Finding the Optimal Shape of an Object Using Design-By-Morphing

by

Sahuck Oh

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering - Mechanical Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Philip S. Marcus, Chair
Professor Ömer Savas
Associate Professor Per-Oolf Persson

Summer 2016

Finding the Optimal Shape of an Object Using Design-By-Morphing

Abstract

Finding the Optimal Shape of an Object Using Design-By-Morphing

by

Sahuck Oh

Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Philip S. Marcus, Chair

We present a new design method, which we call design-by-morphing, for the optimal design of the shape of an object. The surface of one or more objects (or the sub-objects from which it is composed) is represented as a truncated series of exponentially-convergent spectral basis functions multiplied by spectral coefficients. A morphed object or sub-object is obtained from a new set of spectral coefficients, which are a weighted average of the spectral coefficients of the original objects or sub-objects from which it is morphed. Optimized designs are created by choosing the weights such that a cost function of the new morphed shape is minimized. The boundaries of an object and the interfaces between sub-objects can be forced to satisfy geometric constraints on their shapes, slopes, curvature, etc. With these constraints, sub-objects can be seamlessly attached to each other to create a complex object. Because design-by-morphing has the flexibility to adjust independently the weights of sub-objects, users or an automated algorithm can choose some subset of sub-objects to be optimized while preserving or restricting the changes of other sub-objects. Our design-by-morphing method can be automated and is computationally efficient, so it requires much less human input than traditional design methods and is therefore not only inexpensive but also free from human bias in finding optimal designs that are radical and non-intuitive.

We have applied optimization via design-by- morphing to aircraft and a turbine-99 draft tube, and reduced drag-to-lift ratio and maximized mean pressure recovery factor by 23.1% and 10.9%, respectively. We believe that this optimization method is applicable to a wide variety of engineering applications in which the performance of an object depends on the aerodynamic or hydrodynamic properties of its shape.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank everyone who supports me during my Ph.D. studies, especially my parents.

Chapter 1

Introduction

1.1 Background

In many engineering fields, there is a need to design products with optimal shapes to enhance their hydrodynamic, aerodynamic, thermal, structural, or mechanical properties since the performance of many objects are effected by their own shapes. For example, minimizing drag force of vehicles such as airplanes, trains and automobiles have been an significant issues to engineers to efficiently operate them and the shape of a wave energy collector in the ocean has been studied for a long time to maximize energy extraction [37, 38]. People have also tried to find the optimal shape of wind turbine blades in terms of the twist angle, chord length, shape of airfoils and thick of a blade to generate more electricity from wind [10, 27] and reduce noise [39].

There are many factors ones should consider in optimizing the shapes of products. For the design to be practical, it needs to be inexpensive to manufacture and maintain, and appeal to a consumer, but here we assume that the main concern is performance. We further assume that performance can be quantitatively computed in terms of a cost function and the cost function can be numerically computed. For example it might be the drag-to-lift ratio of an airplane, the pressure recovery factor in a turbine draft tube, or the drag of a train. In fact, these three example objects and their cost functions are what we use to illustrate our optimization.

To obtain the performance-based optimized shape of an object, design variables which modifies the shape of an object need to be defined and cost functions of new shaped objects created by varying design variables should be able to compute. In addition, optimization scheme to predict optimum weights generating the best performed shape of an object is also required. The commonly used optimum design process in industry incorporating all of these requirements are shown in Figure 1.1. The START in the figure assumes that the manufacturer needs a totally new design and that the first attempt at a new design is provided. Then, design process starts with this initially designed object which is considered as the first attempt of a new design. In preparing an initial design process and in subsequent

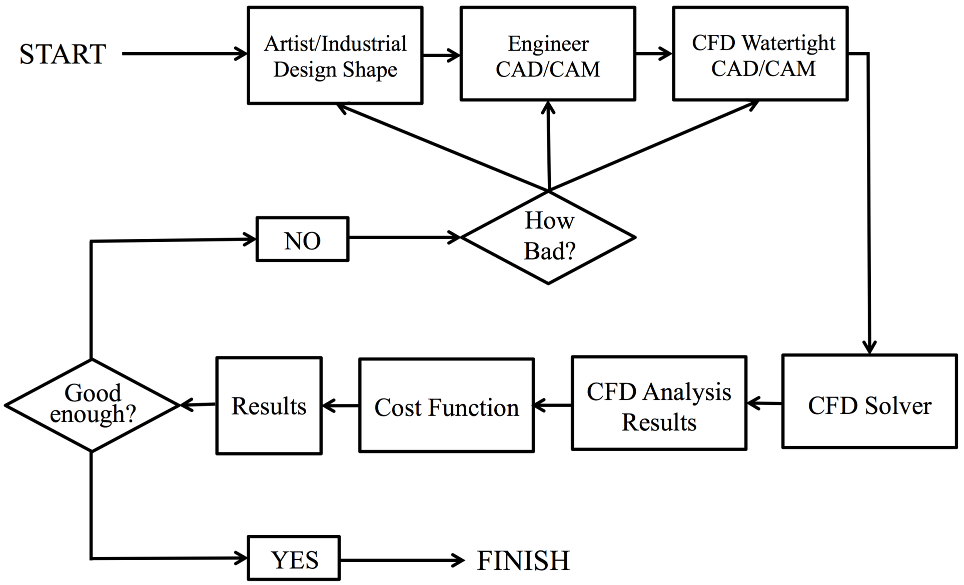


Figure 1.1: Flowchart of current engineering design process.

iterations of the design process, mesh should be generated for CFD simulation. Once the mesh is generated, the designer uses CFD tools to simulate performance of a given object and compute the cost function. If the cost function is not sufficiently low, designer uses optimization scheme to compute values of optimum design variables expected to induce a better performed object and then a new shaped object is created with optimum design variables. This process is repeated until the cost function becomes sufficiently small. To have successfully working optimization process, although all of these steps should work well, our research focused on how to create a new shaped object efficiently. The definition of an efficient way of creating a new design is altering the shape of an object with fewer design variables but encouraging creative shape change which can bring radical shape change from an initially designed object called a baseline object.

However, there are currently two major problems in creating a new shaped object. First, when an existing design under-performs and needs to be improved, there is no systematic way of modifying the design. For example, engineers face the difficulty of trying to decide whether a surface should be made more concave or convex, whether another piece should be lengthened or shortened, or whether a part should be more circular or more elliptical, etc. Further vexing the engineers is the fact that once an approach for improving the design is adopted, creating the new design with existing CAD/CAM tools is often time-consuming and problematic if the design change is a radical departure from an baseline design. The latter is true because many commonly used push-pull tools that are used to change the shape of an object work only locally. A piece of a surface's shape can be represented locally by splines [23] and changes can be made by pushing or pulling points on the original surface closer to or farther away from selected control points. Not only are the changes made with these tools confined to a small part of the overall surface of an object, thereby prohibiting radical changes, but also these modest changes often lead to shapes that are problematic when used with other computer packages such as mesh generators. In particular, often surfaces generated with push-pull packages are not watertight. That is the mesh points along the newly designed surface are interpreted by some computer packages to be multiple, disconnect surfaces rather than a single continuously connected surface.

Our design-by-morphing methodology addresses both of these problems. First, it systematically improves a design by searching over a finite parameter space subject to weights of given baseline objects and other imposed geometric constraints, but the changes in shape are not just local but can be global. Our new design method is so robust, proposed objects (or sub-objects) are not limited to interpolation among multiple existing designs; the proposed object can be extrapolated away from existing designs so that a radical new design can be created. This is significantly important because the characteristics of the existing designs can be enhanced, weakened or removed for the proposed design.

The essential idea of design-by-morphing is to reduce the effectively huge amount of dimensional parameters related to the vertices or control points of the shape of an object to a more manageable, but carefully chosen, reduced number of design parameters consisting of N existing shapes or the sub-objects of the existing shapes. New morphed shapes are created by assigning N weights to the N existing shapes of objects or sub-objects. The weights could all be positive and less than unity, which would create morphed shapes that are interpolations of the existing shapes, or the weights could be negative or much greater than unity, which would create morphed shapes that are extrapolations of the existing shapes. The cost function of the optimal design problem is implicitly a function of the N values of the weights. A optimization method such as a gradient method or genetic algorithm coupled with an artificial neural network for creating a response surface can then be used to find the optimal values of the weights that minimize the cost function. This methodology allows for a systematic way of finding the optimal shape with affordable amount of computational cost.

One key advantage of our method is the way in which shapes are represented. Rather than representing a shape as a mesh of points, each shape or each sub-object from which a shape is made in our method is represented as a truncated spectral sum of basis functions multiplied by spectral coefficients. Because spectral methods are so efficient [9] the truncated sum representing the shape converges exponentially fast so that with only a few thousand terms the truncated sum can spatially resolve a surface with high accuracy. Furthermore, when the object is represented with its basis function, geometric properties such as shape, slope and curvature at the boundary of the object is controlled to satisfy geometric constraints of the design problems or to join sub-object with no discontinues in shape, slope or curvature where they are joined. This joining sub-object makes our method flexible enough to keep some sub-objects of the original design fixed or force the sub-object be a specific shape while radically changing others, which allows to have the more wide range of shape alteration, and make breakthrough performance improvements of the object.

The main purpose of our research is to propose a new way to design and optimize an object which minimizes the cost function or maximizes the performance of the design problem. The cost or performance function we have considered is computational fluid dynamic (CFD) related quantities such as drag, lift and pressure over the objects. The optimized object is the shape of the new design improved their performance compared to the existing objects. Compared to other design method, our method is not only easy to automated, but is also explores wide shape alteration with less design variables, therefore, finding radical new designs that significantly improve current performance of the object. We believe our method can be applied various engineering objects and produce admirable performance improvement which has not been founded by other design methods.

1.2 Related work

Current methodology for design and shape optimization

The common way to make a change in shape in optimal design problem is to deform the shape using free-form deformation (FFD) or variants originated from it. [1, 2]. With FFD, a user chooses the locations of certain number of control points. The volume enclosed by these control points is parameterized and represented as an analytical function. The geometry wanted to be redesigned is embedded in the volume, and then, by moving the control points formed the volume, the geometry inside the enclosed volume are indirectly deformed. A number of the shape optimization researches implemented based on this approach. For example, Poole et al. [40] optimized the shape of the two dimensional airfoil to minimize the drag force by deforming control points on the airfoil surface. Lyu et al. [30] carried out the shape optimization of the Common Research Model wing respect to minimum drag coefficient by considering 720 FFD control points as design parameters. However, as commented by Anderson et al. [2] and as seen in own experience with volumetric deformation, FFD-based deformation techniques are not easy and intuitive to handle, and require many design variables to achieve reasonable shape modification.

Another widely used way in inducing shape modification to find the optimum shape of an object is using parametric based deformation. In parametric based deformation, the shape deformation of an object is described by parametric variables such as height, thickness, angle and etc. Then, optimization is implemented by computing optimum values of these parameters. For example, Agnew draft tube is parameterized in terms of its height and angle [48] and Long et al. used four geometric variables, main chamber diameter,height, and attachment angles for the inlet and outlet arms, to parameterize the shape of a pulsatile ventricular assist device [29]. These parameterization enables to variate the shape with a few design variables, but creating designs which cannot be obtained by incrementally varying design parameters is impossible.

It should be noted that Kang [26] attempted to optimize the shapes of ships using a method similar to our design-by-morphing method. Kang developed new ship designs by morphing among different ships, but the shapes of the ships were represented on triangular meshes rather than as a truncated spectral sum. Furthermore, Kang did not impose any geometric constraints. Although Kang demonstrated a capability to morph multiple objects, he did not report the results of his shape optimization. Besides, the shape editing method [6, 59, 60] have great attributes in allowing creative shape exploration to be sought. However, the shape editing method has not been automated with CFD simulation nor has they been used to optimize a shape.

Current methodology for stitching objects

In our design method, a shape can be represented by many sub-objects as well as one whole object. If the shape is represented as many parts, each sub-object is stitched together with constraints such that the shape, slope or curvature at the joints should be matched. To implement these constraints, we solve partial differential equation (PDE) whose boundary conditions are defined as the constraints what we want to impose on an sub-object. we call this process stitching.

The most relevant work to our stitching approach is found in creating blending objects based on solving PDE. Similar to our method, [51, 57, 58] solve a fourth-order partial differential equation in creating C_1 continuous blending objects where the objects' boundaries are controlled by the boundary conditions of PDE. In their work, they show various examples of blended objects, but surface shapes of examples are simple or analytically defined; therefore, they are inapplicable when it comes to designing engineering objects. Besides of them, [4] developed a software which creates generic aircraft shapes from a numbers of curves defining the shapes of aircrafts and showed various aircraft configurations by adjusting parameters related with boundary conditions of PDE. Although they showed many different generations of aircrafts, their method requires many control curves to create complex shaped aircrafts, which makes it difficult.

In design-by-morphing, stitching method is used to connect sub-objects with respect to matching shape, slope and curvature at a joint as well as to impose geometric constrains on an object. Creating a new object by stitching sub-objects is very beneficial in creating a variety of innovative objects because it allows to collect advantages or remove disadvantages of baseline objects by choosing optimum weights of sub-objects of them. For example, when we consider to morph a spoon with a fork, if the tip of a fork is stitched to the spoon whose tip is removed, it creates a sfork that possesses all pros of a spoon and fork. We believe morphing and stitching sub-object by sub-object helps in creating a new types of objects that can possess all good properties of baseline objects.

Funkhouser et al. [15], Kalogerakis et al. [25] and Schulz et al. [47] have conducted similar approach where a new shape is created by synthesizing components from existing shapes, which is usually called design-by-example. However, ways that components from existing shapes are composed together are somewhat difficult to be applied to an engineering design problem. One way to stitch two different components in their studies is connecting them by specifying where a component is attached to the other without high derivative continuity conditions. Or, components are smoothed by a user by introducing fillet at the joint. Both methods are not suitable for engineering purpose because stitching method is required to preserve prescribed geometric constraints in terms of location, slope and curvature. For example, back of a first car of a train is demanded to exactly satisfy specified shape, slope or curvature to be followed by a passenger train without degrading aerodynamic performance. Dealing these conditions with their methods is difficult, while it is straightforward in design-by-morphing.

Current methodology for morphing

A common way in spectrally representing and morphing objects is using eigenfunctions of Laplace-Beltrami operator of an object as well described in [14, 43]. In fact, spherical harmonics, which are chosen as one of our spectral basis functions, are a special case of eigenfunctions of Laplace-Betrami operator when it applied to a sphere. As Lévy [28] showed, eigenfunctions of Laplace-Betrami operator works well in reconstructing geometries and transfer pose of objects. However, it cannot be utilized in optimum design process because computing those eigenfunctions not only requires huge calculation time, but eigenfunctions should be also recalculated as a new shaped object is proposed during optimization process. In design-by-morphing, instead using eigenfunctions of Laplace-Beltrami operator, we accomplish spectral representation of an object relying on our spectral basis functions. Our basis functions need much less computing time and are so independent of the shape of an object to represent that the basis functions unnecessary to be recalculated in optimization process. The idea of enabling to use our spectral basis functions is representing and morphing simple shaped objects that our spectral basis functions easy to handle. A complex shaped object is segmented into simple shaped sub-objects and the the surface of sub-objects are approximated and morphed by our spectral basis functions, and then, the sub-objects are stitched together to reconstruct an entire object. Due to our stitching method allowing sub-objects are easily connected, our approach can be expended to a complex shaped object. The more detail knowledge about our spectral basis function and the quantitative comparison to represent the shape of an object using our basis function to using eigenfunctions of Laplace Beltrami operator are described in Chapter 2.

In most commercial morphing packages, the 2D surface of a 3D object is represented by a set of points or vertices of a triangular mesh. Morphing between two objects works best when there is a one-to-one correspondence between the vertices of the two objects that are being morphed. If that identification can be made, the vertices on the morph's surface is created by interpolating between each pair of corresponding vertices of the two surfaces that are being morphed. If a one-to-one correspondence is not known or does not exist because the two surfaces have an unequal number of vertices, the software, or more commonly the user must make a one-to-one correspondence by providing a script. It is easy to imagine the strange morph between the shapes of two humans that would occur if the vertex at the tip of one human's nose was incorrectly paired with the vertex at the tip of the other human's toe. Yet, that type of incorrect pairing occurs regularly in commercial software. Figure 1.2

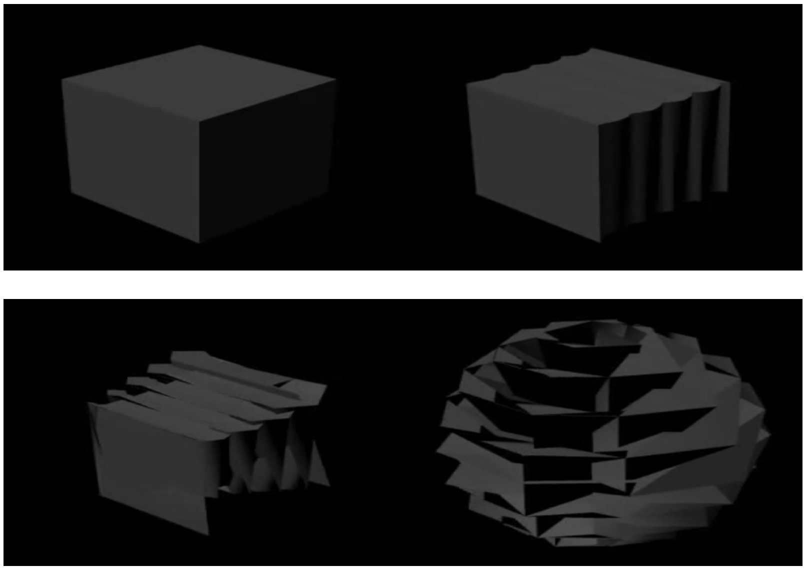


Figure 1.2: Autodesk Maya fails to morph a cube into a sphere.

is an example of a failed morph between a cube and a sphere that was carried out with the Autodesk Maya package. Similarly bad results are obtained with the Houdini morphing software. However, both Houdini and Maya give excellent morphs when the user provides them with scripts on how to carry out the morph. Unfortunately, a software package that attempts to produce optimal designs by morphing shapes together must be able to carry out the morphing in an automated manner without relying on human intervention. We have found that when the shapes of objects are represented by truncated sums of spectral basis functions multiplied by spectral coefficients, a morph between two or more shapes can be carried out by creating a new truncated spectral sum with new spectral coefficients that are the weighted averages of the spectral coefficients of the shapes being morphed. We have found that morphing with this method always creates sensible shapes and the procedure never breaks down, so it can be fully automated.

Usage of spherical harmonics

The most popular basis function among our basis functions is spherical harmonics, which have been applied to represent an object in various fields, from biological molecules [45], heads and brains [5, 55, 56], and medical images of tissue obtained by MRI [19]. However, not all of these works preserve the exponential convergence of the truncated series and are using the fast transform method. In our study, well-established methods [7, 9, 21, 32, 46] are used for obtaining the transforms required for efficiently computing the spectral expansion coefficients used in the truncated series. These well-known, unitary transforms are not only

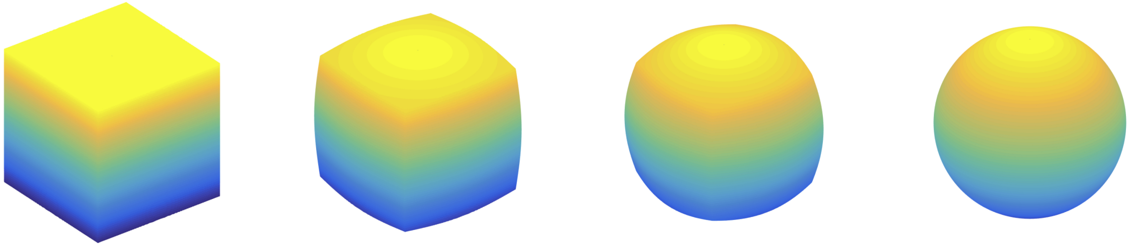


Figure 1.3: Morphing from a cube to a sphere using our morphing method

fast, but are a much more accurate way to compute the spectral expansion coefficients than using a least squares method, as many of the authors cited here have done. Similarly, we use well-established methods for choosing the spectral basis functions in which to expand a function. Usually, one chooses the basis functions to be eigenmodes of a Sturm-Liouville equation [31]. This choice generally allows the imposition of boundary conditions without destroying the exponential convergence of the truncated spectral sum.

1.3 What do we mean by spectral representation?

In this section, we illustrate what we mean by a spectral representation by considering the pedagogical case of the 1-dimensional boundary of a 2D object in a flat 2D world. Consider a flat, 2D starfish as shown in Figure 5.2a. Its boundary lies in a plane. Using a polar coordinate system (r, θ) with the origin in the center of the starfish, the starfish's shape can be described by the radius $r(\theta)$ of its boundary. This representation breaks down if $r(\theta)$ is not a single-valued function of θ, and we consider those cases later. Objects in which $r(\theta)$ is [not] a single-valued function of θ are known as [non-]star-shaped objects. In non-spectral representations, the curve $r(\theta)$ would be represented traditionally by placing a large number of points or vertices on the curve and then approximating the shape of the starfish by connecting those points with straight lines or low-order polynomials. In a spectral method, $r(\theta)$ is approximated as a truncated Fourier series as a function of azimuthal angle θ:

$$r(\theta) \simeq \sum_{m=-M}^{M} a_m \, e^{i\, m\, \theta}. \tag{1.1}$$

Spectral methods have the advantage that if the shape of the object is sufficiently smooth, the series converges exponentially. In practical terms, exponential convergence means that to represent the boundary to a given amount of accuracy, the number of spectral coefficients a_m needed in the truncated sum is much smaller than the number of traditional vertices needed on the boundary. If an object is not star-shaped, it is still often possible to use

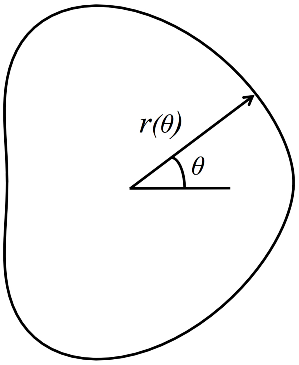


Figure 1.4: Schematic of a 2D star-shaped object. The radius of the object, r, is a function of azithumal angle θ.

a truncated spectral sum to represent $r(\theta)$ by mapping from θ to a new variable (say, the arc-length of the boundary [52]) such that r is single-valued in the new variable and the object is star-shaped with respect to the new mapped variable. However, not all mappings leave the expansion exponentially convergent and in those cases many of the advantages of spectral methods are lost.

Spectral representations of shapes easily extend to 2D boundaries of 3D, star-shaped objects. One of the simplest methods, and the most common in the literature, uses spherical harmonic functions $Y_l^m(\theta,\phi)$ as the spectral basis functions so that the spherical radius $R(\theta,\phi)$ (denoting the distance of the boundary at spherical coordinates $(\theta,\ \phi)$ from the coordinate system origin inside the object) is

$$R(\theta,\phi) \simeq \sum_{l=0}^{L} \sum_{m=-l}^{l} \ a_l^m \ Y_l^m(\theta,\phi), \tag{1.2}$$

where $Y_l^m(\theta,\phi)$ is the spherical harmonic function of degree l and order m, L is the truncation degree of the spherical harmonics, and a_l^m is the (l,m) spectral coefficient. For non-star-shaped objects, some researchers (c.f., [8]) have developed mappings for θ and ϕ that turn non-star-shaped 3D objects into star-shaped ones, but the mappings are complex and do not always preserve the exponential convergence of the truncated series.

Sometimes it is necessary to impose constraints on shapes. For example, it may be necessary for a bubble to attach itself to a wall such that the 2D region of attachment has a prescribed shape or such that the contact angle at each point is constrained to have a certain value. We refer to problems in which there is one region where constraints are imposed as a 1-hole problem. The radial distance r is represented as a double truncated sum in terms of Jacobi polynomials and Fourier expansions,

$$r(\theta, \phi) \simeq \sum_{l=0}^{L} \sum_{m=-N}^{N} a_l^m u^{|m|} P_l^{(0,|m|)}(2u^2 - 1) e^{im\phi}, \tag{1.3}$$

where

$$u = \frac{2\theta}{\pi} \tag{1.4}$$

and $P_l^{(\alpha,\beta)}(s)$ is Jacobi polynomials. Some researchers have tried to represent the shape of a constrained 1-hole problem with spherical harmonics, various types of hemispherical functions, or shifted associated Legendre polynomials coupled with a Fourier expansion [17], but these methods have not been very successful in that they are not very flexible, do not have exponential convergence, fail to satisfy accurately the imposed boundary conditions, and/or have very large shape distortions near the imposed boundary.

Sometimes it is necessary to impose constraints on a shape at two locations. An example of this would be a neck that connects a head and a torso. The boundary of the neck must be constrained at the points of attachment to the head and at the points of attachment to the torso. We refer to this as the 2-hole problem. Using a cylindrical coordinate system (r, z, ϕ) in which the "neck" is centered and aligned with the z-axis, the surface of the neck is located at the radial distance $r(z, \phi)$ from the z-axis. The radial distance r is represented as a double truncated sum in terms of Chebyshev polynomials $T_j(z)$ in z and Fourier modes:

$$r(z, \phi) \simeq \sum_{j=0}^{L} \sum_{m=-M}^{M} a_j^m \, T_j(z) \, e^{i\,m\,\phi}. \tag{1.5}$$

Patches such as a sheet of a paper cab be represented in a Cartesian (x, y) domain by having their heights z approximated as a double Chebyshev series in x and y:

$$z(x, y) \simeq \sum_{l=0}^{L} \sum_{m=0}^{M} a_{lm} \, P_l(x) \, P_m(y), \tag{1.6}$$

where $P_l(x)$ and $P_m(y)$ are respectively the l^{th} and m^{th} Chebyshev polynomials, and where $-1 \leq x \leq 1$ and $-1 \leq y \leq 1$. To produce the patch outside of these domains, the x and y coordinates should be appropriately mapped. All of these basis functions show spectral accuracy in representing their own objects.

In summarizing the previous work on representing the shapes of objects as truncated spectral sums, the main difficulty is that there appears to have been a trial-and-error approach. This is surprising because truncated spectral sums have been used for over a few decades in computational fluid dynamics (CFD) to represent fluid flows. Therefore a large body of applied mathematics has been developed to assure that truncated spectral sums are both exponentially convergent and satisfy boundary constraints [7, 21]. For example, well-known unitary transforms, which are not only fast, but are a much more accurate way to compute the spectral expansion coefficients than using a least squares method as many of the authors cited here have done, were used. Further, one chooses the basis functions to be eigenmodes of a Sturm-Liouville equation to allow the imposition of boundary conditions without destroying the exponential convergence of the truncated spectral sum, which is key in creating efficient and robust design-by-morphing method.

Chapter 2

Basis Functions and morphing an object

In design-by-morphing, an object is classified by the number of its open boundaries and spectral basis functions used to approximate the surface of an object are determined by a type of an object. Topologically, an object belonging to a sphere with no boundaries is called a *0-hole object*. An object belonging to a sphere with one boundary such as a drinking glass is defined as a *1-hole object*, and an object belonging to a sphere with two boundaries is defined such as a drinking straw is defined as a *2-hole object*. If an object has four boundary components, such as sheet of paper, it is called a *patch*. Simple examples of a 0-, 1- and 2-hole objects and a patch are shown in Figure 2.1. The surfaces of 0-, 1-, 2-hole objects and a patch are represented by their own spectral basis functions. In this chapter, we will describe basis functions what we have used for a 0- 1- and 2-hole objects and a patch, and operators related to those basis functions, which will be used in solving PDEs to impose geometric constraints. When we approximate the surface of an object using our spectral basis functions, we will write mapping parameters as u and v as a general expression. Here, u and v mean parameters to map the surface of an object into spectral basis functions' functional domain. For a star-shaped object, it can be respectively θ and ϕ as expressed in Equation 1.2 or can be respectively z and ϕ and x and y as expressed in Equations 1.5 and 1.6. For a non-star shaped object u and v are special mapping like arc-length of an object.

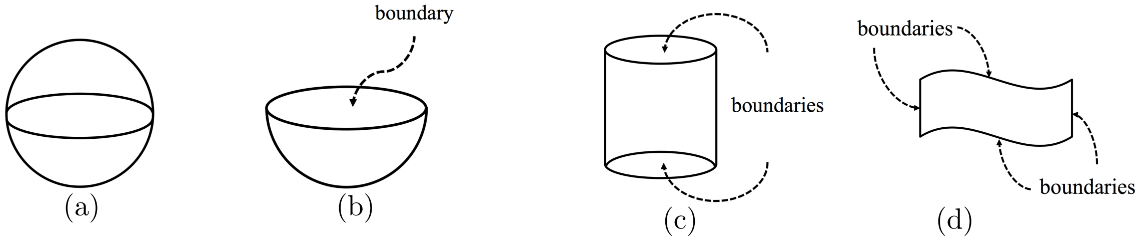


Figure 2.1: Simple examples of n-hole objects and patches: (a) 0-hole object, (b) 1-hole object, (c) 2-hole object, (d) patch. The boundaries of objects are pointed out.

2.1 0-hole Object

Representation

When an object has no boundary components, an object is classified as an 0-hole object. The spherical radius of 0-hole objects, which can be mapped as a function of the latitude angle and azimuthal angle of the surface, is approximated with the spherical harmonics with their coefficients

$$r(u,v) = \sum_{l=0}^{L} \sum_{m=-l}^{l} a_l^m Y_l^m(u,v), \tag{2.1}$$

where $Y_l^m(u,v)$ is spherical harmonic of degree l and order m, defined as

$$Y_l^m(u,v) = P_l^m(cosu)e^{imv}, \tag{2.2}$$

and a_l^m are spectral coefficients and P_l^m are the associated Legendre polynomials. Here, if a 0-hole object is star-shaped, u and v are latitude angle θ and azimuthal angle ϕ of a 0-hole object, respectively. Spherical harmonics are orthonormal basis function over a sphere and show orthogonality

$$\int_0^{2\pi} \int_0^{\pi} Y_l^m(u,v) Y_n^{k*}(u,v) sinududv = \delta_{ln}\delta_{mk} \tag{2.3}$$

Due to the orthogonality property of the spherical harmonics, spherical harmonics coefficients a_l^m is obtained by multiplying $Y_n^{k*}(u,v)sinu$ in Equation 2.3 and integrating respect to u and v over 0 to π and 0 to 2π, respectively,

$$a_l^m = \int_0^{2\pi} \int_0^{\pi} r(u,v) Y_l^{m*}(u,v) sin\theta dudv \tag{2.4}$$

To use the fast spherical transform, Equation 2.2 is rewritten

$$a_l^m = \int_0^{2\pi} f_l^m(v) e^{-imv} dv \tag{2.5}$$

where

$$f_l^m(\phi) = \int_0^{\pi} r(u,v) P_l^m(cosu) sinudu \tag{2.6}$$

Equation 2.6 is calculated with the use of the Gauss-Legendre quadrature, then it is obtained by

$$f_l^m(v) = \sum_{k=1}^{N_u} r(u_k,v) P_l^m(cosu_k)\omega_k \tag{2.7}$$

where u_k and ω_k are Legendre-Gauss points and weights, respectively. Once the $f_l^m(v)$ are calculated from 2.7, the values of a_l^m can be obtained using inverse fast Fourier transform over m in Equation 2.5.

2.2 1-hole Object

Representation

When an object has one boundary, an object is classified as an 1-hole object. The radius of a 1-hole object, which can be mapped as a function of parameterization variables u and v, is approximated with the one-sided Jacobi polynomial with Fourier series with their coefficients. The spherical radius of a 1-hole objects can be expressed as,

$$r\,(u, v) = \sum_{l=0}^{L} \sum_{m=-N}^{N} a_l^m W_l^m(u, v), \tag{2.8}$$

with spectral coefficients a_l^m, where

$$W_l^m(u, v) = V_l^m(u) e^{imv} \tag{2.9}$$

and $V_l^m(u)$ is defined as

$$V_l^m(u) = u^{|m|} P_l^{(0,|m|)}(2u^2 - 1), \tag{2.10}$$

where $P_l^{(\alpha,\beta)}(s)$ is Jacobi polynomials. We call $V_l^m(u)$ as one-sided Jacobi polynomials. Note that here the radius of a 1-hole object can be spherical radius or cylindrical radius. Let's define s as $s = 2u^2 - 1$, where s is Legendre-Gauss points. Then, parameterization variable u is defined as

$$u = \sqrt{\frac{s+1}{2}} \tag{2.11}$$

whose the range is $0 \leq u \leq 1$. The parameter v is equally spaced from 0 to 2π. Any mappings can be used in u and v, but if an object is a star-shaped object and spherical radius of a 1-hole object is approximated, a linear mapping between u and latitudinal angle θ, defined as,

$$\theta = \frac{\pi}{2} u \tag{2.12}$$

and equal azimuthal angle works well for v.

Calculation of Jacobi polynomials

The one-sided Jacobi polynomials is calculated by the three term recursion relation

$$\begin{aligned} &2l(l+\alpha+\beta)(2l+\alpha+\beta-2)P_l^{(\alpha,\beta)}(s) \\ &= (2l+\alpha+\beta-1)\{(2l+\alpha+\beta)(2l+\alpha+\beta-2)s + \alpha^2 - \beta^2\}P_{l-1}^{(\alpha,\beta)}(s) \\ &\qquad\qquad - 2(l+\alpha-1)(l+\beta-1)(2l+\alpha+\beta)P_{l-2}^{(\alpha,\beta)}(s) \end{aligned} \tag{2.13}$$

with the initial starting values

$$P_0^{(\alpha,\beta)}(s) = 1, \;\; P_1^{(\alpha,\beta)}(s) = \frac{1}{2}\{(\alpha-\beta)(\alpha+\beta+2)s\} \tag{2.14}$$

In our 1-hole object basis function, α and β are chosen as $\alpha = 0$ and $\beta = |m|$ because Matsushima and Marcus [36] showed spectral accuracy of these functions and derived efficient linear operators, which we can use in solving PDEs later. In other words, by choosing the same values of α and β with Matsushima and Marcus [36], we will use well developed transformation and linear operators described in [36]. Under this setting, α and β can be substituted to 0 and $|m|$, respectively. Then, Equations 2.13 and 2.14 become

$$\begin{aligned}&2l(l+|m|)(2l+|m|-2)P_l^{(0,|m|)}(s) = \\ &(2l+|m|-1)\{(2l+|m|)(2l+|m|-2)s - m^2\}P_{l-1}^{(0,|m|)}(s) \\ &\qquad\qquad - 2(l-1)(l+|m|-1)(2l+|m|)P_{l-2}^{(0,|m|)}(s)\end{aligned} \tag{2.15}$$

and

$$P_0^{(0,|m|)}(s) = 1, \;\; P_1^{(0,|m|)}(s) = -\frac{|m|(|m|+2)s}{2} \tag{2.16}$$

Equation 2.15 can be rewritten as

$$\begin{aligned}P_l^{(0,|m|)}(s) = &\frac{(2l+|m|-1)(2l+|m|)s}{2l(l+|m|)}P_{l-1}^{(0,|m|)}(s) \\ &- \frac{(2l+|m|-1)m^2P_{l-1}^{(0,|m|)}(s) + 2(l-1)(l+|m|-1)(2l+|m|)P_{l-2}^{(0,|m|)}(s)}{2l(l+|m|)(2l+|m|-2)}\end{aligned} \tag{2.17}$$

with simple mathematical expansion. This is a three-term recursion relation of Jacobi polynomials $P_l^{(\alpha,\beta)}(s)$ when $\alpha = 0$ and $\beta = |m|$. The first two terms start from Equation 2.16 and then last terms are obtained from this three-term recursion relation.

Transformation of 1-hole basis functions

In obtaining spectral coefficients of an 1-hole object, orthogonal property of 1-hole basis functions are used. Polynomials V_l^m have the orthogonality such that

$$\int_0^1 V_l^m(u)V_n^m(u)udu = h_{lm}\delta_{ln}, \tag{2.18}$$

where

$$h_{lm} = \frac{1}{4l+2m+2}. \tag{2.19}$$

If we multiply $uW_n^{k*}(u,v)$ in Equation 2.3 and integrating respect to u and v over 0 to 1 and 0 to 2π, where superscript * means complex conjugate, it gives

$$\int_0^1\int_0^{2\pi} r(u,v)W_n^{k*}(u,v)udvdu = \int_0^1\int_0^{2\pi} a_l^m V_l^m(u)V_n^k(u)e^{i(m-k)v}udvdu. \tag{2.20}$$

Rearranging the right hand side term of Equation 2.20 allows to have

$$\int_0^1 \int_0^{2\pi} r(u,v) W_n^{k^*}(u,v) u dv du = a_l^m \int_0^1 V_l^m(u) V_n^k(u) \Big(\int_0^{2\pi} e^{i(m-k)v} dv \Big) u du. \tag{2.21}$$

Since

$$\int_0^{2\pi} e^{i(m-k)v} dv = 2\pi \delta_{mk}, \tag{2.22}$$

Equation 2.21 can be rewritten as

$$\int_0^1 \int_0^{2\pi} r(u,v) W_n^{m*}(u,v) u dv du = 2\pi a_l^m \int_0^1 V_l^m(u) V_n^m(u) u du. \tag{2.23}$$

Since the integral of the right hand side of Equation 2.23 is equal to $h_{lm}\delta_{ln}$, we can finally obtain

$$a_l^m = \frac{1}{2\pi h_{lm}} \int_0^{2\pi} \int_0^1 r(u,v) W_l^{m*}(u,v) u du dv. \tag{2.24}$$

To use the fast spherical transform, Equation 2.24 is rewritten

$$a_l^m = \frac{1}{2\pi h_{lm}} \int_0^{2\pi} f_l^m(v) e^{-imv} dv, \tag{2.25}$$

where

$$f_l^m(v) = \int_0^1 r(u,v) V_l^m(u) u du. \tag{2.26}$$

Once Equation 2.26 is calculated with Legendre-Gauss quadrature method, Equation 2.25 is computed with fast Fourier transform.

Derivatives of 1-hole basis functions

For the later usage, here, we want to derive the first and second derivatives of 1-hole basis functions. To derive derivative formulas for 1-hole basis functions, we start from the derivative relation of one-sided Jacobi polynomial, given as

$$\frac{d^k}{ds^k} P_l^{(\alpha,\beta)}(s) = \frac{\Gamma(l+k+\alpha+\beta+1)}{2^k \Gamma(l+\alpha+\beta+1)} P_{l-k}^{(\alpha+k,\beta+k)}(s), \tag{2.27}$$

where $\Gamma(n) = (n-1)!$. After substitute s into $2u^2-1$, expanding Equation 2.27 for the first derivative of Jacobi polynomials gives

$$\frac{d}{du} P_l^{(\alpha,\beta)}(2u^2-1) = 2u(l+\alpha+\beta+1) P_{l-1}^{(\alpha+1,\beta+1)}(2u^2-1) \tag{2.28}$$

Differentiating Equation 2.28 respect to u gives

$$\begin{aligned}\frac{d^2}{du^2}P_l^{(\alpha,\beta)}(2u^2-1) = 4u^2(l+\alpha+\beta+1)(l+\alpha+\beta+2)P_{l-2}^{(\alpha+2,\beta+2)}(2u^2-1)\\ +2(l+\alpha+\beta+1)P_{l-1}^{(\alpha+1,\beta+1)}(2u^2-1).\end{aligned} \tag{2.29}$$

By substituting respectively α and β to 0 and $|m|$, Equations 2.28 and 2.29 are simplified as

$$\frac{d}{du}P_l^{(0,|m|)}(2u^2-1) = 2u(l+|m|+1)P_{l-1}^{(1,|m|+1)}(2u^2-1) \tag{2.30}$$

$$\begin{aligned}\frac{d^2}{du^2}P_l^{(0,|m|)}(2u^2-1) = 4u^2(l+|m|+1)(l+|m|+2)P_{l-2}^{(2,|m|+2)}(2u^2-1)\\ +2(l+|m|+1)P_{l-1}^{(1,|m|+1)}(2u^2-1)\end{aligned} \tag{2.31}$$

Since $W_l^m(u,v)$ is defined as

$$W_l^m(u,v) = u^{|m|}P_l^{(0,|m|)}(2u^2-1)e^{imv}, \tag{2.32}$$

the first derivative of $W_l^m(u,v)$ can be obtained by differentiating Equation 2.32 respect to u,

$$\frac{d}{du}W_l^m(u,v) = \Big(|m|u^{|m|-1}P_l^{(0,|m|)}(2u^2-1) + u^{|m|}\frac{d}{du}P_l^{(0,|m|)}(2u^2-1)\Big)e^{imv}. \tag{2.33}$$

Putting Equation 2.30 into Equation 2.33 gives

$$\frac{d}{du}W_l^m(u,v) = \Big(|m|u^{|m|-1}P_l^{(0,|m|)}(2u^2-1)+u^{|m|+1}(l+|m|+1)P_{l-1}^{(1,|m|+1)}(2u^2-1)\Big)e^{imv}. \tag{2.34}$$

Similarly, differentiating Equation 2.32 twice respect to u allows to derive the second derivative of $W_l^m(u,v)$

$$\begin{aligned}\frac{d^2}{du^2}W_l^m(u,v) = \Big(|m|(|m|-1)u^{|m|-2}P_l^{(0,|m|)}(2u^2-1)\\ +|m|u^{|m|-1}\frac{d}{du}P_l^{(0,|m|)}(2u^2-1)+u^{|m|}\frac{d^2}{du^2}P_l^{(0,|m|)}(2u^2-1)\Big)e^{imv}.\end{aligned} \tag{2.35}$$

Then, substituting Equation 2.30 and 2.31 into Equation 2.35 gives

$$\begin{aligned}\frac{d^2}{du^2}W_l^m(u,v) = u^{|m|-2}\Big(|m|(|m|-1)P_l^{(0,|m|)}(2u^2-1)+(|m|+2)u^2(l+|m|+1)P_{l-1}^{(1,|m|+1)}(2u^2-1)\\ +4u^4(l+|m|+1)(l+|m|+2)P_{l-2}^{(2,|m|+2)}(2u^2-1)\Big)e^{imv}\end{aligned} \tag{2.36}$$

Here, polynomials $P_{l-1}^{(1,|m|+1)}(2u^2-1)$ and $P_{l-2}^{(2,|m|+2)}(2u^2-1)$ can be calculated from the three-term recursion relationship of Jacobi polynomials.

Linear Operators of 1-hole basis functions

Suppose any function g can be represented with one-hole basis function with spectral coefficients c_l^m, written as

$$g(u,v) = \sum_{l=0}^{L} \sum_{m=-N}^{N} c_l^m W_l^m(u,v). \tag{2.37}$$

Let's consider any linear operator L can be defined as

$$Lg(u,v) = \sum_{l=0}^{L} \sum_{m=-N}^{N} d_l^m W_l^m(u,v) \tag{2.38}$$

What we want to do in this moment is finding d_l^m in terms of c_l^m at specified operator L. First, let's consider when $L = \nabla^2$ where ∇^2 is defined as

$$\nabla^2 = \frac{1}{u}\frac{\partial}{\partial u}(u\frac{\partial}{\partial u}) + \frac{1}{u^2}\frac{\partial}{\partial v^2}. \tag{2.39}$$

This operator will be used to impose geometric constraints on an 1-hole object. Taking ∇^2 in Equation 2.37 gives

$$\nabla^2 g(u,v) = \sum_{l=0}^{L} \sum_{m=-N}^{N} c_l^m \nabla^2 W_l^m(u,v). \tag{2.40}$$

In [53], Verkley showed

$$\nabla^2 W_l^m(u,v) = \sum_{n=0}^{l-1} 4(m+2n+1)(l-n)(n+m+l+1)W_n^m(u,v) \tag{2.41}$$

By substituting 2.41 into 2.40, we obtain

$$\nabla^2 g(u,v) = \sum_{l=0}^{L} \sum_{m=-N}^{N} 4c_l^m \sum_{n=0}^{l-1}(m+2n+1)(l-n)(n+m+l+1)W_n^m(u,v) \tag{2.42}$$

By changing summation range of indexes, Equation 2.42 can be rewritten as

$$\nabla^2 g(u,v) = \sum_{n=0}^{L} \sum_{m=-N}^{N} 4(m+2n+1) \sum_{l=n+1}^{N} (l-n)(n+m+l+1)c_l^m W_n^m(u,v). \tag{2.43}$$

Without loss the generality, we can exchange index l and n in Equation 2.43, giving

$$\nabla^2 g(u,v) = \sum_{l=0}^{L} \sum_{m=-N}^{N} d_l^m W_l^m(u,v), \tag{2.44}$$

where

$$d_l^m = 4(m+2l+1)\sum_{n=l+1}^{N}(n-l)(l+m+n+1)c_n^m. \tag{2.45}$$

Therefore, $\nabla^2 g$ of any function g is easily obtained from spectral coefficients of g using Equation 2.45.

If we consider when $L = u^2$, recurrence formula in [36],

$$d_l^m = \frac{l(l+m)}{(2l+m-1)(2l+m)}c_{l-1}^m + \left(1+\frac{m^2}{2(2l+m)(2l+m+2)}\right)c_l^m + \frac{(l+1)(l+m+1)}{(2l+m+3)(2l+m+2)}c_{l+1}^m, \tag{2.46}$$

are utilized to compute $u^2 g(u,v)$, which is approximated as

$$u^2 g(u,v) = \sum_{l=0}^{L}\sum_{m=-N}^{N} d_l^m W_l^m(u,v), \tag{2.47}$$

where g is an arbitrary function defined in Equation 2.37.

2.3 2-hole object

Transformation of 2-hole basis functions

When an object topologically belongs to a sphere with two open boundaries, an object is classified as a 2-hole object. The cylindrical radius of 2-hole objects mapped as a function of parameterization variables u and v is approximated with Chebyshev polynomial with Fourier series with their coefficients. The cylindrical radius of 2-hole objects can be expressed as,

$$r(u,v) = \sum_{l=0}^{L}\sum_{m=-N}^{N} a_l^m T_l(u)e^{imv} \tag{2.48}$$

Here, T_l is the Chebyshev polynomials. For a star-shaped object, simple parameterization such that u is linear mapping of the height of an object and v is the azimuthal angle works well. To compute spectral coefficient a_l^m, multiply Equation 2.48 by $T_n(u)e^{-ikv}\frac{1}{\sqrt{1-u^2}}$ and integrate respect to u and v over -1 to 1 and 0 to 2π, and then, it gives

$$\begin{aligned}&\int_0^{2\pi}\int_{-1}^{1} r(u,v)T_n(u)e^{-ikv}\frac{1}{\sqrt{1-u^2}}dudv \\ &= \int_0^{2\pi}\int_{-1}^{1}\sum_{l=0}^{L}\sum_{m=-N}^{N} a_l^m T_l(u)T_n(u)\frac{1}{\sqrt{1-u^2}}e^{i(m-k)v}dudv\end{aligned} \tag{2.49}$$

By rearranging terms, Equation 2.76 can be rewritten as

$$\begin{aligned}\sum_{l=0}^{L}\sum_{m=-N}^{N} a_l^m \int_{-1}^{1} T_l(u)T_n(u)\frac{1}{\sqrt{1-u^2}}du \int_0^{2\pi} e^{i(m-k)v}dv \\ = \int_0^{2\pi}\Big(\int_{-1}^{1} r(u,v)T_n(u)\frac{1}{\sqrt{1-u^2}}du\Big)e^{-ikv}dv\end{aligned} \tag{2.50}$$

It should be noticed that

$$\int_0^{2\pi} e^{i(m-k)v}dv = 2\pi\delta_{mk} \tag{2.51}$$

and

$$\int_{-1}^{1} T_l(u)T_n(u)\frac{1}{\sqrt{1-u^2}}du = c_l\delta_{ln} \tag{2.52}$$

where

$$c_l = \begin{cases} \pi, \ l=0 \\ \frac{\pi}{2}, \ l\neq 0 \end{cases} \tag{2.53}$$

Equations 2.51 and 2.52 allow to deduce Equation 2.50 further. By using these relations, we finally obtain a_l^m, defined as

$$a_l^m = \frac{1}{2\pi c_l}\int_0^{2\pi} f_l(v)e^{-imv}dv, \tag{2.54}$$

where

$$f_l(v) = \int_{-1}^{1} r(u,v)T_l(u)\frac{1}{\sqrt{1-u^2}}du \tag{2.55}$$

Once Equation 2.55 is calculated from Chebyshev-Gaussian quadrature rule, a_l^m is obtained in Equation 2.54 using fast Fourier transform.

Derivatives of Chebyshev polynomials

The first derivative of Chebyshev polynomials is derived from its definition. The l^{th} order Chebyshev polynomial is defined as

$$T_l(u) = cos(l\theta), \tag{2.56}$$

where $u = cos(\theta)$. If $T_l(u)$ is differentiated in u, it is written as

$$\frac{d}{du}T_l(u) = l\frac{sin(l\theta)}{sin\theta}. \tag{2.57}$$

Due to the fact that $sin(l\theta)$ can be decomposed into

$$sin(l\theta) = sin((l-1)\theta)cos(\theta) + cos((l-1)\theta)sin(\theta), \tag{2.58}$$

if we use Equation 2.58 repeatedly, the right hand side of Equation 2.57 can be decomposed into summations of low order terms. If we generalize the decomposed terms, Equation 2.57 can be classified into two cases depending on whether l is even or odd, given as

$$l\frac{sin(l\theta)}{sin\theta} = 2l\Big(cos[(l-1)\theta] + cos[(l-3)\theta] + \cdots \begin{cases} +cos(3\theta) + cos(\theta), & l\ even \\ +cos(4\theta) + cos(2\theta) + \frac{1}{2}, & l\ odd \end{cases}\Big) \quad (2.59)$$

From the definition of Chebyshev polynomials, Equation 2.59 becomes

$$l\frac{sin(l\theta)}{sin\theta} = 2l\Big(T_{l-1}(u) + T_{l-3}(u) + \cdots \begin{cases} +T_3(u) + T_1(u), & l\ even \\ +T_4(u) + T_2(u) + \frac{1}{2}, & l\ \ odd \end{cases}\Big). \quad (2.60)$$

It is straightforward to show that Equation 2.60 is equivalent with

$$\frac{dT_l(u)}{du} = 2l \sum_{\substack{l+p=odd \\ p=0}}^{l-1} T_p(u)\alpha_p, \quad (2.61)$$

where

$$\alpha_p = \begin{cases} \frac{1}{2},\ p = 0 \\ 1,\ otherwise \end{cases} \quad (2.62)$$

If we define an arbitrary function $f(u)$ approximated by Chebyshev polynomials, a function $f(u)$ can be written as

$$f(u) = \sum_{l=0}^{N} c_l T_l(u). \quad (2.63)$$

Then, the first derivative of $f(u)$ respect to u is

$$\frac{df(u)}{du} = \sum_{l=0}^{N} c_l \frac{dT_l(u)}{du} \quad (2.64)$$

Substituting Equation 2.61 into Equation 2.64 gives

$$\frac{df(u)}{du} = \sum_{l=0}^{N} c_l \Big(2l \sum_{\substack{l+p=odd \\ p=0}}^{l-1} T_p(u)\alpha_p\Big) \quad (2.65)$$

By rearranging indexes and changing their ranges, we obtain

$$\frac{df(u)}{du} = \sum_{p=0}^{N} \Big(2\alpha_p \sum_{\substack{l=p+1 \\ l+p=odd}}^{N} lc_l\Big) T_p(u) \quad (2.66)$$

After index l and p are interchanged, the spectral coefficient c_l^m of a function $f(u)$ is related with the the spectral coefficient d_l^m of the first derivative of $f(u)$ such that

$$\frac{df(u)}{du} = \sum_{l=0}^{N} d_l T_l(u), \tag{2.67}$$

where

$$d_l = 2\alpha_l \sum_{\substack{p=l+1 \\ p+l=odd}}^{N} pc_p. \tag{2.68}$$

With the matrix representation, a matrix $\mathbf{d} = [d_0, d_1, \cdots, d_N]^\intercal$ is obtained by $\mathbf{d} = \mathbf{Dc}$, where $\mathbf{c} = [c_0, c_1, \cdots, c_N]^\intercal$ and $(i, j) - th$ components of $\mathbf{D}$, defined as D_{ij}, is calculated from

$$D_{ij} = \alpha_i \sum_{\substack{j=i+1 \\ i+j=odd}}^{N} j, \tag{2.69}$$

where $i = 0,\ 1, \cdots,\ N+1$ and $j = 0,\ 1, \cdots,\ N+1$. This matrix $\mathbf{D}$ is called differentiation matrix of Chebyshev polynomials.

Linear operator of 2-hole basis functions

Suppose any function g can be represented with 2-hole basis function with spectral coefficients c_l^m, written as,

$$g(u, v) = \sum_{l=0}^{L} \sum_{m=-N}^{N} c_l^m T_l(u) e^{imv} \tag{2.70}$$

Also, let's consider any linear operator L which can be defined as

$$Lg(u, v) = \sum_{l=0}^{L} \sum_{m=-N}^{N} d_l^m T_l(u) e^{imv} \tag{2.71}$$

Then, we want to find a relation between coefficients c_l^m and d_l^m. If we consider $L = \nabla^2$, defined as

$$\nabla^2 = \frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2}, \tag{2.72}$$

$\nabla^2 g(u, v)$ becomes

$$\nabla^2 g(u, v) = \sum_{l=0}^{L} \sum_{m=-N}^{m=N} c_l^m \Big(\frac{\partial^2}{\partial u^2} - m^2\Big) T_l(u) e^{imv}. \tag{2.73}$$

If we equate Equation 2.71 and Equation 2.73, we can seek a relation between spectral coefficients of a 1-hole object, given as

$$d_l^m = (D_{lk}^2 - m^2 \delta_{lk}) c_k^m, \tag{2.74}$$

where D_{lk} is a $(l, k)^{th}$ component of the differentiation matrix defined in Equation 2.69.

2.4 3-hole Object

Representation

When an object has four boundaries, an object is classified as a patch. The height of an object z which is a function of u an v is represented as

$$z(u,v) = \sum_{l=0}^{L}\sum_{m=0}^{N} a_l^m T_l(u) T_m(v), \tag{2.75}$$

where u and v are Chebyshev-Gauss Lobatto points and a_l^m is spectral coefficients of double Chebyshev polynomials. To compute double Chebyshev coefficients a_l^m, we multiply $T_n(u)T_k(u)\frac{1}{\sqrt{1-u^2}}\frac{1}{\sqrt{1-v^2}}$ to Equation 2.75 and integrate respect to u and v over -1 to 1. Then, it gives

$$\begin{aligned} &\int_{-1}^{1}\int_{-1}^{1} z(u,v)T_n(u)T_k(u)\frac{1}{\sqrt{1-u^2}}\frac{1}{\sqrt{1-v^2}}dudv \\ &= \int_{-1}^{1}\int_{-1}^{1}\sum_{l=0}^{L}\sum_{m=0}^{N} a_l^m T_l(u)T_n(u)\frac{1}{\sqrt{1-u^2}}T_m(u)T_k(u)\frac{1}{\sqrt{1-v^2}}dudv \end{aligned} \tag{2.76}$$

Due to the orthogonality of Chebyshev polynomials, spectral coefficients a_l^m are calculated from

$$a_l^m = \frac{1}{4\pi^2 c_l c_m}\int_{-1}^{1} f_l(v)T_m(v)\frac{1}{\sqrt{1-v^2}}dv, \tag{2.77}$$

where

$$f_l(v) = \int_{-1}^{1} r(u,v)T_l(u)\frac{1}{\sqrt{1-u^2}}du \tag{2.78}$$

Here, the definition of c_l and c_m is equal to the one shown in Equation 2.53.

Chebyshev differentiation matrix for collocation method

When a patch is parameterized by double Chebyshev polynomials, it is mapped to a square domain. When geometric constraints are imposed on a patch whose surface is mapped into a square domain, constraints should be physically reasonable. For example, let say a patch's height is mapped by its functional parameters (u, v), where u, v are Chebyshev grid points ranging from $-1 \leq u, v \leq 1$. Further suppose the heights of a path are demanded to be $f_{\pm1}(v)$ when $u = \pm1$ and $g_{\pm1}(u)$ when $v = \pm1$ respectively. Then, $f_{\pm1}(\pm1)$ should be equal to $g_{\pm1}(\pm1)$ because a patch should have the same height at the corners. Besides the height matching conditions at the corner, if we consider higher derivative constraints such as slope and curvature, there are more dependent constraints which have to be considered at the corner. Without considering these conditions at the corners, solving PDE with arbitrary geometric constraints can be an over-constrained problem.

When PDEs are solved to impose geometric constraints on a patch, both tau method and collocation method can be used. However, we focus more on collocation method than tau method because collocation method has an advantage in dealing with an over-constrained problem. Since tau method is so global, a constrained shape becomes blows-up or very oscillatory when an over-constrained problem is faced. However, if collocation method is used in solving PDEs, oscillation is only confined near the corners of a patch, and then, manipulating the shape to remove the oscillation is quite easy. Thus, in this work, we will use collocation method in solving PDEs for imposing constraints on a patch. Here we will describe how to obtain a differentiation matrix for a collocation method and describe how to solve PDEs based on collocation method in the next chapter.

Suppose $p(x)$ is a given function and $\mathbf{p} = [p_0,\ p_1,\ \cdots,\ p_N]^\intercal$ is a column matrix whose components are values of $p(x)$ on Chebyshev grid points, defined as,

$$x_k = cos\Big(\frac{k\pi}{N}\Big), \quad j = 0,\ 1,\ \cdots,\ N. \tag{2.79}$$

Further suppose $q(x)$ is derivative of $p(x)$ respect to x and $\mathbf{q} = [q_0,\ q_1,\ \cdots,\ q_N]^\intercal$ is a column matrix whose components are values of $q(x)$ on Chebyshev grid points. Then, we want to find a differentiation matrix $\mathbf{D_p}$ for a collocation method, which satisfies the following equation.

$$\mathbf{q} = \mathbf{D_p}\mathbf{p}. \tag{2.80}$$

Here, $\mathbf{D_p}$ is $(N+1) \times (N+1)$ and column matrices $\mathbf{p}$ and $\mathbf{q}$ are $(N+1) \times 1$. When $N = 1$, Chebyshev grid points become $x_0 = 1$ and $x_1 = -1$, and then, Lagrangian form of polynomial passing points $(1, p_0)$ and $(-1, p_1)$ are

$$p(x) = \frac{1}{2}(1+x)p_0 + \frac{1}{2}(1-x)p_1 \tag{2.81}$$

The derivative of Equation 2.81 respect to x becomes

$$q(x) = \frac{1}{2}p_0 - \frac{1}{2}p_1, \tag{2.82}$$

Therefore, differentiation matrix when $N = 1$ is

$$\mathbf{D_p} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix} \tag{2.83}$$

When $N = 2$, collocation points $x_0 = 1$, $x_1 =$ and $x_1 =$, and then, Lagrangian form of polynomial passing points $(1, p_0)$, $(0, p_1)$ and $(1, p_2)$ are

$$p(x) = \frac{1}{2}x(x+1)p_0 + (1+x)(1-x)p_1 + \frac{1}{2}x(x-1)p_2 \tag{2.84}$$

The derivative of Equation 2.84 respect to x becomes

$$q(x) = \Big(x + \frac{1}{2}\Big)p_0 - 2xp_1 + \Big(x - \frac{1}{2}\Big)p_2, \tag{2.85}$$

Therefore, differentiation matrix when $N = 2$ is

$$\mathbf{D_p} = \begin{bmatrix} \frac{3}{2} & -2 & \frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 2 & -\frac{3}{2} \end{bmatrix} \tag{2.86}$$

If we generalize it, we can finally obtain the differentiation matrix $\mathbf{D_p}$, given as

$$\mathbf{D_p} = [d_{ij}] = \begin{cases} d_{ij} = \frac{c_i}{c_j}\frac{(-1)^{i+j}}{(x_i - x_j)}, & 0 \leq i, j \leq N,\ i \neq j \\ d_{ii} = -\frac{x_i}{2(1-x_i^2)}, & 1 \leqq i \leqq N-1 \\ d_{00} = -d_{NN} = \frac{2N^2+1}{6} & \end{cases} \tag{2.87}$$

2.5 Morphing objects

Spectral morphing

When the surfaces of two or more 0-, 1-, 2-holed objects, sub-objects or patches are represented by a truncated spectral sum, morphing is particularly easy and robust. A morphed object, sub-object or patch is created from N baseline objects, patches or sub-objects by creating a new set of spectral coefficients from the baseline coefficients. For example, spectral coefficients of a morphed 0-hole object b_l^m from N baseline 0-hole objects is calculated by

$$b_l^m = \sum_{k=1}^{N} \omega_k \ (a_l^m)_k, \tag{2.88}$$

where ω_k is the weight of the k^{th} baseline 0-hole object and $(a_l^m)_k$ is spectral coefficients of the k^{th} 0-hole object. Then, the radius of a morphed 0-hole object, $r_m(u, v)$, is computed from

$$r_m(u, v) = \sum_{l=0}^{L} \sum_{m=-l}^{l} b_l^m Y_l^m(u, v). \tag{2.89}$$

It is often useful, but not necessary to make the sum of the weights equal to unity. For $N = 2$, if $0 \leq \omega_1 \leq 1$ and $\omega_2 = 1 - \omega_1$, the morph is an interpolation between the two objects (or patches or sub-objects). If $\omega_1 < 0$ or $\omega_1 > 1$, the morph is an extrapolation. The definitions of interpolation and extrapolation can be extended to the case $N > 2$. Because this method of morphing is so robust, extrapolation can easily be done without the method breaking down. Extrapolations are particularly useful for creating radical new designs of objects, as shown in Figure 2.2.

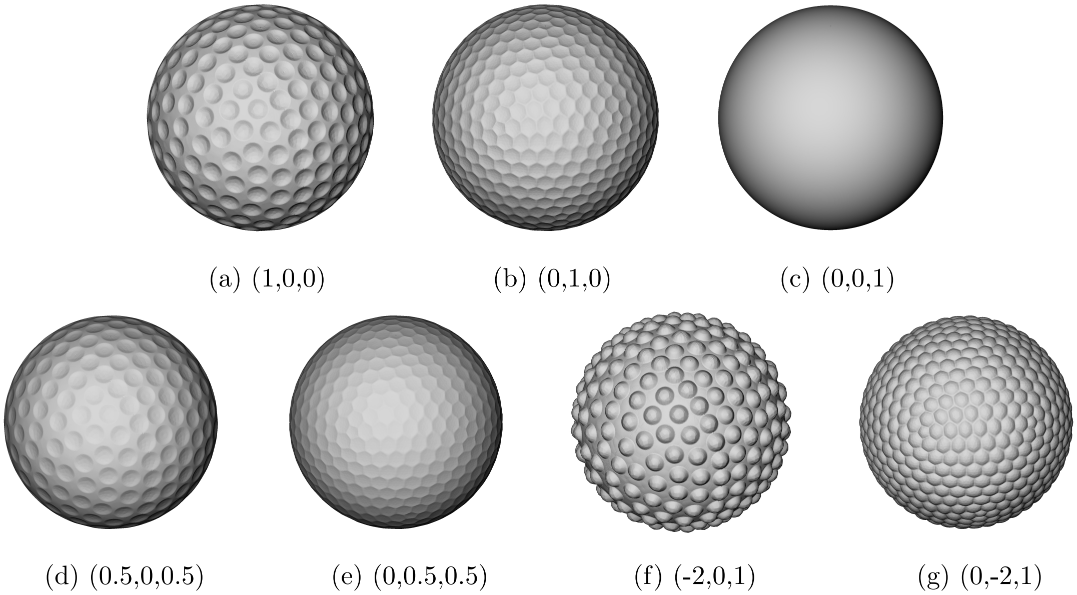
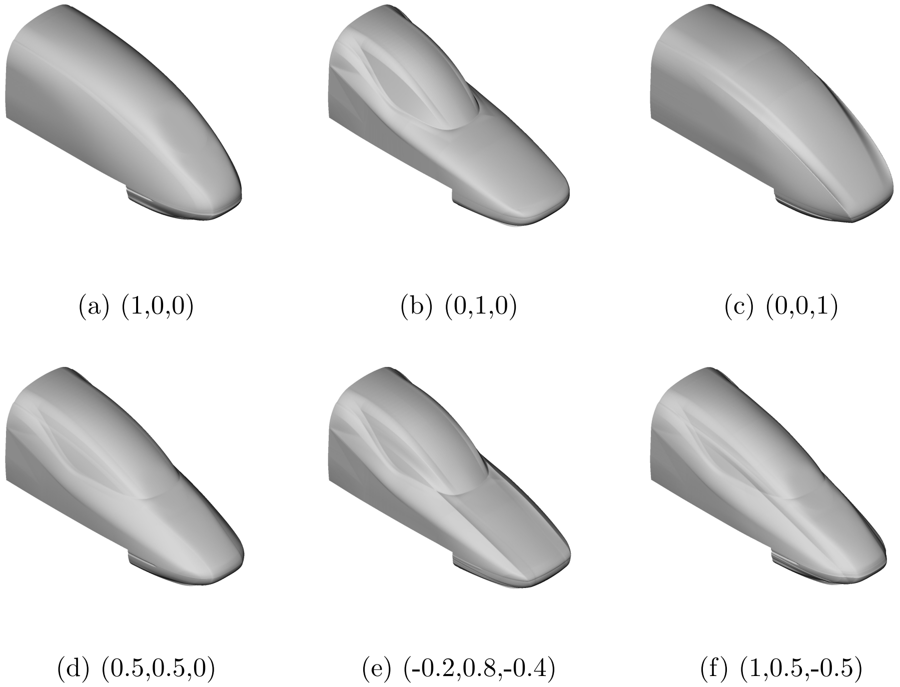


Figure 2.2: Morphing of golf balls: (a) the baseline circular dimpled golf ball, (b) the baseline hexagonal dimpled golf ball, (c) the baseline smooth golf ball. A morphed interpolation of one of the dimpled baseline balls and the smooth ball will produce a ball with shallower dimples, as shown in (d). A morphed extrapolation of one of the dimpled balls and the smooth ball will produce a ball with deeper dimples as shown in (e) and a ball with bumps, rather than dimples, as shown in (f) and (g). Weights of each golf ball are given in its own subcaption.

In the figure, three baseline golf balls, plotted in (a), (b) and (c) , whose shapes are represented as truncated sums of 0-hole objects are morphed. Among them, two baseline golf balls are dimpled, and one is smooth. The morphed golf balls are shown in panels from (d) to (g). Two morphed golf balls, shown in (d) and (e), that are interpolations of the baseline shapes are dimpled and two morphed golf balls, shown in (f) and (g), that are extrapolations of a dimpled golf ball and a smooth golf ball do not have dimples, rather they have bumps.

Another example of morphed objects is presented in Figure 2.3. Three actual train engines (without the wheels and undercarriages), which are 0-hole objects and plotted in (a), (b) and (c), are represented with truncated spherical harmonics. We use these as the three baseline objects in our morphing. An example of morphed train engine made by interpolation among the 3 baseline trains is plotted in (d) and two examples of morphed train engines made by extrapolation of the baseline trains are plotted in (e) and (f). As illustrated in Figures 2.2 and 2.3, design-by-morphing works well for extrapolation of baseline objects as well as for interpolation.



(a) (1,0,0)

(b) (0,1,0)

(c) (0,0,1)

(d) (0.5,0.5,0)

(e) (-0.2,0.8,-0.4)

(f) (1,0.5,-0.5)

Figure 2.3: Morphed train engines (i.e., the first car of a high-speed train). In (a) (b) and (c), three baseline train engines are represented as truncated sums of 0-hole objects. The morphed shape (d) is created by interpolating the three baseline objects, while the morphed shapes in (e) and (f) are extrapolations of the baseline objects. Weights of each train engin are given in its own subcaption.

Representation comparison of spherical harmonics and eigenfunctions of Laplace-Betrami operator

One main reason we introduce and use these basis functions in design-by-morphing is to represent the surface of an object accurately but efficiently, and to take advantages of exponential convergence rate and fast forward and backward transform of our spherical basis functions. These basis functions have advantages over currently using spherical basis functions with respect to representation accuracy and computation time to obtain basis functions. To show these advantages, we calculate the representation accuracy and computation time in representation the surface of a synthetic object and compare these values between one of our basis function, spherical harmonics, to currently commonly used spectral basis function, eigenfunctions of Laplace-Beltrami operator. The surface of a synthetic object is given as

$$\frac{x^2}{a^2 s(z)^2} + \frac{y^2}{b^2 s(z)^2} + \frac{z^2}{c^2} = 1, \tag{2.90}$$

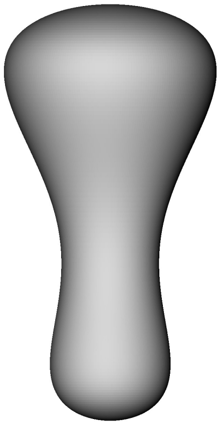


Figure 2.4: A Synthetic object

where $a = 0.8,\ b = 0.9,\ c = 1.0$ and

$$s(z) = 1 - 0.7cos\Big[\frac{\pi}{2}\Big(\frac{z}{c} + 0.2\Big)\Big] \tag{2.91}$$

To quantitatively compare the representation accuracy, we define error E, given as,

$$E = \Big[\frac{1}{N_v}\sum_{i=1}^{N_v}\Big(\frac{\tilde{x_i}^2}{a^2 s(\tilde{z}_i)^2} + \frac{\tilde{y_i}^2}{b^2 s(\tilde{z}_i)^2} + \frac{\tilde{z_i}^2}{c^2} - 1\Big)^2\Big]^{\frac{1}{2}}, \tag{2.92}$$

where N_v is number of vertices of the synthetic object and $\tilde{x}_i$, $\tilde{y}_i$, and $\tilde{z}_i$ are positions of vertices approximated by using spherical harmonic or eigenfunctions of LBO. To compute eigenfunctions of LBO, we use knowledge and algorithm described in [44].

The comparison results are presented in Table 2.1. The representation accuracy is slightly better (at $N = 8, 16, 32$) when the surface of the synthetic object is approximated by using eigenfunctions of LBO than using spherical harmonics, but both basis functions show spectral accuracy. The reason the representation accuracy using eigenfunctions of LBO is trapped around the order of 10^{-6} at high number of modes used (at $N = 64, 128$) is due to the fact eigenfunctions of LBO are calculated under the single precision accuracy. Thus, the order of 10^{-6} accuracy is the highest representation accuracy we can expect from using eigenfunctions of LBO.

The more important fact is huge discrepancy of calculation time between two methods. As an example, we represent the surface of a hexagonal dimpled golf ball (shown in Figure 2.5b) by using two methods and compare the calculation time. The calculation time required to compute spherical harmonics and eigenfunctions of Laplace-Betrami operator are presented in Table 2.2 and the shape of a golf ball approximated by these two basis functions are plotted in Figure 2.5. As Figure 2.5 shows, it is obvious to claim that the number of modes

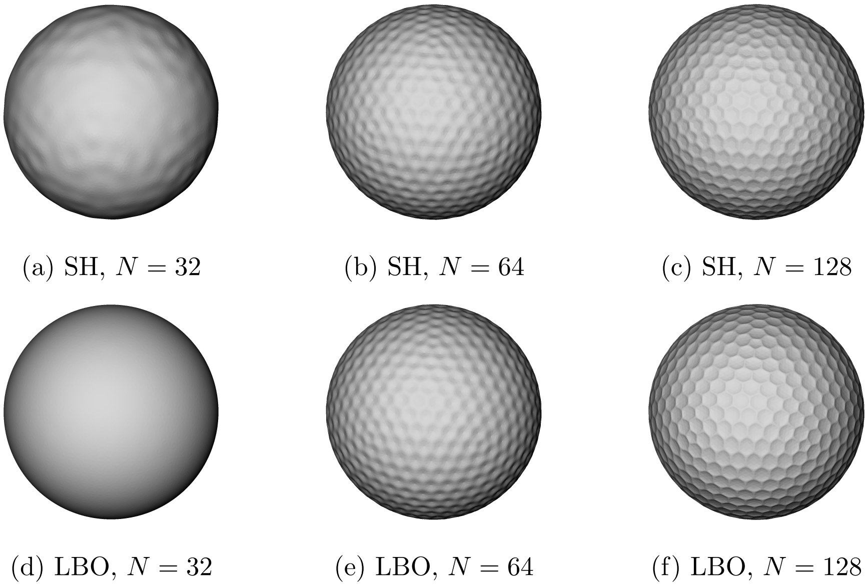
N	error	
	spherical harmonics	eigenfunctions of LBO
8	1.47×10^{-2}	9.10×10^{-3}
16	1.17×10^{-3}	3.02×10^{-4}
32	1.29×10^{-5}	5.98×10^{-6}
64	2.44×10^{-9}	1.79×10^{-6}
128	7.22×10^{-14}	.

Table 2.1: Representation comparison of a synthetic object using spectral harmonics and eigenfunctions of Laplace-Betrami operator in terms of represetation error and computation time. Here, LBO means Laplace-Beltrami operator

N	computation time	
	spherical harmonics (sec)	eigenfunctions of LBO (sec)
16	0.886	1.897×10^{2}
32	0.931	3.306×10^{2}
64	1.041	5.436×10^{3}
128	2.033	1.508×10^{5}

Table 2.2: Representation comparison of a hexagonal golf ball, shown in Figure 2.2b, using spectral harmonics and eigenfunctions of Laplace-Betrami operator in terms of computation time. Here, LBO means Laplace-Beltrami operator

to be used to capture the shapes of small dimples of the golf ball should be equal to or higher than $N = 128$. However, the amount of time required to compute eigenfunctions of LBO of the hexagonal golf ball so rapidly increases as N grows that approximately 42 hours are needed to obtain them, while spherical harmonics only needs approximately 2 seconds. Due to this huge computation time, spectral representation using eigenfunctions of LBO cannot be applied to optimization process where hundreds of different shapes of objects should be represented.



(a) SH, $N = 32$ (b) SH, $N = 64$ (c) SH, $N = 128$

(d) LBO, $N = 32$ (e) LBO, $N = 64$ (f) LBO, $N = 128$

Figure 2.5: As a function of N, the shapes of golf balls represented by spherical harmonics are presented in (a), (b) and (c), while the shapes of golf balls represented by Laplace-Betrami operator are presented in (d), (e) and (f). Here, SH and LBO mean representation using spherical harmonics and eigenfunctions of Laplace-Beltrami operator, respectively. The original shape of the golf ball is plotted in Figure 2.2 (b).

Chapter 3

Imposing geometric constraints on an object

3.1 Differential equation

No geometric constraints are imposed on the morphed object so far. However, in many cases, a morphed object needs to be constrained to meet certain pre-described geometric conditions. At this point, we want to define the terms *unconstrained object* and *constrained object*. We use the term *unconstrained object* to refer to the shape of an n-hole object or a patch before constraints are imposed on the boundaries and *constrained object* to refer to the shape of an n-hole object or a patch after constraints are imposed on the boundaries. We impose geometric constraints on an object by spectrally solving the partial differential equation with boundary conditions that are incorporated with geometric constraints. In this chapter, we will describe how to impose geometric constraints on n-hole objects and a patch. We will explain for a 2-hole object first and move onto for a 1-hole object and a patch.

Governing equation to impose geometric constraints on 1- and 2-hole objects

To impose geometric constraints on a 2-hole object, let's consider a hyperdiffusion-like partial differential equation

$$\frac{\partial r(u,v)}{\partial t} = \mu_1 \nabla^2 r(u,v), \tag{3.1}$$

where μ_1 is dynamic diffusivity, $r(u,v)$ is cylindrical radius of a 2-hole object and ∇^2 is a Laplace operator defined as

$$\nabla^2 = \frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2}. \tag{3.2}$$

Because Equation 3.1 has a time-derivative term, we assume that a constrained surface can be created by marching the solution in time. If backward Euler method is used to discretize in time, the surface of a constrained object can be obtained by solving

$$(1 - \nu_1 \nabla^2) r^c(u,v) = r^u(u,v), \tag{3.3}$$

where ν_1 is defined as the kinematic diffusivity and $r^c(u,v)$ and $r^u(u,v)$ respectively represent the cylindrical radius of the unconstrained and constrained object. For simplicity, we will refer to the kinematic diffusivity, ν_1, as diffusivity (not μ_1) in the rest of the paper.

Solving the second order equations only allows shape constraints to be imposed on an object. To deal with higher order derivatives, for example, to achieve C_n continuity, a higher order PDE needs to be solved. If we generalize a governing equation enabling to impose n^{th} order geometric constraints, the generalized governing equation can be rewritten as

$$\frac{\partial r(u,v)}{\partial t} = \sum_{i=0}^{n} (-1)^i \mu_{i+1} \nabla^{2(i+1)} r(u,v) \tag{3.4}$$

Similarly, the time discretized governing equation is easily re-derived from Equation 3.4, given as

$$\Big(1 - \sum_{i=0}^{n} (-1)^i \nu_{i+1} \nabla^{2(i+1)}\Big) r^c(u,v) = r^u(u,v) \tag{3.5}$$

We tested the equation up to the second order geometric constraints ($n = 2$) and ascertained that shape, slope and curvature constraints are well imposed on an object.

In design-by-morphing, Equation 3.5 is solved spectrally. Due to the fast transform [46] and fast solver in the spectral method [21], we can solve this parabolic PDE efficiently.

Governing equation to impose geometric constraints on a patch

Unlike imposing geometric constraints on 1- or 2-hole objects which parabolic PDE is solved to impose geometric constraints, elliptical PDE is solved to impose constraints on a patch. The reason we used elliptical PDE is that a $\frac{\partial^2}{\partial v^2}\frac{\partial^2}{\partial u^2}$ term generated from parabolic equation requires huge computational work in our mathematical approach. Removing this term via considering elliptical PDE allows to solve governing equation efficiently. To impose shape constraint on a patch whose height z is parameterized by mapping variables (u, v), we solve

$$\frac{\partial z(u,v)}{\partial t} = \mu \left(\frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2} \right) z(u,v). \tag{3.6}$$

If we define z^u as the height of an unconstrained patch and z^c as the height of a constrained patch, using the implicit first order time stepping scheme gives

$$z^c(u,v) - \nu \left(\frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2} \right) z^c(u,v) = z^u(u,v). \tag{3.7}$$

Equation 3.96 can be expanded to higher governing equation to impose shape, slope or curvature constraints. To imposed up to n^{th} order geometric constraints, governing equation is given as

$$\frac{\partial z(u,v)}{\partial t} = \sum_{i=0}^{n} (-1)^i \mu_{i+1} \left(\frac{\partial^{2(i+1)}}{\partial u^{2(i+1)}} + \frac{\partial^{2(i+1)}}{\partial v^{2(i+1)}} \right) z(u,v). \tag{3.8}$$

Similarly, using implicit first order time stepping scheme allows to obtain

$$z^c(u,v) - \sum_{i=0}^{n} (-1)^i \nu_{i+1} \left(\frac{\partial^{2(i+1)}}{\partial u^{2(i+1)}} + \frac{\partial^{2(i+1)}}{\partial v^{2(i+1)}} \right) z^c(u,v) = z^u(u,v). \tag{3.9}$$

3.2 Imposing geometric constraints on a 2-hole object

Boundary conditions

To solve Equation 3.5, boundary conditions should be defined. This indicates that if ones define boundary conditions to be equivalent to geometric constraints that they want to apply on objects, the geometric constraints can be imposed on the objects. For example, suppose a 2-hole object whose cylindrical radius $r(u,v)$ is approximated by

$$r(u,v) = \sum_{l=0}^{L} \sum_{m=-N}^{N} a_l^m T_l(u) e^{imv}, \tag{3.10}$$

is parameterized be the bottom and top ends when $u = -1$ and $u = 1$, respectively. Further, suppose the radius of a 2-hole object should be $f^0(v)$ and $g^0(v)$ at the bottom and top, respectively, where $f^0(v)$ and $g^0(v)$ are given functions, and $f^0(v)$ and $g^0(v)$ can be approximated by a truncated Fourier series as a function of v, given as

$$f^0(v) = \sum_{m=-N}^{N} \hat{f}_m^0 e^{imv} \tag{3.11}$$

and

$$g^0(v) = \sum_{m=-N}^{N} \hat{g}_m^0 e^{imv}. \tag{3.12}$$

Here, $\hat{f}_m^0$ and $\hat{g}_m^0$ are respectively Fourier coefficients of the functions $f^0(v)$ and $g^0(v)$. Since when $u = -1$ and $u = 1$, the radius of a 2-hole object should be equal to $f(v)$ and $g(v)$, the right hand sides of Equations 3.11 and 3.12 can be equated with Equation 3.10 when $u = -1$ and $u = 1$. In other words, when $u = -1$, the following condition should be obeyed.

$$\sum_{l=0}^{L} \sum_{m=-N}^{N} a_l^m T_l(-1) e^{imv} = \sum_{m=-N}^{N} \hat{f}_m^0 e^{imv} \tag{3.13}$$

Similarly, when $u = 1$, the following condition also should be obeyed.

$$\sum_{l=0}^{L} \sum_{m=-N}^{N} a_l^m T_l(1) e^{imv} = \sum_{m=-N}^{N} \hat{g}_m^0 e^{imv}. \tag{3.14}$$

Rearranging Equation 3.13 gives

$$\sum_{m=-N}^{N} e^{imv} \Big(\sum_{l=0}^{L} a_l^m T_l(-1) - \hat{f}_m^0 \Big) = 0. \tag{3.15}$$

Without loss generality, to satisfy Equation 3.15 for all m modes, the parenthesis in the equation should be zero, can be written as

$$\sum_{l=0}^{L} a_l^m T_l(-1) = \hat{f}_m^0. \tag{3.16}$$

Similarly, when $u = 1$, we can obtain

$$\sum_{l=0}^{L} a_l^m T_l(1) = \hat{g}_m^0. \tag{3.17}$$

For the higher order of geometric constraints, similar conditions can be derived. For example, if i^{th} derivative of radius respect to u is required to be equal to $f^i(v)$ at the bottom and i^{th} derivative of radius respect to u is required to be equal to $g^i(v)$ at the top, where functions $f^i(v)$ and $g^i(v)$ are known and approximated by Fourier series with their own Fourier coefficients, $\hat{f}_m^i$ and $\hat{g}_m^i$, these conditions can be written as

$$f^i(v) = \sum_{m=-N}^{N} \hat{f}_m^i e^{imv}, \tag{3.18}$$

and

$$g^i(v) = \sum_{m=-N}^{N} \hat{g}_m^i e^{imv}, \tag{3.19}$$

it is straightforward to derive the following conditions.

$$\sum_{l=0}^{L} a_l^m \frac{\partial^i T_l(u)}{\partial u^i}\Big|_{u=-1} = \hat{f}_m^i \tag{3.20}$$

$$\sum_{l=0}^{L} a_l^m \frac{\partial^i T_l(u)}{\partial u^i}\Big|_{u=1} = \hat{g}_m^i. \tag{3.21}$$

Equations 3.20 and 3.21 are boundary conditions of a 2-hole object in solving Equation 3.4. When Equation 3.4 is spectrally solved, these boundary conditions are imposed by substituting a few highest modes of l at each m mode, which is well known as tau method. More detail knowledge on how to solve it is described in the following subsection.

Solving PDE with geometric conditions

To explain how to solve PDE spectrally, we first focus on the second order diffusion equation, which is written in Equation 3.3, and then, we will move onto solving high order diffusion equation. Imposing shape constraint on a 2-hole object can be implemented by solving diffusion-like equation

$$\frac{\partial r(u,v)}{\partial t} = \mu \nabla^2 r(u,v), \tag{3.22}$$

where

$$\nabla^2 = \frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2}. \tag{3.23}$$

Let's define r^u as the cylindrical radius of an unconstrained 2-hole object and r^c as cylindrical radius of a constrained 2-hole object. We represent $r^u$and r^c by truncated sum of Chebyshev polynomials and Fourier expansions, which can be written as

$$r^u(u,v) = \sum_{l=0}^{N} \sum_{m=-N}^{N} b_l^m T_l(u) e^{imv} \tag{3.24}$$

and

$$r^c(u,v) = \sum_{l=0}^{N} \sum_{m=-N}^{N} a_l^m T_l(u) e^{imv}, \tag{3.25}$$

where a_l^m and b_l^m are spectral coefficients of r^c and r^u, respectively. Using the backward Euler method in resolving Equation 3.22 in time allows to impose geometric constraints by assuming that geometric constraints can be imposed on an object by marching the equation one time step in time, which gives

$$(1 - \nu \nabla^2) r^c(u,v) = r^u(u,v), \tag{3.26}$$

where $\nu = \mu \Delta t$ and Δt is time-step of backward Euler scheme. Substituting Equations 3.24 and 3.25 into 3.26 gives

$$\sum_{l=0}^{N} \sum_{m=-N}^{N} (1 - \nu \nabla^2) a_l^m T_l(u) e^{imv} = \sum_{l=0}^{N} \sum_{m=-N}^{N} b_l^m T_l(u) e^{imv}. \tag{3.27}$$

If we rearrange Equation 3.27 and express the rearranged equation as the matrix form, we obtain

$$\sum_{m=-N}^{N} [(\mathbf{I} - \nu(\mathbf{D}^2 - m^2 \mathbf{I}) \mathbf{a}(m) - \mathbf{b}(m)] e^{imv} = 0, \tag{3.28}$$

where $\mathbf{a}(m) = [a_0^m,\ a_1^m, \cdots,\ a_N^m]^T$, $\mathbf{b}(m) = [b_0^m,, b_1^m, \cdots,\ b_N^m]^T$, $\mathbf{I}$ is an identity matrix and $\mathbf{D}$ is a differentiation matrix of Chebyshev polynomials. Without loss of generality, in order to satisfy Equation 3.28 independent of v, the terms inside the parenthesis should be equal to zero at every m. Therefore, Equation 3.28 can be rewritten as

$$\boldsymbol{\Theta}(m)\mathbf{a}(m) = \mathbf{b}(m), \tag{3.29}$$

where $\boldsymbol{\Theta}(m) = \mathbf{I} - \nu(\mathbf{D}^2 - m^2\mathbf{I})$. Equation 3.29 cannot be solved because $\boldsymbol{\Theta}(m)$ is a singular matrix. To solve it, boundary conditions should be imposed on $\boldsymbol{\Theta}(m)$ and $\mathbf{b}(m)$, which are given at Equations 3.16 and 3.17. By defining a matrix $\boldsymbol{\Theta}^*(m)$ to be equal to $\boldsymbol{\Theta}(m)$ except the two most bottom rows which are replaced with boundary conditions and a column vector $\mathbf{b}^*(m)$ to be equal to $\mathbf{b}(m)$ except the two bottom most rows which are also replaced with boundary conditions.

$$\boldsymbol{\Theta}^\star(m)\mathbf{a}(m) = \mathbf{b}^\star(m) \tag{3.30}$$

Once Equation 3.30 is solved at each m-th mode, we can finally obtain all coefficients , a_l^m, of constrained geometries.

If ones have to constrain an object higher than C_0 geometric constraints, higher order of PDEs than the second order should be considered. For example, if shape and slope conditions are demanded to be imposed on an object, a fourth order PDE should be solved, while an object is constrained with shape, slope and curvature constraints, a sixth order PDE should be solved. For a fourth order diffusion equation, from Equation 3.4, we need to solve

$$\frac{\partial r(u,v)}{\partial t} = \nu_1 \nabla^2 r(u,v) - \nu_2 \nabla^4 r(u,v), \tag{3.31}$$

Similar to solving a second order diffusion equation, we obtain the following equation

$$\left(1 - \nu_1 \nabla^2 + \nu_2 \nabla^4\right) r^c = r^u, \tag{3.32}$$

and it can be represented like Equation 3.29 as a matrix form, where $\boldsymbol{\Theta}(m) = (1 + m^2\nu_1 + m^4\nu_2)\mathbf{I} + \nu_2\mathbf{D}^4 - (\nu_1 + 2m^2\nu_2)\mathbf{D}^2$. After, the four most bottom rows of the $\boldsymbol{\Theta}(m)$ matrix and $\mathbf{b}(m)$ column vector are replaced with boundary conditions, spectral coefficients of a constrained object can be obtained.

Similarly, for a sixth order diffusion equation, we need to solve

$$\frac{\partial r(u,v)}{\partial t} = \nu_1 \nabla^2 r(u,v) - \nu_2 \nabla^4 r(u,v) + \nu_3 \nabla^6 r(u,v), \tag{3.33}$$

As we did to solve second and fourth order PDEs, backward Euler time-stepping method allows to obtain the equation specified by the radius of an unconstrained and constrained objects

$$\big(1 - \nu_1 \nabla^2 + \nu_2 \nabla^4 - \nu_3 \nabla^6\big) r^c = r^u. \tag{3.34}$$

Equation 3.34 also can be represented a matrix form given in Equation 3.29, where $\mathbf{\Theta}(m) = (1+m^2\nu_1+m^4\nu_2+m^6\nu_3)\mathbf{I}-\nu_3\mathbf{D}^6+(\nu_2+3\nu_3m^2)\mathbf{D}^4-(\nu_1+2m^2\nu_2+3m^4\nu_3)\mathbf{D}^2$. After, the six most bottom rows of the $\mathbf{\Theta}(m)$ matrix and $\mathbf{b}(m)$ column vector are replaced with boundary conditions, spectral coefficients of a constrained object can be obtained.

As described so far, the surface of a constrained object can be obtained by inverting the matrix $\mathbf{\Theta}^\star$. However, this approach is inefficient and induces considerable numerical error because the condition number of the matrix $\mathbf{\Theta}$ is big especially when a higher order PDE is solved. The better way is using a trick allowing to diagonalize the matrix $\mathbf{\Theta}^\star$ and obtain even and odd modes of spectral coefficients separately from the diagonalized matrix. Here, we will be described this method with a fourth-order PDE as an example. Suppose we define p^{th} derivative of r^c respect to u as

$$\frac{d^p r^c}{du^p} = \sum_{l=0}^{N} \sum_{m=-N}^{N} (a_l^m)^{(p)}\, T_l(u) e^{im\phi}, \tag{3.35}$$

where $a_l^{m(0)}$ is equal to a_l^m. Due to the definition of ∇^2 and ∇^4, Equation 3.32 can be rewritten as

$$\Big[1-\nu_1\Big(\frac{\partial^2}{\partial u^2}+\frac{\partial^2}{\partial v^2}\Big)+\nu_2\Big(\frac{\partial^4}{\partial u^4}+2\frac{\partial^2}{\partial u^2}\frac{\partial^2}{\partial v^2}+\frac{\partial^4}{\partial v^4}\Big)\Big]r^c(u,v) = r^u(u,v) \tag{3.36}$$

By substituting Equation 3.35 into Equation 3.36, we obtain

$$\nu_2\,(a_l^m)^{(4)} - \big(2\nu_2 m^2+\nu_1\big)\,(a_l^m)^{(2)} + \big(\nu_2 m^4+\nu_1 m^2+1\big)\,a_l^m = b_l^m \tag{3.37}$$

The boundary conditions of Equation 3.37 are obtained from Equations 3.20 and 3.21, given as

$$\sum_{l=0}^{L} a_l^m T_l(-1) = \hat{f}_m \tag{3.38}$$

$$\sum_{l=0}^{L} a_l^m T_l(1) = \hat{g}_m \tag{3.39}$$

$$\sum_{l=0}^{L} a_l^m \frac{\partial T_l(u)}{\partial u}\Big|_{u=-1} = \hat{f}_m^1 \tag{3.40}$$

$$\sum_{l=0}^{L} a_l^m \frac{\partial T_l(u)}{\partial u}\Big|_{u=1} = \hat{g}_m^1 \tag{3.41}$$

Since $T_l(-1) = (-1)^l$ and $T_l(1) = 1$, adding Equations 3.38 and 3.39, and subtracting Equation 3.38 from Equation 3.39 respectively give

$$\sum_{\substack{l=0 \\ l\,even}}^{N} a_l^m = \frac{\hat{g}_m + \hat{f}_m}{2} \tag{3.42}$$

$$\sum_{\substack{l=1 \\ l\,odd}}^{N} a_l^m = \frac{\hat{g}_m - \hat{f}_m}{2} \tag{3.43}$$

Similarly, since $\left.\frac{\partial T_l(u)}{\partial u}\right|_{u=-1} = (-1)^{l+1} l^2$ and $\left.\frac{\partial T_l(u)}{\partial u}\right|_{u=1} = l^2$, adding Equations 3.40 and 3.41, and subtracting Equation 3.40 from Equation 3.41 respectively give

$$\sum_{\substack{l=1 \\ l\,odd}}^{N} l^2 a_l^m = \frac{\hat{g}_m^1 + \hat{f}_m^1}{2} \tag{3.44}$$

$$\sum_{\substack{l=0 \\ l\,even}}^{N} l^2 a_l^m = \frac{\hat{g}_m^1 - \hat{f}_m^1}{2} \tag{3.45}$$

It should be noticed that there is a well-known three term recursion relation between $a^{(p)}$ and $a^{(p-1)}$ at each m [9]

$$c_l \left(a_l^m\right)^{(p)} - \left(a_{l+2}^m\right)^{(p)} = 2(l+1)\left(a_{l+1}^m\right)^{(p-1)}, \quad l \geqq 0, \tag{3.46}$$

where

$$c_l = \begin{cases} 2, & l = 0 \\ 1, & l > 0 \end{cases} \tag{3.47}$$

Equation 3.46 is used to derive a formula to solve Equation 3.32 efficiently. After defining $p = 4$ in Equation 3.46, substituting Equation 3.46 into Equation 3.37 gives

$$\left(a_l^m\right)^{(3)} = A\left(a_l^m\right)^{(1)} - \frac{1}{2l}\left[B\left(c_{l-1}a_{l-1}^m - a_{l+1}^m\right) - P\left(c_{l-1}b_{l-1}^m - b_{l+1}^m\right)\right], \tag{3.48}$$

where

$$\begin{aligned} A &= \frac{2\nu_2 m^2 + \nu_1}{\nu_2} \\ B &= \frac{\nu_2 m^4 + \nu_1 m^2 + 1}{\nu_2} \\ P &= \frac{1}{\nu_2} \end{aligned} \tag{3.49}$$

The highest derivative term of Equation 3.48 is one order lower than that of Equation 3.37. By repeating the same calculation procedure until every high derivative term becomes to zeroth derivative term, we finally obtain

$$\begin{aligned} &\alpha_{l-4} a_{l-4}^m - \left(\frac{A}{B} D_{l-2} + \beta_{l-2} \right) a_{l-2}^m + \left(\frac{1 + AE_l}{B} + \gamma_l \right) a_l^m \\ &\quad - \left(\frac{A}{B} F_{l+2} + \gamma_{l+2} \right) a_{l+2}^m + \epsilon_{l+4} a_{l+4}^m = K_l^m, \quad 4 \leqq l \leqq N, \end{aligned} \tag{3.50}$$

where

$$\begin{aligned} K_l^m &= \frac{P}{B} \left(\alpha_{l-4} b_{l-4}^m - \beta_{l-2} b_{l-2}^m + \gamma_l b_l^m - G_{l+2} L_{l+2} b_{l+2}^m + G_{l+4} \epsilon_{l+4} b_{l+4}^m \right) \\ \alpha_{l-4} &= c_{l-4} D_{l-2} D_{l-4} \\ \beta_{l-2} &= D_{l-2}(E_{l-2} + E_l) \\ \gamma_l &= D_{l-2} F_l + E_l^2 + F_{l+2} D_l \\ L_{l+2} &= (E_l + E_{l+2}) F_{l+2} \\ \epsilon_{l+4} &= F_{l+2} F_{l+4} \\ D_l &= \frac{1}{4(l+1)(l+2)} \\ E_l &= \frac{1}{2l(l-1)(l+1)} \\ F_l &= \frac{1}{4l(l-1)} \\ G_l &= \begin{cases} 1, & l \leqq N \\ 0, & l > N \end{cases} \end{aligned} \tag{3.51}$$

Here, note that even and odd modes can be obtained independently because even and odd modes of Equation 3.50 are uncoupled. For example, for even modes, the structure of it looks like

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & & & & & \cdots & & & & & 1 \\ 0 & 4 & 16 & 36 & 64 & & & & & \cdots & & & & & N^2 \\ x & x & x & x & x & & & & & & & & & & \\ & x & x & x & x & x & & & & & & & & & \\ & & x & x & x & x & x & & & & & & & & \\ & & & & & & & \vdots & & & & & & & \\ & & & & & & & & x & x & x & x & x & & \\ & & & & & & & & & x & x & x & x & x & \\ & & & & & & & & & & x & x & x & x & x \\ & & & & & & & & & & & x & x & x & x \\ & & & & & & & & & & & & x & x & x \end{pmatrix} \begin{pmatrix} a_0^m \\ a_2^m \\ a_4^m \\ a_6^m \\ a_8^m \\ \vdots \\ a_{N-8}^m \\ a_{N-6}^m \\ a_{N-4}^m \\ a_{N-2}^m \\ a_N^m \end{pmatrix} = \begin{pmatrix} \frac{(\hat{g}_m)+(\hat{f}_m)}{2} \\ \frac{(\hat{g}_m^1)-(\hat{f}_m^1)}{2} \\ K_4^m \\ K_6^m \\ K_8^m \\ \vdots \\ K_{N-8}^m \\ K_{N-6}^m \\ K_{N-4}^m \\ K_{N-2}^m \\ K_N^m \end{pmatrix}$$

where x means non-zero elements of the matrix, which can be calculated from Equations 3.49, 3.50 and 3.51. The first and second rows are replaced by boundary conditions we have derived in Equations 3.42 and 3.45. This penta-diagonal matrix can be efficiently solved with a LU decomposition in $O(N)$ operations.

Effect of diffusivities

The shape of a constrained object is affected by diffusitivies, as well as the shape of an unconstrained object and imposed geometric constraints. To study the effect of diffusivities, we introduce a simple two-hole object constraint problem. In this problem, a cylinder is demanded to have constraints such that the shape of the cylinder must be a rectangle (or star or heart) with a perpendicular slope to the horizontal plane maintained at the upper end, while the cylinder must be unchanged in terms of its shape and slope at the bottom end. To accomplish this, fourth-order PDEs are solved with boundary conditions incorporating these requirements. The constrained cylinders are presented in figure 3.1, where the color indicates the strength of deformation before and after imposing constraints, defined as

$$C(u,v) = \frac{\sqrt{(R^c(u,v) - R^u)^2}}{|R^c(u,v) - R^u|_\infty} \tag{3.52}$$

Here, $R^c(u,v)$ is the radius of the constrained cylinder, R^u is the radius of the unconstrained cylinder, and $C(u,v)$ is the color in Figure 3.1, where red indicates high deformation and blue indicates small deformation. The first row of the figure shows constrained cylinders with a weak diffusivity effect, while the bottom row shows a constrained cylinder with a strong diffusivity effect. The boundary layer, defined as the depth of shape deformation propagating from one end to the other end (in this case from the top end to the bottom end), is thicker when diffusivities are strong than in the weak diffusitivies case. In addition, the constrained object more smoothly obeys imposed constraints when diffusitivies are strong. Thus, diffusivities are associated with a smoothing effect, thereby determining the smoothness of constrained objects near the boundaries. This is consistent with the physical

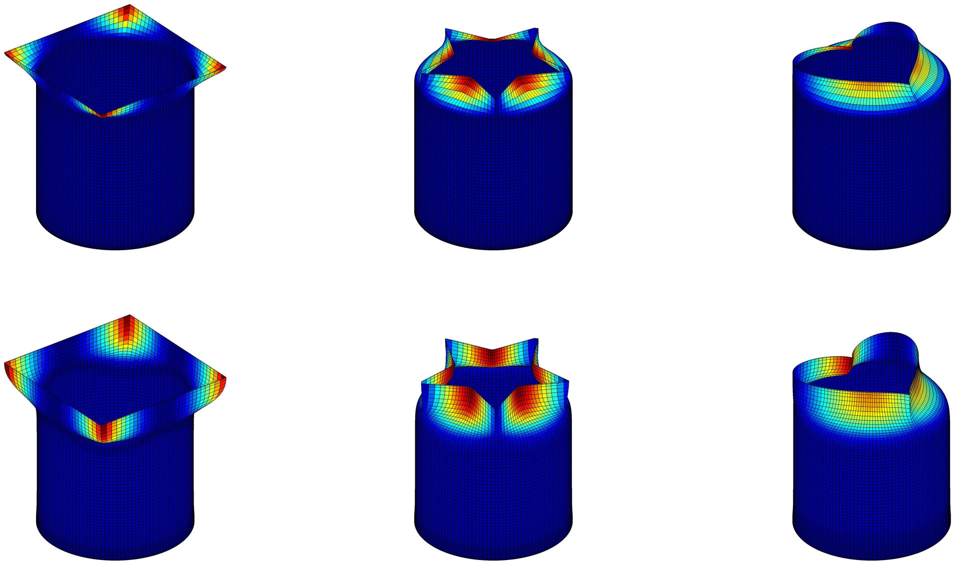


Figure 3.1: The diffusivity effect of constrain solver: A cylinder is required to be square (leftmost), star (middle), and heart-shaped (rightmost) at the top end and unchanged at the bottom end. The slope at both ends are required to be straight along the vertical direction. The diffusivities applied to objects in the first row is smaller than the diffusivities applied to objects in the bottom row. Here, color demonstrates the degree of deformation, where red indicates strong deformation and blue indicates weak deformation.

meaning of diffusitivy, quantifying the transfer rate of thermal and fluidic quantities from one area to another.

So far, we have dealt with constant diffusitivies in space. However, non-constant diffusivities in space can be used instead. By employing varying diffusivities, users are able to adjust the deformation of an object over the space. For example, if the upper and bottom end of a two-hole object are mapped to $u = 1$ and $u = -1$, respectively, the non-constant viscosity

$$\nu = \bar{\nu} + a(u+1)^2, \tag{3.53}$$

where $\bar{\nu}$ is the constant diffusivity and a is positive constant, assists in deforming the upper end with minimized the shape change near the bottom end.

Objects connected by a *neck*

We have applied the constraint solver to various examples to connect two or more separate objects through a *neck*. The *neck* here is named after a neck of human because *neck* splices different objects as the human neck connects a head an body. When objects are connected by a neck, geometric constraints should be imposed on a neck to force a neck stitch objects smoothly in terms of matching shape, slope or curvature at joints. To impose those geometric constraints, shape, slope or curvature at joints should be numerically computed. To do this, diagnostic objects at the joint are needed to be considered. The purpose of creating diagnostic objects are just to numerically compute shape, slope or curvature of connected objects at joints. Once these values are computed, diagnostic objects are no longer necessary. For easy understanding of diagnostic objects, diagnostic objects are colored by purple in this subsection, while shapes of an unconstrained and constrained objects are presented as green. For example, the object colored by purple in Figure 3.2 is a diagnostic object. The diagnostic object is part of a connected object (in this case, it is a part of horse hindquarters) whose one of the boundaries becomes the joint of a constrained problem (in this case, lower boundary is the joint of the constrained problem). The surface of this diagnostic object is approximated by our 2-hole basis functions. Due to the fact that one of the boundaries of a diagnostic object is the joint of the constrained problem whose surface is approximated by spectral basis functions, shape, slope or curvature at the joint can be easily obtained from spectral calculation.

We applied creation of neck approach to stitch a two or more objects in various shaped objects. For example, Figure 3.3 shows creation of a centaur by stitching upper human body to horse hindquarters thorough a neck. To create a centaur, we first consider the shape of a human and the shape of a horse, which are presented in Figure 3.3a and 3.3b. The boxes here mean the these objects are sliced by a flat plane. To slice them, we used a software called Netfabb. The sliced objects will be connected by imposing geometric constraints, assuring upto C_2 continuity at the joints, on a unconstrained neck, shown as a blue object in Figure 3.3c. A way how to create an unconstrained neck may be debatable, but choosing the shape of lower belly of a human as a unconstrained neck works well in this case. The constrained neck are obtained as follows.

Let's define $\underline{X}_N(u,v) = [x_N(u,v),\, y_N(u,v),\, z_N(u,v)]^\intercal$ as position functions of an unconstrained neck and $\underline{X}_{HD}(u,v) = [x_{HD}(u,v),\, y_{HD}(u,v),\, z_{HD}(u,v)]^\intercal$ as position functions of a diagnostic object for the hindquarters of the horse, where all objects are mapped to be the bottom of objects at $u = -1$ and the top of objects at $u = 1$. Here each of position functions' component is approximated by 2-hole basis function. For example, $x_N(u,v)$ is approximated as

$$x_N(u,v) = \sum_{l=0}^{N} \sum_{m=-N}^{N} (\hat{x}_N)_l^m T_l(u) e^{imv}, \tag{3.54}$$

where $(\hat{x}_N)_l^m$ is spectral coefficients of x_N. Similarly, if we define $(\hat{y}_N)_l^m$ and $(\hat{z}_N)_l^m$ as spectral coefficients of $y_N(u,v)$ and $z_N(u,v)$ respectively, generalized spectral approximation of an unconstrained object' positions can be written as

$$\underline{X}_N(u,v) = \sum_{l=0}^{N} \sum_{m=-N}^{N} (\hat{\underline{X}}_N)_l^m T_l(u) e^{imv}, \tag{3.55}$$

where $(\hat{\underline{X}}_N)_l^m = [(\hat{x}_N)_l^m,\ (\hat{y}_N)_l^m,\ (\hat{z}_N)_l^m]^\intercal$. A constrained neck $\underline{X}_N^C(u,\tau)$ can be obtained by solving

$$(1-\nu_1\nabla^2+\nu_2\nabla^4-\nu_3\nabla^6)\underline{X}_N^C(u,\tau)=\underline{X}_N(u,\tau), \tag{3.56}$$

where $\nabla^2 = \frac{\partial^2}{\partial u^2}+\frac{\partial^2}{\partial v^2}$, with the following boundary conditions.

$$\underline{X}_N^c|_{u=1} = \underline{X}_N^u|_{u=1} \tag{3.57}$$

$$\underline{X}_N^c|_{u=-1} = \underline{X}_{HD}|_{u=1} \tag{3.58}$$

$$\frac{d\underline{X}_N^c}{du}|_{u=1} = \frac{d\underline{X}_N^u}{du}|_{u=1} \tag{3.59}$$

$$\frac{d\underline{X}_N^c}{du}|_{u=-1} = k_N\frac{d\underline{X}_{HD}}{du}|_{u=1} \tag{3.60}$$

$$\frac{d^2\underline{X}_N^c}{du^2}|_{u=1} = \frac{d^2\underline{X}_N^u}{du^2}|_{u=1} \tag{3.61}$$

$$\frac{d^2\underline{X}_N^c}{du^2}|_{u=-1} = k_N^2\frac{d^2\underline{X}_{HD}}{du^2}|_{u=1}, \tag{3.62}$$

where k_N is constant and $\underline{X}_{HD}(u,v) = [x_{HD}(u,v),\ y_{HD}(u,v),\ z_{HD}(u,v)]^\intercal$ is position functions of a diagnostic object whose surface is represented by 2-hole basis functions with spectral coefficients $(\hat{\underline{X}}_{HD})_l^m = [(\hat{x}_{HD})_l^m,\ (\hat{y}_{HD})_l^m,\ (\hat{z}_{HD})_l^m]^\intercal$ where $(\hat{x}_{HD})_l^m$, $(\hat{y}_{HD})_l^m$ and $(\hat{z}_{HD})_l^m$ are spectral coefficients of $x_{HD}(u,v)$, $y_{HD}(u,v)$ and $z_{HD}(u,v)$ respectively. We assume that Equations 3.57, 3.59 and 3.61 are sufficient conditions to retain C_2 continuity at the upper joint since the unconstrained neck was brought from the original geometry of the human sharing the same geometric characteristics with the connected human object at the joint. The reason why we multiply k_N and k_N^2 in Equations 3.60 and 3.62 is that they assist to control the boundary layer of the constrained neck, while it does not violate C_2 continuity geometric conditions.

This imposing constraint process is graphically summarized in Figure 3.3. In the figure, it shows how we have made a centaur by connecting two unconstrained one-holed sub-objects together via a neck that are constrained to join smoothly to the open boundaries of the 1-holed sub-objects. A centaur is created from the head, arms and upper torso (an unconstrained 1-hole sub-object) that are detached from a human in Figure 3.3a, and the hindquarters of a horse (another unconstrained 1-hole sub-object) that is detached from a horse in Figure 3.3b. These two sub-objects, shown in Figure 3.3c, are then connected to each other by a lower torso of another human (a 2-hole sub-object) that is constrained at both of its open end boundaries to match the shape, slope, and curvature at the open boundaries of the 1-hole sub-objects to which they are attached.

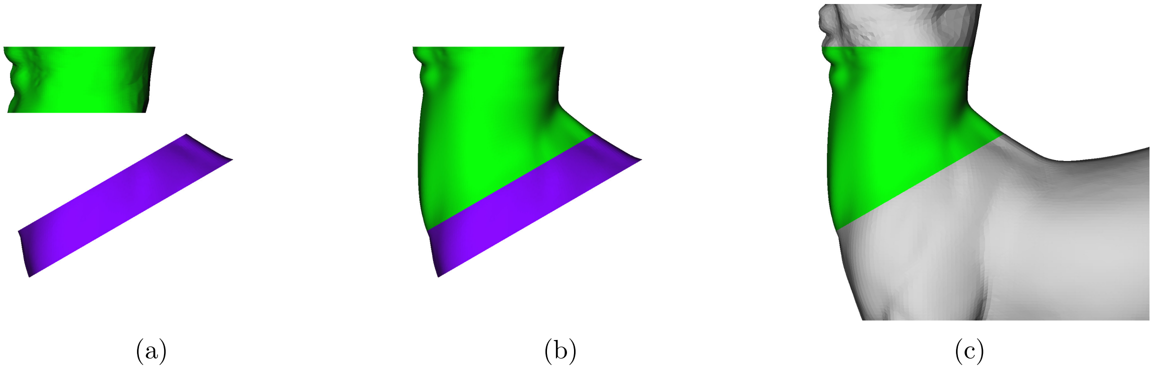


Figure 3.2: Unconstrained and constrained necks with the diagnostic object for a centaur. Unconstrained neck, shown in (a), is constrained to be equal to the shape, slope and curvature with those of the diagnostic object, colored as purple, at the attachment. Upper human body and horse hindquarters are then smoothly connected by the constrained neck, which is shown in (c).

Figure 3.4 is a repeat of the centaur construction shown in Figure 3.3 with one change. In Figure 3.4a only the shape, but not the slope or curvature of the 2-holed lower torso is constrained to match the open boundaries of the human upper torso and the hindquarters of the horse. In Figure 3.4b, the shape and slope, but not curvature of the 2-holed lower torso is constrained to match the open boundaries of the human upper torso and the hindquarters of the horse. The centaur in Figure 3.4c is identical to the one in Figure 3.3e where the shape, slope, and curvature of the 2-holed lower torso is constrained to match the open boundaries of the human upper torso and the hindquarters of the horse. The constrained 2-holed lower torso in Figure 3.4a was computed using Equation 3.56 with $\nu_2 = \nu_3 = 0$. This is a second, rather than sixth-order equation, so imposing the shapes at the boundaries is sufficient. Similarly, the centaur in Figure 3.4b was computed using Equation 3.56 with $\nu_3 = 0$, and this fourth-order equation only requires imposing the shapes and slopes at the two boundaries. The purpose of this figure is to show how the increased order of the partial differential equation and the increased number of constraints makes the joints between the sub-objects smoother.

Besides of a centaur object, two more examples are presented in Figures 3.5, and 3.6. In Figure 3.5, an animal is made up of a horse head (an unconstrained 1-hole sub-object) that is removed from a horse and the headless body (an unconstrained 1-hole sub-object) of Batman which is detached from its head as shown in Figure 3.5a. An unconstrained neck (a 2-hole sub-object) is placed between the Batman body and the horse head in Figure 3.5b. The neck is then constrained at both ends to match the shape, slope, and curvature of the horse head and Batman body (Figures 3.5c, d, and e).

Figure 3.6 shows a hand in which each fingertip is replaced by a smaller version of the original hand. In this case, the original hand in Figure 3.6a, which is a 1-hole sub-object with the hole being the open boundary at the wrist, has its five fingertips sliced off creating

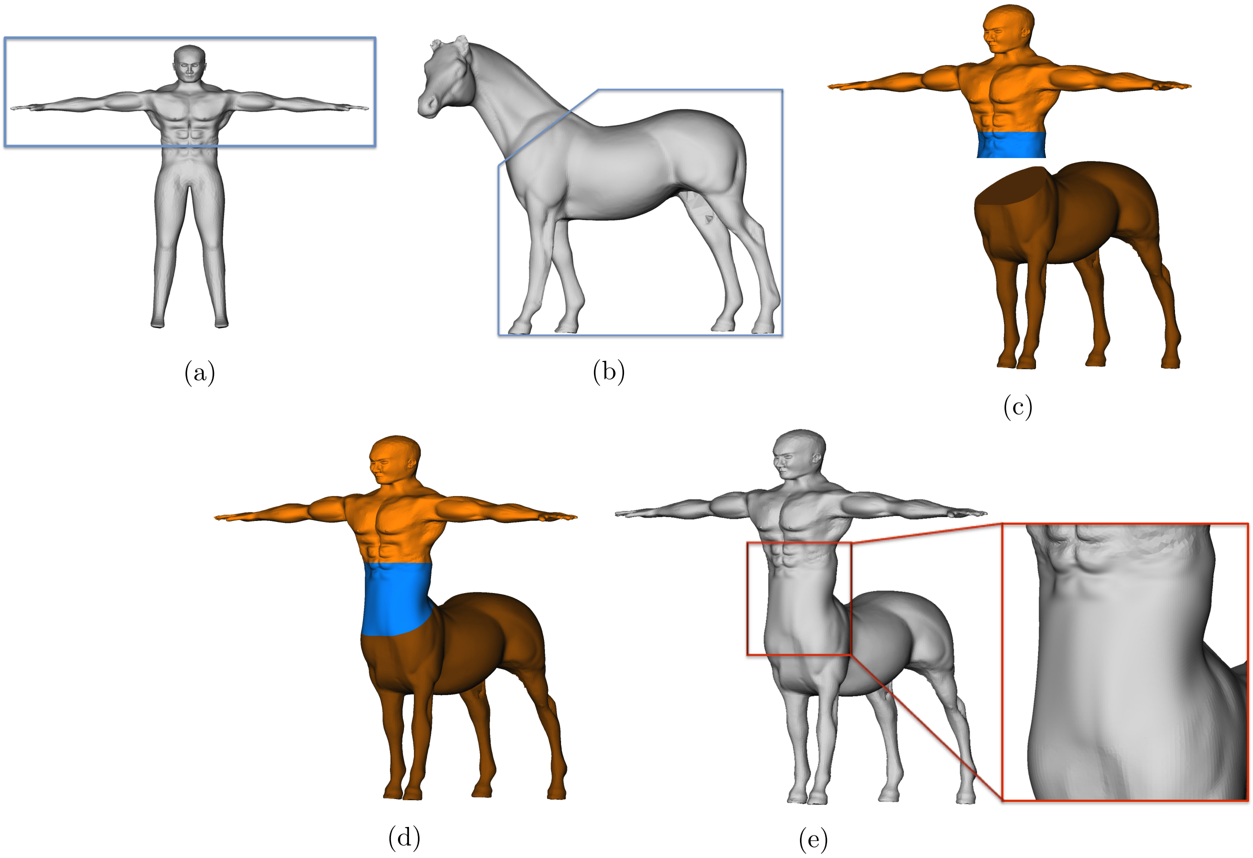


Figure 3.3: Creation of a centaur. The upper human body (a) and hindquarters of the horse (b) are detached creating two unconstrained 1-hole sub-objects. (c) An unconstrained lower torso (a 2-holed sub-object and shown in blue) is placed between the upper human body and horse hindquarters. (d) The (blue) 2-holed sub-object is then constrained at both ends so that its open boundaries match the shapes, slopes, and curvatures of the open boundaries of the upper human body and horse hindquarters. (e) Same as (d) but with a grey palette and with a blow-up showing the constrained lower torso.

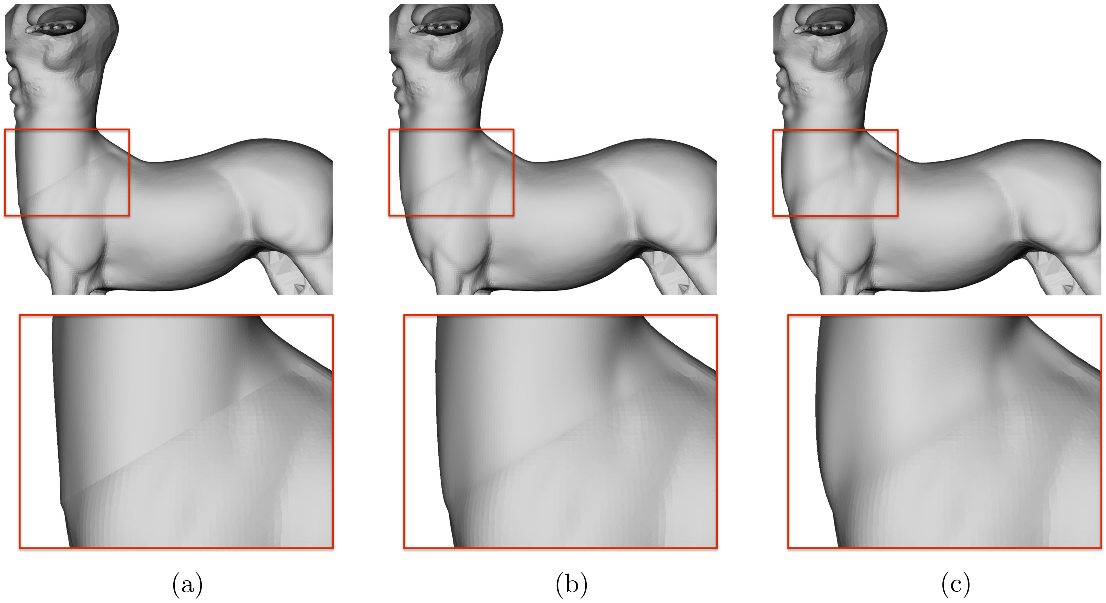
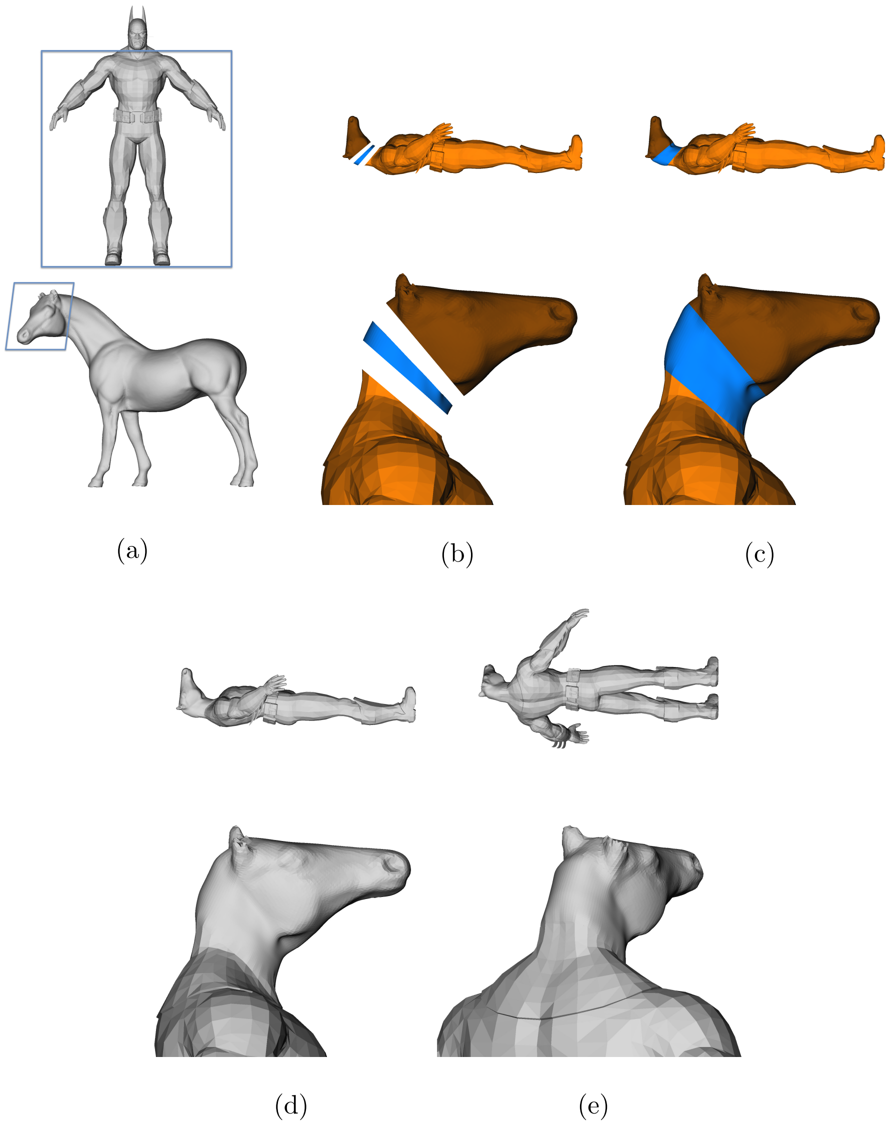


Figure 3.4: Re-creation of the centaur in Figure 3.3 with different constraints. (a) The lower torso that connects the upper human body to the horse's hindquarters is constrained to match the shapes of the open boundaries of the hindquarters and the upper human torso. (b) Same as (a) but the slopes are also constrained to match. (c) Same as (b) but the curvatures are also constrained to match. The regions where the lower torso is attached to the horse and human are much less visible in (c).

a 6-hole sub-object (Figure 3.6b). Five smaller duplicates of the original hand in Figure 3.6a are then placed near the five sliced fingers of the original hand as in Figure 3.6c. Each of the five duplicate hands are then attached to sliced fingers via a connecting *knuckle*, rather than a neck, which is an unconstrained 2-holed sub-object (Figure 3.6d). The five knuckles are then constrained at each end to match the positions, and slopes of sliced fingers and wrists (Figure 3.6d, e and f).

3.3 Imposing geometric constraints on an 1-hole object

A way of imposing geometric constraints on a 1-hole object is similar to the way of imposing geometric constraints on a 2-hole object. The only difference is the definition of Laplace operator, ∇^2, and formulas to solve governing PDEs resulted from the usage of different basis functions. In this section, we will re-derive boundary conditions of a 1-hole object and show how governing PDEs are solved for a 1-hole object.



(a) (b) (c)

(d) (e)

Figure 3.5: Horse head attached to a Batman body. (a) the horse head and Batman body are detached to create two 1-holed sub-objects. (b) An unconstrained 2-holed neck (blue) is placed between the horse head and body. (c) The neck is constrained to match the shapes, slopes and curvatures of the open boundaries of the head and body. (d) Same as (c) but shown with a grey palette. (e) Same as (d) but from a different point of view. In panels (b) – (e) the full animal is shown in the upper row and a blow-up of the head and neck region are shown in the lower row.

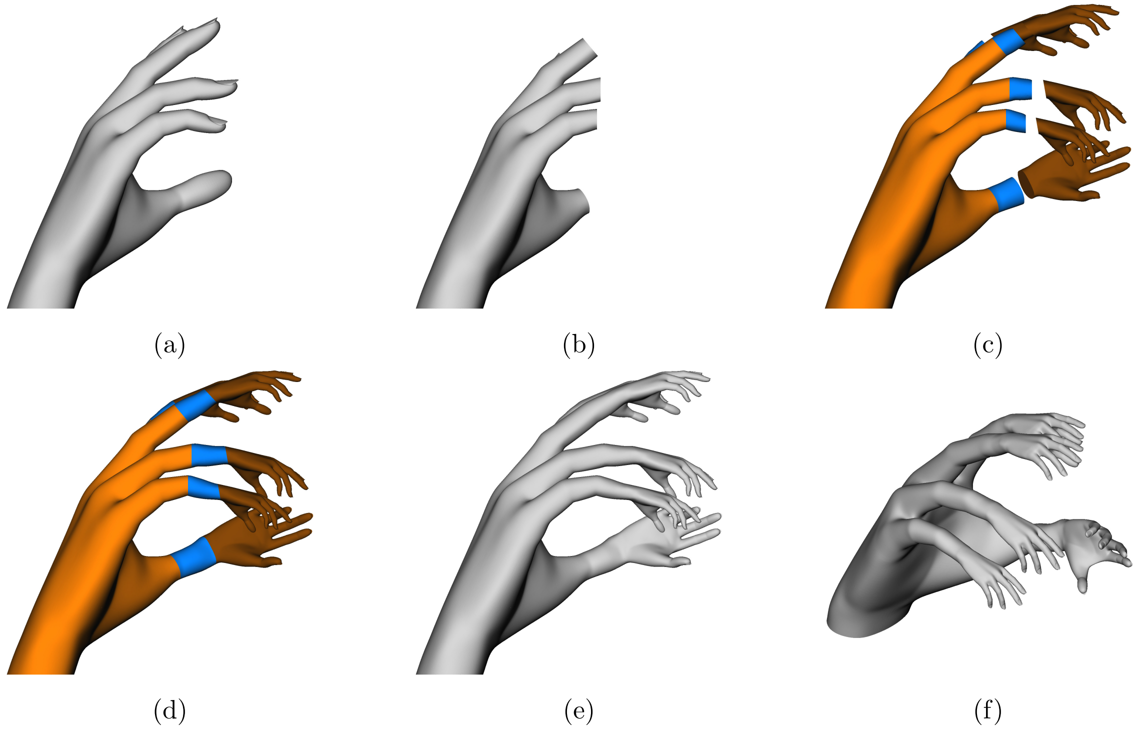


Figure 3.6: A hand in which the fingertips are replaced by hands. (a) The original 1-holed hand with the open boundary at the wrist. (b) The fingertips are sliced off creating holes. (c) Duplicate smaller 1-holed hands are placed near the severed fingers with unconstrained 2-holed knuckles (blue) between them. (d) the knuckles are constrained at both ends to match the shapes, slopes and curvatures of the open boundaries of the five sliced fingers and five wrists. (e) Same as (d) but with a grey palette. (f) Same as (e) but from a different point of view.

Imposing boundary conditions

As described in Section 3.2, imposing shape constraint on an object can be implemented by solving a diffusion-like PDE with boundary conditions. Let consider the following diffusion equation

$$\frac{\partial r(u,v)}{\partial t} = \mu \nabla^2 r(u,v), \tag{3.63}$$

where $r(u,v)$ is a cylindrical radius of a 1-hole object, μ is dynamic diffusivity and

$$\nabla^2 = \frac{1}{u}\frac{\partial}{\partial u}\left(u\frac{\partial}{\partial u}\right) + \frac{1}{u^2}\frac{\partial^2}{\partial v^2}. \tag{3.64}$$

Here, $r(u,v)$ is approximated by Equation 2.48. The ranges of u and v are respectively $0 \leq u \leq 1$ and $0 \leq v \leq 2\pi$, and the south or north pole of a 1-hole object is parameterized to be $u = 0$, while the equator of a 1-hole object is parameterized to be $u = 1$. Suppose the radius of a 1-hole object is demanded to be a known function $f(v)$ at the equator. If $f(v)$ is approximated by Fourier series, $f(v)$ can be written as

$$f(v) = \sum_{m=-N}^{N} \hat{f}_m e^{imv}. \tag{3.65}$$

By equating Equation 3.63 at the equator with Equation 3.65, we obtain

$$\sum_{l=0}^{L} \sum_{m=-N}^{N} a_l^m W_l^m(1,v) = \sum_{m=-N}^{N} \hat{f}_m e^{imv}. \tag{3.66}$$

Since $W_l^m(1,v) = V_l^m(1)e^{imv}$, by substituting $V_l^m(1)e^{imv}$ into $W_l^m(1,v)$ and factoring e^{imv} out, Equation 3.66 becomes

$$\sum_{m=-N}^{N} \Big(\sum_{l=0}^{L} a_l^m V_l^m(1) - \hat{f}_m \Big) e^{imv} = 0. \tag{3.67}$$

Then, it is obvious to claim that inside of the parenthesis of Equation 3.67 should be equal to zero at each m, giving

$$\sum_{l=0}^{L} a_l^m V_l^m(1) = \hat{f}_m. \tag{3.68}$$

For the high derivative geometric constraints, the same process allows to derive boundary conditions for them. If we suppose that i^{th} derivatives of $r(u,v)$ respect to u are demanded to be $f^i(v)$ at the equator and $\hat{f}_m^i$ is defined as m^{th} Fourier coefficients of $f^i(v)$, boundary conditions of PDEs for geometric constraints imposition of a 1-hole object are

$$\sum_{l=0}^{L} a_l^m \frac{\partial^i V_l^m}{\partial u^i}\Big|_{u=1} = \hat{f}_m^i \tag{3.69}$$

Solving PDE with geometric conditions

Suppose we define r^u as radius of an unconstrained 1-hole object and r^c as radius of a constrained 1-hole object. The approximations of $r^u$and r^c with truncated sum of $W_l^m(u,v)$ can be expressed as

$$r^u\ (u,v) = \sum_{l=0}^{N} \sum_{m=-N}^{N} b_l^m W_l^m(u,v) \tag{3.70}$$

$$r^c(u,v) = \sum_{l=0}^{N} \sum_{m=-N}^{N} a_l^m W_l^m(u,v) \tag{3.71}$$

By using the implicit first order Euler method in Equation 3.63, we can obtain

$$\left(1 - \nu\nabla^2\right) r^c{=}r^u \tag{3.72}$$

where $\nu = \mu\triangle t$, $\triangle t$ is artificial time step. Substituting Equations 3.70 and 3.71 into Equation 3.72 gives

$$\sum_{l=0}^{N}\sum_{m=-N}^{N}\left[(1-\nu\nabla^2)a_l^m - b_l^m\right] W_l^m(u,v) = 0 \tag{3.73}$$

If we use Equation 2.45 to expand ∇^2 in Equation 3.73, we obtain

$$\sum_{l=0}^{N}\sum_{m=-N}^{N}\left[\left(\mathbf{I} - 4k(m+2l+1)\sum_{n=l+1}^{N}(n-l)(l+m+n+1)\right)a_n^m - b_l^m\right] W_l^m(u,v) = 0, \tag{3.74}$$

where $\mathbf{I}$ is an identity matrix. Without loss of generality, in order to satisfy Equation 3.74 independent on u and v, the terms inside the parenthesis should be equal to zero at every l and m. Therefore,

$$\left(\mathbf{I} - 4\nu(m+2l+1)\sum_{n=l+1}^{N}(n-l)(l+m+n+1)\right)a_n^m - b_l^m = 0 \tag{3.75}$$

should be satisfied. The matrix form of Equation 3.75 at m^{th} mode can be represented by

$$\mathbf{\Theta}(m)\mathbf{a}\left(m\right) = \mathbf{b}\left(m\right), \tag{3.76}$$

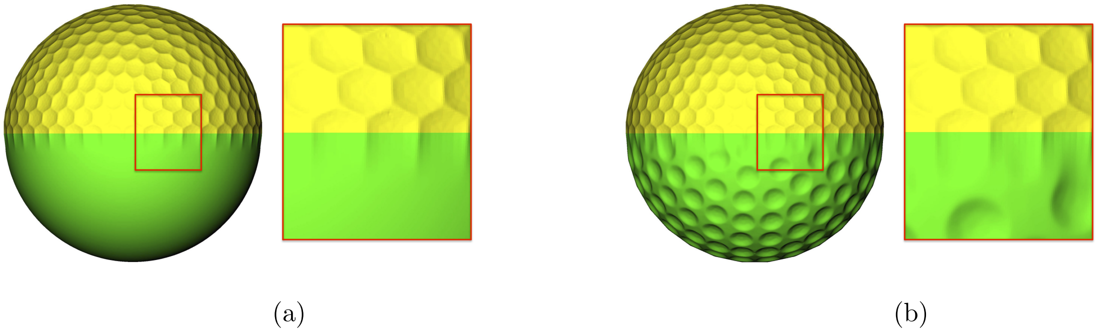
where $\mathbf{\Theta}(m) = \mathbf{I} - \nu\mathbf{A}\left(m\right)$, a matrix $\mathbf{A}(m)$ is

$$\mathbf{A}(m) = [A(m)_{ij}] = \begin{cases} 4(m+2i+1)(j-i)(i+m+j+1), & i \leqq j-1 \\ 0, & otherwise \end{cases} \tag{3.77}$$

$\mathbf{a}(m) = [a_0^m,\ a_1^m, \cdots,\ a_N^m]^T$ and $\mathbf{b}(m) = [b_0^m,, b_1^m, \cdots,\ b_N^m]^T$. Equation 3.76 cannot be solved because $\mathbf{\Theta}(m)$ is a singular matrix. To solve it, boundary conditions should be imposed on $\mathbf{\Theta}(m)$ and $\mathbf{b}\left(m\right)$. We impose boundary conditions by replacing the bottom most row of matrix $\mathbf{\Theta}(m)$ with $V_n^m(1)$ and the highest mode of $\mathbf{b}(m)$ with $\hat{f}_m$, derived at Eq.(3.68). By defining a matrix $\mathbf{\Theta}^\star(m)$ to be equal to $\mathbf{\Theta}(m)$ with the bottom most row replaced and a column vector $\mathbf{b}^\star(m)$ to be equal to $\mathbf{b}(m)$ with the highest mode replaced, we can obtain the boundary condition imposed equation

$$\mathbf{\Theta}^\star(m)\mathbf{a}(m) = \mathbf{b}^\star(m) \tag{3.78}$$

Once Equation 3.78 is solved at each m-th mode, we can finally obtain all coefficients , a_l^m, of constrained geometries.



(a)

(b)

Figure 3.7: Morphing hemispherical golf balls. Two diffrent types of golf balls are morphed at the mid-plane with C_1 continuity: (a) a smooth ball is morphed with a hexagonal golf ball, (b) a circular golf ball is morphed with a hexagonal golf ball.

Solving 2^{nd} order PDE with the shape constraint can be easily extended in solving higher order PDE with shape, slope or curvature geometric constraints. For example, to impose shape and slope geometric constraints, the following 4^{th} order PDE is solved.

$$\left(1-\nu_1\nabla^2+\nu_2\nabla^4\right)r^c{=}r^u \tag{3.79}$$

Equation 3.79 also can be rewritten as the matrix form, given as

$$\mathbf{H}(m)\mathbf{a}\left(m\right)=\mathbf{b}\left(m\right), \tag{3.80}$$

where $\mathbf{H}(m)=\mathbf{I}-\nu_1\mathbf{A}\left(m\right)+\nu_2\mathbf{A}^2\left(m\right)$. To impose the boundary conditions on $\mathbf{H}$, the most bottom two rows of matrix $\boldsymbol{\Theta}(m)$ are replaced with $V_l^m(1)$ and $\left.\frac{\partial V_l^m(u)}{\partial u}\right|_{u=1}$, and the two highest modes of $\mathbf{b}(m)$ are replaced with $\hat{f}_m$ and $\hat{f}_m^1$, respectively. Defining a matrix $\mathbf{H}^\star(m)$ to be equal to $\mathbf{H}(m)$ with the two bottom most rows replaced and a column vector $\mathbf{b}^\star(m)$ to be equal to $\mathbf{b}(m)$ with two highest modes replaced gives

$$\mathbf{H}^\star(m)\mathbf{a}(m)=\mathbf{b}^\star(m) \tag{3.81}$$

By solving Equation 3.81 at every m^{th} mode, a constrained object assuring satisfaction of shape and slope constraints at the equator can be obtained.

Our method of imposing boundary constraints is quite versatile. It can be utilized to create new types of objects from existing objects. For example, Figure 3.7 shows the surface of a hybrid golf ball. In Figure 3.7 (a), the top half of the hybrid ball is the top half of the baseline hexagonally-dimpled golf ball shown in Figure 2.2 (b). The bottom half of the hybrid ball is the bottom half of a smooth golf ball in Figure 2.2 (c). The halves are joined at the equator by making the bottom half of the hybrid golf ball, which is a 1-hole sub-object, constrained so that its boundary at the equator has the same radial location, slope, and curvature at the equator as the unconstrained top half of the hybrid golf ball , which is also a 1-hole sub-object. In Figure 3.7 (b), the top half of this hybrid ball is still the top half of the baseline, hexagonally-dimpled golf ball shown in Figure 3.7 (b), but the bottom half of the hybrid is now the the bottom half of the baseline, round-dimpled golf ball shown in Figure 3.7 (a). They are joined at the equator by the same way how the hybrid golf ball in Figure 3.7 (a) is joined.

3.4 Imposing geometric constraints on a patch

To explain how to impose geometric constraints on a patch, we first explain a way in solving a fourth order elliptical PDE. Then, we will describe a fourth order PDE to impose constraints on a patch. Although we focused our explanation here to a fourth order PDE, it can be easily modified to second or six order PDEs according to geometric constraints ones have to imposed on a patch.

Solving elliptical equation with basis functions for a patch

Suppose the following 4^{th} order elliptical PDE is needed to be solved in a rectangular domain whose ranges of u and v are $-1 \leq u, v \leq 1$

$$\frac{d^4 z(u,v)}{du^4} + \frac{d^4 z(u,v)}{dv^4} = F(u,v) \tag{3.82}$$

with the following Dirichlet and Neumann boundary conditions

$$\begin{aligned} z(\pm 1, v) &= z^c_{u=\pm 1}(v) \\ z(u, \pm 1) &= z^c_{v=\pm 1}(u) \\ \tfrac{\partial z}{\partial u}(\pm 1, v) &= s^c_{u=\pm 1}(v) \\ \tfrac{\partial z}{\partial v}(u, \pm 1) &= s^c_{v=\pm 1}(u) \end{aligned} \tag{3.83}$$

By choosing the Chebyshev-Gauss-Lobatto collocation points [9], Equation 3.82 can be rewritten as

$$\frac{\partial z^4(u_i, v_j)}{\partial u^4} + \frac{\partial z^4(u_i, v_j)}{\partial v^4} = F(u_i, v_j) \tag{3.84}$$

where $= 0, \cdots, N$ and $j = 0, \cdots, N$. In a matrix form, Equation 3.84 can be expressed as

$$\mathbf{D_p^4 z{+}z} \left(\mathbf{D_p^4}\right)^{\mathbf{T}} \mathbf{=F} \tag{3.85}$$

$\mathbf{D_p}$, $\mathbf{z}$ and $\mathbf{F}$ are $(N+1)$ by $(N+1)$ matrices, where $\mathbf{D_p}$ is a differentiation matrix defined in Equation 2.87. The components of $\mathbf{z}$ and $\mathbf{F}$ are

$$\mathbf{z} = \begin{bmatrix} z_{00} & z_{10} & \cdots & z_{N0} \\ z_{01} & z_{11} & \cdots & z_{N1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{0N} & z_{1N} & \cdots & z_{NN} \end{bmatrix}, \mathbf{F} = \begin{bmatrix} F_{00} & F_{10} & \cdots & F_{N0} \\ F_{01} & F_{11} & \cdots & F_{N1} \\ \vdots & \vdots & \ddots & \vdots \\ F_{0N} & F_{1N} & \cdots & F_{NN} \end{bmatrix} \tag{3.86}$$

Here, we used notation z_{ij} and F_{ij} instead of $z(u_i, v_j)$ and $F(u_i, v_j)$, respectively, for convenience. Then, let define $\mathbf{E} = \mathbf{D_p^4}$, which makes Equation 3.85

$$\mathbf{Ez} + \mathbf{zE^T} = \mathbf{F} \tag{3.87}$$

Solving Equation 3.87 is well explained in [9]. Let's assume a modified version of Equation 3.87 can be written as

$$\mathbf{\Theta^\star z^\star} + \mathbf{z^\star \Theta^{\star T}} = \mathbf{F^\star} \tag{3.88}$$

where $\mathbf{z}^\star$ is an interior matrix $(2 \leq i, j \leqq N-1)$ of $\mathbf{z}$ and $\mathbf{\Theta}^\star$ and $\mathbf{F}^\star$ are matrices which can be obtained from Equation 3.87 by doing some calculation after boundary conditions are incorporated to the equation. To solve Equation 3.88 efficiently**,** $\mathbf{\Theta}^\star$ is diagonalized by decomposing $\mathbf{\Theta}^\star$ into

$$\mathbf{\Theta^\star} = \mathbf{PQP^{-1}} \tag{3.89}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are eigenvector and eigenvalue matrices of $\mathbf{\Theta}^\star$, respectively. Substituting Equation 3.89 into Equation 3.88 and multiplying $\mathbf{P^{-1}}$ give

$$\mathbf{Q\hat{z}} + \mathbf{\hat{z}\Theta^{\star T}} = \mathbf{\hat{F}} \tag{3.90}$$

where $\mathbf{\hat{z}} = \mathbf{P^{-1}z^\star}$ and $\mathbf{\hat{F}} = \mathbf{P^{-1}F^\star}$**.** After transposing Equation 3.90, the k^{th} column vector can be represented as

$$(\mathbf{\Theta^\star} + \lambda_k \mathbf{I})\mathbf{\hat{z}}_k^{\mathbf{T}} = \mathbf{\hat{F}}_k^{\mathbf{T}}, \quad k = 0, \cdots, N-3 \tag{3.91}$$

where λ_k is k-th eigenvalue of $\mathbf{\Theta}^\star$. Thus, for each k^{th} column, the $\mathbf{\hat{z}}_k$ can be calculated. Once $\mathbf{\hat{r}}$ is calculated, $\mathbf{z}^\star$ can be easily obtained from

$$\mathbf{z^\star} = \mathbf{P\hat{z}} \tag{3.92}$$

Imposing geometric constraints

To impose shape and slope constraints on three hole objects, we solve

$$\frac{\partial z(u,v)}{\partial t} = \mu_1 \nabla^2 z(u,v) - \mu_2 L z(u,v), \tag{3.93}$$

where μ_1 and μ_2 are dynamic diffusivities. Here, ∇^2 is defined as

$$\nabla^2 = \frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2}, \tag{3.94}$$

and L is defined as

$$L = \frac{\partial^4}{\partial u^4} + \frac{\partial^4}{\partial v^4}. \tag{3.95}$$

If we substitute Equations 3.94 and 3.95 in to Equation 3.96, we obtain

$$\frac{\partial z(u,v)}{\partial t} = v_1 \left(\frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2} \right) z(u,v) - v_2 \left(\frac{\partial^4}{\partial u^4} + \frac{\partial^4}{\partial v^4} \right) z(u,v) \tag{3.96}$$

Define z^u as height of an unconstrained patch and r^c as height of a constrained patch. Using the implicit first time step method gives

$$z^c(u,v) - \nu_1 \left(\frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2} \right) z^c(u,v) + \nu_2 \left(\frac{\partial^4}{\partial u^4} + \frac{\partial^4}{\partial v^4} \right) z^c, (u,v) = z^u(u,v) \tag{3.97}$$

where ν_1 and ν_2 are diffusitivies.

Similar to previous study, as a matrix form, Equation 3.97 can be written as

$$(\mathbf{I} + \mathbf{E})\, \mathbf{z}^c + \mathbf{z}^c \mathbf{E}^{\mathbf{T}} = \mathbf{z}^{\mathbf{u}}, \tag{3.98}$$

where $\mathbf{E} = -\nu_1 \mathbf{D}_{\mathbf{p}}^{\mathbf{2}} + \nu_2 \mathbf{D}_{\mathbf{p}}^{\mathbf{4}}$. Each matrix in Equation 3.98 is a $N+1$ by $N+1$ matrix. After imposing shape and slope constraints on Equation 3.98 and diagonalizing in u, we finally obtain

$$(\mathbf{\Theta}^{\star} + (\lambda_k + 1)\mathbf{I})\hat{\mathbf{z}}_k^T = \hat{\mathbf{F}}_k^T \tag{3.99}$$

as a counterpart of Equation 3.91. It can be efficiently solved with the same procedure which described in Subsection 3.4.

Chapter 4

Construct and optimize complex non-star-shaped objects

We have explained how simple a star-shaped object is represented as $0, 1, 2$-hole object or a patch with our basis functions and how geometric constraints are imposed on those objects. Our method can be easily applied to represent a complex non-star shaped object. Our philosophy in representing a complex non-star shaped object is avoiding a complex mapping and join several constrained sub-objects together. In other words, a complex object is constructed by several sub-objects and the sub-objects are constrained to have the equal shape, slope and/or curvature at the joints. In this chapter, we will describe the way how a non-star shaped object is represented, how non-star shaped objects are morphed and eventually how a morphed non-star shaped object is created. To assist our understanding, we will present specific examples of a non-star shape objects. What we choose as examples is two different shaped airplanes, an SR-71 and one that assembles the shape of a Naboo Cruiser in the Star Wars movie, which is not real but imaginary airplane. These two airplanes are presented in Figure 4.2.

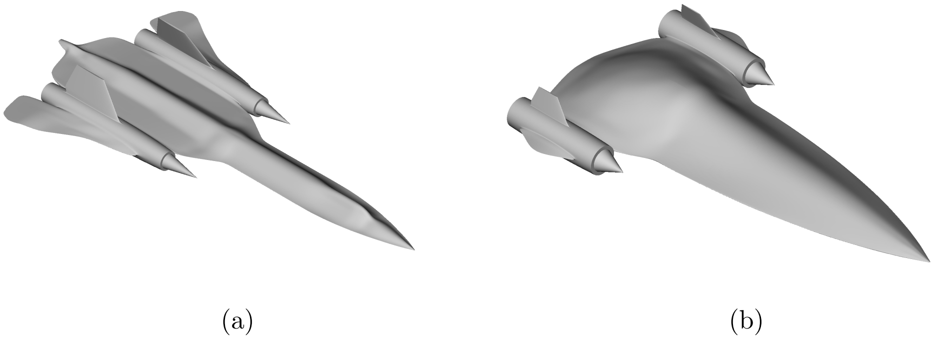
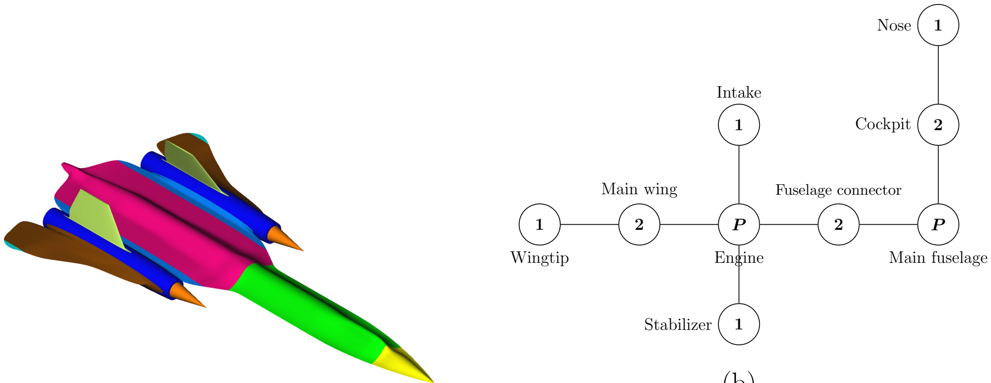


Figure 4.1: The 3D shapes of an (a) SR-71 and (b) Naboo fighter.



(b)

Figure 4.2: An SR-71 can be made by joining several constrained sub-objects. (a) An SR-71 airplane is made from fifteen sub-objects, each shown with a different color. (b) The wiring diagram for the SR-71. Only the center and left half of the airplane is depicted in the wiring diagram. The number n, or letter "P" or "M" inside the circles in the wiring diagrams indicate that the sub-objects are an n-hole object, a Patch, or a Multi-hole object, respectively.

4.1 Construction of non-star shaped objects

In our method, a complex object is constructed by stitching several constrained sub-object together by matching shape, slope or curvature at the interfaces. The structure of a complex object shows how the object is reconstructed from sub-objects. This structure is plotted as a wiring diagram where sub-objects are presented as circles and lines connecting circles means joints of two or more sub-objects. The numbers or letter "P“ inside of circles indicates that sub-objects are represented as n-hole objects or patches respectively. For example, an SR-71 we considered is created by joining 15 constrained sub-objects together. The left panel of Figure 4.2 (a) shows 15 constrained sub-objects of an SR-71 illustrated by different colors. The wiring diagram presented in the right panel of Figure 4.2 (b) shows types of primitives used for each sub-object. At each joint, every sub-object is stitched to assure C_1 continuity except the joint where intake and engine are connected. At that joint, we believe C_0 continuity is good enough. Here, it should be mentioned that only half of an SR-71 is shown in the diagram because the airplane has the same connectivity at the other half.

Another complex shaped object example is presented in Figure 4.3. The left panel of Figure 4.3 shows a hand that was created by joining 11 constrained sub-objects together. The constrained open boundaries make the shape and slope of the hand continuous at all locations. The wiring diagram of the hand shown in the right panel of Figure 4.3 shows that the hand was made up of a multi-holed palm joined to five fingers with each finger made up of a constrained 2-holed part joined to a constrained 1-holed fingertip. Here, multi-holed palm is created by stitching two patches.

The number and type of sub-objects needed to represent an object and how they are

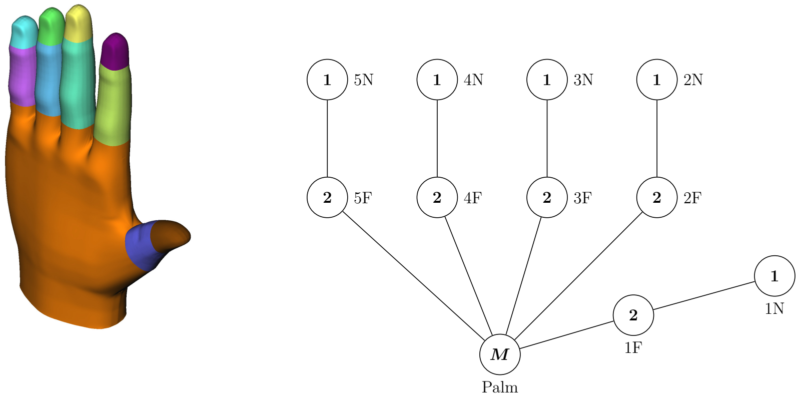


Figure 4.3: A hand can be made by joining several constrained sub-objects. (a) A hand is made from eleven constrained sub-objects. Each sub-object has a different color. (b) The wiring diagram that shows what each sub-object is and how it is connected to the other sub-objects.

wired together are not unique and somewhat arbitrary. The rule of thumb that we have adopted is to try to have as few sub-objects as possible. For example, it might seem more natural to represent a finger with three sub-objects because a finger has three joints, but breaking the finger into only two pieces gives an adequate representation. One motivation for choosing a particular wiring diagram for a complex object made of sub-objects is that at the current time, a necessary condition for us to morph between or among two or more objects is that all of objects have the same wiring diagram. Once all objects have the same wiring diagram, morphing of a whole object is implemented by morphing each sub-object with its correspondence object. For example, we can animate a hand by morphing a straight hand, shown at the leftmost figure of Figure 4.4, with a curved hand, shown at the rightmost figure of Figure 4.4. Morphing of these hands is implemented by morphing each sub-object with its correspondence object. In this case, it is essential that the hands with the straight and bent fingers should have the same wiring diagrams. Similarly, Figure 4.5 shows an SR-71 airplane morphing into a fictional Naboo cruiser airplane from the movie Star Wars. Since these two airplanes have identical wiring diagrams, morphing occurs each sub-object is morphed wit its correspondence object.

The fact that morphing of sub-objects is implemented by morphing them with their correspondence sub-objects is the biggest limitation in our morphing methodology because we are currently unable to morph two or more objects that are made up of sub-objects unless all of the objects have the same wiring diagram. For example, a hand with four fingers has a different wiring diagram than one with five fingers. Therefore in general, it would not be possible to morph a four-fingered hand into a five-fingered hand. One possible way to overcome this difficulty might be to have a more general wiring diagram that could represent

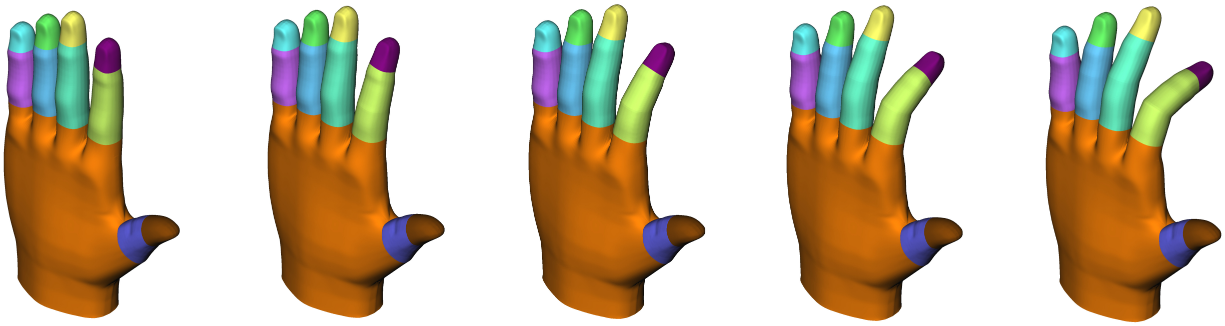


Figure 4.4: A hand with straight fingers (leftmost) bending its fingers. The hand is made of 11 constrained sub-objects like the hand in Figure 4.3 (a).

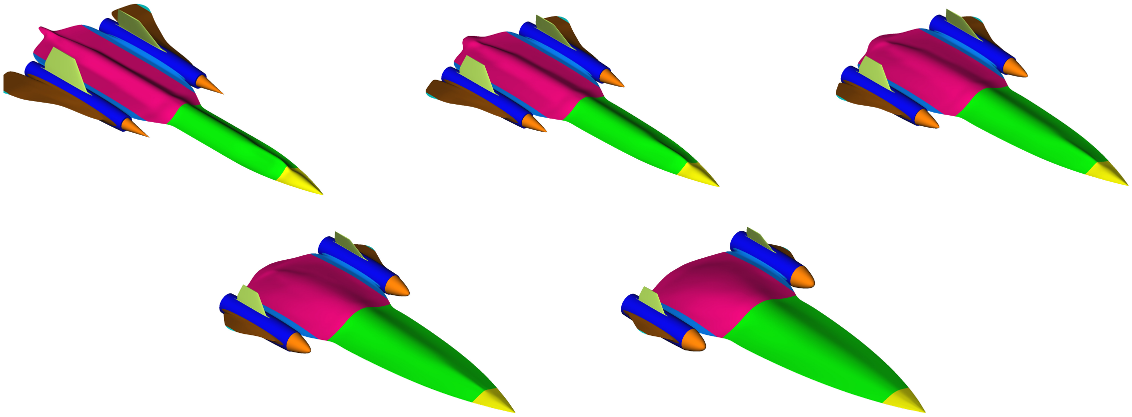


Figure 4.5: An SR-71 (leftmost panel) made up of 15 constrained sub-objects as in Figure 4.3 (b) is morphed into a Naboo cruiser (rightmost panel). The three airplanes between the SR-71 and Naboo cruiser are shown at various stages of the morphing.

both types of hands (with one of the fingers vestigial in the four-fingered hand). In general, the question of how to break an object efficiently into sub-objects and create a sensible wiring diagram is difficult although there has been some recent progress in this area, such as the skeleton-based and convex-based segmentation methods [41, 42]. Our future work will focus on robust methods for breaking up an objects into sub-objects where those sub-objects are specifically n-hole constrained sub-objects and constrained patches.

4.2 Optimization of an airplane by morphing an SR-71 and Naboo cruiser

Imposition of geometric constraints on sub-objects of an airplane

With the representation of airplanes described in Section 4.1, we implemented shape optimization of an airplane by morphing an SR-71 and Naboo cruiser to maximize its lift-to-drag ratio, L/D, where L and D are lift and drag force of an airplane, respectively. A way of morphing an SR-71 and Naboo cruiser is morphing each sub-object of an SR-71 with its correspondence sub-object of a Naboo cruiser. When each sub-object is morphed, if we use an independent weight per sub-object, a morphed airplane before geometric constraints are imposed, which is called an unconstrained morphed airplane, may have very different shape at interfaces. For example, Figure 4.6 shows an example of an unconstrained airplane in the left panel. As this figure illustrate, an unconstrained morphed airplane may have very distinctive shape per sub-object, therefore, simply connecting these sub-objects creates an airplane that under-performs due to phenomenons such as flow separation, flow recirculation and stress concentration at discontinuous locations. However, if we impose geometric constraints on sub-objects by using constraint solvers described in Section 3, we can obtain a constrained airplane whose interfaces of sub-objects are seamlessly connected. An example of a constrained airplanes is plotted in the right panel of Figure 4.6. In this section, before we explain design variables, simulation conditions and results of the shape optimization problem of an airplane, we first want to how each part of a morphed airplane is approximated by our spectral basis functions and how shape and slope of a morphed airplane be continuous at all joints. To help our understanding, we painted a diagnostic objects as purple, while unconstrained and constrained objects are painted as green.

Cockpit

The surface of a morphed unconstrained cockpit is a 2-hole object. If we define $\underline{X}_{CP} = [x_{CP}(u),\, y_{CP}(u,v),\, z_{CP}(u,v)]^\intercal$ as morphed position functions of a morphed unconstrained cockpit (green object in Figures 4.7a and 4.7b), it can be written as

$$\underline{X}_{CP}(u,v) = \sum_{l=0}^{N} \sum_{m=-N}^{N-1} \left(\hat{\underline{X}}_{CP}\right)_l^m T_l^m(u) e^{imv}, \tag{4.1}$$

where $(\hat{\underline{X}}_{CP})_l^m = [(\hat{x}_{CP})_l^m,\, (\hat{y}_{CP})_l^m,\, (\hat{z}_{CP})_l^m]^\intercal$ and $(\hat{x}_{CP})_l^m$, $(\hat{y}_{CP})_l^m$ and $(\hat{z}_{CP})_l^m$ are respectively (l,m)-th spectral coefficients of $x_{CP}(u)$, $y_{CP}(u,v)$ and $z_{CP}(u,v)$, and u is Chebyshev-Gauss-Lobatto grid in Chebyshev domain and maps the surface of the cockpit in x-direction by

$$x_{CP}(u) = \frac{x_{CP,e} - x_{CP,s}}{2} u + \frac{x_{CP,e} + x_{CP,s}}{2}. \tag{4.2}$$

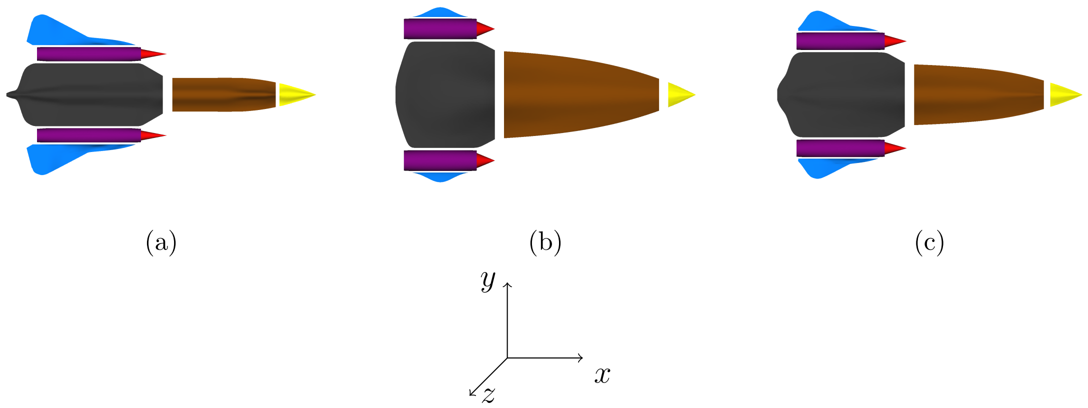

(a) (b) (c)

Figure 4.6: An SR-71 shown in (a) and Naboo cruiser shown in (b) are decomposed to one-hole, two-hole objects and patches. The corresponding objects of an SR-71 and a Naboo fighter are morphed each other. A nose (yellow) is one hole object. The cockpit (brown), stablizer and engines (plume) are two hole objects. Main fuselage (black) and wing (blue) are represented as upper and lower patches. The decomposed airplane in (c) is created by morphing an SR-71 and Naboo cruiser.

Here, $x_{CP,s}$ is the left most value of x_{CP}, which is mapped to $u = 1$, and $x_{CP,e}$ is the right most value of x_{CP}, which is mapped to $u = -1$. The value v is equally spaced from 0 to 2π in Fourier domain and is parameterized with azimuthal angle, ϕ, given by

$$\phi = v - \frac{\sin(2v)}{2}. \tag{4.3}$$

To demand $\underline{X}_{CP}$ have the same shape and slope at $u = -1$ with those of main fuselage on the interface, we create a 2-hole diagnostic object, shown as a purple object in Figure 4.7. We define position functions of this diagnostic object as $\underline{X}_{FD}(u, v) = [x_{FD}(u),\, y_{FD}(u, v),\, z_{FD}(u, v)]^{\intercal}$. Similarly, $\underline{X}_{FD}(u, v)$ is represented by 2-hole basis functions with the following parameterization.

$$\begin{aligned} x_{FD}(u) &= \frac{x_{FD,e} - x_{FD,s}}{2} u + \frac{x_{FD,e} + x_{FD,s}}{2} \\ y_{FD}(u, v) &= \frac{y_{FD,e}(u) - y_{FD,s}(u)}{2} \cos v + \frac{y_{FD,e}(u) + y_{FD,s}(u)}{2}, \end{aligned} \tag{4.4}$$

where $x_{FD,s}$ is the left most value of x_{FD}, $x_{FD,e}$ is the right most value of x_{FD}, $y_{FD,s}(u)$ is the lowermost value of y_{FD} at each u and $y_{FD,e}(u)$ is the uppermost value of y_{FD} at each u. The height of the diagnostic object $z_{FD}(u, v)$ is correspondence positions on points (x_{FD}, y_{FD}). With the representation of a morphed unconstrained cockpit and a diagnostic object for the cockpit, a morphed constrained cockpit can be created by solving 4^{th} order PDEs, given as

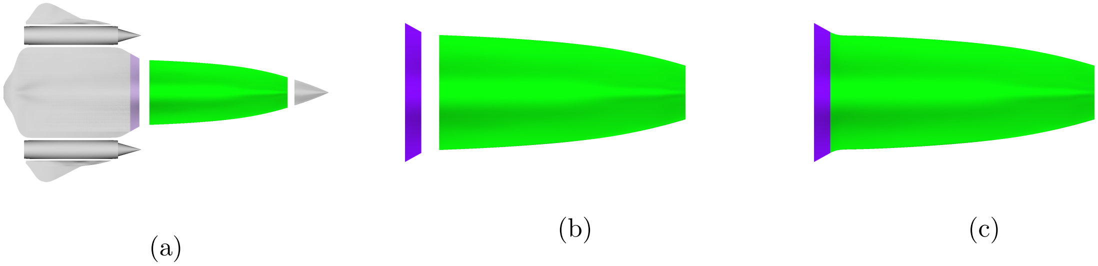


Figure 4.7: The shape of the cockpit (green object) with its diagnostic object (purple object). (a) overall shape of cockpit and its diagnostic object (b) unconstrained cockpit with diagnostic object (c) constrained cockpit with diagnostic object. The diagnostic object is removed after the constrained cockpit is obtained.

$$(1-\nu_1\nabla^2+\nu_2\nabla^4)\underline{X}^C_{CP}(u,v)=\underline{X}_{CP}(u,v), \tag{4.5}$$

where ∇^2 is defined as

$$\nabla^2=\frac{\partial^2}{\partial u^2}+\frac{\partial^2}{\partial v^2} \tag{4.6}$$

and $\underline{X}_{CP}(u,v)^C$ is position functions of a morphed constrained cockpit. The boundary conditions of Equation 4.29 inducing C_1 continuity at the joint are given as

$$\begin{aligned}\underline{X}^C_{CP}|_{u=-1}&=\underline{X}_{FD}|_{u=1}\\ \underline{X}^C_{CP}|_{u=1}&=\underline{X}_{CP}|_{u=1}\\ \frac{d\underline{X}^C_{CP}}{du}|_{u=-1}&=k_{CP}\frac{d\underline{X}_{FD}}{du}|_{u=1}\\ \frac{d\underline{X}^C_{CP}}{du}|_{u=1}&=\frac{d\underline{X}_{CP}}{du}|_{u=1}\end{aligned} \tag{4.7}$$

Therefore, by solving Equation 4.29 with boundary conditions in Equation 4.7, a morphed constrained cockpit can be obtained.

Nose

If position functions of a morphed unconstrained nose, defined as $\underline{X}_N(u,v)=[x_N(u),\, y_N(u,v),\, z_N(u,v)]^\intercal$ in Cartesion coordinate, is approximated by 1-hole object basis functions, which can be written as

$$\underline{X}_N(u,v)=\sum_{l=0}^{N}\sum_{m=-N}^{N-1}\left(\hat{\underline{X}}_N\right)_l^m W_l^m(u,v), \tag{4.8}$$

where $(\underline{\hat{X}}_N)_l^m = [(\hat{x}_N)_l^m,\ (\hat{y}_N)_l^m,\ (\hat{z}_N)_l^m]^\intercal$ are spectral coefficients' functions of the nose and $(\hat{x}_N)_l^m$, $(\hat{y}_N)_l^m$ and $(\hat{z}_N)_l^m$ are respectively (l, m)-th spectral coefficients of $x_N(u)$, $y_N(u, v)$ and $z_N(u, v)$, and

$$W_l^m(u, u) = V_l^m(u) e^{imv}. \tag{4.9}$$

Here, $V_l^m(u) = \xi^{|m|} P_l^{(0,|m|)}(s)$, $s = 2u^2 - 1$ and s is Chebyshev grid points. Since the range of is $-1 \leq s \leq 1$, the range of u becomes $0 \leq u \leq 1$. The function $P_l^{(0,|m|)}(s)$ is one set of Jacobi polynomials, $P_l^{(\alpha,\beta)}(s)$, where α and β are set to be $\alpha = 0$ and $\beta = |m|$. When the surface of the nose is approximated by 1-hole basis functions, it is parameterized to be a pole when $u = 0$ and to be the bottom of the nose when $u = 1$. Further, u linearly maps x coordinate of the surface of a morphed unconstrained nose, given as

$$x_N(u) = (x_{N,e} - x_{N,s})u + x_{N,s}, \tag{4.10}$$

where $x_{N,s}$ and $x_{N,e}$ are rightmost and leftmost values of the nose. For v, we use the same parameterization which is used to represent the cockpit, written in Equation 4.3. To obtain a constrained morphed nose, we solve

$$(1 - \nu_1 \nabla_u^2 + \nu_2 \nabla_u^4) \underline{X}_N^C(u, v) = \underline{X}_N(u, v), \tag{4.11}$$

where $\underline{X}_N^C$ is position functions of a morphed constrained connector and ∇_u^2 is defined as

$$\nabla_u^2 = \frac{1}{u} \frac{\partial}{\partial u} \left(u \frac{\partial}{\partial u} \right) + \frac{1}{u^2} \frac{\partial^2}{\partial v^2}. \tag{4.12}$$

The boundary conditions of Equation 4.29 are

$$\begin{aligned} \underline{X}_N^C|_{u=1} &= \underline{X}_{CP}^C|_{u=1} \\ \frac{d\underline{X}_N^C}{du}|_{u=1} &= k_N \frac{d\underline{X}_{CP}^C}{du}|_{u=1}. \end{aligned} \tag{4.13}$$

It should be noticed that only two boundary conditions are needed instead of four because when u is zero, the object is forced to be located on the pole. Both unconstrained and constrained morphed noses are presented in Figure 4.8 with the constrained cockpit, shown in Figure 4.8a. After solving Equation 4.11 with boundary conditions given in Equation 4.13, the unconstrained nose, shown in Figure 4.8b, is constrained. After constraints are imposed, its surface is seamlessly stitched with the constrained cockpit.

Fuselage

We define the surface of a fuselage represented in Figure 4.9a as $\underline{X}_F^p = [x_F^p,\ y_F^p,\ z_F^p]^\intercal$, where $p = 1$ or 2 indicates upper and lower patch of the fuselage, respectively. Then, the fuselage $\underline{X}_F^p$ is decomposed into main fuselage $\underline{X}_{MF}^p$, shown as a gray object in Figure 4.9b, and two

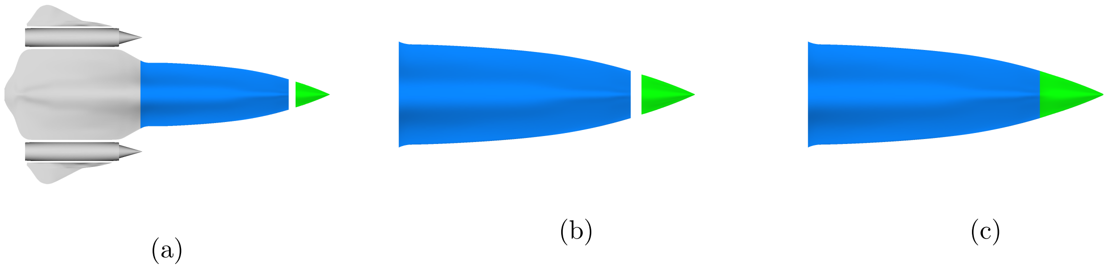


Figure 4.8: The shape of the nose (green object) with constrained cockpit (blue object). The overall shape of the nose and the constrained fuselage are presented in (a). After solving Equation 4.11 with boundary conditions given in Equation 4.13, the unconstrained nose, shown in (b), is constrained and modified to the constrained nose as shown in (c).

connectors, shown as green objects in Figure 4.9b. The position functions of a connector in Cartesion coordinate is defined as $\underline{X}_{CN}(u,v) = [x_{CN}(u,v), y_{CN}(u), z_{CN}(u,v)]^\intercal$. Here, a connector is made by stitching the green colored upper and lower patches in Figure 4.9b so that a connector $\underline{X}_{CN}$ is a 2-hole object. Since a SR-71 plane is mirrored along y plane, we only consider one connector. If we summarize our definition so far, mathematically, $\underline{X}_F^p$, $\underline{X}_{MF}^p$ and $\underline{X}_{CN}$ have the following relation:

$$\underline{X}_F^1 \cup \underline{X}_F^2 \equiv \underline{X}_{MF}^1 \cup \underline{X}_{MF}^2 \cup \underline{X}_{CN} \cup (\underline{X}_{CN})^{MIR}, \tag{4.14}$$

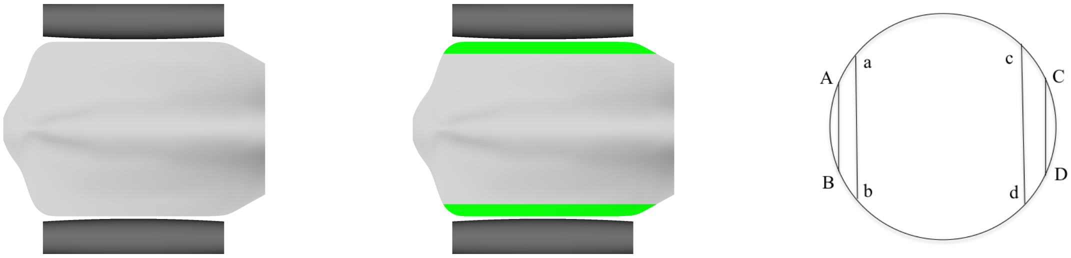
where $(\underline{X}_{CN})^{MIR}$ is the mirrored surface of $\underline{X}_{CN}$ respect to y-plane. The surface of the connector $\underline{X}_{CN}(u,v)$ is represented by Chebyshev polynomial and Fourier series, using the same parameterization with $\underline{X}_{FD}(u,v)$. Thus, parameterization for $\underline{X}_{CN}(u,v)$ can be written as

$$\begin{aligned} y_{CN}(u) &= \frac{y_{CN,e} - y_{CN,s}}{2} u + \frac{y_{CN,e} + y_{IW,s}}{2} \\ x_{CN}(u,v) &= \frac{x_{CN,e}(u) - x_{CN,s}(u)}{2} \cos v + \frac{x_{CN,e}(u) + x_{CN,s}(u)}{2}, \end{aligned} \tag{4.15}$$

where $y_{CN,s}$ and $y_{CN,e}$ are the lowermost and uppermost values of y_{CN}, and $x_{CN,s}(u)$ and $x_{CN,e}(u)$ are the leftmost and rightmost values of x_{CN} at each u, respectively. Then, $z_{CN}(u,v)$ is the height of the object at the correspondence positions on (x_{CN}, y_{CN}). With this parameterization, the surface of a morphed unconstrained connector is approximated as a summation of our 2-hole basis functions multiply spectral coefficients,

$$\underline{X}_{CN}(u,v) = \sum_{l=0}^{N} \sum_{m=-N}^{N-1} \left(\hat{\underline{X}}_{CN}\right)_l^m T_l^m(u) e^{imv}, \tag{4.16}$$

where $(\hat{\underline{X}}_{CN})_l^m = [(\hat{x}_{CN})_l^m, (\hat{y}_{CN})_l^m, (\hat{z}_{CN})_l^m]^\intercal$ and $(\hat{x}_{CN})_l^m$, $(\hat{y}_{CN})_l^m$ and $(\hat{z}_{CN})_l^m$ are respectively (l,m)-th spectral coefficients of $x_{CN}(u,v)$, $y_{CN}(u)$ and $z_{CN}(u,v)$.



(a) (b) (c)

Figure 4.9: The shape of the fuselage and connectors. A fuselage, shown as a gray object in (a), is decomposed into a main fuselage and two connectors as shown and colored as green in (b). The cross-section of an engine is shown in (c). Here, “ABba” is the location where a connector is asked to be joined with the engine.

To demand $\underline{X}_{CN}$ have the same shape and slope at the interface, we create a 2-hole diagnostic object, shown as a purple object in Figure 4.10. The cross-sectional view of a diagnostic object by the x-plane is plotted in Figure 4.9c. We define position functions of this diagnostic object as $\underline{X}_{IWD}(u, v) = [x_{IWD}(u),\, y_{IWD}(u, v),\, z_{IWD}(u, v)]^\intercal$. The surface of a diagnostic object $\underline{X}_{IED}(u, v)$ is then represented by Chebyshev polynomials and Fourier series, mapped to be on ”a-b” plane at $u = 1$ and on ”A-B” plane at $u = -1$. Since the shape and slope of a morphed unconstrained connector are required to be equal to those of a diagnostic object at the interface, we can define the following boundary conditions as geometric constrains of a morphed unconstrained connector,

$$\begin{aligned} \underline{X}^C_{CN}|_{u=-1} &= \underline{X}_{CN}|_{u=-1} \\ \underline{X}^C_{CN}|_{u=1} &= \underline{X}_{IED}|_{u=-1} \\ \frac{d\underline{X}^C_{CN}}{du}|_{u=-1} &= \frac{d\underline{X}_{CN}}{du}|_{u=-1} \\ \frac{d\underline{X}^C_{CN}}{du}|_{u=1} &= k_{CN}\frac{d\underline{X}_{IED}}{du}|_{u=-1}, \end{aligned} \tag{4.17}$$

where $\underline{X}^C_{CN}$ is position functions of a morphed constrained connector. Then, the surface of a morphed constrained connector can be obtained by solving

$$(1 - \nu_1\nabla^2 + \nu_2\nabla^4)\underline{X}^C_{CN}(u, v) = \underline{X}_{CN}(u, v), \tag{4.18}$$

with boundary conditions, given in Equation 4.17, where ∇^2 is defined in Equation 4.6. Figure 4.10 shows a constrained connector from the unconstrained connector. It should be noticed that here a diagnostic object is created to calculate geometric constraints for a morphed unconstrained connector. After geometric constraints are obtained, the diagnostic object is no longer necessary.

(a) (b)

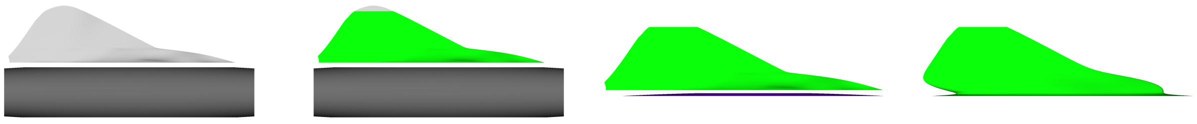
Figure 4.10: The shape of the unconstrained and constrained connectors. An unconstrained connector in (a) is constrained and the constrained connector is shown in (b). A diagnostic object in (a) is created to calculate geometric conditions at the interface of the connector and engine.

Wing

Imposing constraints on a morphed unconstrained wing, shown in Figure 4.11a, is similar with imposing constraints on the fuselage. Similar to representing the surface of the fuselage, we define position functions of the wing in Cartension coordinate as $\underline{X}_W^p = [x_W^p,\, y_W^p,\, z_W^p]^\intercal$, shown as a gray object in Figure 4.11a, where p is 1 or 2 indicates the upper or lower patch of the wing, respectively. Then, $\underline{X}_W^p$ is divided to a morphed unconstrained main wing $\underline{X}_{MW}^p$ and a morphed unconstrained wingtip $\underline{X}_{WT}$ which are shown as a green and gray objects in Figure 4.11b, respectively. Again, remember that $\underline{X}_W^p$ are patches, while $\underline{X}_{MW}$ is a 2-hole object. The surface of the main wing is approximated by our 2-hole basis functions where its position functions are mapped with the same parameterization what have been used to represent the fuselage. So, if we define the surface positions of a morphed unconstrained main wing as $\underline{X}_{MW}^p = [x_{MW},\, y_{MW},\, z_{MW}]^\intercal$, it is approximated by our 2-hole basis functions,

$$\underline{X}_{MW}(u,v) = \sum_{l=0}^{N} \sum_{m=-N}^{N-1} \left(\hat{\underline{X}}_{MW}\right)_l^m T_l^m(u) e^{imv}, \tag{4.19}$$

where $(\hat{\underline{X}}_{MW})_l^m = [(\hat{x}_{CN})_l^m,\, (\hat{y}_{MW})_l^m,\, (\hat{z}_{MW})_l^m]^\intercal$ and $(\hat{x}_{MW})_l^m$, $(\hat{y}_{MW})_l^m$ and $(\hat{z}_{MW})_l^m$ are respectively (l,m)-th spectral coefficients of $x_{CN}(u,v)$, $y_{CN}(u)$ and $z_{MW}(u,v)$. To demand $\underline{X}_{MW}$ have the same shape and slope at the interface, we create a 2-hole diagnostic object, shown as a purple object in Figure 4.11. We define position functions of this diagnostic object as $\underline{X}_{OED}(u,v) = [x_{OED}(u),\, y_{OED}(u,v),\, z_{OED}(u,v)]^\intercal$. The surface of a diagnostic object $\underline{X}_{OED}(u,v)$ is then represented by Chebyshev polynomials and Fourier series, mapped to be on "c-d" plane at $u = -1$ and on "C-D" plane at $u = 1$ in Figure 4.9c. Since the shape and slope of a morphed unconstrained connector are required to be equal to those of a diagnostic object at the interface, we can define the following boundary conditions as geometric constrains of a morphed unconstrained main wing,



(a) (b) (c) (d)

Figure 4.11: The shape of the unconstrained and constrained main wings, wingtip and engine. The wing, shown as a gray object in (a), is decomposed into the unconstrained main wing and wingtip which are plotted as green and gray objects in (b). To impose geometric constraints on the unconstrained main wing, a diagnostic object is created in (c). As a result of constraints imposition, the unconstrained main wing in (c) is changed to the constrained main wing plotted in (d).

$$\begin{aligned} \underline{X}_{MW}^{C}|_{u=-1} &= \underline{X}_{OED}|_{u=-1} \\ \underline{X}_{MW}^{C}|_{u=1} &= \underline{X}_{MW}|_{u=1} \\ \frac{d\underline{X}_{MW}^{C}}{du}|_{u=-1} &= k_{MW}\frac{d\underline{X}_{OED}}{du}|_{u=-1} \\ \frac{d\underline{X}_{MW}^{C}}{du}|_{u=1} &= \frac{d\underline{X}_{MW}}{du}|_{u=1}, \end{aligned} \tag{4.20}$$

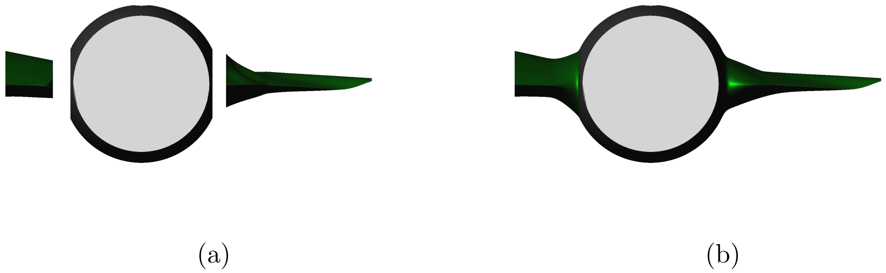
where $\underline{X}_{MW}^{C}(u,v)$ is position functions of a morphed constrained main wing. By solving the following equation with boundary conditions, given in Equation 4.20, we can obtain the surface of a morphed constrained main wing

$$(1-\nu_1\nabla^2+\nu_2\nabla^4)\underline{X}_{MW}^{C}(u,v) = \underline{X}_{MW}(u,v), \tag{4.21}$$

where ∇^2 is given in Equation 4.6. The shape of the constrained main wing with the constrained main fuselage is presented in Figure 4.12. As a result of the imposition of geometric constraints , the fuselage, engine and wing are smoothly connected maintaining C_1 continuity.

Stablizer

A morphed unconstrained stablizer can be treated as a 2-hole object. We define surface of the morphed unconstrained stablizer as $\underline{X}_S$ and unit normal vector at the bottom plane of the stablizer as $\hat{n_s}$. Also, suppose we define rotation matrix, $\mathbf{R}$, which makes unit normal vector $\hat{n_s}$ be $\hat{z}$ and rotated shape of $\underline{X}_S$ by $\mathbf{R}$ as $\underline{X}_S^R$. Here, we define surface of the rotated stablizer as $\underline{X}_S^R(u,v) = [x_S^R(u,v),\, y_S^R(u,v),\, z_S^R(u)]^{\intercal}$. We first parameterize z_S^R and $x_S^R(u)$ with Chebyshev polynomial grid point u and equally spaced (from 0 to 2π) Fourier grid points v



(a) (b)

Figure 4.12: Deformation of the main fuselage and main wing after geometric constraints are imposed. The unconstrained main fuselage and main wing presented in (a) become the constrained main fuselage and main wing as presented in (b) after the imposition of geometric constraints. After they are constrained, the fuselage, engine and wing are smoothly connected by maintaining C_1 continuity at the interface.

$$\begin{aligned} z_S^R(u) &= \frac{z_{S,e}^R - z_{S,s}^R}{2}u + \frac{z_{S,e}^R + z_{S,s}^R}{2} \\ x_S^R(u,v) &= \frac{x_{S,e}^R(u) - x_{S,s}^R(u)}{2}\cos v + \frac{x_{S,e}^R(u) + x_{S,s}^R(u)}{2}, \end{aligned} \tag{4.22}$$

where $z_{S,s}^R$ and $z_{S,e}^R$ are the lowermost and uppermost values of z_S^R, and $x_{S,s}^R(u)$ and $x_{S,e}^R(u)$ are the leftmost and rightmost values of x_S^R at each u. Then, $y_S^R(u,v)$ is represented as the correspondence positions of rotated stablizer in y on (x_S^R, z_S^R). Under the this parameterization, the stablizer is required to be connected with the engines when $u = -1$. Once $\underline{X}_S^R(u,v)$ is obtained, $\underline{X}_S(u,v)$ can be easily calculated from

$$\underline{X}_S(u,v) = \mathbf{R}^{\mathsf{T}}\underline{X}_S^R(u,v) \tag{4.23}$$

Then, position functions of $\underline{X}_S$, defined as $\underline{X}_S(u,v) = [x_S(u,v),\ , y_S(u,v),\ , z_S(u,v)]^{\mathsf{T}}$, are approximated as a summation of 2-hole basis functions multiplies by spectral coefficients

$$\underline{X}_S(u,v) = \sum_{l=0}^{N}\sum_{m=-N}^{N-1}\left(\hat{\underline{X}}_S\right)_l^m T_l^m(u)e^{imv}, \tag{4.24}$$

where $(\hat{\underline{X}}_S)_l^m = [(\hat{x}_S)_l^m,\ (\hat{y}_S)_l^m,\ (\hat{z}_S)_l^m]^{\mathsf{T}}$ and $(\hat{x}_S)_l^m$, $(\hat{y}_S)_l^m$ and $(\hat{z}_S)_l^m$ are respectively (l,m)-th spectral coefficients of $x_S(u,v)$, $y_{CN}(u,v)$ and $z_S(u,v)$.

To insert a stablizer to engine by matching the shape and slope at the interface, we first create a rotated diagnostic object. Suppose we define $\underline{X}_{SD,B}^R(v)$ as the projected bottom shape of the rotated stablizer onto z-plane, which can be written as

$$\underline{X}_{SD,B}^R(v) = [x_S^R(-1,v),\ y_S^R(-1,v),\ 0]^{\mathsf{T}} \tag{4.25}$$

Also, let define a center point of $\underline{X}^R_{SD,B}(v)$ as $\underline{X}^R_{SP,C} = [x_{SD,C}, y_{SD,C}, 0]^\intercal$. Then, we create a rotated diagnostic object whose position functions are defined as $\underline{X}^R_{SD} = [x^R_{SD},\ y^R_{SD},\ z^R_{SD}]^\intercal$ as follows. We parameterize x^R_{SD} and y^R_{SD} of the rotated diagnostic object as

$$\begin{aligned} x^R_{SD}(u,v) &= \Big(\frac{k_s-1}{2}u + \frac{k_s+1}{2}\Big)(x^R_S(-1,v) - x^R_{SD,C}) + x^R_{SD,C} \\ y^R_{SD}(u,v) &= \Big(\frac{k_s-1}{2}u + \frac{k_s+1}{2}\Big)(y^R_S(-1,v) - y^R_{SD,C}) + y^R_{SD,C}, \end{aligned} \tag{4.26}$$

where k_s is a scale factor. Since the surface of the rotated diagnostic object is defined, defining the surface of the engine rotated by the same rotation matrix makes the calculation easy. If $\underline{X}^R_E = [x^R_E, y^R_E, z^R_E]^\intercal$ is defined as the position functions of the engine rotated by $\mathbf{R}$, $z^R_{SD}(u,v)$ can be obtained by interpolating z values on grid $(x^R_{SD}(u,v),\ y^R_{SD}(u,v))$ from $\underline{X}^R_E$. Since $\underline{X}^R_{SD}(u,v)$ is computed, the diagnostic object for the stablizer can be easily obtained from

$$\underline{X}_{SD}(u,v) = \mathbf{R}^\intercal \underline{X}^R_{SD}(u,v), \tag{4.27}$$

where $\underline{X}_{SD}(u,v)$ is position functions of a diagnostic object for a morphed unconstrained stablizer. To assure C_1 continuity conditions at the interface, the following constraints should be imposed on a stablizer,

$$\begin{aligned} \underline{X}^C_S|_{u=-1} &= \underline{X}_{SD}|_{u=-1} \\ \underline{X}^C_S|_{u=1} &= \underline{X}_S|_{u=1} \\ \frac{d\underline{X}^C_S}{du}|_{u=-1} &= k_{SD}\frac{d\underline{X}_{SD}}{du}|_{u=-1} \\ \frac{d\underline{X}^C_S}{du}|_{u=1} &= \frac{d\underline{X}_S}{du}|_{u=1}, \end{aligned} \tag{4.28}$$

when the following PDE is solved

$$(1 - \nu_1\nabla^2 + \nu_2\nabla^4)\underline{X}^C_S(u,v) = \underline{X}_S(u,v), \tag{4.29}$$

where $\underline{X}^C_S(u,v)$ is position functions of a morphed constrained stablizer and ∇^2 is given in Equation

4.3 Shape optimization of airplanes

In Section 4.2, ways of how morphed sub-objects are parameterized and constrained are explained. In this section, based on predescribed ways, we will try to find the optimum shape of an airplane by morphing an SR-71 and Naboo cruiser to maximize lift-to-drag

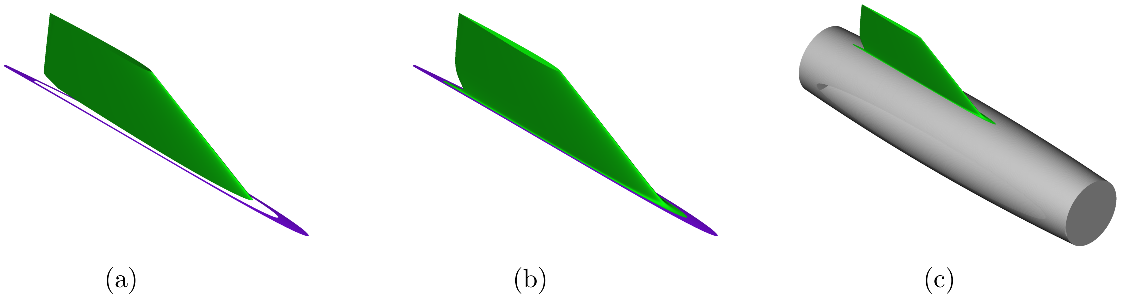


Figure 4.13: The shape of a stablizer, shown as a green object, and its diagnostic object, shown as a blue object. An unconstrained stablizer, presented in (a), is constrained to match the shape and slope of a diagnostic object where they are joined. A constrained stablizer, presented in (b), is smoothly inserted into the top of an engine, as shown in (c).

ratio. The main purpose of this optimization is to demonstrate the flexibility of our design-by-morphing method so that any physical constraints related in designing airplanes such as a supersonic area rule have not been considered in finding the optimal shape of an airplane.

When a SR-71 is morphed with a Naboo cruiser and the morphed airplanes are simulated to find the optimum shape of an airplane, although each sub-object can have independent weights, we do not consider 15 different weights to optimize the shape of airplanes. Rather, we divide the sub-objects into three groups, roughly modifying shapes of the fuselage plus engines, the cockpit plus nose and the wings. Every sub-object in the same group always have the same weight. Therefore, there are only three independent weights. We define ω_1, ω_2, and ω_3 as weights of the fuselage plus engines, the cockpit plus nose, and wings of the SR-71, respectively, while $1-\omega_1$, $1-\omega_2$, and $1-\omega_3$ are defined as weights of those of the Naboo cruiser. The optimization problem we solve in this example is defined as follows:

$$\begin{aligned} \text{minimize} \quad & -\frac{L}{D} \\ & 0 \le \omega_1 \le 1 \\ & 0 \le \omega_2,\ \omega_3 \le 1.25, \end{aligned} \tag{4.30}$$

where L and D are the lift and drag force of the morphed airplane, respectively. The ranges of weights are chosen to prevent from creating morphed airplanes having ridiculous shapes like very thin wings which cannot endure drag force while they fly. The procedure creating a morphed airplane is graphically presented in Figure 4.14.

The adjustment of the morphed airplane shape by these three weights are presented in Figure 4.15. Since the values of three weights are independent, the geometry features at the interfaces are not matched at the beginning. However, by solving PDEs, these geometric discrepancies at the interfaces are removed, and seamlessly connected morphed airplanes are finally obtained. Some of the morphed airplanes along with their weights and lift-to-drag ratios are shown in Figure 4.17. The lift-to-drag ratios were computed when the airplane flies at a speed of Mach 1.4 and at 4-degree angle of attack. Our optimization scheme maximized

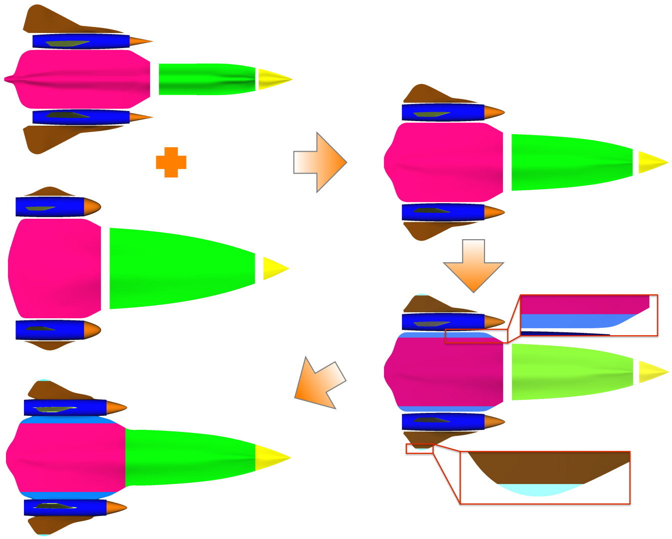


Figure 4.14: Procedures showing how a morphed airplane is created (top view). Sub-objects of an SR-71 and Naboo cruiser are morphed with their own corresponces, which creates morphed unconstrained airplane. The morephd unconstrained airplane is segmented further to have connectors and wingtips, and then, sub-objects are stitched continously to construct a morphed airplane.

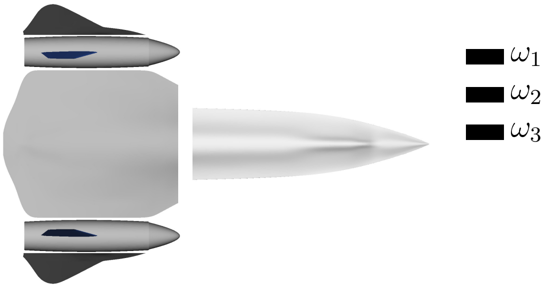


Figure 4.15: Three design groups of an airplane painted by different colors. The airplane here is before constrained, thus, sub-objects of the airplane are not matched geometric features at the attachment. Several examples of constrained airplanes are plotted in Figure 4.16.

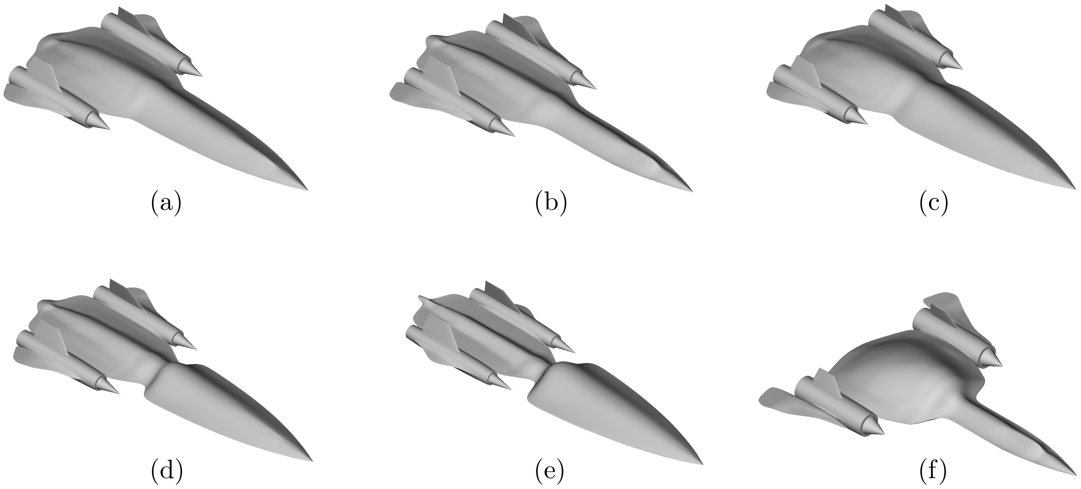


Figure 4.16: The shapes of morphed airplanes. Weights of morphed airplanes are (a) $[0.5, 0.5, 0.5]$ (b) $[0.77, 0.87, 0.71]$ (c) $[0.36, 0.19, 0.51]$ (d) $[0.75, 0.25, 0.5]$ (e) $[1, 0, 0]$ (f) $[0, 1, 1]$ where weights are ordered $[\omega_1, \omega_2, \omega_3]$.

the lift-to-drag ratio, and the optimal weights were found based on the optimization process that we used in optimizing the shape of a first car of a train. The results of the optimization are shown in Figure 4.18. The optimized airplane has a lift-to-drag ratio that is 23.12% greater than the ratio of the SR-71 and 51.73% greater than the ratio of a Naboo cruiser. In this optimization example, resolution of the simulation is low to avoid heavy computation work related to high resolved mesh. Due to this setting, values of the cost function may have low fidelity. However, through this study, it is shown that our design-by-morphing is so robust that it works well even an object undergone huge shape change and deformation at the interfaces.

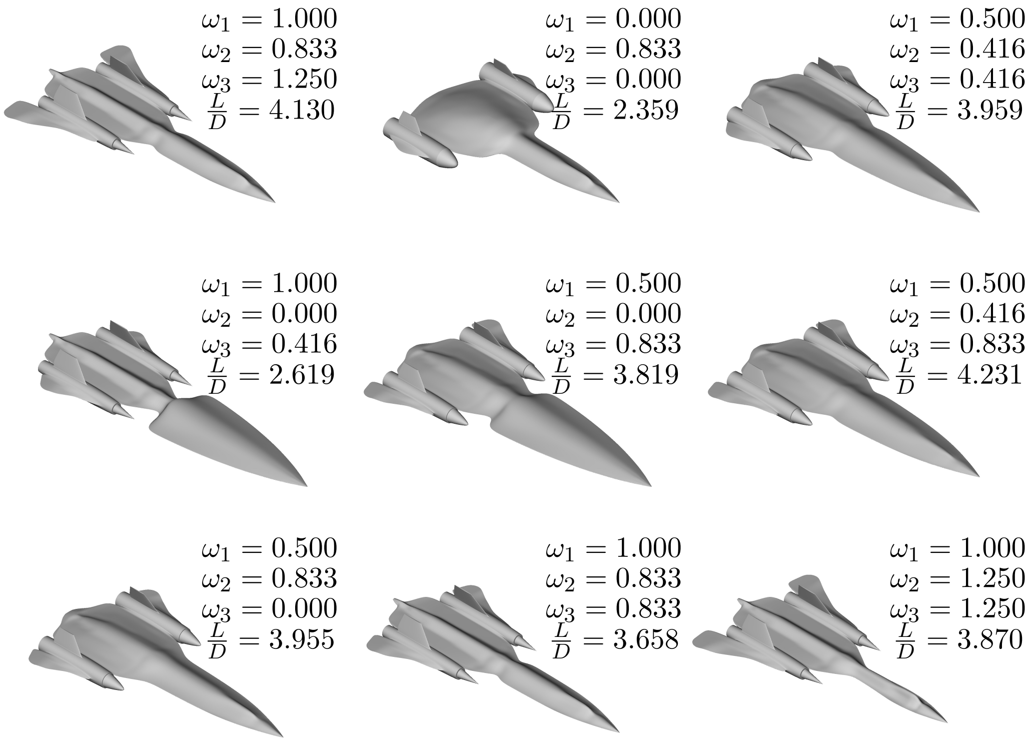


Figure 4.17: Examples of morphed airplanes created by morphing an SR-71 and Naboo cruiser. The weights and lift-to-drag ratios are also presented with the configurations of the morphed airplanes.

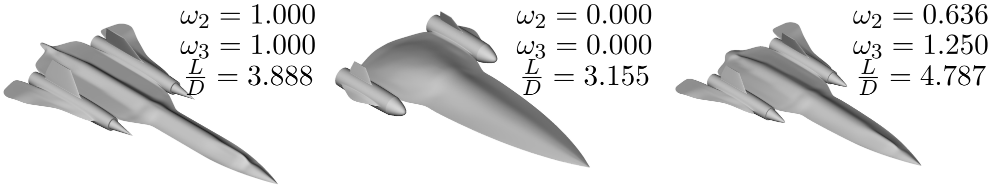


Figure 4.18: The SR-71 (left), naboo cruiser (middle), and the optimized airplane (right) are presented with their weights and lift-to-drag ratios.

Chapter 5

Shape Optimization of a Turbine-99 Draft tube

We applied design-by-morphing method to find the optimal shape of a turbine-99 draft tube to maximize its mean pressure recovery factor. We conducted this optimization in cooperation with NUMECA USA, INC.

5.1 Related work

The role of a draft tube in hydro-power plants is converting dynamic pressure of flow into static pressure by decelerating the velocity of the fluid at the runner passing through the draft tube. The influence of a draft tube becomes more significant at Kaplan turbine draft tubes [3] because entire performance of hydro-power plants is highly effected by the performance of the draft tubes. For example, a turbine-99 draft tube, which is installed at the exit of the runner of a Kaplan turbine and has been utilized in many hydro-power plants (about 20 were installed in Swedish power stations), is estimated to generate $3 \sim 4$ TWh more electricity per year if $0.3 - 2.3\%$ performance improvement of the draft tube is accomplished [3]. According to Andersson [3], this is the equal amount of electricity generation about 5% of the total Swedish hydropower production. However, a turbine-99 draft tube were simply created based on human intuition and simple analytic method. The shape of a turbine-99 draft tube has a sharp heel across the sections and is far away from the optimal configuration, which makes it underperformed for a long time. Therefore, simulating flow in the draft tube and optimizing the shape of it are significantly important in perspective of efficiently generating, converting and managing energy.

Several researchers have been carried out the shape optimization of a turbine-99 draft tube. The most common way related to redesigning this sharp heel draft tube is removing a sharp heel by introducing radius R of the heel to create a rounded heel draft tube. In fact, in 1970's, Gubin [22] reported that $0.3 - 2.3\%$ efficiency loss takes place due to the sharp heel of a turbine-99 draft tube. About two decades later, Dahlbäck [12] performed model tests to measure the turbine efficiency improvement with two redesigned draft tubes having rounded heels at three different blade angle opening conditions. With the shape chosen from the model tests, Dahlbäck carried out a prototype test and achieved in order of 0.5% improved turbine efficiency. Based on this work, Marjavaara and Lundström [34] attempted to compute the optimum radius R of the heel that induces maximum mean pressure recovery factor or minimum energy loss coefficient. They chose six different radius R for initial design of experiment, constructed individual cubic response curves of two objective functions and computed optimum radius using the constructed response curves. However, the improvements rates of mean pressure recovery factor and energy loss coefficient are smaller than their simulation uncertainty. It indicates either their computational modeling of fluid flow is not good enough to reflect effects of a rounded heel turbine-99 draft tube or a rounded heel turbine-99 draft has no effects compared to a sharp heel turbine-99 draft tube in terms of pressure recovery factor and energy loss coefficient. In the later work of Marjavaara et al. [33], instead of optimizing the shape of a turbine-99 draft tube, they optimized the simplified draft tube to maximize mean pressure recovery factor at two different operating modes. The simplified draft tube they considered is a 3D extension of a cone section of a turbine-99 draft tube which is connected directly to a simplified diffuser with the exclusion of an elbow section. They took this simplified draft tube into account because the simplified draft tube makes CFD easier and less computation work is needed to simulate the flow. Although, with five design parameters, pressure recovery factor was improved about 14.4% and 8.9% at each operation mode, the exclusion of an elbow part causes huge flow field discrepancy with the one of a turbine-99 draft tube, so that, the result obtained from the simplified draft tube cannot be applied and compared to a turbine-99 draft tube.

As described so far, ways of optimizing a turbine-99 draft tube have been very limited. There is no big difference even we broaden the scope to shape optimization of other types of draft tubes. For a Francis turbine draft tube, Marjavaara and Lundström [35] optimized the shape of it with three design parameters (left, right and upper end diffuser wall angles) based on CFD simulation by using standard $k - \epsilon$ model with wall function and surrogate based optimization technique. However, in their study, improvement rate resulting from the redesigning is not that impressive. Recently, Shojaeefard et al. [48] maximized the pressure recovery factor and minimized the energy loss coefficient for a draft tube of Agnew microhydro turbines using the Pareto solution [20] by adjusting cone angle and the height of the draft tube.

The most interesting work may be implemented byCoelho [11]. In their work, they proposed a new draft tube of a bulb turbine having the same inlet and outlet area ratio and approximately the same efficiency, but whose length is significantly reduced. This is meaningful achievement because people can more economically manufacture the draft tube than before.

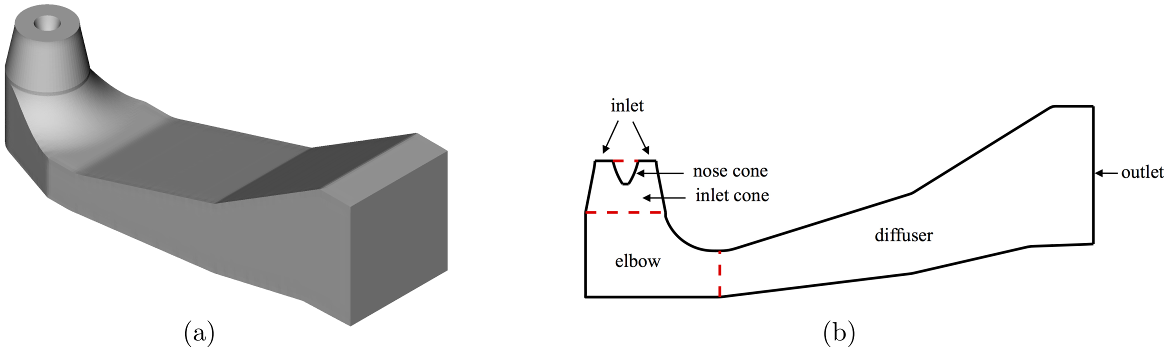


Figure 5.1: The shape of a turbine-99 draft tube. (a) the 3D shape of a turbine-99 draft tube (b) cross sectional view of a turbine-99 draft tube at the mid-plane.

5.2 Proposed design method

What do we mean by morphing?

The new proposed design method here, design-by-morphing, is based on morphing techniques. Although morphing technique is not unique, morphing should be robust, automated without human interaction to be used in optimization process. Since geometric constraints, for example shapes at certain locations are requested to be predetermined shapes, are often demanded to be imposed on a newly designed object, a way to impose geometric constraints is also needed in morphing technique. In our design-by-morphing, all of these requirements are reflected. It is so robust that do not break down and automated which does not require human intervention. Furthermore, any geometric constraints such as shape, slope or curvature constraints can be easily imposed on the boundaries of an object.

To explain how we morph an object, we will first illustrate how to represent 1D boundary of a 2D object, shown in Figure 5.2a. Using a polar coordinate system (r, θ) with the origin in the center of the object, the object's shape can be described by the radius $r(\theta)$ of its boundary. We approximated $r(\theta)$ as a truncated Fourier series as a function of azimuthal angle θ, which can be expressed as

$$r(\theta) \simeq \sum_{m=-M}^{M} a_m \, e^{i\, m\, \theta}. \tag{5.1}$$

This method has the advantage that if the shape of the object is sufficiently smooth, the series converges exponentially, which is called spectral representation.

Spectral representations of shapes can be easily extended to 3D space. Consider a cylindrical object as shown in Figure 5.2b. The shape of this object can be described by the cylindrical radius $r(u, \theta)$ of its surface along the centerline of the object, where u is mapping from the bottom to top boundaries of the object along the centerline. The radius $r(u, \theta)$ is approximated by

$$r(u, \theta) \simeq \sum_{l=0}^{L} \sum_{m=-M}^{M} a_{lm} T_l(u) e^{im\theta}, \tag{5.2}$$

where $T_l(u)$ is Chebyshev polynomial order of l and a_{lm} is the (l, m) spectral coefficient.

Our design-by-morphing method treats multiple baseline objects and a morphed object is created by linearly combining baseline objects with different weights. When the surfaces of objects are represented by a truncated spectral sum, morphing is implemented by computing a new set of spectral coefficients that are weighted averages of the spectral coefficients of the baseline objects. For example, a morph created from N baseline objects where the k^{th} baseline object has spectral coefficients $a_{lm}(k)$ has

$$r(u, \theta) \simeq \sum_{l=0}^{L} \sum_{m=-M}^{M} A_{lm}\ P_l(u)\, e^{i\,m\,\theta}, \tag{5.3}$$

where

$$A_{lm} \equiv \sum_{k=0}^{K} \omega_k\ a_{lm}(k). \tag{5.4}$$

Here, ω_k is the weight of the k^{th} baseline object. It is often useful, but not necessary to make the sum of the weights equal to unity. For example, when $K = 1$, ω_1 is dependent on ω_0 by $\omega_1 = 1 - \omega_0$. Some examples of morphed draft tubes when $K = 1$ with weights' summation dependency are presented in Figure 5.3. As the figure shows, by varying ω_0 from zero to unity, shapes of morphed draft tubes translates from one baseline object to the other. This is called interpolation morphing because morphed objects are shaped between two baseline objects.

As Equation 5.4 shows, the shape of a morphed object is effected by weights and spectral coefficients of baseline objects. Since spectral coefficients of baseline objects are quantities related to how baseline objects look like, once baseline objects are chosen, spectral coefficients of baseline objects are no longer changed. As a result, the shape of a morphed object is only effected by weights of baseline objects; thereby, the optimal shape of an object can be found by seeking the optimum values of weights which produces the best performed object.

Imposition of geometric constraints

Often, new designed objects are demanded to obey geometric constraints. For example, a back of a leading car of a train is requested to be a certain predescribed shape to be connected by a following car of a train. A leading car of a train may be also needed to have certain slope and curvature at the back to seamlessly connects a following car of a train in order to avoid any unnecessary aerodynamic features such as flow separation and

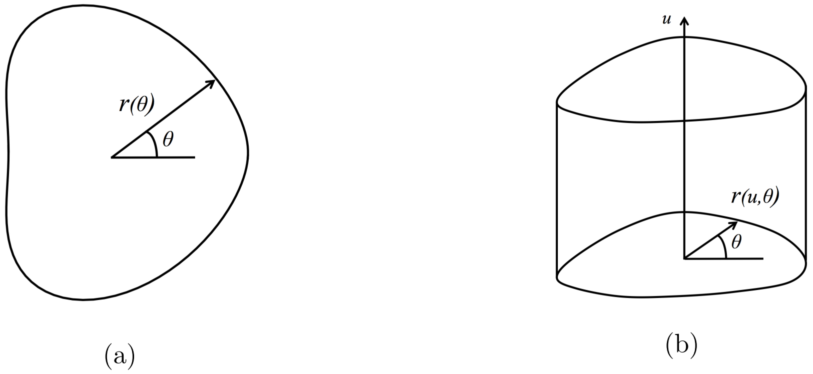


Figure 5.2: Schematic to illustrate an arbitrary shape of a (a) 2D object (b) 3D object. The radius of a 2D object is approximated by Fourier expansion while the cylindrical radius of a 3D object is approximated by Chebyshev polynomials and Fourier expansion.

recirculation at the connection. To deal with these requirements, our new design method should have a capability of imposing geometric constraints on an object. We handle it by solving high order PDEs with the proper boundary conditions where boundary conditions are incorporated with geometric constraints what we want to impose on an object. By solving PDEs, regardless of satisfactory of geometric constraints of an unconstrained object, a constrained object is forced to obey all imposed geometric constraints. Simple example shapes of constrained objects are illustrated in Figure 5.4. Initially, each object was a right circular cylinders. Boundary conditions are then imposed at the top and bottom of each cylinder where top of the cylinder is constrained to be a square and bottom is constrained to be remained as a circle as it was and the slopes of the boundaries are constrained to be vertical. The difference between the shapes of Figures 5.4a and 5.4b are the values of the diffusivities where hte values are smaller in Figures 5.4a than the values in Figure 5.4b. The comparison of Figures 5.4a and 5.4b shows that the smaller values of diffusivities create thinner boundary layers. The fact the thickness of boundary layer is adjustable in design-by-morphing allows it to have more freedom in shape exploration since the thickness of a boundary layer determines how smoothly geometric constraints are imposed on objects and the surface of a constrained object is modified to obey imposed geometric constraints.

5.3 Morphing of draft tubes

In this section, as a preliminary step of finding the optimal shape of a turbine-99 draft tube, we will explain which parts of a turbine-99 draft tube are redesigned and how, and what geometric constraints are demanded to be imposed on the redesigned draft tubes. Then, we will show a variety of morphed draft tubes as examples.

When a turbine-99 draft tube is optimized, we did not redesign the shape of an inlet cone section of a turbine-99 draft tube. In fact, what we optimized is the shape of a nose

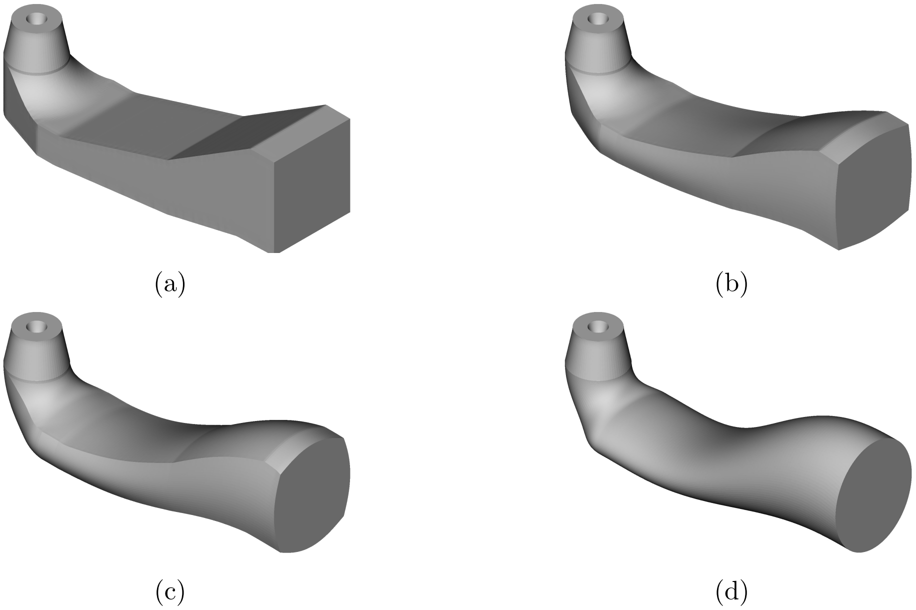


Figure 5.3: Morphed draft tubes from two baseline objects shown in (a) and (d). The weight of the original draft tube in (a) ,is defined as ω_1 and the weight the draft tube in (d), is defined as $1 - \omega_1$. The weights of morphed draft tubes from (a) to (d) are 0, 0.333, 0.667 and 1, respectivly.

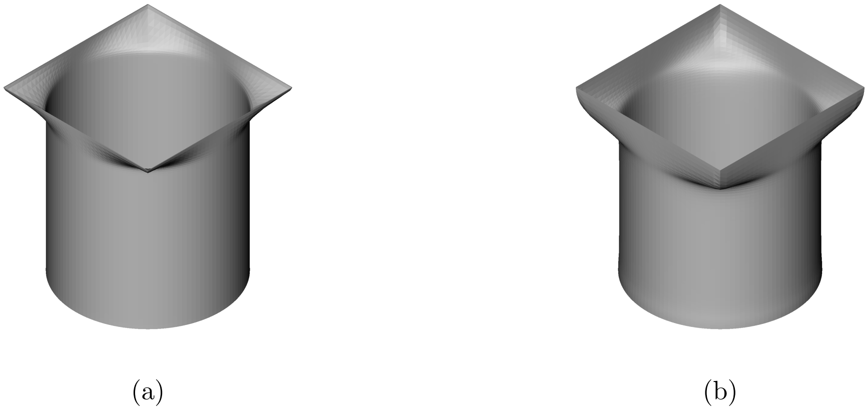


Figure 5.4: A cylinder constrained with geoemtric requirements. A cylinder is required to be a square at the top end and unchanged at the bottom. The slope at both ends are required to be straight along the vertical direction. The diffusivities applied to objects in (a) is smaller than the diffusivities applied to object in (b).

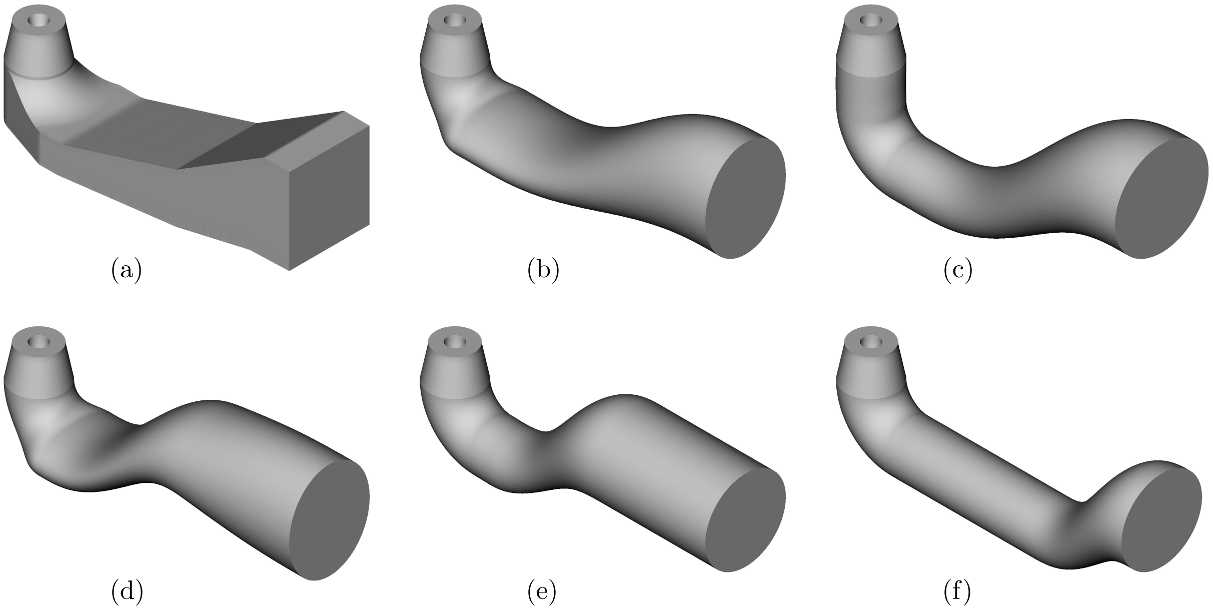


Figure 5.5: Six baseline draft tubes create from morphing. The weights of these draft tubes from (a) to (f) are respectively defined as ω_0, ω_1, $\cdots$, ω_5.

cone as well as the shapes of elbow and diffuser sections of a turbine-99 draft tube. Strictly speaking, ones may argue that a nose cone is not a part of a draft tube (rather it is a part of Kaplan turbine), but we assume that the interior shape of a turbine-99 draft tube is defined by the shape of a nose cone because the dented shape of an ingress part of a draft tube is defined by the configuration of a nose cone. In this sense, morphing was carried out twice; (1) to create the exterior shape of a draft tube and (2) to create the interior shape of a draft tube defined by the shape of a morphed nose cone.

Morphing to define the exterior shape of a draft tube

The new exterior shape of a draft tube is defined by the surfaces of morphed elbow and diffuser sections. To obtain the morphed exterior shape of a draft tube, we considered six baseline draft tubes and treated them as baseline draft tubes. These six baseline draft tubes are presented in Figure 5.5. The one shown in Figure 5.5a is an original turbine-99 draft tube, whose weight is defined as ω_0. We refer the original turbine-99 draft tube as a zeroth baseline draft tube. The other five baseline draft tubes shown in Figure 5.5b to Figure 5.5f are created base on intuition. For example, the shapes of these baseline draft tubes are defined by the accumulation of 2D cross-sectional boundaries. These 2D shapes are either perfect circles or ellipses whose radii or major and minor radii are a function of u which is a mapping parameter of each baseline draft tubes in its streamline direction. Also, each baseline draft tube has its own profile in streamline direction. It should be noticed that the shape of the baseline draft tube represented in Figure 5.5c is invented by mimicking the shape of the draft tube installed at Yngeredsforsen hydro-power plant in Sweden [35], which has a deeper elbow section than other baseline draft tubes. The weights of baseline draft tubes shown in Figure 5.5b to Figure 5.5f, which are respectively called the first and fifth baseline draft tubes, are defined as ω_1 to ω_5.

When baseline draft tubes are morphed, we demanded that a sum of the weights should be equal to unity, therefore, the weight of the fifth baseline draft tube, ω_5, can be obtained from

$$\omega_5 = 1 - \sum_{k=0}^{4} \omega_k. \tag{5.5}$$

Then, the exterior surface of a morphed draft tube is created by calculating a new set of spectral coefficients of spectral basis functions. If we define spectral coefficients of the k^{th} baseline draft tube as $a_{lm}(k)$, where double summations of $a_{lm}(k)$ multiplied by spectral basis functions approximates the radius of elbow and diffuser sections of the k^{th} baseline draft tube, radius of elbow and diffuser sections of a morphed draft tube, R, can be computed from

$$R(u,\theta) = \sum_{l=0}^{L} \sum_{m=-N}^{N} A_{lm} T_l(u) e^{im\theta}, \tag{5.6}$$

where

$$A_{lm} = \sum_{k=0}^{5} \omega_k a_{lm}(k). \tag{5.7}$$

Here, we want to remind that ω_5 is slaved to other weights by Equation 5.5 so that a new exterior shape of a draft tube is created from five independent parameters.

Several examples of morphed draft tubes are presented in Figure 5.7 with the weights. Since all weights in Figures 5.7a and 5.7b are positive, the morphed draft tubes shown in these figures are created from interpolation morphing. However, the morphed draft tubes shown in Figures 5.7c and 5.7d are created from extrapolation morphing because some of their weights are negative. Due to extrapolation morphing in design-by-morphing, characteristics of baseline draft tubes can be strengthened or weakened based on values of weights. One factor should be noticed here is that the shapes and slopes at egresses of the morphed draft tubes vary according to the values of weights used to create morphed draft tubes, which shows the necessity of geometric constraints impositions.

Morphing to define the interior shape of a draft tube

The new interior shape of a draft tube is defined by the surface of a nose cone. To obtain a morphed interior shape of a draft tube, we considered three differently shaped nose cones, which are presented in Figure 5.6. The nose cone shown in Figure 5.6a is the one of an original turbine-99 draft tube, which is called an original nose cone, while the others are

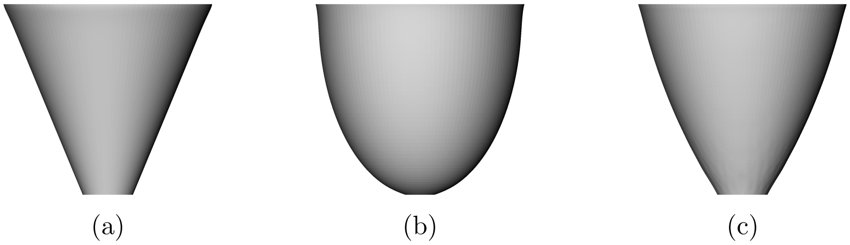


Figure 5.6: Six baseline draft tubes cover for morphing. The weights of these draft tube cover from (a) to (c) are defined as ω_{c1}, ω_{c2} and ω_{c3}.

invented from intuition. For example, the nose cone shown in Figure 5.6b is created by linearly connecting 2D profile of the original nose cone at the top to the bottom, while the nose cone shown in Figure 5.6c is made up from a part of an ellipsoid. By respectively treating the nose cones presented in Figure 5.6a to 5.6c as zeroth baseline nose cone to the second baseline nose cone, we define weights of these nose cones as ω_0^c, ω_1^c and ω_2^c, respectively and demand a sum of the weights be equal to unity. Here, we slaved the weight of a zeroth baseline nose cone to the others, which can be written as

$$\omega_0^c = 1 - \sum_{k=1}^{2} \omega_k^c. \tag{5.8}$$

Similar to creating the morphed exterior surface of a draft tube, the morphed interior surface is created by calculating a new set of spectral coefficients. However, for the morphing of a nose cone, we introduce one more design parameter which adjusts the height of a morphed nose cone. Then, if we define spectral coefficients of the k^{th} baseline nose cone as $a_{lm}^c(k)$, where double summations of $a_{lm}^c(k)$ multiplied by spectral basis functions approximates the radius of the k^{th} baseline nose cone, radius of a morphed nose cone, R^c, can be computed from

$$R^c(z(u), \theta) = \sum_{l=0}^{L} \sum_{m=-N}^{N} A_{lm}^c T_l(u(z)) e^{im\theta} \tag{5.9}$$

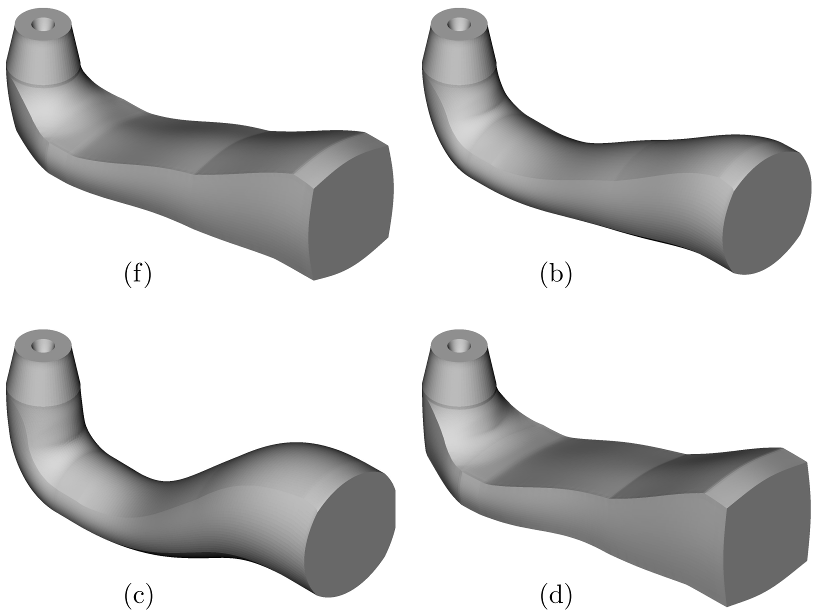
where

$$A_{lm}^c = \sum_{k=0}^{2} \omega_k^c a_{lm}^c(k), \tag{5.10}$$

and

$$z = \omega_h \frac{Hu}{2}. \tag{5.11}$$

Here, H is the height of the original nose cone and ω_h is the weight to control the height of a morphed nose cone. Since ω_0^c is subordinated to ω_1^c and ω_2^c as written in Equation 5.8, the number of independent parameters in determining the interior shape of a draft tube is three.



(f) (b)

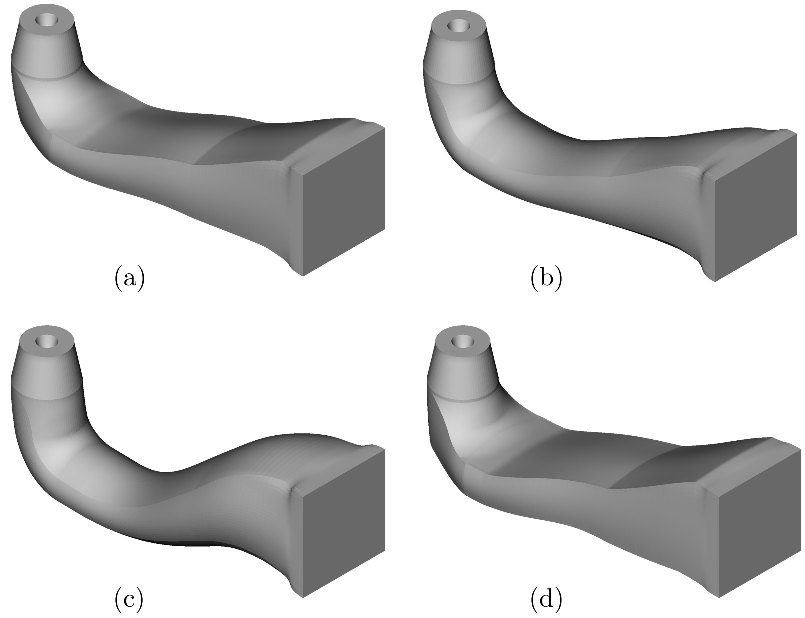
(c) (d)

Figure 5.7: Several examples of morphed draft tubes. The morphed draft tubes in (a) and (b) are created by interpolating baseline draft tubes, while the morphed draft tube in (c) and (d) are created by extrapolating baseline draft tubes. Weights of all morphed draft tubes from ω_1 to ω_6 are (a) (0.54, 0.12, 0.02, 0.18, 0.05, 0.09) (b) (0.14, 0.22, 0.28, 0.08, 0.12, 0.16) (c) (0.09, 0.34, 0.61, -0.18, 0.29, -0.15) (d) (0.69, -0.24, 0.17, -0.29, 0.58, 0.09).

Geometric constraints

In redesigning the shape of a turbine-99 draft tube, shapes and slopes at ingress and egress of a morphed draft tube and a morphed nose cone are demanded to be unchanged from those of the original draft tube to avoid any installation issues of a newly designed draft tube or mismatch between the new draft tube and other components of a hydro-power plant. Therefore, morphing should be carried out subject to imposition of geometric constraints at the ingress and egress of a new draft tube and nose cone.

To deal with the demanded constraints, a morphed draft tube and a morphed nose cone are imposed with the given geometric constraints. Examples of morphed draft tubes after imposition of geometric constraints are presented in Figure 5.8. The draft tubes shown in Figure 5.8 have the same weights with the ones shown in Figure 5.7, but the draft tubes in Figure 5.8 are the shapes after geometric constraints are imposed on them. Unlike the unconstrained draft tubes in Figure 5.7, the shape and slope at the boundaries of the constrained morphed draft tubes are always kept unchanged from those of the original draft tube. Similarly, several examples of morphed nose cones after imposition of geometric constraints are presented in Figure 5.9. Due to the imposition of geometric constraints, shape and slope at the ingress of the draft tube is always identical to those of the original draft



(a) (b)

(c) (d)

Figure 5.8: The morphed draft tubes presented in Figure 5.7 are constrained with geometric constraints.

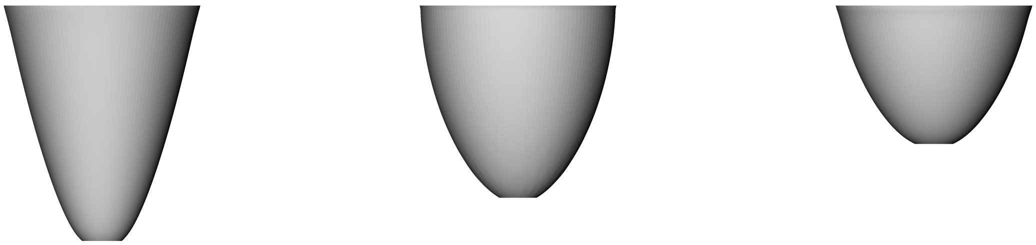


Figure 5.9: Several examples of morphed nose cones. Weights of all morphed nose cones from ω_1^c, ω_2^c to ω_h^c are (a) (0.78, 0.39, 1.31) (b) (−0.35, 0.53, 1.07) (c) (0.25, 0.48, 0.77).

tube.

5.4 Simulation and validation

Computational simulation

To carry out computational simulation, three dimensional Reynold-Averaged Navier-Stokes (RANS) equation is solved with hexagonally constructed grids inside a draft tube. To simulate flow, we used Numeca's multi-physics simulation software, FineOpen [24]. The second order time stepping scheme is used to evolve flow in time and turbulence is modeled and resolved by applying $k - \epsilon$ with extended wall function. The flow was simulated under the

T operational mode of a turbine-99 draft tube and the inlet boundary conditions were used provided in [18]. At the outlet, an averaged static pressure boundary condition was applied.

There are two significant quantities in measuring the performance of a turbine-99 draft tube. One is mean pressure recovery factor, C_{pm}, and the other is energy loss coefficient, ζ. These two quantities are defined as

$$C_{pm} = \frac{\frac{1}{A_{\mathbf{out}}} \iint_{A_{out}} p dA - \frac{1}{A_{\mathbf{in}}} \iint_{A_{\mathbf{in}}} p dA}{\frac{1}{2}\rho \left(\frac{Q_{\mathbf{in}}}{A_{\mathbf{in}}}\right)^2}, \tag{5.12}$$

and

$$\zeta = \frac{\int_{A_{\mathbf{in}}} p_{\mathbf{t}} \mathbf{u} \cdot \mathbf{n} dA - \int_{A_{\mathbf{out}}} p_{\mathbf{t}} \mathbf{u} \cdot \mathbf{n} dA}{\frac{1}{2}\rho \left(\frac{Q_{\mathbf{in}}}{A_{\mathbf{in}}}\right)^2}, \tag{5.13}$$

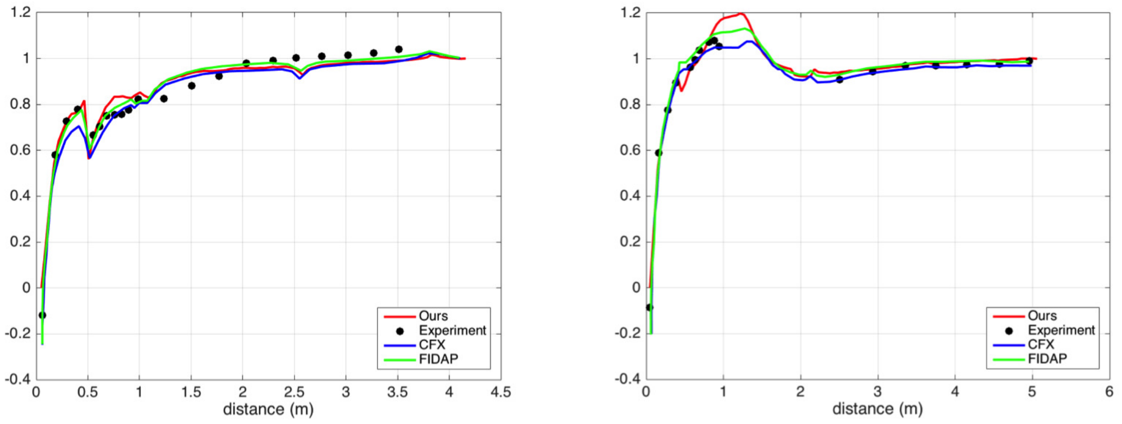
where p is static pressure, $p_{\mathbf{t}}$ is total pressure, ρ is density, A_{in} is area at inlet, A_{out} is area at outlet and Q is mass flow rate, $\mathbf{u}$ is flow velocity and $\mathbf{n}$ is unit normal vector. Mean pressure recovery factor shows that the degree of energy conversion from kinetic energy at the exit of the runner into pressure energy at the outlet of the draft tube. On the other hand, energy loss coefficient indicates the total energy loss of the flow by passing the draft tube. We believe these two quantities are highly correlated each other, which will be quantitatively shown in Section 5.6.

Grid sensitivity study and simulation verification

To simulate the flow in a draft tube, non-uniformly distributed mesh was generated. Many grid points cluster around the inlet, inlet cone and elbow sections of a draft tube where 70 $\sim$ 80% of mean pressure recovery occurs [16, 34] as well as near the boundary of a draft tube. To obtain grid resolution free results, grid sensitivity test was conducted. Three different meshes - low, intermediate and high resolved meshes- were created and mean pressure recovery factor was computed at each case. Then, we defined the mean pressure recovery factor disparity, which can be written as

$$\Delta C_{pm}(i) = \left| \frac{(C_{pm})_i - (C_{pm})_3}{(C_{pm})_3} \right| \times 100. \tag{5.14}$$

Here, $(C_{pm})_1$, $(C_{pm})_2$ and $(C_{pm})_3$ are respectively mean pressure recovery factors computed from simulation results with low, intermediate and high resolved meshes. The grid sensitivity test results are presented in Table 5.1. As the resolution of the mesh increases from low to intermediate, disparity of the mean pressure recovery factor is reduced approximately from 0.5% to 0.07%. These results imply the simulation results in terms of the pressure recovery factor is not sensitive to the grid resolution and the values of mean pressure recovery factor is converging the a certain value as the numbers of cells of meshes increase. To compromise



(a) (b)

Figure 5.10: Normalized pressure recovery factors at mid-plane. (a) Normalized pressure recovery factor along upper mid-plane wall (b) Normalized pressure recovery factor along lower mid-plane wall.

Mesh resolution	C_{pm}	Number of cells	$\Delta C_{pm}(\%)$
Low	0.8989	1870216	0.4981
Intermediate	0.9027	2879054	0.0675
High	0.9034	3717977	-

Table 5.1: Effects of resolution of the grids. As resolution of simulation increases, mean pressure recovery approaches a certain value.

the efficiency and accuracy of the simulation, we use the intermediate mesh for the further simulation.

In addition, we plotted normalized pressure recovery factor as a function of the distance along the upper centerline wall in Figure 5.10a and the lower centerline wall in Figure 5.10b. Then, our results are compared to the data from experiments and the results from other computation simulations as shown in Figure 5.10. Although the normalized pressure recovery factors from our simulation slightly more variate around sharp heel region compared to other simulation results, but our simulation results generally well agree with experimental data and other CFD simulation results. In fact, any simulation results have not been validated near the sharp heel region due to the absence of experimental data at that region.

5.5 Optimization

In the proposed method, the optimum shape of an object is found by determining the optimum weights of baseline objects, which maximizes the performance of the object. To find

design parameters	lower bound	upper bound
ω_0 (1)	0	1
ω_1 (0)	0	1
ω_2 (0)	0	1
ω_3 (0)	-0.5	0.5
ω_4 (0)	-0.5	0.5
ω_1^c (0)	-0.5	1.5
ω_2^c (0)	-0.5	1.5
ω_h^c (1)	0.5	1.5

Table 5.2: Eight design parameters with their ranges. The values in the parentesis indicate the weights for the original draft tube.

optimum weights, morphing method is incorporated with artificial neural network and genetic algorithm in the optimization process which basically consists of simulating, predicting and searching processes [49, 54].

Figure 5.11 shows the overall optimization procedure in design-by-morphing. First, simulation is implemented at the initial selected points and cost functions are computed at each selected point, called design of experiment. Then, artificial neural network constructs a surrogate model for the prediction of cost functions. Once the surrogate model is constructed, genetic algorithm searches optimum weights and then morphing algorithm creates an estimated optimum object with the found optimum weights. After that, simulation is implemented for the estimated optimum object and cost functions is obtained. Then, by adding newly computed cost functions to design of experiment, surrogate model is updated. This process is repeated until cost functions become sufficiently small or iteration reaches a specified maximum iteration number. We used this process to find the optimal shape of a turbine-99 draft tube to maximize mean pressure recovery factor. The optimal shape is found by seeking values of eight design parameters where five of them are the weights to obtain a morphed exterior shape of a draft tube and the rest three are the weights to obtain a morphed interior shape of a draft tube. The ranges of design parameters were defined as shown in Table 5.2.

Design of Experiment

The design of experiment is conducted to construct a surrogate model to predict a cost function as a function of input design variables. There are different method to perform DOE such as factorial, Plackett-Burmann and cubic face centered [50], but we used a discrete level method [24] generating user-defined sampling variables over the design space. To create as accurately approximated surrogate model as possible, many number of initially selected points is required. However, it causes huge amounts of simulation work. As a compromise, we selected initial 500 sampling points in the design space to build surrogate model. This

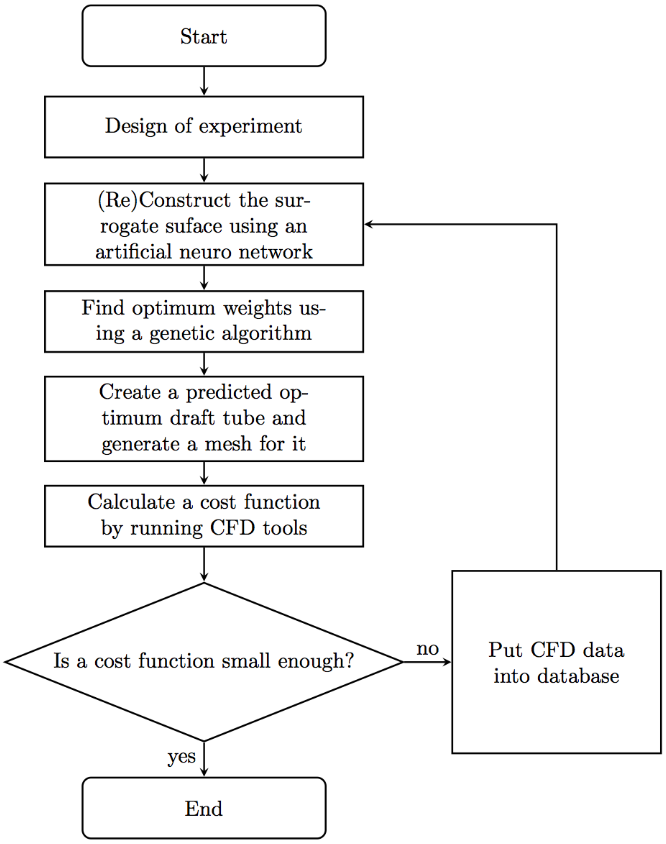

Start
Design of experiment
(Re)Construct the surrogate suface using an artificial neuro network
Find optimum weights using a genetic algorithm
Create a predicted optimum draft tube and generate a mesh for it
Calculate a cost function by running CFD tools
Is a cost function small enough?
no
Put CFD data into database
yes
End

Figure 5.11: Flow chart of the optimization process.

may not be enough number of sampling points to construct accurate surrogate model for eight design variables optimization problem because only two or three sample layers are distributed at each design variable. However, this is moderate number to deal with the eight dimensional optimization problem with limited computation time and resources.

Artificial Neural Network

An artificial neural network is utilized to estimate a cost function which is effected by input design parameters. An artificial neural network is built by several layers and each layer consists of a number of neurons. The numbers of input, hidden and output layers used in optimizing the shape of a turbine-99 draft tube are given in Table 5.2. The high quality of artificial neural network with these numbers of layers is validated in [13]. To investigate the accuracy of artificial neural network in our design problem, we defined the normalized error measuring discrepancy of mean pressure recovery factor obtained from CFD simulation and artificial neural network, given as

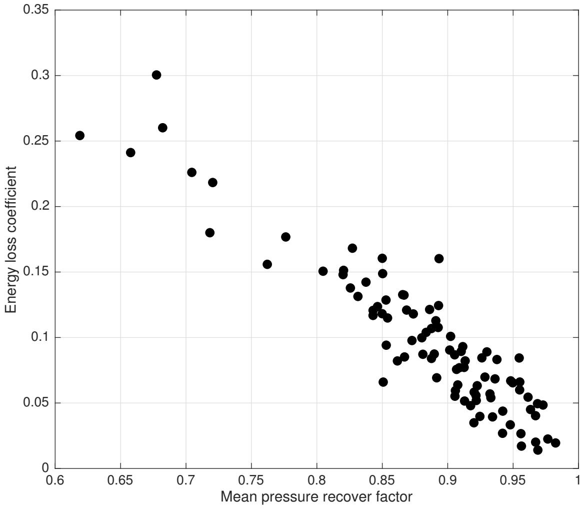


Figure 5.12: Correlation between mean pressure recovery factor and energy loss coefficient at the first several points among initially selectd 500 points. These two factors show highly negative correlation. The correlation coefficient of these two parameters is $C(C_{pm}, \zeta) = -0.9296$.

$$E = \sqrt{\frac{\sum_{j=1}^{J} \left(C_{pm,j} - C_{pm,j}^{ANN} \right)^2}{J}}, \tag{5.15}$$

where $C_{pm,j}$ and $C_{pm,j}^{ANN}$ are a j^{th} experimental mean pressure recovery factor computed from the CFD simulation and predicted from artificial neural network, respectively, and J is a number of initial selected points, in our case $J = 500$. The normalized error of the initial five hundred design of experiment is $E = 0.0072$. It indicates that artificial neural network is well trained by a design of experiment and is expected to predict mean pressure recovery factor with acceptable accuracy.

5.6 Results

As mentioned in subsection 5.4, mean pressure recovery factor and energy loss coefficient are significant quantities which should be considered to evaluate the performance of a turbine-99 draft tube. Although Pareto front method can be applied by defining double valued cost functions for both pressure recovery factor and energy loss coefficient, we only take account of maximizing pressure recovery factor based on the assumption that pressure recovery factor and energy loss coefficient are inversely correlated. As the verification of our assumption, we plotted mean pressure recovery factor and energy loss coefficient of the first several

Name	ω_0	ω_1	ω_2	ω_3	ω_4	ω_1^c	ω_2^c	ω_h^c	C_{pm}
baseline DT 0	1	0	0	0	0	0	0	1	0.903
baseline DT 1	0	1	0	0	0	0	0	1	0.945
baseline DT 2	0	0	1	0	0	0	0	1	0.965
baseline DT 3	0	0	0	1	0	0	0	1	0.922
baseline DT 4	0	0	0	0	1	0	0	1	0.903
baseline DT 5	0	0	0	0	0	0	0	1	0.904
best DT among ISP	0	0	1	0.5	-0.5	1.5	1.5	1.5	0.986
ODT	0	0.6	1	0.337	-0.407	-0.3	0.778	1.5	0.994
ODT with EPR	0.096	0.267	0.801	0.522	-0.500	0.572	0.719	2.015	1.001

Table 5.3: Weights, mean pressure recovery factor of the baseline, best draft tube among initial 500 selected points and the optimal draft tubes. Here, "DT", "ISP" and "ODT" stand for draft tube, initial selected points and the optimal draft tube, respectively.

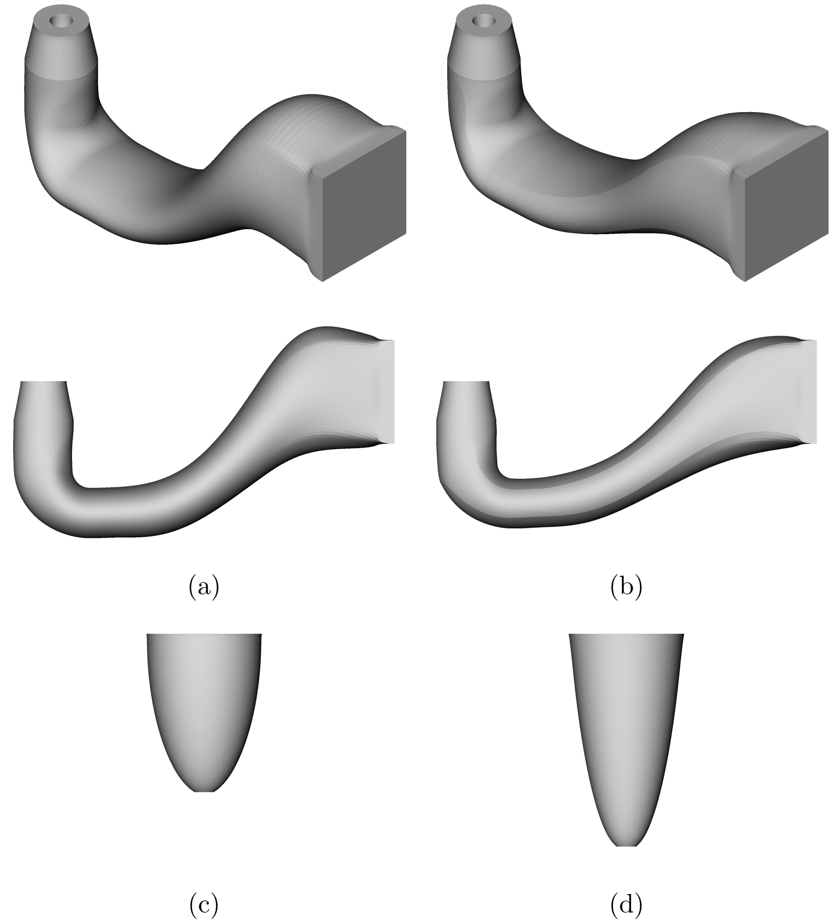
points of initially selected 500 points in Figure 5.12. If we define $C_{pm,j}$ and ζ_j as the mean pressure recovery factor and energy loss coefficient at the j^{th} design of experiment, correlation coefficient of mean pressure recovery factor and energy loss coefficient, $C(C_{pm}, \zeta)$, which measures linear dependency of them is given as

$$C(C_{pm}, \zeta) = \frac{1}{J-1} \sum_{j=1}^{J} \left(\frac{C_{pm,j} - \mu_1}{\sigma_1} \right) \left(\frac{\zeta_j - \mu_2}{\sigma_2} \right), \tag{5.16}$$

where J is number of points and μ_1 and σ_1 are the mean and standard deviation of mean pressure recovery factor of the several points among initially selected 500 points, respectively, and μ_2 and σ_2 are the mean and standard deviation of energy loss coefficient of them, respectively. As a result of computing the correlation coefficient, we obtained $C(C_{pm}, \zeta) = -0.9296$, which indicates that mean pressure recovery factor and energy loss coefficient are very negatively correlated. Therefore, by maximizing mean pressure recovery factor, we can expect to minimize energy loss coefficient as well, justifying our assumption.

Simulation and optimization results are shown in Table 5.3. In the table, weights, mean pressure recovery factors and energy loss coefficients of baseline draft tubes and the optimal draft tube are presented. Compared to the original draft tube, mean pressure recovery factor of the second baseline draft tube showing the highest mean pressure recovery factor among baseline draft tubes is 6.95% better. Other baseline draft tubes also performs better (at least similar) than the original draft tube in terms of mean pressure recovery factor. The fact that the original draft tube is worse than baseline draft tubes which are created without any preliminary studies and simulations of the flow but only based on physical intuition shows that high necessity of a turbine-99 draft tube's optimization.

Among 500 initial experiments, we found the best draft tube whose mean pressure recovery factor is improved 9.17% respect to mean pressure recovery factor of the original draft tube. After optimization was carried out, the optimum draft tube was found and mean



(a)

(b)

(c)

(d)

Figure 5.13: The shape of the optimal draft tubes. (a) the optimal draft tube without extended parameter ranges (b) the optimal draft tube with extended parameter ranges

pressure recovery factor was improved upto 10.11%. Since some of the optimum weights are negative, the optimum draft tube was created from extrapolation morphing. One factor should be noticed here is that some of the weights of the best draft tube among 500 initial experiments and the optimal draft tube are on the bounds of weights. Then, ones can imagine that better draft tubes may be obtainable by extending ranges of design parameters so that optimum weights are located inside (not on the boundaries of) design space. For this purpose, we defined new ranges of design parameters extended to the directions where the averaged values of each weights of the best draft tube among 500 initial experiments and the optimized draft tube are close to the their upper or lower bounds. We extended the ranges if the averaged weights deviate more than ±20% from the mean value of the ranges. The new defined design parameter ranges are shown in Figure 5.14.

With the newly defined ranges of design parameters, another 500 initial experiments were conducted. Then, we added these experiments to previously conducted 500 experiments with

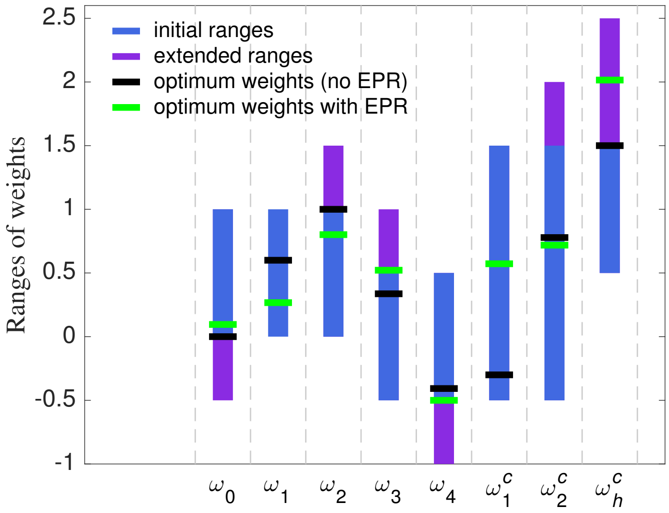
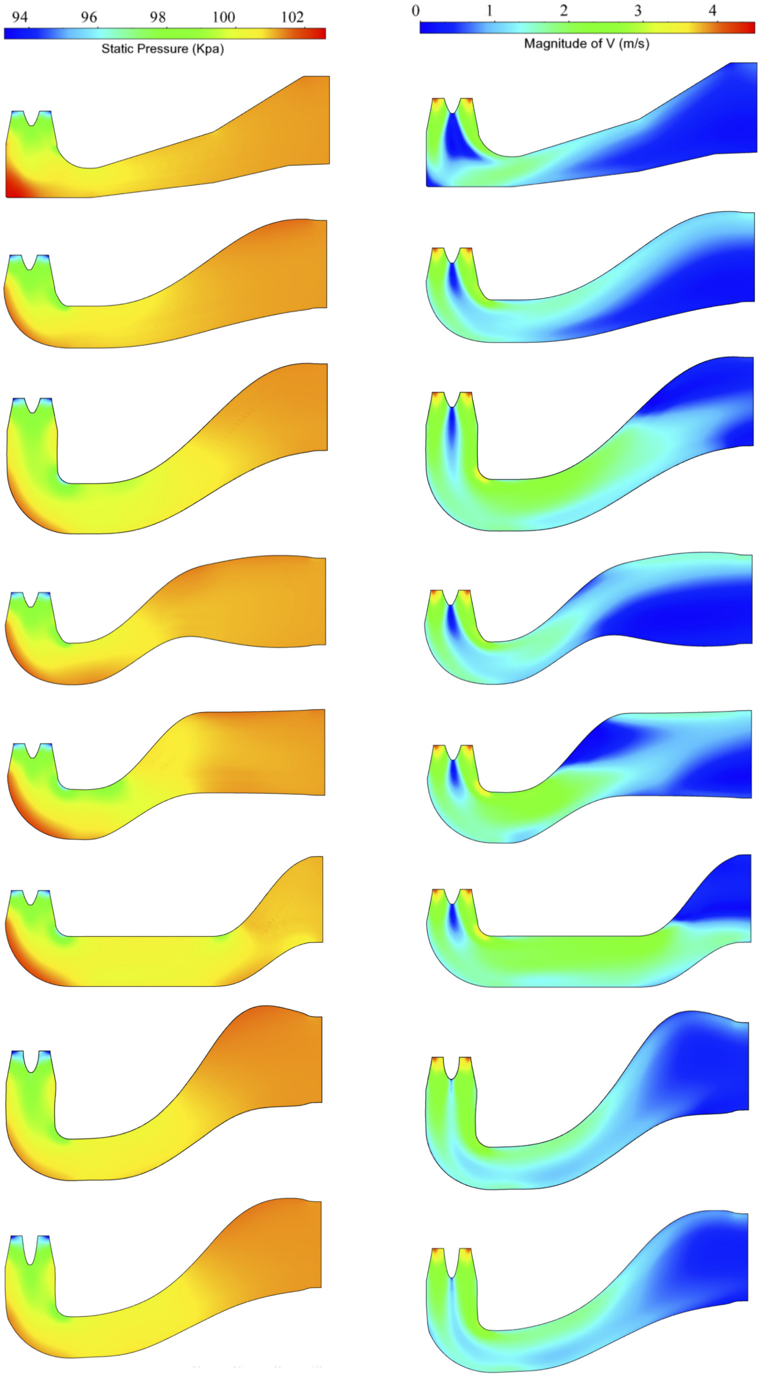


Figure 5.14: Ranges of design parameters. Here “EPR” means extended parameter ranges.

narrow ranges of design parameters, and so total experiments become a thousand. With the thousand experiments, we constructed the surrogate model using artificial neural network and carried out thirty optimized loops. As a result of the optimization, we achieved a new optimal draft tube which shows 10.92% improvement rate respect to the original draft tube. We will refer this optimal draft tube as the optimal draft tube with extended parameter ranges. Due to the extension of design space, approximately 0.8 percentage point more improvement was obtained.

The optimal draft tubes with and without extended parameters ranges are presented in Figure 5.13 with the shapes of their nose cones. As illustrated by Figure 5.13, configurations of the two optimized draft tubes are quite distinguishable. This configuration discrepancy results from the weights of these draft tubes which were converged to different values with respect to different ranges of design parameters. It indicates that several local maximums exist over the design space and the maximum value can be effected by defined ranges of design parameter. To obtain the global maximum independent on upper and lower bounds of design parameters, parameters ranges should be properly defined to include the weights producing the global maximum.

The static pressure, magnitude of velocity and dynamic pressure at the mid-plane of the original draft tube and the optimized draft tubes with and without extended parameter ranges are represented in Figure 5.15. The high static pressure region near a sharp heel of the former is apparently removed at the optimal draft tubes. Simply removing a sharp heel does not help in getting rid of the high pressure region. The static pressure contour carried out on a smoothed heel turbine-99 draft tube by Marjavaara and Lundström [34] does not show any high pressure region removal. Further, as shown in Figure 5.15, the fourth and fifth baseline drafts which do not have a sharp heel in the elbow section still have high static



(a)

(b)

Figure 5.15: Flow contour plots of several draft tubes at the mid-plane : (a) static pressure (b) magnitude of velocity. From the first row to sixth rows, contour plots show the flow of the zeroth to fifth baseline draft tubes, respectively. The contour plots shown in the first and second rows from the bottom are respectively from the optimal draft tube with and without the extended parameter ranges. The color maps of all static pressure and magnitude of velocity contour plots are identical.

pressure regions. Removing high pressure region and improving the performance of a draft tube result from all combination of the factors such as shape of the cross section, the depth of elbow and shape of diffuser not just smoothing sharp heel of a draft tube.

5.7 Discussion

We proposed a new method of shape optimization to find the optimal shape of an object and applied the new method in optimizing the shape of a turbine-99 draft tube to maximize a mean pressure recovery factor. Using our method, the surfaces of baseline draft tubes were spectrally represented and a morphed draft tube was created by combining baseline draft tubes with different weights. After optimization was carried out, we found the optimal draft tubes whose pressure recovery factors are increased more than 10% compared to the original turbine-99 draft tube. The flow field analysis showed that the high static pressure region of the original turbine-99 draft tube is significantly reduced in the optimal draft tubes, and the flow is smoother and more uniform than it was. We believe that this optimization method is applicable to many engineering applications in which the performance of an object depends on its shape.

Chapter 6

Conclusion

In design-by-morphing, we represent the surface of an object or sub-object as a truncated sum of spectral basis functions multiplied by spectral coefficients. An object or sub-object is classified by the number of its boundaries. Depending upon the number of boundaries of the object has, an object or sub-object belongs to a 0-, 1-, 2-hole object or a patch and the surface of the object is approximated by spherical harmonics, one-sided Jacobi polynomials and Fourier expansion, Chebyshev polynomials and Fourier expansion, and double Chebyshev polynomials, respectively. A morphed object is created from one or more baseline objects by finding new spectral coefficients that are weighted averages of the spectral coefficients of the baseline objects. An optimum shape of an object is created by finding weights of baseline objects which minimizes a cost function or maximizes a performance function.

The weights of baseline objects can be not only between zero to unity, which means a morphed object is created by interpolating given baseline objects, but the ranges of the weights can be also greater than unity and less than zero, which means a morphed object is created by extrapolating the baseline objects. With this range of weights, design-by-morphing is encouraged to explore an innovative new design shape, which may bring high performance improvement compared to other design methods.

Shape constraints are easily imposed on the boundaries of objects and sub-objects by solving a second order parabolic PDE with the boundary conditions which are defined by shape geometric constraints demanded to be imposed on the objects and sub-objects. By solving higher order PDEs, higher geometric constraints such as slope and curvature constraints can be also imposed on the boundaries of objects and sub-objects. When geometric constraints are imposed, boundary layer meaning the healing distance from the boundaries where the constraints are imposed to the location where the influence of the imposed constraints disappear are adjustable by the values of diffusitivies. Therefore, final shape of a constrained object is determined by the values of diffusitivies as well as shape, slope or curvature geometric constraints imposed on the object. A complex object can be also made up of many sub-objects whose geometries are constrained such that the overall object is smooth with a continuous shape, slope, and curvature. For a complex object, design-by-morphing works by morphing several baseline objects together sub-object by sub-object and choosing the weights of the sub-objects so that the morphed object minimizes a cost function or maximizes a performance function.

Our design-by-morphing is very flexible, and it can produce radical or conservative (or partially radical or partially conservative) shape changes by adjusting weights of objects and sub-objects. Further, morphing can occur part by part so that only one or a few parts can be replaced with morphed objects while the other parts are maintained. In addition, geometric constraints can be imposed easily on new designed shapes, and every optimization process is automated, making it an inexpensive way to implement optimum design.

We applied this optimization method to an airplane and to a turbine draft tube to maximize their performances. In both cases, the optimized shapes improved performance by more than 10%, whereas typical design improvements of the draft tube found by conventional methods are less than 1%. Design-by-morphing is a robust method and often produces optimized designs that are radically different from current designs. We believe that *design-by-morphing* can be used in a wide variety of engineering applications and achieve significant improvements of under-performed designs.

One limitation in design-by-morphing methodology is the fact that currently we are unable to morph two or more objects that are made up of sub-objects unless all of the objects have the same wiring diagram. Therefore, to morph two objects which have different wiring diagrams, the objects should be intentionally split the objects to make them have the same wiring diagram, but this process is quite tedious and need human intervention. One possible way to overcome this difficulty is inventing more general way of shape representation which is not effected by the structure of wiring diagrams. Besides of the wire diagram related issue, the question of how to break an object efficiently into sub-objects is difficult although there has been some recent progress in this area, such as the skeleton-based and convex-based segmentation methods[41, 42]. Our future work will focus on robust methods for breaking up an objects into sub-objects where those sub-objects are specifically n-hole constrained sub-objects and constrained patches.

Bibliography

[1] George R Anderson, Michael J Aftosmis, and Marian Nemec. Parametric deformation of discrete geometry for aerodynamic shape design. *AIAA Paper*, 965, 2012.

[2] GR Anderson, MJ Aftosmis, and M Nemec. Constraint-based shape parameterization for aerodynamic design. In *7th International Conference on Computational Fluid Dynamics, Big Island, Hawaii*, 2012.

[3] Urban Andersson. *An experimental study of the flow in a sharp-heel Kaplan draft tube.* Luleå tekniska universitet,, 2009.

[4] Michael Athanasopoulos, Hassan Ugail, and Gabriela González Castro. Parametric design of aircraft geometry using partial differential equations. *Advances in Engineering Software*, 40(7):479–486, 2009.

[5] Artemy Baxansky and Nahum Kiryati. Calculating geometric properties of three-dimensional objects from the spherical harmonic representation. *Pattern recognition*, 40 (2):756–770, 2007.

[6] Martin Bokeloh, Michael Wand, Hans-Peter Seidel, and Vladlen Koltun. An algebraic model for parameterized shape editing. *ACM Transactions on Graphics (TOG)*, 31(4): 78, 2012.

[7] John P Boyd. *Chebyshev and Fourier spectral methods.* Courier Corporation, 2001.

[8] Ch Brechbühler, Guido Gerig, and Olaf Kübler. Parametrization of closed surfaces for 3-d shape description. *Computer vision and image understanding*, 61(2):154–170, 1995.

[9] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics.* Scientific Computation. Springer Berlin Heidelberg, 2007. ISBN 9783540307280. URL `https://books.google.com/books?id=7COgEw5_EBQC`.

[10] Jin Chen, Quan Wang, Wen Zhong Shen, Xiaoping Pang, Songlin Li, and Xiaofeng Guo. Structural optimization study of composite wind turbine blade. *Materials & Design*, 46:247–255, 2013.

[11] JG Coelho. Numerical simulation of draft tube flow of a bulb turbine. *Journal homepage: www. IJEE. IEEFoundation. org*, 4(4):539–548, 2013.

[12] N Dahlbäck. Redesign of sharp heel draft tuberesults from tests in model and prototype. In *Hydraulic Machinery and Cavitation*, pages 985–993. Springer, 1996.

[13] Alain Demeulenaere, Alban Ligout, and Charles Hirsch. Application of multipoint optimization to the design of turbomachinery blades. In *ASME Turbo Expo 2004: Power for Land, Sea, and Air*, pages 1481–1489. American Society of Mechanical Engineers, 2004.

[14] Tamal K Dey, Pawas Ranjan, and Yusu Wang. Eigen deformation of 3d models. *The Visual Computer*, 28(6-8):585–595, 2012.

[15] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 652–663. ACM, 2004.

[16] Sergio Galván, Marcelo Reggio, and Francois Guibault. Assessment study of k- turbulence models and near-wall modeling for steady state swirling flow analysis in draft tube using fluent. *Engineering Applications of Computational Fluid Mechanics*, 5(4): 459–478, 2011.

[17] Pascal Gautron, Jaroslav Krivanek, Sumanta N Pattanaik, and Kadi Bouatouch. A novel hemispherical basis for accurate and efficient rendering. *Rendering Techniques*, 2004:321–330, 2004.

[18] Rikard Gebart, Håkan Gustavsson, and Rolf Karlsson. Proceedings of turbine-99 workshop on draft tube flow. 2000.

[19] Aurobrata Ghosh, Elias Tsigaridas, Bernard Mourrain, and Rachid Deriche. A polynomial approach for extracting the extrema of a spherical function and its application in diffusion mri. *Medical image analysis*, 17(5):503–514, 2013.

[20] Tushar Goel, Rajkumar Vaidyanathan, Raphael T Haftka, Wei Shyy, Nestor V Queipo, and Kevin Tucker. Response surface approximation of pareto optimal front in multi-objective optimization. *Computer methods in applied mechanics and engineering*, 196 (4):879–893, 2007.

[21] David Gottlieb and Steven A Orszag. *Numerical analysis of spectral methods: theory and applications*, volume 26. Siam, 1977.

[22] Maksimillian Fedorovich Gubin. *Draft tubes of hydro-electric stations.* Published for the Bureau of Reclamation, US Dept. of the Interior and National Science Foundation, Washington, DC by Amerind Pub. Co., 1973.

[23] Thomas JR Hughes, John A Cottrell, and Yuri Bazilevs. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, 194(39):4135–4195, 2005.

[24] NUMECA International. User manual fine-design3d v9.1. *NUMECA International*.

[25] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun. A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics (TOG)*, 31(4):55, 2012.

[26] J.Y. Kang. Design of marine structures through morphing method and its supporting techniques. *Marine Technology Society Journal*, 48(2):81–89, 2014. doi: 10.4031/MTSJ.48.2.7. cited By 0.

[27] Seongkyu Lee. Inverse design of horizontal axis wind turbine blades using a vortex line method. *Wind Energy*, 18(2):253–266, 2015.

[28] Bruno Lévy. Laplace-beltrami eigenfunctions towards an algorithm that" understands" geometry. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pages 13–13. IEEE, 2006.

[29] CC Long, AL Marsden, and Y Bazilevs. Shape optimization of pulsatile ventricular assist devices using fsi to minimize thrombotic risk. *Computational Mechanics*, 54(4): 921–932, 2014.

[30] Zhoujie Lyu, Gaetan KW Kenway, and Joaquim RRA Martins. Aerodynamic shape optimization investigations of the common research model wing benchmark. *AIAA Journal*, 53(4):968–985, 2014.

[31] VA Marchenko. Sturm-liouville operators and their applications. *Kiev Izdatel Naukova Dumka*, 1, 1977.

[32] Philip Marcus. Description and philosophy of spectral methods. In Karl-Heinz Winkler and Michael Norman, editors, *Astrophysical Radiation Hydrodynamics, NATO ASI Series*, volume 188, pages 359–386. Springer, 1986. doi: 10.1007/978-94-009-4754-2˙10.

[33] B Daniel Marjavaara, T Staffan Lundström, Tushar Goel, Yolanda Mack, and Wei Shyy. Hydraulic turbine diffuser shape optimization by multiple surrogate model approximations of pareto fronts. *Journal of fluids engineering*, 129(9):1228–1240, 2007.

[34] BD Marjavaara and TS Lundström. Redesign of a sharp heel draft tube by a validated cfd-optimization. *International journal for numerical methods in fluids*, 50(8):911–924, 2006.

[35] Daniel Marjavaara and Staffan Lundström. Response surface-based shape optimization of a francis draft tube. *International Journal of Numerical Methods for Heat & Fluid Flow*, 17(1):34–45, 2007.

[36] T Matsushima and PS Marcus. A spectral method for polar coordinates. *Journal of Computational Physics*, 120(2):365–374, 1995.

[37] AP McCabe. Constrained optimization of the shape of a wave energy collector by genetic algorithm. *Renewable energy*, 51:274–284, 2013.

[38] AP McCabe, GA Aggidis, and MB Widden. Optimizing the shape of a surge-and-pitch wave energy collector using a genetic algorithm. *Renewable Energy*, 35(12):2767–2775, 2010.

[39] Stefan Oerlemans, Murray Fisher, Thierry Maeder, and Klaus Kögler. Reduction of wind turbine noise using optimized airfoils and trailing-edge serrations. *AIAA journal*, 47(6):1470–1481, 2009.

[40] J Poole, CB Allen, and TCS Rendall. Application of control point-based aerodynamic shape optimization to two-dimensional drag minimization. In *52nd AIAA Aerospace Sciences Meeting, National Harbor, Maryland*, pages 2014–0413, 2014.

[41] Zhou Ren, Junsong Yuan, Chunyuan Li, and Wenyu Liu. Minimum near-convex decomposition for robust shape representation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 303–310. IEEE, 2011.

[42] Dennie Reniers, Jarke J Van Wijk, and Alexandru Telea. Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. *Visualization and Computer Graphics, IEEE Transactions on*, 14(2):355–368, 2008.

[43] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. Laplace–beltrami spectra as shape-dnaof surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.

[44] Martin Reuter, Silvia Biasotti, Daniela Giorgi, Giuseppe Patanè, and Michela Spagnuolo. Discrete laplace–beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3):381–390, 2009.

[45] David W Ritchie and Graham JL Kemp. Fast computation, rotation, and comparison of low resolution spherical harmonic molecular surfaces. *Journal of Computational Chemistry*, 20(4):383–395, 1999.

[46] Nathanaël Schaeffer. Efficient spherical harmonic transforms aimed at pseudospectral numerical simulations. *Geochemistry, Geophysics, Geosystems*, 14(3):751–758, 2013.

[47] Adriana Schulz, Ariel Shamir, David IW Levin, Pitchaya Sitthi-Amorn, and Wojciech Matusik. Design and fabrication by example. *ACM Transactions on Graphics (TOG)*, 33(4):62, 2014.

[48] Mohammad Hasan Shojaeefard, Ammar Mirzaei, and Ali Babaei. Shape optimization of draft tubes for agnew microhydro turbines. *Energy Conversion and Management*, 79: 681–689, 2014.

[49] Wei Shyy, Nilay Papila, Rajkumar Vaidyanathan, and Kevin Tucker. Global design optimization for aerodynamics and rocket propulsion components. *Progress in Aerospace Sciences*, 37:59 – 118, 2001.

[50] Wei Shyy, Nilay Papila, Rajkumar Vaidyanathan, and Kevin Tucker. Global design optimization for aerodynamics and rocket propulsion components. *Progress in Aerospace Sciences*, 37(1):59–118, 2001.

[51] Hassan Ugail, Malcolm IG Bloor, and Michael J Wilson. Manipulation of pde surfaces using an interactively defined parameterisation. *Computers & Graphics*, 23(4):525–534, 1999.

[52] Robert VanBuskirk and Philip Marcus. Spectrally accurate contour dynamics. *Journal of Computational Physics*, 115(2):302–318, 1994.

[53] WTM Verkley. A spectral model for two-dimensional incompressible fluid flow in a circular basin. *Journal of Computational Physics*, 136(1):100–114, 1997.

[54] XD Wang, C Hirsch, Sh Kang, and C Lacor. Multi-objective optimization of turbomachinery using improved nsga-ii and approximation model. *Computer Methods in Applied Mechanics and Engineering*, 200(9):883–895, 2011.

[55] Emily L Williams, Ayman El-Baz, Matthew Nitzken, Andrew E Switala, and Manuel F Casanova. Spherical harmonic analysis of cortical complexity in autism and dyslexia. *Translational neuroscience*, 3(1):36–40, 2012.

[56] Rachel Aine Yotter, Robert Dahnke, Paul M Thompson, and Christian Gaser. Topological correction of brain surface meshes using spherical harmonics. *Human brain mapping*, 32(7):1109–1124, 2011.

[57] LH You, Hassan Ugail, and Jian J Zhang. Controllable c1 continuous blending of time-dependent parametric surfaces. *The Visual Computer*, 28(6-8):573–583, 2012.

[58] Lihua You, Jian J Zhang, and Peter Comninos. Blending surface generation using a fast and accurate analytical solution of a fourth-order pde with three shape control parameters. *The Visual Computer*, 20(2-3):199–214, 2004.

[59] Mehmet Ersin Yumer and Levent Burak Kara. Co-constrained handles for deformation in shape collections. *ACM Transactions on Graphics (TOG)*, 33(6):187, 2014.

[60] Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K. Hodgins, and Levent Burak Kara. Semantic shape editing using deformation handles. *ACM Trans. Graph.*, 34(4):86:1–86:12, July 2015. ISSN 0730-0301. doi: 10.1145/2766908. URL `http://doi.acm.org/10.1145/2766908`.