding distance in descending order instead of adding the whole F_i into \bar{P}_t . Fifth, if the termination condition, i. e, the number of iterations, reaches its maximum, the procedure ends. Otherwise, the \bar{P}_t is set as the initial population and the first step is restarted.

Latin hypercube sampling

In the second stage of the multi-objective aerodynamic optimization process, the local optimization is obtained by adjusting the three shape control parameters in PDE (3.1) to improve the local shapes of PDE surface patches of the high-speed train head. In order to obtain the applicable values of the three shape control parameters, LHS method is applied to initialize them in the defined design space. LHS is a stratified sampling technology to approximate the random sampling from multiple parameter distributions [McKay et al. 2000], and it is the generalization of Latin square to multi-dimensions and each axis-aligned hyperplane contains only one sample, which ensures all portions of the sample space are sampled and improves the sampling accuracy.

The three shape control parameters in PDE (3.1) are taken as three different input variables in the LHS process. In order to avoid an overlarge deformation of the PDE surface patch, the design space of each variable is defined as $[-3, 0.1]$ after a dozen experiments. The range of each variable is divided into 5 intervals with equal marginal probability, and

a sample is randomly selected from each interval. Since each variable generates 5 samples, totally 125 combinations of the three variables are obtained. Each combination including a sample set of a_1 , a_2 and a_3 can construct a different shape of the PDE surface patch, which results in a different high-speed train head. Therefore, 125 high-speed train models are automatically generated and applied to the CFD simulation in the local optimization process. The LHS result of the three shape control parameters is shown in Fig. 3.16.

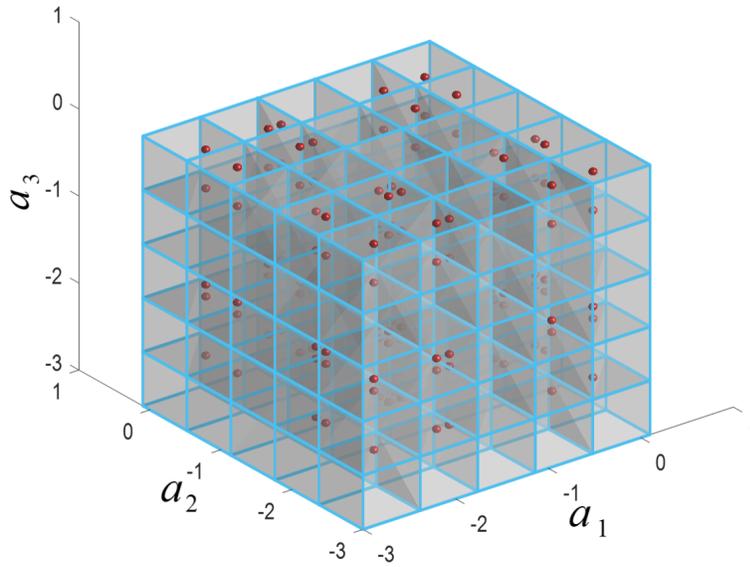


Figure 3.16: LHS for three shape control parameters a_1 , a_2 and a_3 .

3.3.4 CFD simulation method

In this work, the CFD simulation is conducted by using two software products: ICEM and FLUENT. ICEM is used to divide the whole high-speed train and the computational domain into aerodynamic meshes as shown in Fig. 3.18, and FLUENT is applied to carry out the subsequent fluid dynamics analysis for the high-speed train. To achieve automation of the optimization design, the process of the mesh generation and the aerodynamic calculation of the high-speed train are executed automatically by invoking the script files of ICEM and FLUENT respectively. These script files can be performed by batch commands.

The model and the computational domain

In order to evaluate the practical aerodynamic performance of the high-speed train head, a whole train model is constructed by adding a middle coach and a tail (same as the head) to the train head. Since the real high-speed train has a complex shape, which may greatly increase the time consumption of the parametric modeling, mesh generation and CFD simulation in the optimization process, only the relationship between the shape of the train head and the aerodynamic drag and lift forces is investigated, and the constructed high-speed train model is simplified by ignoring bogies and other auxiliary structures, as shown in Fig. 3.17(a). In this work, the generated high-speed train model is on the same scale of a real high-speed train, which runs at the speed of 300km/h in the open air without a crosswind.

First, a computational domain is constructed to simulate the flow field around the high-speed train. The train length is indicated with the symbol L ($L = 78\text{ m}$) and taken to be a characteristic length. The height and width of the train are $0.04435L$ and $0.04470L$, respectively. Then, the size of the computational domain is determined according to the characteristic length L . As shown in Fig. 3.17(b), the distance between the entrance of the computational domain and the nose cone of the train head is $1L$, and the distance between the nose cone of the train tail and the exit of the computational domain is $1.5L$. The distance from the ground to the top boundary of the computational domain is $0.5L$, and the distance from the train center to the boundary on both sides of the computational domain is $0.5L$. The distance between the train wheel and the ground is $0.00235L$, which represents the height of rail tracks.

CFD simulation

The software FLUENT is adopted to carry out the CFD simulation after obtaining the mesh file generated from the ICEM, and calculate the coefficients of aerodynamic drag and lift of the high-speed train model. Since the wind tunnel or real vehicle experiments have not yet been conducted by this research, there is no experimental data available. In

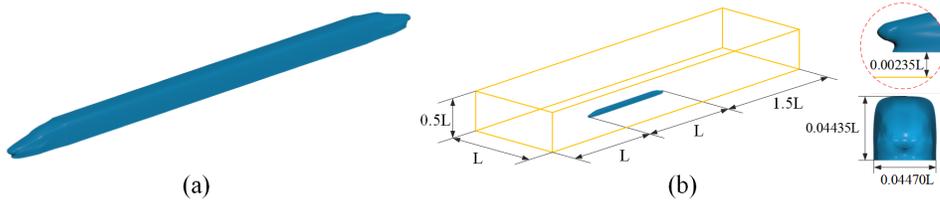


Figure 3.17: *The simplified model and the computational domain of the high-speed train. (a) The simplified model of the whole train. (b) The computational domain.*

order to ensure that the simulation results are reliable, the strategy of the simulation setup makes reference to previous similar work [Li et al. 2016; Zhang et al. 2018; Morden et al. 2015; Yao et al. 2014; Cheli et al. 2010], which has indicated that the CFD computation is able to predict the flow correctly.

Governing equations of fluid flows The Navier-Stokes (N-S) equations are the governing equations of fluid flows and have different forms for incompressible and compressible flows [Cebeci et al. 2005]. In this work, the flow around the high-speed train is considered to be an incompressible flow in the CFD simulation based on the following reasons. 1) The speed of the high-speed train is 300 km/h and the resultant Mach number is 0.245. 2) When the high-speed train is in the open air without passing other trains or going through a tunnel, the impact of the air density on the flow can be ignored. 3) The numerical simulation using an incompressible flow solver is accurate compared with experimental data [Cheli et al. 2010; Morden et al. 2015]. 4) The incompressible flow solver is commonly used in the studies of the aerodynamic optimization of high-speed train heads [Sun et al. 2010; Li et al. 2016; Zhang et al. 2018]. The incompressible N-S equations take the following tensor forms:

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (3.24)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\nu \frac{\partial u_i}{\partial x_j} \right) \quad (3.25)$$

where u_i is the velocity components in the i direction, ρ is the density, p is the pressure, and ν is the fluid kinematic viscosity.

Approximate approaches for N-S equations The high-speed train has a large Reynolds number so that the flow of the high-speed train is highly turbulent [Wang et al. 2017]. Since the high turbulent flow has fluctuations of pressure, temperature and velocity over a wide range of frequencies, solving the incompressible N-S equations is a formidable challenge [Cebeci et al. 2005]. The commonly used approximate computational approaches include direct numerical simulation (DNS), large eddy simulation (LES) and Reynolds-averaged Navier-Stokes (RANS) equations. Due to three reasons, the RANS approach is adopted in this work. These three reasons are: (1) Compared with DNS and LES, RANS has lower computational cost. (2) RANS is good at accurately predicting the pressure distribution and the frictional resistance [Bensow et al. 2006]. (3) The research here focuses on studying the drag and lift of the high-speed train, not the flow details. The RANS equations apply the Reynolds decomposition on the instantaneous incompressible N-S equations (3.24) and (3.25), which splits the flow into its mean and fluctuating components [Morden et al. 2015]. The incompressible RANS equations are defined as:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \quad (3.26)$$

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\nu \frac{\partial \bar{u}_i}{\partial x_j} \right) + \frac{1}{\rho} \frac{\partial \tau_{ij}}{\partial x_i} \quad (3.27)$$

where $\tau_{ij} = \overline{u_i u_j}$ is the Reynolds stress tensor which cannot be formally expressed in terms of mean flow variables and thus a turbulent model is used to close the equations.

Turbulent model The frequently used turbulent models include $k - \varepsilon$, $k - \omega$, and shear stress transport (SST) $k - \omega$ models. The SST $k - \omega$ model is selected in this research due to the following reasons. (1) The SST $k - \omega$ model blends the advantages of classical $k - \omega$ and $k - \varepsilon$ models, and can better model flows on the smooth surfaces of a high-speed train [Wang et al. 2017]. (2) The SST $k - \omega$ model is recommended as the

optimal RANS model based on experimental verifications [Morden et al. 2015]. (3) The SST $k - \omega$ model is commonly selected as the turbulence model in the studies of aerodynamic optimization of high-speed train heads [Yao et al. 2014, 2016; Zhang et al. 2018]

Wall function Since the gradients of velocity near the wall are steep, a large number of thin meshes should be used to accurately capture the gradients. However, these thin meshes will result in poor mesh quality and high computational cost. In order to solve this problem, a single large mesh instead of many thin meshes plus a nonlinear function called the wall function is used to simulate the gradient variation. In this work, the standard wall function is adopted because it works reasonably well for a broad range of wall-bounded flows [ANSYS 2009] and has been most widely used in the CFD simulation of high-speed train heads [Yao et al. 2014; Muñoz-Paniagua et al. 2014; Yao et al. 2015; Li et al. 2016; Zhang et al. 2018]. The standard wall function is defined as [ANSYS 2009]:

$$U^* = \begin{cases} y^*, & y^* < 11.225 \\ \frac{1}{\kappa} \ln(Ey^*), & y^* > 11.225 \end{cases} \quad (3.28)$$

where U^* is the dimensionless velocity, y^* is the dimensionless wall distance, $\kappa = 0.4187$ is the von Kármán constant, and $E = 9.793$ is the empirical constant.

Other CFD setup strategies The pressure-based segregated solver is used, and the SIMPLE scheme is introduced to couple the pressure and velocity. In the spatial discretization scheme, the least squares cell based gradient with the second order interpolation such as the second order pressure and the second order upwind momentum is applied. As shown in Fig. 3.17(b), the left and right sides of the computational domain are set as the velocity inlet and pressure outlet boundaries, respectively, and the two sides and the top are set as the symmetric boundary. The high-speed train surface is set as the non-slip wall boundary conditions, and the ground is set as the slip wall boundary conditions to simulate the ground effect and the slip velocity is equal to the speed of the train. In addition,

the far-field pressure is set to 1 atm, the temperature is set to 288 k , and the reference area is taken to be the maximum cross-sectional area of the train, which is 12.0638 m^2 .

In previous aerodynamic research work of high-speed trains, Cheli et al. [2010] and Morden et al. [2015] have conducted wind tunnel tests to verify their CFD simulation results. For example, Cheli et al. [2010] studied the static aerodynamic coefficients of a high-speed train with different wind angles, and showed the lateral and vertical force coefficients from the simulation and experiment are very close when the wind angle is below 20° . In the work of Morden et al. [2015], the C_d and C_l from simulation and experiment results are 0.14 against 0.13 and 0.17 against 0.23, respectively, which show a good agreement. The CFD setup strategies of their verified simulation models give us a solid reference. Therefore, the CFD setup strategies adopted in this thesis are similar to their work as shown in Table 3.4, and they ensure that the obtained results are as reliable as possible, though there is no experiment data support.

3.3.5 Mesh generation

When conducting a specific division of spatial meshes in ICEM, the quality and quantity of meshes have a significant influence on the computational efficiency, astringency and precision of the CFD simulation results.

Hexahedral meshes are used to divide the whole computational domain and prism meshes are distributed in the area around the body of the high-speed train. Since the train head bears most of the aerodynamic drag in simulation process, the mesh division around the train head is refined for improving the computational accuracy. In addition, in order to reduce computational cost, the mesh size of the middle coach is slightly larger than that of the train head and tail. The area around the body of the whole train utilizes five layers of fine prism meshes to accurately simulate the flow field around the train body. The y^+ is non-dimensional wall distance of the first cell from the wall based on the fluid local velocity. The range of the y^+ values obtained around the train body is from 5.99 to 77.91. Since it is difficult to get all surfaces of the train model to

Table 3.4: Comparison of CFD setup strategies of verified simulation models

	Cheli et al. [2010]	Morden et al. [2015]	The proposed method
Computational approach	Incompressible RANS equations	Incompressible RANS equations	Incompressible RANS equations
Turbulence model	$k - \varepsilon$	SST $k - \omega$	SST $k - \omega$
Pressure-velocity coupling algorithm	SIMPLE	SIMPLE	SIMPLE
Discretization scheme	Second-order upwind	Second-order upwind	Second-order upwind
Solver	Segregated solver	Potential flow solver	Segregated solver

have the desired y^+ value exactly, the majority of y^+ are made within the recommended ranges. Note that the y^* in Eq. (3.28) plays the same role as y^+ in measuring the dimensionless distance. The difference is that the velocity scale of y^* is based on the turbulent kinetic energy and the velocity scale of y^+ is based on the wall shear stress [ANSYS 2009]. The number of cell volumes within the boundary layer region around the train body is about 2.1 million. The mesh details are shown in Fig. 3.18.

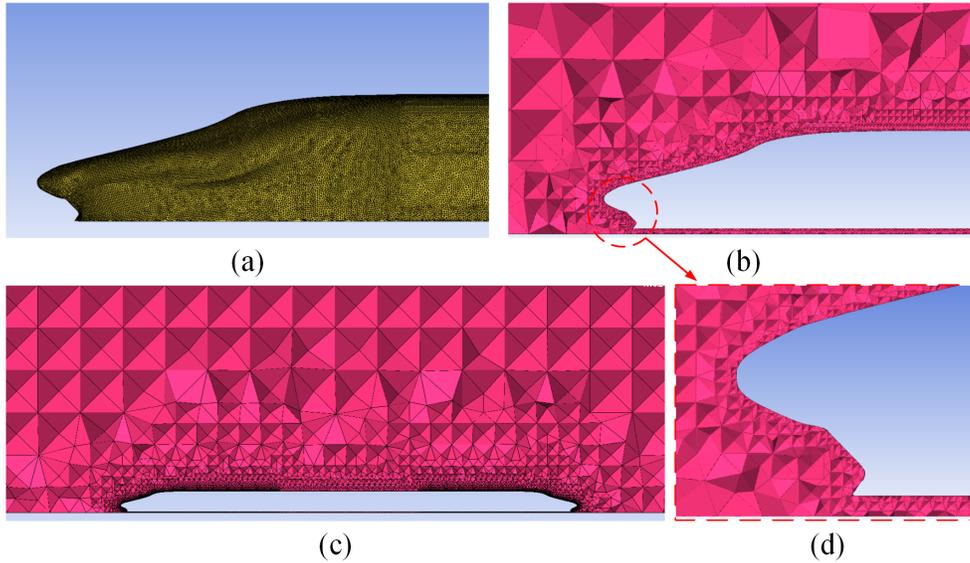


Figure 3.18: *The meshes for CFD simulation. (a) The surface meshes of the train head. (b) The longitudinal section of the train head in the computational domain. (c) The longitudinal section of the whole train in the computational domain. (d) The closer view of the nose cone and front spoiler of the train head.*

Since too many meshes will increase the simulation time and too few meshes will reduce the computational accuracy in CFD simulation process, it is important to generate a proper number of meshes [Yao et al. 2016]. Four sets of meshes with different mesh quantities, i. e., 7.2 million, 9.8 million, 14.3 million, and 22.2 million, are built to evaluate the influence of different meshes on the aerodynamic drag and lift. Table 3.5 shows the results of C_d and $|C_l|$ of the four sets of meshes. The values of C_d and $|C_l|$ obtained from the second set of meshes are 10.04%, 9.49% larger and 5.99%, 9.25% smaller than those obtained from the third and fourth sets of meshes, respectively. Due to the high number of simulations

needed to be carried out for the optimization algorithm, which require a large amount of computational cost, a compromise is made between the accuracy and the computational cost, and a discrepancy of approximated 10% is accepted. With these considerations, the second set of meshes with 9.8 million meshes is adopted for all flow field calculations in this work.

Table 3.5: *Computational results of the four sets of meshes.*

Sets of meshes	1	2	3	4
Mesh quantity (million)	7.2	9.8	14.3	22.2
C_d	0.1688	0.1557	0.1415	0.1422
$ C_l $	0.0052	0.0157	0.0167	0.0173

A small scale computation is also carried out for 22.2 million meshes following the same optimization flow of 9.8 million meshes. The optimized train head with 22.2 million meshes has a smaller drag and a larger lift compared with 9.8 million meshes, which is in accordance with Table 3.5, and the error of optimized C_d between 9.8 and 22.2 million meshes is within 10%, which indicates that the acceptable error of 10% covers the error caused by 9.8 million meshes.

3.3.6 Results and discussion

The multi-objective optimization process of the high-speed train head includes the global optimization and local optimization stages. The research first gets the optimized framework of the train head in the global optimization stage, then optimizes the local shape of PDE surface patches on the train head with the framework and obtains the optimized shape of the train head in the local optimization stage, and finally compares and analyses the aerodynamic performance of the train head in different stages.

Global optimization

In NSGA-II, C_d and C_l are set as the two optimization objectives. The population size is set to be 40 ($N = 40$) and the number of generations is

set to be 10. Moreover, the crossover and mutation probabilities are set as 0.9 and 0.1, respectively. The threshold value of the convergence criterion is set as $\varepsilon = 0.01$. The ranges of the ten design variables controlling the deformation of the train head framework are shown in Table 3.6. In order to avoid the distortion of the train head shape, the suitable lower and upper bounds of each design variable are found after a dozen experiments.

Table 3.6: *The ranges of the ten design variables.*

Design variable	Lower bound	Upper bound
δ_1	-0.15	0.20
δ_2	-0.10	0.15
δ_3	-0.20	0.10
δ_4	-0.15	0.15
δ_5	-0.15	0.15
δ_6	-0.10	0.10
δ_7	-0.10	0.10
δ_8	-0.10	0.10
δ_9	-0.05	0.10
δ_{10}	-0.30	0.25

By using NSGA-II, 400 solutions are obtained in which there are 13 Pareto-optimal solutions constructing a Pareto-optimal front as shown in Fig. 3.19. In order to find the most satisfactory solution among Pareto-optimal solutions and inspired by the minimum distance algorithm discussed in Li et al. [2016], a cost function is proposed, which assigns a proper weight to each objective and aggregates all of the objectives together. The cost function is defined by

$$D = \min \{D_1, D_2, \dots, D_S\}$$

$$D_i = \sqrt{(1 - \omega_c) \left(\frac{f_{C_d}(i)}{\min f_{C_d}} - 1 \right)^2 + \omega_c \left(\frac{|f_{C_l}(i)|}{\min |f_{C_l}|} - 1 \right)^2} \quad (i = 1, 2, \dots, S) \quad (3.29)$$

where i represents the i^{th} Pareto-optimal solution, S is the total number of Pareto-optimal solutions, $f_{C_d}(i)$ and $f_{C_l}(i)$ are C_d and C_l of the i^{th} Pareto-optimal solution, respectively, $\min f_{C_d} = \min \{f_{C_d}(1), f_{C_d}(2), \dots, f_{C_d}(S)\}$, $\min |f_{C_l}| = \min \{|f_{C_l}(1)|, |f_{C_l}(2)|, \dots, |f_{C_l}(S)|\}$, and ω_c is the user-defined weight.

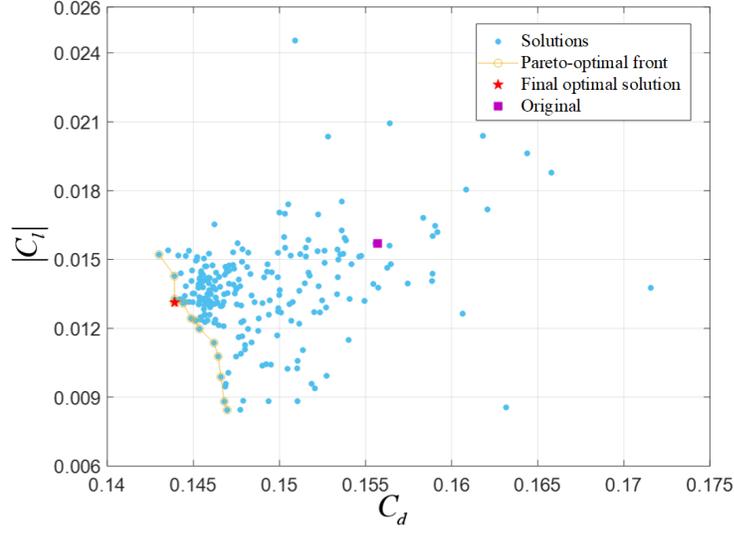


Figure 3.19: *The Pareto-optimal front in the global optimization stage.*

By applying different values of ω_c , different global optimized results are obtained from the 13 Pareto-optimal solutions. Since C_d plays a more important role in reducing the air resistance to the forward motion of a train compared with C_l , ω_c is set to 0.001 so that C_d is the main contributor based on previous work [Brockie & Baker 1990; Schetz 2001]. The final solution corresponding to the minimum D is selected as the global optimization result, which is shown as the red star marker in Fig. 3.19. The ten design parameters of the global solution construct the optimized framework of the high-speed train head, which will be used in the next local optimization stage.

Local optimization

In order to demonstrate the process of the local optimization, the patch $A_3A_4B_4B_3$ is taken to be an example. The design variables are the three shape control parameters of the patch $A_3A_4B_4B_3$. Same as the global optimization, the optimization objectives for the local optimization are C_d and C_l . LHS is employed to sample the design variables into 125 sample sets and use a dominated sorting to select the acceptable solutions whose C_d and C_l are not both dominated by the results of the optimized framework of the high-speed train head. In addition, Eq. (3.29)

is employed to determine the final solution from the sample sets. Since C_d is the main contributor in the global optimization stage, the influence of C_l in the local optimization stage is enhanced by setting $\omega_c=0.5$. The final solution is shown as the red star marker in Fig. 3.20.

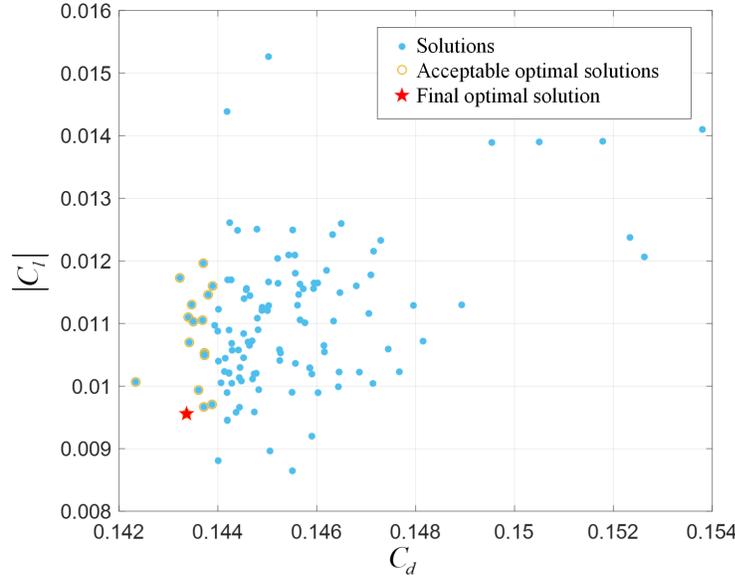


Figure 3.20: *The optimization results in the local optimization stage.*

Discussion

Since the proposed multi-objective optimization method based on the PDE parametric modeling includes global and local optimization stages, the results from the two stages are compared to discuss the advantages of the proposed method.

Fig. 3.21 shows the original (a), the globally optimized (b) and the locally optimized (c) high-speed train head models. After the global optimization, there are distinct deformations on the train head comparing (b) with (a). For example, the height of the cab decreases and the nose cone and the front spoiler move forward, as shown in (d). The local optimization further optimizes PDE surface patches and the locally optimized train head is shown in (c). It can be seen clearly from (d) that the locally optimized shape of the patch $A_3A_4B_4B_3$ representing the cab window on the train head becomes concave.

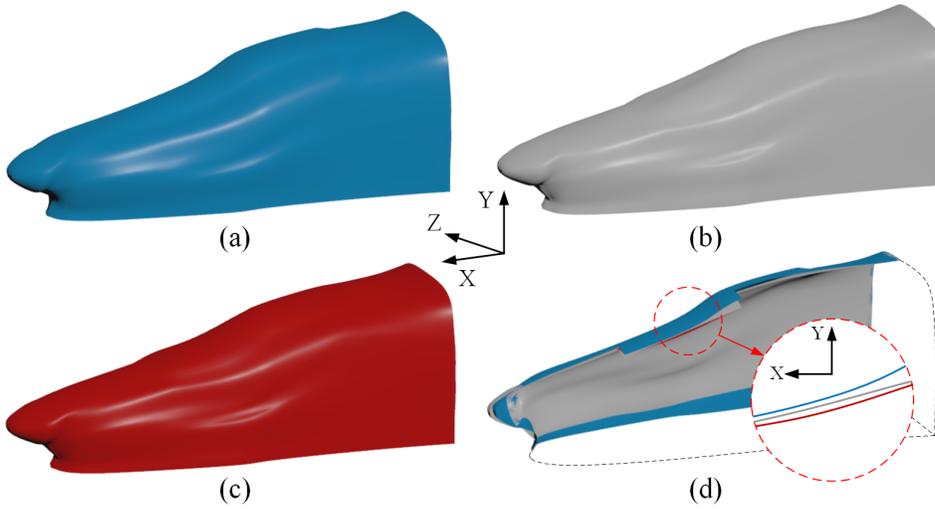


Figure 3.21: Comparison of the high-speed train head models. (a) The original model. (b) The globally optimized model. (c) The locally optimized model. (d) The cross-sections of the three models (the blue, grey and red lines represent original, globally optimized and locally optimized models, respectively).

The results of the optimization objectives C_d and C_l of the original, the globally optimized and the locally optimized high-speed trains are shown in Table 3.7. Compared with the original train, the C_d and $|C_l|$ of the globally optimized train are reduced by 7.58% and 16.56%, respectively. After the local optimization of the globally optimized train head, the two optimization objectives are further reduced. Compared with the original train, the C_d and $|C_l|$ of the locally optimized train are reduced by 7.90% and 38.85%, respectively.

Table 3.7: Aerodynamic drag and lift coefficients of the original, globally optimized (GO) and locally optimized (LO) trains.

	Original	GO	Reduction	LO	Reduction
C_d	0.1557	0.1439	7.58%	0.1434	7.90%
$ C_l $	0.0157	0.0131	16.56%	0.0096	38.85%

Aerodynamic drag and lift forces are mainly caused by the pressure force, which mainly exists on the surface of train head and tail. The shape deformation of the train head has a direct impact on the pressure distributions [Li et al. 2016]. In order to discuss the aerodynamic perfor-

mance of the train before and after the multi-objective optimization, the pressure distributions of the train head and tail are presented in Fig. 3.22. Since the scales of the pressure distribution of the train head and tail are different, two color bars in different scales are used for the train head and tail to clearly indicate the changes before and after the optimization.

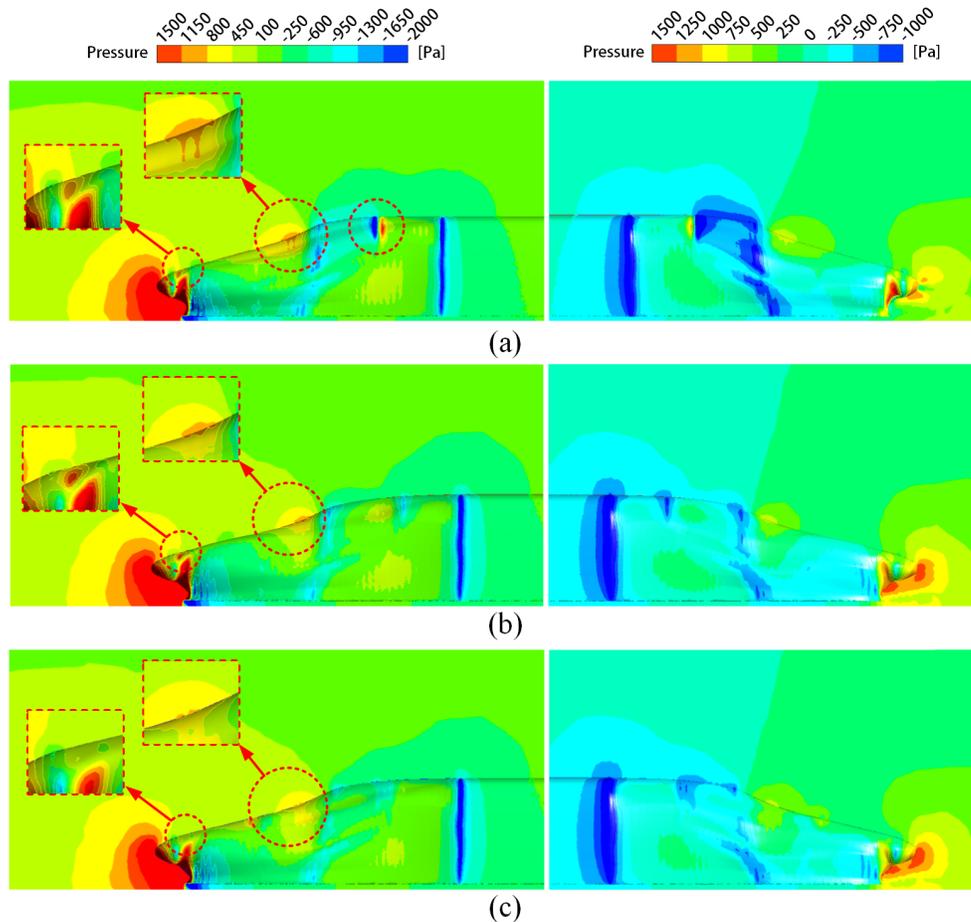


Figure 3.22: Comparison of the pressure distributions of the original (a), globally optimized (b) and locally optimized (c) high-speed train heads (the left column) and tails (the right column).

The left column in Fig. 3.22 shows the pressure distributions of the original, globally optimized and locally optimized train heads. There are mainly three high-pressure zones, near the nose cone, cab window and roof, respectively, which are indicated by the circle of red dashed lines. After the global optimization, the pressure near the cab window significantly decreases and the pressure variation near the roof, i. e., a negative pressure followed by a positive pressure, almost disappears,

but the pressure near the nose cone is a little larger than that of the original train head. After the local optimization, the pressure near the nose cone is reduced in addition to further decrease of the pressure near the cab window. Similarly, the right column in Fig. 3.22 shows the pressure distributions of train tails. After the global optimization, the high-pressure zone in front of the nose cone is larger than that of the original train, which gives the train tail a forward and an upward push. Moreover, the pressure near the cab window is decreased in both global and local optimizations, and the region with negative pressure near the roof is significantly reduced after the global optimization and slight improved after the local optimization. Through the multi-objective optimization process with the global and local optimization stages, the final optimized shape of the high-speed train head is obtained, which has an improved pressure distribution and small aerodynamic drag and lift forces.

Fig. 3.23 shows the streamlines around the high-speed train before and after the multi-objective optimization. From the overall views and the closer views of the train heads and tails, the streamlines are smooth near the train heads, and a flow separation occurs near the front spoiler of the train tail, which produces a trailing wake vortex as shown in (a). After global and local optimizations, the wake vortex is reduced and the streamlines around the train tail become smoother, which enlarge the high-pressure zone in front of the nose cone and give a forward push to reduce the drag, as shown in Fig. 3.22. The shape of the high-speed train head has a significant effect on the drag. By optimizing the shape, the air flow around the train can be smoother and the drag can be reduced.

In order to demonstrate the advantages of the proposed method, the comparison is made between the proposed method and the current shape optimization methods of high-speed train heads in term of the optimizing objectives C_d and C_l , as shown in Table 3.8. Note that the results obtained from the proposed method in Table 3.8 are based on the choice of ω_c in Eq. (3.29), i. e., $\omega_c=0.001$ in the global optimization and $\omega_c=0.5$ in the local optimization, and different choices of ω_c will produce different values of C_d and C_l as well as their reduction ratios. The high-speed

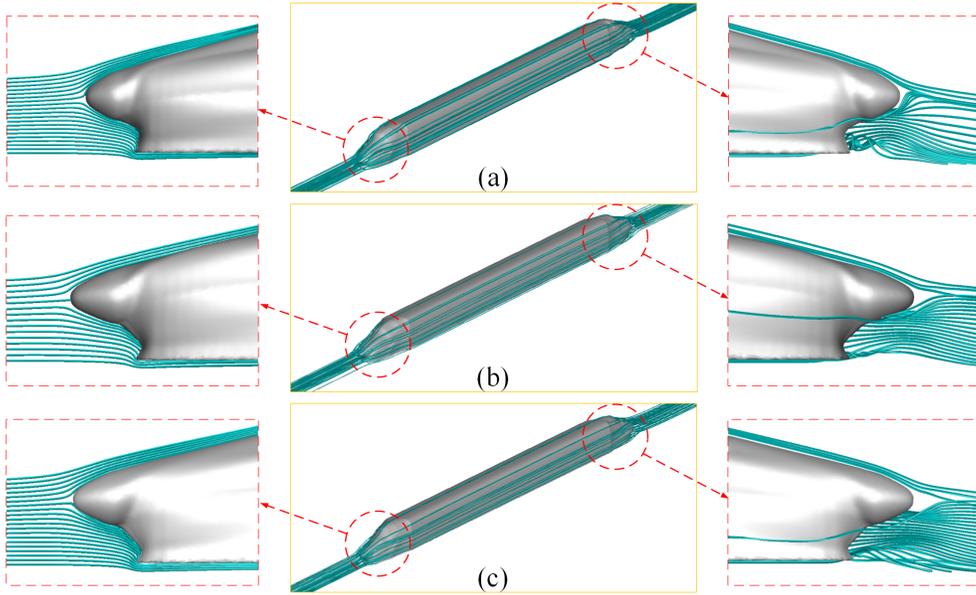


Figure 3.23: Comparison of the streamlines around the original (a), globally optimized (b) and locally optimized (c) train heads (the left column), whole trains (the middle column) and train tails (the right column).

trains in different studies are under the same running conditions, i. e., the high-speed train is in the open air without passing each other or going through a tunnel and the running speed is 300 km/h . The data in Table 3.8 indicate that although the original model used in this research already has good aerodynamic performance, i. e., the smallest values of C_d and $|C_l|$ in comparison with existing optimization studies, the proposed method can still achieve the maximum reduction rates 7.9% and 38.85% of C_d and $|C_l|$ among all the optimization methods and improve the drag and lift further. Therefore, the proposed method is more effective in improving the aerodynamic performance of high-speed train heads.

Table 3.8: Comparison of the proposed method with other optimization methods.

	The proposed method	Li et al. [2016]	Yao et al. [2014]	Yao et al. [2015]	Li et al. [2020]	Zhang et al. [2018]	Sun et al. [2010]
PM	Framework+PDE	FFD	ASD	Framework	RBF morph	FFD	ASD
OA	NSGA-II	NSGA-II	NSGA-II	MPSO	NSGA-II	NSGA-II	MOGA-II
ND	10+3	5	5	6	3	7	4
NM	9.8	3	11	8	3	22.66	0.73
TM	SST $k-\omega$	$k-\varepsilon$	SST $k-\omega$	SST $k-\omega$	$k-\varepsilon$	SST $k-\omega$	$k-\varepsilon$
RN	1.94×10^7	-	3×10^5	1.68×10^7	1.96×10^7	1.85×10^6	-
TG	Prototype	CRH2	CRH380A	-	Prototype	CRH2	CRH3
OR	C_d	0.1557	0.3998	0.1686	0.428	0.7262	0.2822
	$ C_l $	0.0157	0.1023	0.0207	-	-	-
OP	C_d	0.1434	0.3856	0.1659	0.4192	0.7087	0.277
	$ C_l $	0.0096	0.0912	0.0146	-	-	-
RE	C_d	7.90	3.55	1.60	2.05	2.42	1.85
	$ C_l $	38.85	10.85	27.86	29.47	-	-

PM: Parameterization method; OA: Optimization algorithm; ND: Number of design variables; NM: Number of mesh cells (million); TM: Turbulence model; RN: Reynolds number; TG: Train geometry; OR: Original shape; OP: Optimal shape; RE: Reduce (%); - : no data available.

3.3.7 Summary

In this section, a novel multi-objective aerodynamic optimization design process of a high-speed train head is proposed. The PDE-based parametric modeling method is applied to construct the parametric model of the high-speed train head, which can describe the complicated shape in detail with few design variables and keep the surface smooth. NSGA-II is adopted to obtain Pareto-optimal solutions in the global optimization stage of the high-speed train head and take the aerodynamic drag of the whole train and the aerodynamic lift of the train tail as the optimization objectives. Then, an optimized framework of the high-speed train head is selected from the Pareto-optimal solutions using an improved minimum distance algorithm. Based on the obtained optimized framework, LHS is introduced into the local optimization stage to obtain the final optimized shape of the train head by generating various sample sets of the three shape control parameters of PDE surface patches and optimizing the shape of each patch.

The proposed optimization method is demonstrated by analysing the aerodynamic characteristics, pressure distributions and streamlines of the optimization solutions in both global and local optimization stages compared with the original high-speed train head, and evaluating the drag and lift coefficients compared with other optimization methods. The analysis results indicate that the proposed method is able to optimize both global and local shapes and significantly improve the aerodynamic performance of the high-speed train head.

3.4 Summary

In this chapter, the engineering applications of numerical PDE surface patches are explored. The numerical PDE surface patch is first constructed using the finite difference method to solve 4-sided boundary conditions of a fourth-order PDE, and then the automatic and optimal conversion of numerical PDE surfaces representing high-speed train heads into NURBS surfaces is developed. Through this conversion, the PDE-based approach can be used in CAD, CAM and CAE systems.

The numerical PDE surface patch is also used to develop a novel multi-objective aerodynamic optimization method of high-speed train heads, which is the first pipeline of using the PDE-based approach to optimize shapes in the CFD simulation and indicates the powerful ability of the PDE-based approach in engineering applications.

Although the numerical solution of PDE-based surface modelling is applied in this chapter to explore the potential of PDE-based modelling in engineering applications, analytical solutions can also play the same role but at a lower computational cost. Since this chapter focuses on engineering applications rather than developing new solution methods, a numerical solution is adopted. It is no doubt that once the mathematical model of the analytical solution of PDE is built, it will be more effective than the numerical solution in engineering applications because of its low computational cost. For this reason, the analytical solution will be discussed in the next chapter and a unified PDE-based surface modelling framework will be developed, which will further improve engineering applications of the PDE-based modelling.

Chapter 4

Unified PDE-based surface modelling framework

Up to now, the existing research studies on analytical PDE surfaces focus on 2-sided PDE patches, i. e., creating a PDE surface patch from boundary conditions on two opposite boundaries. Various accurate and approximate analytical solutions of 2-sided PDE patches have been developed [Bloor & Wilson 1990; Zhang & You 2004; Ugail et al. 1999; Athanasopoulos et al. 2009; Bloor & Wilson 1989b; You et al. 2004b; You & Zhang 2003; Kubiesa et al. 2004; You et al. 2004a; Bloor & Wilson 2005; Zhang & You 2002]. Using analytical 2-sided PDE patches only is not applicable to all 3D modelling tasks. For example, 2-sided PDE patches are incapable in creating branched models. Here, a branched model is a 3D model with branching structures that are connected to but not part of the central body of the model, such as the trunk of a tree with branches and a human body with limbs. In order to release the potential of PDE-based surface modelling, analytical 3- and 4-sided patches with different continuity requirements should be developed.

In this chapter, a novel unified PDE mathematical model with analytical 2-, 3- and 4-sided surface patches will be proposed and the boundary conditions of C^n continuity between adjacent PDE surface patches will be formulated. By using the unified model, a PDE-based surface reconstruction method will be developed to analytically generate optimal surfaces

with C^n continuity under the constraint of the feature curves in automotive styling design, and a user interface of PDE-based surface modelling will be developed as a plug-in in the 3D animation and modelling software Blender, which demonstrates the potential of the developed technique for future commercial applications.

4.1 Unified PDE mathematical model

The PDE mathematical model usually chooses an elliptic PDE to solve the surface generation problem because the elliptic PDE is regarded as an averaging process throughout the entire surface [Castro et al. 2008]. Due to different orders and numbers of control parameters involved in a PDE, the PDE mathematical model has several forms. In this chapter, the target is to generate PDE surface patches with C^n continuity. In practice, the C^0 , C^1 and C^2 are frequently used continuities and C^3 is only required in some high-level surfaces designs such an automobile exterior [Hosaka 2012]. A higher degree of continuities, i. e., C^n ($n > 3$), are rarely used in surface modelling. Therefore, the symbol n of C^n continuity is defined up to 4.

Since a fourth-order PDE is sufficient in achieving C^n ($n = 0, 1, 2, 3, 4$) continuities as discussed below, the proposed unified mathematical model uses a vector-valued fourth-order PDE combined with three shape control parameters that offer enough degrees of freedom to satisfy arbitrary order of continuity. The vector-valued fourth-order PDE is defined as

$$(a_1 \frac{\partial^4}{\partial u^4} + a_2 \frac{\partial^4}{\partial u^2 \partial v^2} + a_3 \frac{\partial^4}{\partial v^4}) \mathbf{S}(u, v) = 0 \quad (4.1)$$

where $\mathbf{S}(u, v) = [x(u, v), y(u, v), z(u, v)]^T$ is a vector-valued position function which represents the generated parametric surface, a_1 , a_2 and a_3 are three shape control parameters, and u and v are the parametric variables defined by $u \in [0, 1]$ and $v \in [0, 1]$.

4.1.1 The boundary conditions

In order to obtain a unified PDE mathematical model, the constructed boundary conditions should involve 2-, 3- and 4-sided PDE surface patches, as shown in Fig. 4.1

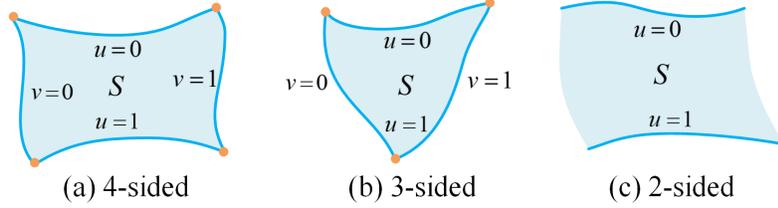


Figure 4.1: 2-, 3- and 4-sided PDE surface patches.

The boundary conditions of a 4-sided PDE surface patch are first constructed and then the boundary conditions of 2- and 3-sided PDE surface patches can be defined according to the 4-sided PDE surface patch. The 4-sided PDE surface patch has four boundary curves, as shown in Fig. 4.1(a), and the position, tangent, curvature and higher-order (C^n) constraints at the boundaries lead to the following boundary conditions

$$\begin{aligned}
 u = 0, \quad \mathbf{S} &= \mathbf{G}_0^0(v), \quad \frac{\partial \mathbf{S}}{\partial u} = \mathbf{G}_0^1(v), \quad \frac{\partial^2 \mathbf{S}}{\partial u^2} = \mathbf{G}_0^2(v), \quad \dots, \quad \frac{\partial^n \mathbf{S}}{\partial u^n} = \mathbf{G}_0^n(v) \\
 u = 1, \quad \mathbf{S} &= \mathbf{G}_1^0(v), \quad \frac{\partial \mathbf{S}}{\partial u} = \mathbf{G}_1^1(v), \quad \frac{\partial^2 \mathbf{S}}{\partial u^2} = \mathbf{G}_1^2(v), \quad \dots, \quad \frac{\partial^n \mathbf{S}}{\partial u^n} = \mathbf{G}_1^n(v) \\
 v = 0, \quad \mathbf{S} &= \mathbf{G}_2^0(u), \quad \frac{\partial \mathbf{S}}{\partial v} = \mathbf{G}_2^1(u), \quad \frac{\partial^2 \mathbf{S}}{\partial v^2} = \mathbf{G}_2^2(u), \quad \dots, \quad \frac{\partial^n \mathbf{S}}{\partial v^n} = \mathbf{G}_2^n(u) \\
 v = 1, \quad \mathbf{S} &= \mathbf{G}_3^0(u), \quad \frac{\partial \mathbf{S}}{\partial v} = \mathbf{G}_3^1(u), \quad \frac{\partial^2 \mathbf{S}}{\partial v^2} = \mathbf{G}_3^2(u), \quad \dots, \quad \frac{\partial^n \mathbf{S}}{\partial v^n} = \mathbf{G}_3^n(u)
 \end{aligned} \tag{4.2}$$

where \mathbf{G}_i^0 , \mathbf{G}_i^1 , \mathbf{G}_i^2 and \mathbf{G}_i^n ($i = 0, 1, 2, 3$) are boundary curves and the first, second, and n^{th} partial derivatives on the boundary curves, which are used to guarantee position, tangent, curvature, and C^n continuities on the boundary curves, respectively.

The boundary conditions in Eq. 4.2 can be decomposed into a linearly independent combination of some basic functions, such as exponential functions, trigonometric functions, power functions, logarithmic functions and the constant 1 [Zhang & You 2004]. After the decomposition, the

boundary conditions in Eq. 4.2 can be rewritten as

$$\begin{aligned}
u = 0, \quad \mathbf{S} &= \sum_{j=0}^J \mathbf{b}_{0j} s_j(v), \quad \frac{\partial \mathbf{S}}{\partial u} = \sum_{j=0}^J \mathbf{b}_{1j} s_j(v), \\
\frac{\partial^2 \mathbf{S}}{\partial u^2} &= \sum_{j=0}^J \mathbf{b}_{2j} s_j(v), \dots, \quad \frac{\partial^n \mathbf{S}}{\partial u^n} = \sum_{j=0}^J \mathbf{b}_{nj} s_j(v) \\
u = 1, \quad \mathbf{S} &= \sum_{j=0}^J \mathbf{c}_{0j} s_j(v), \quad \frac{\partial \mathbf{S}}{\partial u} = \sum_{j=0}^J \mathbf{c}_{1j} s_j(v), \\
\frac{\partial^2 \mathbf{S}}{\partial u^2} &= \sum_{j=0}^J \mathbf{c}_{2j} s_j(v), \dots, \quad \frac{\partial^n \mathbf{S}}{\partial u^n} = \sum_{j=0}^J \mathbf{c}_{nj} s_j(v) \\
v = 0, \quad \mathbf{S} &= \sum_{m=0}^M \mathbf{d}_{0m} g_m(u), \quad \frac{\partial \mathbf{S}}{\partial v} = \sum_{m=0}^M \mathbf{d}_{1m} g_m(u), \\
\frac{\partial^2 \mathbf{S}}{\partial v^2} &= \sum_{m=0}^M \mathbf{d}_{2m} g_m(u), \dots, \quad \frac{\partial^n \mathbf{S}}{\partial v^n} = \sum_{m=0}^M \mathbf{d}_{nm} g_m(u) \\
v = 1, \quad \mathbf{S} &= \sum_{m=0}^M \mathbf{e}_{0m} g_m(u), \quad \frac{\partial \mathbf{S}}{\partial v} = \sum_{m=0}^M \mathbf{e}_{1m} g_m(u), \\
\frac{\partial^2 \mathbf{S}}{\partial v^2} &= \sum_{m=0}^M \mathbf{e}_{2m} g_m(u), \dots, \quad \frac{\partial^n \mathbf{S}}{\partial v^n} = \sum_{m=0}^M \mathbf{e}_{nm} g_m(u)
\end{aligned} \tag{4.3}$$

where $s_j(v)$ and $g_m(u)$ are the linearly independent basic functions, \mathbf{b}_{ij} , \mathbf{c}_{ij} , \mathbf{d}_{im} and \mathbf{e}_{im} ($i = 0, 1, 2, \dots, n$) are known constants, n is the order of parametric continuity, and J and M represent the number of the basic functions.

For the 3-sided patch shown in Fig. 4.1(b), the curve at $u = 1$ becomes a point, which is the intersecting point of the curve at $v = 0$ and the curve at $v = 1$. In this case, the boundary conditions at $u = 1$ of Eq. 4.3 are not required. For the 2-sided PDE surface patch defined by two boundary curves as shown in Figure 4.1(c), its boundary constraints only contain $u = 0$ and $u = 1$.

4.1.2 Solutions

Normally, it is extremely difficult to obtain the analytical solution of Eq. 4.1 subject to boundary conditions in Eq. 4.3. To effectively solve

this PDE mathematical model, an approximate analytical solution in a composite form to represent the PDE surface \mathbf{S} is proposed, which has the form of

$$\mathbf{S}(u, v) = \sum_{j=0}^J \sum_{m=0}^M \mathbf{r}_{jm} u^m s_j(v) \quad (4.4)$$

where \mathbf{r}_{jm} are the unknown constants to be determined.

The approximate analytical solution combines the basic functions of variable v and the power functions of variable u . Substituting the approximate analytical solution in Eq. 4.4 into the first two boundary constraints of Eq. 4.3 and solving for the unknown constants \mathbf{r}_{jm} , the approximate analytical solution is changed into

$$\mathbf{S}(u, v) = \sum_{j=0}^J \left\{ \sum_{k=0}^n \frac{1}{k!} \mathbf{b}_{kj} u^k + \sum_{l=n+1}^{2n+1} \mathbf{r}_{jl} u^l + \sum_{m=2n+2}^M \mathbf{r}_{jm} u^m \right\} s_j(v) \quad (4.5)$$

where

$$\mathbf{r}_{jl} = \varphi_l(\mathbf{b}_{ij}, \mathbf{c}_{ij}) + \sum_{m=2n+2}^M \gamma_l(m) \mathbf{r}_{jm} \quad (i = 0, 1, 2, \dots, n) \quad (4.6)$$

In Eq. 4.6, $\varphi_l(\mathbf{b}_{ij}, \mathbf{c}_{ij})$ is a function consisting of \mathbf{b}_{ij} and \mathbf{c}_{ij} , and $\gamma_l(m)$ is a function of m . With the different values of the order n , the forms of $\varphi_l(\mathbf{b}_{ij}, \mathbf{c}_{ij})$ and $\gamma_l(m)$ are different. An application example of Eq. 4.5 for the 2-sided PDE surface patch is introduced in Section 4.2. By substituting Eq. 4.5 into Eq. 4.1, the PDE mathematical model can be effectively solved.

Since the PDE (4.1) is not an accuracy solution for $\mathbf{S}(u, v)$, $\mathbf{E}(u, v)$, which represents the left-hand side terms of PDE (4.1), is introduced below to describe the error of the PDE

$$\mathbf{E}(u, v) = (a_1 \frac{\partial^4}{\partial u^4} + a_2 \frac{\partial^4}{\partial u^2 \partial v^2} + a_3 \frac{\partial^4}{\partial v^4}) \mathbf{S}(u, v) \quad (4.7)$$

For the 3- and 4-sided PDE patches, when $M = 2n + 1$ and $J = 2n + 1$, all the unknown constants \mathbf{r}_{jm} can be obtained by using the boundary constraints at $v = 0$ and $v = 1$ of Eq. (4.3). When $M \geq 2n + 2$ and $J \geq$

$2n+2$, the amount of rest unknown constants is $(J-2n-1) \times (M-2n-1)$, and the rest unknown constants are determined by

$$\frac{\partial}{\partial \mathbf{r}_{jm}} \left[\sum_{j=2n+2}^J \sum_{m=2n+2}^M \mathbf{E}^2(u, v) \right] = 0 \quad (4.8)$$

For a 2-sided PDE patch, its boundary constraints are at $u = 0$ and $u = 1$ of Eq. (4.3). When $M = 2n + 1$, all the unknown constants \mathbf{r}_{jm} can be obtained. When $M \geq 2n + 2$, the amount of rest unknown constants is $(J + 1) \times (M - 2n - 1)$, and the rest unknown constants are determined by

$$\frac{\partial}{\partial \mathbf{r}_{jm}} \left[\sum_{j=0}^J \sum_{m=2n+2}^M \mathbf{E}^2(u, v) \right] = 0 \quad (4.9)$$

4.1.3 C^n continuity

For 2-, 3- and 4-sided PDE surface patches, the C^n continuity needs to be satisfied when two adjacent surface patches whose boundary curves are connected at the same points. In order to meet the C^n continuity requirement, the adjacent 4-sided PDE surface patches \mathbf{S} and $\bar{\mathbf{S}}$ meet at their common boundary curve \widetilde{AB} , as shown in Fig. 4.2. The C^n continuity between two adjacent surface patches can be achieved in the u or v direction. Here, only the C^n continuity in the v direction is introduced below. The C^n continuity in the u direction can follow the similar strategy.

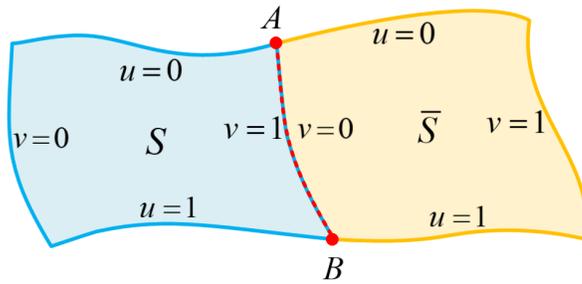


Figure 4.2: Two adjacent PDE surface patches.

At the joint vertices A and B , the PDE surface patches \mathbf{S} and $\bar{\mathbf{S}}$ should satisfy up to C^n continuity with respect to the parametric variable v

which gives

$$\begin{aligned}
\sum_{j=0}^J \mathbf{b}_{ij} \frac{\partial^i s_j(v=1)}{\partial v^i} &= \sum_{j=0}^{\bar{J}} \bar{\mathbf{b}}_{ij} \frac{\partial^i \bar{s}_j(v=0)}{\partial v^i} \\
\sum_{j=0}^J \mathbf{c}_{ij} \frac{\partial^i s_j(v=1)}{\partial v^i} &= \sum_{j=0}^{\bar{J}} \bar{\mathbf{c}}_{ij} \frac{\partial^i \bar{s}_j(v=0)}{\partial v^i} \quad (4.10)
\end{aligned}$$

$(i = 0, 1, 2, \dots, n)$

where \mathbf{b}_{ij} , \mathbf{c}_{ij} and $\bar{\mathbf{b}}_{ij}$, $\bar{\mathbf{c}}_{ij}$ are the constants in the boundary conditions of the PDE surface patches \mathbf{S} and $\bar{\mathbf{S}}$, respectively, and i is the order of partial derivatives, which represents the order of continuities.

Except the joint vertices A and B , the PDE surface patches \mathbf{S} and $\bar{\mathbf{S}}$ on the boundary curve \widetilde{AB} should also satisfy up to C^n continuity. These up to C^n continuities with respect to parametric variable v at the boundary curve \widetilde{AB} are found to be

$$\sum_{j=0}^J \mathbf{r}_{jm} \frac{\partial^i s_j(v=1)}{\partial v^i} = \sum_{j=0}^{\bar{J}} \bar{\mathbf{r}}_{jm} \frac{\partial^i \bar{s}_j(v=0)}{\partial v^i} \quad (i = 0, 1, 2, \dots, n) \quad (4.11)$$

Although only the C^n continuity of 4-sided PDE surface patches is presented here, Eqs. (4.10) and (4.11) are also applicable to 2- and 3-sided PDE surface patches.

4.2 PDE-based automotive styling design

In order to demonstrate the effectiveness of the proposed unified PDE mathematical model (Eq. 4.1) and the developed approximate analytical solution (Eq. 4.5), this section investigates the application of the proposed approximate analytical solution. The unified model is adopted to reconstruct surfaces from scanned data for automotive styling design. Since the feature curve of an automobile plays an important role in styling design, which determines the main shape of the automobile body and produces aesthetically pleasing forms [Tovey 1997], as shown in Fig. 4.3, the research will focus on surface reconstruction from two opposite feature

curves in this section.



Figure 4.3: *The feature curves in the styling design. The images taken from [Ford Media 2013; Jaguar 2017] show automotive clay models with black tapes, which represent feature curves.*

The 2-sided PDE surface patch is adopted as the type of the reconstructed surface, which has the C^1 continuity in the u direction and C^n continuity in the v direction. Hence, Eqs. (4.5) and (4.6) become

$$\mathbf{S}(u, v) = \sum_{j=0}^J \left\{ \mathbf{b}_{0j} + \mathbf{b}_{1j}u + \sum_{l=2}^3 \mathbf{r}_{jl}u^l + \sum_{m=4}^M \mathbf{r}_{jm}u^m \right\} s_j(v) \quad (4.12)$$

$$\mathbf{r}_{jl} = \varphi_l(\mathbf{b}_{ij}, \mathbf{c}_{ij}) + \sum_{m=4}^M \gamma_l(m) \mathbf{r}_{jm} \quad (i = 0, 1) \quad (4.13)$$

where

$$\begin{aligned} \varphi_2(\mathbf{b}_{ij}, \mathbf{c}_{ij}) &= -3\mathbf{b}_{0j} - 2\mathbf{b}_{1j} + 3\mathbf{c}_{0j} - \mathbf{c}_{1j} \\ \gamma_2(m) &= m - 3 \\ \varphi_3(\mathbf{b}_{ij}, \mathbf{c}_{ij}) &= 2\mathbf{b}_{0j} + \mathbf{b}_{1j} - 2\mathbf{c}_{0j} + \mathbf{c}_{1j} \\ \gamma_3(m) &= m - 2 \end{aligned} \quad (4.14)$$

By substituting Eq. (4.12) into Eq.(4.1) and using Eq. (4.9) if $M \geq 4$, the 2-sided PDE surface patch \mathbf{S} is obtained. For a PDE surface patch $\bar{\mathbf{S}}$ which is adjacent to \mathbf{S} in the v direction, the C^n continuity between them can be guaranteed using Eqs. (4.10) and (4.11).

4.2.1 Optimization

Given a set of data points $\mathbf{P} = \{\mathbf{p}_i\}_{i=0}^{I_p}$, the reconstructed surface $\mathbf{S} = \{\mathbf{s}_i\}_{i=0}^{I_s}$ is usually compared to \mathbf{P} according to some error measure methods, such as the point-wise Euclidean distance. However, for a set of scanned data, the distribution of point positions is irregular, and it is also difficult to make the number of the points in \mathbf{S} exactly the same as \mathbf{P} , i. e., $I_s = I_p$, even though I_p is known. In constrast, Hausdorff distance provides a measure of dissimilarity between two arbitrary point sets without determining the one-to-one correspondence between them [Aspert et al. 2002]. The classic Hausdorff distance will be taken to be the objective function H , which needs to be minimized during the optimization process through the following equation

$$\min_{a_1, a_2, a_3} H = \max_{\mathbf{s} \in \mathbf{S}} (\min_{\mathbf{p} \in \mathbf{P}} \|\mathbf{s} - \mathbf{p}\|) \quad (4.15)$$

where a_1 , a_2 and a_3 are shape control parameters in the PDE (4.1).

Note that Eq. (4.15) is a one-sided distance. Since \mathbf{P} is a set of scanned data points that usually contains some noise points, the one-sided distance only considers the closest point in \mathbf{P} for each point in \mathbf{S} , which can effectively avoid the influence of noise points from \mathbf{P} . Therefore, the one-sided distance is applied in Eq. (4.15).

The surface reconstruction procedure contains the following steps:

1. Input. A set of scanned data points $\mathbf{P} = \{\mathbf{p}_i\}_{i=0}^{I_p}$ representing a part of the automotive body and two opposite feature curves $\mathbf{G}_0^0(v)$ and $\mathbf{G}_1^0(v)$ defined by scanning the tapes' position, or section curve from scanned data or discretized CAD curve, etc.
2. Setup of boundary conditions. The feature curves $\mathbf{G}_0^0(v)$ and $\mathbf{G}_1^0(v)$ are regarded as the boundary curves, and the boundary tangents $\mathbf{G}_0^1(v)$ and $\mathbf{G}_1^1(v)$ are two functions of the tangent vectors along the unconstrained boundaries of the scanned data. By decomposing $\mathbf{G}_0^0(v)$, $\mathbf{G}_1^0(v)$, $\mathbf{G}_0^1(v)$ and $\mathbf{G}_1^1(v)$ into linearly independent combinations of basic functions $s_j(v)$, the boundary conditions (4.3) at $u = 0$ and $u = 1$ are obtained. Note that for achieving up to

C^n continuity between two adjacent surface patches \mathbf{S} and $\bar{\mathbf{S}}$, the known constants \mathbf{b}_{ij} , \mathbf{c}_{ij} and $\bar{\mathbf{b}}_{ij}$, $\bar{\mathbf{c}}_{ij}$ and unknown constants \mathbf{r}_{jm} and $\bar{\mathbf{r}}_{jm}$ must satisfy Eq. 4.10 and Eq. 4.11, respectively.

3. Surface generation. After initializing the three shape control parameters a_1 , a_2 and a_3 in Eq. 4.1, the unknown constants \mathbf{r}_{jm} can be obtained by substituting Eq. 4.12 into Eq. (4.1) to obtain Eq. 4.9 and solving Eq. (4.9), and then the PDE surface patch \mathbf{S} is generated using Eq. 4.4.
4. Optimization. The surface shape is changed to find the optimal surface through minimizing the objective function H in Eq. (4.15) with respect to the three design variables, i. e., the shape control parameters a_1 , a_2 and a_3 .
5. Surface conversion. Since the optimized PDE surface cannot be directly used in CAD systems for downstream engineering and manufacturing operations, the PDE surface is converted into NURBS format by applying the optimal NURBS conversion method presented in Chapter 3.

4.2.2 Experiments

In this experiment, power functions are used as the linearly independent basic functions to construct the boundary conditions, i. e., $f_j(v) = v^j$, the number of the basic functions is set to $J = 4$, and the terms of the power series are set to $M = 4$. To solve the nonlinear optimization problem, the classic genetic algorithm (GA) [Goldberg 1989] is used because it is a good solution to find a global minimum for highly nonlinear problems. In this research, the range of input design variables a_1 , a_2 and a_3 is set to $[-10, 10]$, and the convergence criterion is: the change in H is less than the specified tolerance 10^{-6} . The proposed method is implemented using MATLAB and runs on a desktop computer with Intel/Xeon E5-1650 (3.5 GHz) CPU.

For evaluating the quality of reconstructed surfaces with the proposed method, the three visual surface analysis tools listed below are used in

the following figures.

- Color map of error: For evaluating the similarity between the obtained results and input data, the surface error is measured with respect to the bounding box diagonal, which is evaluated with Metro tool [Cignoni et al. 1998]. Green and blue colors are used to represent the maximum and minimum errors, respectively.
- Zebra map: It is used to visualize curvature on surfaces and understand the shape and quality of surfaces especially check the C^0 , C^1 , and C^2 continuities at the join of two adjacent surfaces.
- Curvature combs: Since C^2 and C^3 continuities have similar zebra map, the curvature comb can help evaluate high-order continuities because it displays the curvature value at a given point.

Reconstruction of single surfaces

Single surface reconstruction experiments are conducted by using two examples of input meshes, each of which represents a part of the hood of two automotive exteriors, respectively, and their feature curves are specified upfront by power functions. Figures 4.4 and 4.5 show the two resulting surfaces for Examples 1 and 2. For each example, the evaluation of the reconstruction accuracy is achieved by error analysis with a color map and a zebra map, which indicates the difference between the resulting surface and input mesh data.

Compared with the input data in (b) of Fig. 4.4, the maximum error, mean error and root mean square error (RMSE) of the reconstructed surface in (c) are 1.25×10^{-2} , 1.94×10^{-3} and 2.59×10^{-3} , respectively. In Fig. 4.5, the maximum error, mean error and RMSE of the reconstructed surface in (c) are 2.90×10^{-3} , 6.44×10^{-4} and 8.44×10^{-4} , respectively. From these errors and Figs. 4.4 and 4.5, the following observations are obtained. (1) The reconstructed surface, especially at the position of feature curves, has high quality and accuracy. (2) The zebra maps of the reconstructed surface and input mesh data are almost identical. They indicate that the proposed method not only matches the requirement of feature curves' preservation but also leads to reconstructed surfaces with

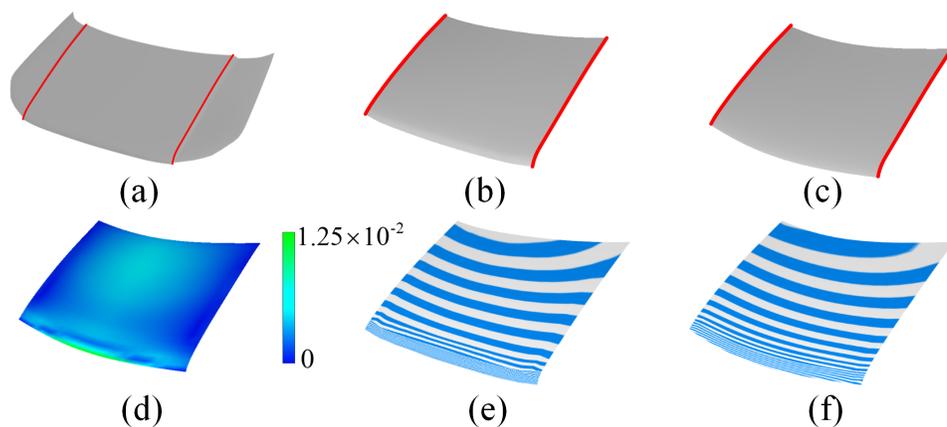


Figure 4.4: *Reconstructed surface and surface analysis for Example 1. (a) Input mesh of a car hood. (b) The part used for reconstruction from (a) with two feature curves (red curve). (c) Reconstructed surface. (d) Color map of error. (e) The zebra map of (b). (f) The zebra map of (c).*

a good curvature flow.

Reconstruction of adjacent surfaces with C^n continuity

In order to demonstrate the effectiveness of the proposed method in achieving C^n continuity between two adjacent surfaces in styling design, two mesh examples (Examples 3 and 4) from two automotive exteriors are presented as shown in Figs. 4.6(a) and 4.7(a). Each example consists of two adjacent parts, i. e, two roof meshes in Example 3 shown in Fig. 4.6(a) and a roof mesh and a rear window mesh in Example 4 shown in Fig. 4.7(a). The part in light grey with the yellow feature curves is named as Part 1 and the part in dark grey with the red feature curves is named as Part 2. In this experiment, the results with different orders of the continuity from C^0 to C^3 are shown in Figs. 4.6 (b)-(e) and 4.7 (b)-(e). Different from single surface reconstruction, the curvature combs on each part are used to evaluate the continuity between two adjacent reconstructed surfaces except the color map and zebra map. The values of three control parameters (CP) a_1 , a_2 and a_3 in the PDE (4.1), computing time (CT), the maximum error (MaxE), mean error (MeanE) and RMSE of the reconstructed C^0 , C^1 , C^2 and C^3 surfaces of Examples 3 and 4 are summarized in Table 4.1.

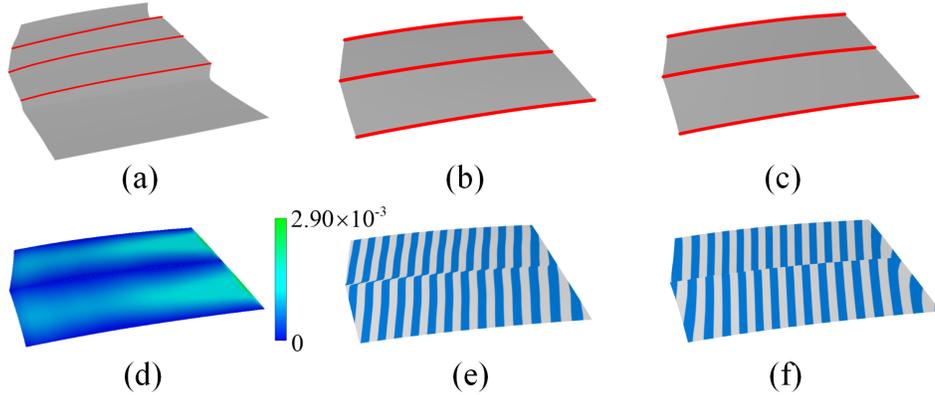


Figure 4.5: *Reconstructed surface and surface analysis for Example 2. (a) Input mesh of a car hood. (b) The part used for reconstruction from (a) with three feature curves (red curve). (c) Reconstructed surface. (d) Color map of error. (e) The zebra map of (b). (f) The zebra map of (c).*

In Table 4.1, the range of the computing time is from 20 to 29 seconds, and the order of magnitude of the maximum error is from 10^{-3} to 10^{-2} and the order of magnitude of the mean error and RMSE are both 10^{-3} . These results indicate that the proposed method can reconstruct surfaces with high precision in fitting the input data in a short time. From Figs. 4.6 and 4.7, the following observations are obtained. (1) In Figs. 4.6(b) and 4.7(b), the curvature combs of the two surfaces are at an angle without connection, and the zebra stripes do not line up due to C^0 continuity. (2) In Figs. 4.6(c) and 4.7(c), the curvature combs of the two surfaces are aligned but the curvature values are different at the join, and the zebra stripes line up but they turn sharply caused by C^1 continuity. (3) In Figs. 4.6(d) and 4.7(d), the curvature combs of the two surfaces are aligned and the curvature values are the same at the join, and the zebra stripes line up and flow smoothly created by C^2 continuity. (4) In Figs. 4.6(e) and 4.7(e), although the zebra stripes have no obvious difference compared with Figs. 4.6(d) and 4.7(d) at the join, the outline of curvature combs has tangential continuity at the join due to C^3 continuity. These observations indicate that the proposed method can achieve C^m continuity at the join of two adjacent reconstructed surfaces.

Table 4.1: Statistical data of C^0 , C^1 , C^2 and C^3 surfaces of Examples 3 and 4

Data	Type	Part	CP			CT(s)	MaxE	MeanE	RMSE
			a_1	a_2	a_3				
C^0	1		5.87	0.31	-0.44	21	7.59×10^{-3}	2.69×10^{-3}	3.34×10^{-3}
	2		-3.19	-7.92	9.76	22	10^{-3}	10^{-3}	10^{-3}
C^1	1		3.55	9.13	-5.52	25	7.56×10^{-3}	2.35×10^{-3}	2.82×10^{-3}
	2		0.48	9.74	-2.11	23	10^{-3}	10^{-3}	10^{-3}
Example 3 C^2	1		4.40	-3.60	-6.85	25	1.02×10^{-2}	2.56×10^{-3}	3.33×10^{-3}
	2		0.60	3.50	1.08	26	10^{-2}	10^{-3}	10^{-3}
C^3	1		1.75	0.08	-2.72	23	1.35×10^{-2}	3.90×10^{-3}	5.35×10^{-3}
	2		0.09	6.63	9.92	21	10^{-2}	10^{-3}	10^{-3}
C^0	1		1.52	2.62	6.19	20	2.15×10^{-2}	1.69×10^{-3}	3.08×10^{-3}
	2		-0.20	9.24	-2.92	25	10^{-2}	10^{-3}	10^{-3}
C^1	1		6.75	0.83	-8.70	29	2.32×10^{-2}	4.39×10^{-3}	7.39×10^{-3}
	2		-1.75	2.01	-9.25	27	10^{-2}	10^{-3}	10^{-3}
C^2	1		0.72	-7.74	-0.26	25	1.91×10^{-2}	3.93×10^{-3}	5.91×10^{-3}
	2		-0.21	-8.91	-1.56	23	10^{-2}	10^{-3}	10^{-3}
C^3	1		4.44	5.50	1.08	24	3.21×10^{-2}	4.08×10^{-3}	8.04×10^{-3}
	2		0.41	-1.90	5.92	21	10^{-2}	10^{-3}	10^{-3}

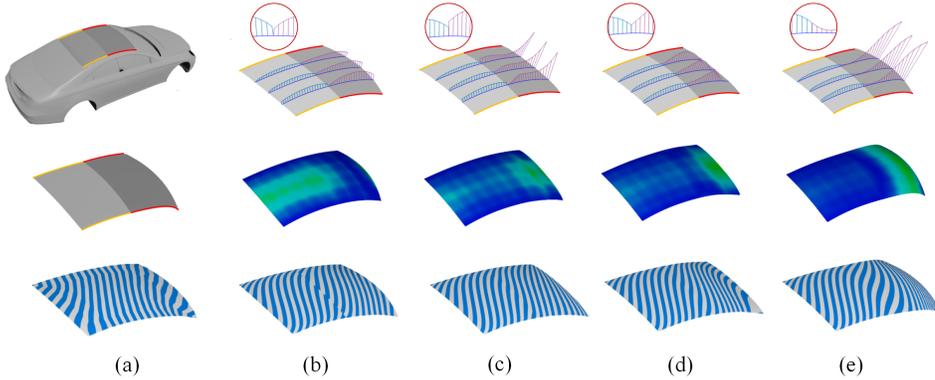


Figure 4.6: Reconstructed surfaces with C^0 , C^1 , C^2 and C^3 continuities and their surface analysis for Example 3. (a) Input mesh data, two parts for reconstruction and original zebra map. (b) Reconstructed C^0 surfaces with curvature combs, color map of error (the error range: $0 - 7.59 \times 10^{-3}$) and zebra map. (c) Reconstructed C^1 surfaces with curvature combs, color map of error (the error range: $0 - 7.56 \times 10^{-3}$) and zebra map. (d) Reconstructed C^2 surfaces with curvature combs, color map of error (the error range: $0 - 1.02 \times 10^{-2}$) and zebra map. (e) Reconstructed C^3 surfaces with curvature combs, color map of error (the error range: $0 - 1.35 \times 10^{-3}$) and zebra map. (The detail of curvature combs is shown in the red circle in the first row, and the blue and green colors in the second row represent the minimum and maximum errors respectively).

4.2.3 Comparison with existing methods

In order to demonstrate the advantages of the proposed method, a comparison with the existing methods is made. Since traditional sweeping and lofting methods generate surfaces from profile curves which can be regarded as feature curves, the proposed method is first compared with sweeping and lofting methods. Figure 4.8 shows loft (b), sweep (c) and PDE (d) surfaces reconstructed from input mesh data (a) with feature curves. The values of the maximum error, mean error and RMSE are listed in Table 4.2.

Table 4.2: Statistical data of loft, sweep and PDE surfaces

Method	Maximum error	Mean error	RMSE
Lofting	2.42×10^{-2}	1.36×10^{-2}	1.59×10^{-2}
Sweeping	2.74×10^{-2}	5.46×10^{-3}	7.54×10^{-3}
The proposed method	8.21×10^{-3}	1.99×10^{-3}	2.53×10^{-3}

In Fig. 4.8, the lofting method can preserve feature curves but cause

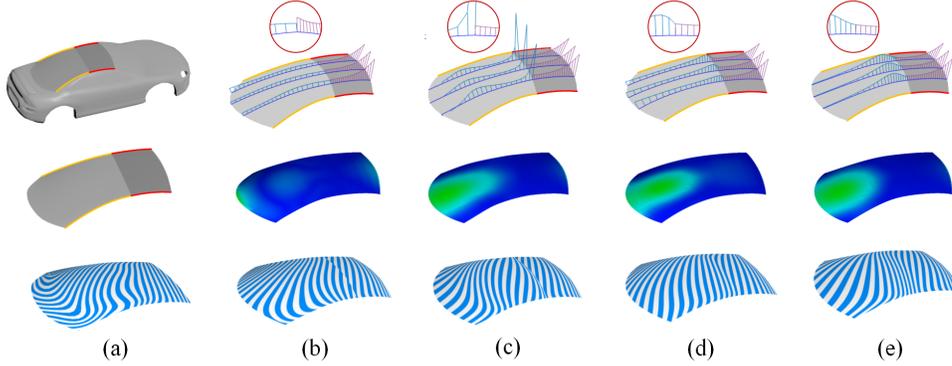


Figure 4.7: *Reconstructed surfaces with C^0 , C^1 , C^2 and C^3 continuities and their surface analysis for Example 4. (a) Input mesh data, two parts for reconstruction and original zebra map. (b) Reconstructed C^0 surfaces with curvature combs, color map of error (the error range: $0 - 2.15 \times 10^{-2}$) and zebra map. (c) Reconstructed C^1 surfaces with curvature combs, color map of error (the error range: $0 - 2.32 \times 10^{-2}$) and zebra map. (d) Reconstructed C^2 surfaces with curvature combs, color map of error (the error range: $0 - 1.91 \times 10^{-2}$) and zebra map. (e) Reconstructed C^3 surfaces with curvature combs, color map of error (the error range: $0 - 3.21 \times 10^{-2}$) and zebra map. (The detail of curvature combs is shown in the red circle in the first row, and the blue and green colors in the second row represent the minimum and maximum errors respectively).*

big error in the middle of the reconstructed surface because there are no design variables to guarantee the surface quality. Although sweeping method can produce less error in the middle of the reconstructed surface, it can only preserve one feature curve and needs the extra help of a backbone curve. In contrast, the proposed method can preserve two feature curves as well as guarantee the surface quality by using the shape control parameters in PDE. Moreover, from the data in Table 4.2, the reconstruction error of the proposed method is small especially the maximum error which is one order of magnitude smaller than other methods. These results manifest that the proposed method generates surfaces with higher accuracy than lofting and sweeping methods.

The proposed method is also compared with the improved sweep-based method presented in [Tsuchie 2019] in automotive styling design. Since both methods use the data of external surfaces from real automobiles and focus on the reconstruction work of automotive hoods and roofs,

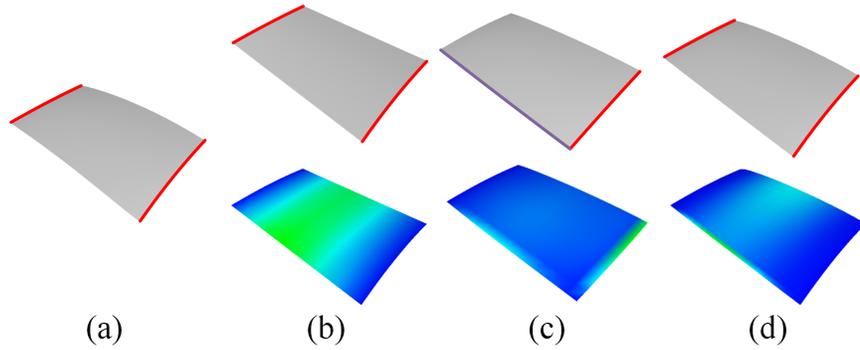


Figure 4.8: *Reconstructed loft, sweep and PDE surfaces with error analysis. (a) Input mesh data. (b) The loft surface with two feature curves (red curve) and color map of error (the error range: $0 - 2.42 \times 10^{-2}$). (c) The sweep surface with one feature curve (red curve) and a backbone curve (purple curve), and color map of error (the error range: $0 - 2.74 \times 10^{-2}$). (d) The PDE surface with two feature curves (red curve) and color map of error (the error range: $0 - 8.21 \times 10^{-3}$). (The blue and green colors in the second row represent the minimum and maximum errors respectively)*

the differences between both methods can be better demonstrated by comparing the statistical data of the reconstruction results from both methods, which are shown in Table 4.3. The order of magnitude of maximum errors with respect to the bounding box diagonal is used for a fair comparison. From the results in Table 4.3, it can be observed that the proposed method has obvious advantages such as a much smaller number of design variables, shorter computing time and higher order of surface continuity. Moreover, the reconstructed surfaces by using the improved sweep-based method need to be trimmed and remove redundant parts before stitching them together, whereas the surfaces generated by the proposed method can be directly applied in the next styling stage.

In this section, the comparison experiment only considers the CAD-generated surfaces rather than the parametric surfaces such as Bézier, B-splines and NURBS surfaces. The reason is that these surfaces are free-form, which cannot be constrained by feature curves. The aim in this research is to reconstruct surfaces whose boundaries exactly meet the feature curves. Therefore, it is fair to conduct the experiment with CAD-generated surfaces only.

To sum up, all above experiments have demonstrated the effectiveness

Table 4.3: *Comparison of the proposed method with another surface reconstruction method.*

	Improved sweep-based method	The proposed method
Number of design variables	30 control points	3 control parameters
Maximum error	$\geq 10^{-3}$	$10^{-3} - 10^{-2}$
Computing time (s)	29 – 315	20 – 29
Surface continuity	C^0	C^n
Need a trim?	Yes	No

of the proposed method. It is the first one able to reconstruct parametric surfaces with C^n continuity as well as preserve feature curves. Furthermore, the shape of reconstructed surface is dependent on the three control parameters in PDE, which greatly reduces the number of design variables in the optimization process and shortens the calculation time.

4.3 User interface

In order to further release the potential of the proposed unified PDE-based surface modelling framework and its approximate analytical solutions, a user interface named PDE Panel is developed as a plug-in in Blender. Blender is a widely known 3D animation and modelling software. It is free and open-source for creating 3D models, animated films, visual effects, art, motion graphics, interactive 3D applications, and computer games. With the powerful 3D interactive capability and a huge user base, Blender provides a good platform for the proposed PDE-based surface modelling technique to be used by worldwide artists.

PDE Panel consists of three functions, i. e., Single Patch Creator, Multi-Patch Creator and Surface Modification, as shown in Fig. 4.9. The developed user interface is a prototype version of the proposed analytical solution of the unified PDE-based surface modelling technique. In this prototype version, 2-, 3- and 4-sided PDE surface patches with C^0 and C^1 continuities are implemented in Single Patch Creator and 4-sided PDE surface patches with C^0 continuity are implemented in Multi-Patch Creator. Each part will be introduced in the following subsections.

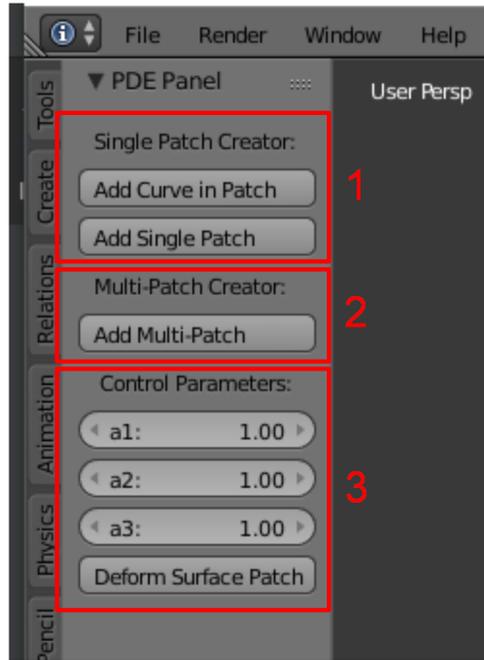


Figure 4.9: *PDE Panel in Blender.*

4.3.1 Single Patch Creator

The function of Single Patch Creator is to create a single 2-, 3- or 4-sided PDE surface patch from the curve network of two, three or four Bézier curves, respectively, as shown in Fig. 4.10. The curve fitting is applied to convert the Bézier curve into the basic functions in Eq. (4.3).

There are three steps for creating PDE surface patches.

(1) **Create Bézier curves** Choose the Bézier option in the Create menu in Blender, and adjust the position and shape of the Bézier curve by moving its two endpoints, and make the endpoints approach or coincide with other curve's endpoints. Then, subdivide the Bézier curve to have at least five control points. In order to correctly parameterize Bézier curves for creating a PDE surface patch, the Bézier curve needs at least five control points, as shown in Fig. 4.11.

(2) **Create a single PDE surface patch** Select and activate a Bézier curve (right mouse click), and then click the 'Add Curve in Patch' button in the PDE panel to mark the selected curve as the boundary curve of the surface patch. Then, add the rest Bézier curves to the patch in the

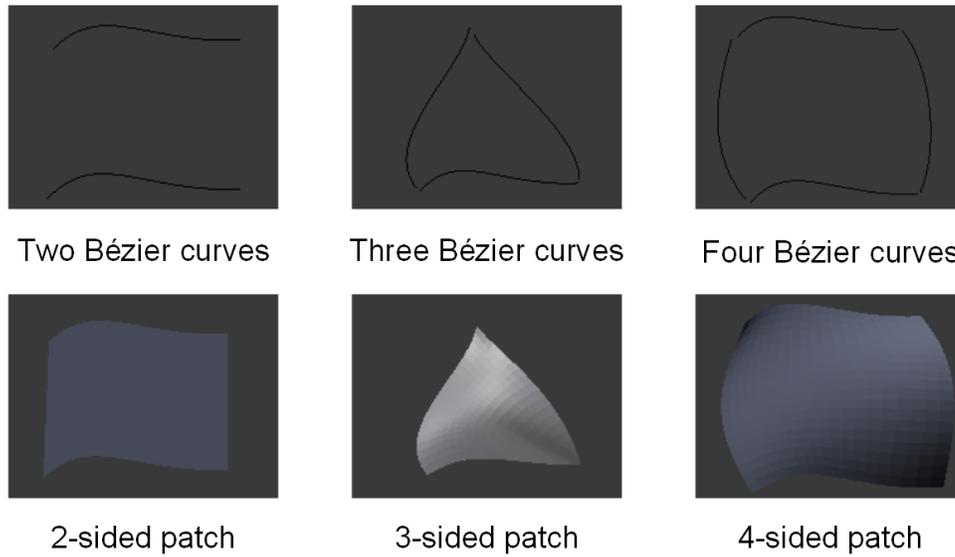


Figure 4.10: *2- 3- and 4-sided PDE surface patches.*

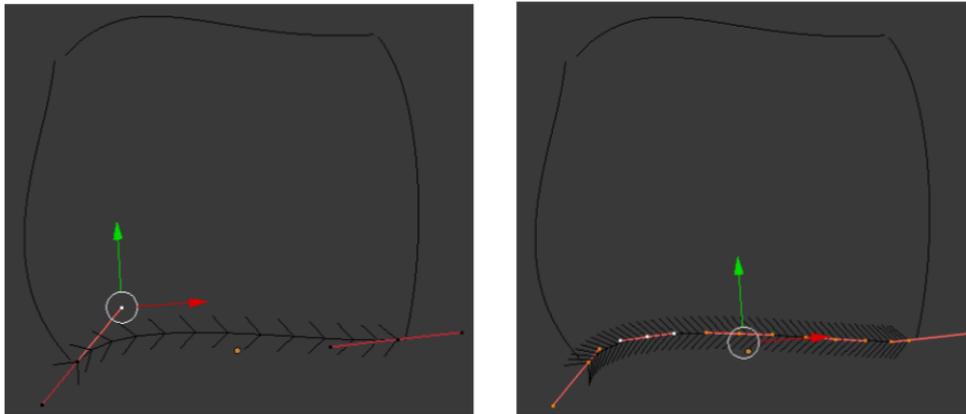


Figure 4.11: *Bézier curves (The left has 2 control points and the right has 5 control points).*

same way. Finally, after all curves are added, click the ‘Create Single Patch’ button in the PDE panel, and a surface patch will be generated quickly. For example, Figure 4.12 shows a 4-sided surface patch which is created from the curve network of four Bézier curves.

(3) **Create adjacent surface patches** A curve network can consist of many two, three and four Bézier curves. Once the curve network is created, the 2-, 3- and 4-sided surface patches will be generated one by one, as shown in Fig. 4.13. Note that if the curve network has both three and four Bézier curves, the 4-sided patch needs to be created first for

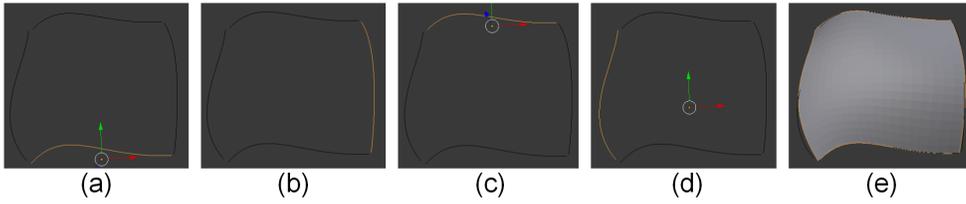


Figure 4.12: *The process of creating a 4-sided surface patch. (a)-(d) indicate the first, second, third and fourth selected curves. (e) is the generated PDE surface patch.*

ensuring the continuity between the two patches.

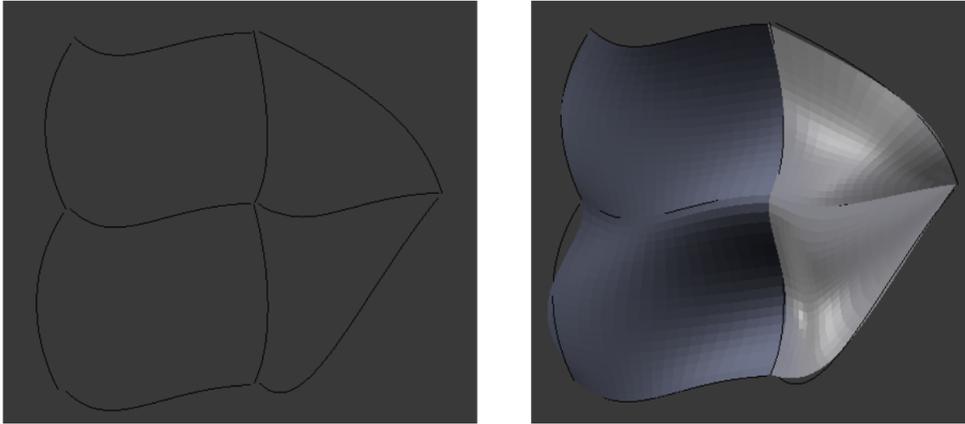


Figure 4.13: *The curve network and the generated adjacent surface patches.*

4.3.2 Multi-Patch Creator

The function of Multi-Patch Creator is to create a complete 3D surface model which consists of multiple surface patches generated from existing curve networks. This type of curve network can be a wireframe OBJ file. Figure 4.14 shows the curve network of a hat which is imported from the local OBJ file to Blender. In this prototype version, the imported curve networks should only be consisted of four closed curves.

The process of creating PDE surface patches from the imported curve network has only two steps. Users can first choose and activate the curve network model by clicking the right mouse, and then click the ‘Add Multi-Patch’ button in the PDE panel. A complete surface model will be generated quickly. Two examples of imported curve networks and created

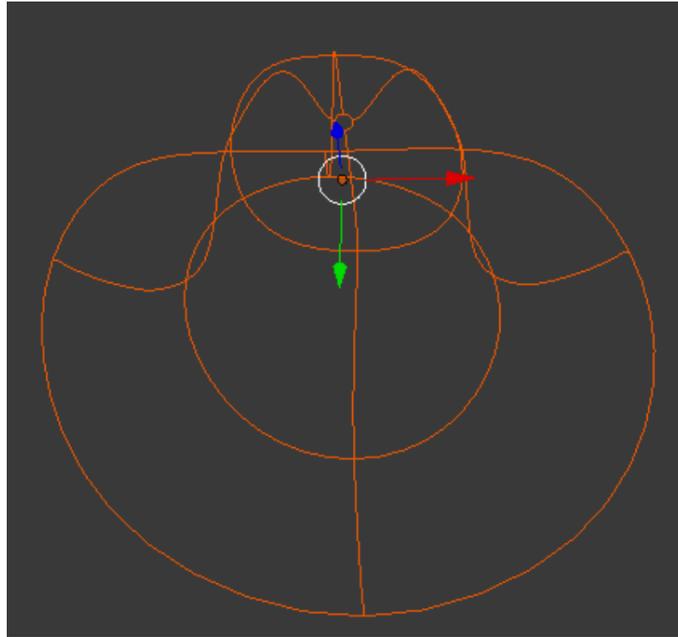


Figure 4.14: *The imported curve network of a hat.*

surface models are shown in Fig. 4.15.

4.3.3 Surface Modification

Since the three control parameters, i. e., a_1 , a_2 and a_3 in the PDE (4.1), define the shape of a PDE surface patch, the surface shape can be easily modified by changing the values of the three control parameters. The PDE Panel provides three input boxes for a_1 , a_2 and a_3 which are set to 1 by default. The shape of any PDE surface patch can be modified no matter whether it is generated by Single Patch Creator or Multi-Patch Creator. As shown in Fig. 4.16, the modification process includes three steps:

- Choose and activate a single surface patch. The patch can be the generated patch from Bézier curves or imported curve networks.
- Modify the values of control parameters through the three input boxes.
- Click the ‘Deform Surface Patch’ button in the PDE panel, and the modified surface is generated as shown in Fig. 4.17.

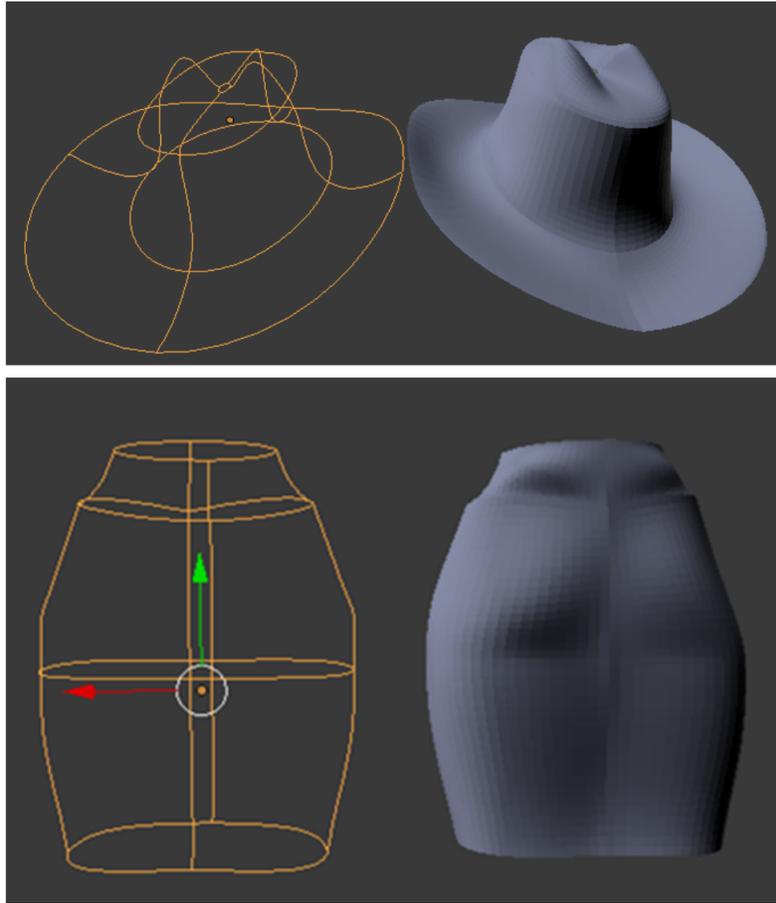


Figure 4.15: *Two examples of imported curve networks and created surface models (The top is a hat model, and the bottom is body model).*

4.4 Summary

In this chapter, the unified PDE-based surface modelling framework is presented, which can not only generate analytical 2-, 3- and 4-sided surface patches but also achieve C^m continuity between adjacent surface patches. In order to demonstrate the effectiveness of the unified model, the 2-sided PDE surface patches are applied to reconstruct surfaces under the constraints of the feature curves in automotive styling design which indicates the ability of PDE-based surface modelling in engineering applications. Besides, a user interface of PDE-based surface modelling is developed as a plug-in in the 3D animation and modelling software Blender, which can create 2-, 3- and 4-sided PDE surface patches. The user interface shows the potential of the developed technique for future

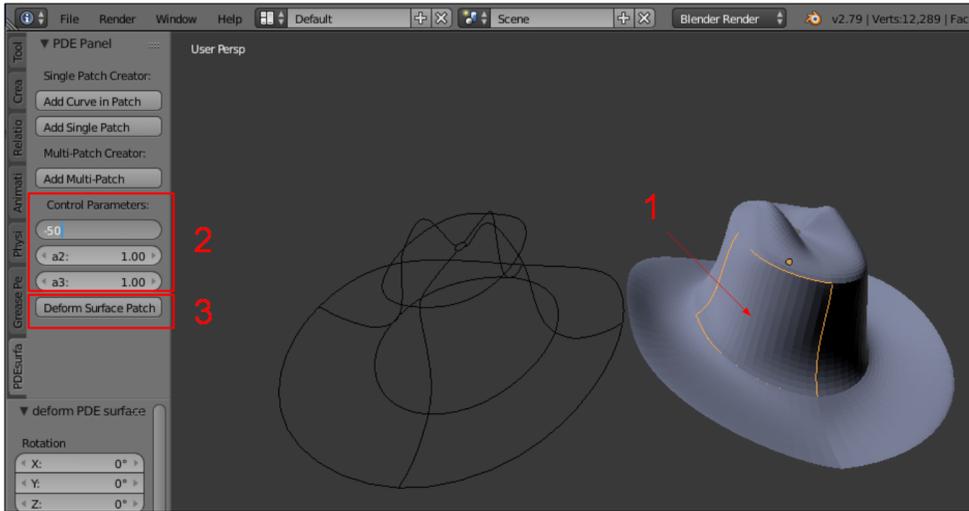


Figure 4.16: *The process of surface modification.*

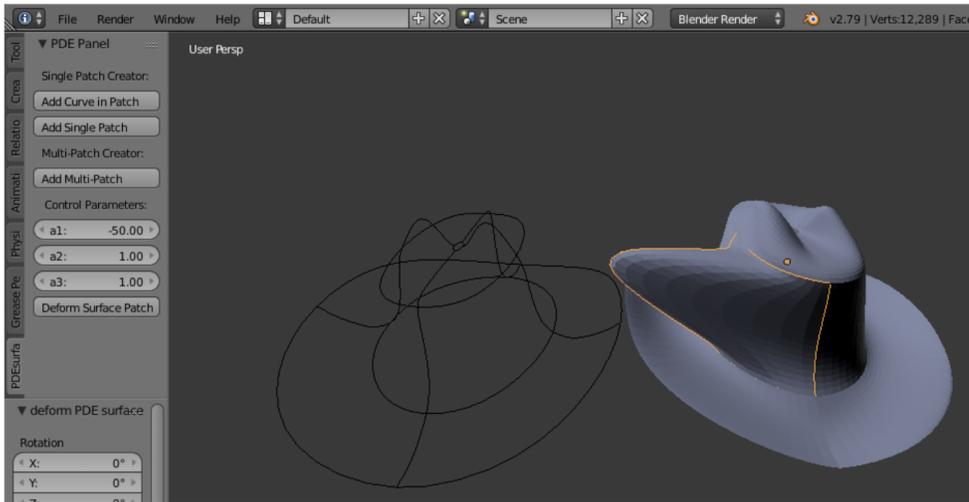


Figure 4.17: *The modified surface patch.*

commercial applications.

Except for creating surfaces from scratch, surface manipulation is also an important topic in surface modelling. It is to deform the surface shape depending on users' intention, which should not be limited by the position and region of deformation. Since the unified PDE-based surface modelling framework is designed for addressing frequently-used surface patches, i. e., 2-, 3- and 4-sided surface patches, it cannot deal with arbitrary boundaries. Therefore, an effective PDE-based surface manipulation method should be developed to tackle complex deformation

regions, which will be discussed in the next chapter.

Chapter 5

Physics-based surface manipulation using PDE

Physics-based surface manipulation, aiming to deform 3D surfaces following underlying physical laws and create more realistic shapes, has attracted considerable attention in the community of geometric modelling. Various existing methods have been proposed for physics-based surface manipulation [Du & Qin 2005a; You et al. 2006; Haixia et al. 2008; McDonnell & Qin 2007; Terzopoulos & Qin 1994]. Nevertheless, two challenges remain. One is how to deform a surface within a local region with an arbitrary boundary shape, and the other is how to find analytical solutions for reducing calculation costs and achieving interactive applications.

To tackle the two challenges, a novel but simple physics-based method using PDE technique will be presented to interactively manipulate surface shapes of 3D models with C^1 continuity. First, a deformation region on a surface in a 3D coordinate space is selected, and the boundary of the deformation region is mapped to a unit circle on a 2D parametric plane. Then a unit circle is defined as a thin elastic plate and the bending deformation of the plate under a sculpting force is simulated. To obtain the deformation, an approximate analytical solution of a fourth-order PDE subjected to the sculpting force and the boundary conditions of the circle is derived for x , y and z components. A length-based method is

used to determine the corresponding relationship between the vertices within the boundary of the deformation region and those within the circle. After obtaining the deformation of the unit circle, the coordinate values of all vertices within the deformation region are renewed to create a new surface shape according to the deformed values of corresponding points on the parametric plane. Finally, an interactive user interface is developed as a plug-in of the 3D modelling software package Maya to achieve surface manipulation.

5.1 Method

5.1.1 Theory of plate bending

The deformations of a surface can be simulated through those of elastic bending of a thin plate. When subjected to a lateral load q , the bending deformation of the plate in the xy plane can be described with the following fourth-order partial differential equation [Timoshenko & Woinowsky-Krieger 1959]

$$D\nabla^4 w = q \quad (5.1)$$

where w is the deflection of the plate in the z direction, and the symbol ∇^4 is a biharmonic differential operator defined by the following equation

$$\nabla^4 w = \frac{\partial^4 w}{\partial x^4} + 2\frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4} \quad (5.2)$$

and the symbol D is called the bending rigidity, which is defined by

$$D = \frac{Eh^3}{12(1 - \mu^2)} \quad (5.3)$$

In Eq. 5.3, E and μ are Young's modulus and Poisson's ratio of the plate, respectively, which are two material properties of the plate and reflect the capacity of the plate against bending deformations. The parameter h is the thickness of the plate.

Besides the applied load, material properties, geometric parameters,

and the boundary constraints of the plate also affect its bending deformations. In surface modeling applications, positional and tangential continuities are usually required. Therefore, in this research, the boundary constraints maintaining such continuities will be considered.

Assuming that the boundary of the plate is indicated by $\partial\Omega$, the fixed support boundary constraints in the plate bending can be written as

$$w = 0, \quad \frac{\partial w}{\partial x} = 0, \quad \frac{\partial w}{\partial y} = 0 \quad \text{on } \partial\Omega \quad (5.4)$$

Solving Eq. 5.1 subjected to boundary constraints (5.4), the bending deflection w for each point (x, y) within the plate is determined by resolving the function of the geometric position (x, y) , i. e., $w = f(x, y)$.

5.1.2 Mathematical model

Based on the theory of plate bending, the mathematical model of surface deformation can be developed. The two variables x, y and the deflection w in Eq. 5.1 form a three-dimensional space. When using a parametric representation to describe a three-dimensional surface, two parametric variables u and v and each component of the coordinate variables x, y and z also form a three-dimensional space. If the relationship between each component of coordinate variables x, y and z and the parametric variables u and v is defined with the same function as that of the plate bending, the deformations of a parametric surface can be determined through Eqs. (5.1-5.4).

Using the variable ξ to stand for each of x, y and z , the equations governing surface deformations become

$$D_\xi \nabla^4 \xi = q_\xi \quad (\xi = x, y, z) \quad (5.5)$$

where the load q_ξ is called the sculpting force, and

$$\nabla^4 \xi = \frac{\partial^4 \xi}{\partial u^4} + 2 \frac{\partial^4 \xi}{\partial u^2 \partial v^2} + \frac{\partial^4 \xi}{\partial v^4} \quad (5.6)$$

Accordingly, boundary constraints (5.4) are changed into

$$\xi = 0, \frac{\partial \xi}{\partial u} = 0, \frac{\partial \xi}{\partial v} = 0 \text{ on } \partial\Omega \quad (5.7)$$

From boundary constraints (5.7), it is known that both the displacements and the rotations of the deformed surface relative to the undeformed surface at the boundary are zero. Therefore, the deformed surface obtained from Eqs. 5.6 and 5.7 keeps both positional and tangential continuities at the boundary $\partial\Omega$. The continuity defined by Eq. 5.7 is C^1 continuity, which is more stringent than positional and tangential continuities on the boundary $\partial\Omega$. In the following subsection, how to solve Eq. 5.5 subjected to boundary constraints (5.7) will be investigated.

5.1.3 Solution

It is known from the theory of plate bending that the analytical solution of Eq. (5.1) subjected to the constraints of an elliptic boundary is obtainable. Since parametric variables u and v are often defined within 0 and 1, i. e., $0 \leq u \leq 1$ and $0 \leq v \leq 1$, in the mathematical representation of parametric surfaces, the boundary defined by parametric variables u and v can be taken to be a unit circle, i. e.,

$$u^2 + v^2 - 1 = 0 \quad (5.8)$$

It should be pointed out that although the boundary defined by parametric variables u and v is a circle, the corresponding boundary in the coordinate system defined by the coordinate variables x , y and z can be a very complicated shape including triangles, rectangles and three-dimensional curves etc., as shown in Fig. 5.1.

For the deformation which has both positional and tangential continuity at the boundary (5.8), it is assumed that the function relationships between each component of coordinate variables x , y and z and parametric variables u and v are

$$\xi = m_\xi(u^2 + v^2 - 1)^2 \quad (\xi = x, y, z) \quad (5.9)$$

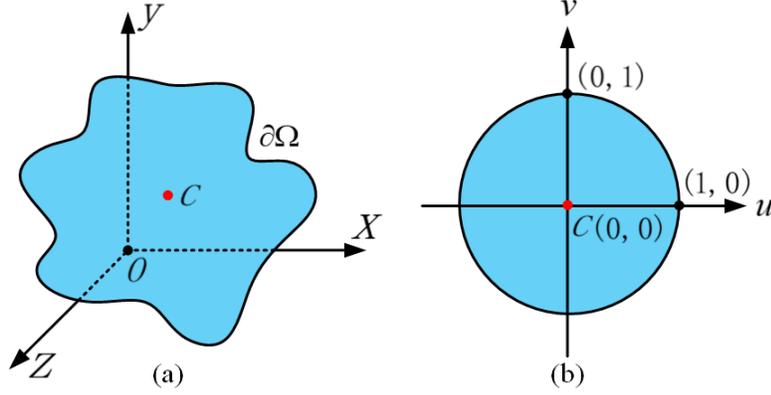


Figure 5.1: *The boundary of a deformation region. (a) the xyz geometric coordinate system. (b) the uv parametric coordinate system. (Point C is the centroid of the deformation region)*

where m_ξ is an unknown constant.

On the boundary $u^2 + v^2 - 1 = 0$, the deformation disappears, i. e., $\xi = m_\xi(u^2 + v^2 - 1)^2 = 0$. Therefore, the first of Eq. 5.7 is satisfied and the positional continuity is guaranteed. Differentiating Eq. 5.9 with respect to the parametric variables u and v , respectively, gives

$$\begin{aligned} \frac{\partial \xi}{\partial u} &= 4m_\xi u(u^2 + v^2 - 1) = 0 \\ \frac{\partial \xi}{\partial v} &= 4m_\xi v(u^2 + v^2 - 1) = 0 \end{aligned} \quad (5.10)$$

Eq. 5.10 indicates that the last two of Eq. 5.7 are also met and the tangential continuity is achieved on boundary $u^2 + v^2 - 1 = 0$.

Substituting Eq. 5.9 into 5.5 and solving for the unknown constant m_ξ , the following equation is obtained.

$$m_\xi = \frac{q_\xi}{64D_\xi} \quad (5.11)$$

Introducing Eq. 5.11 back into Eq. (5.9), the deformation of the surface within boundary $u^2 + v^2 - 1 = 0$ is found to be

$$\xi = \frac{q_\xi}{64D_\xi}(u^2 + v^2 - 1)^2 \quad (5.12)$$

Eq. (5.12) indicates that after applying a sculpting force q_ξ in the region within the boundary $u^2 + v^2 - 1 = 0$, the deformation in the region can be determined analytically.

5.2 Interaction design

In this section, a user interface based on the proposed surface manipulation method is developed so that users can directly apply the proposed method to interactively manipulate surface shapes. The core algorithm of the user interface, i. e., the deformation algorithm, is introduced in Section 5.1, and the remaining algorithms behind the user interface include the projective transformation, boundary extraction and mapping relationship.

In order to use Eq. 5.12 to create surface deformations, a deformation region on a 3D surface defined by coordinate variables x , y and z must be mapped to a circle in a 2D coordinate system defined by parametric variables u and v . However, it is difficult to directly find the mapping relationship between the two spaces. Instead, three steps are adopted to achieve this goal. First, all vertices within the deformation region are projected to a local 2D plane, and then the boundary curve of the projective deformation region is extracted. Finally, a length-based method is applied on the boundary curve to find the mapping relationship between the projective deformation region and parametric plane. The details are described in the following subsections.

5.2.1 Projective transformation

When users select the deformation region on the surface of a 3D model, it is assumed that the equation of users' view plane is $ax + by + cz + d = 0$, as shown in Fig. 5.2(a). For each vertex $\mathbf{p} = [p_x \ p_y \ p_z]^T$ within the deformation region, its projective vertex on the view plane is defined as $\bar{\mathbf{p}}_1 = [\bar{p}_x \ \bar{p}_y \ \bar{p}_z]^T$, which can be obtained through following projective transformation:

$$\bar{\mathbf{p}}_1 = -\frac{ap_x + bp_y + cp_z + d}{a^2 + b^2 + c^2} \mathbf{n} + \mathbf{p} \quad (5.13)$$

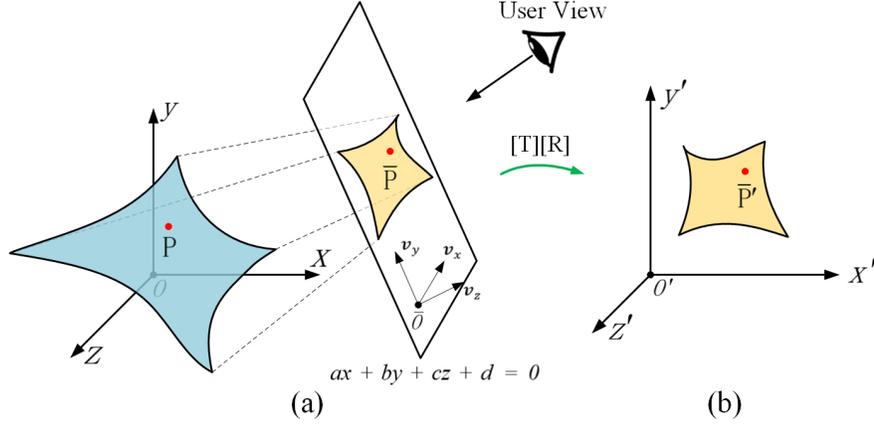


Figure 5.2: Projective transformation of the deformation region. (a) the global xyz coordinate system. (b) the local $x'y'z'$ coordinate system.

where $\mathbf{n} = [a \ b \ c]^T$ is the normal vector of the view plane.

The projective vertex $\bar{\mathbf{p}}_1$ is located in the global xyz coordinate system, and this coordinate system is transformed to a local $x'y'z'$ coordinate system by using a translation matrix $[\mathbf{T}]$ and a rotation matrix $[\mathbf{R}]$, as shown in Fig. 5.2(b). The origin of the local coordinate is $\bar{\mathbf{o}} = [\bar{o}_x \ \bar{o}_y \ \bar{o}_z]^T$ and its unit vectors along x' , y' and z' axes are $\mathbf{v}_x = [v_{xx} \ v_{xy} \ v_{xz}]^T$, $\mathbf{v}_y = [v_{yx} \ v_{yy} \ v_{yz}]^T$ and $\mathbf{v}_z = [v_{zx} \ v_{zy} \ v_{zz}]^T$, respectively. The transformation equation can be written as

$$\bar{\mathbf{p}}' = \bar{\mathbf{p}}_2 [\mathbf{T}] [\mathbf{R}] \quad (5.14)$$

where $\bar{\mathbf{p}}' = [\bar{p}'_x \ \bar{p}'_y \ \bar{p}'_z \ 1]$ and $\bar{\mathbf{p}}_2 = [\bar{p}_x \ \bar{p}_y \ \bar{p}_z \ 1]$, and

$$[\mathbf{T}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\bar{o}_x & -\bar{o}_y & -\bar{o}_z & 1 \end{bmatrix}, \quad [\mathbf{R}] = \begin{bmatrix} v_{xx} & v_{yx} & v_{zx} & 0 \\ v_{xy} & v_{yy} & v_{zy} & 0 \\ v_{xz} & v_{yz} & v_{zz} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.15)$$

Since all transformed vertices are on the $x'y'$ plane, the final projective vertex $\bar{\mathbf{p}}' = [\bar{p}'_x \ \bar{p}'_y]$ within the deformation region is obtained by removing the z' component.

5.2.2 Boundary extraction

In order to define the mapping relationship between boundary curve on the $x'y'$ plane and the circle on the 2D parametric plane, the key step is to extract the boundary curve of the deformation region. The alpha shape (α -shape) method is adopted to extract the boundary curve, which is to generate the convex hull of a finite set of points [Edelsbrunner et al. 1983]. The boundary extraction process is shown in Fig. 5.3. Given a set P of vertices $\bar{\mathbf{p}}'_i$ ($i = 1, 2, \dots, N$) from the projective deformation region, a Voronoi diagram is constructed as a set of cells (Fig. 5.3(a)), which is defined by

$$V_i = \{\mathbf{p} \in \mathbb{R}^2 \mid \|\mathbf{p} - \bar{\mathbf{p}}'_i\| \leq \|\mathbf{p} - \bar{\mathbf{p}}'_j\|, \forall j \neq i\} \quad (5.16)$$

where V_i is the locus of the \mathbf{p} closer to $\bar{\mathbf{p}}'_i$ than any other vertices.

Since the Delaunay triangulation is the dual shape of the Voronoi diagram, the Delaunay triangulation can be obtained by connecting all the vertices in P that share common Voronoi faces, as shown in Fig. 5.3(b). Then, by giving the parameter $\alpha = d_P \alpha_0$ where d_P is the distance between two closest vertices in P and α_0 is the threshold value, if the length of any edge of a triangle is larger than 2α , this triangle is removed. After that, the circles of radius α containing two end vertices of edges of the rest triangles are constructed. If a circle contains no vertices from P in its interior, this edge is regarded as a valid boundary edge. Finally, all valid boundary edges form the boundary curve, as shown in Fig. 5.3(c).

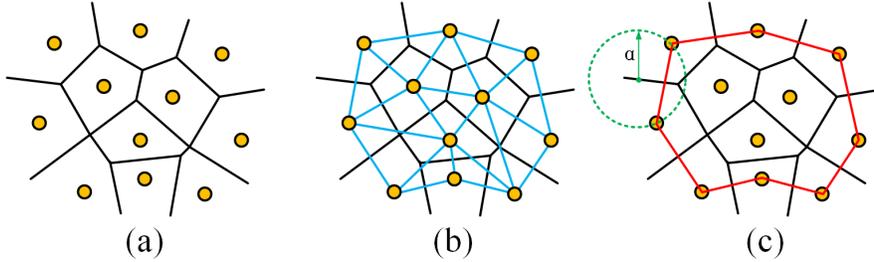


Figure 5.3: Extraction of boundary curve. (a) Voronoi diagram for a set of vertices. (b) Delaunay triangulation. (c) The boundary curve of α -shape.

5.2.3 Mapping relationship

The mapping, in computer graphics, is the key to achieve the parametrization [Sheffer et al. 2006]. Commonly used mapping methods include minimizing angle distortions (the conformal mapping) and area distortions (the equiareal mapping) [Fong 2015]. However, in this research, the goal is to make the surface deformation in the 3D space as similar as possible to the deformation in the 2D parametric space as well as keep the original shape information, rather than reduce the angle or area distortions. In order to achieve this goal, the one-to-one mapping between the projective vertices on the $x'y'$ plane and the parametric points on the parametric plane is necessary and the elastic deformation value of each parametric point needs to be exactly exerted on the corresponding original vertex. Through the one-to-one mapping, the physic-based deformation in the parametric space can be accurately and smoothly reconstructed in the 3D space.

In the parametric space, after generating physic-based deformation within the circle, the deformed shape becomes bigger when moving from the circle to the center, as shown in Fig. 5.7. For reconstructing the deformation in the 3D space, the distance from the point to the center of the circle on the parametric plane should be consistent with the distance from the projective vertex to the centroid on the $x'y'$ plane. The radial mapping [Fong 2015] has the similar idea which makes points only move along radial lines from the center of the disc. However, this method can only deal with the mapping between a square and a disc. Here, a simple and effective length-based mapping method is proposed to achieve the research goal.

The deformation region on the $x'y'$ plane has N vertices. As shown in Fig. 5.4(a), it is defined that the vertices on the boundary curve are A_i ($i = 1, 2, \dots, M$) and the rest vertices within the boundary curve are B_j ($j = 1, 2, \dots, N - M$). The centroid of the deformation region is C , and the point D_j is the intersecting point of the line $\overline{A_i A_{i+1}}$ and the extended line of $\overline{CB_j}$.

The corresponding points \tilde{A}_i , \tilde{B}_j , \tilde{C} and \tilde{D}_j of A_i , B_j , C and D_j on

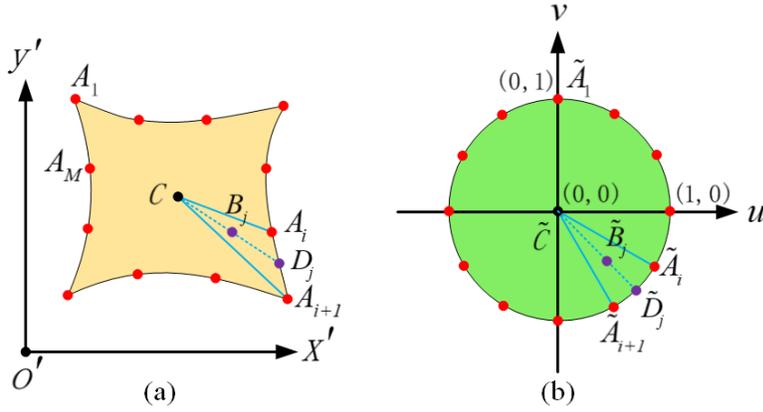


Figure 5.4: *The mapping relationship. (a) The $x'y'$ plane. (b) The uv plane.*

uv plane are shown in Fig. 5.4(b). All the points on the boundary curves can be mapped to the unit circle with the method below. Here, the point \tilde{D}_j is taken as an example to discuss how to determine its position on the unit circle.

The total arc length of the boundary curve is $L_{sum} = L_{A_1A_2} + \dots + L_{A_iA_{i+1}} + \dots + L_{A_MA_1}$. The arc length from the starting point A_1 to the point D_j is $L_{A_1D_j} = L_{A_1A_2} + L_{A_2A_3} + \dots + L_{A_iD_j}$. The perimeter of the unit circle is $L_{circle} = 2\pi$. And the arc length from the starting point \tilde{A}_1 to the point \tilde{D}_j is assumed to be $L_{\tilde{A}_1\tilde{D}_j}$. Since $L_{A_1D_j}/L_{sum}$ should be equal to $L_{\tilde{A}_1\tilde{D}_j}/(2\pi)$, $L_{\tilde{A}_1\tilde{D}_j} = 2\pi L_{A_1D_j}/L_{sum}$ is obtained.

Having determined the position of the point \tilde{D}_j on the circle, its parametric values $u_{\tilde{D}_j}$ and $v_{\tilde{D}_j}$ can be calculated. The parametric values of the point \tilde{C} are $u_{\tilde{C}} = 0$ and $v_{\tilde{C}} = 0$. The coordinate values x'_{D_j} and y'_{D_j} of the point D_j on the boundary curve are determined by the intersection of the line $\overline{A_iA_{i+1}}$ and the extended line of $\overline{CB_j}$, and the coordinate values x'_C and y'_C of the point C and the coordinate values x'_{B_j} and y'_{B_j} of the point B_j are known. Assuming the parametric values of the point \tilde{B}_j are $u_{\tilde{B}_j}$ and $v_{\tilde{B}_j}$, $(u_{\tilde{B}_j} - u_{\tilde{C}})/(u_{\tilde{D}_j} - u_{\tilde{C}}) = (x'_{B_j} - x'_C)/(x'_{D_j} - x'_C)$ and

$(v_{\tilde{B}_j} - v_{\tilde{C}})/(v_{\tilde{D}_j} - v_{\tilde{C}}) = (y'_{B_j} - y'_C)/(y'_{D_j} - y'_C)$ are obtained, which give

$$\begin{aligned} u_{\tilde{B}_j} &= u_{\tilde{D}_j} \frac{x'_{B_j} - x'_C}{x'_{D_j} - x'_C} \\ v_{\tilde{B}_j} &= v_{\tilde{D}_j} \frac{y'_{B_j} - y'_C}{y'_{D_j} - y'_C} \end{aligned} \quad (5.17)$$

Substituting Eq. (5.17) into (5.12), the deformation values of the original vertex of B_j in the global xyz coordinate system can be calculated, and finally a new surface shape can be obtained after solving all vertices within the boundary curve.

5.2.4 User interface

By integrating the above algorithms, a user interface is developed as a plug-in of the popular 3D modelling software Maya, as shown in Fig. 5.5. Users can select their desired deformation regions on the surface of a 3D model and generate new shapes using the interface in Maya.

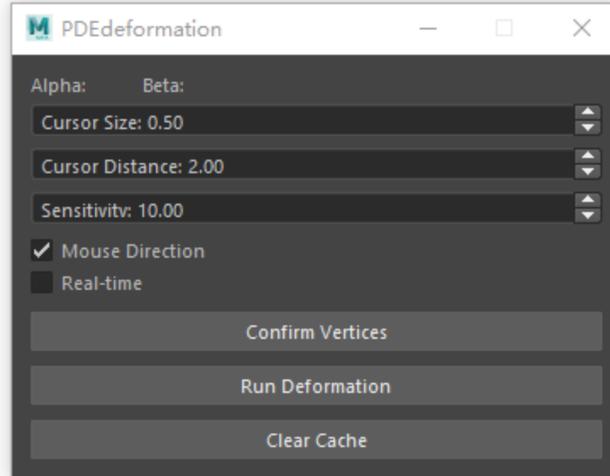


Figure 5.5: *The user interface.*

Since directly inputting the value of the sculpting force to manipulate surface shapes is not intuitive for users, a cursor (a visual cube) is applied to achieve deformations, as shown in Fig. 5.6. In order to control the deformation direction, only the moving direction of the cursor needs to

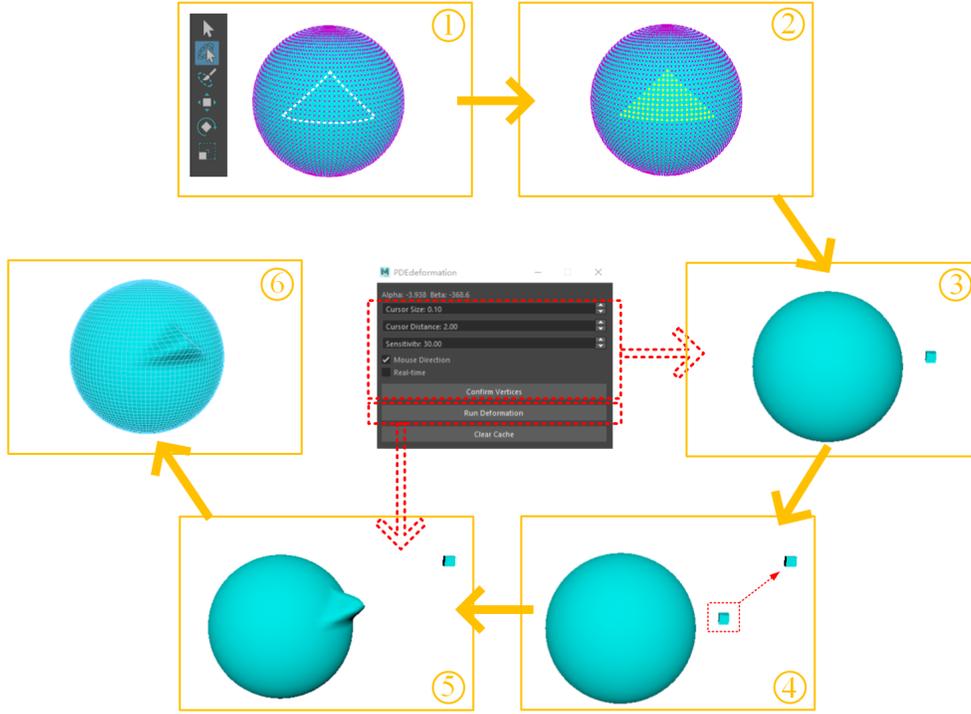


Figure 5.6: *The surface manipulation process of the user interface.*

be controlled. There are two ways to control the cursor. First, the cursor can be moved by using the move tool in Maya. Once the move tool is activated, the cursor can be directly dragged on the current view plane of the user. Second, the moving values in x , y and z directions can be input into the attributes editor of the cursor in Maya. In Fig. 5.5, there are three input parameters for the cursor on the interface, i. e., cursor size, cursor distance and sensitivity. Cursor size is used to control the size of the cursor and avoid that the cursor is much larger or smaller than the target object. Cursor distance is a parameter to change the initial position of the cursor, and sensitivity is a scale factor to control the magnitude of the sculpting force. The sculpting force in Eq. (5.12) can be obtained by $q_{\xi} = s d_{\xi}$ ($\xi = x, y, z$) where d_{ξ} is the moving vector of the cursor and s is the sensitivity. Fig. 5.7 illustrates the influence of the moving direction and sensitivity of the cursor on the surface shape.

Except the three input parameters, two options, i. e. Mouse Direction and Real-time, are also provided in the user interface. 1) Mouse Direction: its function is to make the deformation direction the same as the moving

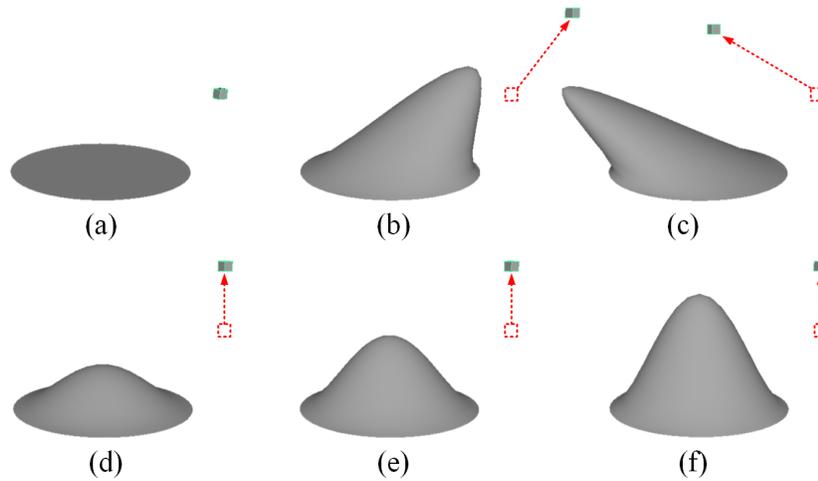


Figure 5.7: *The influence of the moving direction and sensitivity of the cursor on the surface shape. (a) The deformation region and the initial position of the cursor. (b)-(c) The deformed shapes with the same sensitivity and different moving directions (the red dotted arrow) of the cursor (the red dotted square is the initial position of the cursor). (d)-(f) The deformed shapes with the same moving direction and different sensitivities of the cursor. (The sensitivities are 40, 70 and 110 in (d), (e) and (f), respectively)*

direction of the cursor. If it is turned off, the deformation direction is the normal of the selected region. 2) Real-time: since the user's view cannot be moved or rotated during moving the cursor, sometimes users have difficulties to estimate whether the cursor arrives at the desired position. Thus, the option of Real-time is provided to turn off the real-time function so that the shape cannot be deformed in real time unless clicking the Run Deformation button.

As shown in Fig. 5.6, the surface manipulation process through the user interface has six steps. 1) Users select the deformation region by using the lasso tool to draw freeform curves around target vertices. 2) The selected vertices are activated with yellow colour. 3) After inputting three parameters of the cursor, users click the Confirm Vertices button to generate the cursor. 4) Users can move the cursor to any position and the deformation will follow the moving direction if the option of Mouse Direction is ticked. Otherwise, the deformation direction is in the normal direction. 5) If the option of real-time is ticked, the surface

shape is automatically deformed in the fourth step once the cursor moves. Otherwise, users need to click the Run Deformation button to deform the surface shape. 6) After obtaining the final new surface shape or in any previous steps, users can click the Clear Cache button to stop this manipulation process.

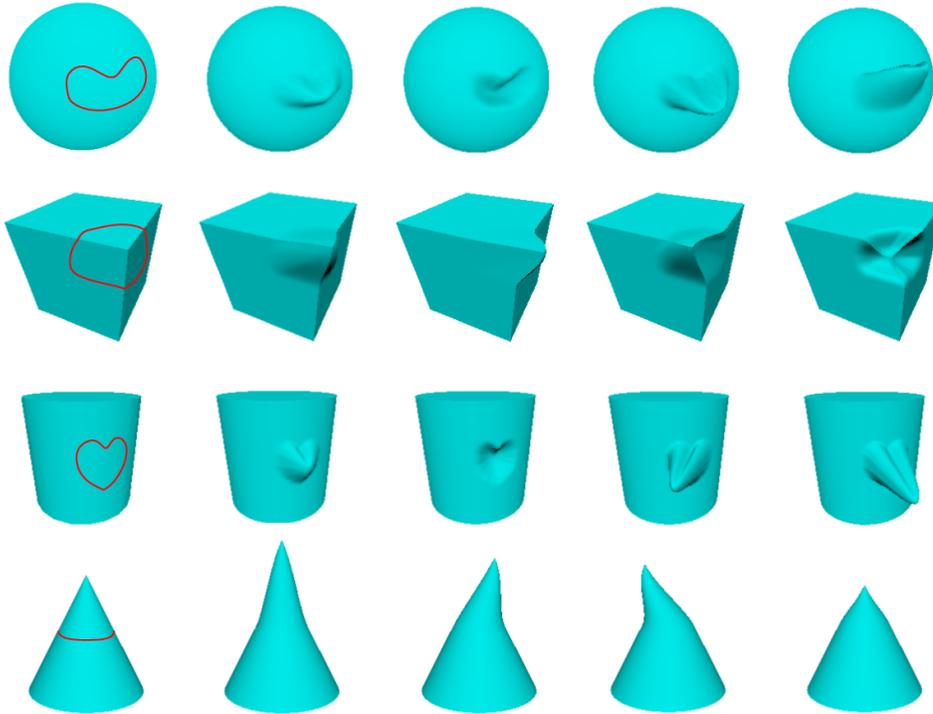


Figure 5.8: Test results of four basic 3D models with different deformation regions surrounded by red curves.

5.3 Results and comparison

In the user interface, the basic parameters of Eq. (5.3) are taken to be: the material properties $E = 10^6$ and $\nu = 0.5$ according to the material rubber, and the geometric thickness $h = 0.1$. The threshold value α_0 is set to 0.86 because this value is applicable for accurately extracting the boundary curve of different deformation regions by trial and error. The user interface has been tested on four basic 3D models in Maya, i. e., the polygon sphere, cube, cylinder and cone, with different deformation regions, as shown in Fig. 5.8. Different deformations on each model are

obtained by moving the cursor with different directions and positions to generate various shapes. These results indicate that the proposed method is effective and convenient to create various surface shapes.

The meshes of above basic 3D models are smooth and uniformly distributed. In order to further test the proposed method, scanned 3D models such as Stanford Bunny, Dragon and Nefertiti are used as the deformation objects. These scanned 3D models are frequently used as test models in the computer graphics community and have non-uniformly distributed vertices. Fig. 5.9 shows the test results. They indicate that the proposed method is also suitable to manipulate the surface shapes of scanned 3D models.

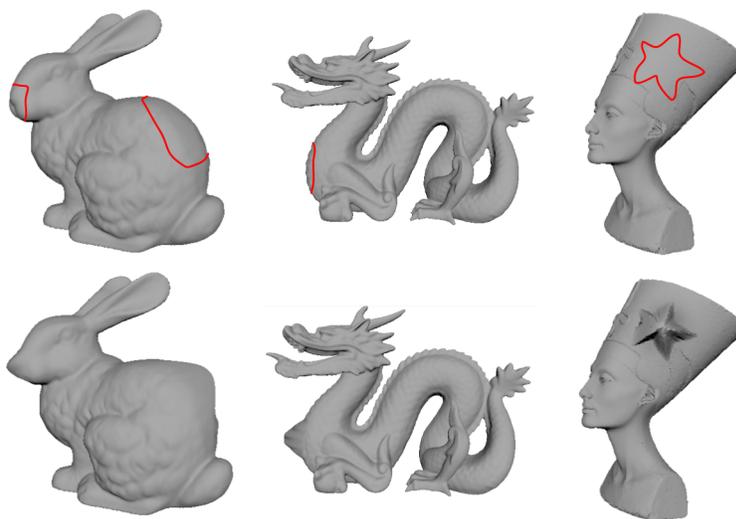


Figure 5.9: *Test results of three scanned 3D models with different deformation regions surrounded by red curves.*

In order to demonstrate the effectiveness of the proposed method, an experiment is made to compare it with commonly used deformation methods, i. e., Delta Mush [Mancewicz et al. 2014] and Laplacian Deformer [Sorkine et al. 2004]. Two examples of deforming the shapes of the square and the pentagram along the normal direction are given in Fig. 5.10. The first and third rows present deformed shapes and the second and fourth rows are contour maps of the deformed shapes in the first and third rows, respectively. As shown in Fig. 5.10, the deformed shape using the proposed method is strictly constrained by the boundary shape

compared with other methods. For example, the shapes of the contour lines with different heights of the square using the proposed method stay the same as the boundary, while contour lines using Delta Mush and Laplacian Deformer gradually become a circle. Therefore, compared with Delta Mush and Laplacian Deformer, the proposed method is better at creating the surface shape which well keeps the features of the boundary shape.

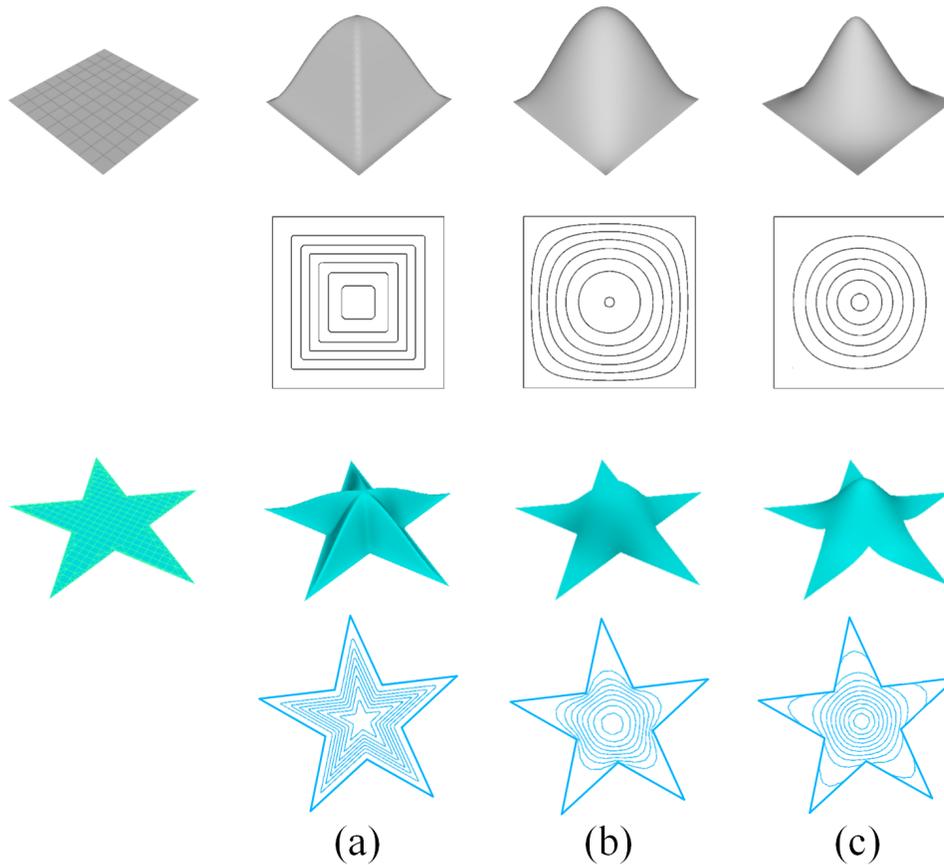


Figure 5.10: *The comparison between different deformation methods. (a) The proposed method. (b) Delta Mush [Mancewicz et al. 2014]. (c) Laplacian Deformer [Sorkine et al. 2004]. (The first and third rows are deformed shapes, and the second and fourth rows are contour maps)*

Since the proposed method is physics-based, it can create more realistic shapes compared with purely geometric methods. The finite element analysis (FEA) is the most accurate and popular numerical method widely applied in scientific research and engineering calculations. Especially, it has been widely used to accurately predict elastic and inelastic deforma-

tions of various objects and structures in engineering fields. Therefore, FEA is used to obtain deformed shapes as ground truth and compared them with the proposed method, Delta Mush and Laplacian Deformer in Fig. 5.11. The left and right columns in the figure present the side view of deformed shapes of a triangle and a rectangle, respectively. The results indicate that the deformed shapes using the proposed method are the closest to the ground truth ones and the most realistic among the three deformation methods used to obtain the deformed shapes shown in the figure. In addition, in order to manipulate the shape as similar as possible to the ground truth, Delta Mush needs to try different values of two parameters, i. e., the number of smoothing iterations and step lengths, which causes extra time, and Laplacian Deformer needs to find appropriate anchor vertices because different anchor vertices may lead to different shapes. In contrast, the proposed method can directly create realistic deformed shapes once the deformation region is selected.

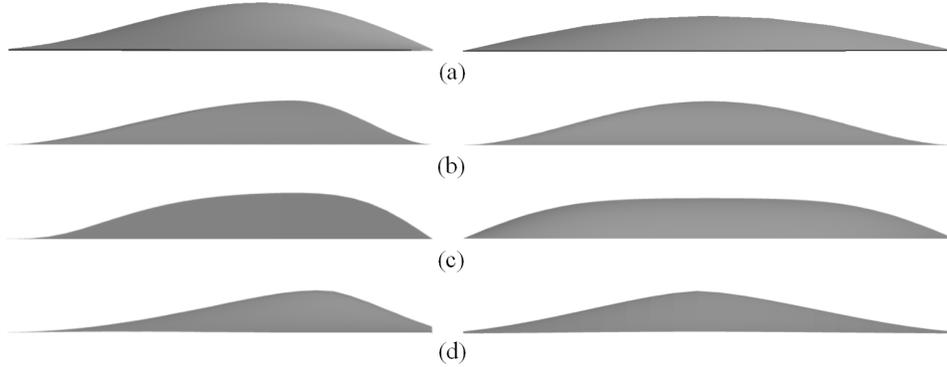


Figure 5.11: *The realism comparison of deformed shapes in the side view. (a) Ground truth. (b) The proposed method. (c) Delta Mush [Mancewicz et al. 2014]. (d) Laplacian Deformer [Sorkine et al. 2004]. (The left and right columns are deformed shapes of a triangle and a rectangle, respectively)*

5.4 Summary

In this chapter, a simple approximate analytical solution of the PDE-based mathematical model defining the underlying physics of surface deformations has been proposed to develop a physics-based deformation

method and used to implement a user interface as a plug-in of the 3D modelling software Maya to interactively manipulate surface shapes of 3D models with C^1 continuity. The validity of the proposed method has been demonstrated by testing different surface deformations on several 3D models. Compared with purely geometric methods, the proposed method is stricter to constrain the deformed shape according to boundary shapes and can generate more realistic shapes.

Chapter 6

Conclusion and future work

6.1 Conclusion

In the community of computer graphics, PDE-based surface modelling has attracted considerable attention in recent decades due to its excellent advantages such as small data, efficient shape deformation, good continuities and physics-based nature. In this thesis, the challenges of using numerical PDE surface patches in engineering applications, constructing and solving a unified PDE-based modelling framework and manipulating surface shapes using an analytical PDE have been tackled.

To use numerical PDE surface patches in engineering applications, the numerical solution of a fourth-order PDE with three control parameters has been presented by using the finite difference method to solve the 4-sided surface patches with C^1 continuity. Then, with the numerical PDE surface patch, an application for the optimal conversion of PDE surfaces representing high-speed train heads into NURBS surfaces has been developed. Through this conversion, the numerical 4-sided PDE surface patch can be used in CAD, CAM and CAE systems. With the same numerical solution, another application for the multi-objective aerodynamic optimization of high-speed train heads has been developed, in which the CFD simulation has been applied to optimize the PDE surface shape of the train head according to the aerodynamic performance, i. e., the aerodynamic drag and lift forces. The proposed method has given

the first pipeline of using the control parameters in PDE and patch boundaries to deform and optimize the shape of the model which consists of several numerical 4-sided PDE surface patches.

Since the numerical solution has a high computational cost and existing accurate and approximate analytical solutions cannot deal with creating complex surfaces on different occasions, the unified PDE mathematical model has been proposed. A novel composite form has been presented to solve the unified model for creating the 2-, 3- and 4-sided PDE surface patches with C^n continuity. With the analytical solution, an application of the surface reconstruction from scanned points has been proposed for automotive styling design, in which the optimal PDE surface patch with C^n continuity under the constraints of the two feature curves has been generated. Several experiments and comparisons with other methods have indicated the potential of the PDE technique in the styling design field. In addition, a user interface of the analytical solution of the unified PDE mathematical model has been developed as a plug-in in Blender. It can create 2-, 3- and 4-sided PDE surface patches from two, three and four 3D curves, respectively, and generate complete 3D models from input curve networks.

The unified PDE-based surface modelling framework can address 2-, 3- and 4-sided surface patches but cannot deal with arbitrary boundaries. In order to manipulate surface shape within arbitrary boundaries and achieve C^1 continuity, a novel but simple physics-based surface manipulation method with a fourth-order PDE involving a sculpting force has been proposed. A mapping method has been presented to map a deformation region from a 3D space to a 2D parametric plane, which can simplify the resolution of the PDE originating from the bending deformation of a thin elastic plate. An interactive user interface has been developed as a plug-in of Maya to achieve surface manipulation. The effectiveness, easiness and better realism of the proposed method have been demonstrated by testing surface deformations on several 3D models and comparing with other methods and ground-truth deformations.

Although this thesis has dealt with the main challenges of PDE-based

surface modelling which have been listed in Section 1.2, there are still some limitations. The most important one is the rule of effecting surface shape by changing the three control parameters in PDE. Although the control parameters have been used as design variables to optimize the surface shape in applications in Chapter 3 and 4, the geometric meaning of the control parameters are still unclear. Since the control parameters are mixed with boundary constraints in the solution process, it is very difficult to quantitatively analyze the influence of each control parameter on the solution result. Another limitation is a general issue of PDE-based surface modelling. Unlike NURBS or Bézier surfaces which are intuitive for users to deform surface shapes by only moving control points, PDE-based surface modelling needs to define the boundary conditions and it is hard to know how to change the conditions, such as tangents and curvatures, to obtain the desired shape in advance if the sculpting force (the right-hand side term) is not used to deform surface shapes.

6.2 Future work

There is some work in this thesis that can be further improved in the future.

- **Numerical solution at irregular grids** In chapter 3, the finite differential method has been adopted as the numerical solution for the 4-sided PDE surface patches. Since the finite difference method is typically defined on a regular grid, it cannot be used for the patches with irregular boundaries, such as the round and pentagram patches. Future work will investigate the finite difference method at irregular grids.
- **Higher order continuity** To construct the unified PDE-based surface modelling framework, the fourth-order PDE has been taken as the PDE mathematical model in Chapter 4. The continuity between adjacent surface patches is up to fourth-order, i. e., C^4 continuity. For a higher continuity requirement, a higher-order PDE such as a PDE mathematical model based on a sixth-order PDE [Zhang & You 2004] will be investigated.

- **Irregular boundary condition** The unified PDE-based surface modelling framework is developed to create 2-, 3- and 4-sided PDE surface patches. n -sided PDE surface patches ($n > 4$) such as 5- and 6-sided PDE surface patches or other irregular boundaries have not been investigated in this thesis. n -sided patches can be decomposed into 3- and 4- sided patches. For an irregular boundary, one solution is to resample it into a unified 4-sided boundary and solve it with the solution of a 4-sided PDE surface patch. However, this process will involve a big error. In Chapter 5, the proposed surface manipulation is used to deform a local shape within an arbitrary irregular boundary by mapping the boundary in the 3D space into a unit circle on the 2D plane. This mapping method can also be applied to create surface patches with irregular boundaries, which will be investigated in the future.
- **Surface manipulation** In Chapter 5, since the proposed method needs to project the surface from a 3D space to a 2D plane, the surface can only be manipulated without overlapping parts. Simple overlapping parts within the deformation region can be avoided by changing the user view before confirming vertices. Besides, the proposed method cannot address the deformation region if its centroid is located outside of its boundaries. One way to solve this limitation is to divide the deformation region into several sub-regions and then construct the mapping relationship of each sub-region independently, which requires further investigation.

Bibliography

- N. Ahmat, et al. (2014). ‘Automatic shape optimisation of pharmaceutical tablets using Partial Differential Equations’. *Computers & Structures* **130**:1–9.
- M. Alexa (2003). ‘Differential coordinates for local mesh morphing and deformation’. *The Visual Computer* **19**(2):105–114.
- I. ANSYS (2009). ‘ANSYS fluent 12.0 User’s Guide’. *New Hampshire: ANSYS INC*.
- N. Aspert, et al. (2002). ‘Mesh: Measuring errors between surfaces using the hausdorff distance’. In *Proceedings. IEEE international conference on multimedia and expo*, vol. 1, pp. 705–708. IEEE.
- M. Athanasopoulos, et al. (2009). ‘Parametric design of aircraft geometry using partial differential equations’. *Advances in Engineering Software* **40**(7):479–486.
- A. H. Barr (1987). ‘Global and local deformations of solid primitives’. In *Readings in Computer Vision*, pp. 661–670. Elsevier.
- R. Bensow, et al. (2006). ‘A comparative study of RANS, DES and LES’. In *Twenty-sixth Symposium on Naval Hydrodynamics*.
- M. Bloor & M. Wilson (1989a). ‘Blend design as a boundary-value problem’. In *Theory and practice of geometric modeling*, pp. 221–234. Springer.
- M. Bloor & M. Wilson (1989b). ‘Generating blend surfaces using partial differential equations’. *Computer-Aided Design* **21**(3):165–171.

- M. I. Bloor & M. J. Wilson (1990). ‘Using partial differential equations to generate free-form surfaces’. *Computer-Aided Design* **22**(4):202–212.
- M. I. Bloor & M. J. Wilson (1993). ‘Functionality in solids obtained from partial differential equations’. In *Geometric modelling*, pp. 21–42. Springer.
- M. I. Bloor & M. J. Wilson (2005). ‘An analytic pseudo-spectral method to generate a regular 4-sided PDE surface patch’. *Computer Aided Geometric Design* **22**(3):203–219.
- M. I. Bloor, et al. (1995). ‘The smoothing properties of variational schemes for surface design’. *Computer Aided Geometric Design* **12**(4):381–394.
- U. Bodenhofer (2003). ‘Genetic algorithms: theory and applications’. *Lecture notes, Fuzzy Logic Laboratorium Linz-Hagenberg, Winter 2004*.
- M. Botsch & O. Sorkine (2008). ‘On linear variational surface deformation methods’. *IEEE transactions on visualization and computer graphics* **14**(1):213–230.
- N. Brockie & C. Baker (1990). ‘The aerodynamic drag of high speed trains’. *Journal of Wind Engineering and Industrial Aerodynamics* **34**(3):273–290.
- J. M. Brown, et al. (1998). ‘The accuracy of B-spline finite element approximations to PDE surfaces’. *Computer methods in applied mechanics and engineering* **158**(3-4):221–234.
- D. Brujic, et al. (2002). ‘Measurement-based modification of NURBS surfaces’. *Computer-Aided Design* **34**(3):173–183.
- G. G. Castro, et al. (2010). ‘Cyclic animation using partial differential equations’. *The Visual Computer* **26**(5):325–338.
- G. G. Castro, et al. (2008). ‘A survey of partial differential equations in geometric design’. *The Visual Computer* **24**(3):213–225.
- T. Cebeci, et al. (2005). *Computational fluid dynamics for engineers*. Springer Berlin Heidelberg.

- G. Celniker & D. Gossard (1991). ‘Deformable curve and surface finite-elements for free-form shape design’. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pp. 257–266.
- K.-H. Chang (2014). *Product design modeling using CAD/CAE: the computer aided engineering design series*. Academic Press.
- F. Cheli, et al. (2010). ‘Aerodynamic behaviour investigation of the new EMUV250 train to cross wind’. *Journal of Wind Engineering and Industrial Aerodynamics* **98**(4-5):189–201.
- C. Chen, et al. (2017). ‘A PDE-based head visualization method with CT data’. *Computer Animation and Virtual Worlds* **28**(1):e1683.
- P. Cignoni, et al. (1998). ‘Metro: measuring error on simplified surfaces’. In *Computer graphics forum*, vol. 17, pp. 167–174. Wiley Online Library.
- S. Coquillart (1990). ‘Extended free-form deformation: A sculpturing tool for 3D geometric modeling’. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pp. 187–196.
- J. Dan & W. Lancheng (2006). ‘An algorithm of NURBS surface fitting for reverse engineering’. *The International Journal of Advanced Manufacturing Technology* **31**(1-2):92–97.
- W. De Wolf & E. Demmenie (1997). ‘A new test facility for the study of interacting pressure waves and their reduction in tunnels for high speed trains’ .
- K. Deb (1999). ‘Multi-objective genetic algorithms: Problem difficulties and construction of test problems’. *Evolutionary computation* **7**(3):205–230.
- K. Deb, et al. (2002). ‘A fast and elitist multiobjective genetic algorithm: NSGA-II’. *IEEE transactions on evolutionary computation* **6**(2):182–197.
- M. Desbrun, et al. (1999). ‘Implicit fairing of irregular meshes using diffu-

- sion and curvature flow’. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 317–324.
- H. Du & H. Qin (2005a). ‘Dynamic PDE-based surface design using geometric and physical constraints’. *Graphical Models* **67**(1):43–71.
- H. Du & H. Qin (2005b). ‘A shape design system using volumetric implicit PDEs’. In *ACM SIGGRAPH 2005 Courses*, p. 116. ACM.
- H. Edelsbrunner, et al. (1983). ‘On the shape of a set of points in the plane’. *IEEE Transactions on information theory* **29**(4):551–559.
- M. Epstein (2017). ‘Partial Differential Equations in Engineering’. In *Partial Differential Equations*, pp. 25–47. Springer.
- C. Fong (2015). ‘Analytical methods for squaring the disc’. *arXiv preprint arXiv:1509.06344* .
- Ford Media (2013). ‘Ford’s Language of Tape’. <https://youtu.be/xnnIVvQt5Ec>. Accessed 20 April 2020.
- A. Gálvez, et al. (2008). ‘Particle swarm optimization for Bézier surface reconstruction’. In *International Conference on Computational Science*, pp. 116–125. Springer.
- A. Gálvez & A. Iglesias (2012). ‘Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points’. *Information Sciences* **192**:174–192.
- A. Gálvez, et al. (2012). ‘Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction’. *Information Sciences* **182**(1):56–76.
- D. E. Goldberg (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York.
- U. G & B.  (1994). ‘Animation of deformable models’. *Computer-Aided Design* **26**(12):868–875.
- D. Haixia, et al. (2008). ‘Pde-based medial axis extraction and shape manipulation of arbitrary meshes’. *Journal of Systems Science and Complexity* **21**(4):609.

- Y. He & H. Qin (2004). ‘Surface reconstruction with triangular B-splines’. In *Geometric Modeling and Processing, 2004. Proceedings*, pp. 279–287. IEEE.
- W. Heidrich, et al. (1996). ‘Fitting uncertain data with NURBS’. In *Proc. 3rd Int. Conf. on Curves and Surfaces in Geometric Design*, pp. 1–8. Citeseer.
- G. Hirota, et al. (2000). ‘Fast volume-preserving free-form deformation using multi-level optimization’. *Computer-Aided Design* **32**(8):499–512.
- M. Hosaka (2012). *Modeling of Curves and Surfaces in CAD/CAM*. Springer Science & Business Media.
- W. M. Hsu, et al. (1992). ‘Direct manipulation of free-form deformations’. *ACM Siggraph Computer Graphics* **26**(2):177–184.
- ISO (1997). ‘Information technology — Computer graphics and image processing — Programmer’s Hierarchical Interactive Graphics System (PHIGS) — Part 1: Functional description’. ISO ISO/IEC 9592-1:1997, International Organization for Standardization, Geneva, Switzerland.
- ISO (2003). ‘Industrial automation systems and integration – product data representation and exchange – part 42: integrated generic resource: geometric and topological representation’. ISO ISO 10303-42:2003, International Organization for Standardization, Geneva, Switzerland.
- A. Jacobson, et al. (2010). ‘Mixed finite elements for variational surface modeling’. In *Computer graphics forum*, vol. 29, pp. 1565–1574. Wiley Online Library.
- Jaguar (2017). ‘Jaguar XE SV Project 8 — SVO Clay Modelling Studio’. <https://youtu.be/g8VdXEbhhBg>. Accessed 06 August 2020.
- Z. Jing, et al. (2009). ‘Optimized NURBS curve and surface fitting using simulated annealing’. In *2009 Second International Symposium on Computational Intelligence and Design*, vol. 2, pp. 324–329. IEEE.
- P. Kalra, et al. (1992). ‘Simulation of facial muscle actions based on rational free form deformations’. In *Computer Graphics Forum*, vol. 11, pp. 59–69. Wiley Online Library.

- P. Kennicott (1996). ‘IGES/PDES organization, US product data association’. *Initial graphics exchange specification, IGES5* **3**.
- J.-P. Kruth & A. Kerstens (1998). ‘Reverse engineering modelling of free-form surfaces from point clouds subject to boundary conditions’. *Journal of Materials Processing Technology* **76**(1-3):120–127.
- Y.-C. Ku, et al. (2010). ‘Optimal cross-sectional area distribution of a high-speed train nose to minimize the tunnel micro-pressure wave’. *Structural and Multidisciplinary Optimization* **42**(6):965–976.
- S. Kubiesa, et al. (2004). ‘Interactive design using higher order PDEs’. *The Visual Computer* **20**(10):682–693.
- H.-b. Kwon, et al. (2001). ‘Nose shape optimization of high-speed train for minimization of tunnel sonic boom’. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing* **44**(3):890–899.
- H. J. Lamousin & N. Waggenspack (1994). ‘NURBS-based free-form deformations’. *IEEE Computer Graphics and Applications* **14**(6):59–65.
- J. Lee & J. Kim (2008). ‘Approximate optimization of high-speed train nose shape for reducing micropressure wave’. *Structural and Multidisciplinary Optimization* **35**(1):79–87.
- R. Li, et al. (2016). ‘Multi-objective optimization of a high-speed train head based on the FFD method’. *Journal of Wind Engineering and Industrial Aerodynamics* **152**:41–49.
- R. Li, et al. (2020). ‘Optimization of the high-speed train head using the radial basis function morphing method’. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* **234**(1):96–107.
- A. Limaïem, et al. (1996). ‘Data fitting using dual kriging and genetic algorithms’. *CIRP annals* **45**(1):129–134.
- C.-Y. Lin, et al. (1997). ‘A surface-lofting approach for smooth-surface reconstruction from 3D measurement data’. *Computers in Industry* **34**(1):73–85.

- Y. Lipman, et al. (2004). ‘Differential coordinates for interactive mesh editing’. In *Proceedings Shape Modeling Applications, 2004.*, pp. 181–190. IEEE.
- D. Liu & G. Xu (2007). ‘A general sixth order geometric partial differential equation and its application in surface modeling’. *Journal of Information and Computational Science* **4**(1):129–140.
- W. Ma & J.-P. Kruth (1995). ‘Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces’. *Computer-Aided Design* **27**(9):663–675.
- W. Ma & J.-P. Kruth (1998). ‘NURBS curve and surface fitting for reverse engineering’. *The International Journal of Advanced Manufacturing Technology* **14**(12):918–927.
- T. Maeda, et al. (1989). ‘Aerodynamic drag of Shinkansen electric cars (series 0, series 200, series 100)’. *Railway Technical Research Institute, Quarterly Reports* **30**(1).
- J. Mancewicz, et al. (2014). ‘Delta mush: smoothing deformations while preserving detail’. In *Proceedings of the Fourth Symposium on Digital Production*, pp. 7–11.
- K. T. McDonnell & H. Qin (2007). ‘A novel framework for physically based sculpting and animation of free-form solids’. *The Visual Computer* **23**(4):285–296.
- M. D. McKay, et al. (2000). ‘A comparison of three methods for selecting values of input variables in the analysis of output from a computer code’. *Technometrics* **42**(1):55–61.
- J. A. Morden, et al. (2015). ‘Comparison of RANS and detached eddy simulation results to wind-tunnel data for the surface pressures upon a class 43 high-speed train’. *Journal of Fluids Engineering* **137**(4).
- J. Muñoz-Paniagua, et al. (2014). ‘Genetically aerodynamic optimization of the nose shape of a high-speed train entering a tunnel’. *Journal of wind engineering and industrial aerodynamics* **130**:48–61.
- A. Nealen, et al. (2006). ‘Physically based deformable models in computer

- graphics’. In *Computer graphics forum*, vol. 25, pp. 809–836. Wiley Online Library.
- Q. Pan, et al. (2014). ‘A unified method for hybrid subdivision surface design using geometric partial differential equations’. *Computer-Aided Design* **46**:110–119.
- I. K. Park, et al. (1999). ‘Constructing NURBS surface model from scattered and unorganized range data’. In *Second International Conference on 3-D Digital Imaging and Modeling (Cat. No. PR00062)*, pp. 312–320. IEEE.
- J. Park, et al. (2015). ‘An algorithm for estimating surface normal from its boundary curves’. *Journal of Computational Design and Engineering* **2**(1):67–72.
- L. Piegl (1991). ‘On NURBS: a survey’. *IEEE Computer Graphics and Applications* **11**(01):55–71.
- L. Piegl & W. Tiller (2012). *The NURBS book*. Springer Science & Business Media.
- PRNewswire (2019a). ‘3D mapping & modeling in 2019: insights into & future of the market to 2024’. <https://www.prnewswire.com/news-releases/3d-mapping--modeling-in-2019-insights-into-future-of-the-market-to-2024-300879291.html>. Accessed: 2021-08-09.
- PRNewswire (2019b). ‘Global and China animation industry report, 2019-2025’. <https://www.prnewswire.com/news-releases/global-and-china-animation-industry-report-2019-2025-300880157.html>. Accessed: 2021-08-09.
- H. Qin & D. Terzopoulos (1995). ‘Dynamic NURBS swung surfaces for physics-based shape design’. *Computer-Aided Design* **27**(2):111–127.
- H. Qin & D. Terzopoulos (1997). ‘Triangular NURBS and their dynamic generalizations’. *Computer Aided Geometric Design* **14**(4):325–347.
- D. Saini & S. Kumar (2014). ‘Free-form surface reconstruction from

- arbitrary perspective images’. In *2014 IEEE International Advance Computing Conference (IACC)*, pp. 1054–1059. IEEE.
- M. Sarfraz (2004). ‘Representing shapes by fitting data using an evolutionary approach’. *Computer-Aided Design and Applications* **1**(1-4):179–185.
- J. A. Schetz (2001). ‘Aerodynamics of high-speed trains’. *Annual Review of Fluid Mechanics* **33**(1):371–414.
- R. Schneider & L. Kobbelt (2001). ‘Geometric fairing of irregular meshes for free-form surface design’. *Computer aided geometric design* **18**(4):359–379.
- T. W. Sederberg & S. R. Parry (1986). ‘Free-form deformation of solid geometric models’. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pp. 151–160.
- A. Sheffer, et al. (2006). ‘Mesh parameterization methods and their applications’. *Foundations and Trends® in Computer Graphics and Vision* **2**(2):105–171.
- Q. Shen, et al. (2018). ‘A PDE patch-based spectral method for progressive mesh compression and mesh denoising’. *The Visual Computer* **34**(11):1563–1577.
- Y. Sheng, et al. (2010). ‘A PDE method for patchwise approximation of large polygon meshes’. *The Visual Computer* **26**(6):975–984.
- Y. Sheng, et al. (2011). ‘Facial geometry parameterisation based on partial differential equations’. *Mathematical and Computer Modelling* **54**(5-6):1536–1548.
- W. Song & X. Yang (2005). ‘Free-form deformation with weighted T-spline’. *The Visual Computer* **21**(3):139–151.
- O. Sorkine, et al. (2004). ‘Laplacian surface editing’. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 175–184.

- T. Stanko, et al. (2016). ‘Surfacing curve networks with normal control’. *Computers & Graphics* **60**:1–8.
- Z. Sun, et al. (2010). ‘Optimization of the head shape of the CRH3 high speed train’. *Science China Technological Sciences* **53**(12):3356–3364.
- M. Suzuki & K. Nakade (2013). ‘Multi-objective design optimization of high-speed train nose’. *Journal of Mechanical Systems for Transportation and Logistics* **6**(1):54–64.
- D. Terzopoulos & K. Fleischer (1988a). ‘Deformable models’. *The visual computer* **4**(6):306–331.
- D. Terzopoulos & K. Fleischer (1988b). ‘Modeling inelastic deformation: viscoelasticity, plasticity, fracture’. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pp. 269–278.
- D. Terzopoulos, et al. (1987). ‘Elastically deformable models’. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 205–214.
- D. Terzopoulos & H. Qin (1994). ‘Dynamic NURBS with geometric constraints for interactive sculpting’. *ACM Transactions on Graphics (TOG)* **13**(2):103–136.
- S. P. Timoshenko & S. Woinowsky-Krieger (1959). *Theory of plates and shells*. McGraw-hill.
- M. Tovey (1997). ‘Styling and design: intuition and analysis in industrial design’. *Design studies* **18**(1):5–31.
- S. Tsuchie (2017). ‘Reconstruction of underlying surfaces from scanned data using lines of curvature’. *Computers & Graphics* **68**:108–118.
- S. Tsuchie (2019). ‘Reconstruction of intersecting surface models from scanned data for styling design’. *Engineering with Computers* pp. 1–12.
- W.-D. Ueng, et al. (1998). ‘Sweep-surface reconstruction from three-dimensional measured data’. *Computer-aided design* **30**(10):791–805.

- H. Ugail, et al. (1999). ‘Techniques for interactive design using the PDE method’. *ACM Transactions on Graphics (TOG)* **18**(2):195–212.
- E. Ulker (2012). ‘NURBS curve fitting using artificial immune system’. *Int J Innov Comput Inf Control* **8**(4):2875–2887.
- T. I. Vassilev (1997). ‘Interactive Sculpting with Deformable Nonuniform B-splines’. In *Computer Graphics Forum*, vol. 16, pp. 191–199. Wiley Online Library.
- K. J. Versprille (1975). *Computer-aided design applications of the rational b-spline approximation form*. Syracuse University.
- R. Wang, et al. (2018). ‘A survey of parametric modelling methods for designing the head of a high-speed train’. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* **232**(7):1965–1983.
- S. Wang, et al. (2017). ‘The performance of different turbulence models (URANS, SAS and DES) for predicting high-speed train slipstream’. *Journal of Wind Engineering and Industrial Aerodynamics* **165**:46–57.
- Y. J. Xiao & Y. Li (2005). ‘Optimized stereo reconstruction of free-form space curves based on a nonuniform rational B-spline model’. *JOSA A* **22**(9):1746–1762.
- S. Yao, et al. (2016). ‘Parametric design and optimization of high speed train nose’. *Optimization and Engineering* **17**(3):605–630.
- S. Yao, et al. (2015). ‘A modified multi-objective sorting particle swarm optimization and its application to the design of the nose shape of a high-speed train’. *Engineering Applications of Computational Fluid Mechanics* **9**(1):513–527.
- S. Yao, et al. (2014). ‘Optimization design for aerodynamic elements of high speed trains’. *Computers & Fluids* **95**:56–73.
- Z. Yin (2004). ‘Reverse engineering of a NURBS surface from digitized points subject to boundary conditions’. *Computers & Graphics* **28**(2):207–212.

- F. Yoshimoto, et al. (1999). ‘Automatic knot placement by a genetic algorithm for data fitting with a spline’. In *Proceedings Shape Modeling International’99. International Conference on Shape Modeling and Applications*, pp. 162–169. IEEE.
- L. You, et al. (2004a). ‘PDE blending surfaces with C2 continuity’. *Computers & Graphics* **28**(6):895–906.
- L. You, et al. (2006). ‘Manipulation of elastically deformable surfaces through maya plug-in’. In *Geometric Modeling and Imaging–New Trends (GMAI’06)*, pp. 15–21. IEEE.
- L. You, et al. (2009). ‘Adaptive physics–inspired facial animation’. In *International Workshop on Motion in Games*, pp. 207–218. Springer.
- L. You, et al. (2020). ‘Fast character modeling with sketch-based PDE surfaces’. *Multimedia Tools and Applications* **79**:23161–23187.
- L. You & J. J. Zhang (2003). ‘Fast generation of 3-D deformable moving surfaces’. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **33**(4):616–625.
- L. You, et al. (2004b). ‘Blending surface generation using a fast and accurate analytical solution of a fourth-order PDE with three shape control parameters’. *The Visual Computer* **20**(2):199–214.
- M.-g. Yu, et al. (2013). ‘Multi-objective optimization design method of the high-speed train head’. *Journal of Zhejiang University SCIENCE A* **14**(9):631–641.
- J. J. Zhang & L. You (2002). ‘PDE based surface representation-vase design’. *Computers & Graphics* **26**(1):89–98.
- J. J. Zhang & L. You (2004). ‘Fast surface modelling using a 6th order PDE’. In *Computer Graphics Forum*, vol. 23, pp. 311–320.
- L. Zhang, et al. (2018). ‘A multiobjective aerodynamic optimization design of a high-speed train head under crosswinds’. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* **232**(3):895–912.

- Y. Zhang, et al. (2020). ‘Proxy-driven free-form deformation by topology-adjustable control lattice’. *Computers & Graphics* **89**:167–177.
- J. Zheng, et al. (2003). ‘Constrained deformation of freeform surfaces using surface features for interactive design’. *The International Journal of Advanced Manufacturing Technology* **22**(1):54–67.
- K. Zhou, et al. (2005). ‘Large mesh deformation using the volumetric graph laplacian’. In *ACM SIGGRAPH 2005 Papers*, pp. 496–503.