Differential Equation-based Shape Interpolation for surface blending and facial blendshapes

Xiangyu You

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

Doctor of Philosophy

Department of Creative Technology
Faculty of Science & Technology
Bournemouth University
Poole, BH12 5BB
United Kingdom

Copyright

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. The materials presented in Section 3.3 have been published by You et al. (2018). The materials presented in Section 3.4 have been published by You et al. (2019a). The materials presented in Chapter 4 have been published by You et al. (2019b). These publications are listed in Subsection 1.6.

Acknowledgements

I would like to first thank my respected supervisors Professor Feng Tian and Professor Wen Tang for giving me their invaluable advice, time and kind support throughout the duration of my PhD study who have helped me to solve various problems and kept me on good progress. It is my great fortunate in my life to become one of their PhD students. Completing my PhD research is inseparable from their tireless instructions and help. I would also like to thank Bournemouth University, Faculty of Science and Technology, and Department of Creative Technology for providing me with a great research environment, and faculty for all their supports to my PhD research.

I greatly appreciate my company's support which enables me to complete my part-time PhD study. Last but not least, I would like to thank my wife for her love, encouragement and strong supports and my children who have made my life enjoyable.

The mesh data used in Figures 32, 35, 36, 38, and 40 of this thesis were made available by Robert Sumner and Jovan Popovic from the Computer Graphics Group at MIT.

Abstract

Differential equation-based shape interpolation has been widely applied in geometric modelling and computer animation. It has the advantages of physics-based, good realism, easy obtaining of high-order continuity, strong ability in describing complicated shapes, and small data of geometric models. Among various applications of differential equation-based shape interpolation, surface blending and facial blendshapes are two active and important topics.

Differential equation-based surface blending can be time-independent and time-dependent. Existing differential equation-based surface blending only tackles time-dependent C^1 continuous surface blending. In many applications, time-dependent C^2 continuous surface blending is required. Another problem is that time-dependent surface blending and time-independent surface blending are investigated separately in existing work, which requires more effort to implement and is not convenient to use. Integrating them together will save implementation effort and time and facilitate their applications.

Differential equation-based facial blendshapes are physics-based, which have the advantages of following underlying physics and generating more different blended facial shapes than those based on geometric interpolation algorithms. Unfortunately, existing physics-based facial blendshapes and facial animation are based on various numerical methods. These numerical methods are extremely difficult and time-consuming to setup since they require special knowledge and skills, additional mesh pre-processing, large computing capacity, and high computational costs and are not easy to learn, implement and use.

This thesis will investigate new techniques of C^2 continuous blending of constant and varying parametric surfaces and facial blendshapes with differential equation-based shape interpolation. For differential equation-based surface blending, three approximate analytical approaches called closed form solution-based approximate analytical approach, variable decomposition-based approximate

analytical approach, and differential property-based approximate analytical approach will be developed to achieve C^2 continuous blending of constant and varying parametric surfaces. For differential equation-based shape interpolation, an efficient physics-based facial blendshape technique will be developed, which is based on a simple and efficient closed form analytical solution to the equation of motion, which considers inertial effects, damping effects and the resistance against deformations. A blending force-based framework consisting of slider force-based, exponentiation force-based and random force-based methods will be developed to achieve highly efficient facial blendshapes.

List of Contents

List of Figures

List of Tables

1. Introduction

Computer graphics is a branch of computer science. Modelling, rendering and animation of geometric objects are three most important elements of computer graphics.

Geometric modelling plays a very important role in computer graphics, and it can be divided into curve modelling, surface modelling and volume modelling. Among them, surface modelling is most widely applied. Currently, surface modelling can be achieved by polygon modelling, spline modelling and subdivision.

- Polygonal modelling (Russo, 2005) is an approach for modelling objects by representing or approximating their surfaces using polygon meshes. The basic element used in polygon meshes is a vertex which is a point in a three-dimensional space. Two vertices connected by a straight line become an edge. Three vertices, connected to each other by three edges, define a triangle, which is the simplest polygon in the Euclidean space. More complex polygons can be created out of multiple triangles, or as a single object with more than 3 vertices. Four sided polygons (generally referred to as quads) and triangles are the most common shapes used in polygonal modelling. Polygon modelling uses some straight-line segments to approximate a curve and flat facets to approximate a curved surface.

- Spline modelling (Piegl and Tiller, 2012) creates a complex curve by stitching some curved segments and a complex surface by stitching some surface patches. Among various spline modelling approaches, Non-uniform rational B-spline (NURBS) is the most popular and has been implemented into various computer graphics software packages and computer-aided design software packages.

- Subdivision modelling (DeRose et al., 1998) generates a curve by repeatedly refining a polyline and a 3D surface by refining a coarse polygon mesh into a denser mesh through approximating or interpolating schemes. A subdivision surface, in the field of 3D computer graphics, represents a smooth surface via the specification of a coarser piecewise linear polygon mesh. The

smooth surface can be calculated from the coarse mesh as the limit of a recursive process of subdividing each polygonal face into smaller faces that better approximate the smooth surface. Subdivision surfaces are defined recursively. The process starts with a given polygonal mesh. A refinement scheme is then applied to this mesh. This process takes that mesh and subdivides it to create new vertices and new faces. The positions of the new vertices in the mesh are computed based on the positions of nearby old vertices. In some refinement schemes, the positions of old vertices might also be altered (possibly based on the positions of new vertices) as stated by Yu et al. (2010). This process produces a finer mesh than the original one, containing more polygonal faces. This resulting mesh can be passed through the same refinement scheme again and so on. The limit subdivision surface is the surface produced from this process being iteratively applied infinitely many times. In practical use however, this algorithm is only applied for a limited number of times. Subdivision surface refinement schemes can be broadly classified into two categories: interpolating and approximating. Interpolating schemes are required to match the original position of vertices in the original mesh. Approximating schemes are not; they can and will adjust these positions as needed. In general, approximating schemes have greater smoothness, but editing applications that allow users to set exact surface constraints require an optimization step.

All the three surface modelling methods create geometric models from scratch. After geometric models have been created, shape interpolation can generate more geometric models from known geometric models very efficiently. It generates transition surfaces or new shapes by interpolating existing ones.

The application of shape interpolation in geometric modelling and computer animation is massive (Bajaj and Ihm, 1992; Lewis et al., 2014). Blending surfaces and morphing between two different geometric models are two examples of the applications in geometric models. Facial blendshapes and keyframe animation are two examples of the applications in computer animation.

Shape interpolation can be divided into two types: one is to interpolate the boundaries of existing surfaces, and the other is to interpolate existing shapes directly. The transition surfaces created by the former is called blending surfaces, and the new shapes generated by the latter is called blend shapes. As the link[1] states, "facial expressions can communicate complex combinations of emotion information via multiplexed facial signals". Facial blendshapes are very useful in generating different facial expressions. This research will introduce differential equations to develop new approaches of surface blending and facial blendshapes.

1.1 Background

Surface blending and facial blendshapes are very important in geometric modeling and computer animation. Surface blending has a wide range of applications in computer-aided design. Facial blendshapes are widely applied in creative industries especially in computer animation and games.

Surface blending is to generate a smooth transition between intersecting surfaces or a smooth connection between disjoint surfaces. The surfaces to be blended are called primary surfaces. The surface which forms a smooth transition or connection between primary surfaces is called a blending surface. The interface curves between a blending surface and primary surfaces are called trimlines (Vida et al., 1994).

Surface blending can be divided into different types. The main types of blends met in practice can be categorized as: surfaces governed by strong functional constraints, aesthetic blends, fairings, and rounds and fillets (Rossignac and Requicha, 1984).

Surfaces used in geometric modelling can be grouped into explicit, implicit, and parametric surfaces. They can be mathematically represented with $z = f(x, y)$, $f(x, y, z) = 0$, and $x = x(u, v)$, $y = y(u, v)$, and $z = z(u, v)$, respectively. Accordingly, surface blending can be grouped into

[1] https://www.sciencealert.com/our-facial-expressions-can-portray-a-broader-range-of-emotions-than-you-might-think

explicit blending, implicit blending (Rockwood, 1989), and parametric blending (Filip, 1989). Since parametric surfaces such as NURBS surfaces are most popular, this research will concentrate on blending of parametric surfaces.

Continuity can be divided into geometric continuity and parametric continuity. Geometric continuity can be further divided into tangent continuity, curvature continuity and higher order geometric continuity. Tangent continuity is G^1 continuity. It requires two surfaces to have a same tangent direction at their shared interface. Curvature continuity is G^2 continuity. It requires two surfaces to have a same curvature at their shared interface. Parametric continuity can be divided into C^1 continuity, C^2 continuity, and higher order parametric continuity. C^1 continuity requires two surfaces to have the same position function and first partial derivative function at their shared interface. It is more stringent than G^1 continuity. C^2 continuity requires two surfaces to have the same position function, first partial derivative function, and the second partial derivative function at their shared interface. C^2 continuity is more stringent than G^2 continuity.

Among position, tangent, curvature, and higher order continuities, the tangent and curvature continuities are most frequently applied in many fields. For example, discontinuous curvature causes problems in numerical control (NC) milling and leads to break points of reflection lines (Aumann, 1995) which are widely used in automotive industry (Tosun et al., 2007). A cam with second-order discontinuity creates abrupt changes in acceleration, and the design of streamlined surfaces of aircraft, ship, and submarine requires curvature continuity to avoid flow separation and turbulence (Pegna and Wolter, 1992). Although G^2 continuous surfaces can meet curvature continuity requirement, higher order continuity such as continuous slope-of-curvature can suppress both laminar and turbulent separation and lead to higher aerodynamic efficiency (Shen et al., 2017). It is also stated that higher order ($>C^2$ continuous) surfaces are often required for certain numerical simulations and to meet visual, aesthetic, and functional requirements (Shen et al., 2017). Due to the importance of C^2 continuities, this thesis will investigate surface blending with C^2 continuities.

A comprehensive literature survey on blending time-independent parametric surfaces has been made in (Vida et al., 1994; Hatna et al., 2001). Although various blending methods have been developed, all these methods can only deal with time-independent primary surfaces which do not change their positions and shapes over time. However, in many situations, primary surfaces are constantly in motion and change shapes. It has been pointed out that the blending surface joining the wing of an aircraft to the fuselage must meet stringent aerodynamic requirements (Rossignac and Requicha, 1984). The blending surfaces connecting the torso and limbs of a running person must be always smooth and seamless.

For the primary surfaces that change motion and shapes over time, the task of surface blending is to generate smooth transition surfaces between these time-dependent surfaces whose mathematical representations involve a time variable. The existing methods of time-independent surface blending are unable to deal with the time variable due to the difficulty in involving the time variable in the surface blending algorithms effectively. For example, when using a constant-radius rolling-ball to blend two time-independent primary surfaces $S_1(u,v)$ and $S_2(u,v)$, the radius of the rolling ball is a constant $r = constant$ (Choi and Ju, 1989). When two primary surfaces become time-dependent, i.e., $S_1(u,v,t)$and $S_2(u,v,t)$, it is difficult to determine how the radius of the rolling ball changes with time, i.e., change $r = constant$ to $r = r(t)$ where t is a time variable. Although time-dependent surfaces can be approximated by time-independent surfaces at many time instants and various time-independent blending methods (vida et al., 1994) can be used to blend time-independent surfaces at the time instants, such treatments cannot smoothly connect the blending surface to the primary surfaces between two adjacent time instants, since the blending surfaces at the two adjacent time instants are interpolated to generate new blending surfaces and these new generated blending surfaces cannot satisfy the required continuities. In contrast, partial differential equations (PDEs) provide an effective solution to cope with time-dependent surface blending by involving the time variable in PDEs (You et al., 2012).

The above discussion highlights the importance of time-dependent surface blending. Since time-dependent surface blending is more difficult than time-independent surface blending, only one paper by You et al. (2012) investigated time-dependent C^1 continuous surface blending before this research. It indicates time-dependent surface blending has not attracted a lot of research attention and the paper by You et al (2012) has two weaknesses: (1) it is not applicable to time-independent surface blending, and (2) it is unable to achieve C^2 continuous surface blending as discussed below.

For the surface blending introduced by You et al. (2012), the first and second partial derivatives of blending surfaces with respect to a time variable t are involved in a vector-valued PDE. Despite the advantage in considering the effects of acceleration and velocity, it will cause the following problem. When primary surfaces are time-independent and do not change their positions and shapes, the blending surface should also be time-independent and does not change its position and shape. However, since the first and second partial derivatives of blending surfaces with respect to the time variable t are involved in the vector-valued partial differential equation, its closed form solution involves the time variable t and the blending surface defined by the closed form solution will change its shape with time. This is contradictory to the real situation and makes the surface blending technique proposed by You et al. (2012) unsuitable for time-independent surface blending.

The blending boundary constraints given in (You et al., 2012) only require the primary and blending surfaces to share the same position functions and first partial derivatives at trimlines. Therefore, the surface blending method developed in (You et al., 2012) can generate C^1 continuous blending surfaces only. The question on how to achieve time-dependent surface blending with C^2 continuity has not been addressed.

Another important problem is how to achieve satisfactory shapes of blending surfaces but still maintain exact satisfaction of blending boundary constraints. Kiciak (2011) introduced a function which is the integral of the square of length of the mean curvature gradient with respect to the surface measure and minimized the functional to achieve a satisfactory shape of a blending surface. The

numerical minimization algorithm involves heavy computations. How to achieve a satisfactory shape of blending surfaces easily and quickly without involving heavy numerical minimization calculations has not been addressed.

Apart from interpolating known boundaries to generate smooth blending surfaces, facial blendshapes interpolate two or more known 3D shapes to create new shapes. They are the predominant choice for realistic facial animation in the movie industry and a standard feature of commercial animation packages (Lewis et al., 2014). They have driven animated characters in Hollywood films and attracted a lot of research attention.

Facial blendshapes can be divided into the geometric and physics-based. Linear interpolation plays a dominant role in geometric facial blendshapes. As discussed in (Lewis et al., 2014), it can be mathematically formulated as $\boldsymbol{x} = \boldsymbol{x}_0 + \sum_{j=1}^{N} w_j(\boldsymbol{x}_j - \boldsymbol{x}_0)$, where $\boldsymbol{x}$, $\boldsymbol{x}_0$ and $\boldsymbol{x}_j$ contains the coordinates of all the vertices of a new facial shape (called the blended shape in this thesis) to be created, a neutral shape, and the j^{th} blendshape (called the target shape), respectively, $0 \leq w_j \leq 1$ is a weight, and N is the total number of all the target shapes.

Linear interpolation based facial blendshapes are very popular since they have the advantages of simplicity, expressiveness, and interpretability. Despite this, the following limitations have been identified by Barrielle et al. (2016). First, linear interpolation constrains the space of facial expressions to lie in an affine subspace. Since not all values of the weight w_j yield plausible facial deformations such as nonlinear and rotational deformations (Kozlov et al., 2017), the space of facial expressions should not be considered affine. Some approaches such as pose space deformation-based corrections (Seol et al., 2012) have been proposed to deal with nonlinearities which are not represented by the affine model. Second, facial blendshapes could be regarded as samples from a hypothesized manifold of facial expressions. Generating a new facial shape requires enough target facial shapes to sample the manifold and define local linear interpolation functions. Creating enough target facial shapes is usually an iterative and labour-intensive process. New techniques of facial

blendshapes are required to create a superset of blended shapes which contain not only the subspace of the blended shapes generated by linear interpolation but also those that linear interpolation is unable to generate. Some efforts have been devoted to extending the subspace such as splitting the face model into segments and applying a local affine model to each segment (Tena et al., 2011) and capturing geometric shapes regulated by physics-inspired deformations (Ma et al., 2011; Ma et al., 2012; Li et al., 2013). However, these approaches struggle to consider complex physical behaviours. Third, linearly combining blendshapes tend to move groups of vertices jointly as blocks over time which cannot render fine temporal behaviours. Linear interpolation changes facial shapes with a same deformation rate. Generating special effects such as acceleration and deceleration effects (Noh and Neumann, 1999) requires animating facial shape changes with different deformation rates.

Different from linear interpolation that is purely geometric, physics-based facial blendshapes (Barrielle et al., 2016; Kozlov et al., 2017) are to add physics to facial blendshapes which has a potential to tackle the above problems. Especially, when physics-based simulations are combined with data-driven approaches, realistic facial animation can be created. Apart from its capacity in tackling the above problems, another important feature of physics-based approaches is that the simulation parameters can be controlled to achieve the desired effects (Ichim et al., 2017). These simulation parameters include mass, damping coefficient, stiffness coefficient, rest shape volume, and static bone structure etc.

Existing physics-based facial animation is obtained by various numerical methods such as the finite element method which simulates facial models as thin shell (Barrielle et al., 2016) or a solid volume (Ichim et al., 2017). As stated in (Kozlov et al., 2017), "*these methods can simulate all the desired dynamic effects but are extremely difficult and time-consuming to set up*" since they require special knowledge and skills, additional mesh pre-processing, large computing capacity, and high computational costs and are not easy to implement and use. Therefore, they are not applicable to the situations where real-time animation or high animation frame rates are required. As stated in (Koch

et al., 1996), “In facial animation, computational complexity is a fundamental constraint, and real-time performance is often much more important than a highly accurate facial shape in terms of a still image.”

1.2 Research questions

As discussed above, differential equation-based surface blending and physics-based facial blendshapes have not answered the following questions.

1. How to develop time-dependent geometric modelling of C^2 continuity? Surface blending with tangent and curvature continuity is frequently required in many applications. Aerodynamics has been used to investigate aerodynamic performance of racing cars, high-speed trains, and airplanes. Aerodynamics is an important application of fluid mechanics, which is based on time-dependent Navier-Stokes equations. To achieve good aerodynamic performance, racing cars, high-speed trains, and airplanes must have streamlined shapes whose transition surfaces can be obtained from C^2 continuous surface blending. Although there are a lot of publications about time-independent surface blending, time-dependent surface blending with C^2 continuity has not been fully developed.

2. How to integrate time-dependent and time-independent surface blending? Current differential equation-based approaches can deal with either time-independent or C^1 continuous time-dependent surface blending. There are no differential equation-based algorithms which can cope with both time-independent and time-dependent surface blending. Separate treatment of time-independent and time-dependent surface blending will increase implementation effort and time and does not facilitate practical applications.

3. How to develop simple, efficient and realistic physics-based facial blendshapes? Physics-based facial blendshapes and animation can generate more realistic results, but they require special knowledge and skills, additional mesh pre-processing, large computing capacity, and high

computational costs and are not easy to implement and use. The challenge on developing efficient and realistic facial blendshapes has not been solved.

1.3 Aim and objectives

To answer the above questions, the work presented in this thesis aims to develop new differential equation-based shape interpolation approaches and apply them in surface blending and facial blendshapes. This aim will be achieved through the following objectives.

- Our first objective is to integrate time-independent and time-dependent C^2 continuous surface blending in a unified mathematical model. This mathematical model consists of a vector-valued sixth-order partial differential equation and C^2 continuous blending boundary constraints.
- Our second objective is to develop three new approaches of C^2 continuous surface blending. They are the closed form solution-based approximate analytical approach, variable decomposition-based approximate analytical approach, and differential property-based approximate analytical approach.
- Our third objective is to develop an analytical approach of dynamic facial blendshape animation. It does not require any special knowledge and skills of numerical calculations and can be used to create facial blendshape animation with high animation frame rates.
- Our fourth objective is to develop a blending force-based facial animation framework. It consists of the slider force-based, exponentiation force-based, and random force-based facial blendshapes.

1.4 Contributions

Through developing new differential equation-based shape interpolation for surface blending and facial blendshapes, the proposed research has made the following contributions.

1. A new mathematical model to integrate both time-dependent and time-independent C^2 continuous blending of parametric surfaces.

2. A closed form solution-based approximate analytical approach to effectively tackle various C^2 continuous blending problems of time-independent constant parametric surfaces.

3. A variable decomposition-based approximate analytical approach which decomposes variables to simplify the problem treatment and creates C^2 continuous blending between time-dependent and time-independent parametric surfaces.

4. A differential equation-based approximate analytical approach which considers different differential properties to speed up the solution process of the mathematical model and integrates time-dependent and time-independent C^2 continuous blending into a unified framework to generate various surfaces efficiently and accurately.

5. A novel mathematical model of dynamic deformations which integrates the equation of motion and the constraints of the source and target shapes.

6. The first analytical approach of dynamic facial blendshape animation which is easy to implement and use by animators without special knowledge and skills of numerical calculations.

7. A simple blending force-based facial animation framework which integrates slider force-based, exponentiation force-based, and random force-based facial blendshapes to create various facial blendshape animation with high animation frame rates and good realism.

1.5 Thesis structure

This thesis contains five chapters. Each chapter is briefly outlined below.

1: Introduction - As presented above, Chapter 1 gives the background information, identifies the research questions, defines the research aim and objectives, and highlights the research contributions. It also gives the thesis structure and lists the publications from this research.

2: Literature review – This chapter presents a literature review about surface blending and facial blendshapes with a focus on differential equation-based approaches.

3: C^2 continuous blending of varying and constant surfaces with differential equation-based shape interpolation – This chapter investigates differential equation-based C^2 continuous blending of time-dependent and time-independent surfaces. It includes proposing a new mathematical model, developing a closed form solution-based approximate analytical approach, variable decomposition-based approximate analytical approach, and differential property-based approximate analytical approach, examining the accuracy and efficiency of the proposed approaches and the effects of the second partial derivatives, and investigating the applications of the developed approaches in time-dependent and time-independent surface blending.

4: Facial blendshapes with differential equation-based shape interpolation – This chapter proposes a new differential equation-based technique of facial blendshapes. It first proposes the mathematical model and derives a simple and efficient closed form analytical solution of the model. After that, it develops a blend force-based, exponentiation force-based, and random force-based facial blendshapes to achieve highly efficient and realistic facial animation.

5: Conclusion and future work – This chapter starts by concluding the work presented in this thesis, followed by future work.

1.6 Publications

- Xiangyu You, Feng Tian, Wen Tang, 2018. A unified approach to blending of constant and varying parametric surfaces with curvature continuity. In Proceedings of Computer Graphics International 2018. ACM Press, pp. 51-56.

- Xiangyu You, Feng Tian, Wen Tang, 2019a. C2 continuous blending of time-dependent parametric surfaces. Journal of Computing and Information Science in Engineering 19(4): 041005. DOI: 10.1115/1.4043042.
- Xiangyu You, Feng Tian, Wen Tang, 2019b. Highly efficient facial blendshape animation with analytical dynamic deformations. Multimedia Tools and Applications 78(18): 25569-25590.

2. Related work

2.1 Surface blending review

Surface blending is a geometric algorithm, which creates a smooth surface connecting different surfaces. These different surfaces, which are connected together by a smooth surface, are called primary surfaces. And the smooth surface is called a blending surface. The interface curves where the blending surface is connected to primary surfaces are called trimlines. The constraints, which the blending surface should satisfy at trimlines, are called blending boundary constraints.

A surface can be mathematically represented explicitly, implicitly, and parametrically. Accordingly, surface blending can be divided into explicit, implicit, and parametric blendings. Explicit blending (You et al., 2004b) is the blending between different explicit primary surfaces, implicit blending (Hartmann, 2001) is to create a smooth surface between implicit surfaces, and parametric blending (Song and Wang, 2007; Kiciak, 2011) is to form a smooth transition between parametric surfaces. Since explicit surfaces has a very weak capacity in representing complicated shapes, it is difficult to find the publications on explicit surface blending except for (You et al., 2004b). In what follows, existing work on implicit blending and parametric blending will be reviewed.

2.1.1 Implicit blending

Some blending methods for implicit surfaces were introduced by Woodwark (1987). It was stated by Liming (1979) and Faux and Pratt (1979) that determining the blending surface relies on finding a conic tangential to two straight lines that define a plane. Later on, Middledich and Sears (1985) applied such a technique to surface blending. The formulation for blending two or three implicit primary surfaces was proposed and named as the potential method by Hoffmann and Hopcroft (1985,

1986, 1987). This potential method was also developed into a projective potential method to blend an arbitrary number of primary surfaces to form a convex corner (Kosters, 1989), and the applicability of the potential method for blending implicitly defined surfaces was extended by Kosters (1991). Ohkura and Kakazu (1992) made a more general formulation for both convex and concave combinations for the blending of three primary surfaces, and extended the projective potential method to produce a convex combination of three surfaces using the two-map explanation of the potential method. Hsu (2018) obtained four families of k-ary Boolean set blends with C^1 continuity which use a single blending operation to create the blending among more than two models. In the work by Angles et al. (2017), implicit blending is used with 2D sketches to obtain 2D shapes or 3D surfaces. Derived from 2D sketches, a gradient-based implicit operator is obtained, which creates 3D implicit blends from a user's sketch with the effect of any existing blending operator.

These methods are effective in blending time-independent implicit primary surfaces. They are not applicable to blending between time-dependent parametric primary surfaces, which will be investigated in this thesis.

Blending of one shape into another called metamorphosis which is based on implicit surfaces or function representation combined with set-theoretic operations, R-functions and displacement functions was also well investigated. By using R-functions consisting of union, intersection, and subtraction and introducing a displacement function, an exact analytical definition of the blending set-theoretic operations over geometric solids with implicitly defined surfaces were proposed by Pasko and Savchenko (1994). Using the control points on the surfaces of two solids or an additional bounding solid to define the shape and location of the blend which is called bounded blending, new analytical formulae of bounded blending were proposed by Paska et al. (2002) for functionally defined set-theoretic operations. By using implicit surfaces to functionally define initial shapes of 3D volume models, the metamorphosis was used to create different shapes of the 3D volume models in (Kazakov et al., 2003). Using implicit surfaces to define simple shapes without holes and self-

intersections and Function representation (FRep) solids to define complicated shapes, a new approach was proposed by Pasko et al. (2004a) to achieve shape transformation between 2D or 3D objects with different topology and positions through increasing dimension, applying bounded blending with added materials, and making cross-section to get intermediate shapes during the transformation. Using bounded blending operations in space-time for shape metamorphosis, two extensions were made by Pasko et al. (2004b). One extension used "smoothed" versions of half-cylinders to solve the problem of "jump" in animation or the rapid transition between shapes in the given interval. The other extension applied the bounded blending union operation to the corresponding "smoothed" 4D space-time half-cylinders. Using two types of functions: FRep functions (Pasko et al., 1995) and parametric functions, a new function-based extension of Virtual Reality Modeling Language (VRML) was made by Liu and Sourin (2005) to obtain function-defined shape metamorphoses in VRML. Using standard R-functions to describe pure set-theoretic operations and introducing localized displacements to the standard R-functions, new analytical formulations for bounded blending set-theoretic operations were proposed by Pasko et al. (2005). Based on the use of analytical implicit, explicit and parametric functions, function-based extensions of VRML and X3D were made by Liu and Sourin (2006), making the programming of function-defined shape metamorphoses very easy in visual cyberworlds. By introducing additional controllable affine transformations to initial objects in space-time to address the fast transition between the shapes and give users more control and an additional non-linear deformation operation to the pure space-time blending, real-time space-time blending with improved user control was achieved by Pasko et al. (2010). By proposing a formula of the pairwise metamorphosis with a variety of functions for different stages of deformation, morphing and offsetting, and extending it to the metamorphosis between groups of shapes with weighted feature elements, an approach was proposed by Sanchez et al. (2013) to generate morphological shapes through user-controlled group metamorphosis. By collapsing the source mesh and the target mesh to approximate skeletal implicit

surfaces (convolution surfaces), transforming shape between the two implicit surfaces, and projecting the obtained implicit surface to the target mesh, the metamorphosis between two skeleton-driven animated meshes with arbitrary topologies was obtained by Kravtsov et al. (2014). A space-time transfinite interpolation was obtained by Sanchez et al. (2015) through extending a transfinite interpolation to spatio-temporal variations in heterogeneous volume objects. The obtained space-time transfinite interpolation was used for volumetric metamorphosis where both geometry and volumetric properties change in time. Space-time blending was extended by Adzhiev et al. (2020) to deal with heterogeneous objects. It interconnects geometry and attribute transformations of volumetric objects with attributes representing their physical properties. Based on a new unifying functionally-based hybrid representation called HFRep, Tereshin et al. (2021) presented a novel framework to model volumetric heterogeneous objects and used it to bend two oscillating 4D geometric shapes.

Most of the above methods investigated time-independent metamorphosis. Few of them such as Sanchez et al. (2015) dealt with time-dependent metamorphosis by involving a time variable. The problems tackled by the above methods are about gradual changes of one shape into another shape represented with implicit surfaces, which are different from the problems of blending between two time-dependent varying parametric surfaces to be investigated in this thesis.

2.1.2 Parametric blending

Various blending methods of time-independent parametric surfaces have been developed. According to the survey paper by Vida et al. (1994), these surface blending methods can be divided into: rolling-ball-based blends, spine-based blends, trimline-based blends, blends based on polyhedral methods and other methods including a cyclide solution, PDE-based blends, and Fourier-based blends.

Among various surface blending approaches above mentioned, rolling ball methods are the most popular and most widely used for rounding edges and corners of mechanical parts (Hatna et

al., 2001). The pioneer work of the rolling ball methods was presented by Rossignac and Requicha (1984). With this method, a blending surface is created by rolling a ball along two primary surfaces. Various rolling-ball blending methods can be divided into constant-radius and variable-radius ones.

Constant-radius rolling-ball blending methods were investigated by Rossignac and Requicha (1984), Choi and Ju (1989), Farouki and Sverrisson (1996), Kós et al. (2000), and Dahl and Krasauskas (2012). Rossignac and Requicha (1984) proposed a new method to incorporate constant-radius blends which is achieved by rolling a sphere in contact with primary surfaces to be blended together. Choi and Ju (1989) mathematically constructed rolling-ball blends through sweeping rational quadratic curves and representing corner blends where three surfaces meet with a convex combination of linear Taylor interpolants. Farouki and Sverrisson (1996) investigated the numerical methods of constant-radius blends to achieve prescribed-precision approximation and guarantee the satisfaction of specified tolerance. Kós et al. (2000) discussed how to determine the radius of rolling ball blends from point data which have been pre-processed and segmented. By applying canal surfaces, Dahl and Krasauskas (2012) developed a general algorithm to parametrize fixed radius rolling ball blends of pairs of natural quadrics.

Constant-radius rolling-ball blending is not applicable to some situations where the distance between two trimlines varies along the trimlines. For such situations, variable-radius rolling-balling blending is effective.

Variable-radius rolling-ball blending methods were examined by Chuang et al. (1995), Chuang and Hwang (1997), Lukács et al. (1997), Lukács (1998), Chuang and Lien (1998), and Kós (2011). Chuang et al. (1995) calculated a parametric form of variable-radius spherical and circular blends by using the derived spine curve and linkage curves. Chuang and Hwang (1997) tackled the problems that the radius for variable-radius blending is difficult to specify and the spine curve is hard to trace by introducing several geometric constraints to specify the variable radius and a paradigm to implement the constraints of tracing the spine curve. Lukács et al. (1997) treated

variable radius rolling ball blending surfaces as the envelopes of one parameter families of varying radius balls which are special cases of discriminant sets. Lukács (1998) used the theory of envelopes and discriminant set to analyse variable radius rolling ball blending surfaces, determine the differential geometric invariants of the surfaces, and characterize the progressive and regressive points on the variable radius rolling ball blending surfaces. Chuang and Lien (1998) presented two formulations to determine general blending between parametric surfaces. The first formulation represents the blend as a sweeping surface whose radii satisfy a specific one-parameter curve, and the second formulation also defines the blend as a sweeping surface but whose radii satisfy a specific two-parameter surface. Kós (2011) generalized the algorithms for reconstructing constant radius rolling ball blends to reconstruct variable radius blends.

Rolling ball blends are also used with automatic segmentation and structural recovery. Zhang et al. (2020) presented a clustering algorithm to extract general quadric and rolling-ball blending regions, proposed a method to automatically segment 3D models to reveal surface structures of the underlying shapes, and developed a skeleton extraction-based algorithm to recover the parameters of the rolling center trajectories and ball radius and fit rolling-ball blending surface patches.

Due to simplicity, rolling-ball blends are very popular. However, they are not suitable to blend some surfaces. For example, when a surface is separated into two parts with a gap between the two parts, rolling-ball blending methods are not suitable to create a blending surface to smoothly connect the two parts together. For such situations, PDE-based surface blending provides an effective means.

Cyclides are effective in dealing with some simple blends such as generating the transition surface for the situation where a cylinder obliquely meets a plane. Implicit quartic equations or parametric expressions represented with trigonometrical parameterisation or rational biquadratic Bézier equations can be used to describe cyclides. Using both Dupin ring cyclides and parabolic cyclides, Allen and Dutta (1997a) investigated the blends between natural quadrics by using a new definition of a pure cyclide blend to force the construction of non-singular cyclide blends which

deliberately exclude cyclide joins and singular surfaces. Later on, they (Allen and Dutta, 1997b) studied the relationship of the existence of cyclide blends to the common inscribed sphere condition. They (Allen and Dutta, 1997c) also examined the supercyclide blends between a plane and a cone and the supercyclide blends between a plane and a cylinder. Shene (1997) presented a necessary and sufficient condition for two cones to have a blending Dupin cyclide and proposed a new construction algorithm to establish the correspondence between the points on one or two coplanar lines for blending cones with Dupin cyclides.

As discussed above, cyclides have been used to smoothly connect two cylinders, blend two cones, and create plane/cone and plane/cylinder blends. These blending problems are simple. For many complicated blending tasks such as rounding a corner and filling a n-sided hole, there are no reports of using cyclides to tackle these blending problems.

Bizzarri et al. (2015) proposed a method of constructing blending surfaces, modified the method for n-way blends, investigated contour curves on ringed surfaces, and used the contour curves to calculate approximate parameterizations of implicitly given blends. They used rational envelope (RE) curves to develop an interpolation algorithm and used the algorithm to produce rational surface blends between canal surfaces (Bizzarri et al., 2016). They also examined RE surfaces, used RE surfaces to develop efficient algorithms, and employed them in rational skinning and blending of sets of spheres and cones/cylinders (Bizzarri et al., 2017).

RE surfaces are especially suitable for smoothly connecting two cylinders. By using an auxiliary sphere, RE surfaces can be used to blend more than two cylinders. By skinning three or more spheres, RE surfaces can be extended to blend disconnected cones/cylinders in various orientations. Although strong capacity of RE surfaces in blending disconnected cones/cylinders, they are incapable in some other blending situations such as blending two or more intersecting planes.

In addition to the above surface blending methods, Roach and Martin (1992) proposed a Fourier-based blending method which predicts and controls curved shapes in a two-dimensional

system. Kim and Elber (1996) presented a symbolic approach for the computation of blending surfaces with good accuracy and meaningful control which provides accurate tangent plane continuities along the entire boundary of blending surfaces.

2.1.3 Noncircular blending

Rolling-ball methods can create circular shapes of blending surfaces only. In general situations, the shapes of blending surfaces are noncircular. Such noncircular blending surfaces can be generated with some other methods such as filling n-sided regions (Schichtel, 1993; Hsu, 1998; Piegl and Tiller, 1999; Hwang and Chuang, 2003; Yang et al., 2006; Shi et al., 2010; Liu, 2015), polyhedral vertex blending with setbacks using rational S-patches (Zhou and Qian, 2009; Zhou, 2010), branching blends between two natural quadrics with Pythagorean normal surfaces (Krasauskas, 2008), and partial differential equation (PDE)-based methods (Bloor and Wilson, 1989).

PDE-based methods are most powerful in creating different noncircular shapes of blending surfaces. They formulate surface blending as a mathematical boundary-value problem and adjust shape control parameters embedded in the PDE to generate different shapes of blending surfaces while still keeping exact satisfaction of boundary constraints. The solution to a vector-valued partial differential equation subjected to the constraints of boundary curves and the first, second and higher order partial derivatives at trimlines represents a blending surface. Since surface blending with tangent and curvature continuity is the most frequently met in engineering (Pegna and Wolter, 1992; Hatna et al., 2001) in comparison with higher order continuities and tangent continuity surface blending has been investigated in (Bloor and Wilson, 1989; Cheng et al., 1990; Bloor et al., 2000; You al., 2012), this thesis examines constant and varying parametric surface blending with curvature continuity.

PDE-based surface blending was pioneered in (Bloor and Wilson, 1989). The main problem with PDE-based surface blending is how to solve partial differential equations effectively and

efficiently. Numerical, accurate, and approximate analytical methods can be used to solve partial differential equations. Various numerical methods such as the finite difference method (Bloor et al., 1995), the finite element method (Brown et al, 1998; Li and Chang, 1999; Jacobson et al., 2010; Pan et al., 2014), and direct discretization with the discrete Laplace operator (Park et al., 2015; Stanko et al., 2016) are most powerful. Despite their powerful capacity, they have the following limitations. First, they generate blending surfaces with discrete boundary representations which are unsuitable for the requirement of good continuity. Second, they involve many design variables and a lot of calculations which cause high requirements for computing devices and slow response. Third, specific knowledge and skills of the numerical methods are required to carry out the numerical calculations. To overcome the limitations of numerical methods in solving partial differential equations for surface blending, some accurate and approximate analytical methods have been proposed in (Zhang and You, 2002; Bloor and Wilson, 1996; You et al., 2004a, 2004b). Most recently, Wu and Zhu (2021) introduced a sixth order partial differential equation into bicubic B-spline surface with uniform and quasi-uniform knots and used the obtained method to construct an open blending surface and a closed blending surface with C^1 continuity.

Compared to numerical methods, accurate and approximate analytical methods can overcome the limitations of numerical methods but with less powerful capacity in complicated surface blending such as filling n-sided holes. Accurate methods obtain the closed form solution of partial differential equations, but only apply to some simple and special cases (Zhang and You, 2002). Approximate analytical methods (Bloor and Wilson, 1996) can solve more complicated surface blending problems and are more powerful than accurate methods. They are more efficient than numerical methods. However, obtaining approximate analytical formulae of PDE-base surface blending is more difficult than using numerical methods to solve the same surface blending problems. In this thesis, approximate analytical methods will be adopted to develop new surface blending approaches.

Most PDE-based methods and all other surface blending approaches only deal with constant surface blending where primary surfaces do not move and change their shapes. Before this research, only one paper by You et al. (2012) dealt with time-dependent surface blending with C^1 continuity and initiated the research to blending of dynamic varying parametric surfaces. In the paper, a vector-valued dynamic fourth-order partial differential equation was used to develop a new approach which tackles blending of varying parametric surfaces with C^1 continuity. The method is not applicable to time-independent surface blending since the solution of the partial differential equation always involves the time variable *t*. In addition, C^2 continuous surface blending has not been investigated.

To tackle the above problems and generalize the technique introduced in (You et al., 2012), this research drops the time variable *t*, proposes sixth-order partial differential equations, introduces the second partial derivative continuity into blending boundary constraints to address C^2 continuity, and develops the first group of approximate analytical solutions of the sixth-order partial differential equations to unify both time-independent and time-dependent C^2 continuous surface blending. The developed solutions have the advantages of good accuracy and high efficiency.

2.2 Facial blending review

The new approach of facial blendshapes that has been proposed in this thesis is related to facial shape interpolation including facial blendshapes and physics-based facial animation. The existing works related to these two topics are reviewed below.

2.2.1 Facial shape interpolation

Facial animation is very important in computer animation. Facial shape interpolation is the most intuitive and commonly used technique in facial animation practice. It can be divided into linear and nonlinear interpolation. Linear interpolation (Bergeron and Lachapelle, 1985; Pighin et al., 1998; Seo and Thalmann, 2003; Liang and Ouhyoung, 2004; Sifakis et al., 2005) plays a dominant role in

facial blendshapes. Region-based linear interpolation of 3D face models proposed in (Tena et al., 2011) increases flexibility for modelling local deformations while keeping the model coherent. A bilinear model was applied to natural spatiotemporal phenomena such as dynamic faces and bodies (Akhter et al., 2012). A higher order generalization of linear model called multilinear was used for modelling identity, expression, and speech independently in (Vlasic et al., 2005). The cosine interpolation proposed in (Waters and Levergood, 1993) and other variations such as spline can provide acceleration and deceleration effects at the beginning and end of an animation. An optimization scheme was proposed in (Liu et al., 2011) to automatically explore the nonlinear relationship of facial blendshape animation from captured facial expressions. Linear blendshapes were used in a statistical model of 3D facial motion to reconstruct 3D face shapes from in-the-wild images (Richardson et al., 2017; Tewari et al., 2017; Tran and Liu, 2018; Booth et al., 2017, 2018). Using linear blendshapes to describe both the motion that is produced by expression (Cheng et al., 2018) and/or motion that is produced by speech (Tzirakis et al., 2019), continuous parameters of the linear blendshapes are used to drive generative adversarial networks (GANs) for synthesizing expressive face images through sliding 3D blendshape parameters (Ververas and Zafeiriou 2020) where sliders were used to control continuous values of statistical blendshape model of facial motion and SliderGAN was used to transform an input face image into a new one and edit a facial image according to expression and speech blendshapes.

In recent years, more research studies were reported. Tu et al. (2019) proposed an automatic approach to select a set of base expressions from a sequence of facial motions by using the Procrustes analysis to estimate the difference among face models and obtain the composition of the base expressions. By selecting a few facial landmarks on an avatar mesh, Onizuka et al. (2019) presented an automatic approach to easily create realistic key-shapes of any avatar with largely different shapes and topology in comparison with the source template mesh for generating facial blendshapes. In order to improve the time-consuming blendshape editing process, Cetinaslan and Orvalho (2020a)

implemented a sketch-based interface, which uses geodesic circles to confine the edits to the local geometry and allows the artists to directly sketch onto the 3D facial model to localize direct manipulation of blendshape models for facial animation. Wang and Ling et al. (2020) used the facial emotion information extracted from an arbitrary facial motion sequence to define semantics and updated facial blendshapes with large-scale and middle-scale face shapes and fine-scale facial details. Given a target neutral face mesh and a small input dataset, Wang and Bradley et al. (2020) developed an approach to synthesize realistic facial expressions by introducing global-local multilinear models to integrate the strengths of global models in coarse motion estimations and local models in expression-specific and identity-specific details where local identity models contain expression variation for a specific identity and local expression models contain identity variation for a specific expression. In order to predict perceptual importance of blendshapes under different activation levels, Carrigan and Zell et al. (2020) investigated the noticeability of blendshapes and proposed new perceptually based models, which use commonly-used geometry and image-based metrics to predict visibility. Carrigan and Zibrek et al. (2020) treated the semantics of facial expressions as an integer optimization of the blendshape weights and used the obtained facial expressions as semantic references during scanning to calculate as-few-as-possible training expressions, which have minimum overlap of activated blendshapes. In order to set up a simulation rig and control material properties based on facial performances more easily and efficiently, Romeo and Schvartzman (2020) proposed a workflow to create an activation map for pseudo-muscles, added physical simulation obtained from an extended position-based dynamics solver to facial animation, and used the activation map to control the simulation behaviour. Chen et al. (2020) used geometric optimization to generate facial blendshapes preserving the identity exaggeration style of an artist-drawn caricature and a conditional generative adversarial network (cGAN) to obtain dynamic textures of target expressions. Wang and Cheng et al. (2020) used a facial blendshape to enhance the performance of facial expression recognition (FER) through proposing a two-stream network called

BlendshapeFERNet. The first stream called image-based FER network extracts the expression information from images and the second stream called blendshape regression network extracts 3D facial muscle movements. In order to avoid unexpected results during facial expression editing by direction weight manipulation interfaces based on the pseudo-inverse of the matrix that contains all the vertex positions of blendshape target shapes, Cetinaslan and Orvalho (2020b) used the transpose of the Jacobian matrix rather than the pseudo-inverse to reduce unexpected movements during blendshape weight editing. Han et al. (2021) used linear regression and an autoencoder to automatically create individually optimized blendshapes from the captured face and selected the trained results that are more similar to the original face.

Facial shape interpolation reviewed above is simpler and more efficient than physics-based facial animation. Due to high efficiency, facial shape interpolation such as facial blendshapes has been widely applied in computer animation industry. However, facial shape interpolation based on linear interpolation and nonlinear interpolation does not consider the underlying physics and is less capable in creating realistic facial animation in comparison with physics-based facial animation.

Unlike these purely geometric face shape interpolation methods which blend source and target shapes through one weight or a set of weights, the approach proposed in this thesis introduces one source shape and one target shape as the constraints of the equation of motion in the physics-based mathematical model, and uses blended forces and different values of the time variable involved in the equation of motion to create different blended shapes.

2.2.2 Physics-based facial animation

Physics-based facial animation is introduced to consider the underlying physics of facial animation and tackle the problems of purely geometric facial animation in creating realistic facial expressions. Various physics-based approaches have been proposed such as the finite element method, finite difference method, boundary element method, finite volume method, mass-spring systems, shape

matching methods, and reduced deformable models based on model analysis (Nealen et al., 2006). Among them, the finite element method is the most accurate and has been used in movies to create realistic deformations, and mass-spring systems are widely applied in computer graphics since they are efficient to calculate (Hong et al., 2016). In this thesis, only the work on mass-spring systems and the finite element method will be reviewed.

Mass-spring systems actuated by vector muscles were introduced to animate facial expressions in (Platt and Badler, 1981). Video-based tracking of facial features was used to calculate muscle actuation parameters in (Terzopoulos and Waters, 1993). For an isotropic linear elastic reference model, Lloyd et al. (2007) derived analytical expressions of the spring parameters for simulating deformable models with a mass-spring system. Natsupakpong and Çavuşoğlu (2010) determined elasticity parameters of mass-spring-damper models for simulating deformable objects by minimizing the error between the stiffness matrices of the linear lumped element model and the linear finite element model of the same object. Physically-plausible shape blending was obtained by linearly interpolating spring rest length parameters of a mass-spring system between source and target shapes (Ma et al., 2011). San-Vicente et al. (2012) adjusted stress-strain and compressibility curves to a given reference behavior and proposed a method to estimate spring stiffness valid for linear and nonlinear material models used in mass-spring systems. Kot et al. (2015) investigated physical properties of a mass-spring system consisting of mass points and linear springs, proposed theoretical predictions of elastic parameters, and presented their verifications in several test cases. Later on, they investigated the construction of Mass-spring models for representing homogeneous isotropic elastic materials whose Poisson's ratio is adjustable (Kot and Nagahashi, 2017). Through modifying the traditional equation for mass spring dampers to introduce nonlinearity and viscoelasticity into the calculation of elastic force, a new method was proposed by (Xu et al., 2018) to simulate soft tissue deformation with mechanical properties of viscoelasticity, nonlinearity and incompressibility etc. for virtual surgery applications. Mass-spring systems were also used to

achieve physical simulations of volumetric skin behavior, lip contacts, and sticky lips for facial animation through the implicit Euler integration (Barrielle and Stoiber, 2019). Regarding the spring deformation energy depends on the volume only and is insensitive to changes of shape, Golec et al. (2020) developed a hybrid 3D mass-spring system to simulate any compressible isotropic elastic material and in particular the nearly incompressible biological soft tissues with Poisson ratio $\nu \cong 0.5$.

Mass-Spring Systems use simplified physical laws and mathematical models to achieve physically inaccurate but real-time simulations. They are simpler, easier and more efficient than the finite element method.

The finite element method is the most popular numerical method especially in computational sciences and engineering because it can simulate complex geometries and irregular shapes subjected to arbitrary constraints with the highest accuracy among various numerical methods. It was introduced to simulate facial surgery in (Koch et al., 1996). 3D finite element simulations were integrated into a computer-aided surgical planning system in (Keeve et al., 1996), and used to develop an anatomically accurate model for facial animation in (Sifakis et al., 2005). Rig-space physics uses the underlying finite element model and explores the interactions between physical models and artistic rigging (Hahn et al., 2012). Physical face cloning was obtained through nonlinear finite element simulations with a neo-Hookean material (Bickel et al., 2012). Recently, finite element simulations were combined with facial blendshapes to preserve volume and avoid self-collisions during facial animation (Barrielle et al., 2016), introduced to develop a new framework for the finite element simulation of facial muscle and flesh to achieve anatomically and biomechanically accurate muscle models (Cong et al. 2016), used to enhance facial blendshape rigs (Kozlov et al., 2017), integrated with a data-driven approach to develop data-driven physics (Kim et al., 2017), and applied with a novel muscle activation model to achieve physics-based face modelling and animation (Ichim et al., 2017). The finite element simulation was also used to simulate facial skeleton and musculature for adding physical detail to improve that anatomical

validity of both blendshapes and blendshape animation in the film Kong (Cong et al., 2017). Physics-based finite element simulations were introduced into animations of arbitrary models with triangle meshes by Li et al., (2017), which include automatically creating simulation meshes, converting triangle mesh animations into animations of the simulation mesh, and dealing with collisions and self-collisions. By using the finite-element method to simulate the nonlinear, quasi-incompressible, isotropic, hyperelastic properties of a tongue and integrating it with the anatomical model consisting of the skeleton, muscle, and skin, a real-time 3D system was proposed by (Jun et al., 2018) to generate facial animation. Smith et al. (2018) designed a novel Neo-Hookean energy and derived closed form eigenvalue and eigenvector expressions to make the new Neo-Hookean model behave well across a wide range of Poisson's ratios and remain stable under large deformations. A simple finite element simulation approach was proposed by (Kozlov et al., 2019) to produce facial animation which reconstructs a very small number of head poses of the actor in 3D and recovers the required simulation parameters to best match a real actor's face motion. Building on a meshless finite elements technique which has already been used successfully for the elastic simulation of surface models in real time, Uhlmann and Brunnett (2019) employed a point-based technique to simulate facial movements. Cong and Fedkiw (2019) presented a physics-based facial retargeting method, which conducts the finite element simulation of an actor's muscle to obtain the best performance-matched blendshape muscles, transfers the obtained blendshape muscles to a creature model to create the corresponding creature blendshape muscles, and uses the creature blendshape muscles to drive a creature muscle simulation to achieve the retargeted performance. Kim et al. (2019) proposed an inversion-safe transversely isotropic energy and an element rehabilitation method for badly-conditioned finite elements to robustly simulate large deformations even when degenerate and zero-volume elements are involved. By using a material space-dependent function to modulate the deformation gradient, Kadleček and Kavan (2019) built a neutral facial volumetric model from magnetic resonance images, and solved an inverse physics problem with linear finite elements to

learn facial mechanical properties, i.e., heterogenous stiffness and prestrain, from the training data. Without using 3D finite elements for physics-based real-time facial animation driven by muscle deformation, a 2D shell element due to the thin structure of facial muscles was used by (Kim et al., 2020) to avoid inefficient caused by 3D finite elements and simulate the movements and deformations of the facial system consisting of four layers of skin, subcutaneous layer, muscles, and skull. Considering the blending of the fine muscles and passive tissue, Dinev et al. (2020) reduced the number of parameters required to build personalized muscle module and used an augmented finite-element-like elastic potential with tetrahedral elements to achieve forward simulation of a face model. Wang and Zheng et al. (2020) combined finite element simulations to consider character's physically based soft-tissue dynamics with linear blend skinning and pose-space deformation for geometric mesh animation. Since the lip plays a critical role in facial aesthetics, Kim et al. (2021) presented a new finite element incremental simulation method, which improves the prediction accuracy in the lip region with a realistic sliding effect.

In comparison with mass-spring systems, the finite element method uses more accurate physical laws and mathematical models. It can easily increase the order of finite elements so that physics fields can be approximated very accurately, locally refine the mesh to get better approximations, and tightly couple multi-physics such as electromagnetic heating to get an accurate solution. Therefore, the finite element method is more powerful and accurate but slower than mass-spring systems.

Different from physics-based facial blendshapes using numerical methods which require an additional preprocess to convert polygon meshes into finite elements or mass-spring meshes, special knowledge and skills to implement and use them, and large computer memory and high computational costs to realize the simulations leading to low animation efficiency, an analytical approach proposed in this thesis directly works on the vertices of polygon models and avoids the problems of numerical methods.

3. C^2 continuous blending of varying and constant surfaces with differential equation-based shape interpolation

In this chapter, how to introduce partial differential equation-based shape interpolation to achieve C^2 continuous blending of varying and constant surfaces will be investigated. Here, "varying surfaces" also called "time-dependent surfaces" mean primary surfaces change their shapes with time and are time-dependent, and "constant surfaces" also called "time-independent surfaces" indicate that primary surfaces do not change their shapes with time and are time-independent.

First, the mathematical model of a unified approach to blending of constant and varying parametric surfaces with C^2 continuity will be proposed in Section 3.1. After that, three different approaches will be developed, including (1) the closed form solution-based approximate analytical approach to be investigated in Section 3.2; (2) the variable decomposition-based approximate analytical approach to be examined in Section 3.3; (3) the differential property-based approximate analytical approach which will be discussed in Section 3.4. Finally, concluding remarks and future work will be given in Section 3.5.

3.1 Mathematical Model

When two primary parametric surfaces are to be connected with C^2 continuity, the blending surface must satisfy the constraints of the position functions and the first and second partial derivatives of the two primary parametric surfaces at the trimlines. If the two primary parametric surfaces change their shape with the time, the position functions and the first and second partial derivatives of the primary parametric surfaces at the trimlines are the functions of time variable t. Therefore, the

position function and the first and second partial derivatives for the first primary parametric surface at the trimline can be written as $\boldsymbol{C}_1(v,t)$, $\boldsymbol{C}_2(v,t)$ and $\boldsymbol{C}_3(v,t)$, and those for the second primary parametric surface can be written as $\boldsymbol{C}_4(v,t)$, $\boldsymbol{C}_5(v,t)$ and $\boldsymbol{C}_6(v,t)$.

Assuming two varying parametric surfaces to be smoothly blended together are $S_1(u,v,t)$ and $S_2(u,v,t)$, with $S_1(u,v,t)$ to be blended at the position $u = u_1$ and $S_2(u,v,t)$ to be blended at the position $u = u_2$. S_1 and S_2 contain three components $[S_{1x}(u,v,t),\ S_{1y}(u,v,t),\ S_{1z}(u,v,t)]^T$ and $[S_{2x}(u,v,t),\ S_{2y}(u,v,t),\ S_{2z}(u,v,t)]^T$ respectively. The position function and the first and second partial derivatives of the varying parametric surface $S_1(u,v,t)$ with respect to the parametric variable u at the position $u = u_1$ are

$$\boldsymbol{C_1}(v,t) = \boldsymbol{S_1}(u_1,v,t),$$

$$\boldsymbol{C_2}(v,t) = \partial \boldsymbol{S_1}(u_1,v,t)/\partial u$$

$$\boldsymbol{C_3}(v,t) = \partial^2 \boldsymbol{S_1}(u_1,v,t)/\partial^2 u$$

where $\boldsymbol{C_1}(v,t), \boldsymbol{C_2}(v,t), \boldsymbol{C_3}(v,t)$ contain three components below, respectively

$$\left[C_{1x}(v,t),\ C_{1y}(v,t), C_{1z}(v,t)\ \right]^T$$

$$\left[C_{2x}(v,t),\ C_{2y}(v,t), C_{2z}(v,t)\ \right]^T$$

$$\left[C_{3x}(v,t),\ C_{3y}(v,t), C_{3z}(v,t)\ \right]^T$$

And the position function and first and second derivatives of the varying parametric surface $S_2(u,v,t)$ at the position $u = u_2$ are

$$\boldsymbol{C_4}(v,t) = \boldsymbol{S_2}(u_2,v,t),$$

$$\boldsymbol{C_5}(v,t) = \partial \boldsymbol{S_2}(u_2,v,t)/\partial u$$

$$\boldsymbol{C_6}(v,t) = \partial^2 \boldsymbol{S_2}(u_2,v,t)/\partial^2 u$$

where $\boldsymbol{C_4}(v,t), \boldsymbol{C_5}(v,t), \boldsymbol{C_6}(v,t)$ also contains three components below, respectively

$$\left[C_{4x}(v,t),\ C_{4y}(v,t), C_{4z}(v,t)\ \right]^T$$

$$\left[C_{5x}(v,t),\ C_{5y}(v,t), C_{5z}(v,t)\ \right]^T$$

$$\left[C_{6x}(v,t),\ C_{6y}(v,t), C_{6z}(v,t)\ \right]^T$$

If the blending surface $S(u,v,t)$ connects two varying parametric surfaces at the positions $u = u_1$ of the first primary surface and $u = u_2$ of the second primary surface smoothly, it must satisfy the following blending boundary constraints, which require the blending surface to share the same boundary curves, the first partial derivatives, and the second partial derivatives with primary surfaces on the trimlines.

$$\begin{aligned} u = 0 \qquad & \partial^n \boldsymbol{S}(u,v,t)/\partial u^n = \boldsymbol{C}_{n+1}(v,t) \\ u = 1 \qquad & \partial^n \boldsymbol{S}(u,v,t)/\partial u^n = \boldsymbol{C}_{n+4}(v,t) \\ & (n = 0,1,2) \end{aligned} \tag{1}$$

where $u = 0$ of the blending surface is the same as $u = u_1$ of the first primary surface $S_1(u,v,t)$, $u = 1$ of the blending surface is the same as $u = u_2$ of the second primary surface $S_2(u,v,t)$, $\partial^0 \boldsymbol{S}(u,v,t)/\partial u^0 = \boldsymbol{S}(u,v,t)$, $\boldsymbol{S}(u,v,t)$ has three components $S_x(u,v,t)$, $S_y(u,v,t)$ and $S_z(u,v,t)$, and $\boldsymbol{C_n}(v,t)$ $(n = 1,2,\ldots,6)$ also have three components $\boldsymbol{C}_{xn}(v,t)$, $\boldsymbol{C}_{yn}(v,t)$ and $\boldsymbol{C}_{zn}(v,t)$.

For example, the two primary surfaces are to be smoothly connected together where the parametric representations for the first (top) primary surface are

$$x = ae^t \sin 2\pi v \qquad y = be^{-t} \cos 2\pi v \qquad z = h_1 + h_2 u^2 \tag{2}$$

and the parametric representations for the second (bottom) primary surface are

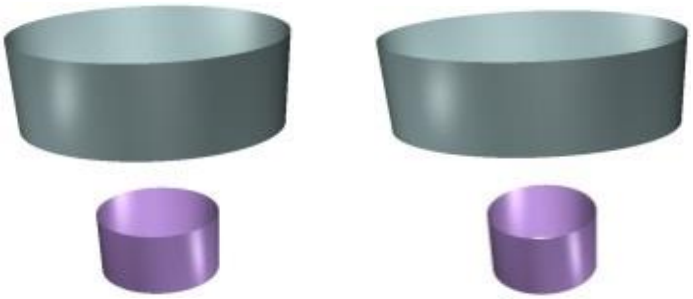
$$x = ce^{-t} \sin 2\pi v \qquad y = de^t \cos 2\pi v \qquad z = -h_3 u^3 \tag{3}$$

where a, b, c, d, h_1, h_2 and h_3 are known geometric parameters.

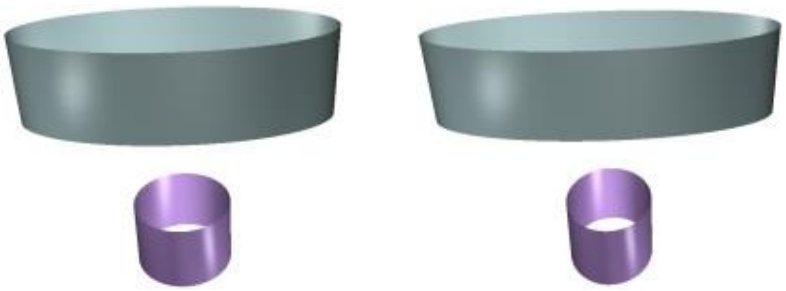
Setting the geometric parameters in Eqs. (2) and (3) to be: $a = 1.6$, $b = 1.0$, $c = 0.8$, $d = 0.6$, $h_1 = 2.0$, $h_2 = 3.0$, and $h_3 = 5.0$, the top primary surface between $u = 0.2$ and $u = 0.65$ and the bottom primary surface between $u = 0.3$ and $u = 0.6$ at the time instants $t = 0$, $t = 0.1$, $t = 0.2$, and $t = 0.3$ are depicted in Figure 1 where the top primary surfaces are obtained from Eq. (2) and the bottom primary surfaces are achieved from Eq. (3).

If the trimline is take to be $u = 0.2$ in Eq. (2) for the top primary surface and $u = 0.3$ in Eq. (3) for the bottom primary surface, the boundary curves $\boldsymbol{C}_1(v,t) = [ae^t \sin 2\pi v \quad be^{-t} \cos 2\pi v \quad h_1 + 0.04h_2]^T$ and $\boldsymbol{C}_4(v,t) = [ce^{-t} \sin 2\pi v \quad de^t \cos 2\pi v \quad -0.027h_3]^T$ are obtained. With Eqs. (2) and (3), the first partial derivatives $\partial x/\partial u$, $\partial y/\partial u$, and $\partial z/\partial u$, and the second partial derivatives $\partial^2 x/\partial u^2$, $\partial y^2/\partial u^2$, and $\partial^2 z/\partial u^2$ can be derived. Setting $u = 0.2$ for the partial derivatives from Eq. (2), $\boldsymbol{C}_2(v,t) = [0 \quad 0 \quad -0.4h_2]^T$ and $\boldsymbol{C}_3(v,t) = [0 \quad 0 \quad 2h_2]^T$ are obtained. Setting $u = 0.3$ for the partial derivatives from Eq. (3), $\boldsymbol{C}_5(v,t) = [0 \quad 0 \quad -0.27h_3]^T$ and $\boldsymbol{C}_6(v,t) = [0 \quad 0 \quad -1.8h_3]^T$ are obtained. Substituting $\boldsymbol{C}_1(v,t)$, $\boldsymbol{C}_2(v,t)$, $\boldsymbol{C}_3(v,t)$, $\boldsymbol{C}_4(v,t)$, $\boldsymbol{C}_5(v,t)$ and $\boldsymbol{C}_6(v,t)$ into (1), the boundary constraints (1) for this blending problem become

$$
\begin{aligned}
&u = 0\\
&S_x = ae^t \sin 2\pi v \qquad S_y = be^{-t} \cos 2\pi v \quad S_z = h_1 + 0.04h_2\\
&\frac{\partial S_x}{\partial u} = 0 \qquad \frac{\partial S_y}{\partial u} = 0 \qquad \frac{\partial S_z}{\partial u} = -0.4h_2\\
&\frac{\partial^2 S_x}{\partial u^2} = 0 \qquad \frac{\partial^2 S_y}{\partial u^2} = 0 \qquad \frac{\partial^2 S_z}{\partial u^2} = 2h_2\\
&u = 1\\
&S_x = ce^{-t} \sin 2\pi v \quad S_y = de^t \cos 2\pi v \quad S_z = -0.027h_3\\
&\frac{\partial S_x}{\partial u} = 0 \qquad \frac{\partial S_y}{\partial u} = 0 \qquad \frac{\partial S_z}{\partial u} = -0.27h_3\\
&\frac{\partial^2 S_x}{\partial u^2} = 0 \qquad \frac{\partial^2 S_y}{\partial u^2} = 0 \qquad \frac{\partial^2 S_z}{\partial u^2} = -1.8h_3
\end{aligned}
\tag{4}
$$



$t = 0$ $t = 0.1$



$t = 0.2$ $t = 0.3$

Figure 1: Primary (top and bottom) surfaces at t=0, t=0.1, t=0.2, and t=0.3.

Achieving both effective shape control of blending surfaces and exact satisfaction of blending boundary constraints is a difficult task. A vector-valued partial differential equation is ideal in achieving such an aim since its solution can satisfy blending boundary constraints exactly and the shape control parameters involving the vector-valued partial differential equation (5) below have a great influence on the shape of the blending surface but have no effect on blending boundary constraints. Here, a vector-valued partial differential equation involving a vector-valued unknown function $\boldsymbol{S}(u, v, t)$ consists of three partial differential equations involving three unknown component functions $S_x(u, v, t)$, $S_y(u, v, t)$ and $S_z(u, v, t)$, respectively. Since the coefficients involved in the vector-valued partial differential equation have strong influences on the shape of a blending surface, they are called shape control parameters.

Unlike in computational sciences and engineering where physically accurate simulations are required, in computer graphics, physics-based simulations, not physically accurate simulations, are popular. Even in computer-aided design and other engineering applications, physics-based blending surfaces are preferable to purely geometric blending surfaces since blending surfaces are in very small regions between two primary surfaces to be connected, the movements and deformations of blending surfaces are subjected to the constraints at trimlines from primary surfaces, and physics-based blending surfaces have a potential to give more reasonable shapes due to the consideration of physics.

Differential equations are used to describe the underlying physics in various fields of engineering and science. For example, fourth-order partial differential equations are used to describe bending deformations of thin elastic plates, sixth-order partial differential equations are used to characterize nonlinear wave dynamics etc., and sixth-order ordinary differential equations are used in astrophysics. Therefore, partial differential equation-based blending surfaces can be regarded as physics-based.

The solution of a vector-valued sixth order partial differential equation contains 6 unknown constants. They can be used to satisfy the 6 constraints of the position functions and the first and second partial derivatives in blending boundary constraints (1). Therefore, the following vector-valued sixth order partial differential equation is chosen for C^2 continuous surface blending. In the equation, γ, η, λ, and ρ are called shape control parameters since they have a big influence on the shapes of blending surfaces.

$$\left(\gamma \frac{\partial^6}{\partial u^6} + \eta \frac{\partial^6}{\partial u^4 \partial v^2} + \lambda \frac{\partial^6}{\partial u^2 \partial v^4} + \rho \frac{\partial^6}{\partial v^6}\right) \boldsymbol{S}(u, v, t) = 0 \tag{5}$$

Putting Eq. (5) and Eq. (1) together, the mathematical model of C^2 continuous surface blending of time-dependent varying parametric surfaces is obtained. When the time variable t in Eqs. (1) and (5) drop, the mathematical model of C^2 continuous surface blending of time-independent constant parametric surfaces is obtained.

3.2 Closed form solution-based approximate analytical approach

One of the advantages for partial differential equation (PDE)-based surface blending is both effective shape control and exact satisfaction of blending boundary constraints. However, it is not easy to solve partial differential Eq. (5) subjected to blending boundary constraints (1). In this section, Eq. (5) subjected to the blending boundary constraints (1) is solved with a closed form

solution and a closed form solution-based approximate analytical approach is developed to achieve surface blending with C^2 continuity. In what follows, we demonstrate how to use the unified mathematical model consisting of the partial differential equation (5) subjected to the blending boundary constraints (1) to solve time-independent surface blending problems. The obtained closed form solution-based approximate analytical approach is also applicable to time-dependent surface blending. This can be achieved by simply replacing $f_{sk}(v)(s = x, y, z; k = 1, 2, 3, \cdots, K_s)$ in this section with $f_{sk}(v, t)$ $(s = x, y, z; k = 1, 2, 3, \cdots, K_s)$.

After the time variable t in the partial different equation (5) and the blending boundary constraints (1) is dropped, and the partial differential equation (5) is converted into the following partial differential equation (7), and the blending boundary constraints (1) are changed into the blending boundary constraints (6) below.

After the conversion, elementary functions in the blending boundary constraints are identified. Taking advantage of the elementary functions identified from blending boundary constraints, the proposed approach first changes blending boundary constraints into a linear combination of the identified elementary functions. Accordingly, the functions for blending surfaces are constructed from these elementary functions, which transform sixth-order partial differential equations for C^2 surface blending into some sixth-order ordinary differential equations. The analytical solutions of these sixth-order ordinary differential equations subjected to corresponding blending boundary constraints are investigated.

With the developed analytical PDE-based method, time-independent C^2 continuous surface blending problems are solved. These surface blending examples indicate that the developed method is simple. It can be used to control the shape of blending surfaces and at the same time exactly satisfy C^2 continuous blending boundary constraints.

In the following subsections, the conversion of the mathematical model for C^2 continuous surface blending of time-independent constant parametric surfaces is first investigated. Then, a

closed form solution of the converted mathematical model. And finally, some examples are presented to demonstrate the applications of the developed closed form solution.

3.2.1 Conversion of mathematical model

Dropping the time variable *t* in Eq. (1), the blending boundary constraints for C^2 continuity surface blending of time-independent constant parametric surfaces can be formulated as

$$\begin{aligned} u = 0 \qquad & \partial^n s(u,v)/\partial u^n = B_{sn+1}(v) \\ u = 1 \qquad & \partial^n s(u,v)/\partial u^n = B_{sn+4}(v) \\ (s = x, y,\ z;\ n = 0, 1, 2) & \end{aligned} \tag{6}$$

where u and v are two parametric variables, $s(u,v)$ stands for the three components $x(u,v)$, $y(u,v)$ and $z(u,v)$ of a blending surface, $B_{s1}(v)$ and $B_{s4}(v)$ are the position components of boundary curves at the boundaries $u = 0$ and $u = 1$, respectively, $B_{s2}(v)$ and $B_{s5}(v)$ are the first partial derivatives and $B_{s3}(v)$ and $B_{s6}(v)$ are the second partial derivatives of the blending surface $\boldsymbol{S}(u,v) = [x(u,v) \quad y(u,v) \quad z(u,v)]^T$ with respect to the parametric variable u at the boundary curves, and

$$\begin{aligned} \partial^0 s(u,v)/\partial u^0 = s(u,v) \\ (s = x, y, z) \end{aligned} \tag{6a}$$

By dropping the time variable *t* in the vector-valued partial differential equation (5), the vector-valued partial differential equation involving the time variable *t* is converted into the following partial differential equations without the time variable *t*.

$$\begin{aligned} \left(\gamma \frac{\partial^6}{\partial u^6} + \eta \frac{\partial^6}{\partial u^4 \partial v^2} + \lambda \frac{\partial^6}{\partial u^2 \partial v^4} + \rho \frac{\partial^6}{\partial v^6}\right) s(u,v) = 0 \\ (s = x,\ y,\ z) \end{aligned} \tag{7}$$

The converted mathematical model of C^2 continuous blending between two time-independent constant parametric surfaces is obtained by putting Eq. (6) and Eq. (7) together. Its closed form solutions will be developed below.

The six equations for each component $s = x, y, z$ in the blending boundary constraints (6) can be decomposed into constants $d_{s0,n}$ $(s = x, y, z;\ n = 1,2,\cdots,6)$ and some elementary functions. These elementary functions can be divided into two groups: the first group has the differential property whose second derivatives are the same as the elementary functions themselves multiplied by a coefficient, and the second group does not have.

If all elementary functions in boundary constraints belong to the first group, the approach developed below can be used to obtain closed form solution of Eq. (7) subject to the boundary constraints (6) and provides the best tool to tackle such surface blending problems. If some elementary functions in boundary constraints belong to the second group, these elementary functions must be first transformed into Fourier series functions with each term having the differential property, i. e., their second derivatives are the same as themselves multiplied by a coefficient. Then, the approach developed below can be used to obtain closed form solution of Eq. (7) subjected to the constraints of these elementary functions.

After transforming the second group of elementary functions into Fourier series functions, all the functions $f_{sk}(v)(s = x, y, z; k = 1, 2, 3, \cdots, K_s)$ in the blending boundary constraints (6) have the differential property

$$\frac{\partial^2 f_{sk}(v)}{\partial v^2} = \xi_{sk} f_{sk}(v)$$

$$(s = x, y, z; k = 1, 2, 3, \cdots, K_s) \tag{8}$$

where ξ_{sk} is the coefficient obtained from the differentiation operation (8). For example, when $f_{sk}(v) = sin(2\pi v)$, $\frac{\partial^2 f_{sk}(v)}{\partial v^2} = -4\pi^2 sin(2\pi v)$. Therefore, $\xi_{sk} = -4\pi^2$.

Introducing the functions $f_{sk}(v)(s = x,\ y,\ z;\ k = 1, 2, 3, \cdots, K_s)$ into the blending boundary constraints and considering the constants $d_{s0,n}$ $(s = x,\ y,\ z;\ n = 1, 2, \cdots, 6)$, Eq. (6) can be changed into

$$u = 0 \qquad \partial^n s(u,v)/\partial u^n = d_{s0,n+1} + \sum_{k=1}^{K_s} d_{sk,n+1}\, f_{sk}(v)$$

$$u = 1 \qquad \partial^n s(u,v)/\partial u^n = d_{s0,n+4} + \sum_{k=1}^{K_s} d_{sk,n+4}\, f_{sk}(v)$$

$$(s = x, y, z; n = 0,1,2) \tag{9}$$

Since the blending surface should satisfy the blending boundary constraints at the boundaries $u = 0$ and $u = 1$, the blending surface functions should: (1) involve the functions $f_{sk}(v)$, and (2) have the functions of the parametric variable u only to satisfy the constraints of the constants $d_{s0,n}$ $(s = x, y, z;\ n = 1, 2, \cdots, 6)$ on the boundaries $u = 0$ and $u = 1$. Therefore, the blending surface functions can be constructed as

$$s(u,v) = G_{s0}(u) + \sum_{k=1}^{K_s} G_{sk}\,(u) f_{sk}(v)$$

$$(s = x,\ y,\ z) \tag{10}$$

where $G_{s0}(u)$ $(k = 0, 1, 2, \ldots, K_s)$ are unknown functions to be determined below.

Substituting Eq. (10) into Eq. (7), the partial differential equations given in Eq. (7) are transformed into the following ordinary differential equations

$$\gamma G_{s0}^{(6)}(u) + \sum_{k=1}^{K_s} \Big[\gamma G_{sk}^{(6)}(u) f_{sk}(v) + \eta G_{sk}^{(4)}(u) f_{sk}^{(2)}(v)$$

$$+\lambda G_{sk}^{(2)}(u) f_{sk}^{(4)}(v) + \rho G_{sk}(u) f_{sk}^{(6)}(v)\Big] = 0$$

$$(s = x, y, z) \tag{11}$$

where $f_{sk}^{(m)}(v) = \frac{d^m f_{sk}(v)}{dv^m}$ $(m = 2, 4, 6)$.

Considering the differential property (8), Eq. (11) can be changed into the following ordinary differential equations (ODEs)

$$G_{s0}^{(6)}(u) = 0$$

$$a_{s0}G_{sk}^{(6)}(u) + a_{s1k}G_{sk}^{(4)}(u) + a_{s2k}G_{sk}^{(2)}(u) + a_{s3k}G_{sk}(u) = 0$$

$$(s = x, y, z; k = 1, 2, 3, \cdots, K_s) \tag{12}$$

where

$$a_{s0} = \gamma \qquad a_{s1k} = \eta\xi_{sk}$$

$$a_{s2k} = \lambda\xi_{sk}^2 \qquad a_{s3k} = \rho\xi_{sk}^3$$

$$(s = x,\ y,\ z;\ k = 1,\ 2,\ 3, \cdots, K_s) \tag{13}$$

Substituting Eq. (10) into the blending boundary constraints (9), the following boundary constraints are obtained

$$u = 0 \qquad \partial^n G_{sk}(u)/\partial u^n = d_{sk,n+1}$$

$$u = 1 \qquad \partial^n G_{sk}(u)/\partial u^n = d_{sk,n+4}$$

$$(s = x, y, z;\ k = 0, 1, 2, 3, \cdots, K_s;\ \ n = 0, 1, 2) \tag{14}$$

where

$$\partial^0 G_{sk}(u)/\partial u^0 = G_{sk}(u)$$

$$(s = x, y, z;\ k = 0, 1, 2, 3, \cdots, K_s)$$

After the above treatment, the mathematical model consisting of Eqs. (7) and (6) is changed into the one consisting of Eqs. (12) and (14), and the original problem of solving the partial differential equations given in Eq. (7) subjected to the boundary constraints (6) is transformed into solving the ordinary differential equations given in Eq. (12) subject to the boundary constraints (14). In what follows, an exact analytical approach will be developed to obtain the closed form solutions of Eq. (12) subjected to Eq. (14).

3.2.2 Closed form solution of mathematical model

Integrating the first of Eq. (12) six times with respect to the parametric variable u, we obtain the unknown function $G_{s0}(u)$, which has the form below:

$$G_{s0}(u) = \sum_{j=0}^{5} C_{sj+1} u^j$$

$$(s = x, y, z) \tag{15}$$

where C_{sj+1} $(s = x, y, z; k = 0, 1, 2, \dots, 5)$ are unknown constants.

Substituting Eq. (15) into (14) and taking $k = 0$ in (14), the following equation is obtained.

$$C_{s1} = d_{s0,1}$$

$$C_{s2} = d_{s0,2}$$

$$2C_{s3} = d_{s0,3}$$

$$\sum_{j=0}^{5} C_{sj+1} = d_{s0,4}$$

$$\sum_{j=1}^{5} jC_{sj+1} = d_{s0,5}$$

$$\sum_{j=2}^{5} j(j-1)C_{sj+1} = d_{s0,6}$$

$$(s = x, y, z) \tag{16}$$

Solving the six linear algebra equations in Eq. (16), the six unknown constants C_{sj} $(j = 1, 2, \cdots, 6)$ are obtained. Substituting them back into Eq. (15), the exact analytical solution for the first ODE of Eq. (12) is found to be

$$G_{s0}(u) = \sum_{j=1}^{6} g_j(u) d_{s0,j}$$

$$(s = x, y, z) \tag{17}$$

where

$$g_1(u) = 1 - 10u^3 + 15u^4 - 6u^5$$

$$g_2(u) = (1 - 6u^2 + 8u^3 - 3u^4)u$$

$$g_3(u) = (0.5 - 1.5u + 1.5u^2 - 0.5u^3)u^2$$

$$g_4(u) = (10 - 15u + 6u^2)u^3$$

$$g_5(u) = (-4 + 7u - 3u^2)u^3$$

$$g_6(u) = (0.5 - u + 0.5u^2)u^3 \tag{18}$$

A popular method in solving linear ordinary differential equations is to transform them into characteristic algebraic equations. This can be achieved by taking the unknown function involved in a linear ordinary differential equation to be an exponential function with the independent variable multiplied by an undetermined coefficient. With this method, the unknown function $G_{sk}(u)$ involved in the second of Eq. (12) can be taken to be

$$G_{sk}(u) = e^{r_{sk}u} \tag{19}$$

For clarity, a_0, a_1, a_2, and a_3 are used to indicate a_{s0}, a_{s1k}, a_{s2k}, and a_{s3k} in Eq. (12), respectively. Then, Eq. (19) is substituted into the second of Eq. (12) to obtain the following nonlinear algebra equation

$$r_{sk}^6 + a_4 r_{sk}^4 + a_5 r_{sk}^2 + a_6 = 0 \tag{20}$$

where

$$a_{j+3} = a_j / a_0$$

$$(j = 1,2,3) \tag{21}$$

Eq. (20) can be changed into

$$(r_{sk}^4 + b_1 r_{sk}^2 + b_2)(r_{sk}^2 + b_3) = 0 \tag{22}$$

Comparing Eq. (22) with Eq. (20), the following equations are obtained

$$b_1 + b_3 = a_4$$

$$b_2 + b_1 b_3 = a_5$$

$$b_2 b_3 = a_6 \tag{23}$$

The three algebra equations given in Eq. (23) can be changed into one cubic equation below

$$b_3^3 - a_4 b_3^2 + a_5 b_3 - a_6 = 0 \tag{24}$$

Some existing research studies (Zhao et al., 2011; Okereke et al., 2014) have discussed how to solve a cubic equation. Here the method given in the link[2] is used to solve a cubic equation. With the root formulas given in the link, the three roots below of the cubic equation (24) are obtained below.

$$b_{3,1} = a_4/3 + (P + T)$$

$$b_{3,2} = a_4/3 - \left[(P + T) - i\sqrt{3}(P - T)\right]/2$$

$$b_{3,3} = a_4/3 - \left[(P + T) + i\sqrt{3}(P - T)\right]/2 \tag{25}$$

where the second subscript 1, 2, and 3 indicates the first, second and third roots, respectively, *i* is an imaginary number, and

$$P = \sqrt[3]{R + \sqrt{D}} \qquad T = \sqrt[3]{R - \sqrt{D}}$$

$$D = Q^3 + R^2 \qquad \mathrm{Q} = (3a_5 - a_4^2)/9$$

$$R = (-9a_4 a_5 + 27a_6 + 2a_4^3)/54 \tag{26}$$

Substituting Eq. (25) back into Eq. (23), $b_{1,j}$ and $b_{2,j}$ $(j = 1, 2, 3)$ can be obtained, which correspond to $b_{3,j}$ $(j = 1, 2, 3)$, respectively. Here $b_{3,1}$ is taken as an example to demonstrate how to determine $b_{1,1}$ and $b_{2,1}$, and the corresponding solution to the second ODE of Eq. (12). Introducing the first of Eq. (25) into (23) and solving for b_1 and b_2, and $b_{2,1}$ are obtain below.

[2] https://mathworld.wolfram.com/CubicFormula.html

$$b_{1,1} = 2a_4/3 - (P+T)$$

$$b_{2,1} = a_5 - [2a_4^2 + 3a_4(P+T) - 9(P+T)^2]/9 \qquad (27)$$

Once again for clarity, b_1, b_2 and b_3 are used to indicate $b_{1,1}$, $b_{2,1}$ and $b_{3,1}$ in the following mathematical operations.

From Eq. (22), the following two nonlinear algebra equations are obtained.

$$r_{sk}^2 + b_3 = 0$$

$$r_{sk}^4 + b_1 r_{sk}^2 + b_2 = 0 \qquad (28)$$

Substituting $b_3 = b_{3,1}$ into the first of Eq. (28), the following two roots $(r_{sk})_{1,2}$ are obtained.

$$(r_{sk})_{1,2} = \pm q_0 \qquad \text{for } b_3 < 0$$

$$(r_{sk})_{1,2} = \pm i q_0 \qquad \text{for } b_3 > 0$$

$$q_0 = \sqrt{|b_3|} \qquad (29)$$

Since $f_{sk}(v)$ has the differential property (8), $a_3 \neq 0$ according to Eq. (8). From Eq. (21), $a_6 \neq 0$. Therefore, $b_3 \neq 0$ according to Eq. (23).

Substituting $b_1 = b_{1,1}$ and $b_2 = b_{2,1}$ into the second of Eq. (28) and solving for r_{sk}^2, the following result is obtained.

$$(r_{sk}^2)_{1,2} = \left(-b_1 \pm \sqrt{b_1^2 - 4b_2}\right)/2 \qquad (30)$$

According to three different cases below, the four roots can be obtained from Eq. (30).

Case 1: $b_1^2 = 4b_2$. For this case, $(r_{sk}^2)_{1,2} = -b_1/2$ and there are three different situations, i. e., $b_1 < 0$, $b_1 = 0$, and $b_1 > 0$. According to $b_1^2 = 4b_2$ $b_1 = 0$ will lead to $b_2 = 0$. According to Eq. (23), $b_1 = 0$ and $b_2 = 0$ will lead to $a_5 = a_6 = 0$. Therefore, b_1 cannot be zero. The roots for the remaining two situations $b_1 < 0$ and $b_1 > 0$ can be summarized as

$$(r_{sk})_{3,4,5,6} = \pm q_1 \qquad \text{for} \qquad b_1 < 0$$

$$(r_{sk})_{3,4,5,6} = \pm i q_1 \qquad \text{for} \qquad b_1 > 0$$

$$q_1 = \sqrt{|b_1|/2} \qquad (31)$$

From the two roots in Eq. (29) and the four roots in Eq. (31), the four different solutions of the second ordinary differential equation of Eq. (12) are indicated below.

1) For $b_3 < 0$, $b_1^2 = 4b_2$, and $b_1 < 0$, $(r_{sk})_{1,2} = \pm q_0$ according to Eq. (29) and $(r_{sk})_{3,4,5,6} = \pm q_1$ according to Eq. (31). The solution of the second ODE of Eq. (12) is

$$G_{sk}(u) = C_{s1k}e^{q_0 u} + C_{s2k}e^{-q_0 u} + (C_{s3k} + C_{s4k}u)$$
$$e^{q_1 u} + (C_{s5k} + C_{s6k}u)e^{-q_1 u} \quad (32)$$

$$\partial G_{sk}(u)/\partial u = q_0 C_{s1k}e^{q_0 u} - q_0 C_{s2k}e^{-q_0 u} + q_1 C_{s3k}e^{q_1 u} + C_{s4k}(1 +$$
$$q_1 u)e^{q_1 u} - q_1 C_{s5k}e^{-q_1 u} + C_{s6k}(1 - q_1 u)e^{-q_1 u} \quad (32a)$$

$$\partial^2 G_{sk}(u)/\partial u^2 = q_0^2 C_{s1k}e^{q_0 u} + q_0^2 C_{s2k}e^{-q_0 u} + q_1^2 C_{s3k}e^{q_1 u} + C_{s4k}q_1(2 + q_1 u)e^{q_1 u} +$$
$$q_1^2 C_{s5k}e^{-q_1 u} - C_{s6k}q_1(2 - q_1 u)e^{-q_1 u} \quad (32b)$$

Substituting Eqs. (32), (32a), and (32b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s2k} + C_{s3k} + C_{s5k} = d_{sk,1}$$

$$q_0 C_{s1k} - q_0 C_{s2k} + q_1 C_{s3k} + C_{s4k} - q_1 C_{s5k} + C_{s6k} = d_{sk,2}$$

$$q_0^2 C_{s1k} + q_0^2 C_{s2k} + q_1^2 C_{s3k} + 2C_{s4k}q_1 + q_1^2 C_{s5k} - 2q_1 C_{s6k} = d_{sk,3}$$

$$C_{s1k}e^{q_0} + C_{s2k}e^{-q_0} + (C_{s3k} + C_{s4k})e^{q_1} + (C_{s5k} + C_{s6k})e^{-q_1} = d_{sk,4}$$

$$q_0 C_{s1k}e^{q_0} - q_0 C_{s2k}e^{-q_0} + q_1 C_{s3k}e^{q_1} + C_{s4k}(1 + q_1)e^{q_1} - q_1 C_{s5k}e^{-q_1} + C_{s6k}(1 -$$
$$q_1)e^{-q_1} = d_{sk,5}$$

$$q_0^2 C_{s1k}e^{q_0} + q_0^2 C_{s2k}e^{-q_0} + q_1^2 C_{s3k}e^{q_1} + C_{s4k}q_1(2 + q_1)e^{q_1} + q_1^2 C_{s5k}e^{-q_1} - C_{s6k}q_1(2 -$$
$$q_1)e^{-q_1} = d_{sk,6} \quad (32c)$$

Solving Eq. (32c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (32), $G_{sk}(u)$ is obtained.

2) For $b_3 < 0$, $b_1^2 = 4b_2$, and $b_1 > 0$, $(r_{sk})_{1,2} = \pm q_0$ according to Eq. (29) and $(r_{sk})_{3,4,5,6} = \pm iq_1$ according to Eq. (31). The solution of the second ODE of Eq. (12) is

$$G_{sk}(u) = C_{s1k}e^{q_0 u} + C_{s2k}e^{-q_0 u} + (C_{s3k} + C_{s4k}u)$$

$$cos(q_1u) + (C_{s5k} + C_{s6k}u)\, sin(q_1u) \tag{33}$$

$$\partial G_{sk}(u)/\partial u = q_0C_{s1k}e^{q_0u} - q_0C_{s2k}e^{-q_0u} - q_1C_{s3k}\, sin(q_1u) + C_{s4k}[cos(q_1u) - q_1u\, sin(q_1u)] + q_1C_{s5k}\, cos(q_1u) + C_{s6k}\,[sin(q_1u) + q_1u\, cos(q_1u)] \tag{33a}$$

$$\partial^2 G_{sk}(u)/\partial u^2 = q_0^2C_{s1k}e^{q_0u} + q_0^2C_{s2k}e^{-q_0u} - q_1^2C_{s3k}\, cos(q_1u) - C_{s4k}[2q_1\, sin(q_1u) + q_1^2u\, cos(q_1u)] - q_1^2C_{s5k}\, sin(q_1u) + C_{s6k}[2\, q_1\, cos(q_1u - q_1^2u\, sin(q_1u)] \tag{33b}$$

Substituting Eqs. (33), (33a), and (33b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s2k} + C_{s3k} = d_{sk,1}$$

$$q_0C_{s1k} - q_0C_{s2k} + C_{s4k} + q_1C_{s5k} = d_{sk,2}$$

$$q_0^2C_{s1k} + q_0^2C_{s2k} - q_1^2C_{s3k} + 2q_1C_{s6k} = d_{sk,3}$$

$$C_{s1k}e^{q_0} + C_{s2k}e^{-q_0} + (C_{s3k} + C_{s4k})\, cos(q_1) + (C_{s5k} + C_{s6k})\, sin(q_1) = d_{sk,4}$$

$$q_0C_{s1k}e^{q_0} - q_0C_{s2k}e^{-q_0} - q_1C_{s3k}\, sin(q_1) + C_{s4k}[cos(q_1) - q_1\, sin(q_1)] + q_1C_{s5k}\, cos(q_1) + C_{s6k}\,[sin(q_1) + q_1\, cos(q_1)] = d_{sk,5}$$

$$q_0^2C_{s1k}e^{q_0} + q_0^2C_{s2k}e^{-q_0} - q_1^2C_{s3k}\, cos(q_1) - C_{s4k}[2q_1\, sin(q_1) + q_1^2\, cos(q_1)] - q_1^2C_{s5k}\, sin(q_1) + C_{s6k}[2q_1\, cos(q_1) - q_1^2\, sin(q_1)] = d_{sk,6} \tag{33c}$$

Solving Eq. (33c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (33), $G_{sk}(u)$ is obtained.

3) For $b_3 > 0$, $b_1^2 = 4b_2$, and $b_1 < 0$, $(r_{sk})_{1,2} = \pm iq_0$ according to Eq. (29) and $(r_{sk})_{3,4,5,6} = \pm q_1$ according to Eq. (31). The solution of the second ODE of Eq. (12) is

$$G_{sk}(u) = C_{s1k}\, cos(q_0u) + C_{s2k}\, sin(q_0u) + (C_{s3k} + C_{s4k}u)\, e^{q_1u} + (C_{s5k} + C_{s6k}u)e^{-q_1u} \tag{34}$$

$$\partial G_{sk}(u)/\partial u = -q_0C_{s1k}\, sin(q_0u) + q_0C_{s2k}\, cos(q_0u) + q_1C_{s3k}e^{q_1u} + C_{s4k}(1 + q_1u)e^{q_1u} - q_1C_{s5k}e^{-q_1u} + C_{s6k}(1 - q_1u)e^{-q_1u} \tag{34a}$$

$$\partial^2 G_{sk}(u)/\partial u^2 = -q_0^2C_{s1k}\cos(q_0u) - q_0^2C_{s2k}\, sin(q_0u) + q_1^2C_{s3k}e^{q_1u} + C_{s4k}q_1(2 + q_1u)e^{q_1u} + q_1^2C_{s5k}e^{-q_1u} - C_{s6k}q_1(2 - q_1u)e^{-q_1u} \tag{34b}$$

Substituting Eqs. (34), (34a), and (34b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s3k} + C_{s5k} = d_{sk,1}$$

$$q_0 C_{s2k} + q_1 C_{s3k} + C_{s4k} - q_1 C_{s5k} + C_{s6k} = d_{sk,2}$$

$$-q_0^2 C_{s1k} + q_1^2 C_{s3k} + 2q_1 C_{s4k} + q_1^2 C_{s5k} - 2q_1 C_{s6k} = d_{sk,3}$$

$$C_{s1k}\, cos(\, q_0) + C_{s2k}\, sin(\, q_0) + (C_{s3k} + C_{s4k}) e^{q_1} + (C_{s5k} + C_{s6k}) e^{-q_1} = d_{sk,4}$$

$$-q_0 C_{s1k}\, sin(\, q_0) + q_0 C_{s2k}\, cos(\, q_0) + q_1 C_{s3k} e^{q_1} + C_{s4k}(1 + q_1) e^{q_1} - q_1 C_{s5k} e^{-q_1} + C_{s6k}(1 - q_1) e^{-q_1} = d_{sk,5}$$

$$-q_0^2 C_{s1k} \cos(q_0) - q_0^2 C_{s2k}\, sin(\, q_0) + q_1^2 C_{s3k} e^{q_1} + C_{s4k} q_1 (2 + q_1) e^{q_1} + q_1^2 C_{s5k} e^{-q_1} - C_{s6k} q_1 (2 - q_1) e^{-q_1} = d_{sk,6} \quad (34c)$$

Solving Eq. (34c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (34), $G_{sk}(u)$ isobtained.

4) For $b_3 > 0$, $b_1^2 = 4b_2$, and $b_1 > 0$, $(r_{sk})_{1,2} = \pm i q_0$ according to Eq. (29) and $(r_{sk})_{3,4,5,6} = \pm i q_1$ according to Eq. (31). The solution of the second ODE of Eq. (12) is

$$G_{sk}(u) = C_{s1k}\, cos(\, q_0 u) + C_{s2k}\, sin(\, q_0 u) + (C_{s3k} + C_{s4k} u)\, cos(\, q_1 u) + (C_{s5k} + C_{s6k} u)\, sin(\, q_1 u) \quad (35)$$

$$\partial G_{sk}(u)/\partial u = -q_0 C_{s1k}\, sin(\, q_0 u) + q_0 C_{s2k}\, cos(\, q_0 u) - q_1 C_{s3k}\, sin(\, q_1 u) + C_{s4k} [cos(\, q_1 u) - q_1 u\, sin(\, q_1 u)] + q_1 C_{s5k}\, cos(\, q_1 u) + C_{s6k} [sin(\, q_1 u) + q_1 u\, cos(\, q_1 u)] \quad (35a)$$

$$\partial^2 G_{sk}(u)/\partial u^2 = -q_0^2 C_{s1k}\, cos(\, q_0 u) - q_0^2 C_{s2k}\, sin(\, q_0 u) - q_1^2 C_{s3k}\, cos(\, q_1 u) - C_{s4k} q_1 [2\, sin(\, q_1 u) + q_1 u\, cos(\, q_1 u)] - q_1^2 C_{s5k}\, sin(\, q_1 u) + C_{s6k} q_1 [2 cos(\, q_1 u) - q_1 u\, sin(\, q_1 u)] \quad (35b)$$

Substituting Eqs. (35), (35a), and (35b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s3k} = d_{sk,1}$$

$$q_0 C_{s2k} + C_{s4k} + q_1 C_{s5k} = d_{sk,2}$$

$$-q_0^2 C_{s1k} - q_1^2 C_{s3k} + 2C_{s6k} q_1 = d_{sk,3}$$

$$C_{s1k}\, cos(\, q_0) + C_{s2k}\, sin(\, q_0) + (C_{s3k} + C_{s4k})\, cos(\, q_1) + (C_{s5k} + C_{s6k})\, sin(\, q_1) = d_{sk,4}$$

$$-q_0 C_{s1k}\, sin(\, q_0) + q_0 C_{s2k}\, cos(\, q_0) - q_1 C_{s3k}\, sin(\, q_1) + C_{s4k}[cos(\, q_1) - q_1\, sin(\, q_1)] + q_1 C_{s5k}\, cos(\, q_1) + C_{s6k}[sin(\, q_1) + q_1\, cos(\, q_1)] = d_{sk,5}$$

$$-q_0^2 C_{s1k}\, cos(\, q_0) - q_0^2 C_{s2k}\, sin(\, q_0) - q_1^2 C_{s3k}\, cos(\, q_1) - C_{s4k} q_1 [2\, sin(\, q_1) + q_1\, cos(\, q_1)] - q_1^2 C_{s5k}\, sin(\, q_1) + C_{s6k} q_1 [2\, cos(\, q_1) - q_1\, sin(\, q_1)] = d_{sk,6} \qquad (35c)$$

Solving Eq. (35c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (35), $G_{sk}(u)$ is obtained.

Case 2: $b_1^2 > 4b_2$. For this case, there are six different situations, i. e., $-b_1 \pm \sqrt{b_1^2 - 4b_2} < 0$, $-b_1 \pm \sqrt{b_1^2 - 4b_2} = 0$, and $-b_1 \pm \sqrt{b_1^2 - 4b_2} > 0$. It is obvious that $-b_1 \pm \sqrt{b_1^2 - 4b_2}$ cannot be zero since $-b_1 \pm \sqrt{b_1^2 - 4b_2} = 0$ will lead to $b_2 = 0$ and $a_6 = 0$ according to Eq. (23). Therefore, only $-b_1 \pm \sqrt{b_1^2 - 4b_2} < 0$ and $-b_1 \pm \sqrt{b_1^2 - 4b_2} > 0$ are considered.

The roots of Eq. (30) for the remaining four situations $-b_1 \pm \sqrt{b_1^2 - 4b_2} < 0$ and $-b_1 \pm \sqrt{b_1^2 - 4b_2} > 0$ can be summarized as

$$\text{For} \quad b_1 < -\sqrt{b_1^2 - 4b_2}, \qquad (r_{sk})_{3,4} = \pm q_2 \qquad (r_{sk})_{5,6} = \pm q_3$$

$$\text{For} \quad -\sqrt{b_1^2 - 4b_2} < b_1 < \sqrt{b_1^2 - 4b_2}, \qquad (r_{sk})_{3,4} = \pm q_2 \qquad (r_{sk})_{5,6} = \pm i q_3$$

$$\text{For} \quad b_1 > \sqrt{b_1^2 - 4b_2}, \qquad (r_{sk})_{3,4} = \pm i q_2 \qquad (r_{sk})_{5,6} = \pm i q_3 \qquad (36)$$

where

$$q_2 = \sqrt{\left| -b_1 + \sqrt{b_1^2 - 4b_2} \right| / 2}$$

$$q_3 = \sqrt{\left|-b_1 - \sqrt{b_1^2 - 4b_2}\right|/2} \tag{37}$$

From the two roots in Eq. (29) and the four roots in Eq. (36), the six different solutions of the second ordinary differential equation of Eq. (12) as obtained below.

5) For $b_3 < 0$, $b_1^2 > 4b_2$, and $b_1 < -\sqrt{b_1^2 - 4b_2}$, $(r_{sk})_{1,2} = \pm q_0$ according to Eq. (29), and $(r_{sk})_{3,4} = \pm q_2$ and $(r_{sk})_{5,6} = \pm q_3$ according to Eq. (36). The solution of the second ODE in Eq. (12) is

$$G_{sk}(u) = C_{s1k}e^{q_0 u} + C_{s2k}e^{-q_0 u} + C_{s3k}e^{q_2 u} + C_{s4k}e^{-q_2 u} + C_{s5k}e^{q_3 u} + C_{s6k}e^{-q_3 u} \tag{38}$$

$$\partial G_{sk}(u)/\partial u = q_0 C_{s1k}e^{q_0 u} - q_0 C_{s2k}e^{-q_0 u} + q_2 C_{s3k}e^{q_2 u} - q_2 C_{s4k}e^{-q_2 u} + q_3 C_{s5k}e^{q_3 u} - q_3 C_{s6k}e^{-q_3 u} \tag{38a}$$

$$\partial^2 G_{sk}(u)/\partial u^2 = q_0^2 C_{s1k}e^{q_0 u} + q_0^2 C_{s2k}e^{-q_0 u} + q_2^2 C_{s3k}e^{q_2 u} + q_2^2 C_{s4k}e^{-q_2 u} + q_3^2 C_{s5k}e^{q_3 u} + q_3^2 C_{s6k}e^{-q_3 u} \tag{38b}$$

Substituting Eqs. (38), (38a), and (38b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s2k} + C_{s3k} + C_{s4k} + C_{s5k} + C_{s6k} = d_{sk,1}$$

$$q_0 C_{s1k} - q_0 C_{s2k} + q_2 C_{s3k} - q_2 C_{s4k} + q_3 C_{s5k} - q_3 C_{s6k} = d_{sk,2}$$

$$q_0^2 C_{s1k} + q_0^2 C_{s2k} + q_2^2 C_{s3k} + q_2^2 C_{s4k} + q_3^2 C_{s5k} + q_3^2 C_{s6k} = d_{sk,3}$$

$$C_{s1k}e^{q_0} + C_{s2k}e^{-q_0} + C_{s3k}e^{q_2} + C_{s4k}e^{-q_2} + C_{s5k}e^{q_3} + C_{s6k}e^{-q_3} = d_{sk,4}$$

$$q_0 C_{s1k}e^{q_0} - q_0 C_{s2k}e^{-q_0} + q_2 C_{s3k}e^{q_2} - q_2 C_{s4k}e^{-q_2} + q_3 C_{s5k}e^{q_3} - q_3 C_{s6k}e^{-q_3} = d_{sk,5}$$

$$q_0^2 C_{s1k}e^{q_0} + q_0^2 C_{s2k}e^{-q_0} + q_2^2 C_{s3k}e^{q_2} + q_2^2 C_{s4k}e^{-q_2} + q_3^2 C_{s5k}e^{q_3} + q_3^2 C_{s6k}e^{-q_3} = d_{sk,6} \tag{38c}$$

Solving Eq. (38c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (38), $G_{sk}(u)$ is obtained.

6) For $b_3 < 0$, $b_1^2 > 4b_2$, and $-\sqrt{b_1^2 - 4b_2} < b_1 < \sqrt{b_1^2 - 4b_2}$, $(r_{sk})_{1,2} = \pm q_0$ according to Eq. (29), and $(r_{sk})_{3,4} = \pm q_2$ and $(r_{sk})_{5,6} = \pm i q_3$ according to Eq. (36). The solution of the second ODE of Eq. (12) is

$$G_{sk}(u) = C_{s1k}e^{q_0 u} + C_{s2k}e^{-q_0 u} + C_{s3k}e^{q_2 u} + C_{s4k}e^{-q_2 u} + C_{s5k}\,cos(q_3 u) + C_{s6k}\,sin(q_3 u) \quad (39)$$

$$\partial G_{sk}(u)/\partial u = q_0 C_{s1k}e^{q_0 u} - q_0 C_{s2k}e^{-q_0 u} + q_2 C_{s3k}e^{q_2 u} - q_2 C_{s4k}e^{-q_2 u} - q_3 C_{s5k}\,sin(q_3 u) + q_3 C_{s6k}\,cos(q_3 u) \quad (39a)$$

$$\partial^2 G_{sk}(u)/\partial u^2 = q_0^2 C_{s1k}e^{q_0 u} + q_0^2 C_{s2k}e^{-q_0 u} + q_2^2 C_{s3k}e^{q_2 u} + q_2^2 C_{s4k}e^{-q_2 u} - q_3^2 C_{s5k}\,cos(q_3 u) - q_3^2 C_{s6k}\,sin(q_3 u) \quad (39b)$$

Substituting Eqs. (39), (39a), and (39b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s2k} + C_{s3k} + C_{s4k} + C_{s5k} = d_{sk,1}$$

$$q_0 C_{s1k} - q_0 C_{s2k} + q_2 C_{s3k} - q_2 C_{s4k} + q_3 C_{s6k} = d_{sk,2}$$

$$q_0^2 C_{s1k} + q_0^2 C_{s2k} + q_2^2 C_{s3k} + q_2^2 C_{s4k} - q_3^2 C_{s5k} = d_{sk,3}$$

$$C_{s1k}e^{q_0} + C_{s2k}e^{-q_0} + C_{s3k}e^{q_2} + C_{s4k}e^{-q_2} + C_{s5k}\,cos(q_3) + C_{s6k}\,sin(q_3) = d_{sk,4}$$

$$q_0 C_{s1k}e^{q_0} - q_0 C_{s2k}e^{-q_0} + q_2 C_{s3k}e^{q_2} - q_2 C_{s4k}e^{-q_2} - q_3 C_{s5k}\,sin(q_3) + q_3 C_{s6k}\,cos(q_3) = d_{sk,5}$$

$$q_0^2 C_{s1k}e^{q_0} + q_0^2 C_{s2k}e^{-q_0} + q_2^2 C_{s3k}e^{q_2} + q_2^2 C_{s4k}e^{-q_2} - q_3^2 C_{s5k}\,cos(q_3) - q_3^2 C_{s6k}\,sin(q_3) = d_{sk,6} \quad (39c)$$

Solving Eq. (39c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (39), $G_{sk}(u)$ is obtained.

7) For $b_3 < 0$, $b_1^2 > 4b_2$, and $b_1 > \sqrt{b_1^2 - 4b_2}$, $(r_{sk})_{1,2} = \pm q_0$ according to Eq. (29), and $(r_{sk})_{3,4} = \pm i q_2$ and $(r_{sk})_{5,6} = \pm i q_3$ according to Eq. (36). The solution of the second ODE in Eq. (12) is

$$G_{sk}(u) = C_{s1k}e^{q_0 u} + C_{s2k}e^{-q_0 u} + C_{s3k}\,cos(\,q_2 u)$$
$$+ C_{s4k}\,sin(\,q_2 u) + C_{s5k}\,cos(\,q_3 u) + C_{s6k}\,sin(\,q_3 u) \quad (40)$$

$$\partial G_{sk}(u)/\partial u = q_0 C_{s1k}e^{q_0 u} - q_0 C_{s2k}e^{-q_0 u} - q_2 C_{s3k}\,sin(\,q_2 u)$$
$$+ q_2 C_{s4k}\,cos(\,q_2 u) - q_3 C_{s5k}\sin\,(q_3 u) + q_3 C_{s6k}\,cos(\,q_3 u) \quad (40a)$$

$$\partial^2 G_{sk}(u)/\partial u^2 = q_0^2 C_{s1k}e^{q_0 u} + q_0^2 C_{s2k}e^{-q_0 u} - q_2^2 C_{s3k}\,cos(\,q_2 u)$$
$$- q_2^2 C_{s4k}\,sin(\,q_2 u) - q_3^2 C_{s5k}\cos\,(q_3 u) - q_3^2 C_{s6k}\,sin(\,q_3 u) \quad (40b)$$

Substituting Eqs. (40), (40a), and (40b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s2k} + C_{s3k} + C_{s5k} = d_{sk,1}$$

$$q_0 C_{s1k} - q_0 C_{s2k} + q_2 C_{s4k} + q_3 C_{s6k} = d_{sk,2}$$

$$q_0^2 C_{s1k} + q_0^2 C_{s2k} - q_2^2 C_{s3k} - q_3^2 C_{s5k} = d_{sk,3}$$

$$C_{s1k}e^{q_0} + C_{s2k}e^{-q_0} + C_{s3k}\,cos(\,q_2) + C_{s4k}\,sin(\,q_2) + C_{s5k}\,cos(\,q_3) + C_{s6k}\,sin(\,q_3) = d_{sk,4}$$

$$q_0 C_{s1k}e^{q_0} - q_0 C_{s2k}e^{-q_0} - q_2 C_{s3k}\,sin(\,q_2) + q_2 C_{s4k}\,cos(\,q_2) - q_3 C_{s5k}\sin\,(q_3)$$
$$+ q_3 C_{s6k}\,cos(\,q_3) = d_{sk,5}$$

$$q_0^2 C_{s1k}e^{q_0} + q_0^2 C_{s2k}e^{-q_0} - q_2^2 C_{s3k}\,cos(\,q_2) - q_2^2 C_{s4k}\,sin(\,q_2) - q_3^2 C_{s5k}\cos\,(q_3)$$
$$- q_3^2 C_{s6k}\,sin(\,q_3) = d_{sk,6} \quad (40c)$$

Solving Eq. (40c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (40), $G_{sk}(u)$ is obtained.

8) For $b_3 > 0$, $b_1^2 > 4b_2$, and $b_1 < -\sqrt{b_1^2 - 4b_2}$, $(r_{sk})_{1,2} = \pm i q_0$ according to Eq. (29), and $(r_{sk})_{3,4} = \pm q_2$ and $(r_{sk})_{5,6} = \pm q_3$ according to Eq. (36). The solution of the second ODE in Eq. (12) is

$$G_{sk}(u) = C_{s1k}\,cos(\,q_0 u) + C_{s2k}\,sin(\,q_0 u) + C_{s3k}e^{q_2 u}$$
$$+ C_{s4k}e^{-q_2 u} + C_{s5k}e^{q_3 u} + C_{s6k}e^{-q_3 u} \quad (41)$$

$$\partial G_{sk}(u)/\partial u = -q_0 C_{s1k}\,sin(\,q_0 u) + q_0 C_{s2k}\,cos(\,q_0 u) + q_2 C_{s3k}e^{q_2 u}$$

$$- q_2 C_{s4k} e^{-q_2 u} + q_3 C_{s5k} e^{q_3 u} - q_3 C_{s6k} e^{-q_3 u} \quad (41a)$$

$$\partial^2 G_{sk}(u)/\partial u^2 = -q_0^2 C_{s1k}\, cos(\, q_0 u) - q_0^2 C_{s2k}\, sin(\, q_0 u) + q_2^2 C_{s3k} e^{q_2 u} + q_2^2 C_{s4k} e^{-q_2 u} + q_3^2 C_{s5k} e^{q_3 u} + q_3^2 C_{s6k} e^{-q_3 u} \quad (41b)$$

Substituting Eqs. (41), (41a), and (41b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s3k} + C_{s4k} + C_{s5k} + C_{s6k} = d_{sk,1}$$

$$q_0 C_{s2k} + q_2 C_{s3k} - q_2 C_{s4k} + q_3 C_{s5k} - q_3 C_{s6k} = d_{sk,2}$$

$$-q_0^2 C_{s1k} + q_2^2 C_{s3k} + q_2^2 C_{s4k} + q_3^2 C_{s5k} + q_3^2 C_{s6k} = d_{sk,3}$$

$$C_{s1k}\, cos(\, q_0) + C_{s2k}\, sin(\, q_0) + C_{s3k} e^{q_2} + C_{s4k} e^{-q_2} + C_{s5k} e^{q_3} + C_{s6k} e^{-q_3} = d_{sk,4}$$

$$-q_0 C_{s1k}\, sin(\, q_0) + q_0 C_{s2k}\, cos(\, q_0) + q_2 C_{s3k} e^{q_2} - q_2 C_{s4k} e^{-q_2} + q_3 C_{s5k} e^{q_3} - q_3 C_{s6k} e^{-q_3} = d_{sk,5}$$

$$-q_0^2 C_{s1k}\, cos(\, q_0) - q_0^2 C_{s2k}\, sin(\, q_0) + q_2^2 C_{s3k} e^{q_2} + q_2^2 C_{s4k} e^{-q_2} + q_3^2 C_{s5k} e^{q_3} + q_3^2 C_{s6k} e^{-q_3} = d_{sk,6} \quad (41c)$$

Solving Eq. (41c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (41), $G_{sk}(u)$ is obtained.

9) For $b_3 > 0$, $b_1^2 > 4b_2$, and $-\sqrt{b_1^2 - 4b_2} < b_1 < \sqrt{b_1^2 - 4b_2}$, $(r_{sk})_{1,2} = \pm i q_0$ according to Eq. (29), and $(r_{sk})_{3,4} = \pm q_2$ and $(r_{sk})_{5,6} = \pm i q_3$ according to Eq. (36). The solution of the second ODE in Eq. (12) is

$$G_{sk}(u) = C_{s1k}\, cos(\, q_0 u) + C_{s2k}\, sin(\, q_0 u) + C_{s3k} e^{q_2 u} + C_{s4k} e^{-q_2 u} + C_{s5k}\, cos(\, q_3 u) + C_{s6k}\, sin(\, q_3 u) \quad (42)$$

$$\partial G_{sk}(u)/\partial u = -q_0 C_{s1k}\, sin(\, q_0 u) + q_0 C_{s2k}\, cos(\, q_0 u) + q_2 C_{s3k} e^{q_2 u} - q_2 C_{s4k} e^{-q_2 u} - q_3 C_{s5k}\, sin(\, q_3 u) + q_3 C_{s6k}\, cos(\, q_3 u) \quad (42a)$$

$$\partial^2 G_{sk}(u)/\partial u^2 = -q_0^2 C_{s1k}\, cos(\, q_0 u) - q_0^2 C_{s2k}\, sin(\, q_0 u) + q_2^2 C_{s3k} e^{q_2 u} + q_2^2 C_{s4k} e^{-q_2 u} - q_3^2 C_{s5k}\, cos(\, q_3 u) - q_3^2 C_{s6k}\, sin(\, q_3 u) \quad (42b)$$

Substituting Eqs. (42), (42a), and (42b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s3k} + C_{s4k} + C_{s5k} = d_{sk,1}$$

$$q_0 C_{s2k} + q_2 C_{s3k} - q_2 C_{s4k} + q_3 C_{s6k} = d_{sk,2}$$

$$-q_0^2 C_{s1k} + q_2^2 C_{s3k} + q_2^2 C_{s4k} - q_3^2 C_{s5k} = d_{sk,3}$$

$$C_{s1k}\, cos(\, q_0) + C_{s2k}\, sin(\, q_0) + C_{s3k} e^{q_2} + C_{s4k} e^{-q_2} + C_{s5k}\, cos(\, q_3) + C_{s6k}\, sin(\, q_3) = d_{sk,4}$$

$$-q_0 C_{s1k}\, sin(\, q_0) + q_0 C_{s2k}\, cos(\, q_0) + q_2 C_{s3k} e^{q_2} - q_2 C_{s4k} e^{-q_2} - q_3 C_{s5k}\, sin(\, q_3) + q_3 C_{s6k}\, cos(\, q_3) = d_{sk,5}$$

$$-q_0^2 C_{s1k}\, cos(\, q_0) - q_0^2 C_{s2k}\, sin(\, q_0) + q_2^2 C_{s3k} e^{q_2} + q_2^2 C_{s4k} e^{-q_2} - q_3^2 C_{s5k}\, cos(\, q_3) - q_3^2 C_{s6k}\, sin(\, q_3) = d_{sk,6} \quad (42c)$$

Solving Eq. (42c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (42), $G_{sk}(u)$ is obtained.

10) For $b_3 > 0$, $b_1^2 > 4b_2$, and $b_1 > \sqrt{b_1^2 - 4b_2}$, $(r_{sk})_{1,2} = \pm i q_0$ according to Eq. (29), and $(r_{sk})_{3,4} = \pm i q_2$ and $(r_{sk})_{5,6} = \pm i q_3$ according to Eq. (36). The solution of the second ODE in Eq. (12) is

$$G_{sk}(u) = C_{s1k}\, cos(\, q_0 u) + C_{s2k}\, sin(\, q_0 u) + C_{s3k}\, cos(\, q_2 u) + C_{s4k}\, sin(\, q_2 u) + C_{s5k}\, cos(\, q_3 u) + C_{s6k}\, sin(\, q_3 u) \quad (43)$$

$$\partial G_{sk}(u)/\partial u = -q_0 C_{s1k}\, sin(\, q_0 u) + q_0 C_{s2k}\, cos(\, q_0 u) - q_2 C_{s3k}\, sin(\, q_2 u) + q_2 C_{s4k}\, cos(\, q_2 u) - q_3 C_{s5k}\, sin(\, q_3 u) + q_3 C_{s6k}\, cos(\, q_3 u) \quad (43a)$$

$$\partial^2 G_{sk}(u)/\partial u^2 = -q_0^2 C_{s1k}\, cos(\, q_0 u) - q_0^2 C_{s2k}\, sin(\, q_0 u) - q_2^2 C_{s3k}\, cos(\, q_2 u) - q_2^2 C_{s4k}\, sin(\, q_2 u) - q_3^2 C_{s5k}\, cos(\, q_3 u) - q_3^2 C_{s6k} \sin\,(q_3 u) \quad (43b)$$

Substituting Eqs. (43), (43a), and (43b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s3k} + C_{s5k} = d_{sk,1}$$

$$q_0 C_{s2k} + q_2 C_{s4k} + q_3 C_{s6k} = d_{sk,2}$$

$$-q_0^2 C_{s1k} - q_2^2 C_{s3k} - q_3^2 C_{s5k} = d_{sk,3}$$

$$C_{s1k}\, cos(\, q_0) + C_{s2k}\, sin(\, q_0) + C_{s3k}\, cos(\, q_2) + C_{s4k}\, sin(\, q_2) + C_{s5k}\, cos(\, q_3) + C_{s6k}\, sin(\, q_3) = d_{sk,4}$$

$$-q_0 C_{s1k}\, sin(\, q_0) + q_0 C_{s2k}\, cos(\, q_0) - q_2 C_{s3k}\, sin(\, q_2) + q_2 C_{s4k}\, cos(\, q_2) - q_3 C_{s5k}\, sin(\, q_3) + q_3 C_{s6k}\, cos(\, q_3) = d_{sk,5}$$

$$-q_0^2 C_{s1k}\, cos(\, q_0) - q_0^2 C_{s2k}\, sin(\, q_0) - q_2^2 C_{s3k}\, cos(\, q_2) - q_2^2 C_{s4k}\, sin(\, q_2) - q_3^2 C_{s5k}\, cos(\, q_3) - q_3^2 C_{s6k} \sin\,(q_3) = d_{sk,6} \quad (43c)$$

Solving Eq. (43c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (43), $G_{sk}(u)$ is obtained.

Case 3: $b_1^2 < 4b_2$. For this case, Eq. (30) is changed into the following one

$$(r_{sk}^2)_{1,2} = \left(-b_1 \pm i\sqrt{4b_2 - b_1^2}\right)/2 \quad (44)$$

To obtain the roots $(r_{sk})_{3,4,5,6}$ from Eq. (44), is Eq. (44) is transformed into

$$(r_{sk}^2)_{1,2} = r(cos\, \varphi \pm i\, sin\, \varphi) \quad (45)$$

where

$$r = \sqrt{|b_2|}$$

$$cos\, \varphi = -|b_1|/\left(2\sqrt{|b_2|}\right) \quad \text{for} \quad b_1 > 0$$

$$cos\, \varphi = |b_1|/\left(2\sqrt{|b_2|}\right) \quad \text{for} \quad b_1 < 0$$

$$sin\, \varphi = \sqrt{4b_2 - b_1^2}/\left(2\sqrt{|b_2|}\right) \quad (46)$$

According to Eq. (45), the four roots $(r_{sk})_{3,4,5,6}$ for Case 3 can be written as

$$(r_{sk})_{3,4,5,6} = \sqrt[4]{|b_2|}\{cos[(\theta + 2j\pi)/2] \pm i\, sin[(\theta + 2j\pi)/2]\}$$

$$(j = 0,1) \quad (47)$$

After introducing

$$q_4 = 0.5\sqrt{2\sqrt{|b_2|} + |b_1|}$$

$$q_5 = 0.5\sqrt{2\sqrt{|b_2|} - |b_1|} \tag{48}$$

the four roots for $b_1 > 0$ are

$$(r_{sk})_{3,4} = q_5 \pm iq_4$$

$$(r_{sk})_{5,6} = -q_5 \mp iq_4 \tag{49}$$

and the four roots for $b_1 < 0$ are

$$(r_{sk})_{3,4} = q_4 \pm iq_5$$

$$(r_{sk})_{5,6} = -q_4 \mp iq_5 \tag{50}$$

From the two roots in (29) and the four roots in (49) and (50), the four different solutions of the second ordinary differential equation of Eq. (12) are obtained below.

11) For $b_3 < 0$, $b_1^2 < 4b_2$, and $b_1 > 0$, $(r_{sk})_{1,2} = \pm q_0$ according to Eq. (29), and $(r_{sk})_{3,4} = q_5 \pm iq_4$ and $(r_{sk})_{5,6} = -q_5 \mp iq_4$ according to Eq. (49). The solution of the second ODE in Eq. (12) is

$$G_{sk}(u) = C_{s1k}e^{q_0 u} + C_{s2k}e^{-q_0 u} + e^{q_5 u}(C_{s3k}cosq_4 u + C_{s4k}\,sin\,q_4\,u) + e^{-q_5 u}(C_{s5k}cosq_4 u + C_{s6k}sin\,q_4\,u) \tag{51}$$

$$\partial G_{sk}(u)/\partial u = q_0 C_{s1k}e^{q_0 u} - q_0 C_{s2k}e^{-q_0 u} + C_{s3k}e^{q_5 u}(q_5 cosq_4 u - q_4 sinq_4 u) + C_{s4k}e^{q_5 u}(q_5\,sin\,q_4\,u + q_4 cos\,q_4\,u) - C_{s5k}e^{-q_5 u}(q_5 cosq_4 u + q_4 sinq_4 u) - C_{s6k}e^{-q_5 u}(q_5 sinq_5 u - q_4 cosq_4 u) \tag{51a}$$

$$\partial^2 G_{sk}(u)/\partial u^2 = q_0^2 C_{s1k}e^{q_0 u} + q_0^2 C_{s2k}e^{-q_0 u} + C_{s3k}e^{q_5 u}(q_5^2 cosq_4 u - 2q_4 q_5 sinq_4 u - q_4^2 cosq_4 u) + C_{s4k}e^{q_5 u}(q_5^2\,sin\,q_4\,u + 2q_4 q_5 cos\,q_4\,u - q_4^2 sinq_4 u) + C_{s5k}e^{-q_5 u}(q_5^2 cosq_4 u + 2q_4 q_5 sinq_4 u - q_4^2 cosq_4 u) + C_{s6k}e^{-q_5 u}(q_5^2 sinq_4 u - 2q_4 q_5 cosq_4 u - q_4^2 sinq_4 u) \tag{51b}$$

Substituting Eqs. (51), (51a), and (51b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s2k} + C_{s3k} + C_{s5k} = d_{sk,1}$$

$$q_0 C_{s1k} - q_0 C_{s2k} + C_{s3k} q_5 + C_{s4k} q_4 - C_{s5k} q_5 + C_{s6k} q_4 = d_{sk,2}$$

$$q_0^2 C_{s1k} + q_0^2 C_{s2k} + C_{s3k}(q_5^2 - q_4^2) + 2q_4 q_5 C_{s4k} + C_{s5k}(q_5^2 - q_4^2) - 2q_4 q_5 C_{s6k} = d_{sk,3}$$

$$C_{s1k} e^{q_0} + C_{s2k} e^{-q_0} + e^{q_5}(C_{s3k} cos q_4 + C_{s4k} \sin q_4) + e^{-q_5}(C_{s5k} cos q_4 +$$

$$C_{s6k} \sin q_4) \ = d_{sk,4}$$

$$q_0 C_{s1k} e^{q_0} - q_0 C_{s2k} e^{-q_0} + C_{s3k} e^{q_5}(q_5 cos q_4 - q_4 sin q_4) + C_{s4k} e^{q_5}(q_5 \sin q_4 +$$

$$q_4 cos q_4) - C_{s5k} e^{-q_5}(q_5 cos q_4 + q_4 sin q_4) - C_{s6k} e^{-q_5}(q_5 sin q_4 - q_4 cos q_4) = d_{sk,5}$$

$$q_0^2 C_{s1k} e^{q_0} + q_0^2 C_{s2k} e^{-q_0} + C_{s3k} e^{q_5}(q_5^2 cos q_4 - 2q_4 q_5 sin q_4 - q_4^2 cos q_4) +$$

$$C_{s4k} e^{q_5}(q_5^2 \sin q_4 + 2q_4 q_5 cos q_4 - q_4^2 sin q_4) + C_{s5k} e^{-q_5}(q_5^2 cos q_4 + 2q_4 q_5 sin q_4 - q_4^2 cos q_4) +$$

$$C_{s6k} e^{-q_5}(q_5^2 sin q_4 - 2q_4 q_5 cos q_4 - q_4^2 sin q_4) = d_{sk,6} \qquad (51c)$$

Solving Eq. (51c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (51), $G_{sk}(u)$ is obtained.

12) For $b_3 < 0$, $b_1^2 < 4b_2$, and $b_1 < 0$, $(r_{sk})_{1,2} = \pm q_0$ according to Eq. (29), and $(r_{sk})_{3,4} = q_4 \pm i q_5$ and $(r_{sk})_{5,6} = -q_4 \mp i q_5$ according to Eq. (50). The solution of the second ODE in Eq. (12) is

$$G_{sk}(u) = C_{s1k} e^{q_0 u} + C_{s2k} e^{-q_0 u} + e^{q_4 u}(C_{s3k} cos q_5 u$$

$$+ C_{s4k} \sin q_5 u) + e^{-q_4 u}(C_{s5k} cos q_5 u + C_{s6k} sin q_5 u) \qquad (52)$$

$$\partial G_{sk}(u)/\partial u = q_0 C_{s1k} e^{q_0 u} - q_0 C_{s2k} e^{-q_0 u} + C_{s3k} e^{q_4 u}(q_4 cos q_5 u - q_5 sin q_5 u) +$$

$$C_{s4k} e^{q_4 u}(q_4 \ \sin q_5 u + q_5 \cos q_5 u) - C_{s5k} e^{-q_4 u}(q_4 cos q_5 u + q_5 sin q_5 u) +$$

$$C_{s6k} e^{-q_4 u}(-q_4 \sin q_5 u + q_5 \cos q_5 u) \qquad (52a)$$

$$\partial^2 G_{sk}(u)/\partial u^2 = q_0^2 C_{s1k} e^{q_0 u} + q_0^2 C_{s2k} e^{-q_0 u} + C_{s3k} e^{q_4 u}(q_4^2 cos q_5 u - 2q_4 q_5 sin q_5 u -$$

$$q_5^2 cos q_5 u) + C_{s4k} e^{q_4 u}(q_4^2 sin q_5 u + 2q_4 q_5 cos q_5 u - q_5^2 sin q_5 u) + C_{s5k} e^{-q_4 u}(q_4^2 cos q_5 u +$$

$$2q_4 q_5 sin q_5 u - q_5^2 cos q_5 u) + C_{s6k} e^{-q_4 u}(q_4^2 \sin q_5 u - 2q_4 q_5 \cos q_5 u - q_5^2 \sin q_5 u) \qquad (52b)$$

Substituting Eqs. (52), (52a), and (52b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s2k} + C_{s3k} + C_{s5k} = d_{sk,1}$$

$$q_0 C_{s1k} - q_0 C_{s2k} + q_4 C_{s3k} + q_5 C_{s4k} - q_4 C_{s5k} + q_5 C_{s6k} = d_{sk,2}$$

$$q_0^2 C_{s1k} + q_0^2 C_{s2k} + C_{s3k}(q_4^2 - q_5^2) + 2q_4 q_5 C_{s4k} + C_{s5k}(q_4^2 - q_5^2) - 2q_4 q_5 C_{s6k} = d_{sk,3}$$

$$C_{s1k} e^{q_0} + C_{s2k} e^{-q_0} + e^{q_4}(C_{s3k} cos q_5 + C_{s4k} \sin q_5) + e^{-q_4}(C_{s5k} cos q_5 +$$

$$C_{s6k} \sin q_5) \ = d_{sk,4}$$

$$q_0 C_{s1k} e^{q_0} - q_0 C_{s2k} e^{-q_0} + C_{s3k} e^{q_4}(q_4 cos q_5 - q_5 sin q_5) + C_{s4k} e^{q_4}(q_4 \sin q_5 +$$

$$q_5 \cos q_5) - C_{s5k} e^{-q_4}(q_4 cos q_5 + q_5 sin q_5) + C_{s6k} e^{-q_4}(-q_4 \sin q_5 + q_5 \cos q_5) = d_{sk,5}$$

$$q_0^2 C_{s1k} e^{q_0} + q_0^2 C_{s2k} e^{-q_0} + C_{s3k} e^{q_4}(q_4^2 cos q_5 - 2q_4 q_5 sin q_5 - q_5^2 cos q_5) + C_{s4k} e^{q_4}(q_4^2 sin q_5 +$$

$$2q_4 q_5 cos q_5 - q_5^2 sin q_5) + C_{s5k} e^{-q_4}(q_4^2 cos q_5 + 2q_4 q_5 sin q_5 - q_5^2 cos q_5) + C_{s6k} e^{-q_4}(q_4^2 \sin q_5 -$$

$$2q_4 q_5 \cos q_5 - q_5^2 \sin q_5) =$$

$$d_{sk,6} \tag{52c}$$

Solving Eq. (52c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (52), $G_{sk}(u)$ is obtained.

13) For $b_3 > 0$, $b_1^2 < 4b_2$, and $b_1 > 0$, $(r_{sk})_{1,2} = \pm i q_0$ according to Eq. (29), and $(r_{sk})_{3,4} = q_5 \pm i q_4$ and $(r_{sk})_{5,6} = -q_5 \mp i q_4$ according to Eq. (49). The solution of the second Ordinary Differential Equation of Eq. (12) is

$$G_{sk}(u) = C_{s1k} cos(\, q_0 u) + C_{s2k} sin(\, q_0 u) + e^{q_5 u}(C_{s3k} cos q_4 u$$

$$+ C_{s4k} \sin q_4 u) + e^{-q_5 u}(C_{t5k} cos q_4 u + C_{s6k} \sin q_4 u) \tag{53}$$

$$\partial G_{sk}(u)/\partial u = -q_0 C_{s1k} sin(\, q_0 u) + q_0 C_{s2k} cos(\, q_0 u) + C_{s3k} e^{q_5 u}(q_5 cos q_4 u -$$

$$q_4 sin q_4 u) + C_{s4k} e^{q_5 u}(q_5 sin q_4 u + q_4 cos q_4 u) + C_{t5k} e^{-q_5 u}(-q_5 cos q_4 u - q_4 sin q_4 u) +$$

$$C_{t6k} e^{-q_5 u}(-q_5 sin q_4 u + q_4 cos q_4 u) \tag{53a}$$

$$\partial^2 G_{sk}(u)/\partial u^2 = -q_0^2 C_{s1k} cos(\, q_0 u) - q_0^2 C_{s2k} sin(\, q_0 u) + C_{s3k} e^{q_5 u}(q_5^2 cos q_4 u -$$

$$2q_4 q_5 sin q_4 u - q_4^2 cos q_4 u) + C_{s4k} e^{q_5 u}(q_5^2 sin q_4 u + 2q_4 q_5 cos q_4 u - q_4^2 sin q_4 u) +$$

$C_{t5k}e^{-q_5u}(q_5^2 cosq_4u + 2q_4q_5 sinq_4u - q_4^2 cosq_5u) + C_{t6k}e^{-q_5u}(q_5^2 sinq_4u - 2q_4q_5 cosq_4u - q_4^2 sinq_4u)$ (53b)

Substituting Eqs. (53), (53a), and (53b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s3k} + C_{t5k} = d_{sk,1}$$

$$q_0 C_{s2k} + q_5 C_{s3k} + q_4 C_{s4k} - q_5 C_{t5k} + q_4 C_{t6k} = d_{sk,2}$$

$$-q_0^2 C_{s1k} + C_{s3k}(q_5^2 - q_4^2) + 2q_4q_5 C_{s4k} + C_{t5k}(q_5^2 - q_4^2) - 2q_4q_5 C_{t6k} = d_{sk,3}$$

$$C_{s1k}\, cos(\, q_0) + C_{s2k}\, sin(\, q_0) + e^{q_5}(C_{s3k} cosq_4 + C_{s4k}\, sin\, q_4) + e^{-q_5}(C_{t5k} cosq_4 + C_{s6k} sin\, q_4) = d_{sk,4}$$

$$-q_0 C_{s1k}\, sin(\, q_0) + q_0 C_{s2k}\, cos(\, q_0) + C_{s3k} e^{q_5}(q_5\, cosq_4 - q_4 sinq_4) + C_{s4k} e^{q_5}(q_5\, sinq_4 + q_4 cosq_4) + C_{t5k} e^{-q_5}(-q_5 cosq_4 - q_4 sinq_4) + C_{t6k} e^{-q_5}(-q_5 sinq_4 + q_4 cosq_4) = d_{sk,5}$$

$$-q_0^2 C_{s1k}\, cos(\, q_0) - q_0^2 C_{s2k}\, sin(\, q_0) + C_{s3k} e^{q_5}(q_5^2\, cosq_4 - 2q_4q_5 sinq_4 - q_4^2 cosq_4) + C_{s4k} e^{q_5}(q_5^2\, sinq_4 + 2q_4q_5 cosq_4 - q_4^2 sinq_4) + C_{t5k} e^{-q_5}(q_5^2 cosq_4 + 2q_4q_5 sinq_4 - q_4^2 cosq_5) + C_{t6k} e^{-q_5}(q_5^2 sinq_4 - 2q_4q_5 cosq_4 - q_4^2 sinq_4) = d_{sk,6}$$ (53c)

Solving Eq. (53c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (53), (u) is obtained.

14) For $b_3 > 0$, $b_1^2 < 4b_2$, and $b_1 < 0$, $(r_{sk})_{1,2} = \pm iq_0$ according to Eq. (29), and $(r_{sk})_{3,4} = q_4 \pm iq_5$ and $(r_{sk})_{5,6} = -q_4 \mp iq_5$ according to Eq. (50). The solution of the second ODE in Eq. (12) is

$$G_{sk}(u) = C_{s1k}\, cos(\, q_0 u) + C_{s2k}\, sin(\, q_0 u) + e^{q_4 u}(C_{s3k} cosq_5 u + C_{s4k}\, sin\, q_5\, u) + e^{-q_4 u}(C_{s5k} cosq_5 u + C_{s6k} sin\, q_5\, u)$$ (54)

$$\partial G_{sk}(u)/\partial u = -q_0 C_{s1k}\, sin(\, q_0 u) + q_0 C_{s2k}\, cos(\, q_0 u) + C_{s3k} e^{q_4 u}(q_4\, cosq_5 u - q_5 sinq_5 u) + C_{s4k} e^{q_4 u}(q_4\, sinq_5 u + q_5 cosq_5 u) - C_{s5k} e^{-q_4 u}(q_4\, cosq_5 u + q_5 sinq_5 u) + C_{s6k} e^{-q_4 u}(-q_4\, sinq_5 u + q_5 cosq_5 u)$$ (54a)

$$\partial^2 G_{sk}(u)/\partial u^2 = -q_0^2 C_{s1k}\, cos(\, q_0 u) - q_0^2 C_{s2k}\, sin(\, q_0 u) +$$

$C_{s3k} e^{q_4 u}(q_4^2 cos q_5 u - 2q_4 q_5 sin q_5 u - q_5^2 cos q_5 u) + C_{s4k} e^{q_4 u}(q_4^2 sin q_5 u + 2q_4 q_5\, cos q_5 u -$

$q_5^2 sin q_5 u) + C_{s5k} e^{-q_4 u}(q_4^2 cos q_5 u + q_4 q_5 sin q_5 u - q_5^2 cos q_5 u) + C_{s6k} e^{-q_4 u}(q_4^2 sin q_5 u -$

$2q_4\, q_5 cos q_5 u - q_5^2 sin q_5 u)$ (54b)

Substituting Eqs. (54), (54a), and (54b) into the boundary constraints (14), the following equations are obtained.

$$C_{s1k} + C_{s3k} + C_{s5k} = d_{sk,1}$$

$$q_0 C_{s2k} + q_4 C_{s3k} + q_5 C_{s4k} - q_4 C_{s5k} + q_5 C_{s6k} = d_{sk,2}$$

$$-q_0^2 C_{s1k} + C_{s3k}(q_4^2 - q_5^2) + 2q_4 q_5 C_{s4k} + C_{s5k}(q_4^2 - q_5^2) - 2q_4\, q_5 C_{s6k} = d_{sk,3}$$

$$C_{s1k}\, cos(\, q_0) + C_{s2k}\, sin(\, q_0) + e^{q_4}(C_{s3k} cos q_5$$

$$+ C_{s4k}\, sin\, q_5) + e^{-q_4}(C_{s5k} cos q_5 + C_{s6k} sin\, q_5) = d_{sk,4}$$

$$-q_0 C_{s1k}\, sin(\, q_0) + q_0 C_{s2k}\, cos(\, q_0) + C_{s3k} e^{q_4}(q_4\, cos q_5 - q_5 sin q_5) + C_{s4k} e^{q_4}(q_4\, sin q_5 +$$

$q_5 cos q_5) - C_{s5k} e^{-q_4}(q_4\, cos q_5 + q_5 sin q_5) + C_{s6k} e^{-q_4}(-q_4\, sin q_5 + q_5 cos q_5) = d_{sk,5}$

$$-q_0^2 C_{s1k}\, cos(\, q_0) - q_0^2 C_{s2k}\, sin(\, q_0) + C_{s3k} e^{q_4}(q_4^2 cos q_5 - 2q_4 q_5 sin q_5 - q_5^2 cos q_5)$$

$$+ C_{s4k} e^{q_4}(q_4^2 sin q_5 + 2q_4 q_5\, cos q_5 - q_5^2 sin q_5) + C_{s5k} e^{-q_4}(q_4^2 cos q_5 + 2q_4 q_5 sin q_5$$

$$- q_5^2 cos q_5) + C_{s6k} e^{-q_4}(q_4^2 sin q_5 - 2q_4\, q_5 cos q_5 - q_5^2 sin q_5)$$

$$= d_{sk,6} \quad (54c)$$

Solving Eq. (54c), the 6 unknown constants C_{s1k}, C_{s2k}, C_{s3k}, C_{s4k}, C_{s5k}, and C_{s6k} are obtained. Substituting them back into Eq. (54), $G_{sk}(u)$ is obtained.

Introducing the above obtained $G_{s0}(u)$ in Eq. (17) and $G_{sk}(u)$ in one of the Eqs. (32)-(35), (38)-(43), and (51)-(54) into Eq. (10), the mathematical equations of $s(u,v)$ $(s = x, y, z)$ are obtained and use them to create blending surfaces. In the following subsection, some examples are given to demonstrate the applications of the analytical solution developed in this subsection in surface blending.

3.2.3 Applications

The example given below is to blend a top surface to a bottom surface. The parametric equations for the top and bottom surfaces are

$$\begin{aligned} x &= au\,sin(2\,\pi v) \\ y &= bu\,cos(2\,\pi v) \\ z &= h_1 + h_2 u^2 \end{aligned} \tag{55}$$

and

$$\begin{aligned} x &= cu\,sin(2\,\pi v) \\ y &= du\,cos(2\,\pi v) \\ z &= -h_3 u^3 \end{aligned} \tag{56}$$

Assuming that the top boundary $u = 0$ of the middle blending surface is taken to be at $u = u_0$ of the top primary surface, letting $u = u_0$ in Eq. (55), the following boundary constraints between the top primary surface and the middle blending surface at the position $u = u_0$ of the top primary surface are obtained.

$$\begin{gathered} u = 0 \\ x = au_0\,sin(2\,\pi v) \qquad y = bu_0\,cos(2\,\pi v) \qquad z = h_1 + h_2 u_0^2 \\ \frac{\partial x}{\partial u} = -a\,sin(2\,\pi v) \qquad \frac{\partial y}{\partial u} = -b\,cos(2\,\pi v) \qquad \frac{\partial z}{\partial u} = -2h_2 u_0 \\ \frac{\partial^2 x}{\partial u^2} = 0 \qquad \frac{\partial^2 y}{\partial u^2} = 0 \qquad \frac{\partial^2 z}{\partial u^2} = 2h_2 \end{gathered} \tag{57a}$$

where the minus symbol "$-$" is due to the opposite directions between the blending surface and the top primary surface. That is to say, when the parametric variable u increases from $u = 0$ to $u = 1$, the isoparametric curve of the parametric variable v of the top surface moves upwards, but the isoparametric curve of the parametric variable v of the blending surface moves downwards.

Assuming that the bottom boundary $u = 1$ of the middle blending surface is taken to be at $u = u_1$ of the bottom primary surface, and letting $u = u_1$ in Eq. (56), the following boundary constraints between the bottom primary surface and the middle blending surface at the position $u = u_1$ of the bottom primary surface are obtained.

$$u = 1$$

$$x = cu_1\, sin(2\,\pi v) \qquad y = du_1\, cos(2\,\pi v) \qquad z = -h_3 u_1^3$$

$$\frac{\partial x}{\partial u} = c\, sin(2\,\pi v) \qquad \frac{\partial y}{\partial u} = d\, cos(2\,\pi v) \qquad \frac{\partial z}{\partial u} = -3h_3 u_1^2$$

$$\frac{\partial^2 x}{\partial u^2} = 0 \qquad \frac{\partial^2 y}{\partial u^2} = 0 \qquad \frac{\partial^2 z}{\partial u^2} = -6h_3 u_1 \tag{57b}$$

The boundary constraints (57a) and (57b) indicate that the elementary functions for x component are $f_{x1}(v) = sin\, 2\,\pi v$, the elementary functions for y component are $f_{y1}(v) = cos\, 2\,\pi v$, and those for z component are constants. Substituting $G_{x0}(u)$= $G_{y0}(u) = 0$, K_x=$K_y = 1$ for x and y components, $f_{x1}(v) = sin(2\,\pi v)$, $f_{y1}(v) = cos(2\pi v)$, and $K_z = 0$ for z component into Eq. (10), the following constructed functions of the blending surface are obtained

$$x(u, v) = G_{x1}(u)\, sin(2\,\pi v)$$

$$y(u, v) = G_{y1}(u)\, cos(2\,\pi v)$$

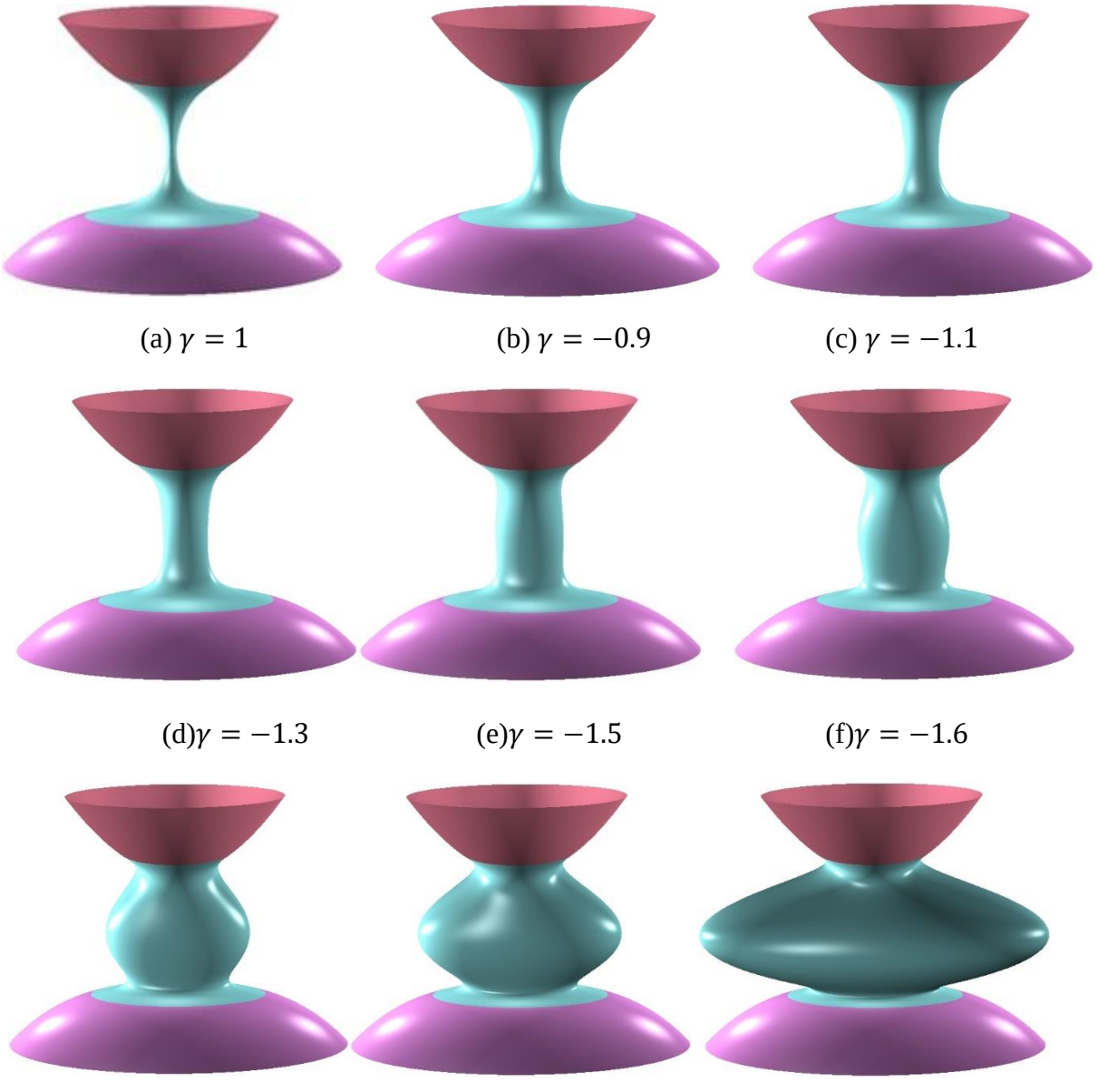
$$z(u, v) = G_{z0}(u) \tag{58}$$

With the method developed in the previous subsection, the unknown functions $G_{x1}(u)$, $G_{y1}(u)$, and $G_{z0}(u)$ are obtained. Substituting them back into Eq. (58), the functions defining the blending surface are obtained.

The developed method provides a powerful tool in quickly creating different shapes of a blending surface but still satisfies boundary constraints. This will be demonstrated this with some examples below. For all these examples, the geometric parameters are $a = 2.6$, $b = 4.5$, $c = h_3 = 5$, $d = h_1 = 2$, $h_2 = 3$, $u_0 = 0.35$, and $u_1 = 0.3$. On average, the CPU time used to determine all

the unknown constants of a blending surface with the above closed form solutions is 20 microseconds on a laptop with 2.5 GHz and 8 GB RAM.

First, the shape control parameters $\eta = \lambda = \rho = 1$ are kept unchanged but the shape control parameter γ is set to 1, -0.9, -1.1, -1.3, -1.5, -1.6, -1.7, -1.75, and -1.8. Different shapes of the blending surface are obtained and depicted in Figure 2.



(a) $\gamma = 1$ (b) $\gamma = -0.9$ (c) $\gamma = -1.1$

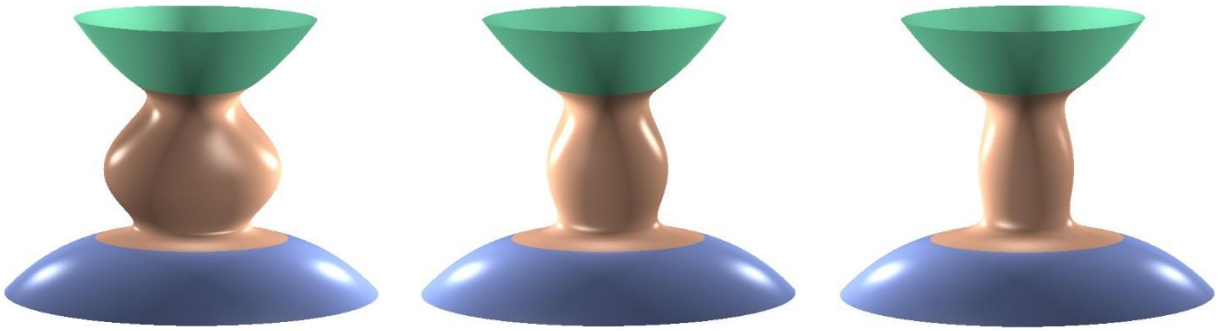
(d)$\gamma = -1.3$ (e)$\gamma = -1.5$ (f)$\gamma = -1.6$

(g) $\gamma = -1.7$ (h) $\gamma = -1.75$ (i) $\gamma = -1.8$

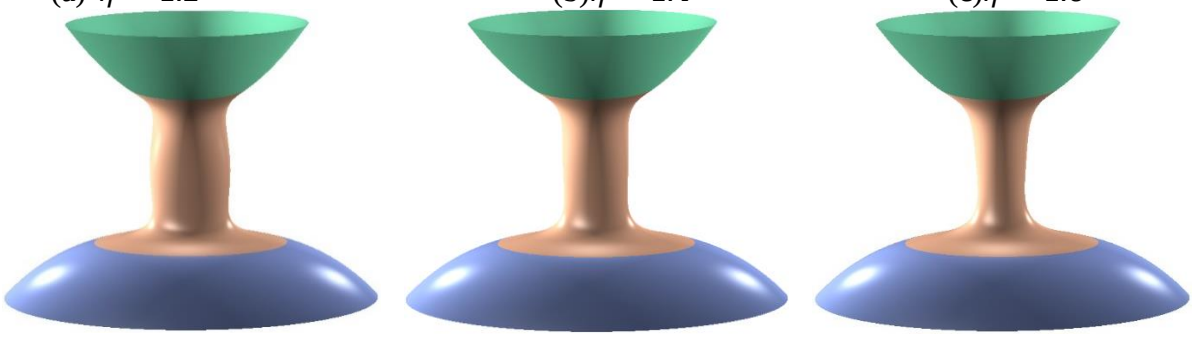
Figure 2: Different shapes of a blending surface determined by the shape control parameters η=λ=ρ=1 and different values of the shape control parameter γ

When the shape control parameter γ changes from 1 to -1.8, the shape of the blending surface changes from concave to convex. At the value $\gamma = -1.5$, the middle part of the blending surface becomes a straight cylinder with an almost constant radius along the height direction. When the shape control parameter γ changes from -1.5 to 1, the blending surface becomes more and more concave. When the shape control parameter γ changes from -1.5 to -1.8, the blending surface becomes more and more convex with a symmetry plane at the middle of the blending surface. Although different shapes of the blending surface are obtained by different values of the shape control parameter γ, the same continuities at the two trimlines are exactly maintained.

Second, the shape control parameters $\gamma, \lambda \; and \; \rho$ are set to $\gamma = -1.8$ and $\lambda = \rho = 1$, and kept unchanged. Then, the shape control parameter η is set to 1.2, 1.4, 1.6, 1.8, 2.2, 3, 4, 4.6, 4.9, 5, 6, 7, 8, and 9. Different shapes of the blending surface are obtained and shown in Figure 3.



(a) $\eta = 1.2$ (b)$\eta = 1.4$ (c)$\eta = 1.6$



(d) $\eta = 1.8$ (e) $\eta = 2.2$ (f) $\eta = 3$

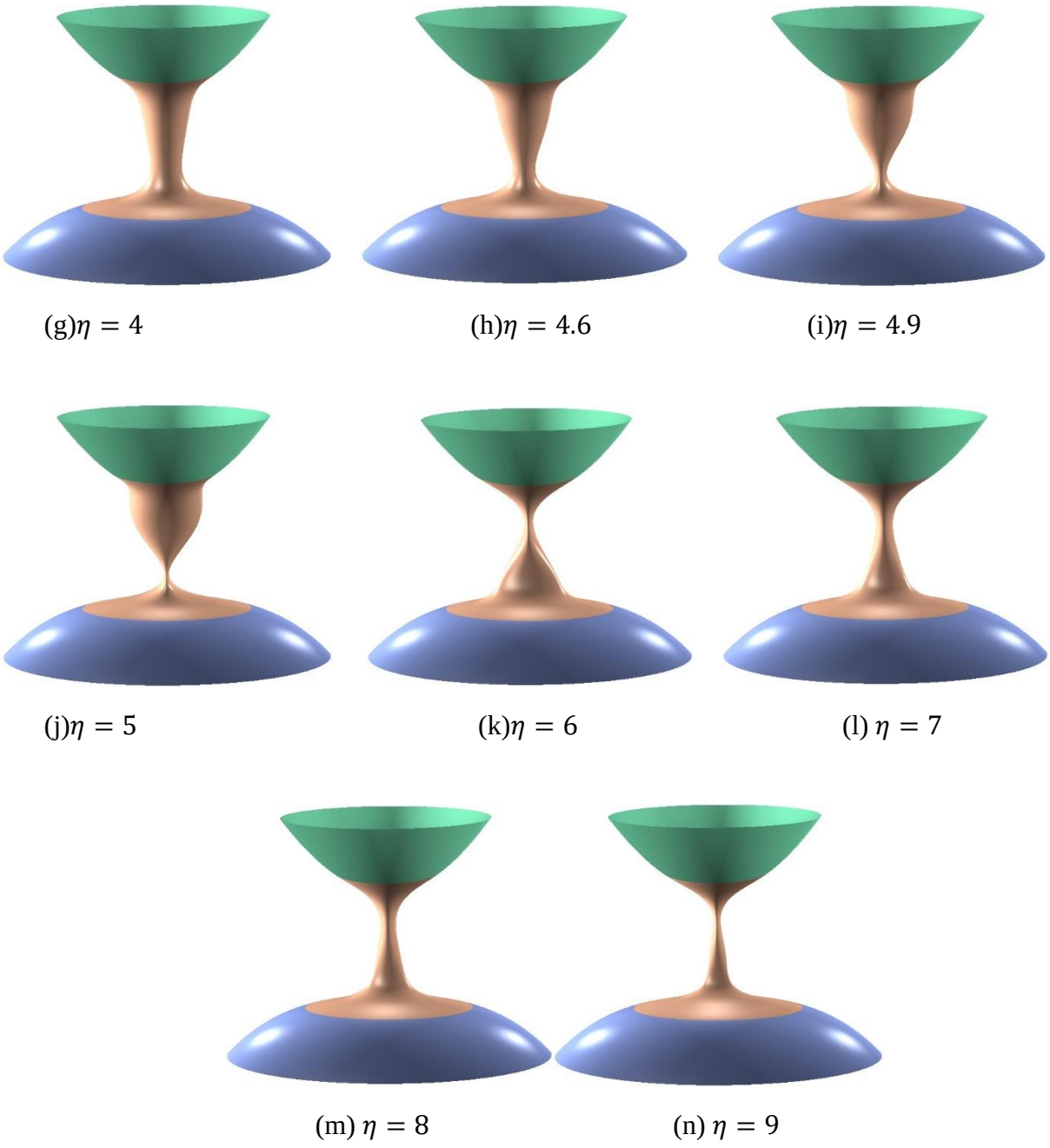


Figure 3: Different shapes of a blending surface determined by the shape control parameters γ=-1.8, λ=ρ=1 and different values of the shape control parameter η

According to Figure 3, the shape change of the blending surface can be divided into four parts. The first part is Figures 3(a)-3(e). In this part, the value of the shape control parameter η increases from 1.2 to 2.2, causing the blending surface to change its shape from a convex one to a straight cylinder with a same cross-section size. The second part is Figures 3(e)-3(g), in which the value of

the shape control parameter η increases from 2.2 to 4, causing the blending surface to change its shape from the straight cylinder into the frustum of a cone with the lower part of the frustum becoming smaller and smaller. The third part is Figures 3(g)-3(j). In this part, the value of the shape control parameter η increases from 4 to 5 which causes the following changes: 1) the lower part continues to become smaller and smaller, and 2) the upper part changes from a straight one into a convex one with bigger cross-section sizes. The fourth part is Figures 3(k)-3(n). There is a jump of the shape change between the third part and the fourth part. In the third part, the blending surface has a large cross-section size at the upper part and a small cross-section size at the lower part. After the jump, the shapes of the blending surface in the fourth part show that the cross-section size at the upper part becomes small, but the cross-section size at the lower part becomes large. In the fourth part, the value of the shape control parameter η increases from 6 to 9 which makes the lower part of the blending surface become smaller and smaller.

For one given combination of the shape control parameters γ and η, changing the shape control parameter λ can obtain different shapes of the blending surfaces. For another given combination of the shape control parameters γ and η, changing the shape control parameter λ can obtain more different shapes of the blending surfaces. Here three different combinations of the shape control parameters γ and η are used to demonstrate this.

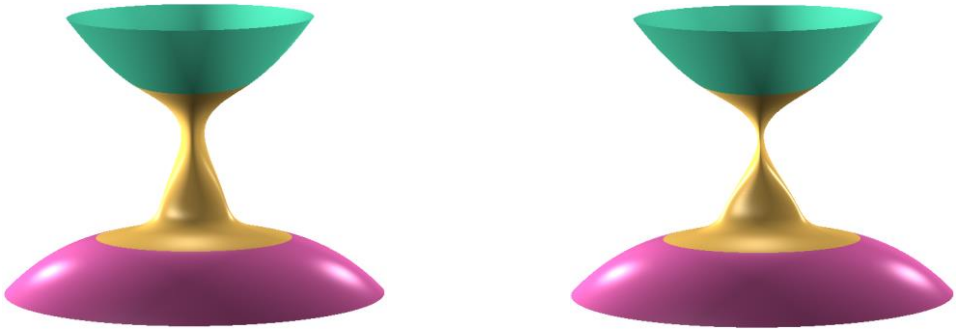
The first combination is the shape control parameters $\gamma = -1.8$ and $\eta = 6$. The shape control parameter $\rho = 1$ is unchanged. Then, the shape control parameter λ is set to 5, 4, 3, 2 and 0.5. The obtained shapes of the blending surface are shown in Figure 4.

It can be seen from the above figure that when the shape control parameter λ reduces its value from 5 to 0.5, the upper part of the blending surface become smaller and smaller. At $\lambda = 0.5$, the upper part of the blending surface becomes very small.

The second combination is the shape control parameters $\gamma = -1.8$ and $\eta = 1$. The shape control parameter $\rho = 1$ is also kept unchanged. Then, the shape control parameter λ is set to 1.1,

1.2, 1.3, 1.4, 1.6, 1.8, 2.2 and 2.6. The obtained shapes of the blending surface are shown in Figure 5.

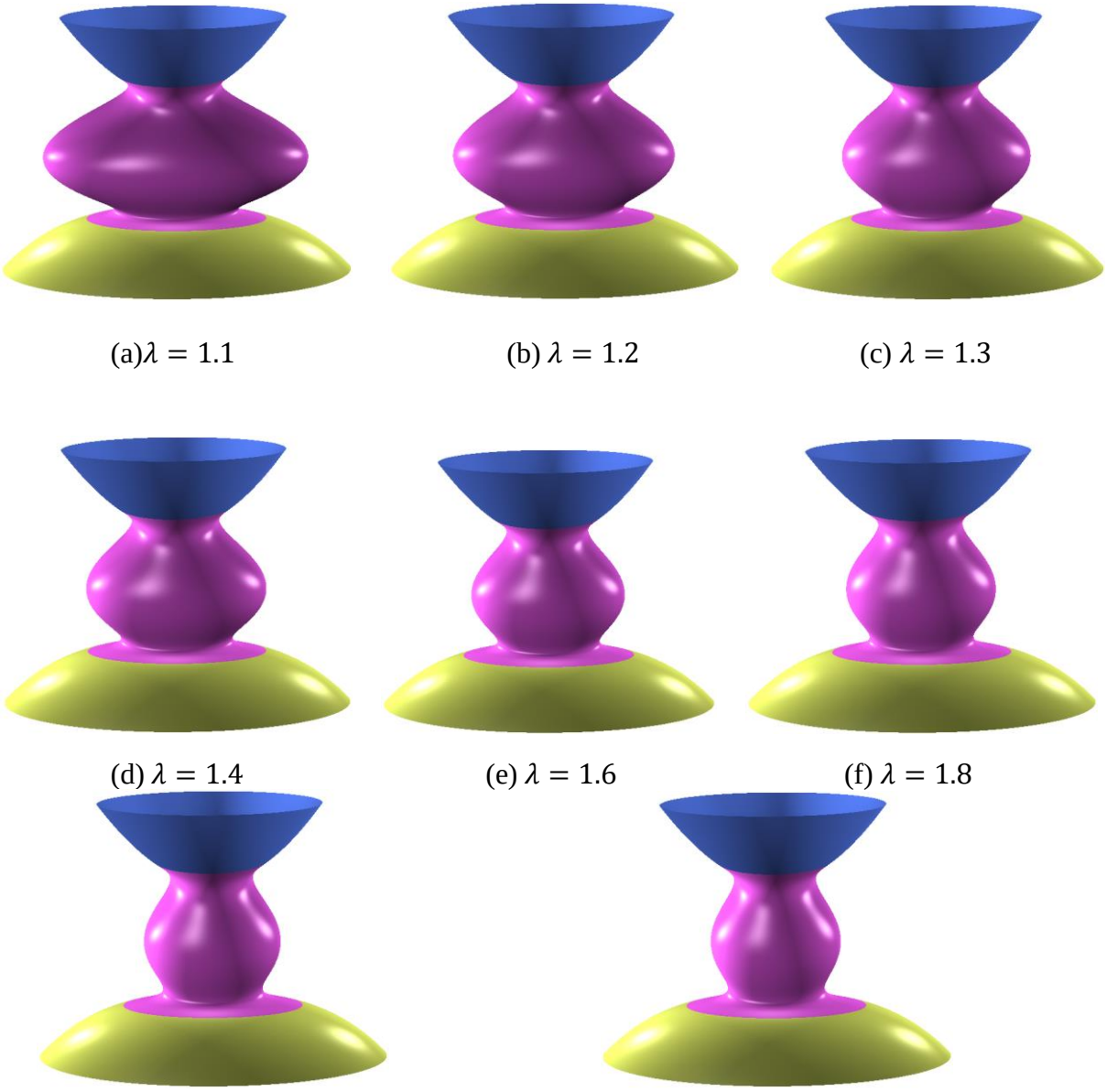
(a) $\lambda = 5$ (b) $\lambda = 4$ (c) $\lambda = 3$



(d) $\lambda = 2$ (e) $\lambda = 0.5$

Figure 4: Different shapes of a blending surface determined by the shape control parameters γ=-1.8, η=6,ρ=1 and different values of the shape control parameter λ

The images depicted in the figure indicate that this combination leads to convex shapes with a biggest cross-section size at the middle of the blending surface. When the shape control parameter λ increase its value from 1.1 to 2.6, the blending surface becomes less convex and the cross-section size at the middle of the blending surface becomes smaller and smaller.



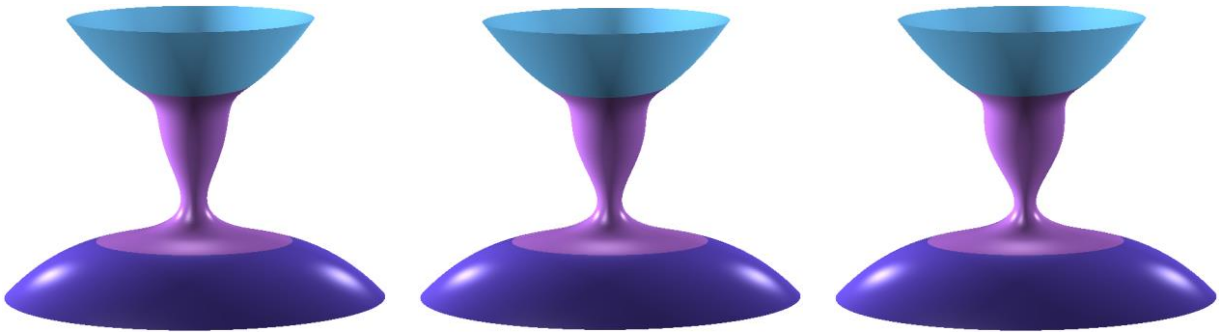
(a)$\lambda = 1.1$ (b) $\lambda = 1.2$ (c) $\lambda = 1.3$

(d) $\lambda = 1.4$ (e) $\lambda = 1.6$ (f) $\lambda = 1.8$

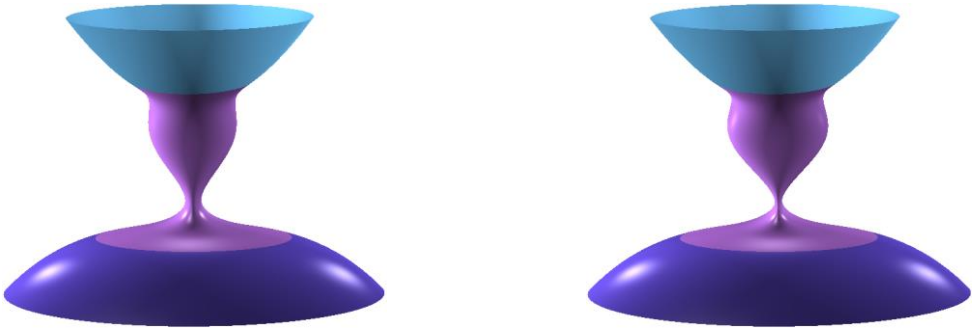
(g) $\lambda = 2.2$ (h) $\lambda = 2.6$

Figure 5: Different shapes of a blending surface determined by the shape control parameters γ=-1.8, η=1,ρ=1 and different values of the shape control parameter λ

The third combination is the shape control parameters $\gamma = -1.8$ and $\eta = 4.6$. The shape control parameter $\rho = 1$ is also kept unchanged. Then, the shape control parameter λ is set to 1.2, 1.4, 1.6, 1.8, and 2. The obtained shapes of the blending surface are shown in Figure 6.



(a) $\lambda = 1.2$ (b) $\lambda = 1.4$ (c) $\lambda = 1.6$



(a) $\lambda = 1.8$ (b) $\lambda = 2$

Figure 6: Different shapes of a blending surface determined by the shape control parameters γ=-1.8, η=4.6,ρ=1 and different values of the shape control parameter λ

The above images indicate that for this combination and the different values of the shape control parameter λ, the upper shape of the blending surface is convex and the lower shape of the blending surface is concave. When the value of the shape control parameter λ in Eq. (54c) increases from 1.2 to 2, the upper part of the blending surface becomes more and more convex, but the lower part of the blending surface becomes more and more concave.

When the combinations change from two shape control parameters to three shape control parameters, more different shapes of the blending surface than the combinations of two shape control parameters can be obtained. Here, two different combinations of three shape control parameters γ, η, and λ are given to demonstrate this.

The first combination is the shape control parameters $\gamma = -1.8$ and $\eta = \lambda = 1$. Then the value of the shape control parameter ρ is set to 1.15, 1.2, 1.25, 1.3, 1.35, and 1.4. The obtained shapes of the blending surface are shown in Figure 7.

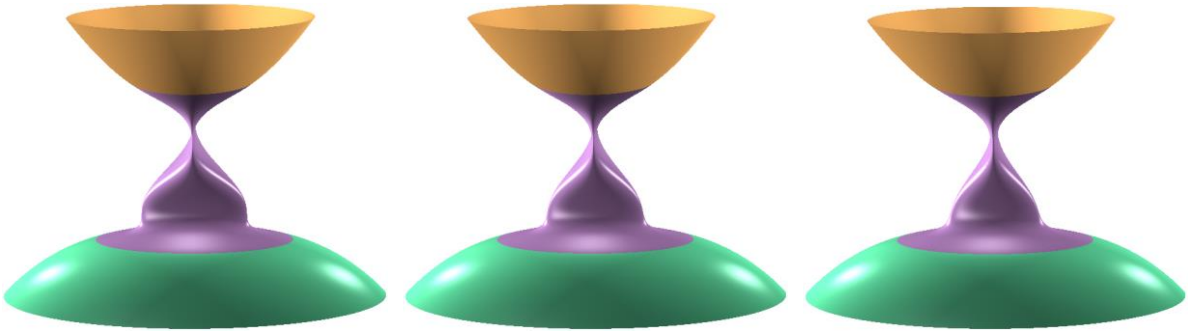
(a) $\rho = 1.15$ (b) $\rho = 1.2$ (c) $\rho = 1.25$

(d) $\rho = 1.3$ (e) $\rho = 1.35$ (f) $\rho = 1.4$

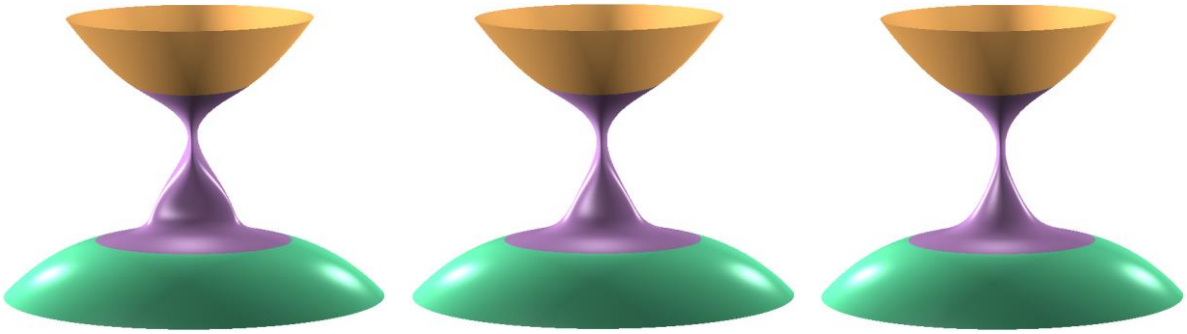
Figure 7: Different shapes of a blending surface determined by the shape control parameters γ=-1.8, η=λ=1 and different values of the shape control parameter ρ

It can be observed from the images depicted in Figure 7 that the shape of the blending surface is convex at $\rho = 1.15$, straight at $\rho = 1.3$, and concave at $\rho = 1.4$. When the value of the shape control parameter ρ increases from 1.15 to 1.3, the blending surface becomes less convex. When the value of the shape control parameter ρ increases from 1.3 to 1.4, the blending surface becomes more concave.

The second combination is the shape control parameters $\gamma = -1.8$, $\eta = 6$, and $\lambda = 1$. Then the value of the shape control parameter ρ is set to 0.1, 0.3, 0.5, 0.7, 1.3, and 2. The obtained shapes of the blending surface are shown in Figure 8.



(a) $\rho = 0.1$ (b) $\rho = 0.3$ (c) $\rho = 0.5$



(d) $\rho = 0.7$ (e) $\rho = 1.3$ (f) $\rho = 2$

Figure 8: Different shapes of a blending surface determined by the shape control parameters γ=-1.8, η=6,λ=1 and different values of the shape control parameter ρ

The images given in the above figure indicate that the upper part of the blending surface has a different shape change in comparison with the lower part of the blending surface. When the shape control parameter ρ in Eq.(54c) increases from 0.1 to 2, the upper part of the blending surface almost maintains a similar shape, but the lower part of the blending surface reduces its cross-section size greatly until its front view shape changes from a convex curved one at $\rho = 0.1$ into a triangle at $\rho = 2$.

From the above discussions, the following conclusions can be drawn. (1) The proposed closed form solutions are correct and efficient. (2) Four shape control parameters involved in the closed

form solutions have a strong influence on the shape of blending surfaces and can be developed into useful shape control handles to obtain different shapes of a same blending surface, and 3) Different combinations of the four shape control parameters can create enormous shapes of a same blending surface.

From Eqs. (29), (31), (37), and (48), we know q_0 is the function of b_3, q_1 is the function of b_1, and $q_2 - q_5$ are the functions of b_1 and b_2. From Eqs. (25)-(27), we know b_1, b_2 and b_3 are the functions of a_4, a_5 and a_6. From Eq. (21), we know a_4, a_5 and a_6 are the functions of $a_0 - a_3$. From Eq. (13), we know $a_0 - a_3$ are the function of the shape control parameters γ, η, λ, and ρ. Therefore, there are no explicit relationships between x, y and z components defining a blending surface and shape control parameters γ, η, λ, and ρ, and it is difficult to theoretically formulate how shape control parameters γ, η, λ, and ρ affect the shape of a blending surface. Even shape control parameters are kept unchanged, different sizes of blending surfaces and different blending boundary constraints also affect shape changes of blending surfaces. Taking all these factors, i. e., different combinations of shape control parameters, different sizes of blending surfaces, and different blending boundary constraints, into account, theoretically formulating how shape control parameters affect the shape of a blending surface would become more difficult.

Although it is difficult to theoretically formulate explicit relationships between shape changes of blending surfaces and shape control parameters, in practice, it is not difficult to manipulate the shape of blending surfaces. For doing this, each of shape control parameters is implemented into a slider. By interactively manipulating four sliders representing the four shape control parameters γ, η, λ, and ρ, users can see shape changes of a blending surface in real-time, choose their preferred shape, and obtain the values of the shape control parameters corresponding to the preferred shape.

3.3 Variable decomposition-based approximate analytical approach

For the blending boundary constraints consisting of trigonometric functions and exponential functions, the differential property (8) is satisfied, and the closed form solution-based approximate analytical approach investigated in previous section is most accurate and efficient among the three surface blending methods developed in this thesis. It provides the best means for blending between time-independent constant parametric surfaces with C^2 continuity investigated in the previous section. However, when the blending boundary constraints consist of some other functions such as power functions of the parametric variable v, these functions must be first converted into trigonometric functions. Such a conversion will cause tedious mathematical derivations, slow the resolving process, and reduce the performance of surface blending. To tackle this problem, this section will develop a new approach. It decomposes the functions in the blending boundary constraints according to different variables and constructs the corresponding blending surface functions accordingly. In what follows, such a variable decomposition-based approximate analytical approach will be introduced.

The variables in the blending boundary constraints can be divided into the constants, the parametric variable *v*, the time variable *t*, and the combination of the parametric variable *v* and the time variable *t*. Accordingly, the functions in the blending boundary constraints can be divided into four types: constants, the functions of the time variable *t*, the functions of the parametric variable *v*, and the functions of the parametric variable *v* and the time variable *t*. According to the divisions, the blending surface functions are decomposed into four components accordingly. They are the functions of the parametric variable *u*, the functions of the parametric variable *v* and the time variable *t*, the functions of the parametric variables *u* and *v*, and the functions of the parametric variables *u* and *v* and the time variable *t*. For each of the decomposed functions, their closed form solutions or

approximate analytical solutions are derived. Finally, some examples are presented to demonstrate their applications in C^2 continuous blending of constant and varying parametric surfaces.

3.3.1 Converted mathematical model and its solution

In order to simplify the solution, each of the functions in the blending boundary constraints is decomposed (1) into: constants $\boldsymbol{C}_n^{(1)}$, the functions $\boldsymbol{C}_n^{(2)}(t)$ of the time variable t, the functions $\boldsymbol{C}_n^{(3)}(v)$ of the parametric variable v, and the functions $\boldsymbol{C}_n^{(4)}(v,t)$ of the parametric variable v and the time variable t.

After the above decomposition, all the functions in the blending boundary constraints can be written as $\boldsymbol{C}_n(v,t) = \boldsymbol{C}_n^{(1)} + \boldsymbol{C}_n^{(2)}(t) + \boldsymbol{C}_n^{(3)}(v) + \boldsymbol{C}_n^{(4)}(v,t)$ $(n = 1,2,3,\dots,6)$. If $\boldsymbol{C}_n^{(2)}$, $\boldsymbol{C}_n^{(3)}$, and $\boldsymbol{C}_n^{(4)}$ are used to indicate $\boldsymbol{C}_n^{(2)}(t)$, $\boldsymbol{C}_n^{(3)}(v)$, and $\boldsymbol{C}_n^{(4)}(v,t)$, respectively, the blending boundary constraints (1) are changed into

$$u = 0 \qquad \partial^n \boldsymbol{S}(u,v,t)/\partial u^n = \sum_{k=1}^{4} \boldsymbol{C}_{n+1}^{(k)}$$

$$u = 1 \qquad \partial^n \boldsymbol{S}(u,v,t)/\partial u^n = \sum_{k=1}^{4} \boldsymbol{C}_{n+4}^{(k)}$$

$$(n = 0,1,2) \tag{59}$$

If w is used to indicate x, y and z, the above blending boundary constraints (59) can be written in the following component form

$$u = 0 \qquad \partial^n S_w(u,v,t)/\partial u^n = \sum_{k=1}^{4} C_{wn+1}^{(k)}$$

$$u = 1 \qquad \partial^n S_w(u,v,t)/\partial u^n = \sum_{k=1}^{4} C_{wn+4}^{(k)}$$

$$(w = x,y,z;\ n = 0,1,2) \tag{59a}$$

Since the blending boundary constraints consist of $C_{wn}^{(1)}$, $C_{wn}^{(2)}(t)$, $C_{wn}^{(3)}(v)$, and $C_{wn}^{(4)}(v,t)$, the function of the blending surface should have the corresponding terms, i.e., $S_w^{(1)}(u)$, $S_w^{(2)}(u,t)$, $S_w^{(3)}(u,v)$, and $S_w^{(4)}(u,v,t)$. That is to say, the function of the blending surface should have the form $S_w(u,v,t) = S_w^{(1)}(u) + S_w^{(2)}(u,t) + S_w^{(3)}(u,v) + S_w^{(4)}(u,v,t)$. Substituting it into Eq. (5) and using $S_w^{(1)}$, $S_w^{(2)}$, $S_w^{(3)}$, and $S_w^{(4)}$ to replace $S_w^{(1)}(u)$, $S_w^{(2)}(u,t)$, $S_w^{(3)}(u,v)$, and $S_w^{(4)}(u,v,t)$, respectively, $S_w(u,v,t) = S_w^{(1)} + S_w^{(2)} + S_w^{(3)} + S_w^{(4)}$ is obtained, and the partial differential equation (5) is changed into

$$\left(\gamma\frac{\partial^6}{\partial u^6} + \eta\frac{\partial^6}{\partial u^4\partial v^2} + \lambda\frac{\partial^6}{\partial u^2\partial v^4} + \rho\frac{\partial^6}{\partial v^6}\right)\left[S_w^{(1)} + S_w^{(2)} + S_w^{(3)} + S_w^{(4)}\right] = 0$$

$$(w = x, y, z) \tag{60}$$

Substituting $S_w(u,v,t) = S_w^{(1)} + S_w^{(2)} + S_w^{(3)} + S_w^{(4)}$ $(w = x, y, z)$ into the blending boundary constraints (59a), the following blending boundary constraints are obtained.

$$u = 0 \qquad \partial^n\left[S_w^{(1)} + S_w^{(2)} + S_w^{(3)} + S_w^{(4)}\right]/\partial u^n = \sum_{k=1}^{4} C_{wn+1}^{(k)}$$

$$u = 1 \qquad \partial^n\left[S_w^{(1)} + S_w^{(2)} + S_w^{(3)} + S_w^{(4)}\right]/\partial u^n = \sum_{k=1}^{4} C_{wn+4}^{(k)}$$

$$(w = x, y, z;\ n = 0,1,2) \tag{60a}$$

Solving the partial differential equation (60) subjected to the blending boundary constraints (60a) can be treated as solving the following equations.

$$\left(\gamma\frac{\partial^6}{\partial u^6} + \eta\frac{\partial^6}{\partial u^4\partial v^2} + \lambda\frac{\partial^6}{\partial u^2\partial v^4} + \rho\frac{\partial^6}{\partial v^6}\right)S_w^{(i)} = 0$$

$$(w = x, y, z;\ i = 1, 2, 3, 4) \tag{60b}$$

subjected to the following blending boundary constraints

$$u = 0 \qquad \partial^n S_w^{(i)} / \partial u^n = C_{wn+1}^{(i)}$$

$$u = 1 \qquad \partial^n S_w^{(i)} / \partial u^n = C_{wn+4}^{(i)}$$

$$(w = x, y, z;\ i = 1, 2, 3, 4; n = 0,1,2) \tag{60c}$$

When $i = 1$, Eq. (60b) becomes

$$\left(\gamma \frac{\partial^6}{\partial u^6} + \eta \frac{\partial^6}{\partial u^4 \partial v^2} + \lambda \frac{\partial^6}{\partial u^2 \partial v^4} + \rho \frac{\partial^6}{\partial v^6}\right) S_w^{(1)} = 0$$

$$(w = x, y, z) \tag{61}$$

and Eq. (60c) becomes

$$u = 0 \qquad \partial^n S_w^{(1)} / \partial u^n = C_{wn+1}^{(1)}$$

$$u = 1 \qquad \partial^n S_w^{(1)} / \partial u^n = C_{wn+4}^{(1)}$$

$$(w = x, y, z;\ n = 0,1,2) \tag{62}$$

where $C_{wn}^{(1)}$ $(w = x, y, z;\ n = 1, 2, 3, 4, 5, 6)$ are known constants.

The closed form solution of the partial differential equation (61) subjected to the blending boundary constraints (62) can be taken to be the form of

$$S_w^{(1)}(u) = \sum_{n=0}^{5} a_{wn}^{(1)} u^n$$

$$(w = x, y, z) \tag{63}$$

The above solution (63) has exactly satisfied the partial differential equation (61). Therefore, it is substituted into corresponding blending boundary constraints (62) to determine the unknown constants $a_{wn}^{(1)}$ $(w = x, y, z)$.

After substituting the determined vector-valued constants $a_{wn}^{(1)}$ $(w = x, y, z)$ back to Eq. (63), $S_w^{(1)}(u)$ is found to be

$$S_w^{(1)}(u) = \sum_{n=1}^{6} g_n(u) C_{wn}^{(1)}$$

$$(w = x, y, z) \tag{64}$$

where $g_1(u) - g_6(u)$ are the same as Eq. (18), i. e.,

$$g_1(u) = 1 - 10u^3 + 15u^4 - 6u^5$$

$$g_2(u) = (1 - 6u^2 + 8u^3 - 3u^4)u$$

$$g_3(u) = (0.5 - 1.5u + 1.5u^2 - 0.5u^3)u^2$$

$$g_4(u) = (10 - 15u + 6u^2)u^3$$

$$g_5(u) = (-4 + 7u - 3u^2)u^3$$

$$g_6(u) = (0.5 - u + 0.5u^2)u^3 \tag{65}$$

When $i = 2$, the boundary functions $C_{wn}^{(2)} = C_{wn}^{(2)}(t)$ $(w = x, y, z; n = 1, 2, \dots, 6)$ are the functions of the time variable *t*. Eq. (60b) becomes

$$\left(\gamma \frac{\partial^6}{\partial u^6} + \eta \frac{\partial^6}{\partial u^4 \partial v^2} + \lambda \frac{\partial^6}{\partial u^2 \partial v^4} + \rho \frac{\partial^6}{\partial v^6}\right) S_w^{(2)}(u, t) = 0$$

$$(w = x, y, z) \tag{66}$$

and Eq. (60c) becomes

$$u = 0 \qquad \partial^n S_w^{(2)} / \partial u^n = \boldsymbol{C}_{wn+1}^{(2)}(t)$$

$$u = 1 \qquad \partial^n S_w^{(2)} / \partial u^n = \boldsymbol{C}_{wn+4}^{(2)}(t)$$

$$(w = x, y, z;\ n = 0, 1, 2) \tag{67}$$

The closed form solution can be taken to be the form of

$$S_w^{(2)}(u, t) = \sum_{n=0}^{5} a_{wn}^{(2)}(t) u^n$$

$$(w = x, y, z) \tag{68}$$

Using the same method as *i*=1, the following closed form solution for *i*=2 is obtained.

$$S_w^{(2)}(u,t) = \sum_{n=1}^{6} g_n(u)\, C_{wn}^{(2)}(t)$$

$$(w = x, y, z) \tag{69}$$

where $g_n(u)$ are given in Eq. (65).

When $i = 3$, Eq. (60b) becomes

$$\left(\gamma \frac{\partial^6}{\partial u^6} + \eta \frac{\partial^6}{\partial u^4 \partial v^2} + \lambda \frac{\partial^6}{\partial u^2 \partial v^4} + \rho \frac{\partial^6}{\partial v^6}\right) S_w^{(3)} = 0$$

$$(w = x, y, z) \tag{70}$$

and Eq. (60c) becomes

$$u = 0 \qquad \partial^n S_w^{(3)} / \partial u^n = C_{wn+1}^{(3)}$$

$$u = 1 \qquad \partial^n S_w^{(3)} / \partial u^n = C_{wn+4}^{(3)}$$

$$(w = x, y, z;\ n = 0,1,2) \tag{71}$$

where $C_{wn}^{(3)}$ $(w = x, y, z;\ n = 1, 2, 3, 4, 5, 6)$ are known constants.

After introducing $d_{wl,j}^{(3)}$ $(l, j = 1, 2, 3, 4, 5, 6)$ and letting $d_{wl,j}^{(3)} = 1$ for $l = j$ and $d_{wl,j}^{(3)} = 0$ for $l \neq j$, Eq. (71) can be further written as

$u = 0$

$$n = 0, \qquad S_w^{(3)} = \sum_{l=1}^{6} \left[d_{wl,1}^{(3)} C_{wl}^{(3)} \right]$$

$$n = 1, \qquad \partial S_w^{(3)} / \partial u = \sum_{l=1}^{6} \left[d_{wl,2}^{(3)} C_{wl}^{(3)} \right]$$

$$n = 2, \qquad \partial^2 S_w^{(3)} / \partial u^2 = \sum_{l=1}^{6} \left[d_{wl,3}^{(3)} C_{wl}^{(3)} \right]$$

$u = 1$

$$n=0,\qquad S_w^{(3)}=\sum_{l=1}^{6}\left[d_{wl,4}^{(3)}C_{wl}^{(3)}\right]$$

$$n=1,\qquad \partial S_w^{(3)}/\partial u=\sum_{l=1}^{6}\left[d_{wl,5}^{(3)}C_{wl}^{(3)}\right]$$

$$n=2,\qquad \partial^2 S_w^{(3)}/\partial u^2=\sum_{l=1}^{6}\left[d_{wl,6}^{(3)}C_{wl}^{(3)}\right]$$

$$(w=x,y,z) \tag{71a}$$

For each of $C_{wl}^{(3)}=C_{wl}^{(3)}(v)$ $(l=1,2,3,4,5,6)$ in Eq. (71a), a trial function $S_{wl}^{(3)}=S_{wl}^{(3)}(u,v)$ can be constructed. Putting all $S_{wl}^{(3)}(u,v)$ $(l=1,2,3,4,5,6)$ together, $S_w^{(3)}(u,v)=S_{w1}^{(3)}(u,v)+S_{w2}^{(3)}(u,v)+S_{w3}^{(3)}(u,v)+S_{w4}^{(3)}(u,v)+S_{w5}^{(3)}(u,v)+S_{w6}^{(3)}(u,v)=\sum_{l=1}^{6}S_{wl}^{(3)}$ is obtained.

Substituting $S_w^{(3)}(u,v)=\sum_{l=1}^{6}S_{wl}^{(3)}$ into Eq. (70), the following equation is obtained.

$$\left(\gamma\frac{\partial^6}{\partial u^6}+\eta\frac{\partial^6}{\partial u^4\partial v^2}+\lambda\frac{\partial^6}{\partial u^2\partial v^4}+\rho\frac{\partial^6}{\partial v^6}\right)\sum_{l=1}^{6}S_{wl}^{(3)}=0$$

$$(w=x,y,z) \tag{71b}$$

Solving the partial differential equation (71b) can be treated as solving the following six partial differential equations

$$\left(\gamma\frac{\partial^6}{\partial u^6}+\eta\frac{\partial^6}{\partial u^4\partial v^2}+\lambda\frac{\partial^6}{\partial u^2\partial v^4}+\rho\frac{\partial^6}{\partial v^6}\right)S_{wl}^{(3)}(u,v)=0$$

$$(w=x,y,z;l=1,2,3,4,5,6) \tag{71c}$$

Substituting $S_w^{(3)}(u,v)=\sum_{l=1}^{6}S_{wl}^{(3)}$ into (71a), the blending boundary constraints become $u=0$

$$n=0,\qquad \sum_{l=1}^{6}S_{wl}^{(3)}=\sum_{l=1}^{6}\left[d_{wl,1}^{(3)}C_{wl}^{(3)}\right]$$

$$n=1,\qquad \partial\left[\sum_{l=1}^{6}S_{wl}^{(3)}\right]\Big/\partial u=\sum_{l=1}^{6}\left[d_{wl,2}^{(3)}C_{wl}^{(3)}\right]$$

$$n=2,\qquad \partial^2\left[\sum_{l=1}^{6}S_{wl}^{(3)}\right]\Big/\partial u^2=\sum_{l=1}^{6}\left[d_{wl,3}^{(3)}C_{wl}^{(3)}\right]$$

$u=1$

$$n=0,\qquad \sum_{l=1}^{6}S_{wl}^{(3)}=\sum_{l=1}^{6}\left[d_{wl4}^{(3)}C_{wl}^{(3)}\right]$$

$$n=1,\qquad \partial\left[\sum_{l=1}^{6}S_{wl}^{(3)}\right]\Big/\partial u=\sum_{l=1}^{6}\left[d_{wl,5}^{(3)}C_{wl}^{(3)}\right]$$

$$n=2,\qquad \partial^2\left[\sum_{l=1}^{6}S_{wl}^{(3)}\right]\Big/\partial u^2=\sum_{l=1}^{6}\left[d_{wl,6}^{(3)}C_{wl}^{(3)}\right]$$

$$(w=x,y,z) \tag{71d}$$

The blending boundary constraints (71d) can be treated as the following six groups of blending boundary constraints (71e)

$u=0$

$$n=0,\qquad S_{wl}^{(3)}(u,v)=d_{wl,1}^{(3)}C_{wl}^{(3)}$$

$$n=1,\qquad \partial\left[S_{wl}^{(3)}(u,v)\right]/\partial u=d_{wl,2}^{(3)}C_{wl}^{(3)}$$

$$n=2,\quad \partial^2\left[S_{wl}^{(3)}(u,v)\right]/\partial u^2=d_{wl,3}^{(3)}C_{wl}^{(3)}$$

$u=1$

$$n=0,\qquad S_{wl}^{(3)}(u,v)=d_{wl,4}^{(3)}C_{wl}^{(3)}$$

$$n=1,\qquad \partial\left[S_{wl}^{(3)}(u,v)\right]/\partial u=d_{wl,5}^{(3)}C_{wl}^{(3)}$$

$$n=2,\qquad \partial^2\left[S_{wl}^{(3)}(u,v)\right]/\partial u^2=d_{wl,6}^{(3)}C_{wl}^{(3)}$$

$$(w=x,y,z;l=1,2,3,4,5,6) \tag{71e}$$

After adding Eq. (71e) for $l=1,2,3,4,5,6$ together, Eq. (71d) is obtained. It indicates that Eq. (71d) can be treated as Eq. (71e).

After the above discussions, it is clear that solving the partial differential equation (70) subjected to the blending boundary constraints (71) can be transformed into solving the above partial differential equations (71c) subjected to the blending boundary constraints (71e).

The analytical solution of the partial differential equation (71c) subjected to the blending boundary constraints (71e) can be taken to be the form below

$$S_{wl}^{(3)} = S_{wl}^{(3)}(u,v) = \left\{\sum_{n=1}^{6} a_{wl,n}^{(3)} u^{n-1} - \sum_{m=1}^{M} b_{wl,m}^{(3)} \sin m\pi u\right\} C_{wl}^{(3)}(v)$$
$$(w = x, y, z; l = 1, 2, 3, 4, 5, 6) \tag{72}$$

Substituting Eq. (72) into the blending boundary constraints (71e), solving for the unknown constants $a_{wl,n}^{(3)}$ $(w = x, y, z; l, n = 1, 2, 3, 4, 5, 6)$, and inserting the obtained unknown constants $a_{wl,n}^{(3)}$ $(w = x, y, z; l, n = 1, 2, 3, 4, 5, 6)$ back into Eq. (72), the following equation is obtained.

$$S_{wl}^{(3)}(u,v) = h_{wl}(u) C_{wl}^{(3)}(v) = \left\{\sum_{n=1}^{6} g_n(u) d_{wl,n}^{(3)} - \sum_{m=1}^{M} b_{wl,m}^{(3)} [m\pi f_m(u) - sin(m\pi u)]\right\} C_{wl}^{(3)}(v)$$
$$(w = x, y, z; l = 1, 2, 3, 4, 5, 6) \tag{73}$$

where $d_{wl,n}^{(3)} = 1$ for $n = l$ and $d_{wl,n}^{(3)} = 0$ for $n \neq l$; $g_l(u)$ $(l = 1, 2, \dots, 6)$ are the same as those given in Eq. (65); and $f_m(u) = g_2(u) + (-1)^m g_5(u)$.

The remaining unknown constants $b_{wl,m}^{(3)}$ $(w = x, y, z; l = 1, 2, 3, 4, 5, 6;$ *m*=1, 2, …, *M*) are determined by substituting Eq. (73) into the partial differential equation (71c) and minimizing the squared error sum at the points (u_i, v_j) $(i = 0, 1, 2, \cdots, I; j = 0, 1, 2, \cdots, J)$ which leads to

$$\sum_{m=1}^{M} \sum_{i=0}^{I} \sum_{j=0}^{J} b_{wl,m}^{(3)} \alpha_{wl,m}^{(3)}(u_i, v_j) \alpha_{wl,k}^{(3)}(u_i, v_j) = \sum_{i=0}^{I} \sum_{j=0}^{J} \varphi_{wl}^{(3)}(u_i, v_j) \alpha_{wl,k}^{(3)}(u_i, v_j)$$
$$(w = x, y, z;\ l = 1, 2, \cdots, 6; k = 1, 2, 3, \dots, M) \tag{74}$$

In the above equation, $\alpha_{wl,m}^{(3)}(u_i, v_j)$ and $\alpha_{wl,k}^{(3)}(u_i, v_j)$ are obtained as follows.

$$\alpha_{wl,m}^{(3)}(u_i, v_j) = \left\{\gamma\left[m\pi\frac{\partial^6 f_m(u_i)}{\partial u^6} + m^6\pi^6(\sin m\pi u_i)\right]C_{wl}^{(3)}(v_j)\right.$$
$$+ \eta\left[m\pi\frac{\partial^4 f_m(u_i)}{\partial u^4} - m^4\pi^4(\sin m\pi u_i)\right]\frac{\partial^2 C_{wl}^{(3)}(v_j)}{\partial v^2}$$
$$+ \lambda\left[m\pi\frac{\partial^2 f_m(u_i)}{\partial u^2} + m^2\pi^2(\sin m\pi u_i)\right]\frac{\partial^4 C_{wl}^{(3)}(v_j)}{\partial v^4}$$
$$\left.+ \rho[m\pi f_m(u_i) - (\sin m\pi u_i)]\frac{\partial^6 C_{wl}^{(3)}(v_j)}{\partial v^6}\right\}$$
$$(w = x, y, z;\ l = 1, 2, 3, 4, 5, 6; m = 1, 2, \dots, M) \quad (75)$$

and $\varphi_{wl}^{(3)}(u_i, v_j)$ is determined by the following equation

$$\varphi_{wl}^{(3)}(u_i, v_j) = \sum_{\boldsymbol{n=1}}^{\boldsymbol{6}} d_{wl,n}^{(3)}\left[\gamma\frac{\partial^6 g_l(u_i)}{\partial u^6}C_{wl}^{(3)}(v_j) + \eta\frac{\partial^4 g_l(u_i)}{\partial u^4}\frac{\partial^2 C_{wl}^{(3)}(v_j)}{\partial v^2} + \lambda\frac{\partial^2 g_l(u_i)}{\partial u^2}\frac{\partial^4 C_{wl}^{(3)}(v_j)}{\partial v^4}\right.$$
$$\left.+ \rho\, g_l(u_i)\frac{\partial^6 C_{wl}^{(3)}(v_j)}{\partial v^6}\right]$$
$$(w = x, y, z; l = 1, 2, 3, 4, 5, 6) \quad (76)$$

By solving Eq. (74), all the unknown constants $b_{wl,m}^{(3)}$ $(w = x, y, z;\ l = 1, 2, 3, 4, 5, 6; m = 1, 2, \dots, M)$ are determined. Substituting them back into Eq. (73), $S_{wl}^{(3)}(u, v)$ is obtained. Substituting the obtained $S_{wl}^{(3)}(u, v)$ into $S_w^{(3)}(u, v) = \sum_{l=1}^{6} S_{wl}^{(3)}$, the following blending surface functions are obtained

$$S_w^{(3)}(u, v) = \sum_{l=1}^{6}\left\{\sum_{n=1}^{6} g_n(u)\, d_{wl,n}^{(3)} - \sum_{m=1}^{M} b_{wl,m}^{(3)}\,[m\pi f_m(u) - (\sin m\pi u)]\right\}C_{wl}^{(3)}(v)$$
$$(w = x, y, z) \quad (77)$$

When $i = 4$, the boundary functions $C_{wn}^{(4)} = C_{wn}^{(4)}(v, t)(\ n = 1, 2, \dots, 6)$ are the functions of the parametric variable v and the time variable t. The partial differential equation (60b) becomes

$$\left(\gamma\frac{\partial^6}{\partial u^6} + \eta\frac{\partial^6}{\partial u^4\partial v^2} + \lambda\frac{\partial^6}{\partial u^2\partial v^4} + \rho\frac{\partial^6}{\partial v^6}\right)S_w^{(4)} = 0$$
$$(w = x, y, z) \quad (78)$$

and the blending boundary constraints (60c) becomes

$$u = 0 \qquad \partial^n S_w^{(4)}/\partial u^n = C_{wn+1}^{(4)}$$

$$u = 1 \qquad \partial^n S_w^{(4)} / \partial u^n = C_{wn+4}^{(4)}$$

$$(w = x, y, z;\ n = 0, 1, 2) \tag{79}$$

where $C_{wn}^{(4)}$ $(w = x, y, z;\ n = 1, 2, 3, 4, 5, 6)$ are known constants.

After introducing $d_{wl,j}^{(4)}$ $(l = 1, 2, 3, 4, 5, 6; j = 1, 2, 3, 4, 5, 6)$ and letting $d_{wl,j}^{(4)} = 1$ for $l = j$ and $d_{wl,j}^{(4)} = 0$ for $l \neq j$, Eq. (79) can be further written as

$u = 0$

$$n = 0, \qquad S_w^{(4)} = \sum_{l=1}^{6} \left[d_{wl,1}^{(4)} C_{wl}^{(4)} \right]$$

$$n = 1, \qquad \partial S_w^{(4)} / \partial u = \sum_{l=1}^{6} \left[d_{wl,2}^{(4)} C_{wl}^{(4)} \right]$$

$$n = 2, \qquad \partial^2 S_w^{(4)} / \partial u^2 = \sum_{l=1}^{6} \left[d_{wl,3}^{(4)} C_{wl}^{(4)} \right]$$

$u = 1$

$$n = 0, \qquad S_w^{(4)} = \sum_{l=1}^{6} \left[d_{wl,4}^{(4)} C_{wl}^{(4)} \right]$$

$$n = 1, \qquad \partial S_w^{(4)} / \partial u = \sum_{l=1}^{6} \left[d_{wl,5}^{(4)} C_{wl}^{(4)} \right]$$

$$n = 2, \qquad \partial^2 S_w^{(4)} / \partial u^2 = \sum_{l=1}^{6} \left[d_{wl,6}^{(4)} C_{wl}^{(4)} \right]$$

$$(w = x, y, z) \tag{79a}$$

For each of $C_{wl}^{(4)} = C_{wl}^{(4)}(v, t)$ $(l = 1, 2, 3, 4, 5, 6)$ in Eq. (79a), a trial function $S_{wl}^{(4)} = S_{wl}^{(4)}(u, v, t)$ can be constructed. Putting all $S_{wl}^{(4)}(u, v, t)$ $(l = 1, 2, 3, 4, 5, 6)$ together, $S_w^{(4)}(u, v, t) = S_{w1}^{(4)}(u, v, t) + S_{w2}^{(4)}(u, v, t) + S_{w3}^{(4)}(u, v, t) + S_{w4}^{(4)}(u, v, t) + S_{w5}^{(4)}(u, v, t) + S_{w6}^{(4)}(u, v, t) = \sum_{l=1}^{6} S_{wl}^{(4)}$ is obtained.

Substituting $S_w^{(4)}(u,v)=\sum_{l=1}^{6}S_{wl}^{(4)}$ into Eq. (78), the following equation is obtained.

$$\left(\gamma\frac{\partial^6}{\partial u^6}+\eta\frac{\partial^6}{\partial u^4\partial v^2}+\lambda\frac{\partial^6}{\partial u^2\partial v^4}+\rho\frac{\partial^6}{\partial v^6}\right)\sum_{n=1}^{6}S_{wl}^{(4)}=0$$

$$(w=x,y,z) \tag{79b}$$

Solving the partial differential equation (79b) can be treated as solving the following six partial differential equations

$$\left(\gamma\frac{\partial^6}{\partial u^6}+\eta\frac{\partial^6}{\partial u^4\partial v^2}+\lambda\frac{\partial^6}{\partial u^2\partial v^4}+\rho\frac{\partial^6}{\partial v^6}\right)S_{wl}^{(4)}(u,v,t)=0$$

$$(w=x,y,z;l=1,2,3,4,5,6) \tag{79c}$$

Substituting $S_w^{(4)}(u,v,t)=\sum_{l=1}^{6}S_{wl}^{(4)}$ into (79a), the blending boundary constraints become

$u=0$

$$n=0,\qquad \sum_{l=1}^{6}S_{wl}^{(4)}=\sum_{l=1}^{6}\left[d_{wl,1}^{(4)}C_{wl}^{(4)}\right]$$

$$n=1,\qquad \partial\left[\sum_{l=1}^{6}S_{wl}^{(4)}\right]\Big/\partial u=\sum_{l=1}^{6}\left[d_{wl,2}^{(4)}C_{wl}^{(4)}\right]$$

$$n=2,\qquad \partial^2\left[\sum_{l=1}^{6}S_{wl}^{(4)}\right]\Big/\partial u^2=\sum_{l=1}^{6}\left[d_{wl,3}^{(4)}C_{wl}^{(4)}\right]$$

$u=1$

$$n=0,\qquad \sum_{l=1}^{6}S_{wl}^{(4)}=\sum_{l=1}^{6}\left[d_{wl,4}^{(4)}C_{wl}^{(4)}\right]$$

$$n=1,\qquad \partial\left[\sum_{l=1}^{6}S_{wl}^{(4)}\right]\Big/\partial u=\sum_{l=1}^{6}\left[d_{wl,5}^{(4)}C_{wl}^{(4)}\right]$$

$$n=2,\qquad \partial^2\left[\sum_{l=1}^{6}S_{wl}^{(4)}\right]\Big/\partial u^2=\sum_{l=1}^{6}\left[d_{wl,6}^{(4)}C_{wl}^{(4)}\right]$$

$$(w=x,y,z) \tag{79d}$$

The blending boundary constraints (79d) can be treated as the following six groups of blending boundary constraints (79e)

$u = 0$

$$n = 0, \quad S_{wl}^{(4)}(u,v,t) = d_{wl,1}^{(4)} C_{wl}^{(4)}$$

$$n=1, \quad \frac{\partial\left[S_{wl}^{(4)}(u,v,t)\right]}{\partial u} = d_{wl,2}^{(4)} C_{wl}^{(4)}$$

$$n=2, \quad \partial^2\left[S_{wl}^{(4)}(u,v,t)\right]/\partial u^2 = d_{wl,3}^{(4)} C_{wl}^{(4)}$$

$u = 1$

$$n = 0, \quad S_{wl}^{(4)}(u,v,t) = d_{wl,4}^{(4)} C_{wl}^{(4)}$$

$$n=1, \quad \partial\left[S_{wl}^{(4)}(u,v,t)\right]/\partial u = d_{wl,5}^{(4)} C_{wl}^{(4)}$$

$$n=2, \quad \frac{\partial^2\left[S_{wl}^{(4)}(u,v,t)\right]}{\partial u^2} = d_{wl,6}^{(4)} C_{wl}^{(4)}$$

$$(w = x, y, z; l = 1, 2, 3, 4, 5, 6) \tag{79e}$$

Adding Eq. (79e) for $l = 1, 2, 3, 4, 5, 6$ together, Eq. (79d) is obtained. It indicates that Eq. (79d) can be treated as Eq. (79e).

After the above discussions, it is clear that solving the partial differential equation (78) subjected to the blending boundary constraints (79) can be transformed into solving the above partial differential equations (79c) subjected to the blending boundary constraints (79e).

The analytical solution of the partial differential equation (79c) subjected to the blending boundary constraints (79e) can be taken to be the form below

$$S_{wl}^{(4)} = S_{wl}^{(4)}(u,v,t) = \left\{\sum_{n=1}^{6} a_{wl,n}^{(4)} u^{n-1} + \sum_{m=1}^{M} b_{wl,m}^{(4)} (sin\, m\, \pi u)\right\} C_{wl}^{(4)}(v,t)$$

$$(w = x, y, z; l = 1, 2, 3, 4, 5, 6) \tag{80}$$

Substituting Eq. (80) into the blending boundary constraints (79e), solving for the unknown constants $a_{wl,n}^{(4)}$ ($w = x, y, z; l = 1, 2, 3, 4, 5, 6$; n=1, 2, 3, …, 6), and inserting the obtained unknown

constants $a_{wl,n}^{(4)}$ ($w = x, y, z; l = 1, 2, 3, 4, 5, 6$; n=1, 2, 3, …, 6) back into Eq. (80), the following equation is obtained

$$S_{wl}^{(4)} = h_{wl}(u)C_{wl}^{(4)}(v,t) = \left\{\sum_{n=1}^{6} g_n(u)d_{wl,n}^{(4)} - \sum_{m=1}^{M} b_{wl,m}^{(4)}\left[m\pi f_m(u) - (\sin m\pi u)\right]\right\} C_{wl}^{(4)}(v,t)$$

$$(w = x, y, z; l = 1, 2, 3, 4, 5, 6) \tag{81}$$

where $d_{wn,l} = 1$ for $n = l$ and $d_{wn,l} = 0$ for $n \neq l$; $g_l(u)$ $(l = 1, 2, \ldots, 6)$ are the same as those given in Eq. (65); and $f_m(u) = g_2(u) + (-1)^m g_5(u)$.

The remaining unknown constants $b_{wl,m}^{(4)}$ ($w = x, y, z; l = 1, 2, 3, 4, 5, 6$; m=1, 2, …, M) are determined by substituting Eq. (81) into the partial differential equation (79c) and minimizing the squared error sum at the points (u_i, v_j) $(i = 0, 1, 2, \cdots, I; j = 0, 1, 2, \cdots, J)$ and the given time t_0 which leads to

$$\sum_{m=1}^{M}\sum_{i=0}^{I}\sum_{j=0}^{J} b_{wl,m}^{(4)}\alpha_{wlm}(u_i, v_j, t_0)\alpha_{wlk}(u_i, v_j, t_0) = \sum_{i=0}^{I}\sum_{j=0}^{J} \varphi_{wl}(u_i, v_j, t_0)\alpha_{wlk}(u_i, v_j, t_0)$$

$$(w = x, y, z;\ l = 1, 2, \cdots, 6; k = 1, 2, 3, \ldots, M) \tag{82}$$

In the above equation, $\alpha_{wlm}(u_i, v_j, t_0)$ and $\alpha_{wlk}(u_i, v_j, t_0)$ are obtained as follows.

$$\begin{aligned}
&\alpha_{wlm}(u_i, v_j, t_0) \\
&\quad = \left\{\gamma\left[m\pi\frac{\partial^6 f_m(u_i)}{\partial u^6} + m^6\pi^6(\sin m\pi u_i)\right] C_{wl}^{(4)}(v_j, t_0)\right. \\
&\quad + \eta\left[m\pi\frac{\partial^4 f_m(u_i)}{\partial u^4} - m^4\pi^4(\sin m\pi u_i)\right]\frac{\partial^2 C_{wl}^{(4)}(v_j, t_0)}{\partial v^2} \\
&\quad + \lambda\left[m\pi\frac{\partial^2 f_m(u_i)}{\partial u^2} + m^2\pi^2(\sin m\pi u_i)\right]\frac{\partial^4 C_{wl}^{(4)}(v_j, t_0)}{\partial v^4} \\
&\quad \left. + \rho[m\pi f_m(u_i) - (\sin m\pi u_i)]\frac{\partial^6 C_{wl}^{(4)}(v_j, t_0)}{\partial v^6}\right\} \\
&\qquad (w = x, y, z;\ l = 1, 2, 3, 4, 5, 6; m = 1, 2, \ldots, M)
\end{aligned} \tag{83}$$

and $\varphi_{wl}(u_i, v_j, t_0)$ is determined by the following equation

$$\varphi_{wl}(u_i, v_j, t_0) = \sum_{n=1}^{6} d_{wl,n}^{(4)} \left[\gamma \frac{\partial^6 g_l(u_i)}{\partial u^6} C_{wl}^{(4)}(v_j, t_0) + \eta \frac{\partial^4 g_l(u_i)}{\partial u^4} \frac{\partial^2 C_{wl}^{(4)}(v_j, t_0)}{\partial v^2} \right.$$

$$\left. + \lambda \frac{\partial^2 g_l(u_i)}{\partial u^2} \frac{\partial^4 C_{wl}^{(4)}(v_j, t_0)}{\partial v^4} + \rho\, g_l(u_i) \frac{\partial^6 C_{wl}^{(4)}(v_j, t_0)}{\partial v^6} \right]$$

$$(w = x, y, z; l = 1, 2, 3, 4, 5, 6) \tag{84}$$

By solving Eq. (82), all the unknown constants $b_{wl,m}^{(4)}$ $(w = x, y, z;\ l = 1, 2, 3, 4, 5, 6; m = 1, 2, \dots, M)$ are obtained. Substituting them back into Eq. (81), $S_{wl}^{(4)}(u, v, t)$ is obtained. Substituting the obtained $S_{wl}^{(4)}(u, v, t)$ into $S_w^{(4)}(u, v) = \sum_{l=1}^{6} S_{wl}^{(4)}$, the following blending surface functions are obtained.

$$S_w^{(4)}(u, v, t) = \sum_{l=1}^{6} \left\{ \sum_{n=1}^{6} g_n(u)\, d_{wl,n}^{(4)} - \sum_{m=1}^{M} b_{wl,m}^{(4)} \left[m\pi f_m(u) - (\sin m\pi u) \right] \right\} C_{wl}^{(4)}(v, t)$$

$$(w = x, y, z) \tag{85}$$

Substituting the obtained solutions (64), (69), (77), and (85) into $S_w(u, v, t) = S_w^{(1)}(u) + S_w^{(2)}(u, t) + S_w^{(3)}(u, v) + S_w^{(4)}(u, v, t)$, the following function $S_w(u, v, t)$ defining a blending surface is obtained.

$$S_w(u, v, t) = \sum_{n=1}^{6} g_n(u) \left[C_{wn}^{(1)} + C_{wn}^{(2)}(t) \right]$$

$$+ \sum_{l=1}^{6} \left\{ \left\{ \sum_{n=1}^{6} g_n(u)\, d_{wl,n}^{(3)} - \sum_{m=1}^{M} b_{wl,m}^{(3)} \left[m\pi f_m(u) - (\sin m\pi u) \right] \right\} C_{wl}^{(3)}(v) \right.$$

$$\left. + \left\{ \sum_{n=1}^{6} g_n(u)\, d_{wl,n}^{(4)} - \sum_{m=1}^{M} b_{wl,m}^{(4)} \left[m\pi f_m(u) - (\sin m\pi u) \right] \right\} C_{wl}^{(4)}(v, t) \right\}$$

$$(w = x, y, z) \tag{86}$$

In the following subsection, the obtained blending surface function will be used to create various blending surfaces of C^2 continuity.

3.3.2 Accuracy, efficiency and effects of the second partial derivatives

In this section, the developed approximate analytical solution will be implemented and compared with the corresponding accurate closed form solution to demonstrate its good accuracy and high efficiency, the influence of second partial derivatives with respect to the parametric variable u on the continuity at timelines will be discussed, and constant surface blending with C^2 continuity will be examined.

3.3.2.1 Comparison between the obtained approximate analytical solution and the corresponding accurate closed form solution

First, a surface blending example is used to make a comparison between the developed approximate analytical solution and the accurate closed form solution. The comparison aims to demonstrate good accuracy and high efficiency of the developed approximate analytical solution.

This example is to create a time-dependent blending surface between two separate elliptic cylinders represented with time-dependent varying primary surfaces. The top primary surface for this example is defined by

$$x = ae^{t}\sin(2\pi v)$$
$$y = be^{-t}\cos(2\pi v)$$
$$z = h_1 + h_2u^2 \tag{87}$$

and the bottom primary surface is represented by the equation of

$$x = ce^{-t}\sin(2\pi v)$$
$$y = de^{t}\cos(2\pi v)$$
$$z = -h_3u^3 \tag{87a}$$

Deriving the first and second derivatives of Eqs. (87) and (87a) with respect to the parametric variable u, setting $u = u_1$ in Eq. (87) and its first and second partial derivatives with respect to the

parametric variable u, and setting $u = u_2$ in Eq. (87a) and its first and second partial derivatives with respect to the parametric variable u, the blending boundary conditions are obtained as

$$
\begin{aligned}
&u = 0\\
&S_x = ae^t \sin(2\pi v) \quad S_y = be^{-t}\cos(2\pi v) \quad S_z = h_1 + h_2 u_1^2\\
&\frac{\partial S_x}{\partial u} = 0 \qquad \frac{\partial S_y}{\partial u} = 0 \qquad \frac{\partial S_z}{\partial u} = -2h_2 u_1\\
&\frac{\partial^2 S_x}{\partial u^2} = 0 \qquad \frac{\partial^2 S_y}{\partial u^2} = 0 \qquad \frac{\partial^2 S_z}{\partial u^2} = 2h_2\\
&u = 1\\
&S_x = ce^{-t}\sin(2\pi v) \quad S_y = de^t\cos(2\pi v) \quad S_z = -h_3 u_2^3\\
&\frac{\partial S_x}{\partial u} = 0 \qquad \frac{\partial S_y}{\partial u} = 0 \qquad \frac{\partial S_z}{\partial u} = -3h_3 u_2^2\\
&\frac{\partial^2 S_x}{\partial u^2} = 0 \qquad \frac{\partial^2 S_y}{\partial u^2} = 0 \qquad \frac{\partial^2 S_z}{\partial u^2} = -6h_3 u_2
\end{aligned}
\tag{88}
$$

The functions in the blending boundary constraints (88) can be divided into $C_{x1}^{(4)} = ae^t sin2\pi v$, $C_{x4}^{(4)} = ce^{-t} sin2\pi v$, $C_{y1}^{(4)} = be^{-t} cos2\pi v$, $C_{y4}^{(4)} = de^t cos2\pi v$, $C_{z1}^{(1)} = h_1 + h_2 u_1^2$, $C_{z2}^{(1)} = -2h_2 u_1$, $C_{z3}^{(1)} = 2h_2$, $C_{z4}^{(1)} = -h_3 u_2^3$, $C_{z5}^{(1)} = -3h_3 u_2^2$, $C_{z6}^{(1)} = -6h_3 u_2$, $C_{x2}^{(4)} = C_{x3}^{(4)} = C_{x5}^{(4)} = C_{x6}^{(4)} = C_{y2}^{(4)} = C_{y3}^{(4)} = C_{y5}^{(4)} = C_{y6}^{(4)} = 0$, and $C_{xn}^{(1)} = C_{xn}^{(2)} = C_{xn}^{(3)} = C_{yn}^{(1)} = C_{yn}^{(2)} = C_{yn}^{(3)} = C_{zn}^{(2)} = C_{zn}^{(3)} = C_{zn}^{(4)} = 0\ (n = 1, 2, \cdots, 6)$.

Substituting $C_{xn}^{(1)} = C_{xn}^{(2)} = C_{xn}^{(3)} = 0$ $(n = 1, 2, \cdots, 6)$ into Eq. (86) for x component, the following equation is obtained.

$$S_x(u, v, t) = \sum_{l=1}^{6}\left\{\sum_{n=1}^{6} g_n(u)\, d_{xl,n}^{(4)} - \sum_{m=1}^{M} b_{xl,m}^{(4)} [m\pi f_m(u) - (\sin m\pi u)]\right\} C_{xl}^{(4)}(v, t) \tag{89}$$

Substituting $C_{x2}^{(4)} = C_{x3}^{(4)} = C_{x5}^{(4)} = C_{x6}^{(4)} = 0$ into the above equation, the above equation is further changed into the following equation

$$S_x(u,v,t) = \left\{\sum_{n=1}^{6} g_n(u)\, d_{x1,n}^{(4)} - \sum_{m=1}^{M} b_{x1,m}^{(4)} [m\pi f_m(u) - (\sin m\pi u)]\right\} C_{x1}^{(4)}(v,t)$$

$$+ \left\{\sum_{n=1}^{6} g_n(u)\, d_{x4,n}^{(4)} - \sum_{m=1}^{M} b_{x4,m}^{(4)} [m\pi f_m(u) - (\sin m\pi u)]\right\} C_{x4}^{(4)}(v,t)$$

Since $C_{x1}^{(4)} = C_{x1}^{(4)}(v,t) \neq 0$, $d_{x1,1}^{(4)} = 1$ and $d_{x1,n}^{(4)} = 0 \quad (n = 2,3,\cdots,6)$. Since $C_{x4}^{(4)} = C_{x4}^{(4)}(v,t) \neq 0$, $d_{x4,4}^{(4)} = 1$ and $d_{x4,n}^{(4)} = 0 \quad (n = 1,2,3,5,6)$. Substituting them into the above equation, the following equation is obtained.

$$S_x(u,v,t) = \left\{ g_1(u) - \sum_{m=1}^{M} b_{x1,m}^{(4)} [m\pi f_m(u) - (\sin m\pi u)]\right\} C_{x1}^{(4)}(v,t)$$

$$+ \left\{ g_4(u) - \sum_{m=1}^{M} b_{x4,m}^{(4)} [m\pi f_m(u) - (\sin m\pi u)]\right\} C_{x4}^{(4)}(v,t) \qquad (89a)$$

Setting $w = x$ and $l = 1$ in Eq. (82) and solving the M linear algebra equations given by Eq. (82), $b_{x1,m}^{(4)}$ $(m = 1,2,\cdots,M)$ is obtained. Setting $w = x$ and $l = 4$ in Eq. (82) and solving the M linear algebra equations given by Eq. (82), $b_{x4,m}^{(4)}$ $(m = 1,2,\cdots,M)$ is obtained. Substituting the obtained $b_{x1,m}^{(4)}$ and $b_{x4,m}^{(4)}$ into Eq. (86), the x component of the blending surface is obtained.

With the same treatment, the y component of the blending surface is obtained which has the form of

$$S_y(u,v,t) = \left\{ g_1(u) - \sum_{m=1}^{M} b_{y1,m}^{(4)} [m\pi f_m(u) - (\sin m\pi u)]\right\} C_{y1}^{(4)}(v,t)$$

$$+ \left\{ g_4(u) - \sum_{m=1}^{M} b_{y4,m}^{(4)} [m\pi f_m(u) - (\sin m\pi u)]\right\} C_{y4}^{(4)}(v,t) \qquad (89b)$$

Setting $w = z$, and substituting $C_{zn}^{(2)} = C_{zn}^{(3)} = C_{zn}^{(4)} = 0$ $(n = 1,2,\cdots,6)$ into Eq. (86), the following equation is obtained

$$S_z(u,v,t) = S_z(u) = \sum_{n=1}^{6} g_n(u) C_{zn}^{(1)} = (h_1 + h_2 u_1^2) g_1(u) - 2h_2 u_1 g_2(u) + 2h_2 g_3(u)$$

$$-h_3 u_2^3 g_4(u) - 3h_3 u_2^2 g_5(u) - 6h_3 u_2 g_6(u) \qquad (89c)$$

Using the approximate analytical solution (89a), (89b) and (89c) and the geometric parameters $a = 1.6, b = 1.0, c = 0.8, d = 0.6, h_1 = 2.0, h_2 = 3.0,$ and $h_3 = 5.0$, and setting $u_1 = 0.2$ for the top surface, $u_2 = 0.3$ for the bottom surface, the time variable to $t = 0.2$, and *M* in Eqs. (89a) and (89b) to 10, 15, and 20, and I and J in Eq. (82) to 100, which leads to 101 x 101 = 10,201 uniformly distributed vertices, the obtained blending surfaces are depicted in Figure 9.

For this example, the boundary constraints for $t = 0.2$ only involve constants, and sine and cosine functions. Therefore, the closed form solution is obtainable. Using the same geometric parameters, the mathematical expressions of the corresponding closed form solution is obtained as

$$S_x(u,v,t) = [C_{x1} e^{q_0 u} + C_{x2} e^{-q_0 u} + e^{q_1 u}(C_{x3} cosq_2 u + C_{x4} sinq_2 u)$$
$$+ e^{-q_1 u}(C_{x5} cosq_2 u + C_{x6} sinq_2 u)]\sin(2\pi v)$$

$$S_y(u,v,t) = \left[C_{y1} e^{q_0 u} + C_{y2} e^{-q_0 u} + e^{q_1 u}\left(C_{y3} cosq_2 u + C_{y4} sinq_2 u\right)\right.$$
$$\left. + e^{-q_1 u}\left(C_{y5} cosq_2 u + C_{y6} sinq_2 u\right)\right]\cos(2\pi v)$$

$$S_z(u) = \textstyle\sum_{n=1}^{6} C_{zn} u^{n-1} \qquad (90)$$

where the constants q_0 , q_1 , q_2 , C_{xn}, C_{yn} , and C_{zn} $(n = 1, 2, 3, 4, 5, 6)$ are determined by substituting Eq. (90) into the partial differential equation (5) and the blending boundary constraints (88).

The blending surface obtained from the accurate closed form solution is also shown in Figure 9 where CFS indicates the closed form solution. The second image of the second row of the figure shows the profile curves of the blending surface obtained from $M = 10,15, and\ 20$ of the proposed approach and the closed form solution. The profile curves are magnified in the third image of the second row with the two innermost profile curves from *M*=10. The second row of Figure 9 shows

no visible difference between profile curves from M=15 and M=20 of the proposed approach and the closed form solution.

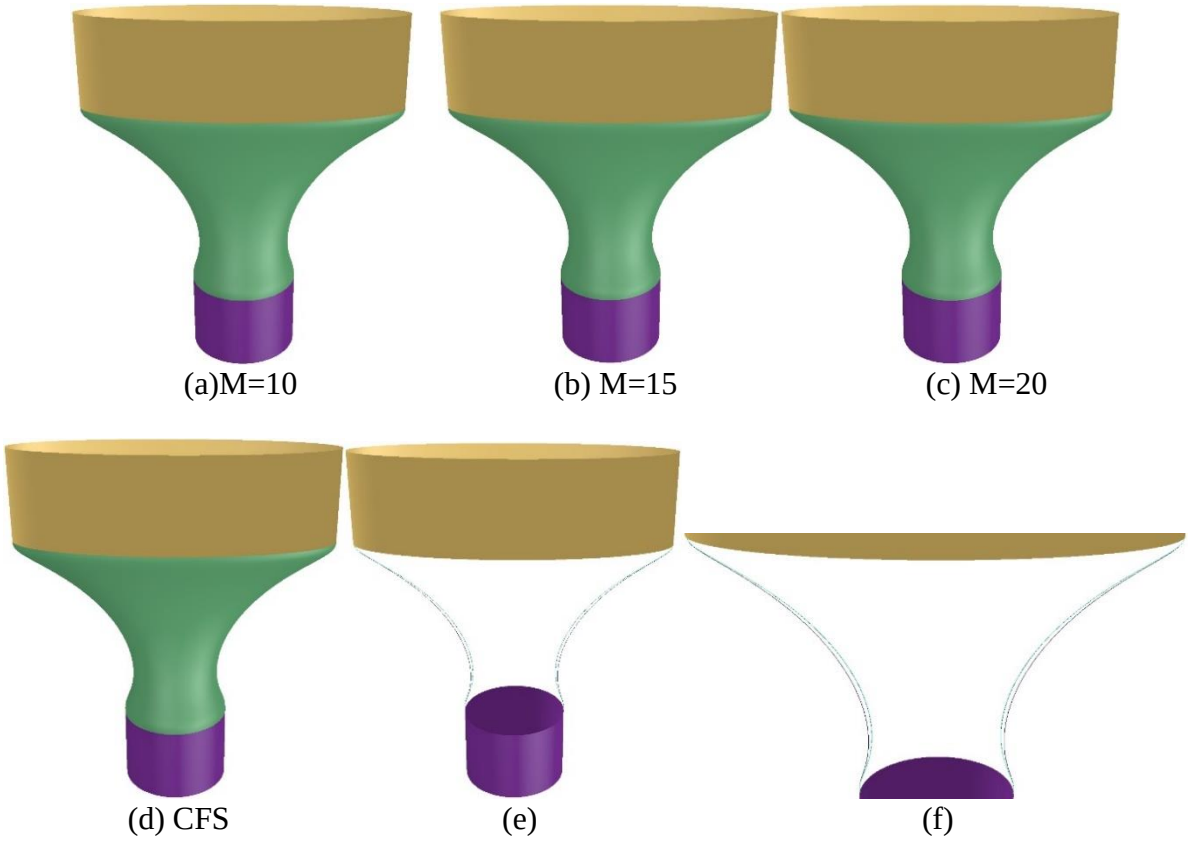


Figure 9: Blending surfaces generated by different values of M in Eqs. (89a) and (89b) and closed form solution (90) where (a), (b) and (c) are from different values of M in Eqs. (89a) and (89b), (d) is from Eq. (90) of the closed form solution, and (e) and (f) depicts the profile curves of the blending surfaces shown in (a), (b), (c) and (d) where (f) is the middle part of (e), and (e) and (f) are used to show the differences between blending surfaces created with the closed form solution and the approximate analytical solution with $M = 10$, 15 and 20.

Next, the errors between the approximate analytical solution and the exact closed form solution are quantified. Since the blending surfaces obtained from the close form solution and the approximate analytical solution are the functions of the parametric variables u and v, the errors between them are also the function of parametric variables u and v. In order to quantify the errors

between them, we discretize the blending surfaces first. If the two methods give the same result, the points at the same parametric positions obtained with the two methods should be the same points. Therefore, the Euclidean distance between two corresponding points is the most suitable in evaluating the errors between the two methods. If $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ are used to indicate the i^{th} point on the blending surfaces respectively obtained from the developed approximate analytical solution and the exact closed form solution, i. e., $\boldsymbol{p}_i = \boldsymbol{S}_M(u_i, v_i, t = 0.2)$ and $\boldsymbol{q}_i = \boldsymbol{S}_{CFS}(u_i, v_i, t = 0.2)$, the Euclidean distance between the i^{th} point on the two blending surfaces is $d(\boldsymbol{p}_i, \boldsymbol{q}_i)$ (Mora et al., 2016). The errors between the proposed approximate analytical solution and the exact closed form solution are calculated with the following equations

$$E_1 = max\{\, d(\boldsymbol{p}_i, \boldsymbol{q}_i)\} \qquad E_2 = \frac{1}{I}\sum_{i=1}^{I} d(\boldsymbol{p}_i, \boldsymbol{q}_i)$$

$$E_3 = max\left\{\frac{d(\boldsymbol{p}_i, \boldsymbol{q}_i)}{D}\right\} \qquad E_4 = \frac{1}{I}\sum_{i=1}^{I} \frac{d(\boldsymbol{p}_i, \boldsymbol{q}_i)}{D}$$

In the equation, E_1, E_2, E_3, and E_4 indicate absolute maximum error (AME), absolute average error (AAE), relative maximum error (RME), and relative average error (RAE), respectively, I is the total number of all the points on a blending surface, and D is the maximum distance between two points of the blending surface with the same parametric values u but different parametric values v. The errors obtained from the above equation are given in Table 1. The computational time (CPU) used to determine all the unknown constants and generate the blending surfaces with $M = 10$, 15 and 20 is also given in the same table.

The data in Table 1 demonstrate good accuracy and high computational efficiency of the proposed approximate analytical solution. When M increases from 10 to 20, the relative average error (RAE) of the proposed approach decreases from 7.95×10^{-3} to 8.31×10^{-6}. That is to say, when $M = 10$, the average error is three orders smaller than the maximum distance D; and when $M = 20$, the average error is six orders smaller than the maximum distance D. They indicate the

errors are small and converge quickly. With the increase of *M*, the computational time for approximate analytical solution rises. When *M* increases from 10 to 20, the computational time of the approximate analytical solution raises from 28.84 ms to 108.22 ms which is at about the same order as the accurate closed form solution (14.88 ms). Unlike the accurate closed form solution which is applicable to simple blending boundary constraints involving constants, sine and cosine functions, and exponential functions only, the proposed approach is applicable to various complicated blending boundary constraints.

Table 1. Accuracy and efficiency of the variable decomposition-based approximate analytical approach at the time instant t = 0.2

M	10 (Fig. 9a)	15 (Fig. 9b)	20 (Fig. 9c)	CFS (Fig. 9d)
AAE	1.71×10^{-2}	1.65×10^{-3}	1.79×10^{-5}	0
AME	4.55×10^{-2}	4.85×10^{-3}	6.40×10^{-5}	0
RAE	7.95×10^{-3}	7.65×10^{-4}	8.31×10^{-6}	0
RME	2.11×10^{-2}	2.25×10^{-3}	2.97×10^{-5}	0
Time (ms)	28.84	59.87	108.22	14.88

3.3.2.2 Effects of second partial derivatives

Unlike the surface blending with tangential continuity presented in (You et al., 2012), which only maintains the continuities of the position functions and first partial derivatives at trimlines, the surface blending with C^2 continuity proposed in this thesis further introduces the second partial derivatives at trimlines to achieve higher order continuity. In this subsection, how the second partial

derivatives at trimlines affect the continuity between the blending surface and primary surfaces will be discussed.

Setting $t = 0.05$ and keeping all the position functions, all the first partial derivatives, and the second partial derivatives of the surface functions x and y for both blending surface and primary surfaces at the timelines same as those given in Eq. (89), the following three different cases of the second partial derivatives of the surface function z shown in Table 2 are considered where TPS, BS and BPS stand for the top primary surface, blending surface, and bottom primary surface, respectively.

Case 1: The blending surface and primary surfaces have the same second partial derivatives at both trimlines (Figure 10(a)).

Case 2: The blending surface has the second partial derivative $-10h_2$ at $u = 0$ and $-5.4h_3$ at $u = 1$ (Figure 10(b)).

Case 3: The blending surface has the second partial derivative $-20h_2$ at $u = 0$ and $-10.8h_3$ at $u = 1$ (Figure 10(c)).

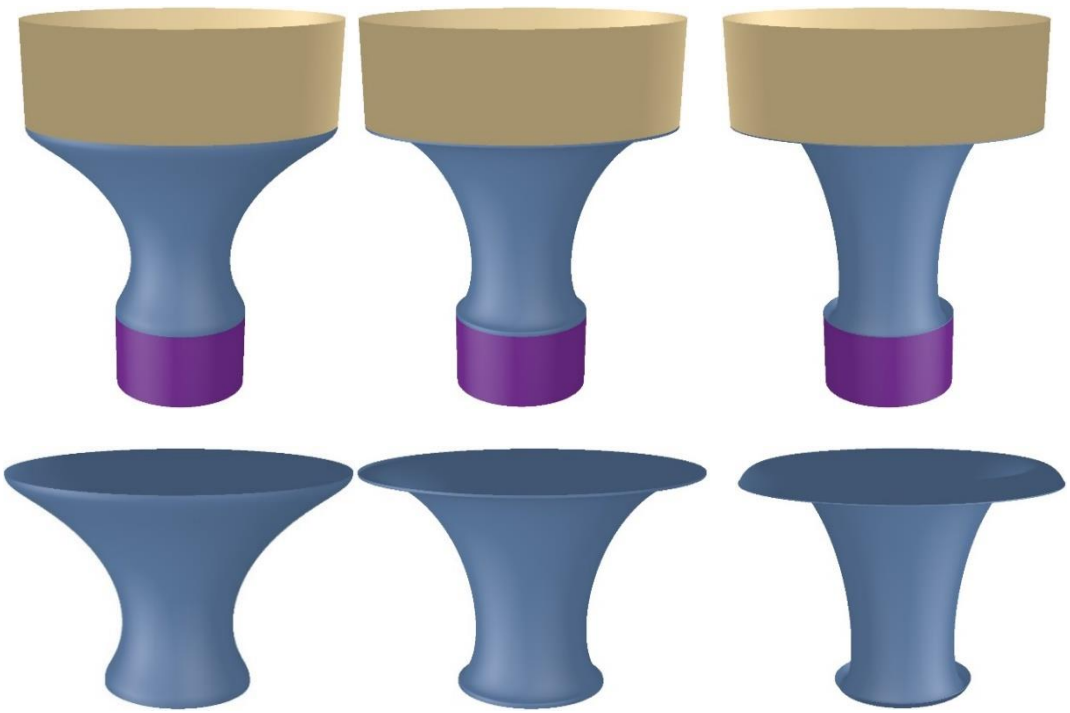
Table 2. Second partial derivatives of the surface function z with respect to u at trimlines used in the variable decomposition-based approximate analytical approach

	TPS $u = 0.2$	BS $u = 0$	BPS $u = 0.3$	BS $u = 1$	Figure
$\frac{\partial^2 z}{\partial u^2}$	$2h_2$	$2h_2$	$-1.8h_3$	$-1.8h_3$	10(a)
	$2h_2$	$10h_2$	$-1.8h_3$	$-5.4h_3$	10(b)
	$2h_2$	$20h_2$	$-1.8h_3$	$-10.8h_3$	10(c)

For all the three cases, the top primary surface has the second partial derivative $2h_2$ at $u = 0.2$ which is the trimline $u = 0$ of the blending surface and the bottom primary surface has the second partial derivative $-1.8h_3$ at $u = 0.3$ which is the trimline $u = 1$ of the blending surface.

The obtained results for the three cases are shown in Figure 10 where the surfaces in the bottom row show the blending shapes only. They are used to demonstrate how the second partial derivatives affect the continuity at the trimlines.

Comparing these images, the following conclusions can be drawn: (1) When the blending surface and primary surfaces have the same second partial derivatives at the trimlines, good continuity between the blending surface and the primary surfaces is obtained as shown in Figure 10(a); (2) Different second partial derivatives between the blending surface and primary surfaces will cause discontinuity at the trimlines as indicated in Figures 10(b) and 10(c); (3) The larger is the difference of the second partial derivatives between the blending surface and primary surfaces, the more serious is the discontinuity as demonstrated in Figure 10(c).



(a) (b) (c)

Figure 10: Effects of second partial derivatives

3.3.3 Applications

The proposed approach can be used to blend both time-dependent varying parametric surfaces and time-independent constant parametric surfaces. In the subsections below, some examples will be given to demonstrate the applications of the proposed approach in surface blending of varying and constant parametric surfaces.

3.3.3.1 Blending of varying parametric surfaces

Surface blending between time-dependent varying parametric surfaces can be well demonstrated with the primary surfaces defined by Eqs. (87) and (88). With the developed method and setting the time variable $t = 0, t = 0.1, \text{and } t = 0.2$, the blending surfaces at these time instants are obtained and depicted in Figure 11 where (a)-(c) are from the front view and (d)-(f) are from the side view.

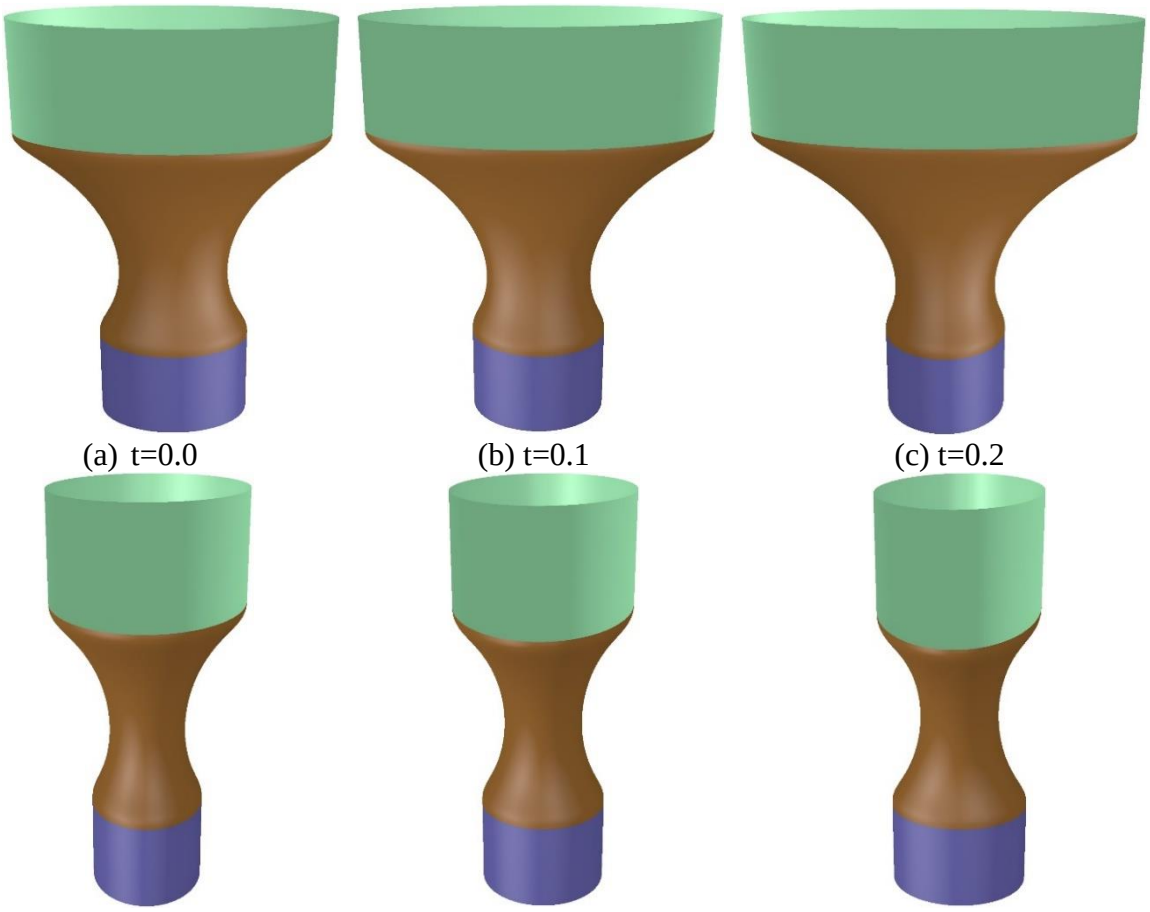


Figure 11: Blending surfaces at different time instants

The images shown in Figure 11 indicate that at different time instants, the proposed approach always creates blending surfaces of C^2 continuity to smoothly connect primary surfaces together.

3.3.3.2 Blending of constant parametric surfaces

In this section, how the developed approach can be used to blend constant parametric surfaces with C^2 continuity will be investigated. This example is to blend two perpendicular planes which is a very common engineering problem. The blending boundary constraints for this surface blending can be taken to be

$$
\begin{array}{lll}
u = 0 & & \\
S_x = 0 & S_y = h_0 + h_1 u_1 & S_z = pv \\
\frac{\partial S_x}{\partial u} = 0 & \frac{\partial S_y}{\partial u} = -h_1 & \frac{\partial S_z}{\partial u} = 0 \\
\frac{\partial^2 S_x}{\partial u^2} = 0 & \frac{\partial^2 S_y}{\partial u^2} = 0 & \frac{\partial^2 S_z}{\partial u^2} = 0 \\
u = 1 & & \\
S_x = s_0 + s_1 u_2^2 & S_y = 0 & S_z = pv \\
\frac{\partial S_x}{\partial u} = 2 s_1 u_2 & \frac{\partial S_y}{\partial u} = 0 & \frac{\partial S_z}{\partial u} = 0 \\
\frac{\partial^2 S_x}{\partial u^2} = 2 s_1 & \frac{\partial^2 S_y}{\partial u^2} = 0 & \frac{\partial^2 S_z}{\partial u^2} = 0
\end{array}
\tag{91}
$$

The above blending boundary constraints can also be written as:

$$
\begin{aligned}
& u = 0 \\
& [S_x \quad S_y \quad S_z] = [0 \quad h_0 + h_1 u_1 \quad pv] \\
& \left[\frac{\partial S_x}{\partial u} \quad \frac{\partial S_y}{\partial u} \quad \frac{\partial S_z}{\partial u}\right] = [0 \quad -h_1 \quad 0] \\
& \left[\frac{\partial^2 S_x}{\partial u^2} \quad \frac{\partial^2 S_y}{\partial u^2} \quad \frac{\partial^2 S_z}{\partial u^2}\right] = [0 \quad 0 \quad 0]
\end{aligned}
$$

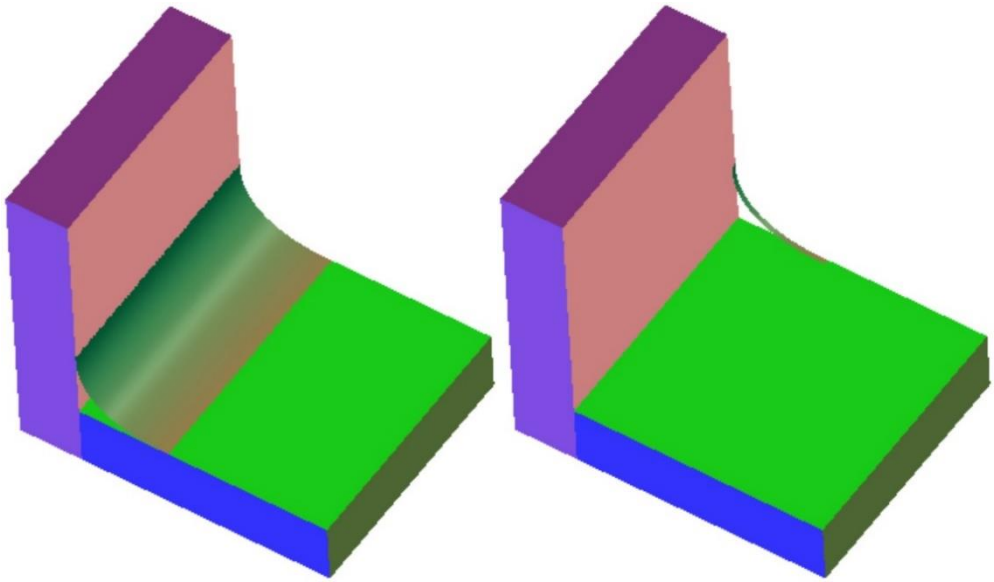
$$u = 1$$

$$[S_x \quad S_y \quad S_z] = [s_0 + s_1 u_2^2 \quad 0 \quad pv]$$

$$\left[\frac{\partial S_x}{\partial u} \quad \frac{\partial S_y}{\partial u} \quad \frac{\partial S_z}{\partial u}\right] = [2s_1 u_2 \quad 0 \quad 0]$$

$$\left[\frac{\partial^2 S_x}{\partial u^2} \quad \frac{\partial^2 S_y}{\partial u^2} \quad \frac{\partial^2 S_z}{\partial u^2}\right] = [2s_1 \quad 0 \quad 0]$$

The geometric parameters in Eq. (91) are taken to be $h_0 = s_0 = 0,\ h_1 = 1.9,\ s_1 = 2\ , p = 2,$ $u_1 = 0.3$ and $u_2 = 0.6$. The developed approximate analytical solution is used to create the blending surface which is shown in Figure 12 where Figure 12(a) shows the whole blending surface and Figure 12(b) shows a very small part of the blending surface. It is clear that the proposed approach is effective in blending constant parametric surfaces with C^2 continuity.



(a) Blending surface (b) Part of blending surface

Figure 12: Blending between perpendicular planes

The second example is to blend the frustum of a declined circular cone and a declined plane at a specified circle with C^2 continuity. The top parametric surface can be described by the following equations

$$x = ru^2 cosv$$

$$y = ru^2 sinv$$

$$z = h_0 + h_1 u + hcosv \tag{92}$$

and the bottom parametric surface is represented by the equation of

$$x = Rucos\alpha cosv$$

$$y = Rusinv$$

$$z = Rusin\alpha cosv \tag{93}$$

If the top surface is to be connected to the blending surface at $u = u_0$, and the bottom surface is to be connected to the blending surface at $u=u_1$, the blending boundary constraints can be formulated as

$$u = 0$$

$$S_x = ru_0^2 cosv \quad S_y = ru_0^2 sinv \quad S_z = h_0 + h_1 u_0 + hcosv$$

$$\frac{\partial S_x}{\partial u} = -2ru_0 cosv \quad \frac{\partial S_y}{\partial u} = -2ru_0 sinv \quad \frac{\partial S_z}{\partial u} = -h_1$$

$$\frac{\partial^2 S_x}{\partial u^2} = 2rcosv \qquad \frac{\partial^2 S_y}{\partial u^2} = 2rsinv \qquad \frac{\partial^2 S_z}{\partial u^2} = 0$$

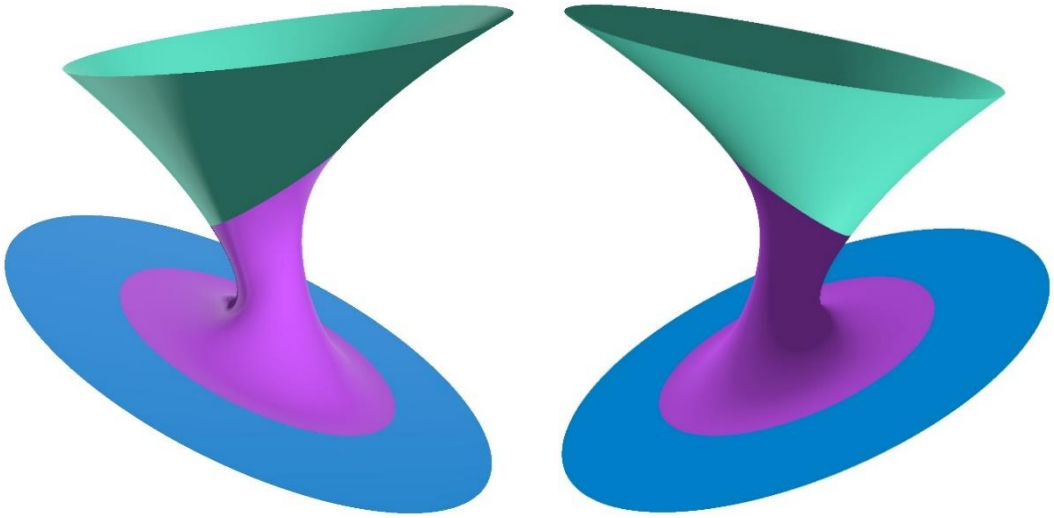
$$u = 1$$

$$S_x = Ru_1 cos\alpha cosv \quad S_y = Ru_1 sinv \quad S_z = Ru_1 sin\alpha cosv$$

$$\frac{\partial S_x}{\partial u} = Rcos\alpha cosv \quad \frac{\partial S_y}{\partial u} = Rsinv \quad \frac{\partial S_z}{\partial u} = Rsin\alpha cosv$$

$$\frac{\partial^2 S_x}{\partial u^2} = 0 \qquad \frac{\partial^2 S_y}{\partial u^2} = 0 \qquad \frac{\partial^2 S_z}{\partial u^2} = 0 \tag{94}$$

Taking $u_0 = 0.6$ and $u_1 = 0.8$ in Eq. (94), and setting the geometric parameters to: $r = 1.8$, $R = 2.0$, $h_0 = 0$, $h = 0.5, h_1 = 3$, and $\alpha = -20^0$, the blending surface is obtained and depict in Figure 13 where the two images shown in Figure 13(a) and Figure 13(b) are from different view angles.



(a) (b)

Figure 13: Blending between the frustum of a declined circular cone and a declined plane at a specified circle

3.4 Differential property-based approximate analytical approach

The variable decomposition-based approximate analytical approach is effective in dealing with various C^2 continuous blending between time-dependent and time-independent parametric surfaces. However, this approach can be improved further.

In the decomposed boundary functions $C_n^{(3)}(v)$ and $C_n^{(4)}(v,t)(n = 1,2,3, \dots 6)$, some of them may have the differential properties

$$\frac{\partial^2 C_n^{(i)}}{\partial v^2} = 0$$

$$(i = 3,4)$$

Or

$$\frac{\partial^{2k} C_n^{(i)}}{\partial v^2} = \beta_n^k C_n^{(i)}$$

$$(i = 3,4)$$

As discussed in Section 3.2, when the blending boundary constraints have the above differential properties, the closed form solutions are obtainable. According to this consideration, the variable-based decomposition discussed in section 3.3 may not be optimal. The optimal decomposition of the blending boundary constraints should consider the above differential properties to further simplify the mathematical treatment.

In this section, this differential property-based decomposition will be investigated, and a differential property-based approximate analytical approach will be developed to tackle C^2 continuous blending between time-dependent and time-independent parametric surfaces. It integrates the advantage of the closed form solution-based approximate analytical approach in considering different differential properties and the strength of the variable decomposition-based approximate analytical approach in obtaining the approximate analytical solutions for the blending boundary functions which have no differential properties.

3.4.1 Transformed mathematical model and its solution

According to different differential properties, the blending boundary functions $f_\xi(\xi = x, y, z; i = 1,2, ...)$ given in the blending boundary constraints (1) can be classified the following three groups

$$\bar{C}_{\xi n} = \bar{C}_{\xi n}(v,t) = \sum_j \bar{a}_{\xi j,n}\bar{f}_{\xi j}(v,t)$$

$$\hat{C}_{\xi n} = \hat{C}_{\xi n}(v,t) = \sum_k \hat{a}_{\xi k,n}\hat{f}_{\xi k}(v,t)$$

$$\tilde{C}_{\xi n} = \tilde{C}_{\xi n}(v,t) = \sum_j \tilde{a}_{\xi j,n}\tilde{f}_{\xi l}(v,t)$$

$$(j,k,l = 1,2,\ldots;1,2,..,6) \tag{95}$$

where the overbar "¯" indicates that the blending boundary functions have the following differential property (97), the overhead symbol "^" denotes that the blending boundary functions have the

following differential property (98), and the overhead symbol "~" shows that the blending boundary functions have no the differential properties (97) and (98).

The first group of functions $\bar{f}_{\xi j}(v,t)$ $(\xi = x,y,z; j = 1,2,\cdots)$ has the following differential properties

$$\begin{gathered}\frac{\partial^2 \bar{f}_{\xi j}(v,t)}{\partial v^2} = 0 \\ (\xi = x,y,z; j = 1,2,\cdots)\end{gathered} \tag{96}$$

The second group of functions $\hat{f}_{k\xi}(v,t)$ $(\xi = x,y,z; k = 1,2,\cdots)$ has the following differential properties

$$\begin{gathered}\frac{\partial^{2\bar{n}} \hat{f}_{k\xi}(v,t)}{\partial v^{2n}} = \beta_{\xi k}^{\bar{n}} \hat{f}_{k\xi}(v,t) \\ (\xi = x,y,z; k = 1,2,\cdots; \bar{n} = 1,2,3)\end{gathered} \tag{97}$$

The third group of functions $\tilde{f}_{\xi l}(v,t)$ $(\xi = x,y,z; l = 1,2,\cdots)$ has none of the differential properties (96) and (97).

After decomposing the functions in the blending boundary constraints (1) into the above the three groups of boundary functions, the blending boundary constraints (1) become

$$\begin{aligned}
u = 0 \qquad & S_\xi(0,v,t) = \bar{C}_{\xi 1} + \hat{C}_{\xi 1} + \tilde{C}_{\xi 1} \\
& \partial S_\xi(0,v,t)/\partial u = \bar{C}_{\xi 2} + \hat{C}_{\xi 2} + \tilde{C}_{\xi 2} \\
& \partial^2 S_\xi(0,v,t)/\partial u^2 = \bar{C}_{\xi 3} + \hat{C}_{\xi 3} + \tilde{C}_{\xi 3} \\
u = 1 \qquad & S_\xi(1,v,t) = \bar{C}_{\xi 4} + \hat{C}_{\xi 4} + \tilde{C}_{\xi 4} \\
& \partial S_\xi(1,v,t)/\partial u = \bar{C}_{\xi 5} + \hat{C}_{\xi 5} + \tilde{C}_{\xi 5} \\
& \partial^2 S_\xi(1,v,t)/\partial u^2 = \bar{C}_{\xi 6} + \hat{C}_{\xi 6} + \tilde{C}_{\xi 6} \\
& (\xi = x,y,z)
\end{aligned} \tag{98}$$

If the mathematical functions $S_\xi(u,v,t)$ $(\xi = x,y,z)$ of the blending surface is decomposed into the corresponding three parts: $\bar{S}_\xi = \bar{S}_\xi(u,v,t)$, $\hat{S}_\xi = \hat{S}_\xi(u,v,t)$ and $\tilde{S}_\xi = \tilde{S}_\xi(u,v,t)$, and substitute $S_\xi(u,v,t) = \bar{S}_\xi + \hat{S}_\xi + \tilde{S}_\xi$ into Eq. (5), the following equation is reached.

$$(\gamma\frac{\partial^6}{\partial u^6}+\eta\frac{\partial^6}{\partial u^4\partial v^2}+\lambda\frac{\partial^6}{\partial u^2\partial v^4}+\rho\frac{\partial^6}{\partial v^6})$$

$$\left[\bar{S}_\xi(u,v,t)+\hat{S}_\xi(u,v,t)+\tilde{S}_\xi(u,v,t)\right]=0 \qquad (99)$$

On the trimlines $u=0$ and $u=1$, $\bar{S}_\xi(u,v,t)$, $\hat{S}_\xi(u,v,t)$ and $\tilde{S}_\xi(u,v,t)$ correspond to $\bar{C}_{n\xi}(v,t)$, $\hat{C}_{n\xi}(v,t)$ and $\tilde{C}_{n\xi}(v,t)$, respectively. Solving Eq. (5) subjected to (1) can be transformed into solving each of the terms in the square bracket of Eq. (99) subjected to the blending boundary constraints consisting of the corresponding terms in Eq. (98).

3.4.1.1 Resolution for the first group of boundary functions

For the first group of boundary functions $\bar{f}_i(v,t)$ $(i=1,2,\cdots)$, the corresponding mathematical expressions of the blending surface can be taken to be

$$\bar{S}_\xi(u,v,t)=\sum_j H_{\xi j}(u)\bar{f}_{\xi j}(v,t)$$

$$(\xi=x,y,z;\ \mathrm{j}=1,2,\cdots) \qquad (100)$$

The partial differential equations (99) corresponding to the mathematical expressions $\bar{S}_\xi(u,v,t)$ are

$$(\gamma\frac{\partial^6}{\partial u^6}+\eta\frac{\partial^6}{\partial u^4\partial v^2}+\lambda\frac{\partial^6}{\partial u^2\partial v^4}+\rho\frac{\partial^6}{\partial v^6})\bar{S}_\xi(u,v,t)=0 \qquad (101)$$

The blending boundary constraints (98) corresponding to the mathematical expressions $\bar{S}_\xi(u,v,t)$ are

$$\mathrm{u}=0 \qquad \bar{S}_\xi(0,v,t)=\textstyle\sum_j \bar{a}_{\xi j,1}\bar{f}_{\xi j}(v,t)$$

$$\partial\bar{S}_\xi(0,v,t)/\partial u=\textstyle\sum_j \bar{a}_{\xi j,2}\bar{f}_{\xi j}(v,t)$$

$$\partial^2\bar{S}_\xi(0,v,t)/\partial u^2=\sum_j \bar{a}_{\xi j,3}\bar{f}_{\xi j}(v,t)$$

$$u=1 \qquad \bar{S}_\xi(1,v,t)=\sum_j \bar{a}_{\xi j,4}\bar{f}_{\xi j}(v,t)$$

$$\partial \bar{S}_{\xi}(1,v,t)/\partial u = \sum_{j} \bar{a}_{\xi j,5}\, \bar{f}_{\xi j}(v,t)$$

$$\partial^2 \bar{S}_{\xi}(1,v,t)/\partial u^2 = \sum_{j} \bar{a}_{\xi j,6}\, \bar{f}_{\xi j}(v,t)$$

$$(\xi = x,y,z) \tag{102}$$

Substituting Eq. (100) into (101) and considering the differential properties (96) and the blending boundary constraints (102), the following equation is obtained.

$$\frac{\partial^6 \bar{H}_{\xi j}(u)}{\partial u^6} = 0$$

$$(\xi = x,y,z; j = 1,2,\cdots) \tag{103}$$

subjected to the following blending boundary constraints

$$u = 0 \qquad \bar{H}_{\xi j}(0) = \bar{a}_{\xi j,1}$$

$$\partial \bar{H}_{\xi j}(0)/\partial u = \bar{a}_{\xi j,2}$$

$$\partial^2 \bar{H}_{\xi j}(0)/\partial u^2 = \bar{a}_{\xi j,3}$$

$$u = 1 \qquad \bar{H}_{\xi j}(1) = \bar{a}_{\xi j,4}$$

$$\partial \bar{H}_{\xi j}(1)/\partial u = \bar{a}_{\xi j,5}$$

$$\partial^2 \bar{H}_{\xi j}(1)/\partial u^2 = \bar{a}_{\xi j,6}$$

$$(\xi = x,y,z; j = 1,2,\cdots) \tag{104}$$

The solution to the sixth order ordinary differential equation (101) can be taken to be

$$\bar{H}_{\xi j}(u) = \sum_{n=0}^{5} \bar{b}_{\xi j,n} u^n$$

$$(\xi = x,y,z; j = 1,2,\cdots) \tag{105}$$

where $\bar{b}_{\xi j,n}$ $(n = 1,2,\cdots,6)$ are unknown constants.

Eq. (105) have exactly satisfied the sixth order ordinary differential equation (103). Substituting Eq. (105) into the boundary conditions (104), all the unknown constants $\bar{b}_{\xi j}$ are

determined. Introducing the determined unknown constants $\bar{b}_{\xi j}$ into (105), and then substituting (105) into (100), the following equation is obtained.

$$\bar{S}_{\xi}(u,v,t) = \sum_{j}\sum_{n=1}^{6} g_n(u)\bar{a}_{\xi j,n}\bar{f}_{\xi j}(v,t)$$

$$(\xi = x,y,z;\ \mathrm{j} = 1,2,\cdots) \tag{106}$$

where

$$g_1(u) = 1 - 10u^3 + 15u^4 - 6u^5$$

$$g_2(u) = (1 - 6u^2 + 8u^3 - 3u^4)u$$

$$g_3(u) = (0.5 - 1.5u + 1.5u^2 - 0.5u^3)u^2$$

$$g_4(u) = (10 - 15u + 6u^2)u^3$$

$$g_5(u) = (-4 + 7u - 3u^2)u^3$$

$$g_6(u) = (0.5 - u + 0.5u^2)u^3 \tag{107}$$

3.4.1.2 Resolution for the second group of boundary functions

For the second group of functions $\hat{f}_{\xi k}(v,t)$ $(k = 1,2,\cdots)$, the corresponding mathematical expressions of the blending surface can be taken to be

$$\hat{S}_{\xi}(u,v,t) = \sum_{k} \hat{H}_{\xi k}(u)\hat{f}_{\xi k}(v,t)$$

$$(\xi = x,y,z) \tag{108}$$

The partial differential equations (99) corresponding to the mathematical expressions $\hat{S}_{\xi}(u,v,t)$ are

$$(\gamma\frac{\partial^6}{\partial u^6} + \eta\frac{\partial^6}{\partial u^4\partial v^2} + \lambda\frac{\partial^6}{\partial u^2\partial v^4} + \rho\frac{\partial^6}{\partial v^6})\hat{S}_{\xi}(u,v,t) = 0 \tag{109}$$

The blending boundary constraints (98) corresponding to the mathematical expressions $\hat{S}_{\xi}(u,v,t)$ are

$$u = 0 \qquad \hat{S}_\xi(0,v,t) = \sum_k \hat{a}_{\xi k,1}\,\hat{f}_{\xi k}(v,t)$$

$$\partial\hat{S}_\xi(0,v,t)/\partial u = \sum_k \hat{a}_{\xi k,2}\,\hat{f}_{\xi k}(v,t)$$

$$\partial^2\hat{S}_\xi(0,v,t)/\partial u^2 = \sum_k \hat{a}_{\xi k,3}\,\hat{f}_{\xi k}(v,t)$$

$$u = 1 \qquad \hat{S}_\xi(1,v,t) = \sum_k \hat{a}_{\xi k,4}\,\hat{f}_{\xi k}(v,t)$$

$$\partial\hat{S}_\xi(1,v,t)/\partial u = \sum_k \hat{a}_{\xi k,5}\,\hat{f}_{\xi k}(v,t)$$

$$\partial^2\hat{S}_\xi(1,v,t)/\partial u^2 = \textstyle\sum_k \hat{a}_{\xi k,6}\,\hat{f}_{\xi k}(v,t) \qquad (110)$$

Substituting Eq. (108) into (109) and considering the differential properties (97) and the boundary constraints (110), the following equation is obtained.

$$\left(\gamma\frac{\partial^6}{\partial u^6} + \eta\beta_{k\xi}\frac{\partial^4}{\partial u^4} + \lambda\beta_{k\xi}^2\frac{\partial^2}{\partial u^2} + \rho\beta_{k\xi}^3\right)\widehat{H}_{k\xi}(u) = 0$$

$$(\xi = x,y,z;\, k = 1,2,\cdots) \qquad (111)$$

And the above equation is subjected to the following blending boundary constraints

$$u = 0 \qquad \widehat{H}_{k\xi}(0) = \hat{a}_{k\xi,1}$$

$$\partial\widehat{H}_{k\xi}(0)/\partial u = \hat{a}_{k\xi,2}$$

$$\partial^2\widehat{H}_{k\xi}(0)/\partial u^2 = \hat{a}_{k\xi,3}$$

$$u = 1 \qquad \widehat{H}_{k\xi}(1) = \hat{a}_{k\xi,4}$$

$$\partial\widehat{H}_{k\xi}(1)/\partial u = \hat{a}_{k\xi,5}$$

$$\partial^2\widehat{H}_{k\xi}(1)/\partial u^2 = \hat{a}_{k\xi,6}$$

$$(\xi = x,y,z;\, k = 1,2,\cdots) \qquad (112)$$

The closed form solution of the sixth order ordinary differential equation (111) subjected to the boundary conditions (112) is obtainable. Due to the different combinations of γ, η, λ, ρ, and

$\beta_{\xi k}$, the closed form solution has many different forms. For the shape control parameters $\gamma = \eta = \lambda = \rho = 1$, and $\beta_{\xi k} = -4\pi^2$, the closed form solution has the form of

$$\hat{S}_\xi(u,v,t) = \sum_k \Big[\hat{b}_{k\xi,1} e^{q_0 u} + \hat{b}_{k\xi,2} e^{-q_0 u} + e^{q_1 u}$$

$$\Big(\hat{b}_{k\xi,3} cos(_2 + \hat{b}_{k\xi,4} \, sin \, q_2 \, u\Big)^{-q_1 u} \big(\hat{b}_{k\xi,5}$$

$$cos \, q_2 \, u + \hat{b}_{k\xi,6} sin \, q_2 \, u)\Big] \hat{f}_{\xi k}(v,t)$$

$$(\xi = x, y, z) \tag{113}$$

where q_0, q_1 and q_2 are determined by substituting Eq. (113) into the sixth order partial differential equation (111), and $\hat{b}_{k\xi,m}$ $(m = 1,2,\cdots,6)$ are determined by substituting Eq. (113) into the blending boundary conditions (112).

3.4.1.3 Resolution for the third group of boundary functions

Unlike the unknown function $\bar{S}_\xi(u,v,t)$ and $\hat{S}_\xi(u,v,t)$ whose exact closed form solutions are obtainable; the exact closed form solutions do not exist for the unknown functions $\tilde{S}_\xi(u,v,t)$. Here, their approximate analytical solutions are derived below.

For the third group of functions $\tilde{f}_{l\xi}(v,t)$, the corresponding mathematical expressions of the blending surface can be taken to be

$$\tilde{S}_\xi(u,v,t) = \sum_l \tilde{H}_{\xi l}(u) \tilde{f}_{\xi l}(v,t)$$

$$(\xi = x, y, z) \tag{114}$$

The partial differential equations (99) corresponding to the mathematical expressions $\tilde{S}_\xi(u,v,t)$ are

$$(\gamma \frac{\partial^6}{\partial u^6} + \eta \frac{\partial^6}{\partial u^4 \partial v^2} + \lambda \frac{\partial^6}{\partial u^2 \partial v^4} + \rho \frac{\partial^6}{\partial v^6}) \tilde{S}_\xi(u,v,t) = 0 \tag{115}$$

The blending boundary constraints (98) corresponding to the mathematical expressions $\tilde{S}_\xi(u,v,t)$ are

$$u = 0 \qquad \tilde{S}_\xi(0,v,t) = \sum_l \tilde{a}_{\xi l,1}\,\tilde{f}_{\xi l}(v,t)$$

$$\partial\tilde{S}_\xi(0,v,t)/\partial u = \sum_l \tilde{a}_{\xi l,2}\,\tilde{f}_{\xi l}(v,t)$$

$$\partial^2\tilde{S}_\xi(0,v,t)/\partial u^2 = \sum_l \tilde{a}_{\xi l,3}\,\tilde{f}_{\xi l}(v,t)$$

$$u = 1 \qquad \tilde{S}_\xi(1,v,t) = \sum_l \tilde{a}_{\xi l,4}\,\tilde{f}_{\xi l}(v,t)$$

$$\partial\tilde{S}_\xi(1,v,t)/\partial u = \sum_l \tilde{a}_{\xi l,5}\,\tilde{f}_{\xi l}(v,t)$$

$$\partial^2\tilde{S}_\xi(1,v,t)/\partial u^2 = \textstyle\sum_l \tilde{a}_{\xi l,6}\,\tilde{f}_{\xi l}(v,t) \qquad (116)$$

Substituting Eq. (114) into the partial differential equation (115), and considering the boundary conditions (116), the following equation is obtained.

$$\gamma\frac{d^6\widetilde{H}_{\xi l}(u)}{du^6}\tilde{f}_{\xi l}(v,t) + \eta\frac{d^4\widetilde{H}_{\xi l}(u)}{du^4}\frac{d^2\tilde{f}_{\xi l}(v,t)}{dv^2}$$

$$+\lambda\frac{d^2\widetilde{H}_{\xi l}(u)}{du^2}\frac{d^4\tilde{f}_{\xi l}(v,t)}{dv^4} + \rho\widetilde{H}_{\xi l}(u)\frac{d^6\tilde{f}_{\xi l}(v,t)}{dv^6} = 0$$

$$(l = 1,2,\cdots) \qquad (117)$$

subjected to the following boundary conditions

$$u = 0 \qquad \widetilde{H}_{\xi l}(0) = \tilde{a}_{\xi l,1}$$

$$\partial\widetilde{H}_{\xi l}(0)/\partial u = \tilde{a}_{\xi l,2}$$

$$\partial^2\widetilde{H}_{\xi l}(0)/\partial u^2 = \tilde{a}_{\xi l,3}$$

$$u = 1 \qquad \widetilde{H}_{\xi l}(1) = \tilde{a}_{\xi l,4}$$

$$\partial\widetilde{H}_{\xi l}(1)/\partial u = \tilde{a}_{\xi l,5}$$

$$\partial^2\widetilde{H}_{\xi l}(1)/\partial u^2 = \tilde{a}_{\xi l,6}$$

$$(\xi = x,y,z; l = 1,2,\cdots) \qquad (118)$$

Since the boundary functions $\tilde{f}_{\xi l}(v,t)$ $(l = 1,2,\cdots)$ are known, Eq. (117) is a sixth order ordinary differential equation for each of $l = 1,2,\cdots$. To solve Eq. (117) subjected to the corresponding blending boundary constraints (118), a trial function $\widetilde{H}_{\xi l}(u)$ is first constructed and make it meet the boundary conditions (118) exactly. Then, the trial function is introduced into Eq. (114) to obtain $\tilde{S}_{\xi}(u,v,t)$. The obtained $\tilde{S}_{\xi}(u,v,t)$ is substituted into Eq. (117) and the error of Eq. (117) is minimized to obtain the required solution.

Since Eq. (118) involves six blending boundary conditions, the trial function $\widetilde{H}_{\xi l}(u)$ can be taken to be a polynomial function of degree 5 plus a sine series, i. e.,

$$\widetilde{H}_{\xi l}(u) = \sum_{n=0}^{5} \tilde{b}_{\xi l,n}\, u^n + \sum_{m=1}^{M} \tilde{c}_{\xi l,m}\, sin(\, m\pi u)$$

$$(\xi = x,y,z; l = 1,2,\cdots) \tag{119}$$

where $\tilde{b}_{\xi l,n}$ and $\tilde{c}_{\xi l,m}$ are unknown constants to de determined by the boundary conditions (118) and the sixth order ordinary differential equations (117).

Substituting Eq. (119) into the boundary conditions (118), solving for $\tilde{b}_{\xi l,n}$, and inserting the obtained $\tilde{b}_{\xi l,n}$ back into Eq. (119), the following function is obtained

$$\widetilde{H}_{\xi l}(u) = \textstyle\sum_{n=1}^{6} \tilde{a}_{\xi l,n} g_n(u) - \sum_{m=1}^{M} \tilde{c}_{\xi l,m}[m\pi f_m(u) - sin\, m\, \pi u] \tag{120}$$

where $g_n(u)$ $(n = 1,2,\cdots,6)$ is determined by Eq. (107), and $f_m(u) = g_2(u) + (-1)^m g_5(u)$ $(m = 1,2,\cdots,M)$.

Substituting Eq. (120) into Eq. (114), then introducing Eq. (114) into Eq. (117), and formulating the squared error sum of Eq. (117) below

$$E_{\xi l} = \left[\phi_{\xi l}(u,v) - \sum_{m=1}^{M} \tilde{c}_{\xi l,m} \alpha_{\xi l,m}\,(u,v)\right]^2$$

$$(\xi = x,y,z; l = 1,2,\cdots) \tag{121}$$

with

$$\phi_{\xi l}(u,v)=\sum_{n=1}^{6}\tilde{a}_{\xi l,n}\Big[\gamma g_n^{(6)}(u)\tilde{f}_{\xi l}(v,t)+\eta g_n^{(4)}(u)$$

$$\tilde{f}_{\xi l}^{(2)}(v,t)+\lambda g_n^{(2)}(u)\tilde{f}_{\xi l}^{(4)}(v,t)+\rho g_n(u)\tilde{f}_{\xi l}^{(6)}(v,t)\Big] \quad (122)$$

and

$$\alpha_{\xi l,m}(u,v)=\gamma m^6\pi^6\tilde{f}_{\xi l}(v,t)\,sin(\,m\pi u)+\eta[m\pi f_m^{(4)}(u)$$

$$-m^4\pi^4 sin(\,m\pi u)]\tilde{f}_{\xi l}^{(2)}(v,t)+\lambda[m\pi f_m^{(2)}(u)+m^2\pi^2$$

$$sin(\,m\pi u)]\tilde{f}_{\xi l}^{(4)}(v,t)+\rho[m\pi f_m^{(0)}(u)-sin(\,m\pi u)]$$

$$\tilde{f}_{\xi l}^{(2)}(v,t) \quad (123)$$

where $g_l^{(k)}(u)=\frac{d^k g_l(u)}{du^k}$, $\tilde{f}_{\xi l}^{(2)}(v,t)=\frac{d^k\tilde{f}_{\xi l}(v,t)}{dv^k}$, $f_m^{(r)}(u)=\frac{d^r f_m(u)}{du^r}$, and $f_m^{(0)}(u)=f_m(u)$, $(l=1,2,\cdots;\quad k=2,4,6;\ m=1,2,\cdots M;\ r=0,2,4)$.

The above error $E_{\xi l}$ is a vector-valued continuous function. In order to quantify the error function, $(I+1)\times(J+1)$ sample points in the solution region $\{0\le u\le 1, 0\le v\le 1\}$ are uniformly allocated, which gives $\Delta u=1/I$, $\Delta v=1/J$, $u_i=i\Delta u=i/I$, and $v_i=j\Delta v=j/J$. The squared error sum of the error function $E_{\xi l}$ at these sample points can be formulated as

$$E_{\xi l}=\sum_{i=0}^{I}\sum_{j=0}^{J}\left[\phi_{\xi l}(u_i,v_j)-\sum_{m=1}^{M}\tilde{c}_{\xi l,m}\alpha_{\xi l,m}\,(u_i,v_j)\right]^2$$

$$(\xi=x,y,z;\ l=1,2,\cdots) \quad (124)$$

With the least squared method, $\partial E_{\xi l}/\partial\tilde{c}_{\xi l,m}=0$ $(l=1,2,\cdots;\ m=1,2,3,4,\cdots,M)$ are calculated, which changes Eq. (124) into the following equation

$$\sum_{m=1}^{M}\sum_{i=0}^{I}\sum_{j=0}^{J}\tilde{c}_{\xi l,m}\alpha_{\xi l,m}\,(u_i,v_j)\alpha_{\xi l,q}(u_i,v_j)$$

$$=\sum_{i=0}^{I}\sum_{j=0}^{J}\phi_{\xi l}(u_i,v_j)\alpha_{\xi l,q}\,(u_i,v_j)$$

$$(q = 1,2,3,4,\cdots,M) \tag{125}$$

There are M linear algebra equations in (125) which can be used to determine M unknown constants $\tilde{c}_{\xi l,m}$ $(m = 1,2,3,4,\cdots,M)$.

Repeating the solution process for $l = 1,2,\cdots$, all the unknown constants $\tilde{c}_{\xi l,m}$ $(l = 1,2,\cdots;\ m = 1,2,\cdots,M)$ are obtained. Substituting them back into Eq. (120), and introducing Eq. (120) into Eq. (114), the mathematical expressions are obtained as

$$\tilde{S}_{\xi}(u,v,t) = \sum_{l}\sum_{n=1}^{6} \tilde{a}_{\xi l,n} g_n(u) - \sum_{m=1}^{M} \tilde{c}_{\xi l,m}$$

$$[m\pi f_m(u) - sin\, m\,\pi u]\tilde{f}_{\xi l}(v,t)$$

$$(\xi = x,y,z) \tag{126}$$

Putting the obtained $\bar{S}_{\xi}(u,v,t)$, $\hat{S}_{\xi}(u,v,t)$, and $\tilde{S}_{\xi}(u,v,t)$ together, the mathematical expression of blending surfaces is obtained. The applications of the proposed approach in time-dependent surface blending will be demonstrated in subsection 3.4.4.1 and in time-independent surface blending will be demonstrated in subsection 3.4.4.2.

3.4.2 Results and discussion

In this subsection, the proposed approach is first implemented and compared with the exact closed form solution to demonstrate its good accuracy and high efficiency. Next, it is compared with the time-dependent C^1 continuous surface blending in (You et al., 2012) to investigate the differences between C^1 and C^2 continuities. Finally, the influence of second partial derivatives on the continuity at timelines is examined.

The obtained mathematical expressions (106), (113) and (126) of blending surfaces were implemented with C++ and OpenGL. All the examples were running on a same desktop with 3.5GHz CPU.

3.4.2.1 Accuracy and efficiency

First, the accuracy and efficiency of the proposed approach are investigated by comparing it with the accurate closed form solution through creating a time-dependent blending surface between two separate elliptic cylinders represented with time-dependent primary surfaces. The parametric representation for the first (top) primary surface is

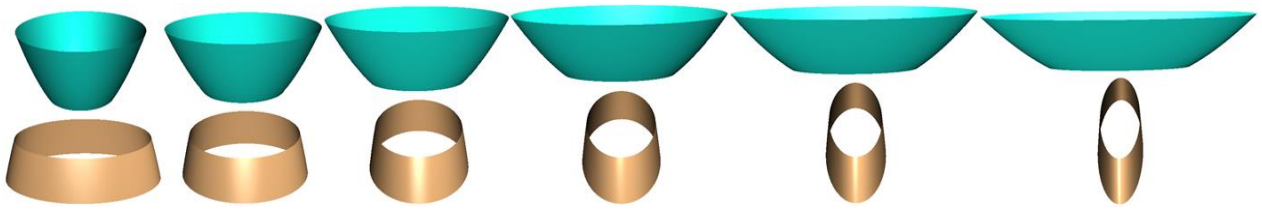
$$x = aue^{t}sin2\pi v \quad y = bue^{-t}cos2\pi v \quad z = h_1 + h_2u^2 \tag{127}$$

And the parametric representation for the second (bottom) primary surface is

$$x = cue^{-t}sin2\pi v \quad y = due^{t}cos2\pi v \quad z = -h_3u^3 \tag{128}$$

where $a, b, c, d, h_1, h_2\ and\ h_3$ are the geometric parameters to be specified, and $0 \le u \le 1$.

Setting the geometric parameters in Eqs. (127) and (128) to be: $a = 3.6$, $b = 5.5, c = 6.0, d = 3.0, h_1 = 2.0, h_2 = 6.0$ and $h_3 = 20.0$, the first primary surface between $u = 0.4$ and $u = 0.75$ and the second primary surface between $u = 0.55$ and $u = 0.65$ at the time instants $t = 0, 0.2, 0.4, 0.6, 0.8, \text{and } 1.0$ are depicted in the Figure 14 where the top and bottom primary surfaces are obtained from Eq. (127) and Eq. (128), respectively.



$t = 0$ $t = 0.2$ $t = 0.4$ $t = 0.6$ $t = 0.8$ $t = 1$

Figure 14: Primary surfaces in cyan and brown at t=0,0.2,0.4,0.6,0.8,and 1

If the trimline is taken to be $u = u_0$ in Eq. (127) for the top primary surface and $u = u_1$ in Eq (128) for the bottom primary surfaces, the boundary curves $C_1(v,t) = [aue^{t}sin2\pi v,\ \ bue^{-t}cos2\pi v,\ h_1 + h_2u_0^2]^T$ and $C_4(v,t) = [cue^{-t}sin2\pi v,\ due^{t}cos2\pi v,\ -h_3u_0^2]^T$ are obtained. With Eqs. (127) and (128), the first partial derivatives

$\partial x/\partial u\,, \partial y/\partial u,\ and\ \partial z/\partial u,$ and the second partial derivatives $\partial^2 x/\partial u^2\,, \partial^2 y/\partial u^2\,, and\ \partial^2 z/\partial u^2$ are derived. Setting $u = u_0$ for the first and second partial derivative from Eq. (127), $C_2(v,t) = [-aue^t sin2\pi v,\ -bue^{-t} cos2\pi v,\ -h_2 u_0]^T$ and $C_3(v,t) = [0\ \ 0\ 2h_2]^T$ are obtained. Setting $u = u_1$ for the first and second partial derivative from Eq. (128), $C_5(v,t) = [ce^{-t} sin2\pi v,\ \ de^t cos2\pi v, -3h_3 u_1^2]^T$ and $C_6(v,t) = [0\ \ 0\ \ -6h_3 u_1]^T$ are obtained. Substituting $C_i(v,t)$ $(i = 1,2,\dots,6)$ into (1), the blending boundary constraints (1) become

$$
\begin{aligned}
&u = 0\\
&S_x = au_0 e^t sin2\pi v \quad S_y = bu_0 e^{-t} cos2\pi v \quad S_z = h_1 + h_2 u_0^2\\
&\frac{\partial S_x}{\partial u} = -ae^t sin2\pi v \quad \frac{\partial S_y}{\partial u} = -be^{-t} cos2\pi v \qquad \frac{\partial S_z}{\partial u} = -2h_2 u_0\\
&\frac{\partial^2 S_x}{\partial u^2} = 0 \qquad\qquad \frac{\partial^2 S_y}{\partial u^2} = 0 \qquad\qquad \frac{\partial^2 S_z}{\partial u^2} = 2h_2\\
&u = 1\\
&S_x = cu_1 e^{-t} sin2\pi v \quad S_y = du_1 e^t cos2\pi v \quad S_z = -h_3 u_1^3\\
&\frac{\partial S_x}{\partial u} = ce^{-t} sin2\pi v \quad \frac{\partial S_y}{\partial u} = de^t cos2\pi v \qquad \frac{\partial S_z}{\partial u} = -3h_3 u_1^2\\
&\frac{\partial^2 S_x}{\partial u^2} = 0 \qquad\qquad \frac{\partial^2 S_y}{\partial u^2} = 0 \qquad\qquad \frac{\partial^2 S_z}{\partial u^2} = -6h_3 u_1
\end{aligned}
\tag{129}
$$

where the minus symbol "−" is due to the opposite directions between the blending surface and the top primary surface.

The functions in the blending boundary constraints (129) can be divided into two types, i. e., $\bar{S}_\xi(u,v,t)$ and $\hat{S}_\xi(u,v,t)$. To investigate the accuracy and efficiency, both the exact closed form solutions (113) and the approximate analytical solutions (126) are used to obtain the unknown functions $\hat{S}_\xi(u,v,t)$. Using the same geometric parameters $a = 3.6,\ b = 5.5,\ c = 6.0,\ d = 3.0,$ $h_1 = 2.0,\ h_2 = 6.0,$ and $h_3 = 20.0$, and setting the shape control parameters to $\gamma = \eta = \lambda = \rho = 1$, the time variable to $t = 0.1$, and M in Eq. (126) to 10, 15, and 20, the blending surfaces

$S_M(u, v, t = 0.1)$ obtained from approximate analytical solution are depicted in Figure 15 where M indicates the total terms used in Eq. (126).

Using the same geometric and shape control parameters, the blending surface $S_{CFS}(u, v, t = 0.1)$ obtained from the exact closed form solution (113) is also shown in Figure 15 where CFS indicates the closed form solution.

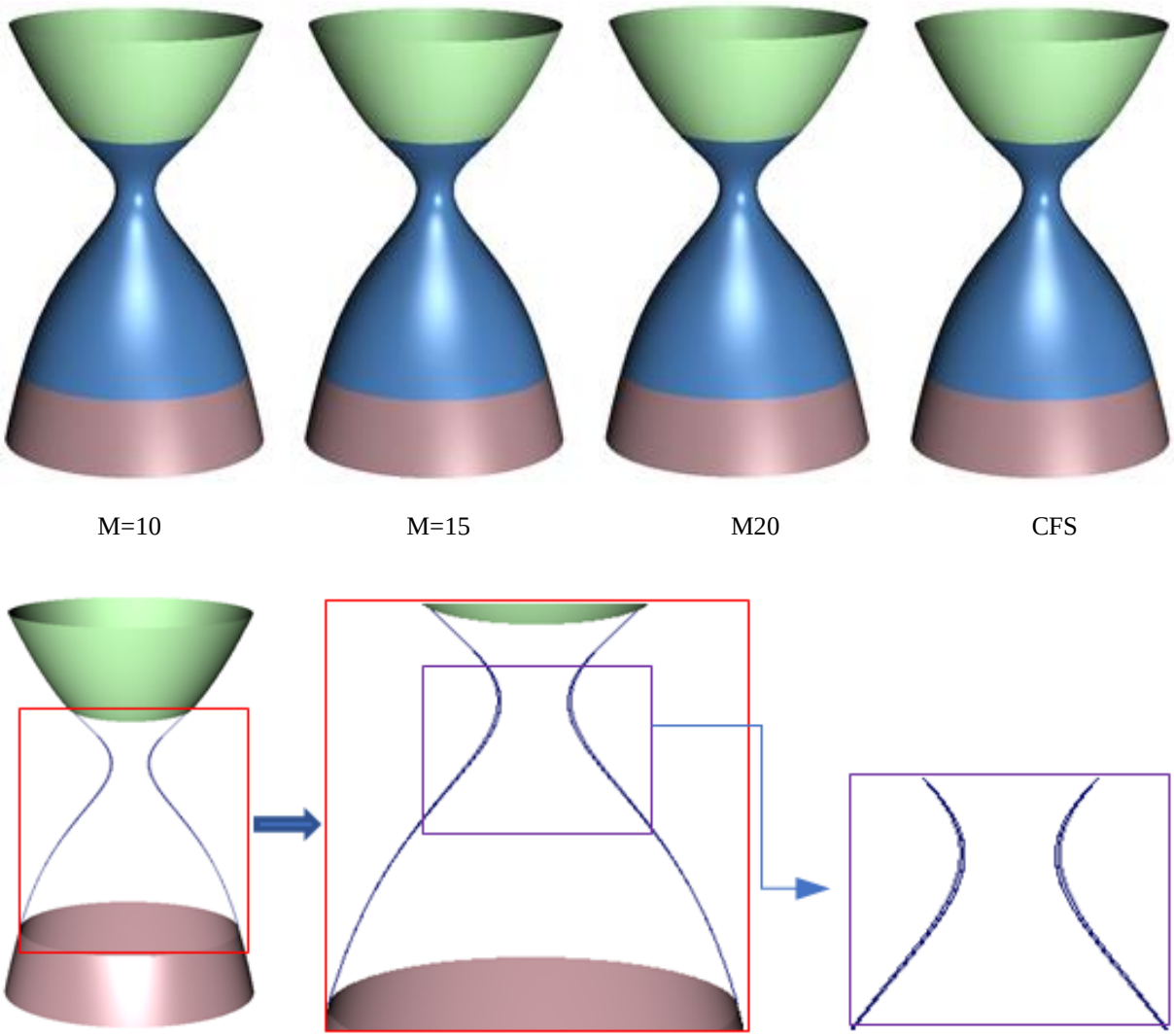


Figure 15: Blending surfaces generated by three different M values and the closed form solution (CFS)

The first image of the second row of the figure shows the profile curves of the blending surface obtained from M=10, 15, and 20 of the proposed approach and the exact closed form solution, and

the profile curves are magnified in the second and third images of the second row with the two innermost profile curves in the last image of the second row from M=10 and the outer profile curves in the last image of the second row from $M = 15$, $M = 20$ and the closed form solution. The second row of Figure 15 shows no visible difference between profile curves from M=15 and M=20 of the proposed approach and the closed form solution.

Next, the error calculations given in Subsection 3.3.2.1 are used to determine the absolute maximum error (E_1), absolute average error (E_2) relative maximum error (E_3), and relative average error(E_4), respectively, and the obtained errors are given in Table 3. The computational time (CPU) used to determine all the unknown constants and generate the blending surfaces with M = 10, 15 and 20 and four different quad meshes is also given in the same table where T1, T2, T3 and T4 stand for the computational time for the quad meshes with $51 \times 51, 101 \times 101, 151 \times 151, and\ 201 \times 201$ vertices, respectively.

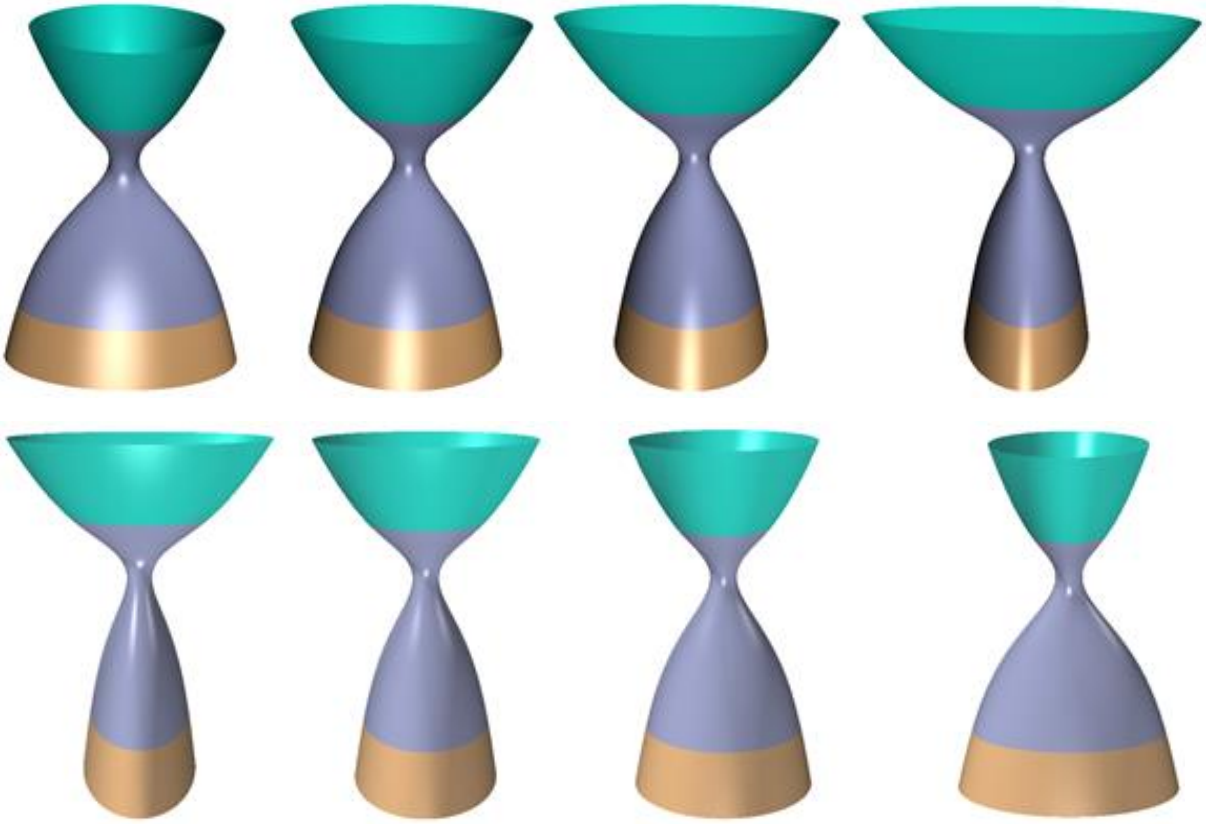
Table 3. Accuracy and efficiency of the proposed approach

M	10	15	20	CFS
E_1	4.35×10^{-2}	3.17×10^{-3}	2.22×10^{-5}	0
E_2	1.82×10^{-2}	1.31×10^{-3}	1.10×10^{-5}	0
E_3	1.11×10^{-2}	8.06×10^{-4}	5.64×10^{-6}	0
E_4	4.63×10^{-3}	3.34×10^{-4}	2.80×10^{-6}	0
T(ms)(51×51)	35	60	98	47
T(ms)(101×101)	69	112	197	109
T(ms)(151×151)	118	184	288	187
T(ms)(201×201)	183	269	394	339

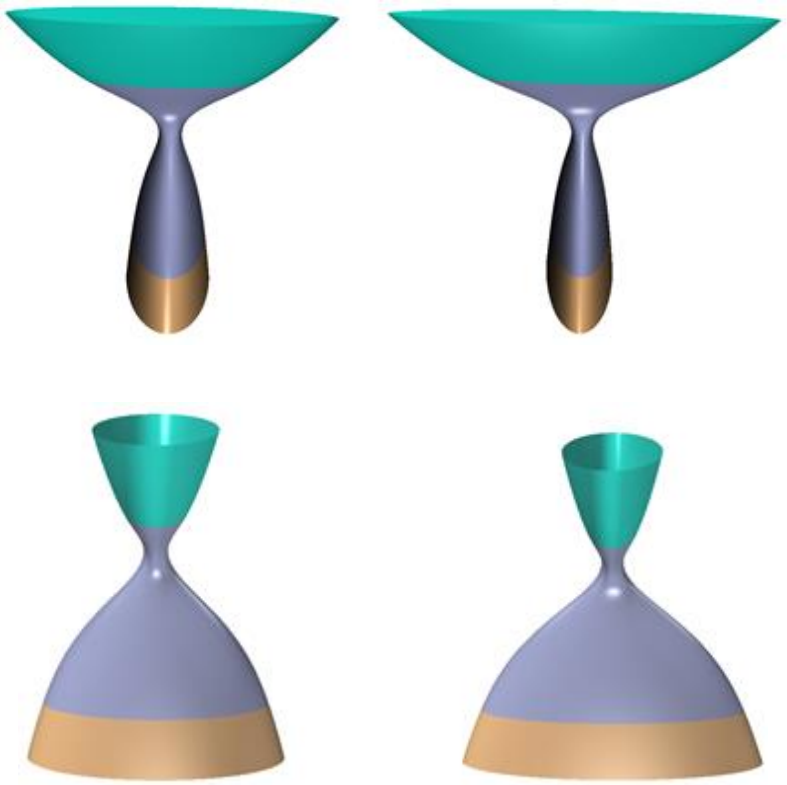
The data in Table 3 demonstrate good accuracy and high computational efficiency of the proposed approximate analytical solution. When M increases from 10 to 20, the relative average error (E_4) of the proposed approach decreases from 4.63×10^{-3} to 2.80×10^{-6} in comparison

with the closed form solution, which is the most accurate. With the increase of the total vertices, the computational time for both approximate analytical solution and exact closed form solution rises. When M=10, the approximate analytical solution is more efficient than the exact closed form solution for all the four meshes. When M=20, the computational time of the approximate analytical solution becomes larger than the exact closed form solution but still at the same order.

With the developed approach and setting the time variable t = 0, 0.2, 0.4, 0.6, 0.8 and 1, the blending surfaces at these time instants are obtained and depicted in Figure 16 where the first and third rows are from the front view and the second and fourth rows are from the side view.



$t = 0.0$ $t = 0.2$ $t = 0.4$ $t = 0.6$



$t = 0.8$ $t = 1.0$

Figure 16: Blending surfaces at different time instants

The images shown in Figure 16 indicate that at different time instants, the proposed approach always create C^2 continuous blending surfaces to smoothly connect time-dependent primary surface together.

3.4.2.2 Comparison with time-dependent C^1 continuous surface blending

In this subsection, the C^2 continuous surface blending developed in this thesis is compared with the C^1 continuous surface blending introduced in (You et al., 2012) through a blending example. It creates a smooth transition between two time-dependent cylinders.

The parametric equations for the first cylinder are

$$x = ae^t sin2\pi v \quad y = ae^{-t} cos2\pi v \quad z = h_1 + h_2 u^2 \tag{130}$$

The parametric equations for the second cylinder are

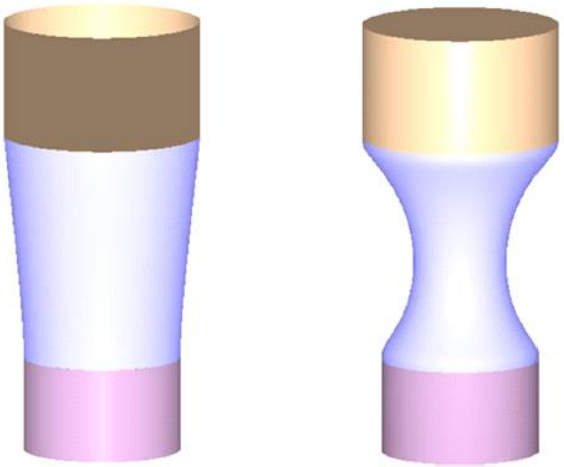
$$x = be^{-t}sin2\pi v \quad y = be^{t}cos2\pi v \quad z = -h_3u^3 \tag{131}$$

In the above Eqs. (130) and (131), the geometric parameters are taken to be $a = 1.0, b = 0.8, h_1 = 2.0, h_2 = 3.0\ and\ h_3 = 5.0$. The trimlines are at $u_0 = 0.2\ and\ u_1 = 0.3$ where u_0 and u_1 stand for the isoparametric lines of the first and second cylinders, respectively.

The position functions and the first partial derivatives at the timelines required by the blending boundary constraints described in (You et al., 2012) can be derived from Eqs. (130) and (131). All the shape control parameters are set to 1, the total terms are k = 10. The C^1 continuous blending surface at the time instant t = 0 created by the approach proposed in (You et al., 2012) is depicted in Figure 17(a), and the computational time (CPU) is 145 milliseconds.

With the approach proposed in this thesis, the second partial derivatives at the trimlines are derived from Eqs. (130) and (131) and added to the blending boundary constraints. All the shape control parameters are also set to 1, the total terms are M=10, and all the geometric parameters are kept unchanged. The C^2 continuous blending surface at the time instant $t = 0$ created by the approach proposed in this thesis is shown in Figure 17(b), and the computational time (CPU) is 178 milliseconds.

Since the two approaches uses different partial differential equations (fourth-order vs sixth order) and different blending boundary constraints (without and with the second partial derivatives), the shapes of the blending surfaces generated with the two different approaches are different and their shapes cannot be compared. Therefore, only the computational efficiency and how the second partial derivatives affect curvature continuity are compared.



(a) C^1 continuity (b) C^2 continuity

Figure 17: Comparison between C^1 and C^2 continuous surface blending approaches

Although the approach proposed in this thesis uses sixth-order partial differential equations and more blending boundary constraints, i.e., the second partial derivatives, the computational time for the two approaches is at the same order. Without the constraint of the second partial derivatives, the curvature continuity at the trimlines cannot be maintained as shown in Figure 17(a). After applying the constraint of the second partial derivatives, good curvature continuity at the trimlines is achieved as shown in Figure 17(b).

Good curvature continuity at the trimlines can also be demonstrated by the same curvature values on trimlines. Using the formulae given in the links[3,4], the maximum curvature and minimum curvature obtained from C^1 and C^2 continuous surface blending approaches are given in Table 4.

For the primary and blending surfaces shown in Figure 17, the maximum curvature is in the horizontal plane in which the cross-section curves are circles, and the minimum curvature is in the vertical direction. Since the upper and bottom primary surfaces are cylindrical surfaces, the minimum curvature are zero. Table 4 shows both maximum and minimum curvature values on the

[3] https://math.stackexchange.com/questions/3639320/find-out-the-maximum-principal-curvature-of-parametric-surface-pu-v
[4] https://en.wikipedia.org/wiki/Parametric_surface

upper and bottom trimlines $u = 0$ and $u = 1$ obtained from the proposed C^2 continuous approach are the same as those of primary surfaces . However, the C^1 continuous approach only keeps the same maximum curvature values, i. e., the same circles in the horizontal cross-sections at the positions of the trimlines, but has totally different minimum curvature values in the vertical direction as shown by 0 and 0.9811757 on the upper trimline $u = 0$ and 0 and -0.631916 on the bottom trimline $u = 1$, indicating that the C^1 continuous approach cannot achieve curvature continuities in the directions crossing the trimlines.

Table 4. Maximum and minimum curvature on trimlines

Trimline		C^2 continuous approach		C^1 continuous approach	
$u = 0$	Top surface	1.0	0.0	1.0	0.0
	Blending surface	1.0	0.0	1.0	0.9811757
$u = 1$	Blending surface	1.25	0.0	1.25	-0.631916
	Bottom surface	1.25	0.0	1.25	0.0

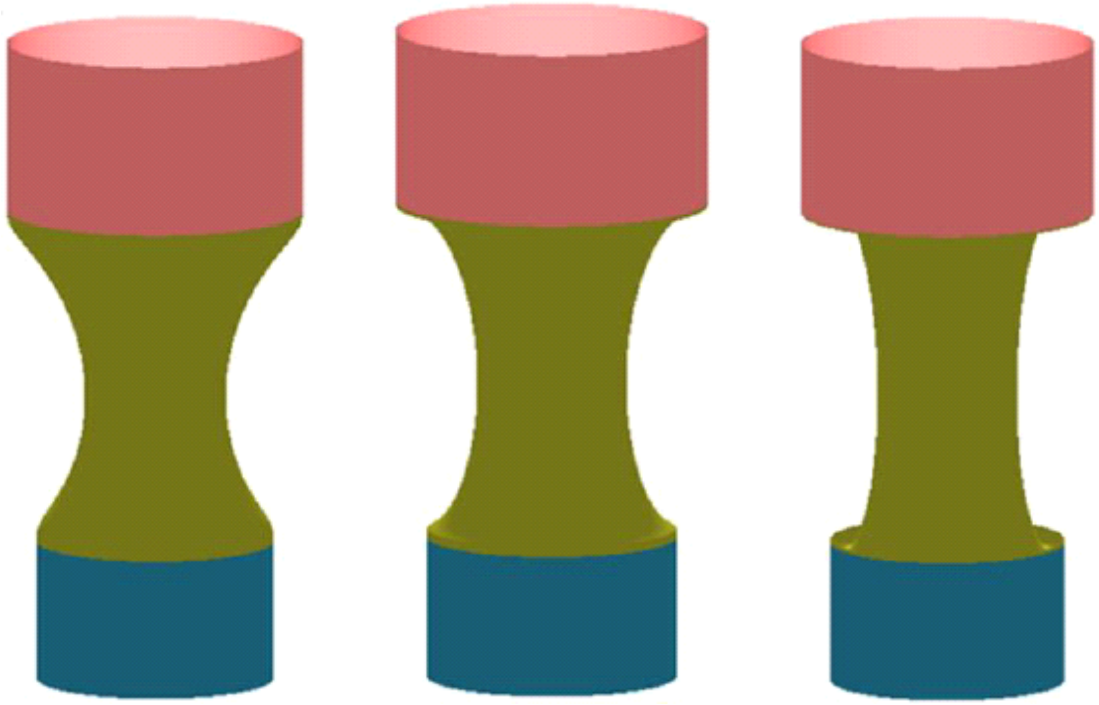
The blending surface shown in Figure 17(b) is very concave. It indicates that for the surface blending between the two cylinders defined by Eqs. (130) and (131), when the area of the blending surface is to be minimized or the radius of curvature is to be maximized, C^2 continuous surface blending may not be as good as C^1 continuous surface blending. However, this problem can be effectively solved by optimizing shape control parameters whose effects on blending surfaces are discussed in the following subsection. With this treatment, the surface area can be minimized, and the radius of curvature can be maximized while the exact C^2 continuity is still satisfied.

3.4.2.3 Effects of second partial derivatives

Unlike the C^1 continuous surface blending presented in (You et al., 2012) which only maintains the continuities of the position functions and first partial derivatives at trimlines, the C^2 continuous

surface blending developed in this subsection introduces second partial derivatives at trimlines to achieve higher order continuity. In this subsection, how second partial derivatives at trimlines affect the continuity between the blending surface and primary surfaces will be investigated.

For the surface blending problem discussed in the above subsection, three different cases of the second partial derivatives for z component are considered. For all three cases, the z component of the primary surfaces and the blending surface always has the same position functions and first derivatives at the trimlines. For the first case, the blending surface and primary surfaces have the same second partial derivatives $-2h_2$ at the trimline u = 0 and $-6h_3u_1$ at the trimline u =1, and the obtained blending surface is shown in Figure 18(a). For the second case, the blending surface increases its second partial derivative to $-10h_2$ at the trimline u = 0 and $-18h_3u_1$ at the trimline u =1, and the obtained blending surface is indicated in Figure 18(b). For the third case, the blending surface further raises its second partial derivative to $-20h_2$ at the trimline u = 0 and $-36h_3u_1$ at the trimline u =1, and the generated blending surface is depicted in Figure 18(c). In the figure, the images in the bottom row show different shapes of the blending surface only. They are used to demonstrate how the second partial derivatives affect the continuity at the trimlines.



(a) (b) (c)

Figure 18: effects of second partial derivatives

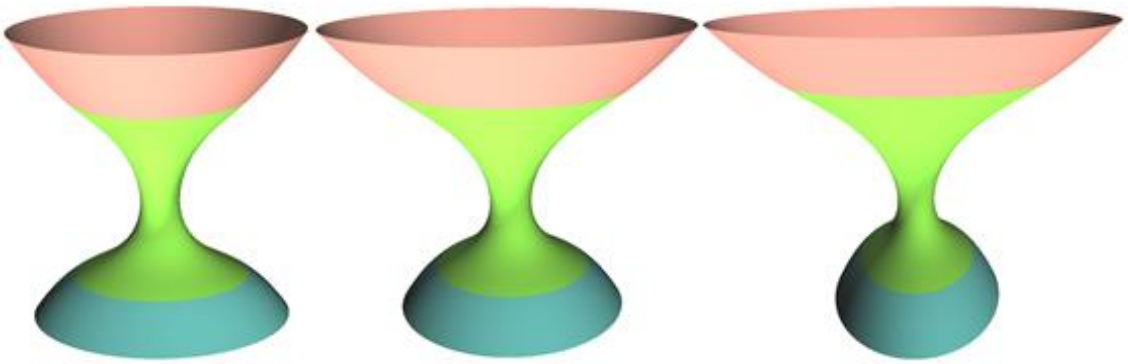
It can be seen from the images that when the blending surface and primary surfaces have the same second partial derivatives at the trimlines, good continuity between the blending surface and the primary surfaces is obtained as shown in Figure 18(a). When the second partial derivatives of the blending surface and primary surfaces at trimlines are different, poor continuity from the blending surface to primary surfaces occurs as indicted in Figure 18(b). If the difference of the second partial derivatives between the blending surface and primary surfaces is bigger, the continuity between the blending surface and primary surfaces becomes worse as shown in Figure 18(c).

3.4.3 Shape control of blending surfaces

One of the main advantages of the proposed approach is it provides useful shape control to obtain different shapes of blending surfaces in comparison with existing surface blending methods. If users do not want to use the shape control to manipulate blending surfaces, they can simply set all the shape control parameters to 1, i.e., $\gamma = \eta = \lambda = \rho = 1$. In this subsection, how to use different shape control parameters to create different shapes of blending surfaces will be investigated.

In order to generate different shapes, the geometric parameters in Eqs. (127), (128) and (129) are changed to: $a = 2.6, b = 4.5, c = h_3 = 5.0, d = h_1 = 2.0,$ and $h_2 = 3.0$, and the trimlines are

changed to $u_0 = 0.5$ and $u_1 = 0.4$. If it is not required to use the shape control parameters to manipulate blending surfaces, they can be set to $\gamma = \rho = \lambda = \eta = 1$. The obtained blending surfaces at the time instants $t = 0.2, 0.4$ and 0.6 are shown in Figure 19.



$t = 0.2$ $t = 0.4$ $t = 0.6$

Figure 19: Blending surfaces with γ=η=λ=ρ=1 at different time instants

If it is required to use the shape control parameters to create different shapes and select the desirable ones from them, the shape control parameters to different values. In what follows, the effects of the shape control parameters $\gamma, \eta, \lambda \; and \; \rho$ and on blending surfaces will be investigated.

3.4.3.1 Effects of Shape Control Parameter γ

In order to investigate the effect of the shape control parameter γ, $\eta = \lambda = \rho = 1$ are kept unchanged, the time t is taken to be 0 and γ is set to different values shown in Figure 20. Different shapes of the blending surface are obtained and depicted in the same figure where the last image shows the profile curves of different shapes of a same blending surface.

Examining the shapes of the blending surface in Figure 20, the following conclusions can be drawn. (1) When the shape control parameter γ changes from -10 to -2.6, the concave blending surface becomes straight and bigger. (2) With further changes from -2.6 to -2, the middle part of the blending surface becomes more and more convex.

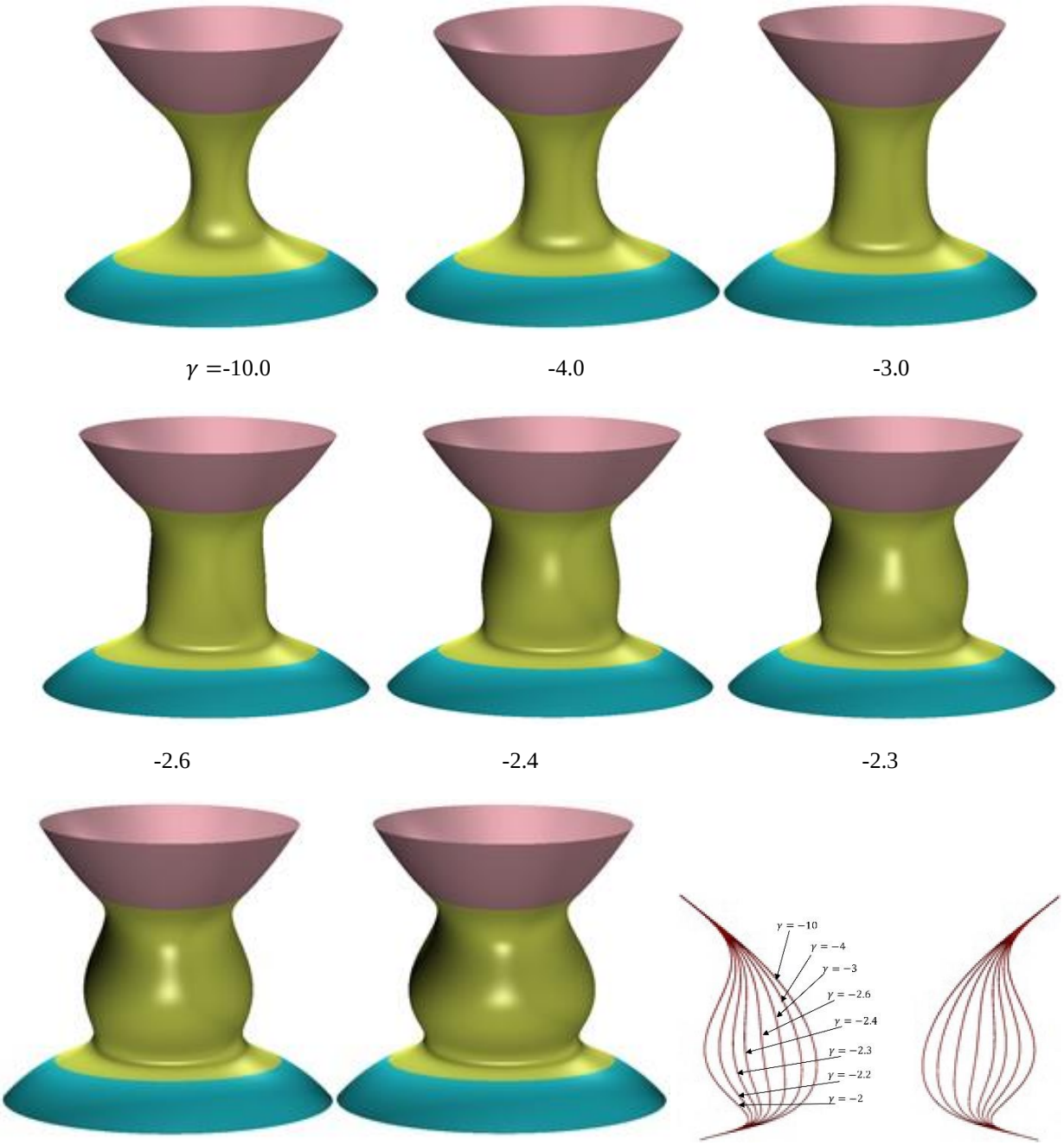
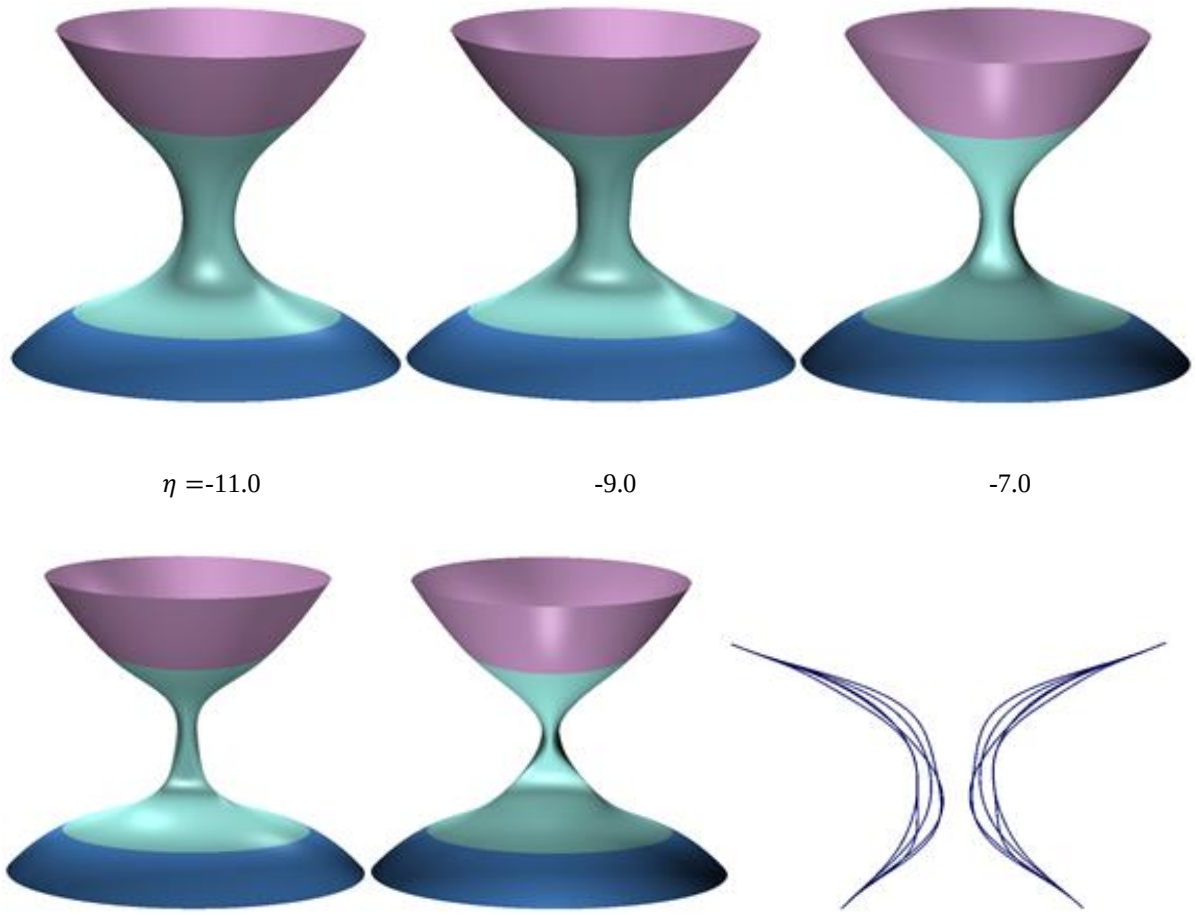


Figure 20: Effect of the shape control parameter γ on the blending surface with η=λ=ρ=1, and t=0

3.4.3.2 Effects of Shape Control Parameter η

Secondly, the shape control parameters $\gamma = \lambda = \rho = 1$ are fixed, and η is set to different values shown in Figure 21. The obtained shapes of the blending surface and their profile curves are shown in the same figure.

The blending surfaces in Figure 21 indicate that when η changes from -11.0 to -7.0, the middle part of the blending surface first becomes straight, then becomes more concave and smaller. With further changes from -7.0 to -4.5, the middle part of the blending surface is changed into a shape like the frustum of a cone first, and finally becomes most concave at $\eta = -4.5$.



η =-11.0 -9.0 -7.0

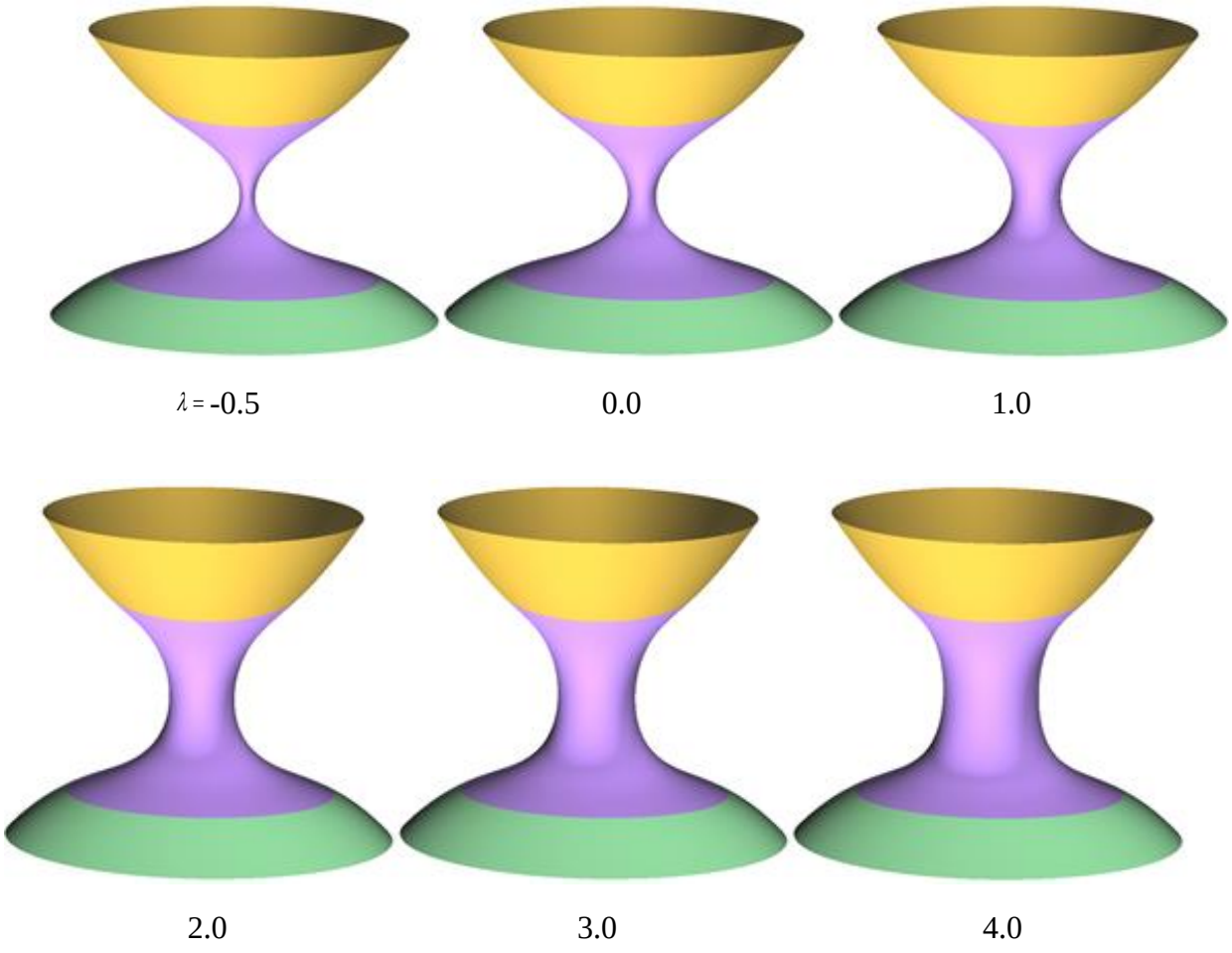
-6.0 -4.5 Profile curves

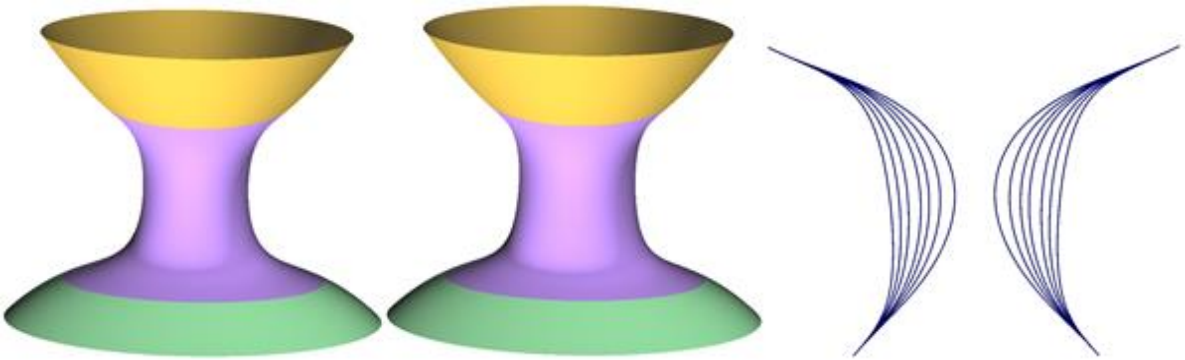
Figure 21: Effect of the shape control parameter η on the blending surface with γ=λ=ρ=1, and t=0

3.4.3.3 Effects of Shape Control Parameter $\boldsymbol{\lambda}$

Thirdly, the shape control parameters $\gamma = \eta = \rho = 1$ are kept unchanged and the shape control parameter λ is set to different values shown in Figure 22. The obtained shapes of the blending surface and their profile curves are also given in the same figure.

From Figure 22, it is found that the middle part of the blending surface is most concave at $\lambda = -0.5$ among the $\boldsymbol{\lambda}$ values between -0.5 and 7.0. When $\boldsymbol{\lambda}$ changes from -0.5 to 7.0, the blending surface becomes less concave until it becomes a cylinder-like shape at $\lambda = 7.0$.





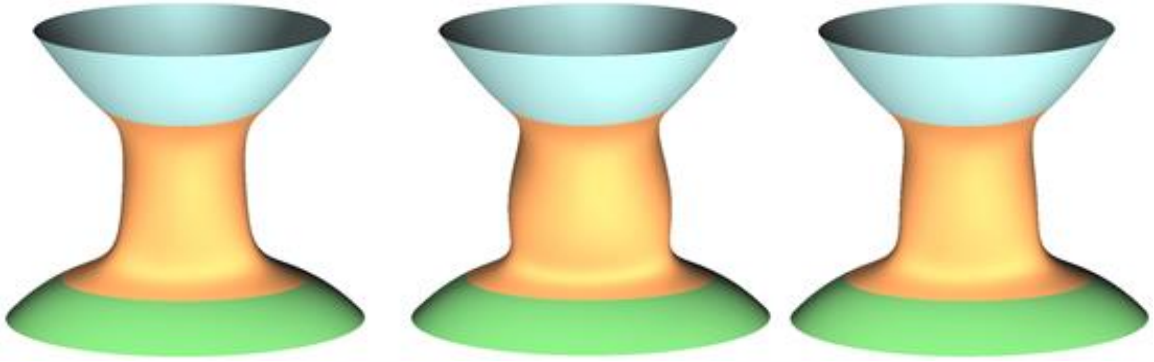
5.0 7.0 Profile curves

Figure 22: Effect of the shape control parameter λ on the blending surface with γ=η=ρ=1, and t=0

3.4.3.4 Effects of Shape Control Parameter ρ

Finally, how the shape control parameter ρ affects the shape of the blending surface is investigated. The shape control parameters $\gamma = \eta = \lambda = 1$ are kept unchanged, and the shape control parameter ρ is set to different values shown in Figure 23. The generated shapes of the blending surface and their profile curves are also depicted in the same figure.

After testing different values of the shape control parameter ρ, the range $-1.5 \leq \rho \leq 3.0$ was chosen to demonstrate the effects of this shape control parameter. The blending surfaces shown in Figure 23 indicate that when the shape control parameter ρ changes from -1.5 to -0.7, the straight middle part of the blending surface first becomes convex, and then changes back to the frustum of a cone but with a bigger cross-section size. When the shape control parameter ρ changes from -0.7 to 3.0, the blending surface becomes more and more concave and reaches most concave at $\rho = 3.0$.



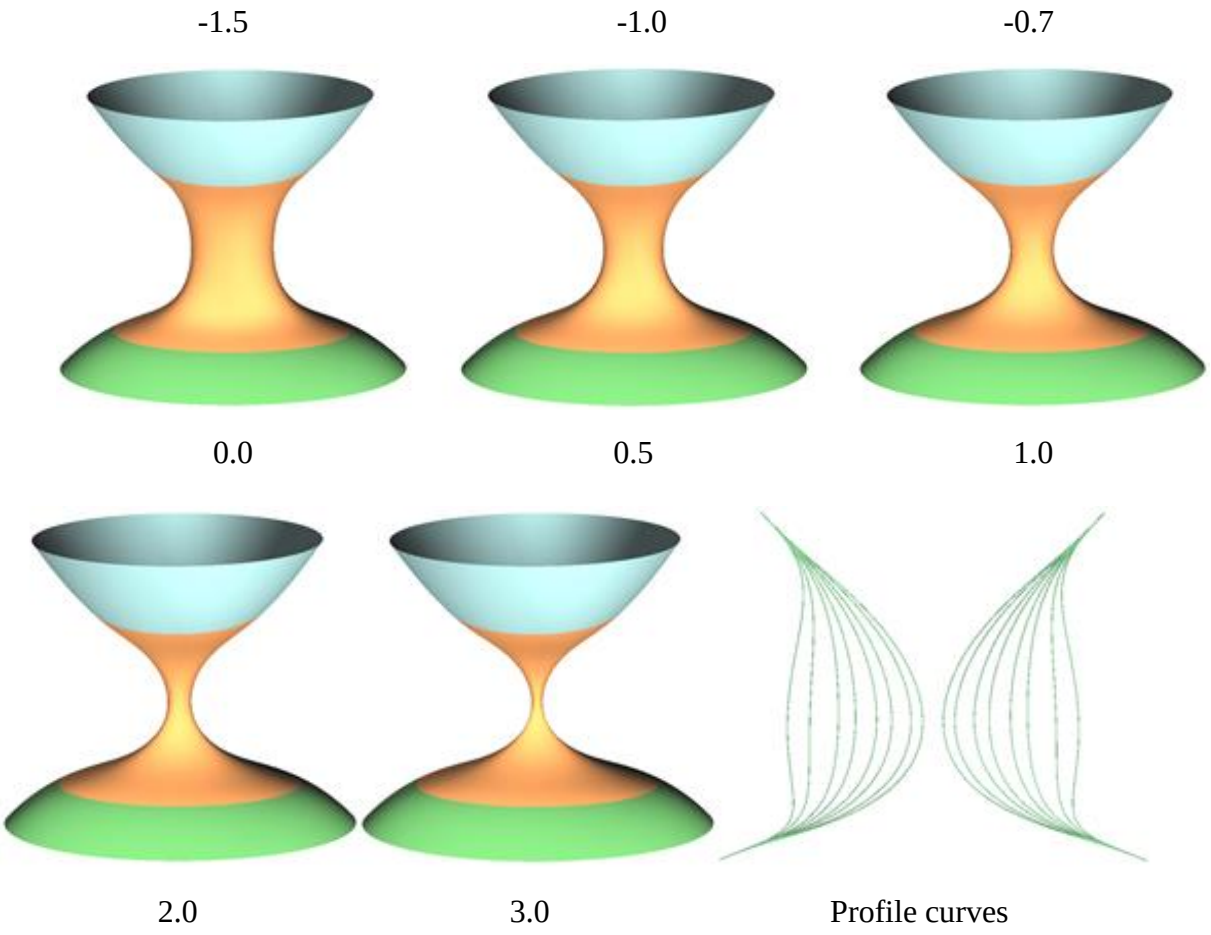


Figure 23: Effect of the shape control parameter ρ on the blending surface γ=η=λ=1 and t=0

As discussed in Section 1.1, how to easily and quickly achieve a satisfactory shape of blending surfaces while keeping the same continuity on the trimlines is an unsolved problem. The above discussions indicate that all the four shape control parameters have a great influence on the shape of the blending surface. They can be developed into useful user handles to control the shape of the blending surface.

3.4.4 Applications of surface blending

In this section, the developed approximate analytical solution will be used to create two time-dependent blending surfaces of C^2 continuity. Then, the developed approximate analytical solution

will be employed to create some time-independent blending surfaces frequently met in engineering applications.

3.4.4.1 Time-dependent surface blending

First, the surface blending between linearly varying primary surfaces will be investigated. Then, a blending surface between non-linearly varying primary surfaces will be created.

3.4.4.1.1 Surface blending between linearly varying primary surfaces

For the first application example, two primary surfaces change their shapes linearly. The first primary surface varies from an open surface to a closed one, and the second primary surface changes from a plane to a cone shaped frustum.

The parametric equations of the first primary surfaces are constructed as

$$x = (1-t)[b_1 \sinh(a_1 v + a_2) + b_2(1+u^3)\sin(a_3 v)] + b_3 t\cos(-v)$$

$$y = (1-t)[b_4 \cosh(a_4 v) + b_2(1+u^3)\cos(a_3 v)] + b_5 t\sin(-v)$$

$$z = (1-t)(h_0 + b_6 + e^u) + (h_1 + h_2 u^2)t \tag{132}$$

The parametric equations for the second primary surface are constructed as

$$x = (0.8+u)(1-t)b_6 \sin(a_5 v) + b_7 t\cos(-v)$$

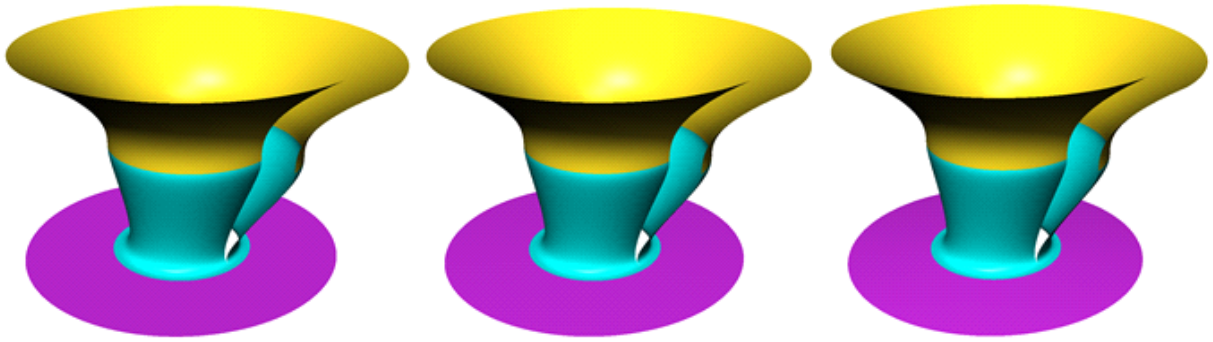
$$y = 0.8(1-t)b_6 \cos(a_5 v) + b_8 t\sin(-v)$$

$$z = (1-t)h_3 + [h_4 + h_5 \sinh(-u)]t \tag{133}$$

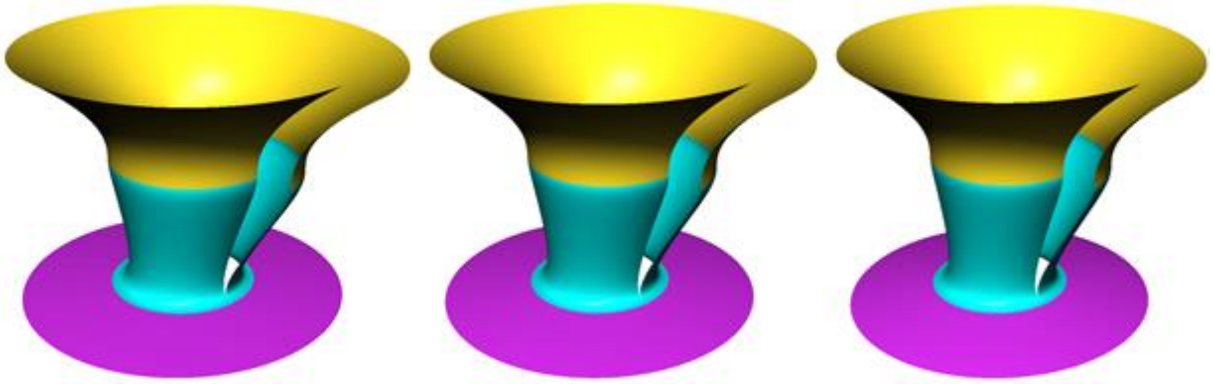
The geometric parameters in the above Eqs. (132) and (133) are taken to be: $a_1 = a_2 = b_1 = 0.1, a_3 = a_5 = b_6 = h_2 = h_5 = 0.1, a_4 = b_4 = 0.3, b_2 = 1.5$, $b_3 = 1.2, b_5 = 0.8, b_6 = 1.6, b_7 = 0.5$, $b_8 = 0.7, h_0 = 0.75, h_1 = 1.7, and\ h_3 = h_4 = -0.5$. The trimlines are taken to be at the isoparametric lines $u_0 = u_1 = 0$ where $u_0\ and\ u_1$ stand for the isoparametric lines of the first and second primary surfaces, respectively.

From the parametric equations of the two primary surfaces, the position functions, and the first and second partial derivatives of the first surface at $u_0 = 0$ and the second surface at $u_1 = 0$ can be

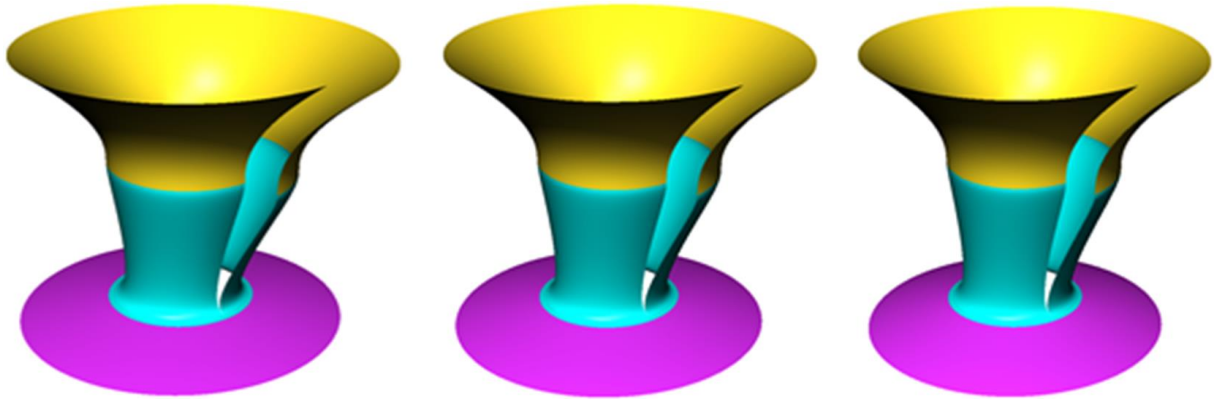
obtained. They are taken to be the boundary constraints of the blending surface at its isoparametric lines u = 0 and u =1, respectively. With the developed approach, the C^2 continuous blending surface is generated whose shapes at the time instants $t = i/25$ (i = 0, 1, 2, ⋯, 24, 25) are depicted in Figure 24.



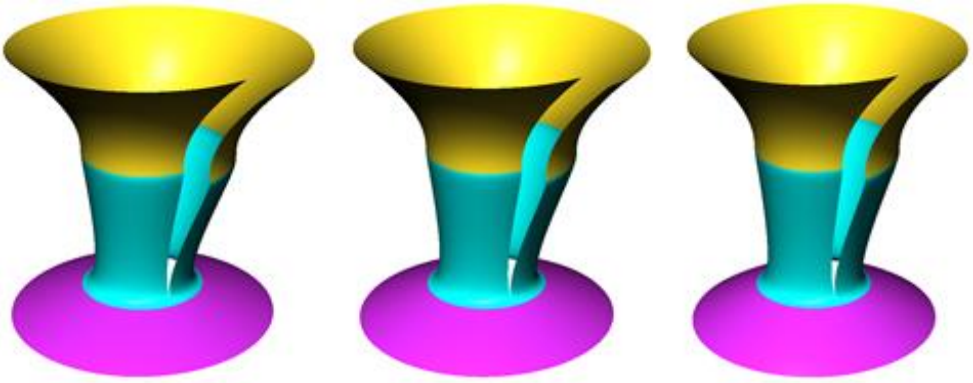
t=0 *t*=0.04 *t*=0.08



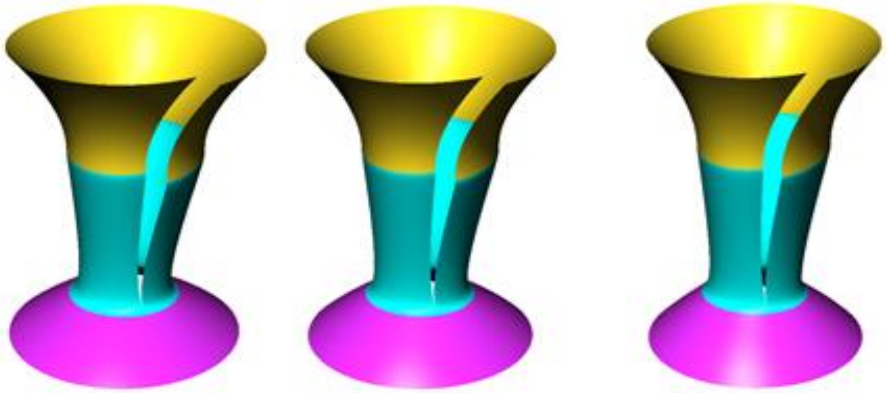
t=0.12 *t*=0.16 *t*=0.20



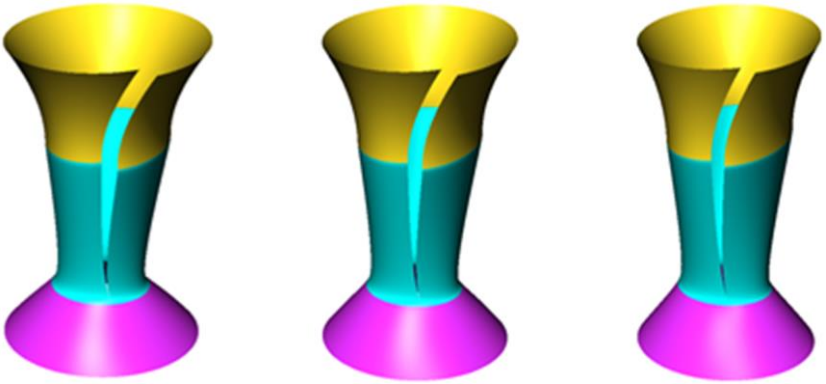
t=0.24 *t*=0.28 *t*=0.32



t=0.36 *t*=0.40 *t*=0.44



t=0.48 *t*=0.52 *t*=0.56



t=0.60 *t*=0.64 *t*=0.68

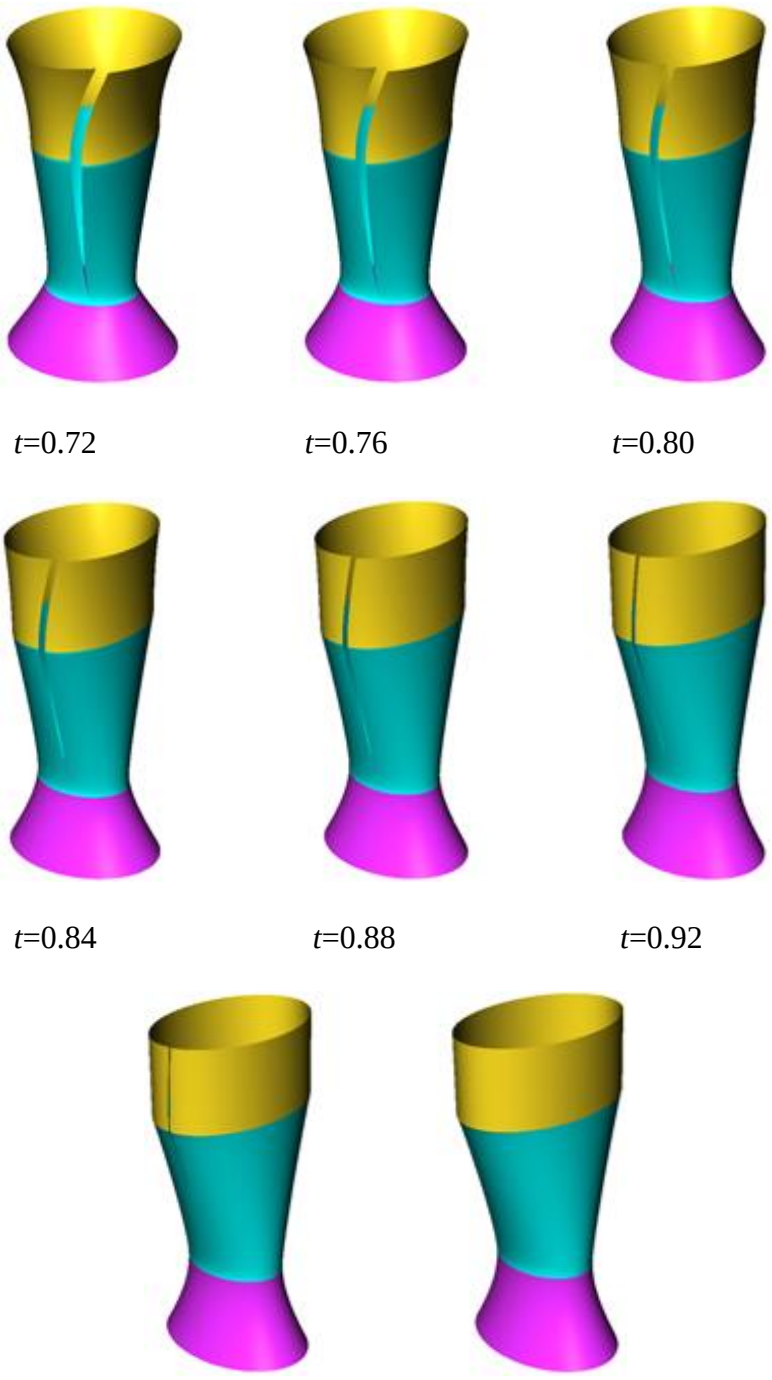


Figure 24: Surface blending between linearly varying primary surfaces

This example indicates that although the two primary surfaces continuously change their shapes, the proposed approach creates a blending surface, which connects the varying shapes together with C^2 continuity.

3.4.4.1.2 Surface blending between nonlinearly varying primary surfaces

For the second application example, two primary surfaces change their shapes following a nonlinear sine variation. The first primary surface initially has some wrinkles, and finally becomes the frustum of a smooth inclined circular cone. The second primary surface changes from a cone-shaped elliptic cylinder to an inclined plane.

The parametric equations of the first primary surface are constructed as

$$x = [(1 - \sin(\pi t/2)](1 + k_1 u^2)[r_0 \cos v + r_1 \cos(kv)] + r \sin(\pi t/2)\,(k_2 + u)^2 \cos v$$

$$y = [(1 - \sin(\pi t/2)](1 + k_1 u^2)[r_0 \sin v + r_1 \sin(kv)] + r \sin(\pi t/2)\,(k_2 + u)^2 \sin v$$

$$z = [(1 - \sin(\pi t/2)](h_0 + h_1 u) + \sin(\pi t/2)\,[h_2 + h_3(k_2 + u) + h_4 cosv] \quad (134)$$

The parametric equations for the second primary surface are constructed as

$$x = [(1 - \sin(\pi t/2)]a(k_3 + u)\cos v + r_2 \sin\left(\frac{\pi t}{2}\right)(k_4 + u)\cos(-\alpha)\cos v$$

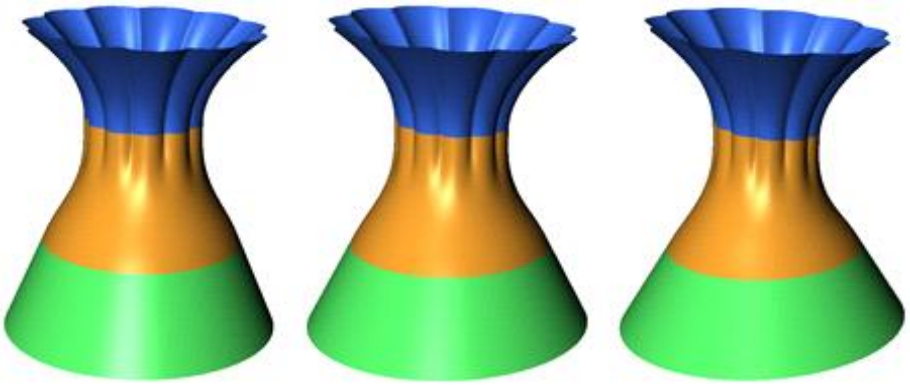
$$y = [(1 - \sin(\pi t/2)]b(k_3 + u)\sin v + r_2 \sin(\pi t/2)\,(k_4 + u)\sin v$$

$$z = [(1 - \sin(\pi t/2)][h_5 - h_6(k_5 + u)] + \sin(\pi t/2)[h_7 - h_8(k_4 + u)\sin(-\alpha) cosv] \quad (135)$$

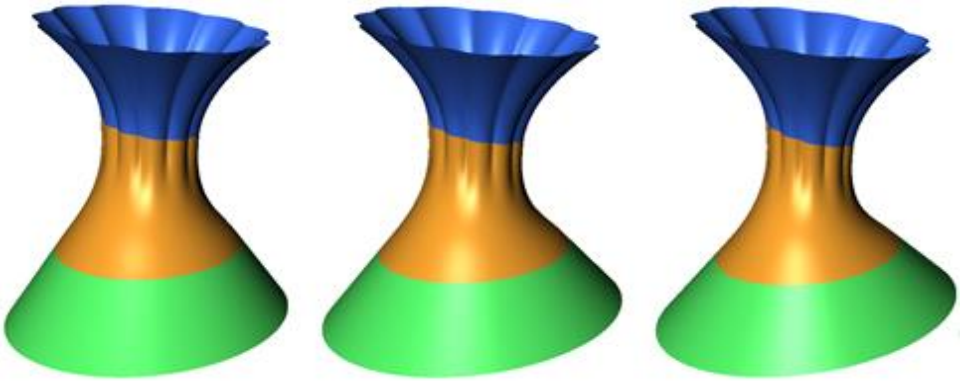
The geometric parameters in the above Eqs. (134) and (135) are taken to be: $a = 0.64, b = k_2 = 0.6, h_0 = 0.3, h_1 = 1.5, h_2 = h_5 = -1.0, h_3 = r_2 = 2.0, h_4 = 0.5, h_6 = 1.5, h_7 = 2.0,\ h_8 = 2.0,\ k = 12.0, k_1 = 1, k_3 = 2.5, k_4 = 0.8, k_5 = 0.5, r = 0.8, r_0 = 0.9, r_1 = 0.05$, and $\alpha = -\pi/9$.

For both primary surfaces, the trimlines are at $u_0 = u_1 = 0$. Using the proposed method, different shapes of the blending surface at the time instants $t = i/25\ (i = 0,1,2, \ldots 24,25)$ are depicted in Figure 25.

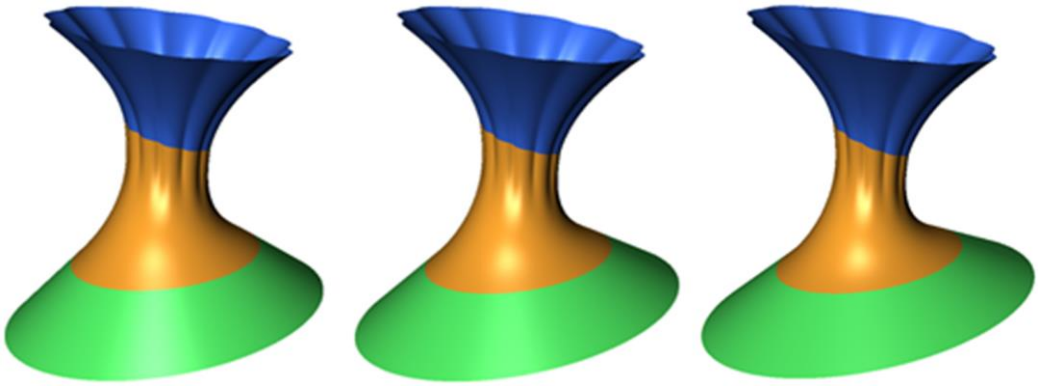
This example also demonstrates the capacity of our proposed approach in connecting two time-dependent primary surfaces together with C^2 continuity.



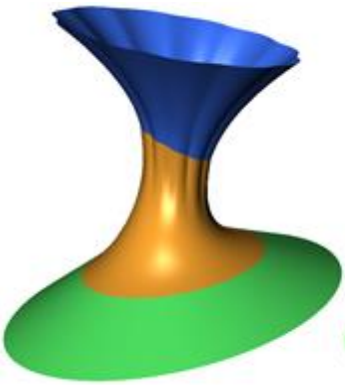
t=0 *t*=0.04 *t*=0.08



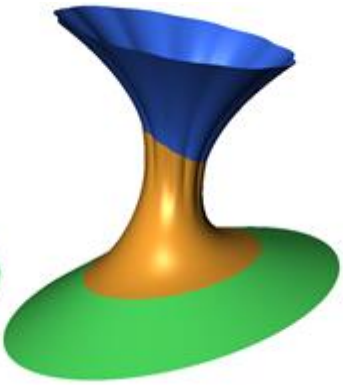
t=0.12 *t*=0.16 *t*=0.20



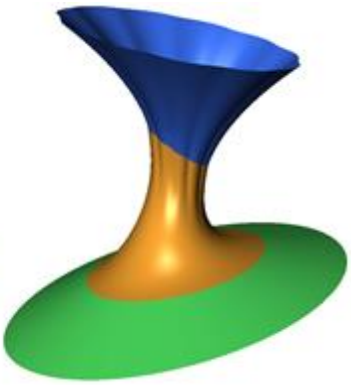
t=0.24 *t*=0.28 *t*=0.32



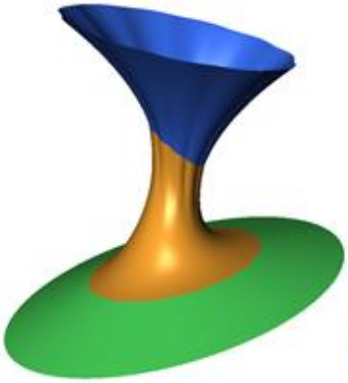
t=0.36



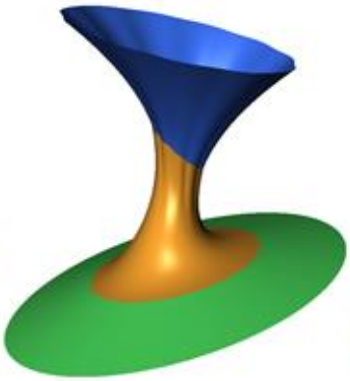
t=0.40



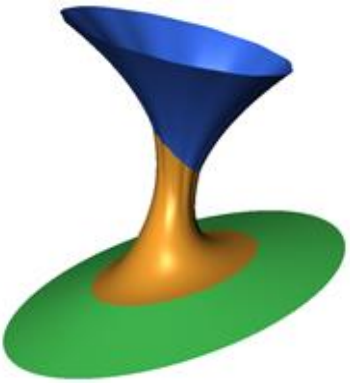
t=0.44



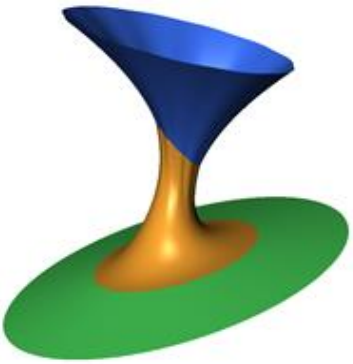
t=0.48



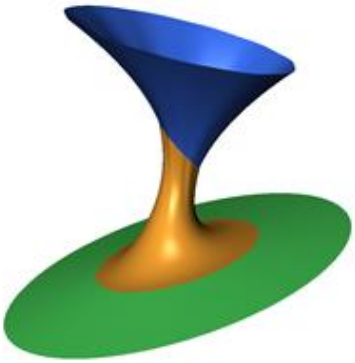
t=0.52



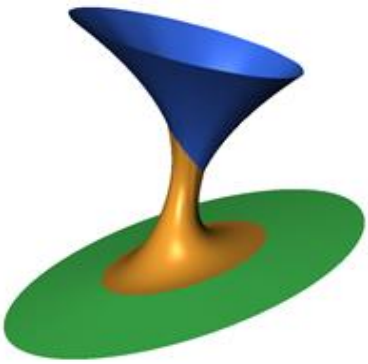
t=0.56



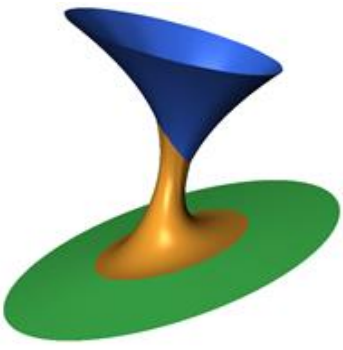
t=0.60

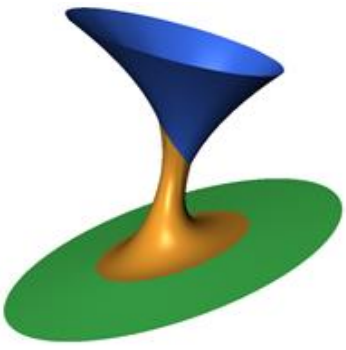


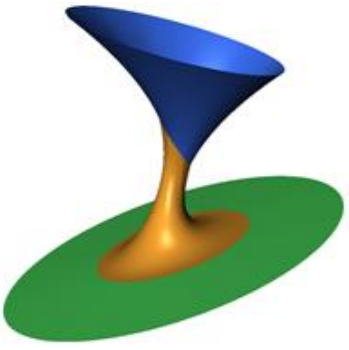
t=0.64



t=0.68







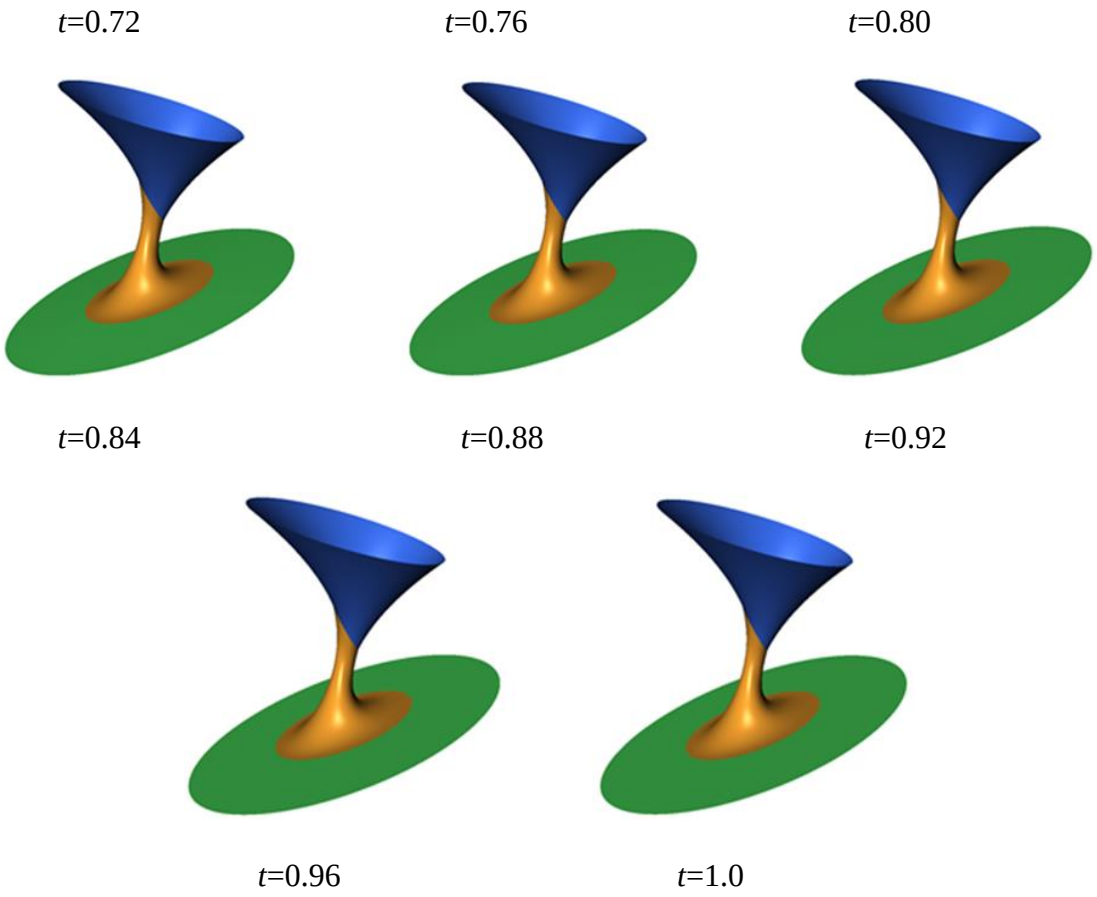


Figure 25: Surface blending between non-linearly varying primary surfaces

3.4.4.2 Time-independent surface blending

Time-independent surface blending is easier than time-dependent surface blending. The proposed approach is also effective in time-independent surface blending. In this section, some examples will be presented to demonstrate this and its industrial applications.

Blending surfaces which blend NURBS surfaces, intersecting planes, and intersecting cylinders are most common in geometric modeling, computer aided design, engineering manufacturing, and computer graphics. In what follows, how to use the above approach to tackle these surface blending problems will be investigated.

3.4.4.2.1 Surface blending between NURBS surfaces

A NURBS surface is a bivariate vector-valued piecewise rational function of the form

$$\boldsymbol{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{ij}\boldsymbol{P}_{ij}/\bar{S}\,(u,v) \tag{136}$$

where

$$\bar{S}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{ij} \tag{137}$$

and

$$N_{k,0}(t) = \begin{cases} 1 & \text{if}(t_i \le t < t_{i+1}) \\ 0 & otherwise \end{cases}$$

$$N_{k,l}(t) = \frac{t - t_i}{t_{k+l} - t_k} N_{k,l-1}(t) + \frac{t_{k+l+1} - t}{t_{k+l+1} - t_{k+1}} N_{k+1,l-1}(t)$$

$$(k = i,j; l = p,q; t = u,v) \tag{138}$$

where $t = u, v;\ l = p$ when $k = i$; and $l = q$ when $k = j$.

In the equation, p and q are the degrees in the u and v directions, respectively, $\boldsymbol{P}_{ij}$ are the control points, and w_{ij} are the weights.

When two NURBS surfaces $\boldsymbol{S}_1(u,v)$ and $\boldsymbol{S}_2(u,v)$ are to be smoothly blended together by a blending surface $\boldsymbol{S}(u,v)$ at the trimlines $\boldsymbol{C}_1(v) = \boldsymbol{S}_1(1,v)$ and $\boldsymbol{C}_4(v) = \boldsymbol{S}_2(0,v)$, the first and second partial derivatives of the two NURBS surfaces are derived, and $\boldsymbol{C}_2(v) = \partial\boldsymbol{S}_1(1,v)/\partial u$, $\boldsymbol{C}_3(v) = \partial^2\boldsymbol{S}_1(1,v)/\partial u^2$, $\boldsymbol{C}_5(v) = \partial\boldsymbol{S}_2(0,v)/\partial u$, and $\boldsymbol{C}_6(v) = \partial^2\boldsymbol{S}_2(0,v)/\partial u^2$ are obtained.

Substituting the obtained $\boldsymbol{C}_i(v)$ $(i = 1,2,\cdots,6)$ into Eqs. (122) and (123), and solving Eq. (125), all the unknown constants and the mathematical expression (126) of the blending surface are obtained.

To tackle various NURBS surface blending problems, a general case where 16 control points and 8 knots were used to generate the first cubic NURBS surface $\boldsymbol{S}_1(u,v)$ highlighted in blue in Figure 26, and 25 control points and 10 knots were used to generate the second quartic NURBS surface $\boldsymbol{S}_2(u,v)$ highlighted in light blue. The obtained blending surface $\boldsymbol{S}(u,v)$ was shown in grey in the same figure where the two images were obtained from two different viewpoints of a same blending surface.

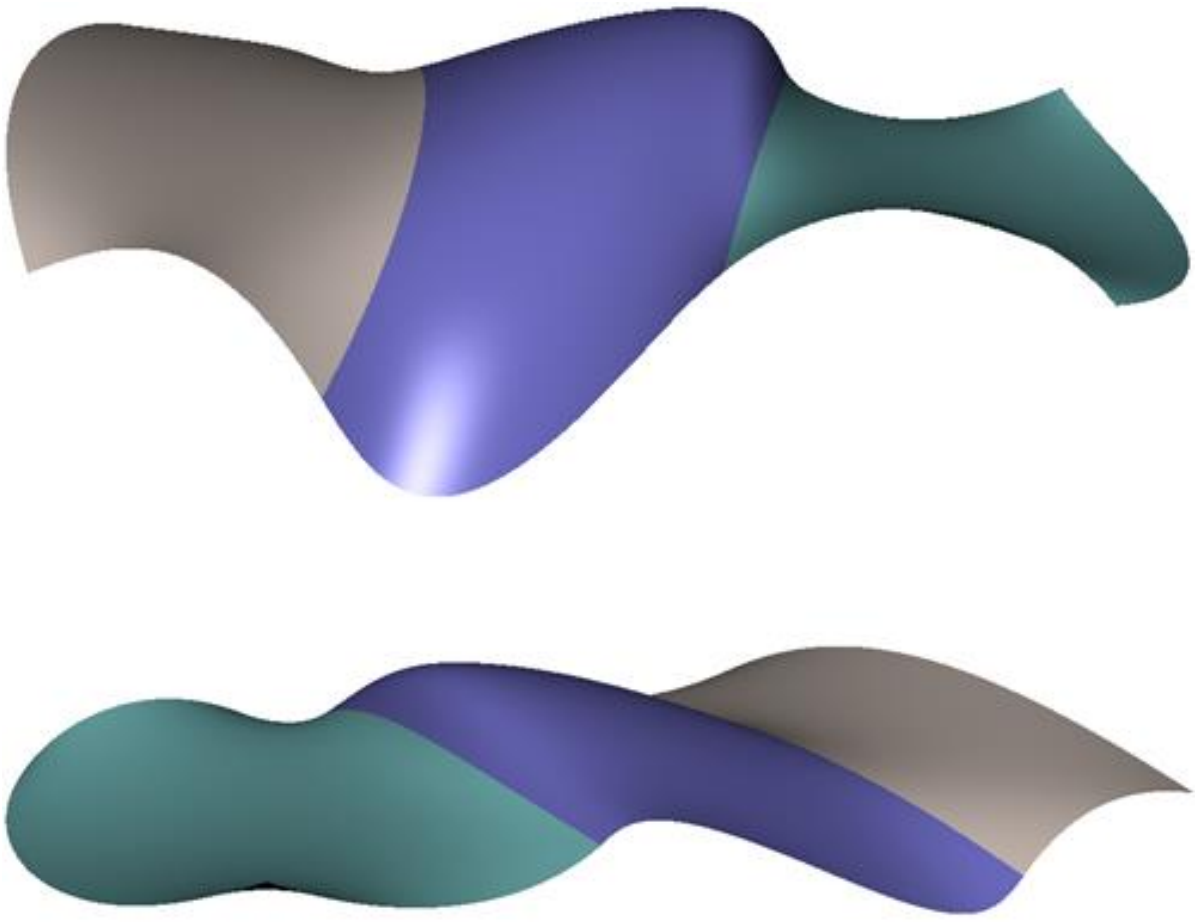


Figure 26: Blending between a cubic NURBS surface and a quartic NURBS surface

The images in Figure 26 show smooth transition from the blending surface to the two NURBS surfaces. It demonstrates the effectiveness of the proposed approach in blending NURBS surfaces with C^2 continuity.

3.4.4.2.2 Surface blending between intersecting planes

Generating a smooth transition surface between two intersecting planes frequently appears in engineering design or manufacturing process to reduce stress concentration at the joint between the two planes. For this surface blending problem, the boundary constraints can be formulated as:

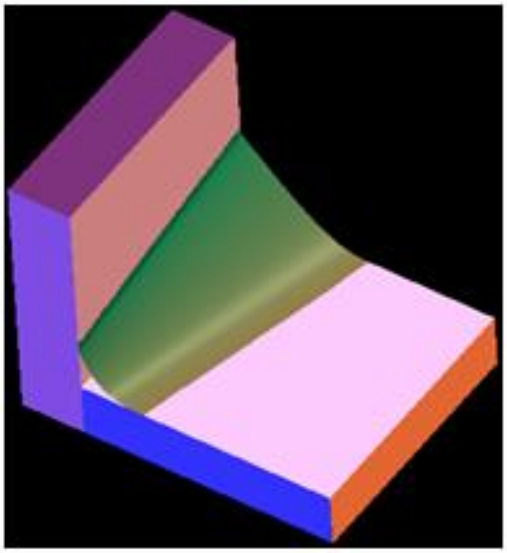
$$u = 0 \quad x = 0 \qquad y = h_0 + h_1 v \qquad z = pv$$

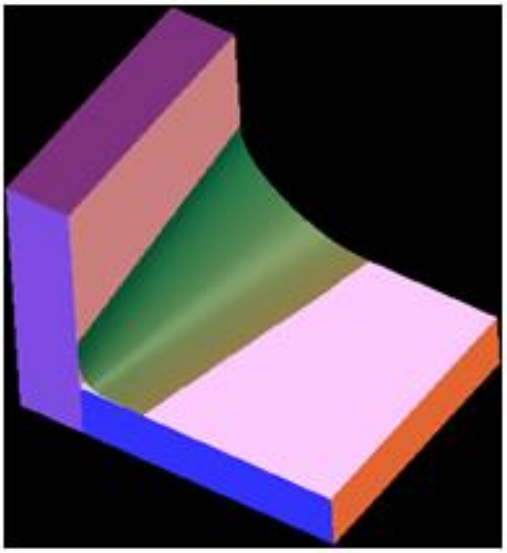
$$\frac{\partial x}{\partial u} = 0 \qquad \frac{\partial y}{\partial u} = -(h_2 + h_3 v) \qquad \frac{\partial z}{\partial u} = 0$$

$$\begin{array}{llll} & \frac{\partial^2 x}{\partial u^2} = 0 & \frac{\partial^2 y}{\partial u^2} = -\eta(h_2 + h_3 v) & \frac{\partial^2 z}{\partial u^2} = 0 \\ u = 1 & x = s_0 + s_1 v & y = 0 & z = pv \\ & \frac{\partial x}{\partial u} = s_2 + s_3 v & \frac{\partial y}{\partial u} = 0 & \frac{\partial z}{\partial u} = 0 \\ & \frac{\partial^2 x}{\partial u^2} = 0 & \frac{\partial^2 y}{\partial u^2} = 0 & \frac{\partial^2 z}{\partial u^2} = 0 \end{array} \quad (139)$$

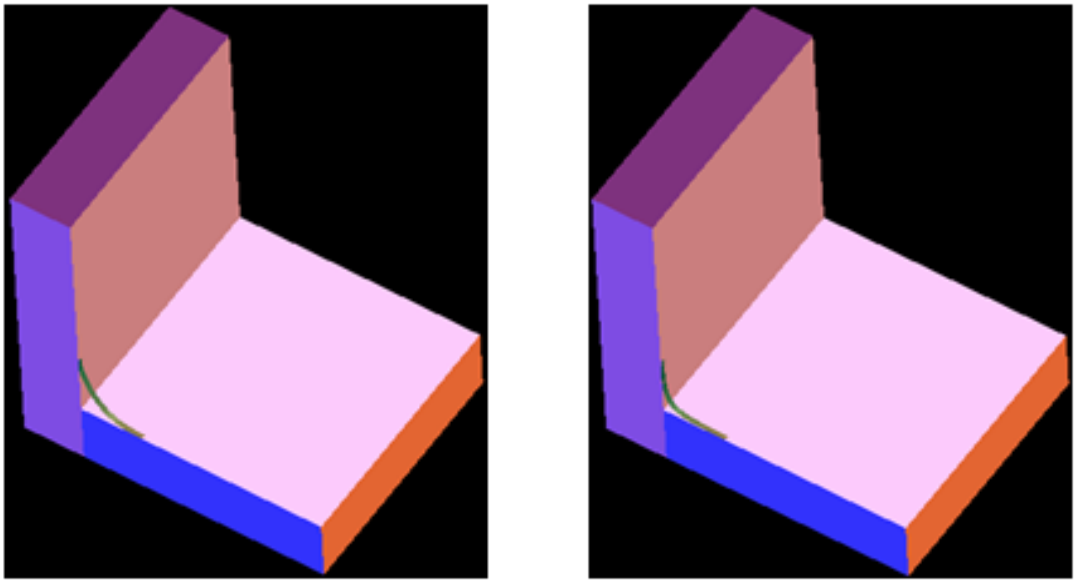
Setting the parameters in Eq. (139) to: $h_0 = 0.9$, $h_1 = -0.4$, $h_2 = 1.1$, $h_3 = s_3 = 0.01$, $s_0 = 1$, $s_1 = -0.5$, $s_2 = 1.2$and $p = 2$, the blending surface demonstrated in Figure 27(a) for $\eta = -10$ and Figure 27(b) for $\eta = 0$ is obtained where Figure 27(c) and 27(d) show a very small local part of the blending surfaces depicted in Figure 27(a) and 27(b). It can be seen that the second derivative in Eq. (139) determined by $\eta = 0$ creates a smoother blending surface than that determined by $\eta = -10$.

The proposed approach is advantageous over constant and variable radius rolling blends since it can achieve different levels of smoothness at trimlines and different shapes of blending surfaces. In contrast, constant and variable radius rolling blends cannot change both the smoothness and shape of the blending surface once the trimlines are specified.





(a): $\eta = -10$ (b): $\eta = 0$



(c): $\eta = -10$ (d) $\eta = 0$

Figure 27: Surface blending between perpendicular planes with inclined trimlines

3.4.4.2.3 Surface blending between intersecting cylinders

Creating a smooth transition between intersecting cylinders is also very common in engineering design and manufacturing. When blending two intersecting cylinders with C^2 continuity, the boundary constraints can be written as

$$u = 0$$

$$x = s\cos v \qquad y = s\sin v \qquad z = f_2(v)$$

$$\frac{\partial x}{\partial u} = 0 \qquad \frac{\partial y}{\partial u} = 0 \qquad \frac{\partial z}{\partial u} = f_3(v)$$

$$\frac{\partial^2 x}{\partial u^2} = 0 \qquad \frac{\partial^2 y}{\partial u^2} = 0 \qquad \frac{\partial^2 z}{\partial u^2} = f_4(v)$$

$$u = 1$$

$$x = (s + l_1)\cos v \qquad y = (s + l_1)\sin v \qquad z = f_5(v)$$

$$\frac{\partial x}{\partial u} = (s + l_1)\cos v \qquad \frac{\partial y}{\partial u} = (s + l_1)\sin v \qquad \frac{\partial z}{\partial u} = f_6(v)$$

$$\frac{\partial^2 x}{\partial u^2} = 0 \qquad \frac{\partial^2 y}{\partial u^2} = 0 \qquad \frac{\partial^2 z}{\partial u^2} = f_7(v) \tag{140}$$

where

$$f_2(v) = \sqrt{f_0(v)} \qquad f_3(v) = -(r + k_1)/\sqrt{f_0(v)}$$

$$f_4(v) = 1/\sqrt{f_0(v)} - (r + k_1)^2/\sqrt[3]{f_0(v)}$$

$$f_5(v) = \sqrt{f_1(v)}$$

$$f_6(v) = -(s + l_1)\cos^2 v/\sqrt{f_1(v)}$$

$$f_7(v) = \cos^2 v/\sqrt{f_1(v)} - (s + l_1)^2 \cos^4 v/\sqrt[3]{f_1(v)} \tag{141}$$

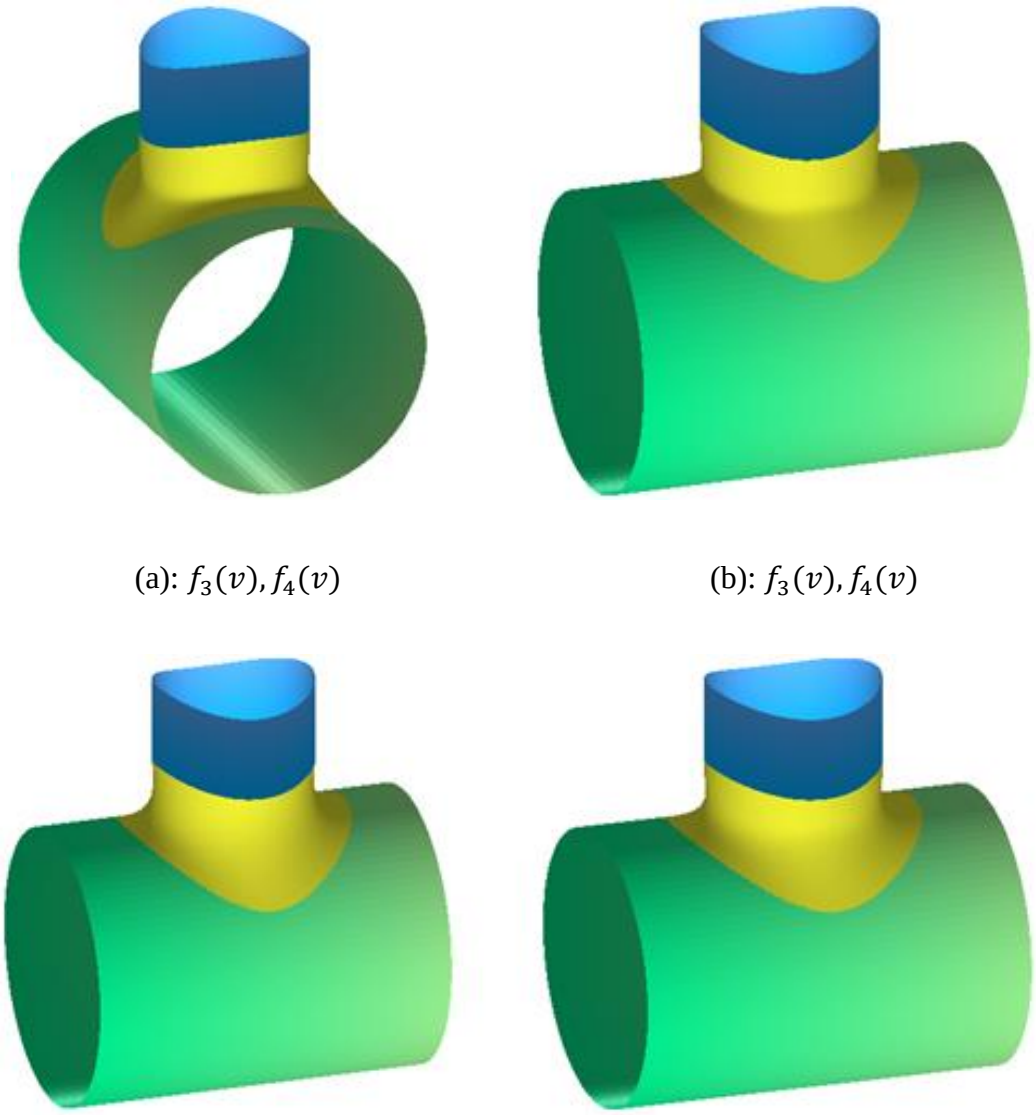
and

$$f_0(v) = (r + k_1)^2 - s^2 \cos^2 v$$

$$f_1(v) = r^2 - (s + l_1)^2 \cos^2 v \tag{142}$$

Setting the shape parameters to $\gamma = \eta = \lambda = \rho = 1$ and geometric parameters to $s = 0.7$, $l_1 = 0.3$, $r = 1.2$, and $k_1 = 0.5$, the blending surface depicted in Figure 28(a) and Figure 28(b) is obtained from $\partial z/\partial u = f_3(v)$ and $\partial^2 z/\partial u^2 = f_4(v)$ where the images shown in Figure 28(a) and Figure 28(b) are from different views of a same blending surface.

How the first and second partial derivatives affect the smoothness and shape of the blending surface can be obtained by scaling them. Setting $\partial z/\partial u = 0.01 f_3(v)$ and keeping $\partial^2 z/\partial u^2 = f_4(v)$ unchanged, the blending surface depicted in Figure 28(c) is obtained. Keeping $\partial z/\partial u = f_3(v)$ unchanged but setting $\partial^2 z/\partial u^2 = 10 f_4(v)$, the blending surface shown in Figure 28(d) is generated. These images demonstrate the effectiveness of the first and second partial derivatives in changing the smoothness and the shape of blending surfaces.



(a): $f_3(v), f_4(v)$ (b): $f_3(v), f_4(v)$

(c): $0.01f_3(v), f_4(v)$ (d): $f_3(v), 10f_4(v)$

Figure 28: Surface blending between intersecting cylinders

3.4.4.2.4 Surface blending between a cylinder and a plane

The final example is to create a time-independent C^2 continuous blending surface smoothly connecting a cylinder to a plane. It is widely applied in mechanical components and parts such as a transmission yoke - drive shaft shown in Figure 29(a) where the blending between the cylinders and planes in the highlighted region is required. The approximate analytical approach developed in this

section is used to obtain the blending surface which is shown in (b) and (c) of Figure 29 where (b) is rendered with a same colour and (c) is rendered with three different colours.

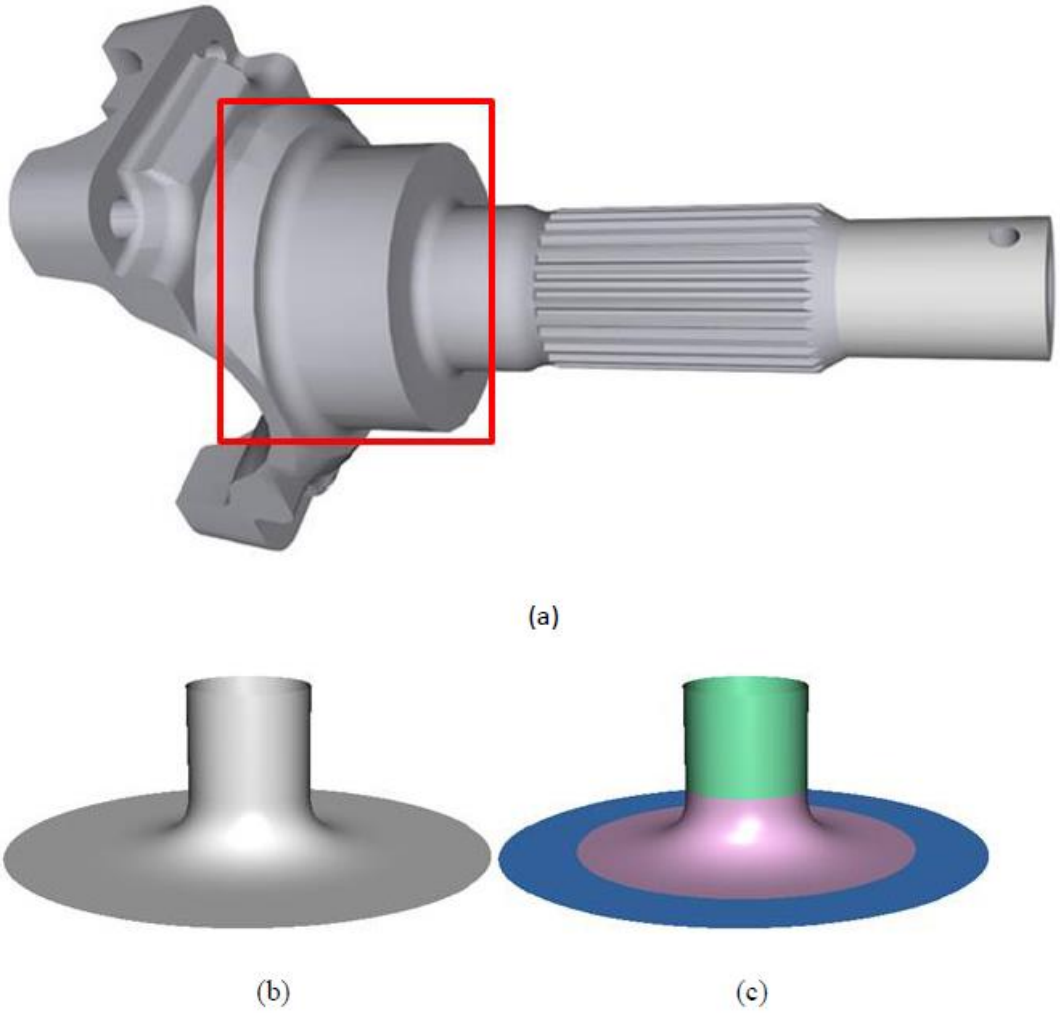


Figure 29: Surface blending between a cylinder and a plane for a transmission yoke - drive shaft

3.5 Conclusions and future work

In this chapter, a unified mathematical model for C^2 continuous blending between both time-dependent and time-independent parametric surfaces has been proposed and three different surface blending approaches were developed to create C^2 continuous blending surfaces. These three surface

blending approaches are: closed form solution-based approximate analytical approach, variable decomposition-based approximate analytical approach, and differential property-based approximate analytical approach. The closed form solution-based approximate analytical approach converts the functions in the blending boundary constraints without closed form solutions into those whose closed form solutions are obtainable. With such a treatment, the closed form mathematical expressions of blending surfaces are obtained and used to create various blending surfaces between two time-independent parametric surfaces. The variable decomposition-based approximate analytical approach groups the functions in the blending boundary constraints according to different variables, constructs blending surface functions accordingly, and develops the approximate analytical approach according to different blending surface functions and the corresponding blending boundary constraints. The differential property-based approximate analytical approach divides the functions in the blending boundary constraints according to different differential properties and obtains the corresponding functions of blending surfaces accordingly.

The accuracy, efficiency, effects of the second derivatives, and how different shape control parameters affect the shape of blending surfaces were investigated. It was found: (1) the proposed approaches have good accuracy and high efficiency; (2) the second partial derivatives play an important role in achieving good continuity; (3) all the shape control parameters have a strong impact on the shape of the blending surface and can be developed into useful shape control handles, i. e. sliders, to achieve the required shapes of blending surfaces. Some examples of time-independent and time-dependent surface blending were presented to demonstrate industrial applications of the proposed approach.

One of the main advantages of the proposed approaches is that the shape control parameters can be optimized to: (1) minimize stress concentrations in engineering applications, and (2) create user's specified shapes for aesthetic or other requirements. Stress concentrations are related to the curvature of blending surfaces between primary surfaces such as two intersecting planes to be

smoothly connected. Small curvature causes low stress concentrations. Therefore, minimizing stress concentrations is to find optimal shape control parameters which minimize the curvature of blending surfaces. To create user's specified shapes, one or more profile curves will be first drawn by users. The difference between the user's drawn profile curves and the corresponding ones of blending surfaces is minimized to obtain the optimal shape control parameters and create the user's specified shapes of blending surfaces. For example, when user draws a profile curve $\boldsymbol{P}(u)$, the corresponding profile curve of the blending surface is $\boldsymbol{S}(u, v_0,\ \gamma,\ \eta,\ \lambda,\ \rho)$. The shape control parameters $\gamma,\ \eta,\ \lambda$, and ρ can be optimized through $\partial\|\boldsymbol{S}-\boldsymbol{P}\|/\partial\,\gamma = 0$, $\partial\|\boldsymbol{S}-\boldsymbol{P}\|/\partial\,\eta = 0$, $\partial\|\boldsymbol{S}-\boldsymbol{P}\|/\partial\,\lambda = 0$, and $\partial\|\boldsymbol{S}-\boldsymbol{P}\|/\partial\,\rho = 0$ to make the blending surface approximate the user's drawn profile curve.

4. Facial blendshapes with differential equation-based shape interpolation

For facial blendshape animation, both realism and efficiency are important. In comparison with facial blendshape animation based on geometric interpolation, physics-based facial blendshape animation has better realism. However, physics-based facial blendshape animation involves heavy numerical calculations. Before this research, there are no research studies on analytical physics-based blendshape animation. The work described in this chapter is the first one of developing such a technique. It makes a good compromise between realism and efficiency. Considering the previous work (Barrielle et al., 2016; Kozlov et al., 2017) on facial animation uses Newton's second law of motion which does not take damping effects into account, the equation of motion adopted in (Park and Hodgins, 2008; Terzopoulos and Waters, 1993) was introduced to incorporate inertial effects, damping effects, and the resistance against shape deformations. The mathematical model of dynamic deformations is obtained by combining the equation of motion with the deformation constraints from source and target facial shapes. An analytical solution of the mathematical model is derived and used to develop an efficient blending force-based animation framework consisting of slider force-based, exponentiation force-based and random force-based facial blendshapes which can create various blended facial shapes highly efficiently. Unlike existing numerical methods that must process polygon models first such as converting polygon meshes into finite element or mass-spring meshes, the proposed analytical approach directly uses polygon vertices for physics-based facial blendshapes.

First, the mathematical model consisting of the equation of motion and the constraints of source and target facial models will be proposed. Next, a simple and efficient closed form analytical solution of the mathematical model will be obtained. Finally, a blend force-based animation

framework will be developed from the obtained analytical solution and slider force-based, exponentiation force-based, and random force-based facial blendshapes.

4.1 Mathematical model and analytical solution

As stated in (Barrielle et al., 2016), "*physics-based methods work off sound physical foundations and generally produce accurate skin motion*". In the paper by (Barrielle et al., 2016), a mass is specified to each of the polygon vertices of a facial mesh, and Newton's second law is used to obtain the temporal evolution of the facial mesh. When the facial mesh changes its shape from a neutral pose to a deformed pose, each of the vertices has a displacement, velocity, and acceleration. The acceleration leads to an inertia force, the displacement results in an elastic restoring force to resist the changes of the positions and shapes, and the facial mesh is deformed from the shape at the neutral pose into the shape at the deformed pose by an external force called an actuation force. The equilibrium of these forces leads to Newton's second law. The weakness of the proposed approach is that it does not consider the effect of the velocity of facial movements on facial deformations.

To tackle this weakness, many other approaches introduce a damping force into Newton's second law to consider the influence of the velocity. The approach proposed in (Warburton and Maddock, 2013) uses the principle of virtual work on the basis of the displacement-based finite element method to derive the equation of motion in matrix form which includes a mass matrix, damping matrix and stiffness matrix to simulate facial movements. The equation of motion involving a damping force is also used to obtain physics-based modelling of facial animation in (Terzopoulos and Waters, 1993; Lee et al., 1995) and combined with data-driven techniques to simulate skin and muscle deformation in (Park and Hodgins, 2008).

Motivated by the existing work, the same equation of motion involving a damping force adopted in (Park and Hodgins, 2008; Terzopoulos and Waters, 1993; Warburton and Maddock, 2013;

Lee et al., 1995) was used to simulate facial deformations, and propose a data-driven technique for synthesizing skin deformation from skeletal motion. The equation of motion can be written as

$$m\frac{d^2x^{(i)}}{dt^2} + c\frac{dx^{(i)}}{dt} + kx^{(i)} = f^{(i)}$$

$$(i = 1,2,3) \tag{143}$$

where m denotes mass which is used to provide inertia effects, c stands for the damping coefficient which is introduced to provide damping effects, k means the stiffness coefficient which reflects the resistance of facial skin against facial deformations, t is the time variable, the superscript i indicates the i^{th} component, i. e., $x^{(1)} = x,\ x^{(2)} = y$, and $x^{(3)} = z$ which represent the three components of deformations, and $f^{(1)} = f_x, f^{(2)} = f_y$, and $f^{(3)} = f_z$ indicate the three components of forces.

Deforming a face model at the neutral pose likes moving an object at the rest position where the displacement and velocity of the object at the rest position are zero. The deformation (displacement) and deformation rate (velocity) of the face model at the neutral pose $t = 0$ are also zero. At the deformed pose $t = 1$, the face model is deformed into various target shapes. Therefore, the following constraints can be obtained.

$$t = 0 \quad x^{(i)} = 0 \quad \frac{dx^{(i)}}{dt} = 0$$

$$t = 1 \quad x^{(i)} = x_2^{(i)} - x_1^{(i)}$$

$$(i = 1,2,3) \tag{144}$$

where the subscripts "1" and "2" indicate the neutral pose and deformed pose, respectively, and $x_1^{(i)}$ and $x_2^{(i)}$stand for the vertex coordinate values of the undeformed and deformed face polygon models respectively.

Eqs. (143) and (144) are the mathematical model of dynamic facial blendshapes. Various numerical methods can be used to solve the mathematical model. However, these numerical methods

have the problems discussed before. In this research, a simple and efficient analytical solution of the mathematical model has been developed.

According to the theory of differential equations[5, 6], the solution to nonhomogeneous second-order ordinary differential Eq. (143) contains a general solution $x_h^{(i)}$to the homogenous version of Equation (143) and a particular solution $x_p^{(i)}$ caused by the right-hand-side mathematical expression. The general solution to the homogenous version of Eq. (143) can be converted into a characteristic equation by letting[7]

$$x_p^{(i)} = e^{rt}$$

$$(i = 1,2,3) \tag{145}$$

Substituting Eq. (145) into the homogenous version of (143), the following equation is obtained.

$$mr^2e^{rt} + cre^{rt} + ke^{rt} = 0$$

After deleting e^{rt} the above equation, the homogenous version of the nonhomogeneous second-order ordinary differential equation (143) is changed into the following characteristic equation

$$mr^2 + cr + k = 0 \tag{146}$$

The two roots of Eq. (146) can be written in the following form

$$r_{1,2} = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m} \tag{147}$$

Depending on the different values of $c^2 - 4mk$, the roots given by Eq. (147) and the analytical solution to Eq. (143) subjected to the constraints (144) can be divided into the following three cases:

First case: $c^2 - 4mk = 0$,

[5] https://math.libretexts.org/Bookshelves/Differential_Equations/Book%3A_Differential_Equations_for_Engineers_(Lebl)/2%3A_Higher_order_linear_ODEs/2.2%3A_Constant_coefficient_second_order_linear_ODEs

[6] https://math.libretexts.org/Bookshelves/Calculus/Book%3A_Calculus_(OpenStax)/17%3A_Second-Order_Differential_Equations/17.2%3A_Nonhomogeneous_Linear_Equations

[7] https://nrich.maths.org/11054

Second case: $c^2 - 4mk > 0$,

Third case: $c^2 - 4mk < 0$.

The above three cases will be discussed in Subsections 4.1.1, 4.1.2, and 4.1.3 below, respectively.

4.1.1 Analytical solution for the first case

For the first case, $c^2 - 4mk = 0$, Eq. (147) gives two repeated real roots below

$$r_{1,2} = \frac{-c}{2m} = q \tag{148}$$

The general solution of the homogenous version of Eq. (143) for this case is[8]

$$x_h^{(i)} = (c_1 + c_2 t)e^{qt}$$

$$(i = 1,2,3) \tag{149}$$

where c_1 and c_2 are two integration constants.

substituting Eq. (149) into the equation $m\frac{d^2x^{(i)}}{dt^2} + c\frac{dx^{(i)}}{dt} + kx^{(i)} = 0$, it is proved that the equation $m\frac{d^2x^{(i)}}{dt^2} + c\frac{dx^{(i)}}{dt} + kx^{(i)} = 0$ is exactly satisfied. Therefore $x_h^{(i)} = (c_1 + c_2 t)e^{qt}$ is the general solution of the homogenous version of Eq. (143).

The particular solution of the nonhomogeneous second-order ordinary differential equation (143) can be[9]

$$x_p^{(i)} = c_3 + c_4 t$$

$$(i = 1,2,3) \tag{150}$$

where $c_3$and c_4 are two unknown constants.

Substituting Eq. (150) into (143), the two unknown constants are determined. Introducing them back into Eq. (150), the particular solution is obtained below.

[8] https://www.math.hkust.edu.hk/~machas/differential-equations-for-engineers.pdf
[9] https://www.efunda.com/math/ode/linearode_undeterminedcoeff.cfm

$$x_p^{(i)} = f^{(i)}/k$$

$$(i = 1,2,3) \tag{151}$$

Putting Eqs. (149) and (151) together, the solution to Eq. (143) is found to be

$$x^{(i)} = x_h^{(i)} + x_p^{(i)} = (c_1 + c_2 t)e^{qt} + f^{(i)}/k$$

$$(i = 1,2,3) \tag{152}$$

The two integration constants c_1 and c_2 can be determined by substituting Eq. (152) into the constraints (144) which gives

$$x^{(i)} = \frac{1 + (-1 + tq)e^{qt}}{t} f^{(i)}$$

$$(i = 1,2,3) \tag{153}$$

where

$$f^{(i)} = \frac{k\left[x_2^{(i)} - x_i^{(i)}\right]}{1 + (-1 + q)e^{q}}$$

$$(i = 1,2,3) \tag{154}$$

4.1.2 Analytical solution for the second case

For the second case, $c^2 - 4mk > 0$, Eq. (147) gives two different real roots below

$$r_1 = \frac{-c + \sqrt{c^2 - 4mk}}{2m} = q_1$$

$$r_2 = \frac{-c + \sqrt{c^2 - 4mk}}{2m} = q_2 \tag{155}$$

The general solution of the homogenous version of Eq. (143) for this case is

$$x_h^{(i)} = c_1 e^{q_1 t} + c_2 e^{q_2 t}$$

$$(i = 1,2,3) \tag{156}$$

Putting the above equation and Eq. (151) together, the solution to Eq. (143) is

$$x^{(i)} = c_1 e^{q_1 t} + c_2 e^{q_2 t} + f^{(i)}/k$$

$$(i = 1,2,3) \tag{157}$$

Substituting Eq. (157) into the constraints (144) to determine the two integration constants $c_1$and c_2, and introducing them back into Eq. (157), the following equation is obtained.

$$x^{(i)} = \frac{q_1 - q_2 + q_2 e^{q_1 t} - q_1 e^{q_2 t}}{k(q_1 - q_2)} f^{(i)}$$

$$(i = 1,2,3) \tag{158}$$

where

$$f^{(i)} = \frac{k(q_1 - q_2)\left[x_2^{(i)} - x_1^{(i)}\right]}{q_1 - q_2 + q_2 e^{q_1} - q_1 e^{q_2}}$$

$$(i = 1,2,3) \tag{159}$$

4.1.3 Analytical solution for the third case

For the third case, $\boldsymbol{c^2 - 4mk < 0}$**,** Eq. (147) gives two different complex roots below

$$r_{1,2} = q_3 \pm j q_4 \tag{160}$$

where *j* is an imagery number, and

$$q_3 = -\frac{c}{2m} \qquad q_4 = \frac{\sqrt{4mk - c^2}}{2m} \tag{161}$$

The general solution of the homogenous version of Eq. (143) for this case is

$$x_h^{(i)} = e^{q_3 t}(c_1 cosq_4 t + c_2 cosq_4 t)$$

$$(i = 1,2,3) \tag{162}$$

Putting the above equation and Eq. (151) together, the solution to Eq. (143) is obtained below.

$$x^{(i)} = e^{q_3 t}(c_1 cosq_4 t + c_2 cosq_4 t) + f^{(i)}/k$$

$$(i = 1,2,3) \tag{163}$$

Substituting Eq. (163) into the constraints (144) to determine the two integration constants c_1 and c_2, and introducing the obtained unknow constants back into Eq. (163), the following solution is obtained.

$$x^{(i)} = \frac{q_4 + e^{q_3 t}(q_3 sinq_4 t - q_4 cosq_4 t)}{kq_4} f^{(i)}$$

$$(i = 1,2,3) \tag{164}$$

where

$$f^{(i)} = \frac{kq_4\left[x_2^{(i)} - x_1^{(i)}\right]}{e^{q3}(q_4 e^{-q_3} + q_3 sinq_4 - q_4 cosq_4)}$$

$$(i = 1,2,3) \tag{165}$$

From one of Eqs. (154), (159) and (165), the force functions are obtained. Substituting them into the corresponding Eqs. (153), (158) or (164), the coordinate values of the blended shapes at time t are calculated. In Section 4.3 below, Eqs. (153-154), (158-159) and (164-165) will be using them to develop an analytical blending force-based animation framework after the experiments and comparisons are investigated in Section 4.2.

4.2 Experiments and comparisons

In this section, the numerical method applied in (Lee et al., 1995) and the linear interpolation method employed in (Seo et al., 2011) will be used in this subsection for the same facial blend shape task. The deformation formulae (153), (158), and (164) and force functions (154), (159) and (165) of our proposed analytical solution, the numerical method, and the linear interpolation method will be implemented and elaborated below. Some experiments are presented in this section to compare the blended shapes obtained from the proposed analytical solution, the numerical method, and the linear interpolation method, which are visualized in Maya.

The numerical method proposed in (Lee et al., 1995) is well-known, explicit Euler method. It is widely used to solve the equation of motion. In order to validate the derived analytical solutions, this numerical method will be used to solve the same differential equation (143) subjected to the constraints (144).

Firstly, the time range [0,1] is uniformly discretized into J equal time intervals leading to a time step length $\Delta t = 1/J$. If $a_{t_j}^{(i)}$ is used to replace the acceleration $d^2x^{(i)}/dt^2$, $v_{t_j}^{(i)}$ is used to replace the velocity $dx^{(i)}/dt$, $d_{t_j}^{(i)}$ is used to replace the deformation $x^{(i)}$, and $f_n^{(i)}(n = 0,1,2,3, \dots)$ is used replace the external forces $f_0^{(i)}$in Eq. (143) at the instant $t_j = j/J(j = 0,1,2,3, \dots, J)$, Eq. (143) is changed into the following equation to determine the acceleration $a_{t_j}^{(i)}$

$$a_{t_j}^{(i)} = \left[\frac{d^2x^{(i)}}{dt^2}\right] = \frac{1}{m}\left[f_n^{(i)} - cv_{t_j}^{(i)} - kd_{t_j}^{(i)}\right]$$

$$(j = 0,1,2,3, \dots, J-1; n = 0,1,2,3, \dots)$$

(166)

At the neutral pose, $j = 0, t_0 = 0$, the constraints (144) give $v_{t_0}^{(i)} = 0$, and $d_{t_0}^{(i)} = 0$. At the final pose, the vertex $x_1^{(i)}$ is moved to $x_2^{(i)}$. The displacement between $x_2^{(i)}$ and $x_1^{(i)}$ is $x_2^{(i)} - x_1^{(i)}$. According to Hooke's law, the restoring force caused by the displacement is $f_0^{(i)} = k\left[x_2^{(i)} - x_1^{(i)}\right]$ where k is the stiffness coefficient. As the first approximation, we take $f_0^{(i)} = k\left[x_2^{(i)} - x_1^{(i)}\right]$, and obtain $a_{t_0}^{(i)} = \frac{1}{m}\left[f_0^{(i)} - cv_{t_0}^{(i)} - kd_{t_0}^{(i)}\right] = \frac{1}{m}f_0^{(i)}$ from Eq. (166). Then the following equations are used to calculate $v_{t_1}^{(i)}$and $x_{t_1}^{(i)}$.

$$v_{t_{j+1}}^{(i)} = v_{t_j}^{(i)} + \Delta t a_{t_j}^{(i)}$$

$$x_{t_{j+1}}^{(i)} = x_{t_j}^{(i)} + \Delta t v_{t_{j+1}}^{(i)}$$

$$d_{t_{j+1}}^{(i)} = x_{t_{j+1}}^{(i)} + x_1^{(i)}$$

$$(j = 0,1,2,3, \dots, J-1) \quad (167)$$

where $v_{t_0}^{(i)} = 0$, $x_{t_0}^{(i)} = x_1^{(i)}$

After obtaining $v_{t_1}^{(i)}$and $x_{t_1}^{(i)}$, Eq. (166) is used to obtain $a_{t_1}^{(i)} = \frac{1}{m}\left[f_0^{(i)} - cv_{t_1}^{(i)} - kd_{t_1}^{(i)}\right]$ and Eq. (167) is used to obtain $v_{t_2}^{(i)}$and $x_{t_2}^{(i)}$. The calculations are repeated until $j = J - 1$. When $j = J -$

1, $x_{t_J}^{(i)} = x_{t_{J-1}}^{(i)} + \Delta t v_{t_J}^{(i)}$ is obtained. Then we set $\bar{x}_0^{(i)}$ to $x_{t_J}^{(i)}$, i. e., $\bar{x}_0^{(i)} = x_{t_J}^{(i)}$ and check whether $x_2^{(i)} - \bar{x}_0^{(i)}$ satisfies the required accuracy ε. If $\left|x_2^{(i)} - \bar{x}_0^{(i)}\right| \le \varepsilon$, $\bar{x}_0^{(i)}$ is very close to $x_2^{(i)}$ and the third constraints $x^{(i)} = x_2^{(i)} - x_1^{(i)}$ at $t = 1$ is satisfied. The obtained $x_{t_j}^{(i)}$ $(j = 1, 2, 3, \cdots, J-1)$ corresponding to $\bar{x}_0^{(i)}$ are the obtained blended shapes. If $\left|x_2^{(i)} - \bar{x}_0^{(i)}\right| > \varepsilon$, the assumed force $f_0^{(i)}$ should be modified to generate more accurate solutions $x_{t_j}^{(i)}$ $(j = 1, 2, 3, \cdots, J-1)$. When $x_2^{(i)} > \bar{x}_0^{(i)}$, the assumed force is too small. We add a positive force increment $\frac{x_2^{(i)} - \bar{x}_0^{(i)}}{x_2^{(i)} - x_1^{(i)}} f_0^{(i)}$ to $f_0^{(i)}$ to make $f_0^{(i)}$ bigger and obtain the modified force $f_1^{(i)} = f_0^{(i)} + \frac{x_2^{(i)} - \bar{x}_0^{(i)}}{x_2^{(i)} - x_1^{(i)}} f_0^{(i)} = \left[1 + \frac{x_2^{(i)} - \bar{x}_0^{(i)}}{x_2^{(i)} - x_1^{(i)}}\right] f_0^{(i)}$. When $x_2^{(i)} < \bar{x}_0^{(i)}$, the assumed force is too big. We add a negative force increment $\frac{x_2^{(i)} - \bar{x}_0^{(i)}}{x_2^{(i)} - x_1^{(i)}} f_0^{(i)}$ to $f_0^{(i)}$ to make $f_0^{(i)}$ smaller and obtain the modified force $f_1^{(i)} = \left[1 + \frac{x_2^{(i)} - \bar{x}_0^{(i)}}{x_2^{(i)} - x_1^{(i)}}\right] f_0^{(i)}$.

After determining $f_1^{(i)}$, we obtain $a_{t_0}^{(i)} = \frac{1}{m}\left[f_1^{(i)} - c v_{t_0}^{(i)} - k d_{t_0}^{(i)}\right] = \frac{1}{m} f_1^{(i)}$ from Eq. (166) and $v_{t_1}^{(i)}$ and $x_{t_1}^{(i)}$ from Eq. (167). Then, we use Eq. (166) to obtain $a_{t_1}^{(i)} = \frac{1}{m}\left[f_1^{(i)} - c v_{t_1}^{(i)} - k d_{t_1}^{(i)}\right]$ and Eq. (167) is used to obtain $v_{t_2}^{(i)}$ and $x_{t_2}^{(i)}$. The calculations are repeated until $j = J - 1$. When $j = J - 1$, $x_{t_J}^{(i)}$ is obtained. Then we set $\bar{x}_1^{(i)}$ to $x_{t_J}^{(i)}$, i. e., $\bar{x}_1^{(i)} = x_{t_J}^{(i)}$ and check whether $x_2^{(i)} - \bar{x}_1^{(i)}$ satisfies the required accuracy ε. If $\left|x_2^{(i)} - \bar{x}_1^{(i)}\right| \le \varepsilon$, the obtained $x_{t_j}^{(i)}$ $(j = 1, 2, 3, \cdots, J-1)$ corresponding to $\bar{x}_1^{(i)}$ are the obtained blended shapes. If $\left|x_2^{(i)} - \bar{x}_1^{(i)}\right| > \varepsilon$, we use $f_2^{(i)} = \left[1 + \frac{x_2^{(i)} - \bar{x}_1^{(i)}}{x_2^{(i)} - x_1^{(i)}}\right] f_1^{(i)}$ to modify the force. Writing the force modification into an iterative form, we obtain

$$f_n^{(i)} = \left[1 + \frac{x_2^{(i)} - \bar{x}_{n-1}^{(i)}}{x_2^{(i)} - x_1^{(i)}}\right] f_{n-1}^{(i)}$$

$$(n = 1, 2, 3, \ldots) \tag{168}$$

The above process is repeated until $\left|x_2^{(i)} - \bar{x}_n^{(i)}\right| < \varepsilon$. The $x_{t_{j+1}}^{(i)} (j = 1, 2, 3, \dots, J-1)$ corresponding to $\bar{x}_n^{(i)}$ are the obtained blended shapes.

The linear interpolation method has two different versions: one is the linear blending between whole face shapes that can be mathematically written as: $x = \sum_{j=0}^{N} w_j x_j$ (Lewis et al., 2014) where x_0 stands for the neutral shape, and x_j $(j = 1, 2, 3, \dots, N)$ indicate N target shapes, and the other is the "delta" or offset blendshape formulation, which is most often used in the current commercial modelling and animation software packages for blendshape applications. The blendshape model used in (Seo et al., 2011) is based on the "delta" blendshape formulation which has the form of Equation $x = x_0 + \sum_{j=1}^{N} w_j (x_j - x_0)$.

In this subsection, the proposed analytical solution is compared with the "delta" blendshape formulation used in (Seo et al., 2011).

All the three methods: the proposed analytical solution, the numerical method, and the linear interpolation method, are implemented with C++ and OpenGL. In the remaining part of this subsection, a comparison among our proposed analytical solution, the numerical method, and the linear interpolation method will be made.

The parameters q and $q_i (i = 1,2,3,4)$ involved in the proposed analytical solution, i.e., Eqs. (153)-(154), (158)-(159) and (164)-(165), and the numerical solution (166)-(168) depend on mass m, damping coefficient c, and the stiffness coefficient k. It indicates that different mass, damping coefficient, and stiffness coefficient can be used to create different facial shapes. Here two groups of different values are used to demonstrate this, and the proposed analytical solution is compared with the numerical method. The first group is: $m = 1.5, c = 8, \text{and } k = 10$, and the second group is: $m = 0.8, c = 7.5, \text{and } k = 20$. These two groups of values are obtained by roughly referring to the papers by Liang and Boppert (2010) for density, Ulusoy et al. (2010) for damping coefficient, and Wei et al. (2017) for stiffness coefficient. The source shape of a cartoon face at the neutral pose

t=0 and its target shape at the deformed pose t=1 obtained from the link[10] are shown in the column (a) and column (f) of Figure 30.

The proposed analytical solution is accurate. First, how the numerical solution converges to the proposed analytical solution is demonstrated. Compared with all the coordinates of the undeformed cartoon face shown in Figure 30(a) and the deformed cartoon face shown in Figure 30(f), the y coordinate of vertex 554 denoted with a small red dot in the first image in Figure 30(a) has the largest displacement since it moves from the position at lower eyelid in Figure 30(a) to the position at upper eyelid in Figure 30(f) when the eye closes. Here it is used to make the comparison. Taking m=1.5, c=8, and k=10 and setting the required accuracy to $\varepsilon = 10^{-13}$ which is infinitesimal in comparison with the maximum size of the cartoon face, which is 40.11, the y coordinate values at t=0.5 obtained from different time step lengths Δt of the numerical method and the proposed analytical solution are given in Table 5 where AS indicated the proposed analytical solution, the second row is the obtained y coordinate values, the third row is the errors of the numerical method relative to the proposed analytical solution, and the fourth row gives the CPU time used to obtain the y coordinate values on a PC with Intel® Xeon® CPU E5-1650 V2 @ 3.5 GHz and 32 GB of memory.

The data in Table 5 indicate that with the decrease of the time step length, the numerical solution converges very slowly to the proposed analytical solution, i. e., from 2.051713 to 1.750848, but the CPU time quickly increases. The slow convergence of the numerical method can be also concluded from the same reduction order. As shown in Table 5, when the step length reduces by one order, the error caused by the numerical method also reduces by one order. When the time step length $\Delta t = 10^{-1}$, the error is 1.718×10^{-1}, which is large, and the CPU time is 1.06×10^{-3} ms. When the time step length decreases to $\Delta t = 10^{-7}$, the error drops to 5.71×10^{-7}, but the CPU time rises to 2,528 ms. In contrast, the proposed analytical solution only takes to 8.6×10^{-8} ms.

[10] https://www.creativebloq.com/3d-world/download-files-3d-world-211-61420631

Table 5. Convergence and timing of the numerical method

Δt	10^{-1}	10^{-2}	10^{-3}	10^{-4}
y	2.051713	1.780207	1.753777	1.75114
Error	1.718×10^{-1}	1.68×10^{-2}	1.67×10^{-3}	1.67×10^{-4}
Time (ms)	0.00106	0.025	0.167	1.4
Δt	10^{-5}	10^{-6}	10^{-7}	AS
y	1.750877	1.750850	1.750848	1.750847
Error	1.71×10^{-5}	1.71×10^{-6}	5.71×10^{-7}	0
Time (ms)	16	218	2,528	8.6×10^{-8}

The above comparison can be extended to the whole cartoon face. Using $\boldsymbol{x}_{0l}$ to indicate the l^{th} vertex on the blended shapes obtained from the proposed analytical solution and $\boldsymbol{x}_{1l}$ to indicate the corresponding l^{th} vertex on the blended shapes obtained from the above numerical solution, the Euclidean distance between $\boldsymbol{x}_{0l}$ and $\boldsymbol{x}_{1l}$ is $d(\boldsymbol{x}_{0l}, \boldsymbol{x}_{1l})$ (Mora et al., 2016) and gives a measure of the error between two corresponding vertices. The average error and the maximum error between two blended shapes obtained from the numerical method and the proposed analytical solution can be calculated from the Euclidean distance through the equations below

$$E_M = max\{d(\boldsymbol{x}_{0l}, \boldsymbol{x}_{1l})\}$$

$$E_A = \frac{1}{L} max\{d(\boldsymbol{x}_{0l}, \boldsymbol{x}_{1l})\}$$

$$(l = 1,2,3, \dots, \bar{L}) \tag{169}$$

where $\bar{L}$ is the number of the total vertices on a blended shape.

Using the mass m=1.5, damping coefficient c=8, and stiffness coefficient k=10, the time step length $\Delta t = 10^{-2}$ and the required accuracy $\varepsilon = 10^{-13}$, the blended shapes at the poses $t = 0.2$, 0.4, 0.6 and 0.8 obtained from the numerical method, the proposed analytical solution, and the

linear interpolation method are shown in the first, second and the third rows of Figure 30. To demonstrate different mass, damping coefficient, and stiffness coefficient can be used to achieve different facial shapes, the mass m=0.8, damping coefficient c=7.5, and stiffness coefficient k=20 are used in the proposed analytical solution to create the blended shapes shown in the fourth row of Figure 30.

The obtained average errors and the maximum errors between the numerical method and the proposed analytical solution are given in Table 5. In the table, the second to fifth column presents the errors of the numerical method with $\Delta t = 10^{-2}$ and $\varepsilon = 10^{-13}$ relative to our proposed analytical solution.

Table 6. Error comparison between the numerical method and the proposed analytical solution

	$t = 0.2$	$t = 0.4$	$t = 0.6$	$t = 0.8$
E_{A1}	4.1×10^{-2}	3.5×10^{-2}	4.2×10^{-2}	3.4×10^{-2}
E_{M1}	8.6×10^{-1}	7.2×10^{-1}	8.7×10^{-1}	7.1×10^{-1}

Even if the time step length is not very small ($\Delta t = 10^{-2}$), any visual differences among the blended shapes shown in the first and third rows of Figure 30 cannot be found. The data in Table 6 also show small errors between the numerical method and the proposed analytical solution. The errors at different time instants are at the same order. It indicates the correctness of the calculations. The errors will be further reduced if a smaller time step length is used. The CPU time used to obtain the blended shapes are 0.87ms for the proposed analytical solution, 0.68ms for the linear interpolation method, and 14.42ms for the numerical method (Even if the required accuracy is set to $\varepsilon = 1$ and the time step length keeps $\Delta t = 10^{-2}$ unchanged, the numerical method still requires 14.42ms to generate the blended shapes). They indicate that the proposed analytical solution is very

efficient, achieves almost as high efficiency as the linear interpolation method, and creates blended shapes much more quickly than the numerical method.

Numerical solution (m=1.5, c=8, k=10)

Proposed analytical solution (m=1.5, c=8, k=10)

Linear interpolation

Proposed analytical solution (m=0.8, c=7.5, k=20)

$t = 0$	$t = 0.2$	$t = 0.4$	$t = 0.6$	$t = 0.8$	$t = 1$
(a)	(b)	(c)	(d)	(e)	(f)

Figure 30: Comparison of blended shapes obtained by the numerical method, the proposed analytical solution, and the linear interpolation method

Comparing the images depicted in Figure 30, three conclusions can be drawn. First, the analytical approach of the dynamic facial blendshapes proposed in this thesis uses the first group of mass, damping coefficient, and stiffness coefficient to achieve the same visual results (Second row of Figure 30) of facial blendshapes as those generated by the linear interpolation method (Third row). It indicates that the proposed approach has a capacity to generate all the blended shapes generated by the linear interpolation method. Second, the analytical approach of the dynamic facial blendshapes uses the second group of mass, damping coefficient, and stiffness coefficient to achieve different results as clearly demonstrated by the images shown in (c), (d) and (e) of the fourth row. It indicates that the blended shapes generated by the linear interpolation method are a subset of the blended shapes generated by the approach proposed in this thesis. The proposed approach can generate a larger superset of blended shapes than the linear interpolation method. When two or more target shapes are blended together, the superset generated by the proposed approach will become much bigger than the subset generated by the linear interpolation method. Third, the linear interpolation method can only generate blended shapes with a same deformation rate. Differently, the proposed approach can generate blended shapes with different deformation rates, i. e., acceleration and deceleration effects stated in (Noh and Neumann, 1999). As shown in the figure, the shape changes from (b) to (d) of the fourth row generated by the proposed approach demonstrate a larger deformation rate than the linear interpolation method, and the shape changes from (d) to (f) of the fourth row generated by the proposed approach demonstrate a smaller deformation rate than the linear interpolation method. This feature is useful since it can be used to create special acceleration and deceleration effects (Noh and Neumann, 1999).

The CPU time of generating a facial animation is also calculated and it is found that obtaining all the new coordinate values of 14,232 polygon vertices for a facial animation of 1,389 frames (1,389 blended facial shapes) only takes one second on the same PC with Intel® Xeon® CPU E5-1650 V2 @ 3.5 GHz and 32 GB of memory. The numerical method based on the physics-based

model of soft tissues including muscle activation described in (Ichim et al., 2017) is powerful since it has the capacity to model the physical interaction of passive flesh, active muscles, and rigid bone structures and integrate collision and contact handling into the simulation. However, it is not efficient since it takes more than two minutes on a laptop with a 3.1 GHz Intel Core i7 processor and 16 GB of main memory to generate a new facial shape for a model with 6,393 surface vertices and 8,098 volumetric vertices. Although the proposed analytical approach of physics-based facial blendshapes has not the capacity, it is highly efficient.

4.3 Blending force-based animation framework

As shown in Eqs. (153-154), (158-159) and (164-165), $\boldsymbol{x}_1^{(i)}$presents a source shape, $\boldsymbol{x}_2^{(i)}$ stands for a target shape, and $\boldsymbol{x}^{(i)}$ is a blended shape. Eqs. (153), (158) and (164) are the functions of time variable *t* only. Therefore, a time slider can be implemented for each of the combinations between one source shape and each of target shapes, and all the time sliders are manipulated to achieve the force interpolation and create new blended shapes. Such a method is called slider force-based facial blendshapes. It is different from existing geometric facial blendshapes which obtain new shapes by interpolating between one source shape and one target shape. The proposed approach uses the time variable *t* in Eqs. (153), (158) and (164) to interpolate the force, which continually deforms one source shape into one or more target shapes.

A production ready blendshape model can involve 100 or more weights (Lewis et al., 2014). Manually manipulating such many weight sliders are time-consuming and can be difficult to find all or desired blended shapes. To avoid manually manipulating time sliders, exponentiation force-based and random force-based facial blendshapes will be proposed to automatically generate uniformly distributed blended shapes for animators to select from them.

The proposed blending force-based animation framework will integrate slider force-based, exponentiation force-based, and random force-based facial blendshapes, which will be introduced in Subsection 4.3.1, 4.3.2 and 4.3.3 below, respectively.

4.3.1 Slider force-based facial blendshapes

Facial blendshapes can be divided into two different types. One is between one source shape and one target shape, and the other is between one source shape and two or more target shapes. The slider force-based method is applicable to both types of facial blendshapes. In this subsection, two examples of facial blendshapes between one source shape and one target shape will be presented. The following two subsections will investigate facial blendshapes between one source shape and two or more target shapes.

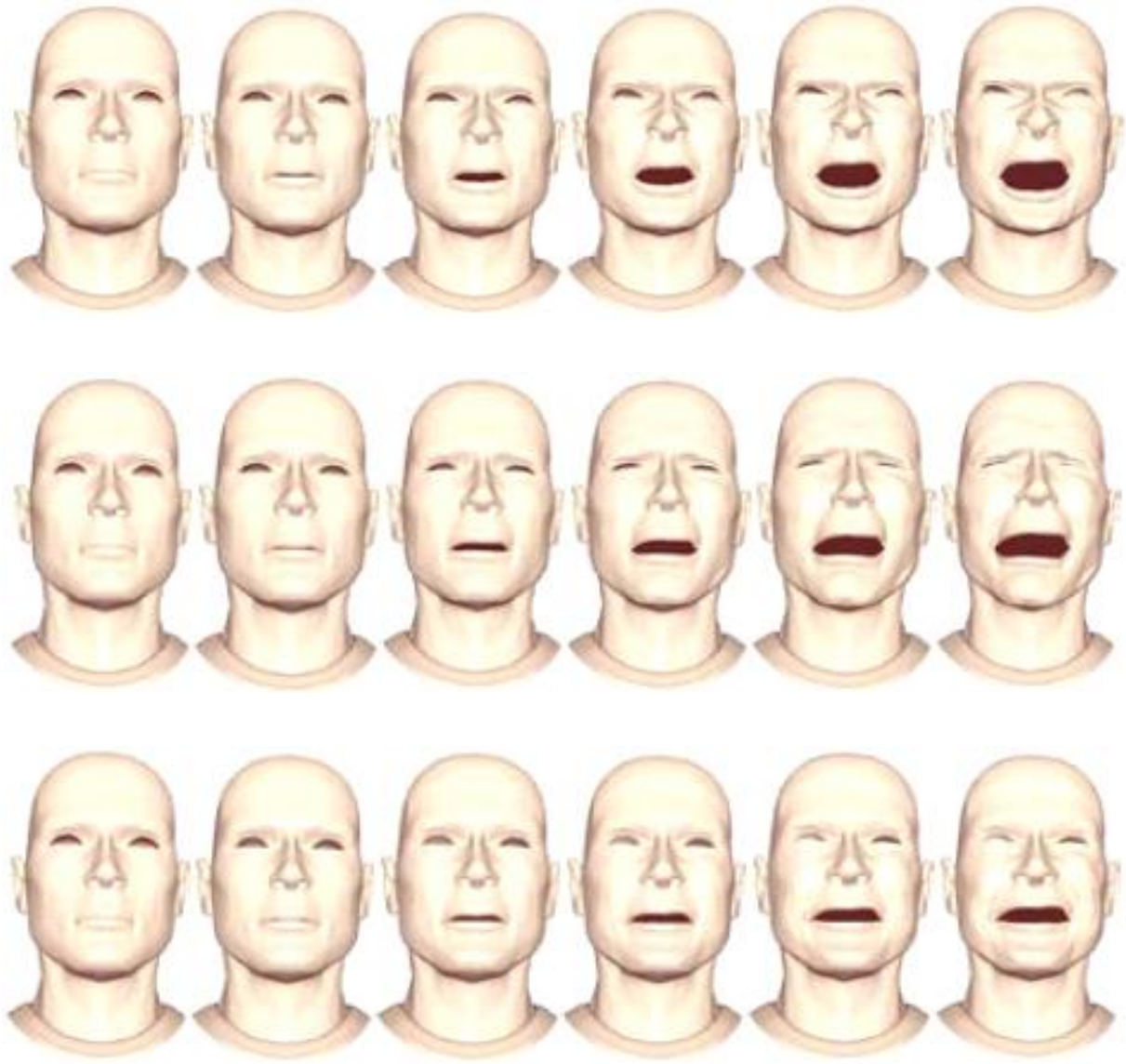
Equations (153), (158) and (164) have been implemented into time sliders to manipulate forces, which deform the source shape into the corresponding target shape. Figure 31 shows the applications of the proposed approach in creating blended shapes between one source shape (first column of Figure 31) and seven target shapes (last column).

The source shape of a human face at the neutral pose $t = 0$ and the target shapes at the deformed pose $t = 1$ obtained from the link[11] are also used to create facial blendshapes. The obtained blended shapes at the poses $t = 0.2, 0.4, 0.6 \text{ and } 0.8$ are depicted in Figure 32 where the first, second and third rows show fury, cry, and laugh expressions, respectively, and the first column shows the source shape, and the last column indicates the target shapes. These images indicate the proposed approach successfully created new blended shapes from a source and a target shape.

[11] http://people.csail.mit.edu/sumner/research/deftransfer/data.html

$t = 0$ $t = 0.2$ $t = 0.4$ $t = 0.6$ $t = 0.8$ $t = 1$

Figure 31: Blended facial shapes of a cartoon face created by using slider force-based facial blendshapes



$t = 0$ $t = 0.2$ $t = 0.4$ $t = 0.6$ $t = 0.8$ $t = 1$

Figure 32: Blended facial shapes of a human face created by using slider force-based facial blendshapes

4.3.2 Exponentiation force-based facial blendshapes

To avoid manually manipulate the implemented time sliders to generate new blended shapes for the situations with a lot of time sliders, this subsection and the following subsection will develop two methods to automatically generate blended shapes.

First, $\boldsymbol{x}_1^{(i)}$ $(i = 1, 2, 3, \dots, M)$ is used to indicate a source shape with M vertices, and $\boldsymbol{x}_j^{(i)}$ $(j = 1, 2, 3, \dots, N;\ i = 1, 2, 3, \dots, M)$ are used to indicate j^{th} target shape with M vertices where N is the total number of target shapes. By calculating the difference $\boldsymbol{x}_j^{(i)} - \boldsymbol{x}_1^{(i)}$ between each of the N target shapes $\boldsymbol{x}_j^{(i)}$ and the source shape $\boldsymbol{x}_1^{(i)}$, one force $\boldsymbol{f}_j^{(i)}$ is obtained from Equations (154), (159) and (165). In total, N forces $\boldsymbol{f}_j^{(i)}(j = 1, 2, 3, \dots, N)$ are obtained. The blended force $\boldsymbol{f}_b^{(i)}$ is obtained from the following linear combination of the N forces

$$\boldsymbol{f}_b^{(i)} = \sum_{1}^{N} w_j \boldsymbol{f}_j^{(i)}$$

$$\sum_{i}^{N} w_j = 1$$

$$(0 \le w_j \le 1) \tag{170}$$

If each of the weight w_j $(j = 1, 2, 3, \dots, N)$ is uniformly discretised into L discrete values $w_{jl} = l/(L-1)(l = 0, 1, 2, 3, \dots, L-1)$, The total weight permutations with repetition will be L^N. For example, if the left one of Figure 33 is taken to be a source shape, and the remaining 5 shapes in the same figure are taken to be target shapes, $N = 5$ is obtained. If each of the weight w_j $(j = 1, 2, 3, \dots, N)$ is discretised into $L = 3$ discrete values, $3^5 = 243$ weight permutations in total are obtained.

Here, L=3 and N=5 are used to explain how L^N is obtained. Since L=3, $w_{jl} = \frac{l}{L-1} = \frac{l}{2}(l = 0,1,2)$, i.e. $w_{jl} = 0$, $w_{j2} = 0.5$, and $w_{j2} = 1$ $(j = 1, 2, 3, 4, 5)$.

First, $w_1 = w_2 = w_3 = w_4 = 0$ are fixed. Only w_5 is changed, 3 combinations (i. e., L combinations): $w_5 = 0, w_5 = 0.5$ and $w_5 = 1$ are obtained.

Then, $w_1 = w_2 = w_3 = 0$ are fixed. For each of $w_4 = 0$, $w_4 = 0.5$, and $w_4 = 1$, 3 combinations (i. e., L combinations) are obtained. In total, 3*L combinations, i. e., $L \times L = L^2$ combinations are obtained.

Next, $w_1 = w_2 = 0$ are fixed. For each of $w_3 = 0$, $w_3 = 0.5$ and $w_3 = 1$, L^2 combinations are obtained. In total, $3 \times L^2$ combinations, i. e., $L \times L^2 = L^3$ combinations are obtained.

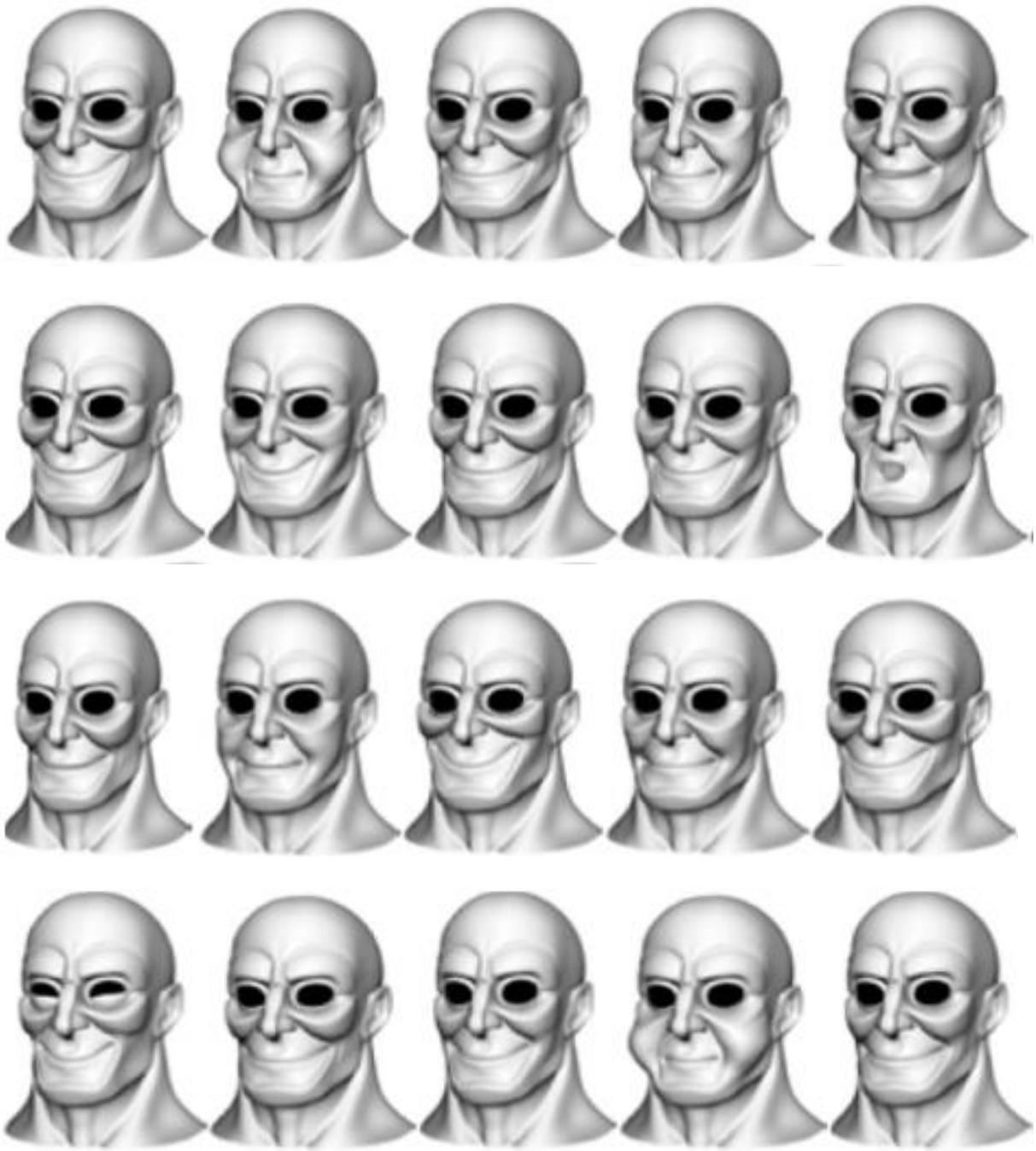
After that, $w_1 = 0$ is fixed. For each of $w_2 = 0$, $w_2 = 0$, $w_2 = 0.5$, and $w_2 = 1$, L^3 combinations are obtained. In total, $3 \times L^3$ combinations, i.e., $L \times L^3 = L^4$ combinations are obtained.

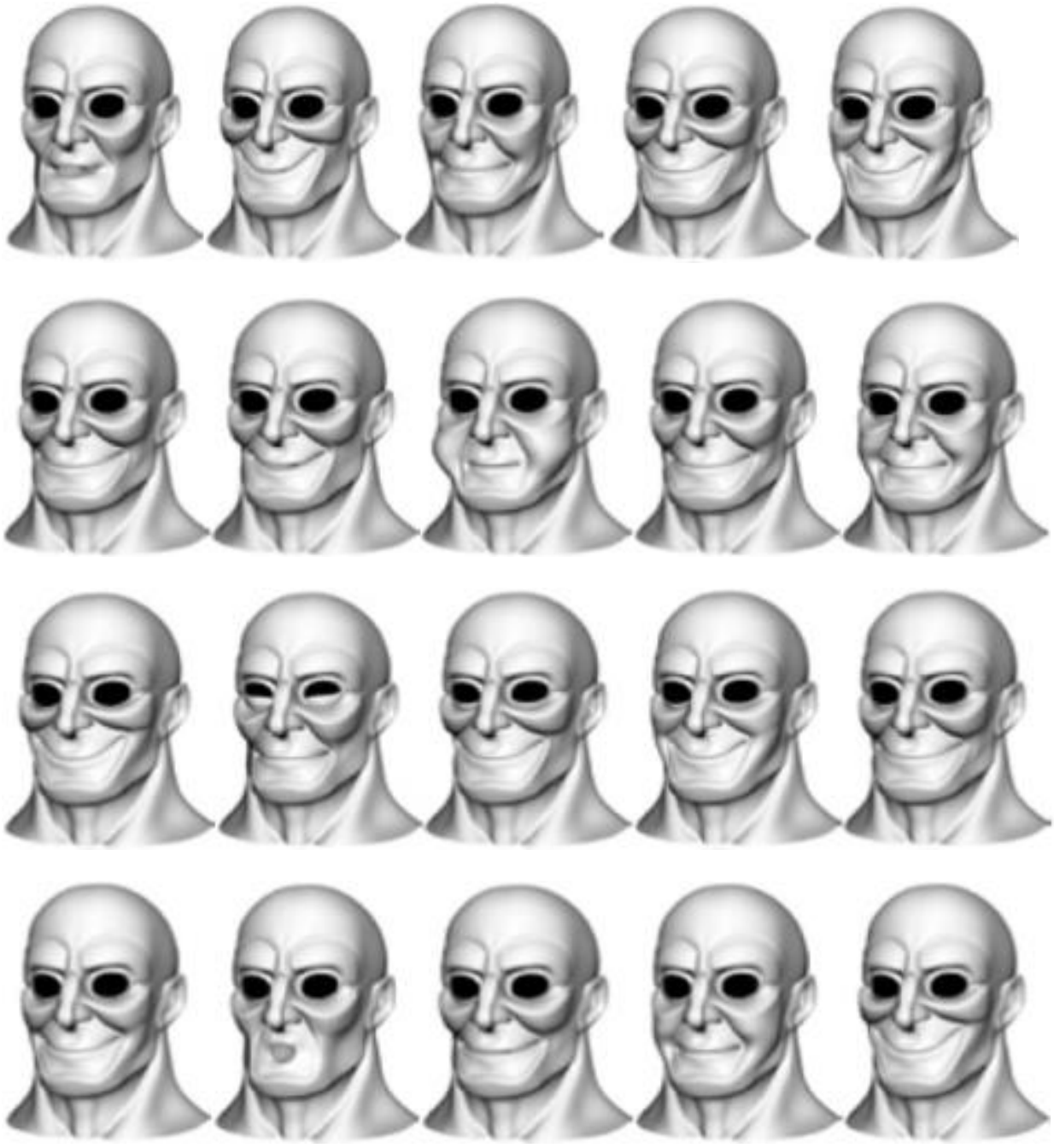
Finally, for each of $w_1 = 0$, $w_1 = 0.5$ and $w_1 = 1$, L^4 combinations are obtained. In total, $3 \times L^4$ combinations, i. e., $L \times L^4 = L^5$ combinations are obtained. Since $N = 5, L^5 = L^N$.

Figure 33: Source shape and target shapes used in exponentiation force-based facial blendshapes (Fig. 34) and random force-based facial blendshapes (Fig. 37 and 39)

Since $w_1 = w_2 = w_3 = w_4 = w_5 = 0$ leads to a zeroed blending force which will not generate any new shapes, the actual blended forces are 242 which can be used to create blended shapes at any poses in the time interval $0 \leq t \leq 1$.

At each time instant t, the 242 blended forces can be used to create 242 blended shapes. If the time instants $t = 0, 0.2, 0.4, 0.6, 0.8$ and 1 are considered, $242 \times 6 = 1{,}452$ blended shapes are obtained. Figure 34 gives the 48 blended shapes taken from every 5 ones of the created 242 blended shapes at $t = 1$.





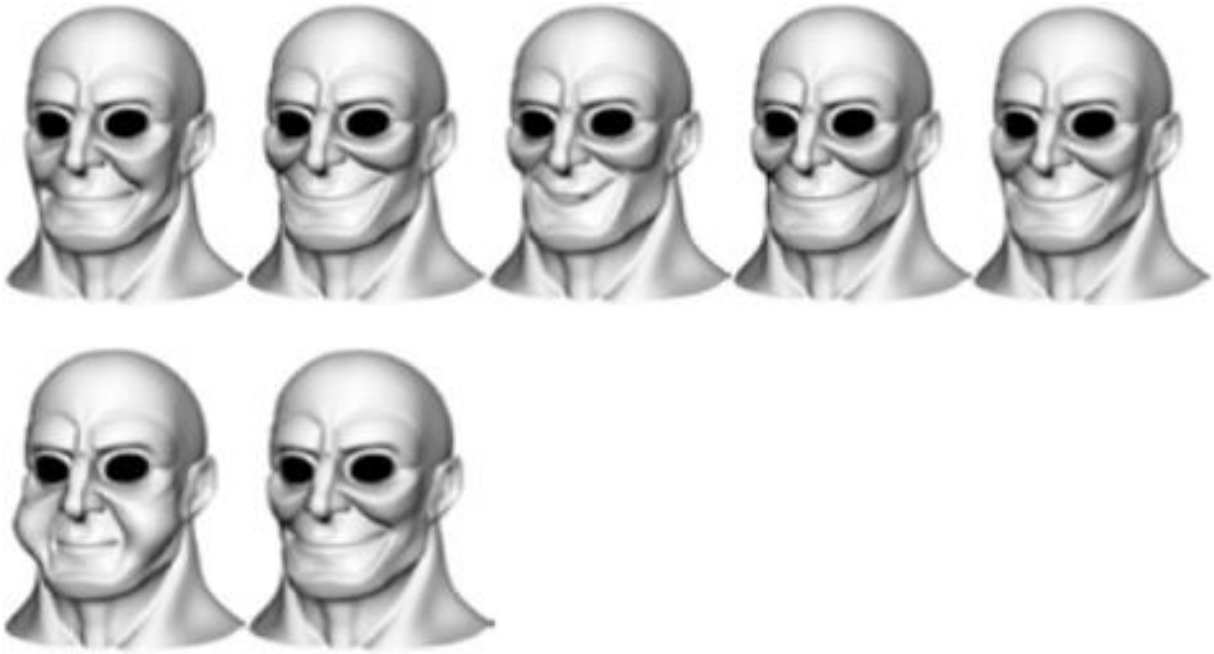
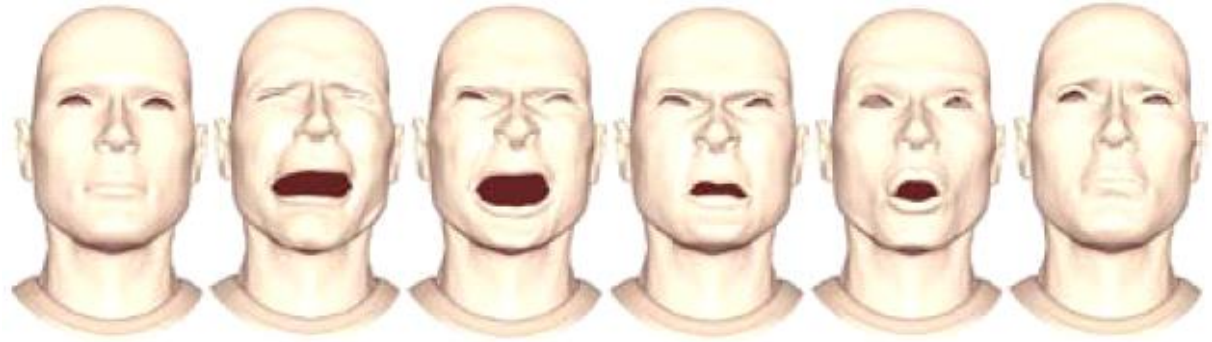


Figure 34: Blended shapes created by using exponentiation force-based facial blendshapes

For the human face shown in Figure 32, more expressions given in Figure 35 are used to demonstrate how to create new blended shapes with the approach of exponentiation force-based facial blendshapes. In Figure 35, the image from the left to the right shows the neutral, cry, fury, rage, surprise, and sad expressions, respectively.



Neutral Cry Fury Rage Surprise Sad

Figure 35: Source shape and target shapes of a human face used in exponentiation force-based facial blendshapes (Figure 36) and random force-based facial blendshapes (Figure 38 and 40)

With the same method used to generate blended shapes shown in Figure 34, the 5 facial expressions shown in Figure 35 determine 242 blended forces at each time instant, which are used to create 242 blended shapes at the time instant. If the time instants 0, 0.2, 0.4, 0.6, 0.8, and 1 are used to create blended shapes, 1,452 blended shapes in total are obtained. Figure 36 gives 24 blended shapes taken from the 1,452 blended shapes.

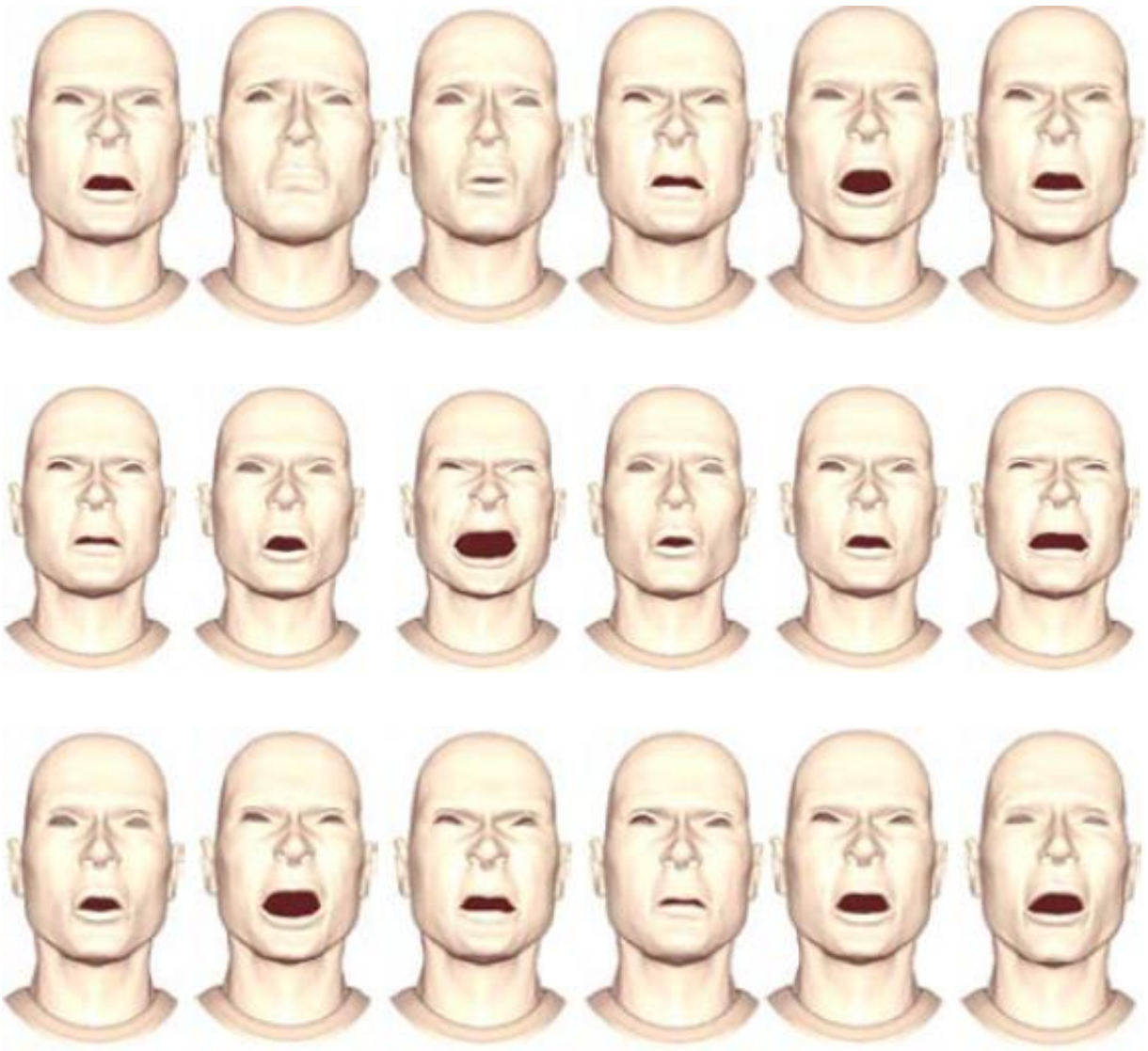


Figure 36: Blended shapes of a human face created by using exponentiation force-based facial blendshapes

The above method of determining new blended forces and creating new blended shapes involves exponentiation. It is named as exponentiation force-based facial blendshapes.

4.3.3 Random force-based facial blendshapes

The exponentiation force-based facial blendshapes give all L^N blended shapes. However, when N and L or one of them are very big, the exponentiation force-based facial blendshapes will generate a very large number of blended shapes. For example, when the discrete weights are increased from 0, 0.5 and 1 to 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0, the blended forces are increased from 242 to 161,050. If these 161,050 blended forces are used to create new blended shapes at 21 time instants t=0, 0.05, 0.1, 0.15, 0.2, ……, 0.95, and 1.0, 3,382,050 blended shapes are obtained. Viewing all 3,383,050 blended shapes and selecting some from them is a time-consuming task. This problem can be overcome by generating uniformly distributed random numbers and using the generated random numbers to create new blended shapes for animators to view and select from them. In what follows, two methods will be proposed to deal with this.

4.3.3.1 First method of random force-based facial blendshapes

If there are N target shapes, the weights will be $w_1, w_2, w_3, \dots, w_n$ according to Eq. (170). All possible combinations of the *N* weights are all the permutations of 1, 2, 3, …, N respectively taken from the N numbers 1, 2, 3,…, *N*. The total number of all the permutations is $\overline{N} = \sum_{r=1}^{N} N_{P^r}$ where $N_{P^r} = \frac{N!}{(N-r)!}$. All the permutations are kept in a 2D array $Per[n_1][n_2]$ ($1 \leq n_1 \leq \overline{N}; 1 \leq n_2 \leq N;$) For example, when *N*=5, the total number of all the permutations will be $\overline{N} = \sum_{r=1}^{N} N_{P^r} = \sum_{r=1}^{N} \frac{N!}{(N-r)!} = \frac{5!}{(5-1)!} + \frac{5!}{(5-2)!} + \frac{5!}{(5-3)!} + \frac{5!}{(5-4)!} + \frac{5!}{(5-5)!} = 5 + 20 + 60 + 120 + 120 = 325$, the permutations of 1 taken from the 5 numbers: 1, 2, 3, 4, 5 will be kept as Per[1][]={1, 0, 0, 0, 0}, Per[2][]={0, 2, 0, 0, 0},…, Per[5][]={0, 0, 0, 0, 5}, the permutations of 2 taken from the 5 numbers will be kept as Per[6][]={1, 2, 0, 0, 0},…, Per[25][]={5, 4, 0, 0, 0}, and the last permutation of 5 taken from the total 5 numbers will be kept as Per[325][]={5, 4, 3, 2, 1}.

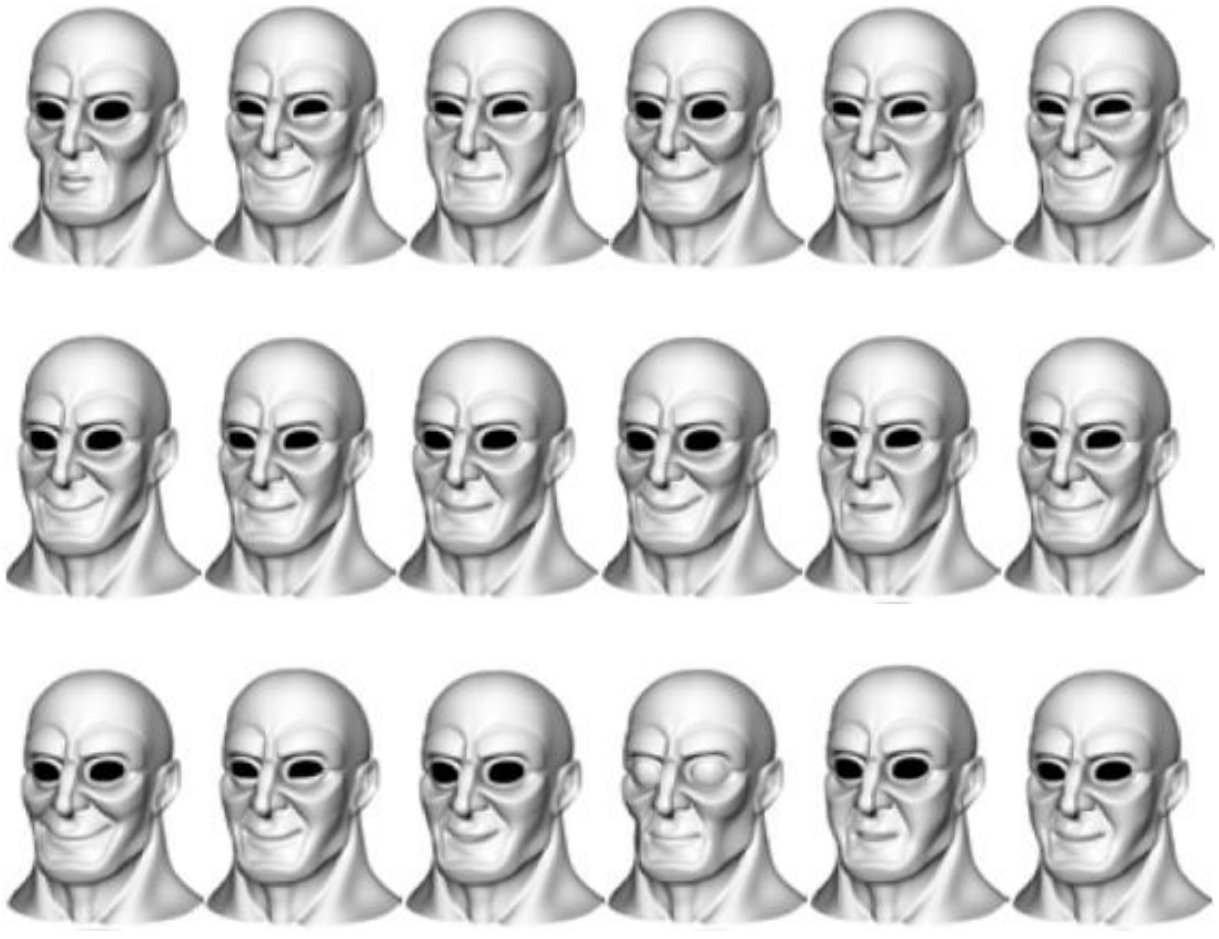
From the source shape and the *N* target shapes, the forces $f_j^{(i)} (j = 1, 2, 3, \dots, N)$ can be obtained. If it is required to select L weight combinations from the total $\overline{N}$ weight combinations to create new blended shapes, L uniformly distributed random numbers within the range between 1 and $\overline{N}$ are generated, and the generated L random numbers are used to identify which weight combinations should be used.

Taking *N*=5 as an example, if the generated random numbers contain 2, 25, and 325, the weight combinations Per[2][]={0, 2, 0, 0, 0}, Per[25][]={5, 4, 0, 0, 0,}, and Per[325][]={5, 4, 3, 2, 1} are identified and converted into Per[2][]= {0, 2/2, 0, 0, 0}={0, 1, 0, 0, 0}, Per[25][]= {5/(5+4), 4/(5+4), 0, 0, 0} = {5/9, 4/9, 0, 0, 0} and Per[325][] ={5/(5+4+3+2+1), 4/(5+4+3+2+1), 3/(5+4+3+2+1), 2/(5+4+3+2+1), 1/(5+4+3+2+1)}={5/15, 4/15, 3/15, 2/15, 1/15}.

Introducing each of the three weight combinations into Equation (170), the three blended forces are: $f_2^{(i)}$, $\left[5f_1^{(i)} + 4f_2^{(i)}\right]/9$, and $\left[5f_1^{(i)} + 4f_2^{(i)} + 3f_3^{(i)} + 2f_4^{(i)} + 4f_5^{(i)}\right]/15$, respectively. Substituting each of the three obtained blended forces into Eqs. (153), (158) and (164), the new blended shapes between $t = 0$ and $t = 1$ can be created.

If taking L = 50 and using the source shape and the 5 target shapes shown in Figure 33, which gives N=5, 50 new blended shapes at each of t values are obtained from 50 blended forces. Figure 30 gives the first 48 ones of the 50 new blended shapes at the pose t=1.

.



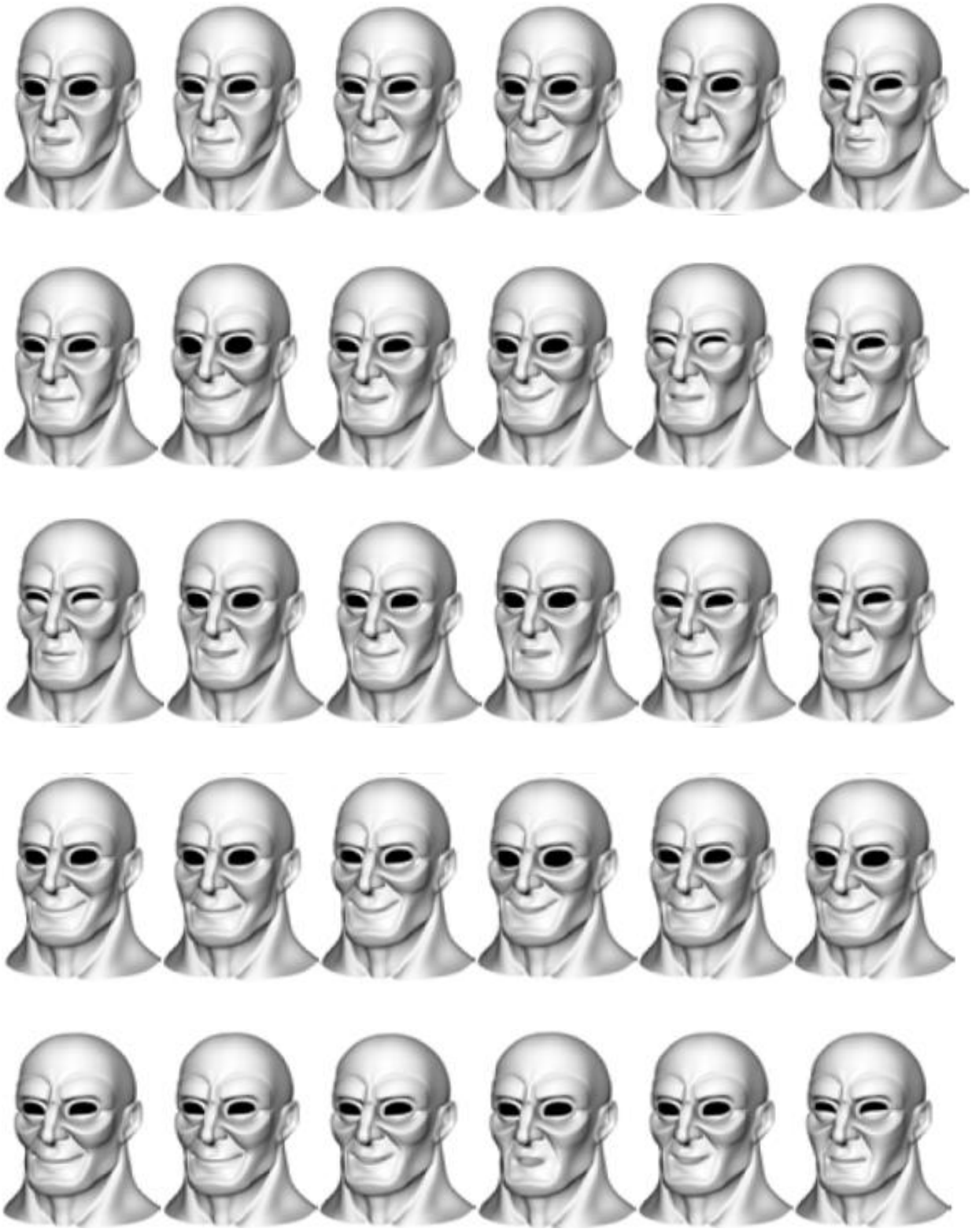


Figure 37: Blended shapes created by using the first method of random force-based facial blendshapes

Still taking L = 50 but using the source (neutral) shape and the 5 target shapes shown in Figure 35, which gives N=5, 50 new blended shapes at each of t values from 50 blended forces can be created. Figure 38 gives 24 blended shapes of the 50 new blended shapes at the time instant $t = 0.8$.

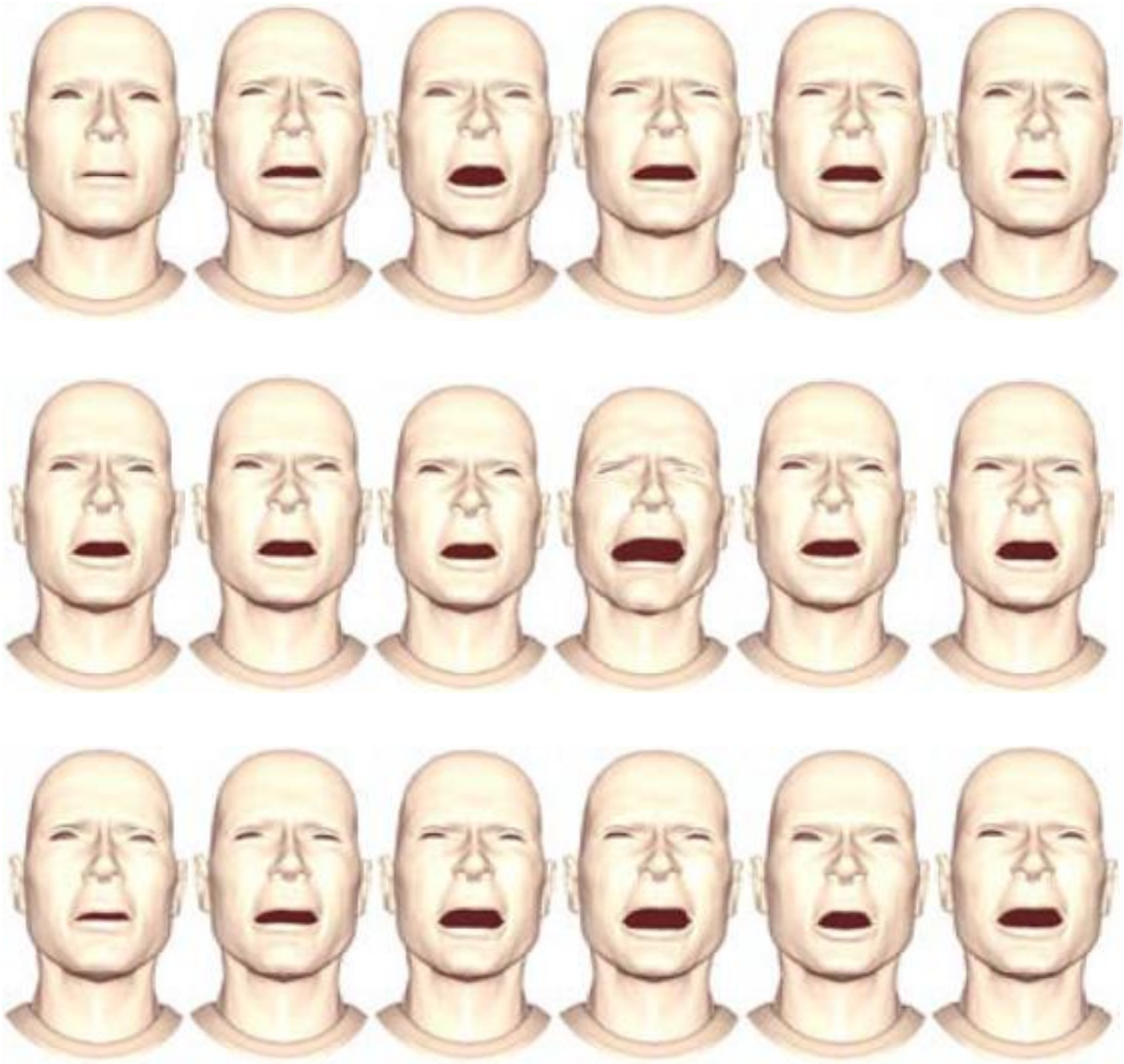


Figure 38: Blended shapes of a human face created by using the first method of random force-based facial blendshapes

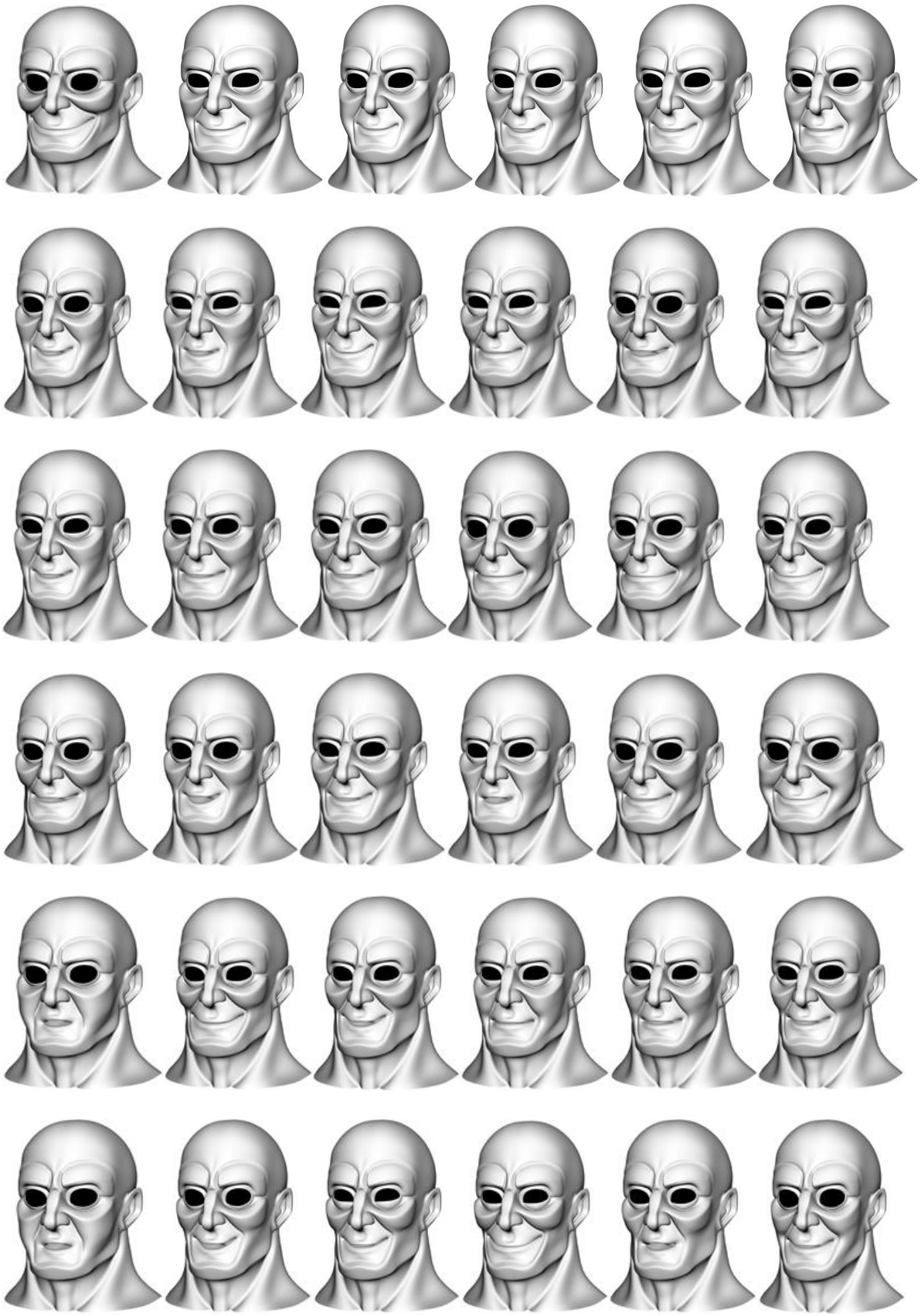
4.3.3.2 Second method of random force-based facial blendshapes

If there are N target shapes and one source shape, N force $\boldsymbol{f}_j^{(i)} (j = 1, 2, 3, \dots, N)$ can be obtained. The weight for each of the N forces is discretized into L discrete values. $M = L^N$ permutations are the total number of all possible weight combinations which will lead to L^N blended forces. Taking $N = 5$ and $L = 11$ as an example, $M = L^N = 11^5 = 161{,}051$ weight combinations are obtained. Here, $w_{1,i} = i \times dw \ \ (i = 0, 1, 2, \dots, 10)$ is used to indicate the i^{th} weight of the first force, , $w_{2,j} = j \times dw \ (j = 0, 1, 2, \dots, 10)$ is used to indicate the j^{th} weight of the second force, $w_{3,k} = k \times dw \ \ (k = 0, 1, 2, \dots, 10)$ is used to indicate the k^{th} weight of the third force, $w_{4,l} = l \times dw \ (l = 0, 1, 2, \dots, 10)$ is used to indicate the l^{th} weight of the fourth force, and $w_{5,m} = m \times dw (m = 0, 1, 2, \dots, 10)$ is used to indicate the m^{th} weight of the fifth force where $dw = \frac{1.0}{10} = 0.1$. if n^{th} is used to indicate the index of the weight combination, $n = \{[(11 \times i + j) \times 11 + k] \times 11 + l\} \times 11 + m + 1$, and the five weights for the n^{th} weight combination can be kept in a two-dimensional array Per[0-161,050][0-4] as Per[n][0]=$w_{1,i}$, Per[n][1]=$w_{2,j}$, Per[n][2]=w_{3k}, Per[n][3]=$w_{4,l}$, and Per[n][4]=$w_{5,m}$ ($i,\ j,\ k,\ l,\ m = 0, 1, 2, 3, \dots, 10;\ n = \{[(11 \times i + j) \times 11 + k] \times 11 + l\} \times 11 + m + 1$). Here, the

first subscript indicates the index of the five forces. For example, $w_{1,i}$ is the i^{th} weight of the first force.

Creating and visualising all the $161{,}051$ blended shapes is tedious and time-consuming. To tackle this problem, uniformly distributed random numbers are introduced to create new blended shapes. If $\tilde{L}$ blended shapes at the time $t = t_0$ are to be viewed among the blended shapes created from the L^N blended forces, $\tilde{L}$ uniformly distributed random numbers are generated and used to determine $\tilde{L}$ weight combinations. Then, the $\tilde{L}$ weight combinations are used to determine $\tilde{L}$ blended forces, which are used to create $\tilde{L}$ blended shapes at $t = t_0$. For example, if an obtained random number is 19,337, it corresponds to $i = 1$, $j = 3$, $k = 5$, $l = 8$, and $m = 9$ since $n = \{[(11 \times i + j) \times 11 + k] \times 11 + l\} \times 11 + m + 1 = \{[(11 \times 1 + 3) \times 11 + 5] \times 11 + 8\} \times 11 + 9 + 1 = 19{,}337$. Accordingly, the weight values $w_{1,1} = 1 \times dw = 1 \times 0.1 = 0.1$ from $w_{1,i} = i \times dw$ and $i = 1$, $w_{2,3} = 3 \times dw = 0.3$, $w_{3,5} = 5 \times dw = 0.5$, $w_{4,8} = 8 \times dw = 0.8$, and $w_{5,9} = 9 \times dw = 0.9$ have already been kept in Per[19,337][0]=0.1, Per[19,337][1]= 0.3 , Per[19,337][2]= 0.5 , Per[19,337][3]= 0.8 , and Per[19,337][4]= 0.9 . Therefore, the value of the five weights can be determined from the obtained random number 19,337. Since the sum of the five weights is not equal to 1.0, the sum of the five weights is calculated, which is 2.6, and each of the five weights is divided by the sum. After that, the normalized weights are used to obtain the blended force $\boldsymbol{f}_b^{(i)} = \frac{1}{2.6}\left[0.1\boldsymbol{f}_1^{(i)} + 0.3\boldsymbol{f}_2^{(i)} + 0.5\boldsymbol{f}_3^{(i)} + 0.8\boldsymbol{f}_4^{(i)} + 0.9\boldsymbol{f}_5^{(i)}\right]$, which corresponds to the obtained random number 19,337.

Taking $\tilde{L} = 50$, the above method was used to create 50 blended shapes at the time t =1.0 from the chosen 50 blended forces. Figure 39 shows 48 blended shapes taken from the 50 blended shapes.



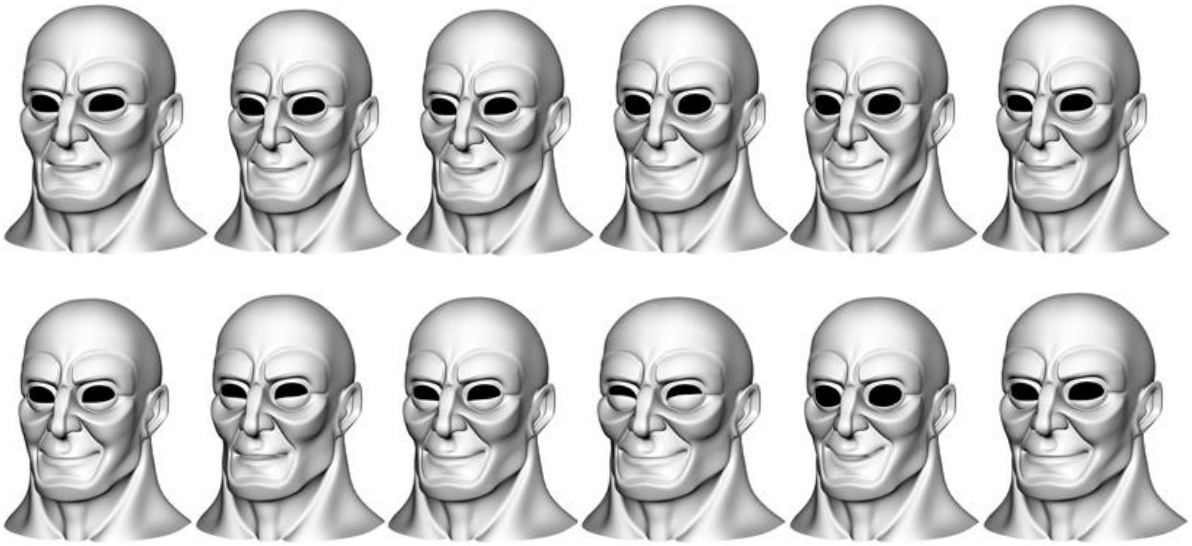


Figure 39: Blended shapes created by using the second method of random force-based facial blendshapes

Still taking $\tilde{L} = 50$ but using the source (neutral) shape and the 5 target shapes shown in Figure 35, N=5 is obtained. New blended shapes from 50 blended forces at any time instants can be created. Figure 40 shows 24 blended shapes taken from the created 50 new blended shapes at the time instant $t = 1.0$.

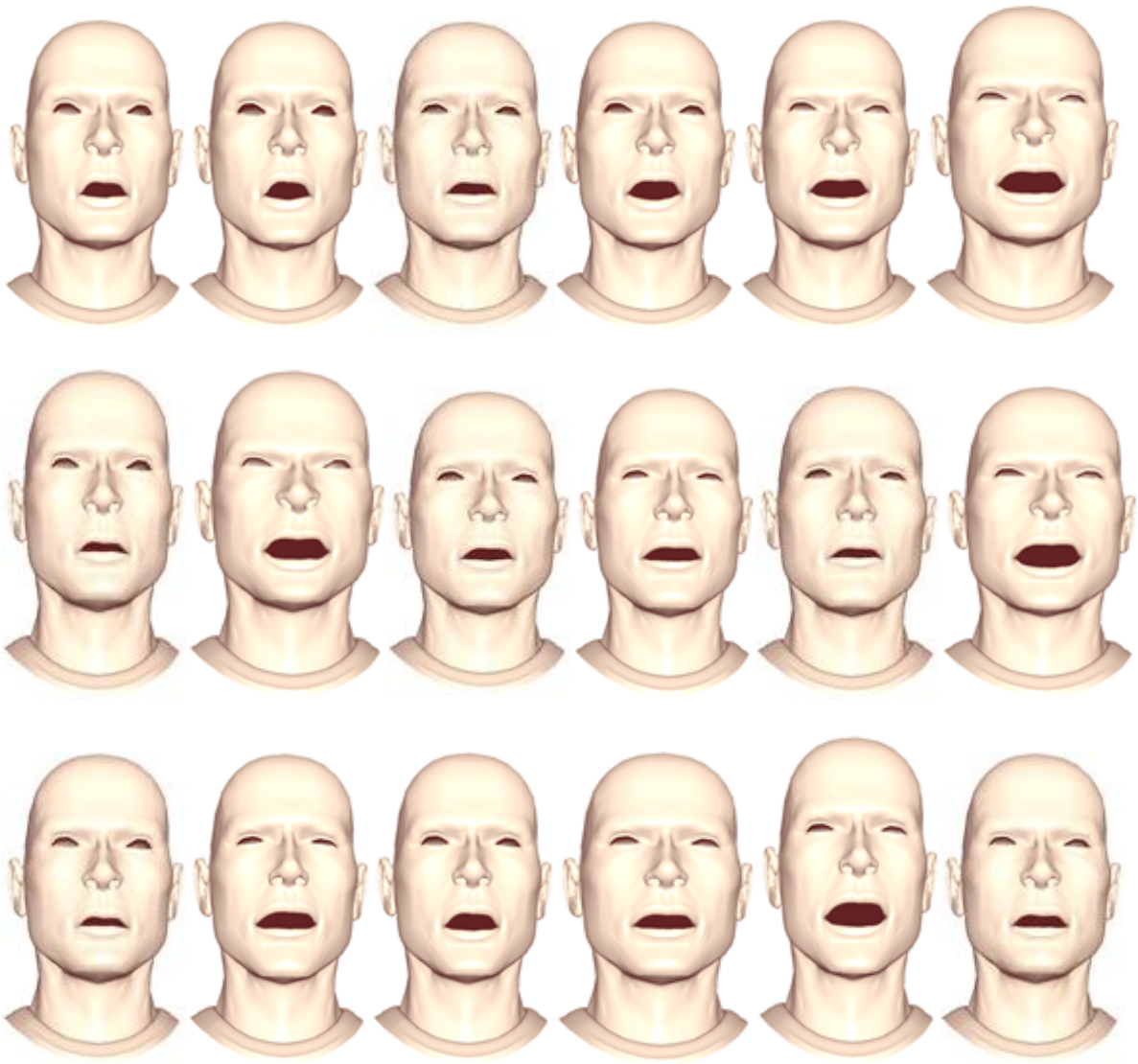


Figure 40: Blended shapes of a human face created by using the second method of random force-based facial blendshapes

4.4 Conclusions and future work

In this chapter, I have developed a novel approach of physics-based facial blendshapes. The approach is based on the analytical solution to the mathematical model integrating the equation of motion and the constraints of source and target facial shapes. It is used to develop an efficient facial animation framework which integrates slider force-based, exponentiation force-based, and random force-based facial blendshapes.

The experiments made in this thesis indicate that the proposed analytical physics-based approach and blending force-based animation framework can create a superset of blended shapes which not only contains the blended shapes generated by linear interpolation, but also those which cannot be generated by linear interpolation. They can create blended shapes with different deformation rates and overcome the limitation of linear interpolation which can only generate blended shapes with a same deformation rate. The analytical equations and their use in facial blendshapes demonstrate simplicity, easiness, low computer capacity requirement, and very high computational efficiency of creating facial animation.

The proposed approach offers a rich set of opportunities for future research. First, the influences of facial muscle, other facial tissues, and rigid bone structures on facial deformations can be introduced into the proposed approach. As elaborated in Section 5.2, this can be achieved by transforming their influences into external forces to be included in the right-hand-side force functions. Second, the physical parameters, i.e., mass, damping coefficient and stiffness coefficient can be estimated to create realistic facial deformations. As detailed in Section 5.2, this can be achieved by minimizing the differences between the captured real facial shapes and the blended facial shapes created with the solution to the ordinary differential equation (143). Third, the developed analytical solutions are used as a shape blending tool in this thesis. Actually, they can be also extended as a shape manipulation tool. As explained in Section 5.2, this can be achieved by using the force functions as sculpting forces. The sculpting forces can be taken to be constants, linear variations, or nonlinear variations and the corresponding solutions can be derived and added to the general solutions to develop powerful physics-based shape manipulation handles and create different deformation effects.

5. Summary and future work

This section summarizes the work presented in this thesis. It draws the conclusions from the work completed by this PhD research, and discusses the future work.

5.1 Conclusion

Through the work presented in this thesis, new differential equation-based shape interpolation techniques have been developed and applied in time-dependent and time-independent surface blending of C^2 continuity and facial blendshapes. These new developed techniques tackle the problems of existing differential equation-based surface blending and physics-based facial blendshapes.

For the new techniques of differential equation-based surface blending with C^2 continuity, a new mathematical model consisting of a vector-valued sixth-order partial differential equation and blending boundary constraints was proposed to unify both time-dependent and time-independent surface blending of C^2 continuity, and three approximate analytical approaches were derived to achieve time-dependent and time-independent surface blending of C^2 continuity easily and efficiently. The comparisons between the approximate analytical approaches and the corresponding accurate closed form solution demonstrate that the developed approximate analytical approaches have good accuracy and high computational efficiency. How the second partial derivatives affect the continuity at trimlines was also investigated. It was found that the same second partial derivatives between the blending and primary surfaces guarantee good continuity at the trimlines, but different second partial derivatives between the blending and primary surfaces cause discontinuity at the trimlines. The developed three approximate analytical approaches were applied to blend time-independent constant surfaces whose positions and shapes do not change with the time and time-dependent

varying surfaces. And the developed variable decomposition-based approximate analytical approach and differential property-based approximate analytical approach were applied to create C^2 continuous blending between both time-dependent and time-independent parametric surfaces. The applications demonstrate the developed approximate analytical approaches are effective and efficient in dealing with C^2 continuous blending of both constant and varying parametric surfaces.

For the new technique of differential equation-based facial blendshapes, a novel mathematical model consisting of the equation of motion and the constraints of source and target facial models was formulated and the analytical solution to the mathematical model was developed. It was used to develop an efficient facial animation framework, which integrates slider force-based, exponentiation force-based, and random force-based facial blendshapes. The experiments carried out in this thesis indicate that the proposed analytical physics-based approach and blending force-based animation framework can create a superset of blended shapes, which not only contains the blended shapes generated by linear interpolation, but also those which cannot be generated by linear interpolation. They can create blended shapes with different deformation rates and overcome the limitation of linear interpolation which can only generate blended shapes with a same deformation rate. Quite a number of examples were presented to demonstrate the applications of the developed technique. These application examples demonstrate that the developed technique is simple and has low requirements for computer resources and very high computational efficiency in generating a lot of facial blendshapes and creating facial animation.

5.2 Further work

The new approaches developed in this thesis not only have a lot of applications in C^2 continuous blending between time-dependent and time-independent parametric surfaces and physics-based facial blendshapes, but also provide many new opportunities for future research studies.

As discussed in Section 1.1, both shape control of blending surfaces and exact satisfaction of blending boundary constraints without heavy numerical minimization calculations has not been addressed. The three new developed surface blending approaches involve four shape control parameters. They can effectively change the shapes of blending surfaces but still well maintain the required continuities between the blending surface and primary surfaces. This advantage can be developed into a useful shape control handle. With the developed shape control handle, the user can achieve their preferred shapes of blending surfaces and required continuities between the blending surface and primary surfaces. Stress concentration is a main problem in practical engineering problems. It is related to the curvature of blending surfaces. Small curvature, i. e. large curvature radius, causes low stress concentration. How to minimize the stress concentration in blending surfaces has not been solved in the existing surface blending approaches. An optimization method can be developed to find the optimal shape control parameters, which lead to the minimum curvature of blending surfaces and minimize stress concentration in blending surfaces. Aesthetic requirements are another aspect of blending surface design, which can be met by creating user's specified shapes through one or more profiles drawn by users. Once these profiles have been drawn, the difference between the user's drawn profiles and the corresponding ones of blending surfaces can be formulated, and a minimization algorithm can be developed to minimize the difference for obtaining the optimal shape control parameters.

The developed differential equation-based facial blendshapes can be extended in many aspects. Here, I list three aspects and explain them below.

First, more complicated facial models can be investigated. These facial models not only include facial skin, but also rigid bone structures and other facial tissues such as facial muscles. The influences of these bone structures and facial tissues can be considered by transforming them into external forces and including them in the right-hand-side force functions. This can be achieved by combining the finite element method with the method extended from Section 4.1. The finite element method can be used to simulate deformations of a finite element model consisting of skin, facial muscle, other tissues, and rigid bones. Since the finite element simulation involves heavy numerical calculations, a large time step can be used to reduce the time of numerical calculations and obtain the deformed shapes at the key poses defined by the large time step. Then, the forces are taken to be an unknown function of the time variable t and the particular solution of the ordinary differential equation (143) is obtained with the method of undetermined coefficients. All the unknown constants involved in the solution of the ordinary differential equation (143) are determined by minimizing the differences between the deformed shapes obtained with the finite element simulation and those described with the solution of the ordinary differential equation (143). Once all the unknown constants in the solution of the ordinary differential equation (143) are obtained, the solution is used to generate new skin shapes, which have not been obtained with the finite element simulation. With such a combination, the time used in finite element simulation can be greatly reduced and the influences of facial muscle, other facial tissues, and rigid bone structures on facial deformations can be considered.

Second, a data-driven technique can be developed from the method proposed in Section 4.1 and used to estimate the physical parameters, i. e., mass m, damping coefficient c, and stiffness coefficient k for more realistic facial animation. This can be achieved by minimizing the differences between the captured real facial shapes and the blended facial shapes generated with Eqs. (153), (158), and (164). With this method, facial shapes at the time instants t_j ($j =$

$0, 1, 2, \ldots, J+1$) are captured and reconstructed. The reconstructed facial shapes are indicated with $x_j^{(i)}$ ($j = 0, 1, 2, \ldots, J+1$). As discussed in Section 4.1, $x_0^{(i)}$ and $x_{J+1}^{(i)}$ are used to obtain the solutions (153), (158), and (164). Taking the solution (164) as an example, the solution involves q_3 and q_4, which are the function of mass m, damping coefficient c, and stiffness coefficient k. At the time instants t_j ($j = 1, 2, \ldots, J$) , the facial shapes determined by Eq. (164) are $x^{(i)}(t_j)$ ($j = 1, 2, \ldots, J$). The squared error sum between the reconstructed facial shapes $x_j^{(i)}$ and the facial shapes $x^{(i)}(t_j)$ determined by Eq. (164) is $E = \sum_{j=1}^{J}\left[x^{(i)}(t_j) - x_j^{(i)}\right]^2$. The mass m, damping coefficient c, and stiffness coefficient k can be determined by minimizing the squared error sum E, i. e., solving the equations $\partial E/\partial m = 0$, $\partial E/\partial c = 0$, and $\partial E/\partial k = 0$.

Third, the sculpting forces involved in the solution of the ordinary differential equation (143) can be used as a shape manipulation handle. In Section 4.1, the sculpting forces are determined by the facial shapes at the neutral pose and target pose. Instead of this, the sculpting forces in the ordinary differential equation (143) can be input by users. These sculpting forces can be time-independent (constant) and time-dependent (linear variations and nonlinear variations with the time variable t). These sculpting forces can be expanded into a unified mathematical expression, i. e., Fourier series. The solution of the ordinary differential equation (143) can be obtained from the unified mathematical expression. When users select a position to apply for a sculpting force, the information is fed into the solution to create a new facial shape. By selecting different positions and applying different sculpting forces, different facial shapes are obtained to realize shape manipulation.

References

Adzhiev, V., Tereshin, A., Anderson, E.F. and Pasko, A., 2020. Space-time blending for heterogeneous objects. In: *Eurographics 2020 Short Paper* [online], pp. 45-48, 25-29 May 2020, Norrköping, Sweden.

Akhter, I., Simon, T., Khan, S., Matthews, I. and Sheikh, Y., 2012. Bilinear spatiotemporal basis models. *ACM Transactions on Graphics* [online], 31(2), pp. 1-12.

Allen, S. and Dutta, D., 1997a. Cyclides in pure blending I. *Computer Aided Geometric Design* [online], 14(1), pp. 51-75.

Allen, S. and Dutta, D., 1997b. Cyclides in pure blending II. *Computer Aided Geometric Design* [online], 14(1), pp. 77-102.

Allen, S. and Dutta, D., 1997c. Supercyclides and blending. *Computer Aided Geometric Design* [online], 14(7), pp. 637-651.

Angles, B., Tarini, M., Wyvill, B., Barthe, L. and Tagliasacchi, A., 2017. Sketch-based implicit blending. *ACM Transactions on Graphics* [online], 36(6), pp. 1-13.

Aumann, G., 1995. Curvature continuous connections of cones and cylinders. *Computer-Aided Design* [online], 27(4), pp. 293-301.

Bajaj, C.L. and Ihm, I., 1992. Algebraic surface design with Hermite interpolation. *ACM Transactions on Graphics* [online], 11(1), pp. 61-91.

Barrielle, V., Stoiber, N. and Cagniart, C., 2016. BlendForces: A Dynamic Framework for Facial Animation. *Computer Graphics Forum* [online], 35(2), pp. 341-352.

Barrielle, V. and Stoiber, N., 2019. Realtime Performance-Driven Physical Simulation for Facial

Animation. *Computer Graphics Forum* [online], 38(1), pp. 151-166.

Bergeron, P., Lachapelle, P., 1985. Controlling facial expressions and body movements in the computer generated animated short 'Tony de Peltrie'. In: *SIGGRAPH '85 Tutorial Notes*, Advanced Computer Animation Course.

Bickel, B., Kaufmann, P., Skouras, M., Thomaszewski, B., Bradley, D., Beeler, T., Jackson, P., Marschner, S., Matusik, W. and Gross, M., 2012. Physical face cloning. *ACM Transactions on Graphics* [online], 31(4), pp. 1-10.

Bizzarri, M. and Lávička, M., 2015. On modeling with rational ringed surfaces. *Computer-Aided Design* [online], 58, pp. 151-161.

Bizzarri, M., Lávička, M. and Kosinka, J., 2016. Medial axis transforms yielding rational envelopes. *Computer Aided Geometric Design* [online], 46, pp. 92-102.

Bizzarri, M., Lávička, M. and Kosinka, J., 2017. Skinning and blending with rational envelope surfaces. *Computer-Aided Design* [online], 87, pp. 41-51.

Bloor, M.I.G and Wilson, M.J., 1989. Generating blend surfaces using partial differential equations. *Computer-Aided Design*, 21(3), pp. 165-171.

Bloor, M.I.G and Wilson, M.J., 1996. Spectral approximations to PDE surfaces. *Computer-Aided Design* [online], 28(2), pp. 145-152.

Bloor, M.I.G, Wilson, M.J. and Hagen, H., 1995. The smoothing properties of variational schemes for surface design. *Computer Aided Geometric Design* [online], 12(4), pp. 381-394.

Bloor, M.I.G, Wilson, M.J. and Mulligan, S.J., 2000. Generating blend surfaces using a perturbation method. *Mathematical and Computer Modelling* [online], 31(1), pp. 1-13.

Booth, J., Antonakos, E., Ploumpis, S., Trigeorgis, G., Panagakis, Y. and Zafeiriou, S., 2017. 3D

face morphable models “in-the-wild”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* [online], pp. 48-57.

Booth, J., Roussos, A., Ververas, E., Antonakos, E., Ploumpis, S., Panagakis, Y. and Zafeiriou, S., 2018. 3D Reconstruction of “In-the-Wild” Faces in Images and Videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online], 40(11), pp. 2638-2652.

Brown, J.M., Bloor, M.I.G., Bloor, M.S. and Wilson, M.J., 1998. The accuracy of B-spline finite element approximations to PDE surfaces. *Computer Methods in Applied Mechanics and Engineering* [online], 158(3-4), pp. 221-234.

Carrigan, E., Zell, E., Guiard, C. and McDonnell, R., 2020. Expression Packing: As-Few-As-Possible Training Expressions for Blendshape Transfer. *Computer Graphics Forum* [online], 39(2), pp. 219-233.

Carrigan, E., Zibrek, K., Dahyot, R. and McDonnell, R., 2022. Investigating perceptually based models to predict importance of facial blendshapes. In: *Proceedings of the 13th Annual ACM SIGGRAPH Conference on Motion, Interaction, and Games* [online]. Association for Computing Machinery, Article No.: 2, pp. 1-6.

Cetinaslan, O. and Orvalho, V., 2020a. Sketching Manipulators for Localized Blendshape Editing. *Graphical Models* [online], 108, p.101059.

Cetinaslan, O. and Orvalho, V., 2020b. Stabilized blendshape editing using localized Jacobian transpose descent. *Graphical Models* [online], 112, p.101091.

Chen, K., Zheng, J., Cai, J. and Zhang, J., 2020. Modeling caricature expressions by 3D blendshape and dynamic texture. In: *Proceedings of the 28th ACM International Conference on Multimedia* [online]. Association for Computing Machinery, pp. 3228-3236.

Cheng, S.Y., Bloor, M.I.G., Saia, A. and Wilson, M.J., 1990. Blending between quadric surfaces using partial differential equations. In: Ravani B (ed) *Advances in design automation*, vol 1. Computer and Computational Design, ASME Press, pp. 257-263.

Cheng, S., Kotsia, I., Pantic, M. and Zafeiriou, S., 2018. 4DFAB: A large scale 4D database for facial expression analysis and biometric applications. In: *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018)* [online]. Salt Lake City, Utah, US.

Choi, B.K. and Ju, S.Y., 1989. Constant-radius blending in surface modelling. *Computer-Aided Design*, 21(4), pp. 213-220.

Chuang, J.H. and Hwang, W.C., 1997. Variable-radius blending by constrained spine generation. *The Visual Computer* [online], 13(7), pp. 316-329.

Chuang, J.H. and Lien, P.L., 1998. One and two-parameter blending for parametric surfaces. *Journal of Information Science and Engineering*, 14, pp. 461-477.

Chuang, J.H., Lin, C.H. and Hwang, W., 1995. Variable-radius blending of parametric surfaces. *The Visual Computer*, 11(10), pp. 513-525.

Cong, M., Fedkiw, R., Bhat, K.S., and Hanrahan, P., 2016. Art-directed muscle simulation for high-end facial animation. In: *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* [online], pp. 119-127.

Cong, M., Lan L. and Fedkiw, R., 2017. Muscle simulation for facial animation in Kong: Skull Island. In: *ACM SIGGRAPH 2017 Talks* [online], (New York, NY, USA), SIGGRAPH ’17, ACM, pp. 21:1-21:2.

Cong, M. and Fedkiw, R., 2019. Muscle-based facial retargeting with anatomical constraints. In: *ACM SIGGRAPH 2019 Talks* [online], (New York, NY, USA, 2019), SIGGRAPH’19, ACM, pp. 5:1-5:2.

Dahl, H.E.I and Krasauskas, R., 2012. Rational fixed radius rolling ball blends between natural quadrics. *Computer Aided Geometric Design* [online], 29(9), pp. 691-706.

DeRose, T., Kass, M. and Truong, T., 1998. Subdivision surfaces in character animation. In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 98)* [online]. pp. 85-94.

Dinev, D., Guo, W., Kadleček, P. and Kavan, L., 2020. Solving for muscle blending using data. *Computers & Graphics* [online], 92, pp. 67-75.

Farouki, R.E.M. and Sverrisson, R., 1996. Approximation of rolling-ball blends for free-form parametric surfaces. *Computer-Aided Design* [online], 28(11), pp. 871-878.

Faux, I.D. and Pratt, M.J., 1979. *Computational geometry for design and manufacture*. Chichester: Ellis Hoewood.

Filip, D.J., 1989. Blending parametric surfaces. *ACM Transactions on Graphics* [online], 8(3), pp. 164-173.

Golec, K., Palierne, J., Zara, F., Nicolle, S. and Damiand, G., 2020. Hybrid 3D mass-spring system for simulation of isotropic materials with any Poisson's ratio. *The Visual Computer* [online], 36(4), pp. 809-825.

Hahn, F., Martin, S., Thomaszewski, B., Sumner, R., Coros, S. and Gross, M., 2012. Rig-space physics. *ACM Transactions on Graphics* [online], 31(4), pp. 1-8.

Han, J.H., Kim, J.-I., Kim, H. and Suh, J.W., 2021. Generate individually optimized blendshapes. In: *Proceedings of 2021 IEEE International Conference on Big Data and Smart Computing (BigComp)* [online], IEEE, pp. 114-120.

Hartmann, E., 2001. Implicit G^n -blending of vertices. *Computer Aided Geometric Design* [online],

18(3), pp. 267-285.

Hatna, A., Grieve, R. and Broomhead, P., 2001. Surface blending for machining purposes: A brief survey and application for machining compound surfaces. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* [online], 215(10), pp. 1397-1408.

Hoffmann, C. and Hopcroft, J., 1985. Automatic surface generation in computer aided design. *The Visual Computer*, 1(2), pp. 92-100.

Hoffmann, C. and Hopcroft, J., 1986. Quadric blending surfaces. *Computer-Aided Design*, 18(6), pp. 301-306.

Hoffmann, C. and Hopcroft, J., 1987. The potential method for blending surfaces and corners. In: *Geometric Modeling: Algorithms and New Trends* [online], eds. G. E. Farm (SIAM, USA, 1987), pp. 347-365.

Hong, M., Jeon, J.H., Yum, H.S. and Lee, S.H., 2016. Plausible mass-spring system using parallel computing on mobile devices. *Human-centric Computing and Information Sciences* [online], 6(1).

Hsu, K.L. and Tsay, D.M., 1998. Corner blending of free-form N-sided holes. *IEEE Computer Graphics and Applications* [online], 18(1), pp. 72-78.

Hsu, P.C., 2018. K-ary Implicit Blends with Increasing or Decreasing Blend Ranges for Level Blend Surfaces. *Journal of Advances in Information Technology* [online], 9(2), pp. 25-32.

Hwang, W.C. and Chuang, J.H., 2003. N-sided hole filling and vertex blending using subdivision surfaces. *Journal of Information Science and Engineering* [online], 19, pp. 857-879.

Ichim, A.E., Kadleček, P., Kavan, L. and Pauly, M., 2017. Phace. *ACM Transactions on Graphics*

[online], 36(4), pp. 1-14.

Jacobson, A., Tosun, E., Sorkine, O. and Zorin, D., 2010. Mixed Finite Elements for Variational Surface Modeling. *Computer Graphics Forum* [online], 29(5), pp. 1565-1574.

Jun, Y., Jiang, C., Li, R., Luo, C. and Wang, Z., 2018. Real-Time 3-D Facial Animation: From Appearance to Internal Articulators. *IEEE Transactions on Circuits and Systems for Video Technology* [online], 28(4), pp. 920-932.

Kadleček, P. and Kavan, L., 2019. Building Accurate Physics-based Face Models from Data. In: *Proceedings of the ACM on Computer Graphics and Interactive Techniques* [online], 2(2), pp. 1-16.

Kazakov, M., Pasko, A. and Adzhiev, V., 2003. Interactive metamorphosis and carving in a multi-volume scene. *GRAPHITE2003* [online], (Melbourne, Australia, 11-14 February, 2003), Publication of ACM SIGGRAPH, pp. 103-110.

Keeve, E., Girod, S., Pfeifle, P. and Girod, B., 1996. Anatomy-based tissue modeling using the finite element method. In: *Proceedings of the 7th Conference on Visualization* [online], pp. 21-27.

Kiciak, P., 2011. Bicubic B-spline blending patches with optimized shape. *Computer-Aided Design* [online], 43(2), pp. 133-144.

Kim, D., Kuang, T., Rodrigues, Y.L., Gateno, J., Shen, S.G.F., Wang, X., Stein, K., Deng, H.H., Liebschner, M.A.K. and Xia, J.J., 2021. A novel incremental simulation of facial changes following orthognathic surgery using FEM with realistic lip sliding effect. *Medical Image Analysis* [online], 72, p.102095.

Kim, J., Choi, M.G. and Kim, Y.J., 2020. Real-time muscle-based facial animation using shell elements and force decomposition. In: *Proceedings of Symposium on Interactive 3D Graphics*

and Games [online]. pp. 1-9.

KIM, K. and ELBER, G., 1996. A Symbolic Approach to Freeform Parametric Surface Blends. *The Journal of Visualization and Computer Animation* [online], 8(2), pp. 69-80.

Kim, M., Pons-Moll, G., Pujades, S., Bang, S., Kim, J., Black, M.J. and Lee, S.H., 2017. Data-driven physics for human soft tissue animation. *ACM Transactions on Graphics* [online], 36(4), pp. 1-12.

Kim, T., De Goes, F.D. and Iben, H., 2019. Anisotropic elasticity for inversion-safety and element rehabilitation. *ACM Transactions on Graphics* [online], 38(4), pp. 1-15.

Koch, R.M., Gross, M.H., Carls, F.R., von Büren, D.F., Fankhauser, G. and Parich, Y.I.H., 1996. Simulating facial surgery using finite element methods. In: *Proceeding of SIGGRAPH 1996* [online], pp. 421-428.

Kós, G., Martin, R.R.and Várady, T., 2000. Methods to recover constant radius rolling ball blends in reverse engineering. *Computer Aided Geometric Design* [online], 17(2), pp. 127-160.

Kos,G., 2000. Recovering Variable Radius Rolling Ball Blends in Reverse Engineering. *Journal for Manufacturing Science and Production*, 3(2-4), pp. 151-158.

Kosters, M., 1989. Quadratic blending surfaces for complex corners. *The Visual Computer*, 5(3), pp. 134-146.

Kosters, M., 1991. High-order implicit blending surfaces of low degree. *Computer Aided Geometric Design*, 8(6), pp. 495-512.

Kot, M., Nagahashi, H. and Szymczak, P., 2015. Elastic moduli of simple mass spring models. *The Visual Computer* [online], 31(10), pp. 1339-1350.

Kot, M. and Nagahashi, H., 2017. Mass spring models with adjustable Poisson's ratio. *The Visual*

Computer [online], 33(3), pp. 283-291.

Kozlov, Y., Bradley, D., Bächer, M., Thomaszewski, B., Beeler, T. and Gross, M., 2017. Enriching Facial Blendshape Rigs with Physical Simulation. *Computer Graphics Forum* [online], 36(2), pp. 75-84.

Kozlov, Y., Xu, H., Bächer, M., Bradley, D., Gross, M. and Beeler, T., 2019. Data-Driven Physical Face Inversion. *arXiv.org* [online]. Available at: <https://arxiv.org/abs/1907.10402> [Accessed 11 February 2022].

Krasauskas, R., 2008. Branching blend of natural quadrics based on surfaces with rational offsets. *Computer Aided Geometric Design* [online], 25(4-5), pp. 332-341.

Kravtsov, D., Fryazinov, O., Adzhiev, V., Pasko, A. and Comninos, P., 2013. Controlled Metamorphosis Between Skeleton-Driven Animated Polyhedral Meshes of Arbitrary Topologies. *Computer Graphics Forum* [online], 33(1), pp. 64-72.

Lee, Y., Terzopoulos, D. and Waters, K., 1995. Realistic modeling for facial animation. In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 1995)* [online], pp. 55-62.

Lewis, J.P., Anjyo, K., Rhee, T., Zhang, M., Oighin, F. and Deng, Z., 2014. Practice and theory of blendshape facial models. In: *Eurographics 2014-State of the Art Reports* [online], Eurographics Association, pp. 199-218.

Li, H., Yu, J., Ye, Y. and Bregler, C., 2013. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics* [online], 32(4), pp. 1-10.

Li, Y., Xu, H. and Barbic, J., 2017. Enriching Triangle Mesh Animations with Physically Based Simulation. *IEEE Transactions on Visualization and Computer Graphics* [online], 23(10), pp. 2301-2313.

Li, Z. C. and Chang, C.S., 1999. Boundary penalty finite element methods for blending surfaces, III. Superconvergence and stability and examples. *Journal of Computational and Applied Mathematics* [online], 110(2), pp. 241-270.

Lin, I., Tu, P., Yeh, J., Liang, R.H.and Ouhyoung, M., 2004. Surface detail capturing for realistic facial animation. *Journal of Computer Science and Technology* [online], 19(5), pp. 618-625.

Liang, X. and Boppart, S.A., 2010. Biomechanical Properties of In Vivo Human Skin From Dynamic Optical Coherence Elastography. *IEEE Transactions on Biomedical Engineering* [online], 57(4), pp. 953-959.

Liming, R.A., 1979. *Mathematics for Computer Graphics*. Fallbrook, California: Aero Publishers.

Liu, Q. and Sourin, A., 2005. Function-defined shape metamorphoses in visual cyberworlds. *The Visual Computer* [online], 22(12), pp. 977-990.

Liu, Q. and Sourin, A., 2006. Function-defined shape metamorphoses in visual cyberworlds. *The Visual Computer* [online], 22(12), pp. 977-990.

Liu, X., 2015. Filling N-Sided Holes With Trimmed B-Spline Surfaces Based on Energy-Minimization Method. *Journal of Computing and Information Science in Engineering*, 15(1).

Liu, X., Xia, S., Fan, Y. and Wang, Z., 2011. Exploring Non-Linear Relationship of Blendshape Facial Animation. *Computer Graphics Forum* [online], 30(6), pp. 1655-1666.

Lloyd, B.A., Szekely, G. and Harders, M., 2007. Identification of Spring Parameters for Deformable Object Simulation. *IEEE Transactions on Visualization and Computer Graphics* [online], 13(5), pp. 1081-1094.

Lukács, G., Hermann, T. and Várady, T., 1997. Continuity and self-intersections of variable radius rolling ball blend surfaces. In: Pratt M.J., Sriram R.D., Wozny M.J. (eds) *Product Modeling*

for Computer Integrated Design and Manufacture. IFIP Advances in Information and Communication Technology. Springer, Boston, MA.

Lukács, G., 1998. Differential geometry of G1 variable radius rolling ball blend surfaces. *Computer Aided Geometric Design* [online], 15(6), pp. 585-613.

Ma, W.G., Wang, Y.H., Fyffe, G., Chen, B. Y.and Debevec, P., 2011. A blendshape model that incorporates physical interaction. *Computer Animation and Virtual Worlds* [online], 23(3-4), pp. 235-243.

Middleditch, A. E.and Sears, K.H., 1985. Blend surfaces for set theoretic volume modelling systems. *ACM SIGGRAPH Computer Graphics*, 19(3), pp. 161-170.

Mora, H., Mora-Pascual, J.M., García-García, A. and Martínez-González, P., 2016. Computational Analysis of Distance Operators for the Iterative Closest Point Algorithm. *PLOS ONE* [online], 11(10), p.e0164694.

Nealen, A., Müller, M., Keiser, R., Boxerman, E. and Carlson, M., 2006. Physically Based Deformable Models in Computer Graphics. *Computer Graphics Forum* [online], 25(4), pp. 809-836.

Natsupakpong, S. and Cenk Çavuşoğlu, M.C., 2010. Determination of elasticity parameters in lumped element (mass-spring) models of deformable objects. *Graphical Models* [online], 72(6), pp. 61-73.

Noh, J.-Y. and Neumann, U., 1999. A survey of facial modeling and animation techniques. *Technical Report USC-TR-99-705* [online], University of Southern California, pp. 1-26.

Ohkura, K. and Kakazu, Y., 1992. Generalization of the potential method for blending three surfaces. *Computer-Aided Design*, 24(11), pp. 599-610.

Okereke, O.E., Iwueze, I.S. and Ohakwe, J., 2014. Some Contributions to the Solution of Cubic Equations. *British Journal of Mathematics & Computer Science*, 4(20), pp. 2929-2941.

Onizuka, H., Thomas, D., Uchiyama, H. and Taniguchi, R., 2019. Landmark-guided deformation transfer of template facial expressions for automatic generation of avatar blendshapes. In: *Proceedings of 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* [online], pp. 2100-2108.

Pan, Q., Xu, G. and Zhang, Y., 2014. A unified method for hybrid subdivision surface design using geometric partial differential equations. *Computer-Aided Design* [online], 46, pp. 110-119.

Park, J., Kim, T., Baek, S.Y. and Lee, K., 2015. An algorithm for estimating surface normal from its boundary curves. *Journal of Computational Design and Engineering* [online], 2(1), pp. 67-72.

Park, S.I. and Hodgins, J.K., 2008. Data-driven modeling of skin and muscle deformation. *ACM Transactions on Graphics* [online], 27(3), pp. 1-6.

Pasko, A.A., Adzhiev, V.D., Sourin, A.I. and Savchenko, V.V., 1995. Function representation in geometric modeling: concepts, implementation and applications. *The Visual Computer*, 11(8), pp. 429-446.

Pasko, A.A. and Savchenko, V.V., 1994. Blending operations for the functionally based constructive geometry. *Set-theoretic Solid Modeling: Techniques and Applications, CSG 94 Conference Proceedings*, pp. 151-161, Information Geometers, Winchester, UK.

Pasko, G., Kravtsov, D. and Pasko, A., 2010. Real-time space-time blending with improved user control. In Motion in Games, R. Boulic, Y. Chrysanthou, T. Komura (Eds.), *Lecture Notes in Computer Science*, 6459, pp. 146-157, Springer.

Pasko, G., Pasko, A., Ikeda, M. and Kunii, T., 2002. Bounded blending operations. In: *Proceedings of Shape Modeling International 2002* [online], pp. 95-103, Banff, Canada, IEEE Computer Society.

Pasko, G., Pasko, A., Ikeda, M. and Kunii, T., 2004. Advanced metamorphosis based on bounded space-time blending. In: *Proceedings of the 10th International Multimedia Modelling Conference* [online]. pp. 211-217, Brisbane, Queensland, Australia.

Pasko, G., Pasko, A.A. and Kunii, T.L., 2004. Space–time blending. *Computer Animation and Virtual Worlds* [online], 15(2), pp. 109-121.

Pasko, G., Pasko, A. and Kunii, T., 2005. Bounded Blending for Function-Based Shape Modeling. *IEEE Computer Graphics and Applications* [online], 25(2), pp. 36-45.

Pegna, J. and Wolter, F.E., 1992. Geometrical Criteria to Guarantee Curvature Continuity of Blend Surfaces. *Journal of Mechanical Design*, 114(1), pp. 201-210.

Piegl, L. and Tiller, W., 1999. Filling n-sided regions with NURBS patches. *The Visual Computer* [online], 15(2), pp. 77-89.

Piegl, L. and Tiller, W., 2012. *The NURBS Book*. 2nd ed. Berlin, Heidelberg: Springer Berlin Heidelberg.

Pighin, F., Hecker, J., Lischinski, D., Szeliski, R. and Salesin, D.H., 1998. Synthesizing realistic facial expressions from photographs. In: *Proceedings of SIGGRAPH 1998* [online], pp. 75-84.

Platt, S.M. and Badler, N.I., 1981. Animating facial expressions. *ACM SIGGRAPH Computer Graphics* [online], 15(3), pp. 245-252.

Richardson, E., Sela, M., Or-El, R. and Kimmel, R., 2017. Learning detailed face reconstruction from a single image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* [online], pp. 5553-5562.

Roach, P.A. and Martin, R.R., 1992. Production of blends and fairings by Fourier methods. *SPIE: Curves and Surfaces in Computer Vision and Graphics III*, 1830, pp. 162-173.

Rockwood, A.P., 1989. The displacement method for implicit blending surfaces in solid models. *ACM Transactions on Graphics* [online], 8(4), pp. 279-297.

Romeo, M. and Schvartzman, S.C., 2020. Data-Driven Facial Simulation. *Computer Graphics Forum* [online], 39(6), pp. 513-526.

Rossignac, J.R. and Requicha, A.A.G., 1984. Constant-radius blending in solid modeling. *ASME Computers in Mechanical Engineering (CIME)* [online], 3(1), pp. 65-73.

Russo, M., 2005. *Polygonal modelling: basic and advanced techniques*, (Worldwide Game and Graphics Library). Jones & Bartlett Learning, 1st edition.

Sanchez, M., Fryazinov, O., Adzhiev, V., Comninos, P. and Pasko, A., 2015. Space-Time Transfinite Interpolation of Volumetric Material Properties. *IEEE Transactions on Visualization and Computer Graphics* [online], 21(2), pp. 278-288.

Sanchez, M., Fryazinov, O., Vilbrandt, T. and Pasko, A., 2013. Morphological shape generation through user-controlled group metamorphosis. *Computers & Graphics* [online], 37(6), pp. 620-627.

San-Vicente, G., Aguinaga, I. and Celigueta, J., 2012. Cubical Mass-Spring Model Design Based on a Tensile Deformation Test and Nonlinear Material Model. *IEEE Transactions on Visualization and Computer Graphics* [online], 18(2), pp. 228-241.

Schichtel, M., 1993. G2 blend surfaces and filling of N-sided holes. *IEEE Computer Graphics and Applications* [online], 13(5), pp. 68-73.

Seo, H. and Thalmann, N.M., 2003. An automatic modeling of human bodies from sizing parameters. In: *Proceedings of the 2003 Symposium on Interactive 3D Graphics* [online], pp. 19-26.

Seo, J., Irving, G., Lewis, J.P. and Noh, J., 2011. Compression and direct manipulation of complex blendshape models. *ACM Transactions on Graphics* [online], 30(6), pp. 1-10.

Seol, Y., Seo, J., Kim, P.H., Lewis, J.P. and Noh, J., 2011. Weighted pose space editing for facial animation. *The Visual Computer* [online], 28(3), pp. 319-327.

Shen, X., Avital, E., Rezaienia, M.A., Paul, G. and Korakianitis, T., 2017. Computational methods for investigation of surface curvature effects on airfoil boundary layer behavior. *Journal of Algorithms & Computational Technology* [online], 11(1), pp. 68-82.

Shene, C.K., 1998. Blending two cones with Dupin cyclides. *Computer Aided Geometric Design* [online], 15(7), pp. 643-673.

Shi, K.L., Yong, J.H., Sun, J.G., Paul, J. and Gu, H., 2010. Filling n-sided regions with G 1 triangular Coons B-spline patches. *The Visual Computer* [online], 26(6-8), pp. 791-800.

Sifakis, E., Neverov, I. and Fedkiw, R., 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Transactions on Graphics* [online], 24(3), pp. 417-425.

Smith, B., Goes, F.D. and Kim, T., 2018. Stable Neo-Hookean Flesh Simulation. *ACM Transactions on Graphics* [online], 37(2), pp. 1-15.

Song, Q. and Wang, J., 2007. Generating parametric blending surfaces based on partial reparameterization of base surfaces. *Computer-Aided Design* [online], 39(11), pp. 953-963.

Stanko, T., Hahmann, S., Bonneau, G.P. and Saguin-Sprynski, N., 2016. Surfacing curve networks

with normal control. *Computers & Graphics* [online], 60, pp. 1-8.

Tena, J.R., De la Torre, F. and Matthews, I., 2011. Interactive region-based linear 3D face models. *ACM Transactions on Graphics* [online], 30(4), pp. 1-10.

Tereshin, A., Pasko, A., Fryazinov, O. and Adzhiev, V., 2021. Hybrid function representation for heterogeneous objects. *Graphical Models* [online], 114, p.101098.

Terzopoulos, D. and Waters, K., 1993. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* [online], 15(6), pp. 569-579.

Tewari, A., Zollhöfer, M., Garrido, P., Bernard, F., Kim, H., Pérez, P. and Theobalt, C., 2017. Self-supervised Multi-level Face Model Learning for Monocular Reconstruction at over 250 Hz. *arXiv.org* [online]. Available at: <https://arxiv.org/abs/1712.02859> [Accessed 12 February 2022].

Tosun, E., Gingold, Y.I., Reisman, J. and Zorin, D., 2007. Shape optimization using reflection lines. In: *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* [online], Eurographics Association, pp. 193-202.

Tran, L. and Liu, X., 2018. *Nonlinear 3D Face Morphable Model*. *arXiv.org* [online]. Available at: <https://arxiv.org/abs/1804.03786> [Accessed 12 February 2022].

Tzirakis, P., Papaioannou, A., Lattas, A., Tarasiou, M., Schuller, B. and Zafeiriou, S., 2019. Synthesising 3D facial motion from “in-the-wild” speech. *Arxiv.org* [online] . Available at: <https://arxiv.org/pdf/1904.07002.pdf> [Accessed 12 February 2022].

Tu Z., Weng D., Cheng D., Bao Y., Liang B. and Luo L., 2019. An automatic base expression selection algorithm based on local blendshape model. *Lecture Notes in Computer Science*, 11902, pp 220-231.

Uhlmann, T. and Brunnett, G., 2019. *Towards Point-based Facial Movement Simulation*. *Diglib.eg.org* [online]. Available at: <https://diglib.eg.org/handle/10.2312/egp20191049> [Accessed 12 February 2022].

Ulusoy, İ., Akagündüz, E. and Weber, G.W., 2010. Inverse solution for parameter estimation of a dynamic volume spline based forehead skin model. *Inverse Problems in Science and Engineering* [online], 18(5), pp. 585-612.

Ververas, E. and Zafeiriou, S., 2020. SliderGAN: Synthesizing Expressive Face Images by Sliding 3D Blendshape Parameters. *International Journal of Computer Vision* [online], 128(10-11), pp. 2629-2650.

Vida, J., Martin, R.R. and Varady, T., 1994. A survey of blending methods that use parametric surfaces. *Computer-Aided Design*, 26(5), pp. 341-365

Vlasic, D., Brand, M., Pfister, H. and Popović, J., 2005. Face transfer with multilinear models. *ACM Transactions on Graphics* [online], 24(3), pp. 426-433.

Wang, B., Zheng, M. and Barbič, J., 2020. Adjustable Constrained Soft-Tissue Dynamics. *Computer Graphics Forum* [online], 39(7), pp. 69-79.

Wang, M., Bradley, D., Zafeiriou, S. and Beeler, T., 2020. Facial Expression Synthesis using a Global-Local Multilinear Framework. *Computer Graphics Forum* [online], 39(2), pp. 235-245.

Wang, S., Cheng, Z., Deng, X., Chang, L., Duan, F. and Lu, K., 2020. Leveraging 3D blendshape for facial expression recognition using CNN. *Science China Information Sciences* [online], 63(2), pp. 1-3.

Wang, Z., Ling, J., Feng, C., Lu, M. and Xu, F., 2020. Emotion-preserving Blendshape Update with Real-time Face Tracking. *IEEE Transactions on Visualization and Computer Graphics*

[online], doi: 10.1109/TVCG.2020.3033838.

Warburton, M. and Maddock, S., 2013. Physically-based forehead animation including wrinkles. *Computer Animation and Virtual Worlds* [online], 26(1), pp. 55-68.

Waters, K. and Levergood, T.M., 1993. DECface: A system for synthetic face applications. *Multimedia Tools and Applications* [online], 1(4), pp. 349-366.

Wei, J.C.J., Edwards, G.A., Martin, D.J., Huang, H., Crichton, M. and Kendall, M., 2017. Allometric scaling of skin thickness, elasticity, viscoelasticity to mass for micro-medical device translation: from mice, rats, rabbits, pigs to humans. *Scientific Reports* [online], 7(1),15885.

Woodwark, J.R., 1987. Blends in geometric modelling. *The Mathematics of Surfaces II*, eds. R. R. Martin, Oxford University Press, New York, pp. 255-297.

Wu, Y. and Zhu, C.G., 2021. Generating bicubic B-spline surfaces by a sixth order PDE. *AIMS Mathematics* [online], 6(2), pp. 1677-1694.

Xu, L., Lu, Y. and Liu, Q., 2018. Integrating viscoelastic mass spring dampers into position-based dynamics to simulate soft tissue deformation in real time. *Royal Society Open Science* [online], 5(2), pp. 1-16.

Yang, Y.J., Yong, J.H., Zhang, H., Paul, J.C. and Sun, J.G., 2006. A rational extension of Piegl's method for filling -sided holes. *Computer-Aided Design* [online], 38(11), pp. 1166-1178.

You, L.H., Comninos, P. and Zhang, J.J., 2004a. PDE blending surfaces with C2 continuity. *Computers & Graphics* [online], 28(6), pp. 895-906.

You, L.H, Ugail, H. and Zhang, J.J., 2012. Controllable C1 continuous blending of time-dependent parametric surfaces. *The Visual Computer* [online], 28(6-8), pp. 573-583.

You, L.H., Ugail, H. and Zhang, J.J., 2012. Controllable C1 continuous blending of time-

dependent parametric surfaces. *The Visual Computer* [online], 28(6-8), pp. 573-583.

You, L.H., Zhang, J.J. and Comninos, P., 2004b. Blending surface generation using a fast and accurate analytical solution of a fourth-order PDE with three shape control parameters. *The Visual Computer* [online], 20(2-3), pp. 199-214.

Yu, F., Lu, Z., Luo, Z. and Wang, P., 2010. *Three-Dimensional Model Analysis and Processing*, Springer, pp. 16.

Zhang, J.J. and You, L.H., 2002. PDE based surface representation—vase design. *Computers & Graphics* [online], 26(1), pp. 89-98.

Zhang, L., Guo, J., Xiao, J., Zhang, X. and Yan, D.M, 2020. Blending Surface Segmentation and Editing for 3D Models. *IEEE Transactions on Visualization and Computer Graphics* [online], doi: 10.1109/TVCG.2020.3045450.

Zhao, T., Wang, D. and Hong, H., 2011. Solution formulas for cubic equations without or with constraints. *Journal of Symbolic Computation* [online], 46(8), pp. 904-918.

Zhou, P. and Qian, W.H., 2009. A vertex-first parametric algorithm for polyhedron blending. *Computer-Aided Design* [online], 41(11), pp. 812-824.

Zhou, P. and Qian, W.H., 2010. Polyhedral vertex blending with setbacks using rational S-patches. *Computer Aided Geometric Design* [online], 27(3), pp. 233-244.