UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA AERONÁUTICA Y DEL ESPACIO

Geometric manipulation of NURBS surfaces for computational meshes

Doctoral Thesis

by

Daniel Redondo García

MADRID, JULY 2018

UNIVERSIDAD POLITÉCNICA DE MADRID
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA AERONÁUTICA Y DEL ESPACIO

Geometric manipulation of NURBS surfaces for computational meshes

Doctoral Thesis

by

Daniel Redondo García

MADRID, JULY 2018

*This thesis is dedicated to my parents
whose endless support made it possible for me
to pursue my scientific curiosity.*

Abstract

This work tackles the problem of the geometric design optimization from an innovative point of view. The developed methodology considers directly the parameters which define the geometry as design variables. State of the art techniques use deformation functions which hamper the full process automation and pose an extra difficulty with intersecting and trimming geometries. This new technique eliminates all these obstacles by construction, and has an even more interesting feature: it provides the final optimized geometry, avoiding the costly work of searching the geometry that better fits the cloud of points from the optimized mesh.

The potential increase in the problem dimensionality has been counteracted by a smart reduction of the design variables, following the inclusion of the weights of the control points into the set of variable to be optimized. This exercise exposes the relevance of a geometric variable which has been disregarded in most of the existing design processes.

Finally, as a complement to the previous developments, the capability to impose analytic boundary constraints between untrimmed NURBS panels was implemented.

Resumen

Este trabajo aborda el problema de la optimización del diseño apoyado en simulaciones computacionales desde un punto de vista novedoso. La metodología desarrollada utiliza directamente los propios parámetros que definen la geometría como variables de diseño, evitando la necesidad de introducir funciones de deformación que, por un lado dificultan la automatización del proceso y, por otro, presentan dificultades aún no resueltas en el manejo de intersecciones entre superficies. El resultado final es la geometría ya optimizada, lo que permite eludir el trabajo manual que supone la búsqueda de la geometría que mejor se adapta a la nube de puntos que conforma la malla superficial ya optimizada.

El posible incremento en la dimensionalidad del problema se ha resuelto mediante una reducción inteligente de las variables de diseño, tras introducir también los pesos de los puntos de control de la NURBS en el conjunto de variables a optimizar, poniendo de manifiesto la relevancia de una variable geométrica menospreciada hasta el momento en la mayor parte de los procesos de diseño.

Por último, y como complemento a todo lo anterior, se ha introducido la capacidad de imponer condiciones de contorno analíticas en paneles NURBS no trimados.

Acknowledgements

First of all, I would like to express my deep gratitude to my advisers Prof. Marta Cordero and Prof. Mariola Gómez for their continuous support, patience and recommendations along all these years.

I will always be in debt with Juan Ramón Pérez Garraleta, he introduced me to the NURBS world, and he showed me the magic behind the most simple solutions.

I also want to address my gratitude to the many colleagues I met working in different research activities during the last years. Eusebio, Esther, Geza, Heiko and Jacques, you illustrated me how to fight for each project; you made me feel part of something bigger while working with you.

Special acknowledgment is made to Carlos, Maria Chiara and Ramón, who were always ready for interesting discussions that led to a real friendship.

I am specially thankful for all the good friends who always helped me to disconnect and to enjoy life. Raúl, Patricia, Elena, Roberto, Willy, Laura, Mamen, Borja, Emilio, Bea, Ivan and Asel, I do not know what I had done without you, but I would not have had such wonderful experiences.

My deepest thanks go to my parents and brother. I would have never been able to accomplish this long journey without their continuous support and encouragement. No matter the difficulties, they never stopped believing in me. Their love and example will always be present in my life.

Contents

1	Introduction	1
1.1	A historical evolution	2
1.1.1	From artisan skills to the the current industrial techniques: today's challenges	3
1.2	Problem definition	6
1.3	Structure of the work	7
2	Geometry	9
2.1	Geometrical description	10
2.2	Geometry parameterization	11
2.2.1	Parametric and implicit form	11
2.2.2	Bézier geometries	12
2.2.3	Rational Bézier geometries	13
2.2.4	B-Spline	14
2.2.5	Rational B-spline	15
2.2.6	Non-uniform rational B-spline (NURBS)	15
2.2.7	T-splines	16
3	Simulation framework	17
3.1	Industrial aerodynamic design process	18
3.2	Optimization	19
3.2.1	Optimization variables	20
3.2.2	Adjoint optimization	21
3.2.3	Adjoint solver	25
4	NURBS	29
4.1	NURBS definition	30
4.1.1	NURBS curve	30
4.1.2	Knotvector	31

4.1.3	Convex hull property	31
4.1.4	NURBS surface	33
4.2	NURBS analysis	35
4.2.1	Continuity definition	35
4.2.2	NURBS curvature: curves	35
4.2.3	NURBS conventions: weights	37
4.2.4	NURBS: getting an uniform knotvector	37
4.2.5	NURBS continuity analysis	38
4.2.6	NURBS analysis: continuity through control points or weights . . .	38
5	Point inversion	61
5.1	Objective	62
5.2	Problem definition	63
5.3	Problem solving	63
5.4	Point inversion techniques: description	66
5.4.1	Grid search	66
5.4.2	“Quad” search	67
5.4.3	“Quadratic” approximation search	68
5.4.4	First order Hu-Wallner algorithm	70
5.4.5	Second order Hu-Wallner algorithm	71
5.5	Point inversion techniques: evaluation	73
5.5.1	Grid search	75
5.5.2	“Quad” search	78
5.5.3	“Quadratic” search	79
5.5.4	Hu-Wallner 1 st order search	81
5.5.5	Hu-Wallner 2 nd order search	82
6	Dimensionality reduction	85
6.1	Analyzing dimensionality	86
6.1.1	Testing setup: curve fitting	86
6.1.2	Increasing the problem dimensionality	88
6.2	NURBS weight in the optimization	92
6.2.1	The current role of the weight of the control points in a NURBS geometry	92
6.2.2	Recovering the role of the weighted control points	93
6.2.3	Using weight values to eliminate control points	97

6.3	Analytical constraints can further reduce the dimensionality	99
6.3.1	Imposing $\mathcal{C}^0$ continuity with a fixed geometry	99
6.3.2	Imposing $\mathcal{C}^1$ continuity with a fixed geometry	101
A	Appendix NURBS expressions	109
A.1	1 st order, 3 control points, knots=[0,0,1,2,2]	110
A.2	2 nd order, 4 control points, knots=[0,0,0,1,2,2,2]	111
A.3	2 nd order, 5 control points, knots=[0,0,0,1,2,3,3,3]	113
A.4	2 nd order, 5 control points, knots=[0,0,0,1,1,2,2,2]	115
A.5	2 nd order, 6 control points, knots=[0,0,0,1,2,3,4,4,4]	117
A.6	2 nd order, 6 control points, knots=[0,0,0,1,1,2,3,3,3]	119
A.7	2 nd order, 7 control points, knots=[0,0,0,1,2,3,4,5,5,5]	122
A.8	2 nd order, 7 control points, knots=[0,0,0,1,1,2,3,4,4,4]	125
A.9	2 nd order, 7 control points, knots=[0,0,0,1,2,2,3,4,4,4]	127
A.10	2 nd order, 7 control points, knots=[0,0,0,1,1,2,2,3,3,3]	129
A.11	3 rd order, 5 control points, knots=[0,0,0,0,1,2,2,2,2]	131
A.12	3 rd order, 6 control points, knots=[0,0,0,0,1,2,3,3,3,3]	133
A.13	3 rd order, 6 control points, knots=[0,0,0,0,1,1,2,2,2,2]	135
A.14	3 rd order, 7 control points, knots=[0,0,0,0,1,2,3,4,4,4,4]	137
A.15	3 rd order, 7 control points, knots=[0,0,0,0,1,1,2,3,3,3,3]	140
A.16	3 rd order, 7 control points, knots=[0,0,0,0,1,1,1,2,2,2,2]	142
A.17	3 rd order, 8 control points, knots=[0,0,0,0,1,2,3,4,5,5,5,5]	144
A.18	3 rd order, 8 control points, knots=[0,0,0,0,1,1,2,3,4,4,4,4]	147
A.19	3 rd order, 8 control points, knots=[0,0,0,0,1,1,1,2,3,3,3,3]	150
A.20	3 rd order, 8 control points, knots=[0,0,0,0,1,1,2,2,3,3,3,3]	152
A.21	3 rd order, 8 control points, knots=[0,0,0,0,1,2,2,3,4,4,4,4]	155

Introduction

Contents

1.1	A historical evolution	2
1.1.1	From artisan skills to the the current industrial techniques: to-day's challenges	3
1.2	Problem definition	6
1.3	Structure of the work	7

This study is aimed to improve the manipulation of shape designs with a special focus on the industrial application of an adjoint optimization approach. It covers all the aspects from geometry parameterization and problem dimensionality, to the tight relation between geometries and computational meshes and the possibility to apply specific design constraints on the geometry parameterization.

This chapter starts with a short review of the current state of the art and its evolution. It will help us to introduce the subject and to better identify the main issues that appear within the existing technology. This brief presentation, will settle the foundations that leads to choose the solutions developed in this work, in opposition to other possibilities.

1.1 A historical evolution

In ancient times, continuous improvement of human manufactured products relied mostly on experience and artisan intuition; and only at the very last stage, the advance was corroborated by the product manufacturing and its in-service life.

As science progressed, this pure empirical methodology was broadened with a growing knowledge of the physical problem and its possible solutions. Then, first simple simulation models appeared and allowed prediction up to certain extent of the product behavior before completely building the final product.

Later on, and in parallel with this progress, industrial revolutions came along with its needs to accurately define the product in order to ensure its repeatability as a cornerstone of the industrial process.

The concept of **parametric definition** appears as a result of all these evolutions: on the one hand, it allows parametric studies which are aimed at searching the optimum among a family of potential solutions, and on the other hand, it ensures repeatability, because the set of parameters contains the complete information to identify our chosen design from the complete family of available designs.

Computational simulation and computer aided design (CAD) carried on a qualitative step change; mathematical models were able to be simulated with a computer, and their size and complexity increased continuously. At this point, the computational formats employed to define the geometry won the battle of the geometry definition and they became standards, thanks to their ability to manipulate more and more complex geometries with a similar effort level. Computational simulation seamlessly approached the real model, and automation of manufacturing processes began using the same for-

mats as the ones employed in the design phase with the computer aided manufacturing (CAM). All these improvements led to considerable costs savings by limiting the number of physical models to be tested in a laboratory, reduction in their development cycles and the possibility of performing assessments of innovative configurations through low cost simulations.

However, the computational simulation has its own limitations. The continuous search for an increasing complexity in the models and the precision and accuracy of the simulations with respect to the real behavior have to be paid with high computational cost: in a daily basis, industry employs simulations that require hundreds of processors to work continuously along several days to reach the requested accuracy level in a certain problem. The specific case of the fluid mechanical problem is nowadays a big challenge where the full simulation of the lowest scale turbulence is still –and probably for many years to come– completely unaffordable.

1.1.1 From artisan skills to the the current industrial techniques: today’s challenges

Old manufacturing techniques have evolved long time ago into increasingly automated and computer-controlled processes; but all those not so new methodologies require solid foundations in order to ensure the continuation and improvement of the existing technology level with a complete new approach.

Naval industry may easily illustrate this change. In its former times, the ships manufacturing process was the origin of the denominated “*lofting*” methodology: the hull shape was defined originally with the help of flexible wooden strips or canes that were deformed with the help of weights and pulleys, forcing them to pass through a certain number of fixed points or clamping them to adapt to a certain tangency. In this example, $\mathcal{C}^1$, $\mathcal{C}^2$ or even $\mathcal{C}^3$ continuity was ensured, provided that the strip had been made in a single piece, and that the required loads were applied with a specific distribution. However, even in those classical processes, further continuity orders or $\mathcal{C}^n$ could not always be achieved and the continuity degree depended on the kind of loads applied on the strip (according to the classical Timoshenko or Euler-Bernoulli beam theory).

Nowadays, the introduction of CAD systems has brought great improvements in the industrial design process: required time for the design phase has been drastically reduced, computational simulations have become the mainstay of the validation and the new CAD standards have broaden their portability and the range of influence of the geometrical design. Nevertheless, when the same classical needs have to be imposed in a not so simple

1.1 A historical evolution

geometry, a strong mathematical background must also be employed in order to ensure that obtained results fulfill all those requirements which could be directly achieved with previous methodologies by the technique construction itself.

In a world where computational simulation rises up as the most promising alternative to the expensive experimental *trial and error* methodology, current design –and simulation– methodology is still a complex and costly process that consumes a significant portion of the total component design and manufacturing costs. This design phase usually implies an iterative process in which several designs and geometries are modified to fulfill a set of restrictions, while simultaneously optimizing one or more objective functions (weight, cost, aerodynamic efficiency,...). This process is commonly performed through successive evaluations of constraints and objective functions. Thanks to the simulation capabilities, part of the experimental testing of the new model can be avoided and thus, the process is shortened while minimizing the experimental cost it would have to face otherwise.

Several geometrical models arose for filling this need of a computational geometry definition request: Bézier, B-splines, rational B-Splines, NURBS, T-NURBS, in increasing complexity, among others.[†]

However, the real description of the geometry is frequently not employed for these simulations, and in its place a discrete model of the geometry with a finite number of points that conform the computational mesh is used for the computations. Examples of this methodology can be found in most CFD simulations [53], and as mentioned by Zhang and Cen [71], “*features of mesh generation and remeshing in the standard CFD software are not usually implemented, [...] communication of flow information between new and old mesh [...] is processed by the interpolation schemes [...] this processing will bring calculation errors and decrease accuracy*”.

The simulation-based optimization strategy that has been classically employed in the industrial designs commonly requests the computation of the gradients of the objective function with respect to each of the design variables. The search of the optimum is performed with successive steps along the descent direction of the objective function which is given by the gradient, which is generally calculated with finite differences. Unfortunately, if the simulation has a high computational cost, this process is doomed due to the fact that each of the gradient computations requires a number of simulations that is

[†]This study will focus on NURBS geometries, mainly because of their wide standardization within the industrial CAD tools.

linked to the number of variables of the system; in such way that the number of variables becomes a limiting factor in the feasibility of the gradient-based optimization problem.

Recent developments in the adjoint methodology have eliminated the limitation imposed by the number of variables, because adjoint methodology allows obtaining the gradient by solving only the direct and the adjoint problems which are, both of them, independent on the number of variables [44, 47]; thus, it could be considered that the classical interest in having a reduced number of variables has vanished and that the designer can freely use more and more complex geometries defined by larger and larger sets of variables; but the reality is extremely different.

The computational simulations are performed on a computational mesh, which is obtained after a costly process from the analytical definition of the geometry. This process has a strongly unidirectional nature and it cannot be reversed; given a certain computational mesh, the search of the initial set of geometric variables that were used to generate this mesh is not straightforward. This lack of bidirectionality leads to some difficulties in the iterative optimization process: the changes in the computational mesh cannot be directly translated to the analytical geometry, which is the required output from the design process.

The increasing computational power has made possible more and more complex simulations, but despite the easily growing number of variables that can be involved in the definition of a geometry, such a spurious high number may easily create numerical noise. In a general case, the variables used for defining a geometry may not be completely independent, or they could constitute a set of variables covering a family of geometries that is much more complex than the one of interest for the specific problem.

Considering this risk of over-complexity in the geometry description, it can be deduced that the old discussion of the geometry description and its parameterization is not completely over. It has recently recovered its importance by the hand of the optimization and the emergence of the iso-geometric analysis. As highlighted by Lovadina [30], “the isogeometric analysis is a recent emerging technology for scientific computing, stemming from old ideas”. With an origin from structural applications [19], this methodology joins geometric design and analysis through the analytical definition of the geometry. Its application to the Navier-Stokes problem [17, 41] is relatively recent, but mesh generation bottlenecks and other issues related to the approximation of the geometry with a given mesh would vanish with this methodology that uses the basis function of the NURBS geometry to perform the analysis.

1.2 Problem definition

Industry continuously considers the need to design smooth shapes: lofting, filleting and unions between different surfaces. This smoothness becomes specifically critical in aeronautical applications where aerodynamic requirements may limit the allowed curvature and impose strong smoothness constraints in order to delay laminar-turbulent transition. Fortunately, despite all these requirements and limitations, usual designs have many degrees of freedom that allow the designer to modify the shape and fulfill different requirements like:

- Cover a system with a surface that minimizes aerodynamic drag.
- Change local aerodynamic behavior.
- Deflect the air flow towards an inlet.

In order to put this study into a practical context, it can be anticipated a situation where a component of an aircraft need to be redesigned to minimize aerodynamic drag, but none of the surrounding components can be modified. An example of this application could be the redesign of a leading edge or nacelle lip, trying to increase the natural laminar region in the complete wing or nacelle respectively. The analysis of this problem shows that, each of the existing sections of the component will be probably defined by several tenths of control points, and such a high dimensionality opens the door to geometries with an undesired level of waviness on the surface.

This growing number of variables that can be involved in the definition of a geometry lead to a high dimensionality that may easily create numerical noise due to the non-orthogonality of the variables employed for defining the geometry; moreover, they could constitute a set of variables covering a family of geometries that is much more complex than the one of interest for the specific problem.

This work tackles the geometry optimization problem from the industrial perspective with an emphasis on its geometric aspects; covering the required preprocessing which is needed in order to optimize the real geometry parameters with an adjoint optimization process and some techniques to apply smart reduction of the design space and impose boundary constraints.

1.3 Structure of the work

Throughout this study, the required missing bricks will be developed in order to complete all the tools needed to squeeze the full potential of the geometry adjoint optimization.

It started with a summary of the problems from the industrial point of view, and it continues with a review of the different geometry approaches that can be considered. It is followed by a description of the optimization process and the advantage of using adjoint methodologies whenever we face optimization problems with an elevated number of variables and with high computational cost for each simulation.

Once the optimization framework has been defined, a detailed analysis of the geometry description will be performed, in order to properly understand the potential of the NURBS and their limitations.

The adjoint optimization methodology applied to the pure geometry definition requires a preliminary preprocessing which is called *point inversion*. Several point inversion techniques will be evaluated in order to build a process that ensures the required reliability to support the adjoint optimization.

The last part of the study is dedicated to the smart dimensionality reduction. The risks derived from high dimensionality problems will be shown and an innovative methodology to progressively reduce this high dimensionality will be presented. This technique uses the adjoint information collected in the optimization to recover the importance of the weights in the NURBS geometries, and jointly with the imposition of analytical boundary conditions, they will allow an astute reduction of the design space which will strongly contribute to remove some spurious solutions.

2

Geometry

Contents

2.1	Geometrical description	10
2.2	Geometry parameterization	11
2.2.1	Parametric and implicit form	11
2.2.2	Bézier geometries	12
2.2.3	Rational Bézier geometries	13
2.2.4	B-Spline	14
2.2.5	Rational B-spline	15
2.2.6	Non-uniform rational B-spline (NURBS)	15
2.2.7	T-splines	16

2.1 Geometrical description

Throughout the last decades, the most relevant progress in computational geometry did not originate from the industrial design but from the entertainment industry in its wide sense: from the NURBS (Non-Uniform Rational B-spline) supporting film animation industry to the tessellation required for the video-games. This tendency came along with an enormous improvement in hardware acceleration, ray tracing, algorithm parallelization, etc; but in exchange other aspects like accuracy or smoothness have been compromised.

Industrial design software market is driven by a few commercial tools (CATIA, ICEM Surf, SolidEdge, etc.) widely used in a multipurpose environment, and some very specific tools with application into a single industry or even only within the company who developed it. While the first subset of tools stand out for their versatility and easiness of use, their main drawback is the limited user control over the analytic definition of the underlying geometry; in contrast to the second subset which thanks to their simplicity give the user a better control and knowledge about the mathematical definition of the geometry and its limitations.

Although there are several methodologies to describe a geometry, NURBS still constitutes the standard of curve and surface description in computer aided design (CAD) systems and the most employed by industrial software.

One of the typical requirements of the implementation of any industrial tool is its ability to obtain some geometrical properties as a function of a few parameters. Aerodynamic designs have a special request regarding the geometrical derivatives of the new geometry, due to their high influence in the aerodynamic coefficients.

Looking into detail what is expected to be able to be imposed in our design, the main requirements can be summarized as following:

- Geometry through a set of lower dimension geometries, *i.e.*
 - points for curves (curves passing through points)
 - points and curves for surfaces (surfaces passing through points and curves)
- First and second order continuity at the junction between two geometries.
 - Continuity imposed in the boundary.
 - Continuity imposed internally for trimmed surfaces.
- Monotony of the new geometry curvature along the flow direction.

A brief analysis of the evolution that made NURBS the current standard in industrial design will help to get a global view about the problem and open issues and show stoppers that may hamper full exploitation of the potential of the technology.

2.2 Geometry parameterization

In this section, the geometry definition evolution will be followed, from the simplest and commonly well known approaches, to the current state of the art.

In most cases, the starting point will be a single curve, only because its expression is a bit simpler. The evolution to surfaces is in general an straightforward adaptation from the curve expressions.

Although not explicitly indicated, a three-dimensional space will be considered along the whole document, where the employed coordinate system is the Cartesian one with coordinates $\{x, y, z\}$.

2.2.1 Parametric and implicit form

Parametric form: The geometry expression is split in three independent expressions, one for each spatial coordinate. The geometry $\mathbf{G}$ can be expressed as:

$$\mathbf{G}(\mathbf{p}) = \begin{cases} x = x(\mathbf{p}) \\ y = y(\mathbf{p}) \\ z = z(\mathbf{p}) \end{cases} \quad (2.1)$$

where $\mathbf{p}$ is a vector with as many dimensions as free geometry parameters (0, 1 or 2 for points, curves and surfaces respectively).

This approach seems to be the best parameterization for mathematical manipulation and is often used in differential geometry. Even so, as frequently studied in the academical examples, parameterization may exhibit anomalies that must be treated carefully. This issue requires special attention depending on each case, and so, it may hinder its use as a computational standard.

Implicit form: The geometry $\mathbf{G}$ is defined as the solution of a system of equations with as many equations as dimension restrictions to the geometry, *i.e.* 1, 2 or 3 for surfaces, curves and points, respectively. The geometry dimension is the space

2.2 Geometry parameterization

dimension minus the dimension restrictions.

$$\mathbf{G} = \{g_1, \dots, g_i\} = \begin{cases} g_1(x, y, z) = 0 \\ \vdots \\ g_i(x, y, z) = 0 \end{cases} \quad (2.2)$$

This implicit form is mathematically equivalent to the parametric one after some parameters elimination. However it is not useful because of its natural difficulty to evaluate. *A system of equations needs to be solved for each evaluation!*

The implicit form will not be well suited for bounded geometries, because there is no way to easily limit the solution; but the parametric definition may solve the problem by setting boundaries or range of variation in the parameters space.

Unfortunately, the manipulation of the parametric geometry descriptions is not always simple. Translations can be easily applied to the parametric form, but rotations and affinity transformations must be stored jointly with the geometry definition, so that it could be applied whenever the new geometry is required. This kind of historic transformation storage is not acceptable for computational purposes; the geometry should be defined with a common format, that is independent on the number and kind of transformation we have needed to obtain it.

At this point, it is important to consider that **polynomial geometries** constitute a subset of the geometries defined parametrically. The easiness of obtaining derivatives in a polynomial would be helpful for solving the issues presented in §1.2. However, it is also known that polynomials cannot be used for representing all geometries that could be defined in parametric form (*e.g.* circumferences). This happens because polynomial geometries are only a subset of the parametric geometries.

2.2.2 Bézier geometries

Popularized by Pierre Étienne Bézier for the automobile industry, but created by Paul de Casteljaou in 1959, Bézier curves are polynomial functions whose coefficients are obtained from a set of control points called *control polygon*.

The generalized expression of a Bézier curve with degree n is:

$$\mathbf{C}(t) = \sum_{i=0}^n \binom{n}{i} \mathbf{P}_i \cdot (1-t)^{n-i} \cdot t^i, \quad t \in [0, 1] \quad (2.3)$$

where $\mathbf{P}_i$ is a set of $n + 1$ control points that define the geometry.

This definition of Bézier curves can be extended to surfaces of order (n, m) that are defined by a set of $(n + 1) \times (m + 1)$ control points $\mathbf{P}_{i,j}$ as:

$$\mathbf{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) \cdot B_{j,m}(v) \cdot \mathbf{P}_{i,j} \quad (2.4)$$

where $B_{i,n}(u)$ are the Bernstein polynomial.

$$B_{i,n}(u) = \binom{n}{i} \cdot u^i \cdot (1 - u)^{n-i} \quad (2.5)$$

Bézier geometries solve the problem derived from simple transformations (rotations, translations and affine transformations) because their definition structure is invariant, and the transformations need to be applied only to the control polygon.

This control polygon approximates the shape of the geometry in such way that the geometry is contained in the convex hull of the control polygon.

These two properties are the biggest advantages that made Bézier curves extensively used graphic design software nowadays. They constitute the grounds of the PostScript format and the computer typography.

Bézier geometries are derivable within their domain, and the tangents at their boundaries are defined by the boundary segments of the control polygon.

The first drawback derives from the nature of the power basis functions which makes that any modification in the control polygon extend up to the whole geometry, *i.e.* local modifications of the shape cannot be achieved only with displacement of the closest control points.

And finally, the last and most important drawback is that certain important geometries like circles, ellipses, hyperbolas, cylinders, cones or spheres cannot be accurately represented by Bézier geometries due to the polynomial structure of their definition.

2.2.3 Rational Bézier geometries

Polynomials can be used to describe a parabola, but not any of the remaining conic shapes. It is known from projective geometry, that any conic curve can be seen under the same structure and properties; the only difference is that *homogeneous coordinates* need to be considered.

$$(x, y, z, \omega) \longleftrightarrow \left(\frac{x}{\omega}, \frac{y}{\omega}, \frac{z}{\omega} \right)$$

With these homogeneous coordinates (x, y, z, ω) , and making them equal to the classical ones (x', y', z') , the expression yields to $(x', y', z') = \left(\frac{x}{\omega}, \frac{y}{\omega}, \frac{z}{\omega} \right)$ and $\mathbf{P} = \omega \cdot \mathbf{P}'$.

2.2 Geometry parameterization

The resulting geometry can be written as:

$$\mathbf{C}(u) = \frac{\sum_{i=0}^n B_{i,n}(u) \cdot \omega_i \cdot \mathbf{P}_i}{\sum_{i=0}^n B_{i,n}(u) \cdot \omega_i}, \quad u \in [0, 1] \quad (2.6)$$

where ω_i is the weight in each control point $\mathbf{P}_i$.

In agreement with projective geometry, all the weights can be multiplied by the same factor with no influence on the final shape, thanks to the denominator, which is a non-negative polynomial with the only objective of normalizing the effect of the weights.

These weights overcome the problem of representing conic curves, including the circle, because they allow the geometry to be defined as one polynomial divided by another [12].

However, there are still some problems. The number of control points is linked to the order of the geometry ($n + 1$ control points for a n -order curve), and if we need to increase the control on the curve shape, the order increases accordingly and this fact leads to numerical instabilities and computational inefficiency. That is, due to the local control, the curve cannot be imposed to pass through a certain number of fixed points without breaking the Bézier geometry into smaller ones or increasing the order, *e.g.*, $(n - 1)$ -degree is needed to pass a Bézier curve through n data points.

2.2.4 B-Spline

Considering the solution of breaking the geometry into smaller parts, a *piecewise polynomial* definition can be obtained. This is the definition of a B-Spline curve.

$$\mathbf{C}(u) = \sum_{i=0}^n f_i(u) \cdot \mathbf{P}_i \quad (2.7)$$

where $f_i(u)$, $i = 0, \dots, n$ are the *B-spline basis functions* (piecewise polynomial functions) with a fixed non-decreasing uniform breakpoint sequence $U = \{u_k\}$, $k = 0, \dots, m$, that is called *knotvector*.

These B-spline basis functions and knotvector determine the degree and continuity of the geometry, hence, the control points affect only the shape of the geometry, but not the continuity.

In order to achieve the *local support* property, *i.e.* only local influence of the control points, it must be ensured that each basis function $f_i(u)$ is nonzero only in a limited number of subintervals, not along the entire domain defined by the knotvector $[u_0, u_m]$,

while maintaining the polynomial shape, so the basis functions $f_i(u)$ are defined as the B-spline basis functions $N_{i,p}(u)$:

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{any other} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} \cdot N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} \cdot N_{i+1,p-1}(u) \end{aligned} \quad (2.8)$$

A deeper study about B-spline basis functions [45] (from now on *basis functions*) shows how the basis functions delimit the influence of the control points on the geometry.

Once the local support issue has been addressed while keeping all the benefits from the Bézier geometries (convex hull, diminishing property, polynomial shape) there is another subject that need to be solved, as explained in §2.2.2, there are geometries that cannot be represented with a B-spline due to its polynomial structure.

2.2.5 Rational B-spline

Following the same procedure that took us from the Bézier geometries to the rational Bézier, a rational B-spline can be obtained, which in case of curves takes the form:

$$\mathbf{C}(\xi) = \frac{\sum_{i=0}^n N_{i,p}(\xi) \omega_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^n N_{i,p}(\xi) \omega_{i,j}} \quad ; \quad 0 \leq \xi \leq 1 \quad (2.9)$$

where $N_{i,p}$ are the *basis functions* as explained in 2.2.4 with $U = \{u_k\}$ as uniform *knotvector*.

2.2.6 Non-uniform rational B-spline (NURBS)

NURBS are obtained after a small generalization of the knotvector, that now includes the possibility of non-uniform monotone knots. The expression that defines NURBS geometries is exactly the same as the one employed for rational B-splines.

So that a NURBS surface is be defined as:

$$\mathbf{S}(\xi, \eta) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(\xi) N_{j,q}(\eta) \omega_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(\xi) N_{j,q}(\eta) \omega_{i,j}} \quad ; \quad 0 \leq \xi, \eta \leq 1 \quad (2.10)$$

2.2 Geometry parameterization

In fact there is no clear difference between rational B-spline and NURBS. The “*non-uniform*” only emphasizes the property that the knotvector does not need to be uniform. The only condition it must accomplish is that knot values cannot decrease (*growing monotony*). This is the reason because most of commercial design software uses B-splines by default even if they are perfectly able to manipulate NURBS.

This uniformity rupture contribute to the reduction of the computational effort by allowing better shape control and definition of a larger class of shapes that the achievable ones from the uniform knotvector used in B-splines.

A complete chapter §4 has been dedicated to this geometry parameterization due to its relevance to the purpose of this study.

2.2.7 T-splines

Even if it seems that there is no need for further improvements, the computational effort is always a good driving force for new approaches, and this is the main reason for the appearance of T-NURBS.

It is always desirable to have a reduced number of surfaces in the design so that inner continuity is maintained automatically. However, in order to include local details on a surface, new control points may need to be added in the area of interest. But NURBS surfaces are linked to a sort of structured grid of control points. This set of control points must lie in a topologically rectangular grid. The problem arises with different scales in a single geometry that force the grid of control points to include refinement traversing the entire surface, even if it is only required in an specific region of the grid.

In 2003 Sederberg published a new approach [60]. By making use of T-junctions, the rectangular topological grid limitation in the NURBS was avoided. This new approach has not been implemented extensively into the commercial software yet, but first results are quite promising.

3

Simulation framework

Contents

3.1	Industrial aerodynamic design process	18
3.2	Optimization	19
3.2.1	Optimization variables	20
3.2.2	Adjoint optimization	21
3.2.3	Adjoint solver	25

3.1 Industrial aerodynamic design process

As already highlighted in §1.1.1, the design phase usually implies an iterative process where every design is tested with a simulation tool in order to check the fitness of the designed component with respect to its main requirements.

The following in-deep review of the complete design process will help to better understand its limitations.

The starting point of the design task is always a preliminary geometry, frequently in CATIA format. But one of the main limiting factors of the design process is linked to the essence of the process itself. Computational simulation models are, in general, not capable to directly handle the real description of the geometry, and in its place, they operate with a discretization of the geometry, so that physical values are calculated on a set of points that conforms the computational mesh, which is constructed from a set of points placed on the design geometry.

The complexity of this initial geometry (number of surface panels, gaps,...) makes the process of obtaining the computational mesh not so straight forward, which is usually translated into a required cleaning and simplification of the geometry before it could be employed as input into the mesh generation module.

Despite the automation improvements in the mesh generation process, the strong dependance of the accuracy of the simulation results with respect to the mesh quality (specially in disciplines like aerodynamics) forces the designer to take special care of the mesh generation process in order to efficiently capture the most relevant characteristics of the flow. In such a way, including a man-in-the-loop hampers the efficiency of any optimization process that contains a remeshing stage in the optimization loop which cannot be fully automatic.

This mesh generation process has a strongly unidirectional nature and it cannot be easily reversed. Given a computational mesh, the search of the initial set of variables that were used to generate this mesh is not straightforward. This lack of bidirectional property leads to some difficulties in the iterative optimization process, because the changes in the computational mesh cannot be directly translated to the analytical geometry, which is the required output from the shape design process.

Solving the computational fluid dynamics simulation (direct solver) permits the evaluation of the objective function for each combination of the design variables. Although it is usually a costly process in terms of time and computational power, it can be considered quite automatic. Unfortunately, the existing automation in the simulation process is not

paired with similar levels in the mesh generation stage. This is the main reason because most employed optimization processes rely on a certain deformation function that acts directly on the mesh.

3.2 Optimization

Complete exploration of design space is usually impossible due to its huge computational cost, and an intelligent search of the minimum or maximum of the objective function through different combinations of the design parameters is required. Simple models with a few parameters are generally manageable with designer skills and experience, so that expert judgment is commonly the best choice after an initial parameters exploration; but as the number of parameters grows, the designer loses the spatial perception and his ability to visualize multidimensional –more than 3-dimensional– parameters space. It is in this context where automatic optimization and design space exploration techniques gain importance.

The most intuitive optimization methodology starts from the uniform sampling distribution in the design space, followed by finite differences from the most promising point. The application of finite differences with respect to all the optimizing variables yields the gradient after $n + 1$ evaluations of the function, being n the number of design variables. The information provided by the obtained gradient allows the construction of a vector in the design space with the local direction of the maximum variation of the objective function; thus, a new set of variables obtained after modifying the initial set with the maximum variation direction should result in a new geometry which, *a priori*, should be better than the former one Fig.3.1.

In the case of high computational cost simulations, this finite differences methodology is doomed due to the fact that each gradient computation requires a number of evaluations of the simulation which is linked to the number of variables of the system; in such a way that the number of variables become a limiting factor in the feasibility of our problem.

Recent developments in adjoint methodology have eliminated the limitation imposed by the number of variables, because adjoint optimization allows the computation of the gradient by solving only the direct and the adjoint problems which are –both of them– independent on the number of variables; thus, it could be considered that the classical interest in having a reduced number of variables will vanish and that more and more complex geometries defined by larger and larger sets of variables can be freely considered;

3.2 Optimization

Figure 3.1: Classical optimization process with finite differences

but the reality is extremely different due to several other constraints: waviness of the final surfaces, lack of convergence and numerical noise due to the higher dimensionality, problems to obtain the optimum geometry from the optimum computational mesh, etc.

3.2.1 Optimization variables

The first critical aspect in the optimization process is the description of the optimization variables. There are three main approaches for aerodynamic –direct and adjoint– optimization problems, depending on the chosen design variables:

Geometry definition parameters Not commonly employed for complex geometries due to the difficulty of the automatic mesh generation. Its main application area is in problems with structured meshes that can be easily parameterized as a function of the geometry; however these meshes usually require a much more intensive manual preprocessing.

Surface grid points coordinates It is only applied in the adjoint methodology, because it results in an explosion in the number of variables.

Deformation functions parameters The deformation functions appear as a methodology to reduce the huge number of variables that define the computational mesh (three coordinates per surface node), and it is currently the most common approach. The optimization parameters are disconnected from the geometry or mesh description, and they act as an intermediate step to deform the employed computational

mesh. There are two relevant examples of deformation functions in the aerodynamic industrial applications [39]: bumps functions and free form deformation (FFD).

Free form deformation (FFD) A control box is defined in the volume surrounding the geometry to be optimized and the variables to optimize are the coordinates of the control points of recently generated box. In a general case the box does not need to be a hexahedron with control points only on its vertex, but the set of grid points of a structured mesh in the hexahedron. The movement of those points is later transferred to the rest of the computational mesh by a predefined influence function. The main interest is to define influence functions that are fully differentiable so that we can introduce these derivatives into the adjoint optimization problem [21, 52, 36].

Bumps or Hicks-Henne functions These functions [15, 66, 24] actuate directly on the surface like a cubic sinus bump. They became popular by its simplicity and the possibility to apply several of them to model small or moderate perturbations of an initial geometry shape. The main interest of these functions is the easy implementation of their derivatives for adjoint problems.

It can be said that in absence of a bidirectional link between the mesh and the geometry, deformation functions appeared as a methodology to avoid the optimization of the huge number of variables that define the computational mesh (2 or 3 coordinates per surface node); being free form deformation (FFD) and bumps or Hicks-Henne functions the two major representatives in the aerodynamic industrial applications [39].

This study develops the first option, whose main advantage is its ability to deal directly with the design geometry, removing the need of complex intermediate steps to transform the optimum mesh into an optimum geometry.

3.2.2 Adjoint optimization

As the design complexity increases and the number of variables grows, obtaining the gradient with finite differences becomes harder. This is due to the noteworthy increase in the required computational power for solving $n+1$ problems in order to obtain the descent direction in a problem with n variables.

Adjoint methodology appears to reduce the cost of this optimization [44, 47]. It allows obtaining the gradient of the objective function after solving only the direct and the adjoint problem, instead of the $n+1$ direct problems that characterize the finite differences.

3.2 Optimization

Fig.3.2 shows the complete adjoint optimization process. The advantages with respect to the classical approach with the gradient obtained with finite differences become obvious after comparison with Fig.3.1

Figure 3.2: Adjoint optimization process

The adjoint optimization process as depicted in Fig.3.2 is composed by:

Initial geometry It should be as simple as possible in order to reduce the dimensionality of the problem and any constraint should be imposed wherever possible in their analytical form in order to further reduce the number of variables §6.[†]

Meshing tool It could be more or less automatic, but the output should always be a compatible format with the CFD solver. As a practical example for this study, the standard format will be NetCDF [67], where all the surface points will be identified with a *boundary marker*[‡] which will be later employed by the CFD solver to impose boundary conditions.

Preprocessing It receives the initial geometry and the surface mesh (or the complete mesh with the boundary markers) and it established a link between each surface

[†]It will be considered that the geometry is provided in NURBS format §4.1 along this study.

[‡]At least a different boundary marker should be employed for each of the surfaces that need to be optimized

mesh node and the corresponding point in the geometrical surface. This methodology will be extensively explained in §5

Solver or direct solver It resolves the fluid dynamics equations (Euler, Navier-Stokes, etc.) in each node of the provided computational mesh.

Adjoint solver Linked to the direct solver it obtains the adjoint variables and, in its continuous form, it provides the sensibility of the objective function with respect to normal displacements of the boundary conditions.[†]

Surface sensibility It computes the movement of each surface mesh node after a modification of the design variables (weights and coordinates of the control points of the geometry).

Optimizer It constitutes the backbone of the complete process. The optimization algorithm receives information from the geometrical surface sensitivity and the boundary sensitivity generated by the adjoint solver, and with all this information it constructs the variation vector of the design variables and it defines the stopping criteria.

Surface mesh deformation tool It receives the new definition of the geometry given by the design variables, and regenerates the new surface mesh (lying on a different geometry) with unchanged topology[‡].

Volume mesh deformation tool It receives as an input the new coordinates of each surface mesh node that delimits the flow field and it modifies accordingly the internal mesh nodes to adapt to the new geometry, while maintaining its topology. It is, in general, an available tool provided within the CFD package.

A detailed analysis of these tools shows that the output from the continuous adjoint solver (as depicted in Fig.3.3) does not fit to the actual needs, because the objective function sensitivity has been obtained in relation with the surface mesh points and not with the real geometrical design variables.

Moreover, this example in Fig.3.3 depicts the first stage of an optimization process where the descent direction $\frac{\partial J}{\partial \mathbf{X}}$ (movement of the nodes in the surface mesh) has been

[†]Many adjoint solvers are still under development and current versions may still require some simplifications like frozen turbulence among others.

[‡]Topology must be maintained in order to keep compatibility with the later volume mesh deformation. Special care must be taken in cases with intersections between surfaces, but in order to maintain mesh topology, the deformation of the surfaces should not give as a result a geometry with a different topology (*small modification approach*)

3.2 Optimization

clearly defined by the gradients. In this case, quick changes of the sensibility are mostly related to the physical model, but as optimization progresses towards minimum C_d , the absolute values of the sensitivity are progressively reduced and, after discarding some singular points (like the trailing edge), small perturbations tend to become predominant and they result in the undesired waviness already tackled by Jameson [22] with the gradient smoothing. The proposed solution will take a different strategy, and instead of filtering the gradient, it will “filter” the accessible design space.

Figure 3.3: C_d surface sensitivity of a NACA0012

Applying the chain rule, to the sensitivity of the surface mesh points we can obtain the real sensitivity of the design parameters:

$$\begin{aligned} \text{Continuous adjoint} &\Rightarrow \frac{\partial J}{\partial \vec{X}} \\ \text{Optimizador} &\Rightarrow \frac{\partial J}{\partial \vec{P}} = \frac{\partial J}{\partial \vec{X}} \cdot \frac{\partial \vec{X}}{\partial \vec{P}} \end{aligned}$$

Where J is the objective function to be optimized, $\vec{X}$ is the displacement vector of the surface control points along the local normal of the surface, and $\vec{P}$ is a vector with each design variable λ , that commonly identifies with the weights and coordinates of each control point defining the surface.

In order to obtain $\frac{\partial \vec{X}}{\partial \vec{P}}$, a relation between the computational mesh and the geometry definition is needed. But as already explained, the mesh generation stage is a complex process that cannot be easily reverted; thus a special pre-processing is needed. This pre-processing aims to pair each surface mesh point with its correspondent point on the surface geometry that was used for constructing the mesh. In an ideal world, all the surface mesh points perfectly lie on the geometrical surface, but due to the accuracy of the mesh generation process, it cannot be assured *a priori*. The identification of the closest point on the surface for each surface mesh node is denominated **point inversion**

and it can be interpreted as the problem of *identifying the minimum distance between a surface and a point*.

Once the point inversion of each surface mesh node has been solved, and the link between the geometry definition and the surface mesh has been established, the sensibility of the surface mesh nodes with respect to the design parameters (parameters defining the geometry, $\frac{\partial \vec{X}}{\partial \vec{P}}$) can be obtained straightforwardly with finite differences with respect to the design variables (it is an extremely quick calculation that only involves two evaluations of the surface).

With the sensibility of the surface mesh nodes with respect to the design parameters $\frac{\partial \vec{X}}{\partial \vec{P}}$ and the output from the adjoint solver $\frac{\partial F}{\partial \vec{X}}$, the sensibility of the objective function with respect to the design variables $\frac{\partial F}{\partial \vec{P}}$ can be obtained. Now, the just obtained descent direction can be entered as an standard gradient in the optimizer, so that the optimizer defines the progression in each optimization step along the descent direction in order to ensure proper convergence.

3.2.3 Adjoint solver

A short description of the adjoint methodology will make possible anticipate its needs when applied to industrial problems as intended in this study.[†]

Let us consider the need to determine the value of

$$\frac{dJ}{d\lambda} = g^T \cdot u + B(X, \lambda) \quad (3.1)$$

where J is the cost or objective function, λ is the design variable or perturbation variable, g is a vector and $u = \frac{dU}{d\lambda}$ is defined as the sensibility. In turn, u value is obtained after resolving the system $A \cdot u = f$, where f is a function of λ .

The real problem lies in $f = f(X, \lambda)$. For each of the n design variables (or perturbations) is needed to solve the system $A \cdot u = f$, to obtain the u value and in such a way we can evaluate the sensibility $\frac{dJ}{d\lambda}$.

Adjoint methodology formulates the dual problem which, *grosso modo*, can be expressed as following:

Obtaining the value of the scalar product $g^T \cdot u$, where u is the solution of $A \cdot u = f$, is equivalent to calculate the scalar product $v^T \cdot f$, where v is the solution of $A^T \cdot v = g$; but the essential difference is that g does not depend on λ .

[†]In order to simplify the understanding of the adjoint optimization, no physical meaning will be taken into consideration but only the algebra involved in the adjoint methodology.

3.2 Optimization

Using the properties of the matrix algebra, a short demonstration can be easily followed:

$$\left. \begin{array}{l} A \cdot u = f \quad \rightarrow \quad u = A^{-1} \cdot f \\ A^T \cdot v = g \quad \rightarrow \quad g^T = v^T \cdot A \end{array} \right\} \quad g^T \cdot u = v^T \cdot A \cdot A^{-1} \cdot f = v^T \cdot f \quad (3.2)$$

For evaluating $\frac{dU}{d\lambda}$ for $f = f_1, f_2, \dots, f_n$, there are different options:

FD	Finite differences (Direct)	$\triangleright$	Solve n times $A \cdot u = f$
DUAL	Control theory (Adjoint)	$\triangleright$	Solve once $A^T \cdot v = g$

If this methodology is applied to the aerodynamic problem, the process is composed by the following steps:

1. **Definition of the fluid mechanics equations (Euler, Navier-Stokes,...)** Being X the geometry of the problem and $U = \{\rho, \rho u, \rho v, \rho w, \rho E\}$ the employed conservative variables, the flux equations can be defined as:

$$N(U, X) = 0 \quad (3.3)$$

2. **Flux variation after geometric modifications –of the surface–** After applying the chain rule to the flux equations, and identifying λ with the perturbation, it is obtained:

$$\frac{\partial N}{\partial U} \cdot \frac{dU}{d\lambda} + \frac{\partial N}{\partial X} \cdot \frac{dX}{d\lambda} = 0 \quad (3.4)$$

$$\frac{\partial N}{\partial U} \cdot \frac{dU}{d\lambda} = -\frac{\partial N}{\partial X} \cdot \frac{dX}{d\lambda} \Rightarrow A \cdot u = f \quad (3.5)$$

Where $A = \frac{\partial N}{\partial U}$, $f = -\frac{\partial N}{\partial X} \cdot \frac{dX}{d\lambda}$ y $u = \frac{dU}{d\lambda}$ is commonly denominated sensibility of the solution with respect to the design variables.

3. **Minimization of the objective function** $J = J(U, X)$ which will be in general a non-linear function of the flow field variables and the aerodynamic coefficients.

$$\frac{dJ}{d\lambda} = \frac{\partial J}{\partial U} \cdot \frac{dU}{d\lambda} + \frac{\partial J}{\partial X} \cdot \frac{dX}{d\lambda} = g^T \cdot u + \frac{\partial J}{\partial X} \cdot \frac{dX}{d\lambda} \quad (3.6)$$

At this stage, it must be noticed that calculating the product $g^T \cdot u$ is extremely costly, due to the fact that its evaluation requires solving for each design variable, the system $A \cdot u = f$ because f is a function of the perturbation.

4. **Setting up the adjoint problem or dual formulation** in order to solve the product $g^T \cdot u$ for as many values of f without any penalty in computation time.

$$A^T \cdot v = g \Rightarrow g^T \cdot u = v^T \cdot f \quad (3.7)$$

Where it can be observed that

$$\frac{dJ}{d\lambda} = v^T \cdot f + \frac{\partial J}{\partial X} \cdot \frac{dX}{d\lambda} \quad (3.8)$$

If the classical finite differences approach was employed, we would need to calculate:

$$\frac{dJ}{d\lambda} = g^T \cdot u + \frac{\partial J}{\partial X} \cdot \frac{dX}{d\lambda} \Rightarrow A \cdot u = f ; A = \frac{\partial N}{\partial U} ; f = -\frac{\partial N}{\partial X} \cdot \frac{dX}{d\lambda} \quad (3.9)$$

However, the dual formulation with origin in the control theory results in:

$$\frac{dJ}{d\lambda} = v^T \cdot f + \frac{\partial J}{\partial X} \cdot \frac{dX}{d\lambda} \Rightarrow A^T \cdot v = g ; A = \frac{\partial N}{\partial U} ; g = -\frac{\partial J}{\partial U} \quad (3.10)$$

Significative advantages appear when the optimization problem involves many design variables (λ):

FD	$f = -\frac{\partial N}{\partial X} \cdot \frac{dX}{d\lambda} \Rightarrow$	It depends on the design variables and it must be evaluated for each variable. $A \cdot u = f$
DUAL	$g^T = \frac{\partial J}{\partial \lambda} \Rightarrow$	It does not depend on the design variables and it must be evaluated only once. $A^T \cdot v = g$

4

NURBS

Contents

4.1	NURBS definition	30
4.1.1	NURBS curve	30
4.1.2	Knotvector	31
4.1.2.1	Normalized knotvector	31
4.1.2.2	Clamped and unclamped knotvector	31
4.1.3	Convex hull property	31
4.1.3.1	Weight points	32
4.1.4	NURBS surface	33
4.2	NURBS analysis	35
4.2.1	Continuity definition	35
4.2.2	NURBS curvature: curves	35
4.2.3	NURBS conventions: weights	37
4.2.4	NURBS: getting an uniform knotvector	37
4.2.5	NURBS continuity analysis	38
4.2.6	NURBS analysis: continuity through control points or weights	38
4.2.6.1	1 st order, 3 control points, knots=[0,0,1,2,2]	39
4.2.6.2	2 nd order, 4 control points, knots=[0,0,0,1,2,2,2]	41
4.2.6.3	2 nd order, 5 control points, knots=[0,0,0,1,2,3,3,3]	46
4.2.6.4	2 nd order, 5 control points, knots=[0,0,0,1,1,2,2,2]	48
4.2.6.5	2 nd order, 6 control points, knots=[0,0,0,1,2,3,4,4,4]	50
4.2.6.6	2 nd order, 6 control points, knots=[0,0,0,1,1,2,3,3,3]	52
4.2.6.7	2 nd order, 7 control points, knots=[0,0,0,1,2,3,4,5,5,5]	55
4.2.6.8	2 nd order, 7 control points, knots=[0,0,0,1,1,2,2,3,3,3]	58

4.1 NURBS definition

There are many choices for geometry definition and a short review was included in §2.2, but NURBS stand out among the others for most of the industrial designs. Due to the predominance of the NURBS, a more detailed analysis will be required in order to properly understand their strengths and limitations.

4.1 NURBS definition

Due to the predominance of the NURBS in the industrial design, this geometry description will be employed along this study.

Although there is a large bibliography about NURBS, a brief review of their principles will help to settle some concepts that will be required afterwards.

4.1.1 NURBS curve

The expression that defines a NURBS curve is:

$$\mathbf{C}(\xi) = \frac{\sum_{i=0}^n N_{i,p}(\xi) \omega_i \mathbf{P}_i}{\sum_{i=0}^n N_{i,p}(\xi) \omega_i} \quad ; \quad 0 \leq \xi \leq 1 \quad (4.1)$$

where:

- $\mathbf{C} = (x, y, z)$ are the coordinates of all the points on the NURBS curve, defined through the parameter $\xi \in [0, 1]^\dagger$,
- $\mathbf{P}_i = (x_i, y_i, z_i)$ are the set of $(n + 1)$ control points. This set of control points is called **control polygon**.
- ω_i are the associated weights to each control point.
- $N_{i,j}(u)$ are the B-Spline base functions,

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{any other} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} \cdot N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} \cdot N_{i+1,p-1}(u) \end{aligned} \quad (4.2)$$

that are linked to the **knotvector** $U = \{u_0, \dots, u_k, \dots, u_r\}$ (§4.1.2).

[†]This range implies the usage of a normalized knotvector §4.1.2.1

4.1.2 Knotvector

The knotvector is a weakly monotone growing set of values that defines the break-points in the NURBS equation and thus the range of influence of each control point through the basis functions.

Its monotonicity and the number of breaking points are the only constraint that must be strictly imposed by definition to the knot values.

$$U = \{u_0, \dots, u_k, \dots, u_r\} \mid u_i \leq u_{i+1} \quad (4.3)$$

The number of knots is given by $r = n + p + 1$, where $n + 1$ is the number of control points (classically numbered from 0 to n) and p is the grade of the NURBS curve.

4.1.2.1 Normalized knotvector

A knotvector $U = \{u_0, \dots, u_k, \dots, u_r\}$ is normalized if $u_0 = 0$ and $u_r = 1$.

The knotvector only indicates the advance of the NURBS parameter. Start and end of the knotvector define the bounding values of this parameter. Hence, any knotvector can be rescaled or translated with no influence on the final shape or the parameterization characteristics.

4.1.2.2 Clamped and unclamped knotvector

A knotvector is clamped if both extreme values have their multiplicities equal to the order of the NURBS $(p + 1)$.[†]

Tangency to the first and last legs of the control polygon can be imposed by clamping, that is, by setting the first and last knot with multiplicity equal to $p + 1$.

An unclamped knotvector will generate curves not touching the extremes of the control polygon.

4.1.3 Convex hull property

Convex set, or **convex hull of a point set** is the portion of space in which any pair of points belonging to the convex set can be connected with a straight line fully contained in the convex set.

[†]Important remark:

grade = p

order = $p + 1$

4.1 NURBS definition

Convex hull property: *Any convex combination of points (where the coefficients for each point, in addition to summing to one are also non-negative) lies always inside the convex hull of the point set.*[12, p. 48]

NURBS curves or surfaces have been obtained as a composition of different B-splines curves or surfaces. Thus, by definition, any point in the NURBS will inherit the properties from the B-spline employed to define it. It means that any NURBS maintains the *convex hull property*; or in other words, any point in a NURBS lies inside the convex hull of the control points that define the B-spline it belongs to.[12, p. 228][45, p. 118,130]

4.1.3.1 Weight points

The control points net defines the convex hull of the geometry, but a tighter domain than the control point convex hull would be desirable.

Let us define *weight points*[12, p. 230] Q_i by setting

$$Q_i = \frac{w_i P_i + w_{i+1} P_{i+1}}{w_i + w_{i+1}} \quad (4.4)$$

Each weight point lies on the straight line that connects the two adjacent control points. Its relative position, close to the first, or to the second point, depends on their relative weights. This behavior is commonly used in design software for providing an intuitive way of modify the weight associated to a control point.

It must be borne in mind that any modification in one of the control point weights affects the whole geometry, (it is not a piecewise influence) in opposition to the control point coordinates modification whose influence extends only to a certain area of the NURBS (see Fig. 4.1).

Figure 4.1: Influence of B-spline parameters[12, p. 230]: a) Control point coordinate change. b) Control point weight change.

Convex hull property is still valid if the control polygon formed by the initial starting and ending point and all the intermediate weight points $(\mathbf{P}_0, \mathbf{Q}_0, \dots, \mathbf{Q}_{n-1}, \mathbf{P}_n)$ are considered. [12, p. 232]

The convex hull property with the set of weight points allows to use a tighter box polygon around the NURBS. This tight polygon increases the convergence in most of the iterative algorithms.

4.1.4 NURBS surface

A NURBS (Non Uniform Rational B-Spline) surface (Fig.4.2) is a biparametric surface defined by the following expression:

$$\mathbf{S}(\xi, \eta) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(\xi) N_{j,q}(\eta) \omega_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(\xi) N_{j,q}(\eta) \omega_{i,j}} \quad ; \quad 0 \leq \xi, \eta \leq 1 \quad (4.5)$$

where $\mathbf{S} = (x, y, z)$ are the coordinates of all the points on the NURBS surface, defined through the parameters (ξ, η) , $\mathbf{P}_{i,j}$ is the set of surface control points, and $\omega_{i,j}$ are the associated weights to each control point.

The functions $N_{i,j}(u)$ are the B-Spline base functions defined in §4.1.1, applied on the knotvectors:

$$\begin{aligned} U &= \{\underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_{p+1}\} \\ V &= \{\underbrace{0, \dots, 0}_{q+1}, u_{q+1}, \dots, u_{s-q-1}, \underbrace{1, \dots, 1}_{q+1}\} \end{aligned} \quad (4.6)$$

where $r = n + p + 1$ and $s = m + q + 1$, being p and q the NURBS grade in each of both directions ξ and η respectively.

The only condition that any knotvector must fulfill is its weakly increasing monotonicity.

$$u_p \leq u_{p+i} \quad \forall u_p \quad \text{and} \quad i > 0 \quad (4.7)$$

Figure 4.2: NURBS surface with its control points (displaced) [45]

4.2 NURBS analysis

There is a common understanding about the *continuity* and *curvature* concepts, but there are some nuances which will be key for the robustness in the geometry manipulation that will follow.

Continuity is a standard requirement in the aerodynamic design, but this continuity request may have some special features, because all aerodynamic problems have a privileged direction following the flow in contact with the surface [51], so that evaluation of curvature along this direction will be very interesting.

This section introduces also a procedure for defining the geometry with an uniform knotvector with the purpose of simplifying manipulation of the mathematical expressions.

Finally, some simple of continuity between NURBS geometries will be analyzed. A more extended analysis has been included in Appendix A

4.2.1 Continuity definition

There are two different continuity concepts extensively employed in geometry [7]:

Parametric continuity $\mathcal{C}^k$ Parametric continuity means smoothness both of the geometry and its parameterization.

Geometric continuity $\mathcal{G}^k$ Geometric continuity refers only to the geometry, with independence on the parameterization.

For example, a $\mathcal{C}^1$ curve has a continuous tangent vector (shape and direction) along its parameter, while a $\mathcal{G}^1$ curve only requires to maintain continuity in the direction of the tangent vector.

$\mathcal{C}^k$ is often much easier to impose than $\mathcal{G}^k$, but it also adds some unnecessary constraints. In design world, and for computational and analysis purposes, it is desirable to have a smooth parameterization but the final goal is the shape, independently on its parameterization, which is an internal feature and remains invisible for the end user.

4.2.2 NURBS curvature: curves

In order to examine the properties of the curvature in a generic NURBS geometry, and for better identifying the different dependencies, the first analysis will be done for NURBS curves.

4.2 NURBS analysis

Remembering the principles of the differential geometry, it would be desirable to get the curve defined by its arch length parameter, s .

$$\overline{C}(u) \longrightarrow \overline{C}^*(s) \quad \text{being} \quad s = s(u) = \int_0^u \left\| \frac{d\overline{C}}{dt} \right\| dt \quad (4.8)$$

As $s = s(u)$ is a continuous strictly monotonic[†] increasing function, the existence of its inverse $u = u(s)$ has been already demonstrated [8, p. 157]. Making use of the Frenet formulas:

$$\begin{aligned} \overline{C}^*(s) &= \overline{C}(u(s)) \\ \vec{t} &= \frac{d\overline{C}^*}{ds} = \frac{d\overline{C}}{du} \cdot \frac{du}{ds} \end{aligned} \quad (4.9)$$

$$K\vec{n} \underbrace{=}_{\left\| \frac{d\overline{C}^*}{ds} \right\|=1} \frac{d^2\overline{C}^*}{ds^2} = \frac{d^2\overline{C}}{du^2} \cdot \left(\frac{du}{ds} \right)^2 + \frac{d\overline{C}}{du} \cdot \frac{d^2u}{ds^2}$$

Now, a short demonstration will guide us to the continuity condition in non-natural parametrization making use of the fundamental theorem of calculus [8, p. 310] and the continuity of inverse function theorem [9, p. 144]. We can derive its extension to $\mathcal{C}^n$.[‡]

Remembering that $s(u)$ is a continuous strictly monotonic function, if $s(u)$ is derivable, then $\frac{ds}{du} \neq 0$. Then, by applying the derivative of inverse function [8, p. 189]

$$(s^{-1})'(b) = \frac{1}{s'(a)} \quad , \quad \text{where} \quad b = s^{-1}(a) \quad (4.10)$$

Then applying further derivatives on this expression we can check that as long as the denominator is not null ($s'(u) \neq 0$), the expression of the derivative of s^{-1} will maintain the continuity of $s(u)$.

$$(s^{-1})''(b) = \left[\frac{1}{s'(a)} \right]' = -\frac{s''(a)}{[s'(a)]^2} \quad (4.11)$$

Further derivatives will keep the same structure with a n -power of $s'(a)$ as denominator and the numerator will be an algebraic expression of functions whose least order of continuity is given by $s^{(n)}$. In conclusion it can be stated that:

$$C(u) \in \mathcal{C}^n \implies s(u) \in \mathcal{C}^{n+1} \underbrace{\implies}_{\frac{ds}{du} \neq 0} u(s) \in \mathcal{C}^{n+1} \quad (4.12)$$

[†]The curve parameterization must have no singularities, *i.e.*, $\frac{d\overline{C}}{du} \neq 0 \Rightarrow \frac{ds}{du} \neq 0$
This is an implicit way of imposing $\mathcal{C}^1$ and not only $\mathcal{G}^1$

[‡]This extension to $\mathcal{C}^n$ can be visually understood as a symmetry with respect to the line $s = u$, thus, this reflection does not affect the continuity of the geometry.

It means that for analyzing the continuity in curvature (K), we can focus into the most critical term $\frac{d^n \bar{C}}{du^n}$, because the term $\frac{d^n u}{ds^n}$ will maintain always one continuity order more.

This short demonstration shows that $\mathcal{C}^k$ and $\mathcal{G}^k$ are equivalent for $k = \{0, 1, 2, \dots, n\}$ in any internal point in a NURBS curve with order n , provided that $\frac{d\bar{C}}{du} \neq 0$.

It must be noted that this methodology cannot be applied when analyzing continuity between two different NURBS, because there is no common parameterization to both curves.

What would happen if $\frac{d\bar{C}}{du} = 0$? We can think that this is a strange situation that never happens, but it is not so hard to find good examples. Let's consider the 2nd order B-spline with its three control points $\{\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2\}$ in an special configuration, so that $\mathbf{P}_0 = \alpha \cdot \mathbf{P}_1 + \beta \cdot \mathbf{P}_2$ where $\alpha, \beta \in \mathbb{R}$; such case shows a continuous parameterization that will lead to geometric continuity. These kind of geometries are mostly degenerated cases and they are fortunately not extensively used, therefore we can consider that for general purposes $\frac{d\bar{C}}{du} \neq 0$

4.2.3 NURBS conventions: weights

All the weights can be assumed to be strictly positive $\omega_i > 0$, because in other case it would imply that the control point may not exist with no impact in the analytical definition of the geometry.

4.2.4 NURBS: getting an uniform knotvector

The methodology for inserting knots in the knotvector of a NURBS [45, p. 141] enables the conversion of a NURBS with a non-uniform knotvector into the same NURBS but expressed with an uniform knotvector by inserting as many intermediate knots as required to get an uniform distribution.

Using this process and without any loss of generality, we can consider that any NURBS can be defined with uniform knotvectors. If we scale this knotvectors, we can go further and state that any NURBS can be expressed with a knotvector with the following definition:

$$U = \{\underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_{p+1}\} \mid \Delta u = u_{i+1} - u_i \Rightarrow \Delta u \in \{0, 1\} \quad (4.13)$$

In such a way we can consider that any NURBS can be defined with uniform knotvector (except knots with multiplicity) and after the rescaling, the distance between consecutive knots would be always 1. This last modification has no other purpose than simplify the analytical expressions.

4.2.5 NURBS continuity analysis

Given a NURBS geometry, its continuity is defined by its degree and the knotvector. It means that, as long as the basis functions are $\mathcal{C}^n$, any composition of them will be $\mathcal{C}^n$, so, the geometry will be at least $\mathcal{C}^n$. It must be noted that the grade of the geometry is implicitly considered within the employed basis functions.

In case of a knotvector $U = \{u_0, u_1, \dots, u_m\}$ with no multiplicities

$$\left. \begin{array}{l} N_{i,0}(u) \notin \mathcal{C}^0 \\ N_{i,1}(u) \in \mathcal{C}^0 \\ N_{i,2}(u) \in \mathcal{C}^1 \\ \vdots \\ N_{i,r}(u) \in \mathcal{C}^{r-1} \end{array} \right\} \forall u, \quad u_0 \leq u < u_m$$

Each multiplicity level in a knotvector will reduce the continuity of the basis functions, in agreement with the expression for the derivatives of a basis function [45, p. 59]

$$N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} \cdot N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} \cdot N_{i+1,p-1}(u) \quad (4.14)$$

This expression also shows the limit of the continuity depending on the order p of the basis functions and the multiplicity in the knotvector, *i.e.* the derivative of a p -order basis function $N'_{i,p}(u)$ will have the lowest continuity between $N_{i,p-1}(u)$ and $N_{i+1,p-1}(u)$, unless any of the knots u_i or u_{i+1} has a multiplicity of order p , in such case, the basis function derivative will be discontinuous because the denominator will be zero.

4.2.6 NURBS analysis: continuity through control points or weights

The continuity order of a NURBS is defined by:

- the order of composing Bézier geometries
- the distribution of breaking points between Bézier geometries

Any of those aspects can be a continuity limiting factor. Inside a Bézier, the continuity order is ensured by the polynomial order (in absence of singularities, like duplicated control points), and usually, it is at the junction between different Béziers where the continuity can be reduced.

The continuity order of a geometry is, by construction, defined by the basis functions and the knotvector; but under certain conditions, the shape of the control polygon or its weights can impose the continuity that cannot be ensured by the knotvector and the order of the NURBS.

In order to better understand this way of working some examples will be explained below. For further information and description about other cases, please refer to the Appendix A

4.2.6.1 1st order, 3 control points curve without knot multiplicities

knotvector=[0,0,1,2,2]

The simplest NURBS curve example is a 1st order curve with 3 control points. The only associated knotvector may not have any multiplicity and it must be $[0, 0, 1, 2, 2]$. This set of data (NURBS order and knotvector) yields to the equation Eq.4.15 that defines the geometry completely.

Looking into detail the shape of the function given in Eq.4.15 and its associated basis functions (Fig.4.3), a clear parallelism can be established. The equation shows a linear dependence with each control point, and this is exactly what can be seen in the plot. The basis functions depict the influence of each control point $\mathbf{P}_i$ over the parameter domain u . *The basis functions can be seen as the normalized and local attraction effect that each control point has over the geometry.*

$u \in \{0, 1, 2\}$ [†] The first control point $\mathbf{p}_0$ has influence equal to 1 in $u = 0$ ($N_{0,1}(0) = 1$), likewise $\mathbf{p}_1$ and $\mathbf{p}_2$ in $u = 1$ and $u = 2$ respectively, where their associated basis functions values are $N_{1,1}(1) = N_{2,1}(1) = 1$. As the control points influence must be normalized and its maximum value cannot be greater than 1, then, no other control point may have any effect in all those points where one of the basis function reaches 1. It means that at the parameter position where $N_{i,p}(u) = 1$ (in our case $u \in \{0, 1, 2\}$), the geometry will go through the control points associated to these basis functions ($\mathbf{p}_0$, $\mathbf{p}_1$ and $\mathbf{p}_2$ respectively).

[†]The geometry is strictly defined in the semi-open interval $u \in [0, 2[$ but considering the limit value when $u \rightarrow 2^-$ we can extend the definition and close the interval where the geometry is defined, so that the curve can be defined in $u \in [0, 2]$

4.2 NURBS analysis

$0 \leq u < 1$ The only non-zero basis functions in this region are $N_{0,1}$ and $N_{1,1}$, that are linked to the control points $\mathbf{p}_0$ and $\mathbf{p}_1$. The basis functions vary lineally, and so, the same behavior will be kept in the geometry that will vary lineally between these two points. This is exactly what was expected from a 1st order NURBS, that connects with straight lines all the control points.

$1 \leq u < 2$ By symmetry reasons, the behavior in this region will be exactly the same as in the interval $0 < u < 1$.

$$\mathbf{knotvector} = [0, 0, 1, 2, 2]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{(u-1)w_0\mathbf{p}_0 - uw_1\mathbf{p}_1}{(u-1)w_0 - uw_1} & 0 \leq u < 1 \\ \frac{(u-2)w_1\mathbf{p}_1 + (1-u)w_2\mathbf{p}_2}{(u-2)w_1 + (1-u)w_2} & 1 \leq u < 2 \\ \text{undefined} & 2 \leq u \end{cases} \quad (4.15)$$

Figure 4.3: Basis functions of §4.2.6.1
(knotvector = [0,0,1,2,2])

Even in this case (polygonal line) where there seems to be no reason to analyze the continuity at its breakpoints, its simple analysis will help us to settle the procedure to follow in more complex cases.

1. The continuity order can be analyzed up to the NURBS order. Trying to impose continuity beyond the order of the NURBS will generally lead to strong constraints whose solutions are degenerated.
2. Calculate the lateral derivatives at each breakpoint by differentiating the analytical expression (Eq.4.15)
3. The balance of the lateral derivatives at both sides of the breakpoint will provide the only equation with information about the relation of the variables that affects the continuity order. The variables in this balancing equation are the control points $\mathbf{p}_i$ and their weights w_i
4. Choose the right strategy to balance the equation.
 - Modify the control points $\mathbf{p}_i$
 - Modify the weights w_i
 - Modify both, control points $\mathbf{p}_i$ and their weights w_i .

Taking as an example the continuity $\mathcal{C}^1$ at the only breakpoint ($u = 1$), the system of equations[†] results as following:

$$\begin{aligned}
 \mathcal{C}^1 \text{ at } u = 1 \implies & -w_0 \cdot \mathbf{p}_0 + (w_0 + w_2) \cdot \mathbf{p}_1 - w_2 \cdot \mathbf{p}_2 = 0 \\
 & -\mathbf{p}_0 + \left(1 + \frac{w_2}{w_0}\right) \cdot \mathbf{p}_1 - \frac{w_2}{w_0} \cdot \mathbf{p}_2 = 0 \\
 & \mathbf{p}_1 - \mathbf{p}_0 = (\mathbf{p}_2 - \mathbf{p}_1) \cdot \frac{w_2}{w_0}
 \end{aligned} \tag{4.16}$$

The result of Eq.4.16 shows that the vectors $\mathbf{p}_1 - \mathbf{p}_0$ and $\mathbf{p}_2 - \mathbf{p}_1$ must be proportional in order to achieve $\mathcal{C}^1$. The only possibility is to set the three control points aligned in a row, or in other words, $\mathbf{p}_1$ must lay on the line between $\mathbf{p}_0$ to $\mathbf{p}_2$.

4.2.6.2 2nd order 4 control points curve without knot multiplicities

knotvector=[0,0,0,1,2,2,2]

This 2nd order curve is in fact quite similar to §4.2.6.1. In the same way that the 1st order NURBS was a polygon line composed by two segments (two 1st order B-splines), which pass through the 3 control points; by increasing the order up to 2, we will enlarge the influence range of the control points and so the number of B-splines that make up

[†]It must be noticed that this is a vectorial equation $\mathbf{p}_i = (p_{i,x}, p_{i,y}, p_{i,z})$

4.2 NURBS analysis

the NURBS geometry will diminish. The minimum number of control points that are required to build a geometry composed by two 2nd order B-splines is 4.

If there are only 4 control points, there cannot be any multiplicity in the knotvector. Given an order and a certain number of control points, there is a limited number of possible knotvectors[†]

The unique set of parameter values where only one of the basis functions has the complete influence are $u = 0$ and $u = 2$, where $N_{0,2}(u)$ and $N_{3,2}(u)$ reach 1. This is the reason because the geometry will start in $\mathbf{p}_0$ and finish in $\mathbf{p}_3$. The coordinates of any point, whose parameter u belongs to the interval $]0, 2[$, will receive influence from three of the control points as shown by Eq.4.17 and Fig.4.4

[†]This limitation does not exist, but it has been imposed with the hypothesis of uniform knotvector §4.2.4

$$\mathbf{knotvector} = [0, 0, 0, 1, 2, 2, 2]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{2w_0\mathbf{p}_0 - 4w_0\mathbf{p}_0u + 2w_0\mathbf{p}_0u^2 + 4w_1\mathbf{p}_1u - 3w_1\mathbf{p}_1u^2 + w_2\mathbf{p}_2u^2}{2w_0 - 4w_0u + 2w_0u^2 + 4w_1u - 3w_1u^2 + w_2u^2} & 0 \leq u < 1 \\ \frac{4w_1\mathbf{p}_1 - 4w_1\mathbf{p}_1u + w_1\mathbf{p}_1u^2 + 8w_2\mathbf{p}_2u - 3w_2\mathbf{p}_2u^2 - 4w_2\mathbf{p}_2 + 2w_3\mathbf{p}_3 - 4w_3\mathbf{p}_3u + 2w_3\mathbf{p}_3u^2}{4w_1 - 4w_1u + w_1u^2 + 8w_2u - 3w_2u^2 - 4w_2 + 2w_3 - 4w_3u + 2w_3u^2} & 1 \leq u < 2 \\ \text{undefined} & 2 \leq u \end{cases} \quad (4.17)$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (4.18)$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies (w_1 + w_2)w_0\mathbf{p}_0 - (4w_2 - w_3 + w_0)w_1\mathbf{p}_1 - (-4w_1 + w_0 - w_3)w_2\mathbf{p}_2 - (w_1 + w_2)w_3\mathbf{p}_3 = 0 \quad (4.19)$$

$$\text{a) } \xrightarrow{w_0=w_3=A} A(w_1 + w_2)\mathbf{p}_0 - 4w_1w_2\mathbf{p}_1 + 4w_1w_2\mathbf{p}_2 - A(w_1 + w_2)\mathbf{p}_3 = 0$$

$$\mathbf{p}_0 - \mathbf{p}_3 = \frac{4w_1w_2}{A(w_1 + w_2)}(\mathbf{p}_1 - \mathbf{p}_2) \quad (4.20)$$

$$\text{b) } \longrightarrow (w_0\mathbf{p}_0 - w_3\mathbf{p}_3) = \frac{1}{w_1 + w_2} [4w_1w_2(\mathbf{p}_1 - \mathbf{p}_2) - (w_3 - w_0)(w_1\mathbf{p}_1 + w_2\mathbf{p}_2)] \quad (4.21)$$

$$\xrightarrow{\mathbf{p}_i^* = \mathbf{p}_i - \mathbf{p}_0} -w_3\mathbf{p}_3^* = \frac{1}{w_1 + w_2} [4w_1w_2(\mathbf{p}_1^* - \mathbf{p}_2^*) - (w_3 - w_0)(w_1\mathbf{p}_1^* + w_2\mathbf{p}_2^*)] \quad (4.22)$$

Figure 4.4: Basis functions of §4.2.6.2
(knotvector = [0,0,0,1,2,2,2])

Imposing continuity at the breaking point leads to Eq.4.18 and Eq.4.19. The first of them shows that the 1st order continuity ($\mathcal{C}^1$) is imposed by construction, *i.e. any NURBS without knot multiplicity keeps its continuity one order below its grade*. The second equation states the conditions in $\mathbf{p}_i$ and w_i that must be satisfied in order to impose 2nd order continuity ($\mathcal{C}^2$). Unfortunately, this second equation cannot be easily solved, so we will start by considering certain hypotheses (a) in order to furtherly extract the maximum information from these equations (b).

- a) The easiest and most representative approach is to consider that the furthest control points have a constant weight equal to A ($w_0 = w_3 = A$). This simplification is not so odd because most of commercial design software define the NURBS weights as 1. This assumption leads to the equation 4.20, that can be understood as a scale relationship between the vectors $\mathbf{p}_0 - \mathbf{p}_3$ and $\mathbf{p}_1 - \mathbf{p}_2$. Once those vectors are parallel and they have the same direction, the length relation can be adjusted with the weights w_1 and w_2 .
- b) The expression in Eq.4.21 shows the general relationship with all the variables and before any artificial assumption. It has been demonstrated that NURBS geometries are independent against translation and rotations [45], *i.e. changes in the reference system*; applying this property, we can find a new reference system where the vectors defined by $\mathbf{p}_0$ and $\mathbf{p}_3$ are proportional ($\mathbf{p}_0 \propto \mathbf{p}_3$), or even better, setting the coordinate origin in $w_0 \mathbf{p}_0$; in such case, part of the left hand side vanishes and it remains as $-w_3 \mathbf{p}_3^*$ [†]. If $\mathbf{p}_1^*$ and $\mathbf{p}_2^*$ are linearly independent, then they constitute a subspace

[†]where $\mathbf{p}_i^* = \mathbf{p}_i - w_0 \mathbf{p}_0$

U with dimension 2 ($U = \{\mathbf{p}_1^*, \mathbf{p}_2^*\} : U \subset \mathbb{R}^2$), and $\{\overrightarrow{\mathbf{p}_1^* - \mathbf{p}_2^*}, \overrightarrow{w_1\mathbf{p}_1^* + w_2\mathbf{p}_2^*}\}$ will also be a base of the subspace U . Then, a solution of Eq.4.21 may exist only if $\mathbf{p}_3^*$ belongs to the subspace U ($\mathbf{p}_3^* \in U$). In other words, *the solution may exist only if the four control points are in the same plane*. However, the weights normalization ($0 < w_i \leq 1$), and the equation structure impose some other restrictions to the solution existence. Let's examine the different terms in the equation:

- $(w_3 - w_0)(w_1\mathbf{p}_1^* + w_2\mathbf{p}_2^*)$ depending on the values of w_1 and w_2 may be any vector with its bound laying in the parallelepiped whose sides are $\mathbf{p}_1$ and $\mathbf{p}_2$ (gray area in Fig.4.5). The factor $(w_3 - w_0)$ may vary from -1 to 1, so the range of possible values is extended and it includes also the reflected parallelepiped.
- $4w_1w_2(\mathbf{p}_1^* - \mathbf{p}_2^*)$ is by definition coplanar with $\mathbf{p}_1$ and $\mathbf{p}_2$. It acts as an offset for $(w_3 - w_0)(w_1\mathbf{p}_1^* + w_2\mathbf{p}_2^*)$.
- $-w_3\mathbf{p}_3^*$ must lie on the right hand side accessible area. An appropriate w_3 value may be imposed to ensure solution existence, unless any of the following:
 - $(\mathbf{p}_3^* \wedge \mathbf{p}_2^*) \cdot (\mathbf{p}_3^* \wedge \mathbf{p}_1^*) > 0$ *i.e.* $\mathbf{p}_3^*$ is between $\mathbf{p}_2^*$ and $-\mathbf{p}_1^*$.
 - $-\mathbf{p}_3^* \cdot \frac{(\mathbf{p}_1^* - \mathbf{p}_2^*)}{\|(\mathbf{p}_1^* - \mathbf{p}_2^*)\|} > 4\frac{w_1w_2}{w_3}(\mathbf{p}_1^* - \mathbf{p}_2^*)$ the projection of $\mathbf{p}_3^*$ on $(\mathbf{p}_1^* - \mathbf{p}_2^*)$ is greater than $(\mathbf{p}_1^* - \mathbf{p}_2^*)$

Figure 4.5: Analysis figure for the range of weights solution existence in §4.2.6.2

4.2.6.3 2nd order, 5 control points curve without knot multiplicities knotvector=[0,0,0,1,2,3,3,3]

Adding a new control point will modify the NURBS expression, thus a new element will appear in the breaking point list ($u = 1$, $u = 2$). Due to symmetry in the basis functions (Fig.4.6), the behavior of both breaking points will be basically the same, in agreement with the continuity conditions in Eq.4.25 and Eq.4.30.

Remembering that *NURBS are invariable against changes in the coordinate system*. We can follow the same method as in §4.2.6.2 for analyzing the breaking point $u = 1$ (Eq.4.27). In this case, the origin will be set in $\mathbf{p}_3$ i.e. ($\mathbf{p}_i^* = \mathbf{p}_i - \mathbf{p}_3$). The left hand side of Eq.4.28 must belong to the subspace U defined by the vectors $\{\mathbf{p}_1^*, \mathbf{p}_2^*\}$. If these two vectors were not independent, it means that $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ are collinear and the only option is that also $\mathbf{p}_0$ lays aligned to them.

Figure 4.6: Basis functions of §4.2.6.3
(knotvector = [0,0,0,1,2,3,3,3])

$$\begin{aligned}
& \mathbf{knotvector} = [0, 0, 0, 1, 2, 3, 3, 3] \\
C(u) = & \begin{cases} \text{undefined} & u < 0 \\ \frac{2w_0\mathbf{p}_0 - 4w_0\mathbf{p}_0u + 2w_0\mathbf{p}_0u^2 + 4uw_1\mathbf{p}_1 - 3u^2w_1\mathbf{p}_1 + w_2\mathbf{p}_2u^2}{2w_0 - 4w_0u + 2w_0u^2 + 4uw_1 - 3u^2w_1 + w_2u^2} & 0 \leq u < 1 \\ \frac{4w_1\mathbf{p}_1 - 4uw_1\mathbf{p}_1 + u^2w_1\mathbf{p}_1 + 6w_2\mathbf{p}_2u - 2w_2\mathbf{p}_2u^2 - 3w_2\mathbf{p}_2 + w_3\mathbf{p}_3 - 2w_3\mathbf{p}_3u + w_3\mathbf{p}_3u^2}{4w_1 - 4uw_1 + u^2w_1 + 6w_2u - 2w_2u^2 - 3w_2 + w_3 - 2w_3u + w_3u^2} & 1 \leq u < 2 \\ \frac{9w_2\mathbf{p}_2 - 6w_2\mathbf{p}_2u + w_2\mathbf{p}_2u^2 - 15w_3\mathbf{p}_3 + 14w_3\mathbf{p}_3u - 3w_3\mathbf{p}_3u^2 + 8w_4\mathbf{p}_4 - 8w_4\mathbf{p}_4u + 2w_4\mathbf{p}_4u^2}{9w_2 - 6w_2u + w_2u^2 - 15w_3 + 14w_3u - 3w_3u^2 + 8w_4 - 8w_4u + 2w_4u^2} & 2 \leq u < 3 \\ \text{undefined} & 3 \leq u \end{cases} \quad (4.23)
\end{aligned}$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (4.24)$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 2(w_1 + w_2)w_0\mathbf{p}_0 + (-7w_2 - 2w_0 + w_3)w_1\mathbf{p}_1 + (7w_1 - 2w_0 + w_3)w_2\mathbf{p}_2 - (w_1 + w_2)w_3\mathbf{p}_3 = 0 \quad (4.25)$$

$$\text{a) } \xrightarrow{w_i=A} A^2(2\mathbf{p}_0 - 4\mathbf{p}_1 + 3\mathbf{p}_2 - \mathbf{p}_3) = 0 \quad (4.26)$$

$$\text{b) } \longrightarrow (w_1 + w_2)(2w_0\mathbf{p}_0 - w_3\mathbf{p}_3) = (7w_2 + 2w_0 - w_3)w_1\mathbf{p}_1 + (-7w_1 + 2w_0 - w_3)w_2\mathbf{p}_2 \quad (4.27)$$

$$\xrightarrow{\mathbf{p}_i^* = \mathbf{p}_i - \mathbf{p}_3} 2(w_1 + w_2)w_0\mathbf{p}_0^* = (7w_2 + 2w_0 - w_3)w_1\mathbf{p}_1^* + (-7w_1 + 2w_0 - w_3)w_2\mathbf{p}_2^* \quad (4.28)$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (4.29)$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies (w_2 + w_3)w_1\mathbf{p}_1 + (-7w_3 + 2w_4 - w_1)w_2\mathbf{p}_2 + (7w_2 + 2w_4 - w_1)w_3\mathbf{p}_3 - 2(w_2 + w_3)w_4\mathbf{p}_4 = 0 \quad (4.30)$$

4.2.6.4 2nd order, 5 control points curve with knot multiplicity knotvector=[0,0,0,1,1,2,2,2]

If we force one of the control points to be a through point, then, the associated basis function will reach value 1 in that point. This condition is imposed through the knot multiplicity. A second order curve with knot multiplicity equal to 1 is equivalent to fix the curve coordinates in that point (*through point*).

However, this solely influence from one of the control points affects the continuity. Each multiplicity level reduces in one the continuity order, *i.e.* in a 2nd order curve, a knot with multiplicity equal to 1 (the knotvector has twice the same internal knot value), will reduce the default continuity order from $\mathcal{C}^1$ to $\mathcal{C}^0$.

According to Eq.4.33, the only way to impose $\mathcal{C}^1$ is by setting up $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ aligned on the same straight line, so that $\mathbf{p}_2^* = \mathbf{p}_2 - \mathbf{p}_1$ and $\mathbf{p}_3^* = \mathbf{p}_3 - \mathbf{p}_1$ are proportional. Considering that $w_i > 0$, we can deduct that $\mathbf{p}_2$ must be between $\mathbf{p}_1$ and $\mathbf{p}_3$.

Figure 4.7: Basis functions of §4.2.6.4
(knotvector = [0,0,0,1,1,2,2,2])

A closer exam at the basis function plot (Fig.4.7) shows its symmetry with respect to $u = 1$. The basis function shape in both sides are exactly the same as the one obtained for a 2nd degree NURBS with 3 control points, that is a conic curve (parabola, ellipse or hyperbola depending on the chosen weights)[12]. It means that the NURBS can be split in two without any modification or loss of information.

A NURBS is composed by two independent NURBS when one of its knots has a multiplicity level that reaches NURBS grade-1. Such examples can be found in Fig.4.7 and Fig.4.10.[†]

[†]A detailed description of this composition can be found in §4.2.6.6

$$\begin{aligned}
& \mathbf{knotvector} = [0, 0, 0, 1, 1, 2, 2, 2] \\
C(u) = & \begin{cases} \text{undefined} & u < 0 \\
\frac{w_0 \mathbf{p}_0 - 2 w_0 \mathbf{p}_0 u + w_0 \mathbf{p}_0 u^2 + 2 u w_1 \mathbf{p}_1 - 2 u^2 w_1 \mathbf{p}_1 + w_2 \mathbf{p}_2 u^2}{w_0 - 2 w_0 u + w_0 u^2 + 2 u w_1 - 2 u^2 w_1 + w_2 u^2} & 0 \leq u < 1 \\
\frac{4 w_2 \mathbf{p}_2 - 4 w_2 \mathbf{p}_2 u + w_2 \mathbf{p}_2 u^2 - 4 w_3 \mathbf{p}_3 + 6 w_3 \mathbf{p}_3 u - 2 w_3 \mathbf{p}_3 u^2 + w_4 \mathbf{p}_4 - 2 w_4 \mathbf{p}_4 u + w_4 \mathbf{p}_4 u^2}{4 w_2 - 4 w_2 u + w_2 u^2 - 4 w_3 + 6 w_3 u - 2 w_3 u^2 + w_4 - 2 w_4 u + w_4 u^2} & 1 \leq u < 2 \\
\text{undefined} & 2 \leq u \end{cases} \quad (4.31)
\end{aligned}$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies -w_1 \mathbf{p}_1 + (w_1 + w_3) \mathbf{p}_2 - w_3 \mathbf{p}_3 = 0 \quad (4.32)$$

$$\xrightarrow{\mathbf{p}_i^* = \mathbf{p}_i - \mathbf{p}_1} (w_1 + w_3) \mathbf{p}_2^* = w_3 \mathbf{p}_3^* \quad (4.33)$$

$$\begin{aligned}
\mathcal{C}^2 \text{ at } u = 1 \implies & w_0 \mathbf{p}_0 w_2 - 2 w_1 (2 w_1 - w_2) \mathbf{p}_1 + (4 w_1^2 - 4 w_3^2 + (w_4 - w_0 - 2 w_1 + 2 w_3) w_2) \mathbf{p}_2 \\
& - 2 w_3 (-2 w_3 + w_2) \mathbf{p}_3 - w_4 \mathbf{p}_4 w_2 = 0 \quad (4.34)
\end{aligned}$$

4.2.6.5 2nd order, 6 control points curve without knot multiplicity

knotvector=[0,0,0,1,2,3,4,4,4]

The addition of a new control point will increase the control along the curve, but it also imposes new breaking points (junction points between B-splines) enlarge the number of possibilities.

As long as we do not have any knot multiplicity, the continuity will be kept as $\mathcal{C}^1$. This example shows clearly the local influence of each control point via the action from the basis functions. The number of control points with influence on a certain point is determined based on the NURBS order and the knotvector with its multiplicities.

Figure 4.8: Basis functions of §4.2.6.5

(knotvector = [0,0,0,1,2,3,4,4,4])

According to this local support property, the continuity in a given point will depend only on a set of control points and their weights. We can add as many control points as we wish but only a set of them can modify the continuity in that point. This can be easily understood watching at the symmetry of the equations, or having a look at the shape of the basis functions. Two points with equally shaped basis functions will keep the same dependences on the associated control points. A good example can be found if we compare §4.2.6.5 and §4.2.6.7. The only difference between both cases is the number of control points; in the second one, a new control point has been added, but no differences can be found between Eq.4.51, Eq.4.53 and Eq.4.39.

$$\mathbf{knotvector} = [0, 0, 0, 1, 2, 3, 4, 4, 4]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{2w_0\mathbf{p}_0 - 4w_0\mathbf{p}_0u + 2w_0\mathbf{p}_0u^2 + 4uw_1\mathbf{p}_1 - 3u^2w_1\mathbf{p}_1 + w_2\mathbf{p}_2u^2}{2w_0 - 4w_0u + 2w_0u^2 + 4uw_1 - 3u^2w_1 + w_2u^2} & 0 \leq u < 1 \\ \frac{4w_1\mathbf{p}_1 - 4uw_1\mathbf{p}_1 + u^2w_1\mathbf{p}_1 + 6w_2\mathbf{p}_2u - 2w_2\mathbf{p}_2u^2 - 3w_2\mathbf{p}_2 + w_3\mathbf{p}_3 - 2w_3\mathbf{p}_3u + w_3\mathbf{p}_3u^2}{-3w_2 + w_3 + 4w_1 - 4uw_1 + u^2w_1 + 6w_2u - 2w_2u^2 - 2w_3u + w_3u^2} & 1 \leq u < 2 \\ \frac{9w_2\mathbf{p}_2 - 6w_2\mathbf{p}_2u + w_2\mathbf{p}_2u^2 - 11w_3\mathbf{p}_3 + 10w_3\mathbf{p}_3u - 2w_3\mathbf{p}_3u^2 + 4w_4\mathbf{p}_4 - 4w_4\mathbf{p}_4u + w_4\mathbf{p}_4u^2}{-11w_3 + 4w_4 + 9w_2 + w_2u^2 - 6w_2u + 10w_3u - 2w_3u^2 - 4w_4u + w_4u^2} & 2 \leq u < 3 \\ \frac{16w_3\mathbf{p}_3 - 8w_3\mathbf{p}_3u + w_3\mathbf{p}_3u^2 - 32w_4\mathbf{p}_4 + 20w_4\mathbf{p}_4u - 3w_4\mathbf{p}_4u^2 + 18w_5\mathbf{p}_5 - 12w_5\mathbf{p}_5u + 2w_5\mathbf{p}_5u^2}{18w_5 + 16w_3 - 32w_4 - 8w_3u + w_3u^2 + 20w_4u - 3w_4u^2 - 12w_5u + 2w_5u^2} & 3 \leq u < 4 \\ \text{undefined} & 4 \leq u \end{cases} \quad (4.35)$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (4.36)$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 2w_0(w_1 + w_2)\mathbf{p}_0 - w_1(2w_0 - w_3 + 7w_2)\mathbf{p}_1 - w_2(-7w_1 + 2w_0 - w_3)\mathbf{p}_2 - w_3(w_1 + w_2)\mathbf{p}_3 = 0 \quad (4.37)$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (4.38)$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies (w_2 + w_3)w_1\mathbf{p}_1 - w_2(w_1 - w_4 + 6w_3)\mathbf{p}_2 - w_3(-6w_2 + w_1 - w_4)\mathbf{p}_3 - w_4(w_2 + w_3)\mathbf{p}_4 = 0 \quad (4.39)$$

$$\mathcal{C}^1 \text{ at } u = 3 \implies 0 = 0 \quad (4.40)$$

$$\mathcal{C}^2 \text{ at } u = 3 \implies w_2(w_3 + w_4)\mathbf{p}_2 - w_3(w_2 + 7w_4 - 2w_5)\mathbf{p}_3 - w_4(-7w_3 + w_2 - 2w_5)\mathbf{p}_4 - 2w_5(w_3 + w_4)\mathbf{p}_5 = 0 \quad (4.41)$$

4.2.6.6 2nd order, 6 control points curve with knot multiplicity

knotvector=[0,0,0,1,1,2,3,3,3]

The set of basis functions shows a peak in $u = 1$. This peak is due to the existence of two knots whose values are 1. A knot with multiplicity equal to 1, in a second order NURBS, defines a through point in the curve. This through point acts as a bounding control point of two smaller NURBS. The comparison between the plots Fig.4.9, Fig.4.7 and Fig.4.6, as depicted in Fig.4.10 shows clearly, how the basis functions set of a 2nd order NURBS with knotvector=[0,0,0,1,1,2,3,3,3] can be understood as the union between two 2nd order NURBS with knotvectors [0,0,0,1,1,2,2,2] and [0,0,0,1,2,3,3,3]

Figure 4.9: Basis functions of §4.2.6.6
(knotvector = [0,0,0,1,1,2,3,3,3])

Figure 4.10: NURBS structure: composition of other NURBS

$$\mathbf{knotvector} = [0, 0, 0, 1, 1, 2, 3, 3, 3]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{w_0 \mathbf{p}_0 - 2w_0 \mathbf{p}_0 u + w_0 \mathbf{p}_0 u^2 + 2uw_1 \mathbf{p}_1 - 2u^2 w_1 \mathbf{p}_1 + w_2 \mathbf{p}_2 u^2}{w_0 - 2w_0 u + w_0 u^2 + 2uw_1 - 2u^2 w_1 + w_2 u^2} & 0 \leq u < 1 \\ \frac{8w_2 \mathbf{p}_2 - 8w_2 \mathbf{p}_2 u + 2w_2 \mathbf{p}_2 u^2 - 7w_3 \mathbf{p}_3 + 10w_3 \mathbf{p}_3 u - 3w_3 \mathbf{p}_3 u^2 + w_4 \mathbf{p}_4 - 2w_4 \mathbf{p}_4 u + w_4 \mathbf{p}_4 u^2}{8w_2 - 8w_2 u + 2w_2 u^2 - 7w_3 + 10w_3 u - 3w_3 u^2 + w_4 - 2w_4 u + w_4 u^2} & 1 \leq u < 2 \\ \frac{9w_3 \mathbf{p}_3 - 6w_3 \mathbf{p}_3 u + w_3 \mathbf{p}_3 u^2 - 15w_4 \mathbf{p}_4 + 14w_4 \mathbf{p}_4 u - 3w_4 \mathbf{p}_4 u^2 + 8w_5 \mathbf{p}_5 - 8w_5 \mathbf{p}_5 u + 2w_5 \mathbf{p}_5 u^2}{9w_3 - 6w_3 u + w_3 u^2 - 15w_4 + 14w_4 u - 3w_4 u^2 + 8w_5 - 8w_5 u + 2w_5 u^2} & 2 \leq u < 3 \\ \text{undefined} & 3 \leq u \end{cases} \quad (4.42)$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies -w_1 \mathbf{p}_1 + (w_1 + w_3) \mathbf{p}_2 - w_3 \mathbf{p}_3 = 0 \quad (4.43)$$

$$\begin{aligned} \mathcal{C}^2 \text{ at } u = 1 \implies & 2w_0 \mathbf{p}_0 w_2 - 4w_1 (2w_1 - w_2) \mathbf{p}_1 + (-8w_3^2 + 8w_1^2 - 4w_1 w_2 + w_2 w_4 + 5w_2 w_3 - 2w_0 w_2) \mathbf{p}_2 \\ & - w_3 (-8w_3 + 5w_2) \mathbf{p}_3 - w_4 \mathbf{p}_4 w_2 = 0 \end{aligned} \quad (4.44)$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (4.45)$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies w_2 (w_3 + w_4) \mathbf{p}_2 - w_3 (4w_4 - w_5 + w_2) \mathbf{p}_3 - w_4 (-4w_3 + w_2 - w_5) \mathbf{p}_4 - w_5 (w_3 + w_4) \mathbf{p}_5 = 0 \quad (4.46)$$

4.2.6.7 2nd order, 7 control points curve without knot multiplicity
 knotvector=[0,0,0,1,2,3,4,5,5,5]

Figure 4.11: Basis functions of §4.2.6.7
 (knotvector = [0,0,0,1,2,3,4,5,5,5])

$$\mathbf{knotvector} = [0, 0, 0, 1, 2, 3, 4, 5, 5, 5]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{2w_0\mathbf{p}_0 - 4w_0\mathbf{p}_0u + 2w_0\mathbf{p}_0u^2 + 4w_1\mathbf{p}_1u - 3w_1\mathbf{p}_1u^2 + w_2\mathbf{p}_2u^2}{2w_0 - 4w_0u + 2w_0u^2 + 4w_1u - 3w_1u^2 + w_2u^2} & 0 \leq u < 1 \\ \frac{4w_1\mathbf{p}_1 - 4w_1\mathbf{p}_1u + w_1\mathbf{p}_1u^2 + 6w_2\mathbf{p}_2u - 2w_2\mathbf{p}_2u^2 - 3w_2\mathbf{p}_2 + w_3\mathbf{p}_3 - 2w_3\mathbf{p}_3u + w_3\mathbf{p}_3u^2}{4w_1 - 4w_1u + w_1u^2 + 6w_2u - 2w_2u^2 - 3w_2 + w_3 - 2w_3u + w_3u^2} & 1 \leq u < 2 \\ \frac{9w_2\mathbf{p}_2 - 6w_2\mathbf{p}_2u + w_2\mathbf{p}_2u^2 - 11w_3\mathbf{p}_3 + 10w_3\mathbf{p}_3u - 2w_3\mathbf{p}_3u^2 + 4w_4\mathbf{p}_4 - 4w_4\mathbf{p}_4u + w_4\mathbf{p}_4u^2}{-6w_2u + w_2u^2 + 10w_3u - 2w_3u^2 - 4w_4u + w_4u^2 + 9w_2 - 11w_3 + 4w_4} & 2 \leq u < 3 \\ \frac{16w_3\mathbf{p}_3 - 8w_3\mathbf{p}_3u + w_3\mathbf{p}_3u^2 - 23w_4\mathbf{p}_4 + 14w_4\mathbf{p}_4u - 2w_4\mathbf{p}_4u^2 + 9w_5\mathbf{p}_5 - 6w_5\mathbf{p}_5u + w_5\mathbf{p}_5u^2}{-6w_5u + w_5u^2 - 8w_3u + w_3u^2 + 14w_4u - 2w_4u^2 - 23w_4 + 16w_3 + 9w_5} & 3 \leq u < 4 \\ \frac{25w_4\mathbf{p}_4 - 10w_4\mathbf{p}_4u + w_4\mathbf{p}_4u^2 - 55w_5\mathbf{p}_5 + 26w_5\mathbf{p}_5u - 3w_5\mathbf{p}_5u^2 + 32w_6\mathbf{p}_6 - 16w_6\mathbf{p}_6u + 2w_6\mathbf{p}_6u^2}{-10w_4u + w_4u^2 + 26w_5u - 3w_5u^2 - 16w_6u + 2w_6u^2 - 55w_5 + 25w_4 + 32w_6} & 4 \leq u < 5 \\ \text{undefined} & 5 \leq u \end{cases} \quad (4.47)$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (4.48)$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 2w_0(w_1 + w_2)\mathbf{p}_0 - w_1(2w_0 - w_3 + 7w_2)\mathbf{p}_1 - w_2(-7w_1 + 2w_0 - w_3)\mathbf{p}_2 - w_3(w_1 + w_2)\mathbf{p}_3 = 0 \quad (4.49)$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (4.50)$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies w_1(w_2 + w_3)\mathbf{p}_1 - w_2(w_1 + 6w_3 - w_4)\mathbf{p}_2 - w_3(-6w_2 + w_1 - w_4)\mathbf{p}_3 - w_4(w_2 + w_3)\mathbf{p}_4 = 0 \quad (4.51)$$

$$\mathcal{C}^1 \text{ at } u = 3 \implies 0 = 0 \quad (4.52)$$

$$\mathcal{C}^2 \text{ at } u = 3 \implies w_2(w_3 + w_4)\mathbf{p}_2 - w_3(w_2 + 6w_4 - w_5)\mathbf{p}_3 - w_4(-6w_3 + w_2 - w_5)\mathbf{p}_4 - w_5(w_3 + w_4)\mathbf{p}_5 = 0 \quad (4.53)$$

$$\mathcal{C}^1 \text{ at } u = 4 \implies 0 = 0 \quad (4.54)$$

$$\mathcal{C}^2 \text{ at } u = 4 \implies w_3(w_4 + w_5)\mathbf{p}_3 - w_4(7w_5 - 2w_6 + w_3)\mathbf{p}_4 - w_5(-7w_4 + w_3 - 2w_6)\mathbf{p}_5 - 2w_6(w_4 + w_5)\mathbf{p}_6 = 0 \quad (4.55)$$

4.2.6.8 2nd order, 7 control points curve with knot multiplicity

knotvector=[0,0,0,1,1,2,2,3,3,3]

This is not a control points NURBS curve but a composition of three parabolas. The multiplicity in the inner points splits the curve in several pieces that are only $\mathcal{C}^0$, similarly to §4.2.6.6

Figure 4.12: Basis functions of §4.2.6.8

(knotvector = [0,0,0,1,1,2,2,3,3,3])

$$\mathbf{knotvector} = [0, 0, 0, 1, 1, 2, 2, 3, 3, 3]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{w_0 \mathbf{p}_0 - 2w_0 \mathbf{p}_0 u + w_0 \mathbf{p}_0 u^2 + 2w_1 \mathbf{p}_1 u - 2w_1 \mathbf{p}_1 u^2 + w_2 \mathbf{p}_2 u^2}{w_0 - 2w_0 u + w_0 u^2 + 2w_1 u - 2w_1 u^2 + w_2 u^2} & 0 \leq u < 1 \\ \frac{4w_2 \mathbf{p}_2 - 4w_2 \mathbf{p}_2 u + w_2 \mathbf{p}_2 u^2 - 4w_3 \mathbf{p}_3 + 6w_3 \mathbf{p}_3 u - 2w_3 \mathbf{p}_3 u^2 + w_4 \mathbf{p}_4 - 2w_4 \mathbf{p}_4 u + w_4 \mathbf{p}_4 u^2}{4w_2 - 4w_2 u + w_2 u^2 - 4w_3 + 6w_3 u - 2w_3 u^2 + w_4 - 2w_4 u + w_4 u^2} & 1 \leq u < 2 \\ \frac{9w_4 \mathbf{p}_4 - 6w_4 \mathbf{p}_4 u + w_4 \mathbf{p}_4 u^2 - 12w_5 \mathbf{p}_5 + 10w_5 \mathbf{p}_5 u - 2w_5 \mathbf{p}_5 u^2 + 4w_6 \mathbf{p}_6 - 4w_6 \mathbf{p}_6 u + w_6 \mathbf{p}_6 u^2}{9w_4 - 6w_4 u + w_4 u^2 - 12w_5 + 10w_5 u - 2w_5 u^2 + 4w_6 - 4w_6 u + w_6 u^2} & 2 \leq u < 3 \\ \text{undefined} & 3 \leq u \end{cases} \quad (4.56)$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies -w_1 \mathbf{p}_1 + (w_1 + w_3) \mathbf{p}_2 - w_3 \mathbf{p}_3 = 0 \quad (4.57)$$

$$\begin{aligned} \mathcal{C}^2 \text{ at } u = 1 \implies w_0 \mathbf{p}_0 w_2 - 2w_1 (2w_1 - w_2) \mathbf{p}_1 + (4w_1^2 - 2w_1 w_2 - 4w_3^2 - w_0 w_2 + w_2 w_4 + 2w_2 w_3) \mathbf{p}_2 \\ - 2w_3 (-2w_3 + w_2) \mathbf{p}_3 - w_4 \mathbf{p}_4 w_2 = 0 \end{aligned} \quad (4.58)$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies -w_3 \mathbf{p}_3 + (w_3 + w_5) \mathbf{p}_4 - w_5 \mathbf{p}_5 = 0 \quad (4.59)$$

$$\begin{aligned} \mathcal{C}^2 \text{ at } u = 2 \implies w_2 \mathbf{p}_2 w_4 - 2w_3 (2w_3 - w_4) \mathbf{p}_3 + (4w_3^2 - w_2 w_4 - 2w_3 w_4 - 4w_5^2 + 2w_4 w_5 + w_4 w_6) \mathbf{p}_4 \\ - 2w_5 (-2w_5 + w_4) \mathbf{p}_5 - w_6 \mathbf{p}_6 w_4 = 0 \end{aligned} \quad (4.60)$$

5

Point inversion

Contents

5.1	Objective	62
5.2	Problem definition	63
5.3	Problem solving	63
5.4	Point inversion techniques: description	66
5.4.1	Grid search	66
5.4.2	“Quad” search	67
5.4.3	“Quadratic” approximation search	68
5.4.4	First order Hu-Wallner algorithm	70
5.4.5	Second order Hu-Wallner algorithm	71
5.5	Point inversion techniques: evaluation	73
5.5.1	Grid search	75
5.5.2	“Quad” search	78
5.5.3	“Quadratic” search	79
5.5.4	Hu-Wallner 1 st order search	81
5.5.5	Hu-Wallner 2 nd order search	82

5.1 Objective

Once the geometry to be optimized has been properly described, it is important to remind that the adjoint solver only provides the sensitivity with respect to the objective function in the mesh nodes, but not in the geometry parameters (see §3.2.2). This chapter builds the link between the geometric parameterization and the mesh definition. The final goal is to establish the relation $\partial\vec{X}/\partial\vec{P}$ as requested by Eq. 5.1.

$$\frac{\partial J}{\partial\vec{P}} = \frac{\partial J}{\partial\vec{X}} \cdot \frac{\partial\vec{X}}{\partial\vec{P}} \quad (5.1)$$

Due to the heuristic nature of this relation search, a complete set of point inversion techniques will be assessed against a benchmark of surfaces, in order to identify the best combination of techniques to ensure the reliability of the solution.

5.1 Objective

The main objective of this task is to establish a link between the coordinates of each surface mesh node $\vec{X}$ and the analytic definition of the surface geometry $\mathbf{S}(\xi, \eta)$, in such a way that, the derivative of the coordinates of any surface mesh point with respect to each of the surface parameters $\frac{\partial\vec{X}}{\partial\vec{P}}$ [†] could be easily obtained. It must be noted that the evaluation of the coordinates of a point $\vec{X} = (x, y, z)$ in a NURBS surface $\mathbf{S}$, given its set of parametric coordinates (η, ξ) , is extremely efficient and finite differences can be employed at relatively low computational cost in order to evaluate these derivatives.

Once this relation between mesh points and geometry definition has been established, there will be no further need of remeshing after any small change in the surface parameters. The obtained set of parametric coordinates (ξ, η) for each surface grid point remains constant against modifications of the geometry parameters which will only affect the shape of the surface. Thus, any geometry parameter modification $\mathbf{S} \rightarrow \mathbf{S}'$ results in a new set of surface mesh nodes with identical parametric coordinates (ξ, η) but different spatial coordinates (x, y, z) . This methodology maintains the topology of the surface grid, so that the just obtained surface mesh (composed by points of $\mathbf{S}'$) can be used as an input for the deformation of the complete volume mesh.

Only if the new deformed surface $\mathbf{S}'$ intersects a trimming surface, the parametric coordinates (ξ, η) of the mesh points will need to be deformed to match the new trimmed parameters space. But in this case, the trimming curve is known as well in the parametric space, and the deformation of the surface mesh can be done in the 2D space of the parameters.

[†] $\vec{P}$ is the set of parameters that define the geometry

5.2 Problem definition

Point inversion is the process of identifying the parametric coordinates (ξ, η) that define the point $\vec{Q}$ as belonging to the surface $\mathbf{S}$, defined as a NURBS, given the spatial coordinates (x, y, z) of this point $\vec{Q}$ that lies on $\mathbf{S}$

Actually, the real point inversion problem applied to the CFD is slightly different because it cannot be ensured that the point $\vec{Q}$ lies exactly on the surface $\mathbf{S}^\dagger$ so that the definition of the problem can be formulated as: *Given the coordinates (x, y, z) of a point $\vec{R}$ in the space and the surface $\mathbf{S}$, defined analytically as a NURBS, we need to obtain the parametric coordinates (ξ, η) of the point $\vec{Q}$ on the surface $\mathbf{S}$ that minimizes its distance to the point $\vec{R}$.*

This problem can be addressed from two perspectives:

- minimum distance between $\vec{R}$ and $\vec{Q}$
- definition of the normal to the surface $\mathbf{S}$ on the point $\vec{R}$

Although the two options may seem equivalent, there could be relevant differences between them if the point to be inverted does not lie in the inner part of the surface. The choice of one method or the other depends on the particular problem (*i.e.* the position of $\vec{R}$). The methodology that employs the normal vector has a very good performance in inner points of the surface, but it could be non-applicable if $\vec{R}$ lies close to the boundary of the geometry; in this case, limits in the parameter space (x, y, z) may prevent the normal to $\mathbf{S}$ passing through $\vec{R}$ to be calculated.

For simplicity reasons, the scope of our point inversion study will be limited to untrimmed NURBS. The more general situation with trimmed NURBS could be solved similarly, but taking into consideration the forbidden regions of the surface, so that the parameters space will be reduced by the trimming curve expressed in the parameters space.

5.3 Problem solving

Point inversion problems do not generally have an analytic solution [45], and they are commonly solved with approximation methods.

The objective of this study is to position each and every point of the surface mesh $\vec{R}$ on the analytic surface $\mathbf{S}$ with its parametric coordinates (ξ, η) . In order to avoid diffi-

[†]The surface mesh point $\vec{Q}$ is the result of a meshing process that may involve some approximations.

5.3 Problem solving

culties close to the borders of the surface $\mathbf{S}$, the chosen approach will be the minimization of the distance between the mesh node $\vec{R}$ and the corresponding point $\vec{Q}$ on the analytic surface $\mathbf{S}$ which $\vec{R}$ supposedly belongs to.

As highlighted by Hu and Wallner [18], the main problems of the point inversion are:

- Initial guess estimation
- Speeding up the searching process

In our specific application where this point inversion is aimed at a geometry deformation for aerodynamic simulation, there are a specific set of drawbacks and advantages that will prioritize the selection of certain methodologies against others for the point inversion problem:

Drawbacks

- A complete optimization (minimization) process is required for each surface mesh point.
- Solution reliability in each mesh point is critical for the whole process chain. A single wrong result of the point inversion could lead to topology changes in the surface mesh and negative volume cells during volume mesh deformation.
- Difficulties to discern local and global minima result in a lack of confidence on the obtained result.

Advantages

- Extremely quick evaluation of the (x, y, z) position of a point $\vec{Q}$ on the surface from its parametric coordinates $\vec{Q} = \mathbf{S}(\xi, \eta)$.
- Very valuable information from the adjacent mesh points whose point inversion has been already performed.

The last two drawbacks are closely linked to each other; Neumaier [40] already highlighted that “*Global optimization is [...] much more difficult than convex programming or finding local minimizers of nonlinear programs, since the gap between the necessary (Karush-Kuhn-Tucker) conditions for optimality and known sufficient conditions for optimality is tremendous*”. Considering that there is not any single methodology that brings solution to this reliability problem, the selected approach is to *cross-validate* the obtained solution with different techniques. It will not provide the required evidence that the

obtained solution is a global minimum of the problem, but it will strongly reduce the probability of getting a wrong solution originated by an ill conditioned technique. The basis of this reasoning is that, in case of obtaining a spurious result $\vec{Q}_0^*$ with one technique, the probability of obtaining exactly the same wrong result $\vec{Q}_0^*$ with a completely independent methodology should be much smaller than the probability of obtaining any other result in general for this point inversion. Considering the impossibility to ensure reliability of the solution, the chosen approach is to substitute *validation* of the point inversion with the reduction of the error probability to the probability of *identical failure* between several independent techniques.

Although the point inversion methodology has been designed to obtain the right values, we accept the probability of obtaining a specific wrong result, denominated event E , whose probability is $P(E)$. Considering a point inversion problem, and a specific wrong result of it, $\vec{Q}_0^*$, so that the distance between $\vec{Q}_0^*$ and the real point $\vec{Q}$ is greater than the expected accuracy of the methodology, $d(\vec{Q}_0^*, \vec{Q}) > \text{Required accuracy}$; it can only be stated that $P(E) \leq 1$. Thus, if there are two independent optimization techniques 1 and 2, the probability of the event E happening with the two methodologies is:

$$P(E_1 \cap E_2) = P(E_1|E_2) \cdot P(E_2) = P(E_2|E_1) \cdot P(E_1) \quad (5.2)$$

But the two methodologies should be independent, $P(E_1|E_2) = P(E_1)$ and $P(E_2|E_1) = P(E_2)$, yielding:

$$P(E_1 \cap E_2) = P(E_1) \cdot P(E_2) \quad (5.3)$$

This methodology cannot confirm the correctness of a point inversion, but it can reduce its failure probability by running independent point inversions that can be used to cross check their results. In case of significant differences (greater than the expected accuracy) between two independent point inversions, we do not have a tool to decide which one is the right one, but based on repetitivity, we can estimate the reliability of the point inversion.

A root analysis of this methodology shows that its weakest point is the ability to ensure complete independence of the different point inversion techniques, because all of them will be strongly dependent on the initial guess point. In order to apply this statistical validation, we need to seed with different candidate guess point one for each point to be inverted. It will be shown later that the points obtained with different discretization of “grid” search §5.4.1 or points with connectivities in the surface mesh, if they have already been inverted, are good candidates.

5.4 Point inversion techniques: description

5.4.1 Grid search

This technique consists of a set of parametric sweeps along the two surface parameters (ξ, η) where the distance between $\vec{R}$ and each $\vec{Q} = \mathcal{S}(\xi_i, \eta_j)$ is evaluated, and the minimum distance is flagged in each iteration step so that the next iterations would take place with a reduced searching area in the surroundings of the previous flagged pair of parameters (ξ_i, η_j) . It is the simplest methodology and one of the most stable ones.

The implemented model is a simplification of the methodology described by Piegl and Tiller [45]. In our implementation, simple evaluations of the geometry on a parametric grid (ξ, η) were employed, in order to avoid calculation of the derivatives and the necessity to solve the linear problem with the Newton method.

The followed process is:

1. Obtain a structured and iso-spaced grid of pairs (ξ, η) , that sweep the whole range of validity of the parameters (**ngrid**+1 points in each parameter direction).
2. Evaluate the coordinates (x_i, y_i, z_i) of every $\vec{Q}_{ij}$ obtained from the pairs (ξ_i, η_j) .
3. Calculate the distance between $\vec{R}$ and every point $\vec{Q}_{ij}$.
4. Reduce the search space (accessible range of the parameters) around the pair (ξ_i, η_j) with the smallest distance to $\vec{R}$.
5. Ensure compatibility of the new search space with the allowable range of the parameters (ξ_i, η_j) . If the new region is not completely inside the allowed region, it must be reduced in order to fit into it.
6. Estimate the uncertainty of the obtained result as the distance in the space parameter between (ξ_i, η_j) and any of the surrounding points $(\xi_{i\pm 1}, \eta_{j\pm 1})$ that have been employed in the last iteration.
7. Repeat from the step 1 until uncertainty reaches the required value.

There are only two parameters that control this optimization algorithm:

Discretization level of the parameter space (ngrid) High discretization values will help to avoid stagnation in local minima due to extensive sampling, but they will increase as well the number of function evaluations.

Law that controls the parameter space reduction High search space reduction rates contribute to make the algorithm more unstable, while smaller reduction rates will slow down the convergence.

In order to simplify the algorithm, it was decided to consider that the relation between the dimensions of the search space in two consecutive iterations would be $\sqrt{\mathbf{ngrid}}$, which shows a good compromise in terms of convergence and stability.

Thus, the only driving parameter of this methodology is the level of discretization of the surface in each iteration step ($\mathbf{ngrid}$), so that the search region of surface will be sampled with $\mathbf{ngrid} + 1$ points in each direction ξ and η ; and the estimated accuracy is $\propto \mathbf{ngrid}^{-n/2}$, where n is the number of iterations.

The main advantage of this methodology is the continuous convergence for smooth surfaces, but it has a relatively high computational cost due to the multiple evaluations of the surface.

5.4.2 “Quad” search

This point inversion methodology is basically an evolution of the “grid” search §5.4.1, where the number of surface evaluations in each iteration is reduced to five.

This technique is composed by the following steps:

1. Evaluate the coordinates (x_i, y_i, z_i) of $\vec{Q}_{ij}$ in each of the pairs (ξ_i, η_j) located in the corners and the center of the search space.
2. Calculate the distance between these five points $\vec{Q}_{ij}$ to the searched point $\vec{R}$.
3. Modify the search space:

Reduce the search space if the point with the smallest distance is the centered one, the next iteration will have an accessible search space whose dimensions have been reduced by the factor $\mathbf{nconv}$.

Move the searching parameter space if the point with the smallest distance is in one of the corners of the search space.

4. Ensure compatibility of the new search space with the allowable range of the parameters (ξ_i, η_j) . If the new region is not completely inside the allowed region, it must be reduced in order to fit into it.

5.4 Point inversion techniques: description

5. Estimation of the uncertainty of the solution. Similarly to the “grid” search §5.4.1, the estimator of the uncertainty is the distance in the space parameter between the center of the searching space and its corners.
6. Repeat from the step 1 until uncertainty reaches the required value.

This methodology has difficulties in the presence of local minima, which may not be able to discern. The main benefit is the much quicker convergence than the “grid” search in areas of the surface that are seen as convex from the target point $\vec{R}$.

In order to avoid stagnation into local minima, this search algorithm needs to be initiated with a good candidate point around the global minimum in order to center the reduced search region. “Grid” search or any other methodology that provides an accurate enough initial guess could be employed; this last option is very useful in case there is already existing information from the adjacent point of the mesh, which can be considered as an initial guess.

5.4.3 “Quadratic” approximation search

This technique is the natural evolution from the “quad” search §5.4.2 through the introduction of an approximation to the sampled surface.

Labeling $\vec{Q}$ as the closest point on the surface $\mathbf{S}$ to the fixed point $\vec{R}$, and if $\vec{Q}$ does not lie on the surface edges, then the normal to the surface $\mathbf{S}$ on $\vec{Q}$ goes through $\vec{R}$. Using this property and considering the plane $\mathbf{\Pi}$ as the first order approximation to the surface $\mathbf{S}$ in $\vec{Q}$, and the point $\vec{Q}'$ each of the approximations to $\vec{Q}$ on the plane $\mathbf{\Pi}$, the function distance has the shape of a two-sheet hyperboloid.

$$d(\overline{Q'R})^2 = d(\overline{QR})^2 + d(\overline{Q'Q})^2 \quad (5.4)$$

Four points in the space could be used to define a bilinear surface, but this geometry would not be a good candidate because it can only approximate regions of a surface with hyperbolic points and all the points of a two-sheet hyperboloid are elliptic. Adding a center point as it was done in §5.4.2 gives valuable information about the surface and opens the possibility of defining a quadric surface with elliptic points.

Considering the generic case of a second order surface to be used as an approximation of the distance function $d(\overline{Q'R})$ between a fixed point $\vec{R}$ and any point $\vec{Q}' = \mathbf{S}(\xi, \eta)$ on the surface $\mathbf{S}$, the obvious candidate would be the hyperboloid, whose equation is as following:

$$z'^2 = f(x', y') = Ax'^2 + By'^2 + 2 \cdot Hx'y' + 2 \cdot Jx' + 2 \cdot Ky' + M \quad (5.5)$$

But obtaining the coefficients of the Eq.5.5 requires six evaluations of the distance to the surface and solving a non-linear system of six equations that considerably increases the computational cost.

Fortunately, there are some simplifications which show good computational results although they introduce some inaccuracy in the methodology:

- The behaviour of an elliptic paraboloid and the two-sheet hyperboloid is very similar in the surroundings of the vertex, and the equation of this elliptic paraboloid results in:

$$z' = f(x', y') = Ax'^2 + By'^2 + 2 \cdot Hx'y' + 2 \cdot Jx' + 2 \cdot Ky' + M \quad (5.6)$$

- If the cross term with x' and y' is eliminated, the equation looks much simpler:

$$z' = f(x', y') = Ax'^2 + By'^2 + 2 \cdot Jx' + 2 \cdot Ky' + M \quad (5.7)$$

- The points employed to evaluate the coefficients could be selected carefully: instead of the corners and center of the searching region employed in the “quad” search §5.4.2, the points that were on the corners in the “quad” search will be on the edges of the searching area, so that the five points form a cross aligned with the parametric lines of the surface.

After these simplifications the process is:

1. Evaluate the coordinates (x, y, z) in of the pairs (ξ_i, η_j) located in the center and edges of the search space $\vec{Q}_i$.
2. Calculate the distance between these five points $\vec{Q}_i$ to the searched point $\vec{R}$.
3. Obtain the coefficients of the distance function.

$$d = f(\xi, \eta) = A\xi^2 + B\eta^2 + 2 \cdot J\xi + 2 \cdot K\eta + M = 0 \quad (5.8)$$

4. Obtain the minimum distance according to the function $d = f(\xi, \eta)$ and its coordinates
5. Modify the search space:

Reduce the search space if the point with the minimum distance is within the searching region.

5.4 Point inversion techniques: description

Move the searching parameter space if the point with the minimum distance is out of the searching region.

6. Ensure compatibility of the new search space with the allowable range of the parameters (ξ_i, η_j) . If the new region is not completely inside the allowed region, it must be reduced in order to fit into it.
7. Estimation of the uncertainty of the solution.
8. Repeat from the step 1 until uncertainty reaches the required value.

5.4.4 First order Hu-Wallner algorithm

This algorithm, developed by Hu and Wallner [18], is an adaptation of the Newton method with a first order approximation to the surface. Thus, the problem is reduced to the obtention of the point $\vec{Q}$ on the plane Π with the minimum distance to a point $\vec{R}$ in the space (Fig.5.1).

Figure 5.1: 1st order Hu Wallner

Derivatives of the surface along ξ and η parameters yield the vectors $\vec{s}_\xi$ and $\vec{s}_\eta$, that define the tangent plane Π to the surface S in the guess point $\vec{Q}'$.

Normal projection of the point $\vec{R}$ on the plane Π gives the point $\vec{Q}''$ which should be a better estimation of the point $\vec{Q}$ with the minimum distance.

The vector $\overline{Q'Q''}$ indicates the increments in both parameters $(\Delta\xi, \Delta\eta)$:

$$\overline{Q'Q''} = \vec{s}_\xi \cdot \Delta\xi + \vec{s}_\eta \cdot \Delta\eta \quad (5.9)$$

The product of the equation Eq.5.9 by the vectors $\vec{s}_i$ ($i = \xi, \eta$) gives a system of equations with the increments $\Delta\xi$ and $\Delta\eta$.

$$\langle \vec{s}_\xi, \vec{s}_\xi \rangle \cdot \Delta\xi + \langle \vec{s}_\eta, \vec{s}_\xi \rangle \cdot \Delta\eta = \langle \overline{Q'Q''}, \vec{s}_\xi \rangle \quad (5.10)$$

$$\langle \vec{s}_\xi, \vec{s}_\eta \rangle \cdot \Delta\xi + \langle \vec{s}_\eta, \vec{s}_\eta \rangle \cdot \Delta\eta = \langle \overline{Q'Q''}, \vec{s}_\eta \rangle \quad (5.11)$$

This process can be repeated iteratively by renaming $\vec{Q}''$ as $\vec{Q}'$ until we achieve the expected uncertainty value in the length of $\overline{Q'Q''}$.

5.4.5 Second order Hu-Wallner algorithm

Hu and Wallner further improved the first order approximation (§5.4.4) by considering the curve $\mathbf{C}$ on the surface $\mathbf{S}$ that projects onto the vector $\overline{Q'Q''_1}$ on the plane $\mathbf{\Pi}$ as shown in Fig.5.2 [18].

Figure 5.2: 2nd order Hu Wallner

The first fundamental form of the surface can be expressed as $g_{ij} = \langle \vec{s}_i, \vec{s}_j \rangle$ with $i, j = \xi, \eta$, so that the normal vector to the surface is

$$\vec{n} = (\vec{s}_\xi \wedge \vec{s}_\eta) / \sqrt{\det(g_{ij})} \quad (5.12)$$

and the second fundamental form is $h_{ij} = \langle \vec{s}_{ij}, \vec{n} \rangle$ with $i, j = \xi, \eta$

Curvature of the surface along the direction $\overline{Q'Q''_1}$ can be obtained as:

$$\kappa_n = \frac{\sum_{i,j=\xi,\eta} h_{ij} \Delta u^i \Delta u^j}{\sum_{i,j=\xi,\eta} g_{ij} \Delta u^i \Delta u^j} \quad (5.13)$$

The curve $\mathbf{C}$ is contained into the plane $\mathbf{\Psi}$ which is normal to the tangent plane $\mathbf{\Pi}$ and passes through the points $\vec{R}$, $\vec{Q}'$ and $\vec{Q}''_1$.

5.4 Point inversion techniques: description

The Taylor series of the curve $\mathbf{C}$ parametrized as $c(\lambda)$, where λ is the arc length parameter of the curve, around the point $\vec{Q}'$ can be used as an approximation of the curve $\mathbf{C}$. Using this Taylor series to define the position of the point $\vec{Q}_2''$ which provides the second order solution for each iteration of the point inversion, yields to:

$$\vec{Q}_2'' = \vec{c}(\lambda_0) + \Delta\lambda \cdot \vec{c}'(\lambda_0) + \frac{\Delta\lambda^2}{2} \cdot \vec{c}''(\lambda_0) + 0(\Delta\lambda^2) \quad (5.14)$$

Where λ_0 is the arc length parameter of the point $\vec{Q}'$ and $\Delta\lambda$ is the distance along the arc between $\vec{Q}'$ and $\vec{Q}_2''$ in each iteration.

After neglecting the higher order terms, the curve results in a circumference arc with radius $1/\kappa_n$, so that $\vec{Q}_2''$ will be the point on the arc with minimum distance to the initial point $\vec{R}$.

$$\vec{Q}_2'' \approx \vec{Q}' + \Delta\lambda \cdot \vec{t} + \frac{\Delta\lambda^2}{2} \cdot \kappa_n \cdot \vec{n} \quad (5.15)$$

Where the vector $\vec{n}$ is the normal to the surface $\mathbf{S}$ in $\vec{Q}'$ and $\vec{t}$ is the unitary vector contained on the tangent plane $\mathbf{\Pi}$ and in the plane $\mathbf{\Psi}$, orthogonal to $\mathbf{\Pi}$, which passes through $\vec{R}$.

$$\vec{t} = \frac{\Delta u^\xi \cdot \vec{s}_\xi + \Delta u^\eta \cdot \vec{s}_\eta}{\|\Delta u^\xi \cdot \vec{s}_\xi + \Delta u^\eta \cdot \vec{s}_\eta\|} = \frac{\overline{Q'Q_1''}}{\|Q'Q_1''\|} \quad (5.16)$$

Reordering the expression Eq.5.15 yields to:

$$\vec{Q}_2'' - \vec{Q}' = \Delta\lambda \cdot \vec{t} + \frac{\Delta\lambda^2}{2} \cdot \kappa_n \cdot \vec{n} \quad (5.17)$$

And if it is multiplied with a cross product with $\vec{n}$

$$\overline{Q'Q_2''} \wedge \vec{n} = \Delta\lambda \cdot \vec{t} \wedge \vec{n} + \frac{\Delta\lambda^2}{2} \cdot \kappa_n \cdot \underbrace{\vec{n} \wedge \vec{n}}_{=0} \quad (5.18)$$

$\vec{t}$ and $\vec{n}$ are two unitary vectors and both terms of the Eq.5.18 have the same direction, thus the vector equation is reduced to the scalar equation:

$$\Delta\lambda = \|\overline{Q'Q_2''} \wedge \vec{n}\| \quad (5.19)$$

Once $\Delta\lambda$ is known, increments in the parameters (ξ, η) can be obtained

$$\Delta\xi = \Delta\lambda \cdot \Delta u^\xi \quad (5.20)$$

$$\Delta\eta = \Delta\lambda \cdot \Delta u^\eta \quad (5.21)$$

5.5 Point inversion techniques: evaluation

A testing benchmark composed by seven surfaces has been prepared for assessing the performance of the different techniques. These surfaces represent the most relevant casuistry that can be found in a point inversion problem.

The considered examples vary from simple surfaces with only elliptical points (the same sign in both principal curvatures) (Figs.5.3a and 5.3d) to saddle shaped surfaces with hyperbolic points (principal curvatures with opposite signs) (Fig.5.3g) or surfaces with parabolic points (one of the principal curvatures is zero) (Figs.5.3b, 5.3c, 5.3e and 5.3f). The following sections will confirm the relevance of the curvature in the efficiency of the different point inversion strategies and the suitability of the chosen validation surfaces.

The importance of curvature will be checked with the comparison of the point inversion results between Fig.5.3a and Fig.5.3d or between Fig.5.3b and Fig.5.3c which are topologically identical but with very different curvature values.

The main difficulty in the point inversion exercise will be found with self-intersecting surfaces. It is a well known problem in the CAD environment that frequently requires to identify the closest point of a curve or surface with respect to the cursor position. Any CAD software can easily overcome this problem if the cursor or the point to be inverted is far enough from the curve intersection, but it may face difficulties when the point is close to the intersection surface. CAD systems classically solve this issue by considering the movement of the cursor and the point inversion with respect to the previous positions. Fortunately industrial design problems does not face the full complexity of the mathematical problem, because self-intersecting surfaces may easily lead to non watertight 3D domains, which by definition are not allowed for most CFD tools. However, there is a closely similar problem if the surface rolls over itself and has two boundaries in a common edge (Fig.5.3f). This problem can be tackled easily as a optimization problem with local minima, and it will be checked that the difficulties can be avoided by a clever selection of the initial guess point or by splitting the surface in two.

Point inversion problem is in fact an optimization problem that searches the minimum distance, so there is always a risk of not being able to reach the global optimum, and two stopping criteria were implemented:

Uncertainty Optimization will stop if the estimated error and the distance between two consecutive iterations with values (η_i, ξ_i) and (η_{i+1}, ξ_{i+1}) is less than 10^{-6} .

Maximum number of iterations It contributes to avoid infinite loops and it has been fixed with 5000 iterations.

5.5 Point inversion techniques: evaluation

Figure 5.3: Benchmark of validation surfaces

It must be noted that the uncertainty criteria works only as an stopping criteria. The real error of the estimation will be later evaluated against the real value that was searched.

Time measurement combined with the achieved accuracy could be a good indicator of the methodology performance; however differences between CPU clock cycles in different evaluations of the point inversion could show a small problem due to the high speed of a single call. This difficulty can be overtaken by unnecessary repetitions of each point inversion, so that external factors related to the operating system, compiler optimization or access to storage could be diminished. However, obtained results reminded that each point inversion is an iterative algorithm that takes itself most of the computational cost, so that the time difference between a single point inversion and the mean time taken by each point inversion after several repetitions is below 5%. Thus, in conclusion, time measurements could be roughly considered independent from external factors, and only related to the methodology.

Error bars in the following plots (time or number of iterations) represent minimum and maximum experimental values.

5.5.1 Grid search

It can be easily seen that required point inversion time scales linearly with the number of surface evaluations (Fig.5.4) and although higher **ngrid** values contribute to reduce the number of iterations (Fig.5.5), the number of surface evaluations which is linked to the square of **ngrid** + 1 predominates in the computation time (Fig.5.6).

According to the tests, the best estimator of the uncertainty is not the difference between consecutive iterations, but the grid size of the last iteration which reduced the search space.

A preliminary analysis of the convergence and its relation with the parameter **ngrid** in the set of testing surfaces, shows that this methodology poses only some difficulties in two cases:

1. Low **ngrid** value that leads to a sparse discretization and a reduction of the search space around a wrong set of parameters.
2. Closed or almost closed surfaces with two completely different set of parameters for a couple of extremely close points in the 3D shape (Fig5.3e).

The first difficulty can be overcome by setting an appropriate **ngrid** level. Values of **ngrid** around 20 can be employed for smooth surfaces with a relatively uniform parame-

5.5 Point inversion techniques: evaluation

Figure 5.4: Relation between point inversion time and the number of surface evaluations. Time has been scaled $\times 100$ with repetitions of each point inversion in order to reduce the influence of external factors. Error bars consider the experimental range of variation.

Figure 5.5: Relation between **ngrid** and the required number of iterations

terization, without an excessive computational cost. It must be noted that, as shown in Fig.5.6, the reduction of the number of iterations cannot compensate the extra computational cost per iteration so that of doubling **ngrid** value results in much more than double total computational cost. This behavior is explained by the stopping criteria which, with high values of **ngrid**, is triggered with an almost constant number of iterations.

All these results lead to the relatively expected conclusions:

Figure 5.6: Relation between `ngrid` and the total point inversion time

- The computational cost is a linear function with the number of evaluations of the surface function (Fig.5.4).
- The number of employed grid points is a key parameter in the first iterations, when it may have the ability to discern global minimum from any local minima.
- Increasing the grid discretization leads to quadratic increments in the number of grid points and by the same reason to quadratic increments in the computational cost per iteration which cannot be compensated by the smaller number of iterations (Fig.5.6).
- Further reduction of the uncertainty in the point inversion is achieved much more efficiently with more iterations (linear response in computational cost) than any increase in the grid discretization.

Fig.5.7 shows the compared behavior of this methodology for a set of different values of `ngrid`. It must be noted that the number of iterations in each point inversion is not fixed, but defined by the convergence criteria.

Figure 5.7: Observed error with expected accuracy of 10^{-5}

5.5.2 “Quad” search

This methodology is considerably faster than the grid search, but it is much more unstable as well. “quad” search methodology exhibits a strong sensitivity with respect to the initial guess point, because it anchors easily to any local minimum.

An initial grid search will be needed to overcome this inherent difficulty of the methodology. Following the results from §5.5.1, a single step iteration of the grid search with `ngrid` equal to 20 is employed to discard local minima and to center the “quad” search around the global minimum. Once the initial local minima have been discarded, the method shows convergence for all the test cases.

The uncertainty stopping criteria was adapted with a similar approach to the grid search technique. The estimator of the uncertainty in this case is not the difference between consecutive iterations, but the size of the searching space in the last iteration that lead to a reduction in the search space.

The chosen uncertainty estimator considers that the searched point should be inside the searching region, which shows reasonable results with some exceptions (§5.5.2 and Fig.5.9a 5.9b).

The only parameter that can influence the behavior of this methodology is the law that drives the reduction of the searching area. This parameter (**nconv**) is the shrinking factor applied to the grid size in two consecutive iterations, so that the maximum achievable convergence is given by Eq.5.22. It must be noted that this maximum convergence can only be achieved when successive candidate points are interior to the searching space and not on its border, which would require only a displacement of the searching space without any further reduction.

$$\Delta\xi, \Delta\eta \geq \text{initial_grid_size} \cdot \frac{1}{\text{nconv}^{\text{iterations}}} \quad (5.22)$$

A detailed analysis of the required number of iterations and the error for each **nconv** value (Fig.5.8) shows that mean error slowly decreases with increasing values of **nconv** up to 1.5, but its variability starts growing from 1.4 onwards. The analysis of the number of iterations shows a similar change in trend between 1.4 and 1.5, with a small reduction in the number of iterations but a greater variability. These behaviors justify the selection of **nconv** value of 1.4 in order to maximize convergence without compromising the stability of the methodology. This plot displays simultaneously mean values of error and number of iterations, and their variability (min and max) in a band and error bars respectively.

Values of **nconv** below 1.2 lead to an growing number of iterations because of the extremely low reduction rate of the search space; however, higher values of **nconv** result in more iterations where the method requires to move the searching space without any reduction of it and a more unstable behaviour as well.

The importance of the smoothness of the surface and its parameterization can be seen in Fig.5.9a and Fig.5.9b, where the second one shows a stronger sensibility to the parameters. Although the intuitive estimation of the uncertainty would consider that the searched point should be inside the searching region, preliminary results indicate that this methodology strongly underestimated this uncertainty in some surfaces.

5.5.3 “Quadratic” search

This methodology has the same number of surface evaluations per iteration than the “quad” search (§5.5.2) but the improvement is achieved with a smaller number of iterations without an excessive extra computational cost.

5.5 Point inversion techniques: evaluation

Figure 5.8: Influence of n_{conv} in the error and number of iterations for all the surfaces except surface 1 in the quad search. Green error bars represent minimum and maximum number of iterations experienced during the execution of the benchmark. Purple band extends from the lowest to the highest observed error in the point inversion benchmark.

Figure 5.9: Influence of n_{conv} in the error

Experimental results show a similar behavior to the “quad” search methodology but with a much smaller dispersion around the theoretical convergence (1×10^{-5}) and with considerably fewer iterations (see Fig.5.10). The dispersion of the number of iterations shows that increasing values of n_{conv} , progressively increase the observed maximum number of iterations.

Convergence in this methodology is almost independent of the `nconv` parameter because the size of the searching region in one step is given by the distance between consecutive candidate points; only in the case that the candidate point lies very close to the border or even outside of the searching region, `nconv` is employed to reduce the size of the next searching region. This special control close to the borders strongly contribute to increase the convergence rate.

A detailed analysis of the real and estimated error of the different point inversions as depicted in Fig.5.11 shows that:

- Cluster of points in the top left quadrant correspond to wrongly inverted points in surface 6 (Fig. 5.3f). Those points lie relatively close to the boundary of the surface, and due to an ill conditioned initial guess point or region it is not possible to make them converge to the real solution.
- All points in the top right quadrant are obtained with point inversion in surface 1 (Fig. 5.3a) which have reached maximum number of iterations (5000) with `nconv` values of 1.2 and 1.3; with the exception of two points whose high value of `nconv` resulted in a lack of convergence in the point inversion. This is a well known behaviour, where surfaces with higher curvature are more sensitive to the initial guess point.

Despite the stiffness of the convergence with `nconv`, this analysis suggests that values of `nconv` around 1.4, are a good compromise between stability (`nconv` > 1.3) and possible increase in the number of iterations.

5.5.4 Hu-Wallner 1st order search

This methodology built on the Hu Wallner description [18] was improved with an special handling of candidate point close to the borders. It follows the same approach as the “quad” methodology §5.5.3, and reduction of search space is given by the displacement of the candidate point, with `nconv` being employed only to improve convergence in the borders.

Similarly to previous methodologies, all point inversion errors (cluster of points in the top right corner in Fig.5.12) correspond to surface 6; which is a clear indicator that the problem is linked to the employed initial guess and the shape of the surface.

Figure 5.12 shows the low influence of `nconv`, which affects only the evaluation of the estimated error.

Figure 5.10: Influence of `nconv` in the error and number of iterations of all the properly converged cases in the quadratic search. Green error bars represent minimum and maximum number of iterations experienced during the execution of the benchmark. Purple band extends from the lowest to the highest observed error in the point inversion benchmark.

5.5.5 Hu-Wallner 2nd order search

In the same way that §5.5.3, §5.5.4 and §5.5.5, the relevance of `nconv` is marginal.

Due to the already low number of iterations required for the first order approach, no real improvement is provided by this second order implementation, resulting in identical number of iterations or even higher in some cases.

Figure 5.13 shows that the effort of the second order approach may slightly increase the accuracy with the same number of iterations, but it indicates a higher risk of uncorrelation between the estimated and real error.

Figure 5.11: Quadratic search: influence of `nconv` in the error Figure 5.12: Hu-Wallner 1st order: Influence of `nconv` in the error

Figure 5.13: Hu-Wallner 2nd order: Influence of `nconv` in the error

6

Dimensionality reduction

Contents

6.1	Analyzing dimensionality	86
6.1.1	Testing setup: curve fitting	86
6.1.2	Increasing the problem dimensionality	88
6.2	NURBS weight in the optimization	92
6.2.1	The current role of the weight of the control points in a NURBS geometry	92
6.2.2	Recovering the role of the weighted control points	93
6.2.3	Using weight values to eliminate control points	97
6.3	Analytical constraints can further reduce the dimensionality	99
6.3.1	Imposing $\mathcal{C}^0$ continuity with a fixed geometry	99
6.3.2	Imposing $\mathcal{C}^1$ continuity with a fixed geometry	101

All the previous chapters have shown how to apply adjoint optimization on a problem defined by geometry parameters, however, most of industrial geometries are defined mathematically by a huge number of parameters. Other optimization approaches like FFD require a manual preprocessing, where the FFD boxes are created with a reduced number of design variables. In our case, we have gained flexibility and automation by the use of the real geometry parameters, but we will review the problems originated by this high dimensionality and we will explore some techniques for reducing the high dimensionality while maintaining enough design flexibility through the design variables.

6.1 Analyzing dimensionality

6.1.1 Testing setup: curve fitting

In order to assess the different effects of the parametrization on the optimization process, a simple experimental setup has been prepared.

Although a classical objective in an aerodynamic optimization process is usually linked to lift or drag evaluations, in this case, the optimization process has been simplified and stripped off in order to ease the assessment of the different effects and to isolate the geometrical problem from other issues with origin in the aerodynamic simulation tools. The CAD fitting by deforming an initial geometry in order to match a given target curve with the specified constraints is the simplest problem that can model more complex behavior like the deformation of a geometry with the surface sensitivity provided by the adjoint simulation.

In this work, the target curve is the upper side of a NACA 0012 profile and a straight NURBS on the center line of the complete airfoil will be the initial curve to be deformed.

The purpose of the objective function is to quantify the goodness of the fitting between the deformed geometry and the target one. Although least-squares technique is often employed for fitting scattered data [28], we have used the area between the curves. This methodology, that evaluates the area between the target curve and the deformed one, has the advantage of not requiring the complete sampling of the target curve nor calculating distance between each of the sampled points and the deformed geometry.

The evaluation of the area between two curves ($\mathbf{C}_A$ and $\mathbf{C}_B$)[†] follows a three main steps process:

[†]Both curves are considered to have normalized knotvectors $\xi \in [0, 1]$

Step 1: The parametrization of the curves is checked, so that the parameter in both curves moves in the same direction. This initial test is performed by measuring distances between points with the same parameter in each curve. Considered test cases showed that sampling with 5 points ($t = 4$) is enough for simple geometries.

$$\text{Revert_flag} = \frac{\sum_{i=0}^t \left| \mathbf{C}_A \left(\frac{i}{t} \right) - \mathbf{C}_B \left(\frac{i}{t} \right) \right|}{\sum_{i=0}^t \left| \mathbf{C}_A \left(\frac{i}{t} \right) - \mathbf{C}_B \left(1 - \frac{i}{t} \right) \right|} ; \quad t \in \mathbb{N} \quad (6.1)$$

If **Revert_flag** is smaller than 1, one of the curves have to revert its parametrization.

Step 2: The curves are split at each intersection point between $\mathbf{C}_A$ and $\mathbf{C}_B$.

Figure 6.1: Area between curves

Step 3: The total area between the curves (A_{TOTAL}) is obtained as the summatory of the ruled surface area (A_i) whose guides are the pair of curves $\mathbf{C}_{A_i}$ and $\mathbf{C}_{B_i}$ between intersections (Fig.6.1).

$$A_{TOTAL} = \sum A_i \quad (6.2)$$

This objective function can be easily generalized for surface fitting; so that area between curves become volume between surfaces and intersection between curves become intersection between surfaces.

The optimization solver employed is NLOPT [38] with local derivative-free constrained optimization by linear approximation (LN_COBYLA[†] method) with two stopping criteria: convergence given by relative or absolute tolerances and maximum number of iterations which was set proportional to the number of optimizing variables.

[†]The employed COBYLA (Constrained Optimization BY Linear Approximations) algorithm is based on Powell's implementation [48, 49], with small adaptations to incorporate new termination criteria, to improve the support for bound constraints, to increase convergence speed by extending the trust-region radius under certain circumstances, and to better support different parameters with very different scales.

6.1 Analyzing dimensionality

An adaptation of the employed optimization algorithm could have been considered for a better convergence or avoidance of local minima in these particular problems. However, these issues help to depict the real situation where industrial problems have unknown solutions, local minima appear quite often and the best displacement vector to initiate the optimization is not generally obvious. Thus, such an imperfect optimization methodology is a good candidate to validate the interest of the findings that may help the industrial designer to find better solutions without the need to develop *ad hoc* adaptations and fine-tuning of the methodology for each specific problem.

6.1.2 Increasing the problem dimensionality

So far, there was no drawback in the “explode number control points” *approach* that simplifies the typical geometry operations without a noticeable loss of accuracy. The problem appears when the result of this excess of information needs to be the input of an optimization problem. We must need to pay the price of the apparently free complexity with a high dimensionality optimization.

The problem of the high dimensionality optimization can be seen from two perspectives:

- Costly evaluation of the gradient whose computational cost in finite differences is proportional to the number of variables.
- Undesired and noisy results due to the existence of spurious variables.

While the first problem may be avoided by the application of adjoint optimization, which only needs to solve the direct and the adjoint problem to obtain the gradient of all the variables, whichever their number; the second one is inherent to the chosen description of the problem, and as we increase the number of control points, it will lead to lower and lower convergence levels and higher possibilities of numerical noise leading to ripples and undesired changes in curvature.

Figure 6.2 shows the result of a typical optimization environment. The optimization example that has been chosen consists on modifying the vertical position of the control points[†] of the **blue** curve, from an starting position given by the **red** curve, in order to fit the target **green** curve, considering that both extremes are clamped. The chosen fitting function is the area between the two curves. The optimization solver employed is

[†]We must note that perfect fitting will be impossible due to the nature of the optimization variables which do not allow vertical tangent in one of the extremes of the curve.

NLOPT [38] with local derivative free constrained optimization by linear approximation (LN_COBYLA method) with two stopping criteria:

Convergence given by relative or absolute tolerances

Maximum number of iterations proportional to the number of optimizing variables.

The frequent high dimensionality of the geometry derives from the need to obtain a specific shape within a given tolerance. Although geometry operations are very efficient, CFD simulations linked to this geometry are extremely costly, and convergence of an optimization problem is commonly linked to the number of variables. An increase in the number of variables results in more iterations needed and a higher risk to stagnate in a local minimum.

The optimization example that has been chosen and represented in Fig.6.2 consists on a typical curve fitting problem with modification of the vertical position of the control points while extremes of the curve are clamped. The initial **red** curve is deformed (**blue** curve), in order to match the target **green** curve.

It must be noted that perfect fitting will be impossible due to the nature of the optimization variables which do not allow vertical tangent in one of the extremes of the curve.

The results presented in Fig.6.2, show an increasing level of undesired waviness in the extremes of the curve as the number of control points grows. This effect can be seen in the curve itself or by the alternate position of the control points above and below the curve. Although there are smoothing techniques that can overcome this problem, they would not remove the source of the problem, which is the topic of interest in this analysis.

Adding new control points is equivalent to adding new variables for describing the geometry, thus new families of geometries become accessible from the new geometry description. The effect of adding new control points becomes evident with the following example: a NURBS geometry defined by only two control points can only be a straight segment, but after adding an extra control point to the curve, results in a geometry that could be any second order curve, while the first order curve (segment) remains accessible as a particular case.

It is clear that a geometry defined through a bigger set of control points is prone to show bad behaviors like waviness (Fig.6.2b-6.2e), and it may suffer a strong loss of convergence speed as the number of variables is progressively increased (Fig.6.3).

The convergence plot in Fig.6.3 shows an *a priori* unexpected result with slower convergence with an increasing number of control points, except for the *Number of control*

6.1 Analyzing dimensionality

Target curve: Green curve with tiny control points

Initial curve: Red curve with small control points

Deformed curve: Blue curve with medium size squared control points

Case	Iterations	Objective function
Initial curve (red color)	0	64.759
Reference curve (Fig.6.2a)	188	1,573
Number of control points x2 (Fig.6.2b)	506	1,414
Number of control points x4 (Fig.6.2c)	1143	1,352
Number of control points x8 (Fig.6.2d)	2003	0,901
Number of control points x16 (Fig.6.2e)	3990	1,334

Figure 6.2: Fitting comparison between the reference case (red curve) and a set of identical geometries defined with a growing number of control points.

points x8 (Fig.6.2d) which initially follows the same tendency but it achieves a much better result of the objective function. This is the real problem of the dimensionality: it allows better fitting to a certain geometry in the same way that it increases the risk of the undesired waviness.

Firstly it should be noticed that, in the test case, the reference geometry does not allow (by definition) a perfect match of the target curve, due to its slope in the left border. The deformed curve would need to change its shape from horizontal tangent to vertical tangent in this extreme, which is not possible if the horizontal coordinate of the second control point remains unchanged. As the number of control points is increased, the horizontal distance between them is reduced and thus increases in the slope in the left border become possible. According to this analysis, any increase in the discretization should yield to a better objective function, but the different results between *Number of*

Figure 6.3: Convergence of the optimization with different number of variables

control points x8 and *Number of control points x16* disagree. This unexpected convergence with *Number of control points x8* that is not reached in *Number of control points x16* is linked to the strong waviness extending from the extremes up to one quarter of the chord, which acts against the improvements in the objective function. These waves generate many local minima where local optimization algorithms (like LN_COBYLA) find difficult to escape from.

If non-independent variables are introduced into the optimization process, we will define a problem prone to show bad behaviors (waviness, convergence...), because the first task of the optimizer should not be looking for the optimum, but looking for the relevant and independent set of variables. Looking at the convergence of the different test cases (Fig: 6.3), there is a strong loss of convergence speed as we further increase the number of variables.

6.2 NURBS weight in the optimization

6.2.1 The current role of the weight of the control points in a NURBS geometry

The importance of the weight has been already highlighted in [26] and an extensive analysis of the particular case of conics has been covered by [4, 11]. The example of the conics is self-explanatory: *There is no NURBS curve —whatever the number of control points— which can define precisely a circumference if all their control point weights are equal.*

If we come back to the industrial design problem, we observe that most of the employed geometries are de-facto NURBS curves and surfaces with identical weight in every control point. Although any commercial software is perfectly able to accept NURBS geometries, simplicity in their operations and the cheap cost of an increasing number of control points led to huge discretization even in low complexity geometries.

The most simple example to support this facts is:

1. Create a revolving cone
2. Intersect the cone with a plane (any plane)
3. Analyze the obtained curve (number of control points employed to define it)

Projective geometry [50] shows that the obtained curve is a conic, which should be undoubtedly defined with only 5 passing points with their weight coordinates. In other words, if we discover that the new curve has been created with more than 100 points, we would perfectly understand that the new curve is not the analytical solution but an approximation to it.

Many commercial CAD tools opted for omitting the weight of the control points in order to simplify some geometrical operations although an increase in the number of control points was needed for maintaining acceptable accuracy levels. However, this artificial increase in the number of variables employed to define a geometry has an adverse effect in geometry optimization processes, as highlighted for the adjoint optimization by Schmidt [59] and Jaworsky [23] *“this further increase in the number of degrees of freedom in the design problem can lead to oscillatory shapes.[22]”*

6.2.2 Recovering the role of the weighted control points

The approach of including the weight as another coordinate is not innovative and was already considered by Blanc and Farin [4, 11]. However, the modification of the initially unused weights brings an extra value in an optimization process in opposition to the classical *iso-weight* geometry definition.

The first problem to be considered is the optimization of a NURBS curve clamped in its extremes and defined with very few control points and with identical and fixed weights for all its control points (the most common situation). Then, some weights are allowed to change: firstly only the weight of one control point (the one closer to the left border) and finally, all weights are set free, except both of the extreme points whose weight is fixed and set equal to 1.

The inclusion of the weights in all the control points does not pose a computational challenge, as it implies only an increase in the number of variables between 50% and 33% for 2D and 3D cases respectively. However, the real price for allowing different weights is the loss of the B-splines simplicity explained in [45, 12], and the dimensionality increase.

As already explained in section §6.1, this parametrization will never be able to shape a geometry with a vertical tangent in the left border whilst the first and second control points have different horizontal coordinates. Although differences are hard to precise between Fig.6.4a and Fig.6.4c, the table of weight values indicates that, adding the weight variable of the second control point results in a 23% faster convergence, despite of the increase in the number of variables, and an improvement of 12% in the objective function in the cases with only y coordinates free. Similar results could be obtained in the case with x and y coordinates free, where convergence is 40% faster and whose objective function drops around 7%. Therefore, the relevance on the weights on the most complex regions of the curve are justified by the obtained results that return a weight of 2,662 and 1,742 for the second control point with only y variables free or x and y variables free respectively, while the rest of the weight values are fixed to 1. These results show that considering weight variables in the regions with bigger discrepancies between the target curve and the obtained one may contribute to a better shaping of the geometry.

Looking at the convergence from Fig.6.6, it can be observed that adding the weights of more and more control points to the optimization problem influences negatively the convergence. This effect is related to the dimensionality of the problem section §6.1. However, there is a clear tendency that freeing the weight of a few points in the most problematic region strongly improves the convergence and the value of the final objective function.

6.2 NURBS weight in the optimization

Case	Iter.	Objective function	w_1	w_2	w_3	w_4	w_5
Fig.6.4a	253	1,546					
Fig.6.4c	195	1,363	2,662				
Fig.6.4e	417	1,347	2,084	2,096	1,560	2,189	2,176
Fig.6.4b	462	1,259					
Fig.6.4d	275	1,175	1,742				
Fig.6.4f	1500	1,056	1,513	1,314	2,030	1,933	0,027

Figure 6.4: Comparison of different solutions with and without weights

Numerical values displayed in Fig.6.4 shows interesting results regarding the weights distribution for all the control points. While in the test case with only y and $weight$ as free coordinates, all $weights$ have the same order, the test case with all the coordinates free (x , y and $weight$) presents a set of $weights$ with all values above 1 except the last one (sixth control point) that reached a weight of 0,027, which is far smaller than the weight of any other control point. This extremely low weight justifies the unexpected location of the control point (Fig.6.4f), which is relatively far from the curve and forces a wrong mathematical slope in the right border. The weight minimizes the influence of this control point so that in the limit case, zero value of weight would indicate a control point with no influence at all in the geometry, and thus a control point which could be deleted without affecting the geometry.

In such a way, this extremely low weight value is an indicator that this region of the curve could be currently over-defined; thus no big penalties could be expected if we completely remove this control point from the initial curve to be optimized. However, this test case result is not as good as expected (Fig.6.5a), its convergence is much worse and the objective function reaches 1,767, around 70% higher than the original situation before removing any control point. This counter-intuitive results emerge from the behavior of the optimization, whose final result is strongly dependent on the initial guess, which in this case makes the the optimizer unable to leave a local minimum; as can be clearly seen because the x coordinate of the second control point has been displaced to the right, in the wrong direction, and even with an apparent accurate fitting in the right part of the curve, the total objective function is much higher.

Case	Iter.	Objective function	w_1	w_2	w_3	w_4	w_5
Fig.6.4f	1500	1,056	1,513	1,314	2,030	1,933	0,027
Fig.6.5a	214	1,767	2,755	1,942	1,842	2,043	
Fig.6.5b	900	1,077	1,537	1,372		1,664	

Figure 6.5: Results of removing control points where the weights have been tagged smaller than the others

Recovering the results of section §6.1, further reduction of the dimensionality should lead to less chances to get trapped into local minima. Although the case with 1 control point removed does not show big differences in the weight values, the process can be repeated and the control point with the lowest weight ($w_3 = 1,842$) is removed again; the obtained solution reaches now the expected benefit of a lower dimensionality as it can be seen in Fig.6.6, where with 40% less iterations than the initial reference we reach an objective function of 1,077 which is only 2% higher than the reference case 6.4f. If the optimization continued until we reach the same number of iterations (1500 iterations) a total value of the objective function of 0,925 would be obtained.

Looking at the convergence plots in Fig.6.3 and Fig.6.6, three important results are obtained. Firstly, any reduction in the number of control points yields to a natural improvement in the first steps of the convergence due to the lower dimensionality (note the logarithmic scale). Secondly, there is always a risk of unexpected results linked to

6.2 NURBS weight in the optimization

No weight:	Only y coordinates free and no weight at all (Fig.6.4a)
1 weight:	Only y coordinates free and weight in the second control point (Fig.6.4c)
All weight:	Only y coordinates free and weights in each control point (Fig.6.4e)
No weight x:	x and y coordinates free and no weight at all (Fig.6.4b)
1 weight x:	x and y coordinates free and weight in the second control point (Fig.6.4d)
All weight x:	x and y coordinates free and weights in each control point (Fig.6.4f)
All weight x 1point deleted:	x and y coordinates free and weights in each control point after removing last control point (Fig.6.5a)
All weight x 2point deleted:	x and y coordinates free and weights in each control point after removing the last and third from last control points (Fig.6.5b)

Figure 6.6: Convergence in reference test cases and after removing the control points whith the lowest weights

the heuristic character of the optimization that does not ensure a global minimum, as it happened in the case where a single control point had been removed. And in the last place, after the stabilization of the test case with two control points removed, its convergence rate is the highest one from all the test cases. This effect is an indicator that a reduction of the number of variables contributes to a better exploration of the design space leading to less local minima with the same optimization algorithm.

6.2.3 Using weight values to eliminate control points

The methodology of removing the control point with the smallest weight can be applied recursively as shown in Fig.6.7, and it results in a better objective function with a smaller number of iterations.

In order to take advantage of this approach, the starting geometry in each case is not the straight line employed up to now, but the NURBS geometry after removing the control point with the smallest weight in each recursion stage.

Figure 6.7 displays the convergence of the fitting curve in successive steps. This convergence plot has a characteristic shape with continuous descent of the objective function up to a minimum value; if the removal of control points goes too far, the modified curve is no longer capable to follow the shape of the target curve and the objective function increases.

In Figure 6.8 it is shown how the reduction in the number of control points contributes to eliminate the waviness caused by the high dimensionality (Fig.6.8a). Despite the clustering of control points in the right border (see Fig.6.8b), the oscillations level has almost dissapeared (Fig. 6.8c-6.8d).

Figure 6.7: Results after successive removals of the control point with the lowest weight

All these results confirm the potential of reincorporating the weight in the geometry description, and the possibility to interpret weight values as an indicator of the points

(a) Reference (49 control points)

(b) Reference after removal of 35 control points

(c) Zoom reference

(d) Zoom after removal of 35 control points

Test case	Objective function	Iterations
Reference with 49 control points (Fig.6.8a, 6.8c):	0,9013	2003
Reference after removal of 35 control points (Fig.6.8b, 6.8d):	0,2231	236

Figure 6.8: Geometry simplification with control point removal based on weights

that can be removed in the geometry simplification, in order to avoid spurious shapes and the excess of optimizing variables.

Although the optimization algorithm and initial curve seem not to be well suited for these test cases because they never reach the right shape in the left part of the curve and they look overdefined in its right border; it has been prepared on purpose. Typical industrial problems with gradient optimization algorithm are also strongly dependent on the initial guess which may drive the optimization towards specific designs.

6.3 Analytical constraints can further reduce the dimensionality

After the reviewing the potential of the control point weights for reducing the dimensionality of the design problem, and following the analysis of §4.2 we will analyze the effect of the weight in the dimensionality reduction by application of boundary conditions.

It is known that weight redistribution effects extend to the complete geometry (Fig.4.1), but in some cases, the modification of the weight close to a boundary condition may have a beneficial effect, helping to achieve the required continuity in a similar way than the tension.

Although only a few cases could be interpreted easily from the geometrical perspective, they are very useful to understand the relations between two NURBS surfaces that should maintain $\mathcal{C}^n$. The real interest of these analytical constraints is their ability to reduce the parametric space in the most common situation[†].

The selected test will employ the same optimization methodology as §6.2.2 for deforming the initial **red** curve so that the new geometry contacts the constraint **cyan** curve while trying to minimize the distance between the **green** curve and the target **blue** curve.

6.3.1 Imposing $\mathcal{C}^0$ continuity with a fixed geometry

The definition of this problem includes an automatic variable reduction thanks to the analytical constraint. In this case, imposing $\mathcal{C}^0$ in one of the extremes of the curve can be done easily by reducing the two or three spatial coordinates (2D or 3D respectively) of the first control point to a single position parameter along the given curve that acts as a constraint.

In the convergence plot Fig. 6.9 we can observe the importance of employing the most relevant variables and not all the possible ones. This figure shows that adding x coordinates to the variables has a relatively high price in terms of speed of convergence due to the increase in the number of variables; but however this higher dimensionality of the problem does not result in a better solution because it leads the optimizer to a local minimum which it is not able to escape.

[†]We must note that high order surfaces are not desired from the computational point of view due to the tendency to generate wrinkles, and only in a very few conditions $\mathcal{C}^3$ will be required, because in most cases $\mathcal{C}^2$ is enough to satisfy aerodynamic requirements.

6.3 Analytical constraints can further reduce the dimensionality

No weight:	Only y variable and no weight at all
1 weight:	Only y variable and weight in the second control point (the one not lying on the constrain curve)
All weight:	Only y variable and weights in each control point
No weight x:	x and y variable and no weight at all
1 weight x:	x and y variable and weight in the second control point (the one not lying on the constrain curve)
All weight x:	x and y variable and weights in each control point

Figure 6.9: Optimization convergence with different sets of variables after imposing $\mathcal{C}^0$ constraint

Nonetheless, the introduction of weight variables made possible a higher final convergence level in both cases, with and without x variables. This result reinforces the idea that adding weight variables is a good idea, while adding variables related to the non-privileged direction of the model (x) could have a negative impact.

The plots in Fig. 6.10 show the obtained results after each of the optimization cases. They show very similar solution with a tendency to merge the last two control points when their x coordinate has been set as free (a result already discussed in 6.3??) In this cases the indicator is not the weight of the control point, which remains with the same order as all the others, but the proximity of the two control point themselves. Two close control points are only required if a strong shape change is required in the surroundings of the control points.

Figure 6.10: Comparison of different solutions of $\mathcal{C}^0$ constrained curve with and without weights

6.3.2 Imposing $\mathcal{C}^1$ continuity with a fixed geometry

This test case has been prepared with identical methodology as the previous one. In this case, analytical $\mathcal{C}^1$ was imposed by automatic reduction of the variables, where the variables of the first two control points are fixed by the two parameters that define the position along the constraint curve and a variable equivalent to the *tension* in the tangent direction.

The results in the convergence plot Fig. 6.11 show the best result when the weight has been considered in a relevant point, and the number of optimization variables has been kept low (Fig. 6.12c); the next two best results are the cases with weight variables but a reduced dimensionality, so that the worst convergence is achieved in the reference case and in the ones without weights or with too many variables. Again, everything tends to indicate that dimensionality reduction is a must in order to achieve good results in the optimizer, and that weight variables are extremely relevant.

No_weight:	Only y variable and no weight at all
1_weight:	Only y variable and weight in the second control point (the one not lying on the constrain curve, but in its tangent)
All_weight:	Only y variable and weights in each control point
No_weight_x:	x and y variable and no weight at all
1_weight_x:	x and y variable and weight in the second control point (the one not lying on the constrain curve, but in its tangent)
All_weight_x:	x and y variable and weights in each control point

Figure 6.11: Optimization convergence with different sets of variables after imposing $\mathcal{C}^1$ constraint

Figure 6.12: Comparison of different solutions of $\mathcal{C}^1$ constrained curve with and without weights

Conclusions

Keeping in mind the target of the industrial application of all these developments, it make sense to take one step backwards and to analyze the achieved progress and the possible future activites to support the full industrialization of the methodology.

Achievements

The starting point of this work substantiated from the current state of the art in the industrial aerodynamic design. Although it is true that optimization is commonly requested, it is nonetheless true that current CFD processes are often not suitable for the most advanced optimization techniques. Designers frequently resort to simplified processes like [Surrogated models + evolutionary algorithms] or [Simplified shapes + gradient base algorithms], because the real problem is either to costly (computation time) or to complex to be automated (human time). This simplification leads to valuable knowledge about trends, but final design relies on the designer ability to implement the features identified with the trends.

A detailed analysis of the situation allowed us to identify the process bricks and its bottlenecks, namely parameterization of the design, the relation between the design definition and the computational simulation and the risk of the high dimensionality and spurious variables.

The first point was considered from a pragmatcal perspective: design industry has a standard nowadays and this standard is built upon NURBS. Although other possibilities may exists, industrial short term applicability of this research requires adherence to this standard. A complete review of the NURBS structure allowed us to derive some relations between geometries that will be needed for a smart application of boundary continuity constraints.

The link between computational simulation and the geometry definition has been solved with the point inversion technique. Accepting the complexity of the meshing process, an efficient methodology has been displayed in order to establish the connection

6.3 Analytical constraints can further reduce the dimensionality

between each surface point in the computational mesh and the geometry parameters. The heuristic nature of this process requires to take special precaution in order to avoid risking the stability of the CFD solver and optimization itself. Two heuristic point inversion algorithms were developed and benchmarked against the classical parameter sweep methodology and Hu-Wallner algorithms. All these algorithm share the same weakness: the reliability of their results. A combined approach between several techniques has been put in place in order to reduce the risk of common failure. The efficiency of these techniques is non-critical because it can be considered a pre-processing task for the adjoint optimization, *i.e.* it has to be performed only once, so that extra computational effort that will pay-off in terms of reliability of the solution would be easily accepted.

After recovering the real geometry definition as optimization variables, a huge increase in the problem dimensionality could be anticipated. A new methodology that uses the information provided by the gradient on the weight variables is presented. Operating on the weights we loose the restricted influence of the NURBS control points, but on the other side, we receive information about the relative weight of each control point with respect to the other. Removing geometry control points based on their weight is a complete new approach which has demonstrated that it strongly contributes to the dimensionality reduction of the problem, and the removal of the spurious solutions and wrinkles in the geometry.

Finally, the application of analytical constraints for imposing continuity $\mathcal{C}^n$, can also contribute to the dimensionality reduction, and to avoid unnecessary iterations in the optimization.

Further extensions to this research line

It might be clear that next step could be the link with a CFD tool with adjoint capabilities, but it is not a research activity by itself.

However, there are still some interesting areas for development:

First of all, the point inversion in this study is applied to untrimmed NURBS. Application in a complete CFD optimization process requires the handling of intersection between different NURBS patches. These intersections, which have not been solved so far for deformation functions, could be almost a straight forward process now. NURBS surfaces intersection can be calculated in the parameter space (ξ, η) of each NURBS patch and this deformation in the inner points can be translated with a 2D deformation tool

(energy, RBF, etc) to obtain the new parametric coordinates of the surface mesh points that can be reconstructed automatically.

Secondly, the concept of the dimensionality reduction based on the weights should be adapted to surfaces, because we will need to remove two complete rows of control points. different approaches for accounting the accumulated weight in each node (considering the line of control points in both directions).

Finally, the mathematical handling of the analytical continuity boundary constraints should also be adapted in order to benefit from the potential dimensionality reduction.

Appendix

NURBS expressions

Contents

A.1	1 st order, 3 control points, knots=[0,0,1,2,2]	110
A.2	2 nd order, 4 control points, knots=[0,0,0,1,2,2,2]	111
A.3	2 nd order, 5 control points, knots=[0,0,0,1,2,3,3,3]	113
A.4	2 nd order, 5 control points, knots=[0,0,0,1,1,2,2,2]	115
A.5	2 nd order, 6 control points, knots=[0,0,0,1,2,3,4,4,4]	117
A.6	2 nd order, 6 control points, knots=[0,0,0,1,1,2,3,3,3]	119
A.7	2 nd order, 7 control points, knots=[0,0,0,1,2,3,4,5,5,5]	122
A.8	2 nd order, 7 control points, knots=[0,0,0,1,1,2,3,4,4,4]	125
A.9	2 nd order, 7 control points, knots=[0,0,0,1,2,2,3,4,4,4]	127
A.10	2 nd order, 7 control points, knots=[0,0,0,1,1,2,2,3,3,3]	129
A.11	3 rd order, 5 control points, knots=[0,0,0,0,1,2,2,2,2]	131
A.12	3 rd order, 6 control points, knots=[0,0,0,0,1,2,3,3,3,3]	133
A.13	3 rd order, 6 control points, knots=[0,0,0,0,1,1,2,2,2,2]	135
A.14	3 rd order, 7 control points, knots=[0,0,0,0,1,2,3,4,4,4,4]	137
A.15	3 rd order, 7 control points, knots=[0,0,0,0,1,1,2,3,3,3,3]	140
A.16	3 rd order, 7 control points, knots=[0,0,0,0,1,1,1,2,2,2,2]	142
A.17	3 rd order, 8 control points, knots=[0,0,0,0,1,2,3,4,5,5,5,5]	144
A.18	3 rd order, 8 control points, knots=[0,0,0,0,1,1,2,3,4,4,4,4]	147
A.19	3 rd order, 8 control points, knots=[0,0,0,0,1,1,1,2,3,3,3,3]	150
A.20	3 rd order, 8 control points, knots=[0,0,0,0,1,1,2,2,3,3,3,3]	152
A.21	3 rd order, 8 control points, knots=[0,0,0,0,1,2,2,3,4,4,4,4]	155

A.1 1st order, 3 control points curve without knot multiplicities

knotvector=[0,0,1,2,2]

$$\text{knotvector} = [0, 0, 1, 2, 2]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{(u-1)w_0\mathbf{p}_0 - uw_1\mathbf{p}_1}{(u-1)w_0 - uw_1} & 0 \leq u < 1 \\ \frac{(u-2)w_1\mathbf{p}_1 + (1-u)w_2\mathbf{p}_2}{(u-2)w_1 + (1-u)w_2} & 1 \leq u < 2 \\ \text{undefined} & 2 \leq u \end{cases} \quad (\text{A.1})$$

Figure A.1: Basis functions of 1st order, 3 control points curve without knot multiplicities (knotvector = [0,0,1,2,2])

$$\begin{aligned} \mathcal{C}^1 \text{ at } u = 1 &\implies -w_0 \cdot \mathbf{p}_0 + (w_0 + w_2) \cdot \mathbf{p}_1 - w_2 \cdot \mathbf{p}_2 = 0 \\ &\quad -\mathbf{p}_0 + \left(1 + \frac{w_2}{w_0}\right) \cdot \mathbf{p}_1 - \frac{w_2}{w_0} \cdot \mathbf{p}_2 = 0 \\ &\quad \mathbf{p}_1 - \mathbf{p}_0 = (\mathbf{p}_2 - \mathbf{p}_1) \cdot \frac{w_2}{w_0} \end{aligned} \quad (\text{A.2})$$

A.2 2nd order 4 control points curve without knot multiplicities

knotvector=[0,0,0,1,2,2,2]

Figure A.2: Basis functions of 2nd order 4 control points curve without knot multiplicities (knotvector = [0,0,0,1,2,2,2])

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.3})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies (w_1 + w_2) w_0 \mathbf{p}_0 - (4 w_2 - w_3 + w_0) w_1 \mathbf{p}_1 - (-4 w_1 + w_0 - w_3) w_2 \mathbf{p}_2 - (w_1 + w_2) w_3 \mathbf{p}_3 = 0 \quad (\text{A.4})$$

$$\begin{aligned} \text{a) } \xrightarrow{w_0=w_3=A} & A (w_1 + w_2) \mathbf{p}_0 - 4 w_1 w_2 \mathbf{p}_1 + 4 w_1 w_2 \mathbf{p}_2 - A (w_1 + w_2) \mathbf{p}_3 = 0 \\ & \mathbf{p}_0 - \mathbf{p}_3 = \frac{4 w_1 w_2}{A (w_1 + w_2)} (\mathbf{p}_1 - \mathbf{p}_2) \end{aligned} \quad (\text{A.5})$$

$$\text{b) } \longrightarrow (w_0 \mathbf{p}_0 - w_3 \mathbf{p}_3) = \frac{1}{w_1 + w_2} [4 w_1 w_2 (\mathbf{p}_1 - \mathbf{p}_2) - (w_3 - w_0) (w_1 \mathbf{p}_1 + w_2 \mathbf{p}_2)] \quad (\text{A.6})$$

$$\xrightarrow{\mathbf{p}_i^* = \mathbf{p}_i - \mathbf{p}_0} -w_3 \mathbf{p}_3^* = \frac{1}{w_1 + w_2} [4 w_1 w_2 (\mathbf{p}_1^* - \mathbf{p}_2^*) - (w_3 - w_0) (w_1 \mathbf{p}_1^* + w_2 \mathbf{p}_2^*)] \quad (\text{A.7})$$

$$\mathbf{knotvector} = [0, 0, 0, 1, 2, 2, 2]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{2 w_0 \mathbf{p}_0 - 4 w_0 \mathbf{p}_0 u + 2 w_0 \mathbf{p}_0 u^2 + 4 w_1 \mathbf{p}_1 u - 3 w_1 \mathbf{p}_1 u^2 + w_2 \mathbf{p}_2 u^2}{2 w_0 - 4 w_0 u + 2 w_0 u^2 + 4 w_1 u - 3 w_1 u^2 + w_2 u^2} & 0 \leq u < 1 \\ \frac{4 w_1 \mathbf{p}_1 - 4 w_1 \mathbf{p}_1 u + w_1 \mathbf{p}_1 u^2 + 8 w_2 \mathbf{p}_2 u - 3 w_2 \mathbf{p}_2 u^2 - 4 w_2 \mathbf{p}_2 + 2 w_3 \mathbf{p}_3 - 4 w_3 \mathbf{p}_3 u + 2 w_3 \mathbf{p}_3 u^2}{4 w_1 - 4 w_1 u + w_1 u^2 + 8 w_2 u - 3 w_2 u^2 - 4 w_2 + 2 w_3 - 4 w_3 u + 2 w_3 u^2} & 1 \leq u < 2 \\ \text{undefined} & 2 \leq u \end{cases} \quad (\text{A.8})$$

A.3 2nd order, 5 control points curve without knot multiplicities

knotvector=[0,0,0,1,2,3,3,3]

Figure A.3: Basis functions of 2nd order, 5 control points curve without knot multiplicities (knotvector = [0,0,0,1,2,3,3,3])

$$\mathbf{knotvector} = [0, 0, 0, 1, 2, 3, 3, 3]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{2w_0\mathbf{p}_0 - 4w_0\mathbf{p}_0u + 2w_0\mathbf{p}_0u^2 + 4uw_1\mathbf{p}_1 - 3u^2w_1\mathbf{p}_1 + w_2\mathbf{p}_2u^2}{2w_0 - 4w_0u + 2w_0u^2 + 4uw_1 - 3u^2w_1 + w_2u^2} & 0 \leq u < 1 \\ \frac{4w_1\mathbf{p}_1 - 4uw_1\mathbf{p}_1 + u^2w_1\mathbf{p}_1 + 6w_2\mathbf{p}_2u - 2w_2\mathbf{p}_2u^2 - 3w_2\mathbf{p}_2 + w_3\mathbf{p}_3 - 2w_3\mathbf{p}_3u + w_3\mathbf{p}_3u^2}{4w_1 - 4uw_1 + u^2w_1 + 6w_2u - 2w_2u^2 - 3w_2 + w_3 - 2w_3u + w_3u^2} & 1 \leq u < 2 \\ \frac{9w_2\mathbf{p}_2 - 6w_2\mathbf{p}_2u + w_2\mathbf{p}_2u^2 - 15w_3\mathbf{p}_3 + 14w_3\mathbf{p}_3u - 3w_3\mathbf{p}_3u^2 + 8w_4\mathbf{p}_4 - 8w_4\mathbf{p}_4u + 2w_4\mathbf{p}_4u^2}{9w_2 - 6w_2u + w_2u^2 - 15w_3 + 14w_3u - 3w_3u^2 + 8w_4 - 8w_4u + 2w_4u^2} & 2 \leq u < 3 \\ \text{undefined} & 3 \leq u \end{cases} \quad (\text{A.9})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.10})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 2(w_1 + w_2)w_0\mathbf{p}_0 + (-7w_2 - 2w_0 + w_3)w_1\mathbf{p}_1 + (7w_1 - 2w_0 + w_3)w_2\mathbf{p}_2 - (w_1 + w_2)w_3\mathbf{p}_3 = 0 \quad (\text{A.11})$$

$$\text{a) } \xrightarrow{w_i=A} A^2 (2\mathbf{p}_0 - 4\mathbf{p}_1 + 3\mathbf{p}_2 - \mathbf{p}_3) = 0 \quad (\text{A.12})$$

$$\text{b) } \longrightarrow (w_1 + w_2)(2w_0\mathbf{p}_0 - w_3\mathbf{p}_3) = (7w_2 + 2w_0 - w_3)w_1\mathbf{p}_1 + (-7w_1 + 2w_0 - w_3)w_2\mathbf{p}_2 \quad (\text{A.13})$$

$$\xrightarrow{\mathbf{p}_i^* = \mathbf{p}_i - \mathbf{p}_3} 2(w_1 + w_2)w_0\mathbf{p}_0^* = (7w_2 + 2w_0 - w_3)w_1\mathbf{p}_1^* + (-7w_1 + 2w_0 - w_3)w_2\mathbf{p}_2^* \quad (\text{A.14})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.15})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies (w_2 + w_3)w_1\mathbf{p}_1 + (-7w_3 + 2w_4 - w_1)w_2\mathbf{p}_2 + (7w_2 + 2w_4 - w_1)w_3\mathbf{p}_3 - 2(w_2 + w_3)w_4\mathbf{p}_4 = 0 \quad (\text{A.16})$$

A.4 2nd order, 5 control points curve with knot multiplicity

knotvector=[0,0,0,1,1,2,2,2]

Figure A.4: Basis functions of 2nd order, 5 control points curve with knot multiplicity (knotvector = [0,0,0,1,1,2,2,2])

$$\begin{aligned}
& \mathbf{knotvector} = [0, 0, 0, 1, 1, 2, 2, 2] \\
C(u) = & \begin{cases} \text{undefined} & u < 0 \\ \frac{w_0 \mathbf{p}_0 - 2 w_0 \mathbf{p}_0 u + w_0 \mathbf{p}_0 u^2 + 2 u w_1 \mathbf{p}_1 - 2 u^2 w_1 \mathbf{p}_1 + w_2 \mathbf{p}_2 u^2}{w_0 - 2 w_0 u + w_0 u^2 + 2 u w_1 - 2 u^2 w_1 + w_2 u^2} & 0 \leq u < 1 \\ \frac{4 w_2 \mathbf{p}_2 - 4 w_2 \mathbf{p}_2 u + w_2 \mathbf{p}_2 u^2 - 4 w_3 \mathbf{p}_3 + 6 w_3 \mathbf{p}_3 u - 2 w_3 \mathbf{p}_3 u^2 + w_4 \mathbf{p}_4 - 2 w_4 \mathbf{p}_4 u + w_4 \mathbf{p}_4 u^2}{4 w_2 - 4 w_2 u + w_2 u^2 - 4 w_3 + 6 w_3 u - 2 w_3 u^2 + w_4 - 2 w_4 u + w_4 u^2} & 1 \leq u < 2 \\ \text{undefined} & 2 \leq u \end{cases} \quad (\text{A.17})
\end{aligned}$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies -w_1 \mathbf{p}_1 + (w_1 + w_3) \mathbf{p}_2 - w_3 \mathbf{p}_3 = 0 \quad (\text{A.18})$$

$$\begin{matrix} \mathbf{p}_1^* = \mathbf{p}_i - \mathbf{p}_1 \\ \longrightarrow \end{matrix} (w_1 + w_3) \mathbf{p}_2^* = w_3 \mathbf{p}_3^* \quad (\text{A.19})$$

$$\begin{aligned}
\mathcal{C}^2 \text{ at } u = 1 \implies & w_0 \mathbf{p}_0 w_2 - 2 w_1 (2 w_1 - w_2) \mathbf{p}_1 + (4 w_1^2 - 4 w_3^2 + (w_4 - w_0 - 2 w_1 + 2 w_3) w_2) \mathbf{p}_2 \\
& - 2 w_3 (-2 w_3 + w_2) \mathbf{p}_3 - w_4 \mathbf{p}_4 w_2 = 0 \quad (\text{A.20})
\end{aligned}$$

A.5 2nd order, 6 control points curve without knot multiplicity

knotvector=[0,0,0,1,2,3,4,4,4]

$N_{0,2}(u) \longleftrightarrow \mathbf{p}_0$
 $N_{1,2}(u) \longleftrightarrow \mathbf{p}_1$
 $N_{2,2}(u) \longleftrightarrow \mathbf{p}_2$
 $N_{3,2}(u) \longleftrightarrow \mathbf{p}_3$
 $N_{4,2}(u) \longleftrightarrow \mathbf{p}_4$
 $N_{5,2}(u) \longleftrightarrow \mathbf{p}_5$

Figure A.5: Basis functions of §4.2.6.5

(knotvector = [0,0,0,1,2,3,4,4,4])

$$\mathbf{knotvector} = [0, 0, 0, 1, 2, 3, 4, 4, 4]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{2w_0\mathbf{p}_0 - 4w_0\mathbf{p}_0u + 2w_0\mathbf{p}_0u^2 + 4uw_1\mathbf{p}_1 - 3u^2w_1\mathbf{p}_1 + w_2\mathbf{p}_2u^2}{2w_0 - 4w_0u + 2w_0u^2 + 4uw_1 - 3u^2w_1 + w_2u^2} & 0 \leq u < 1 \\ \frac{4w_1\mathbf{p}_1 - 4uw_1\mathbf{p}_1 + u^2w_1\mathbf{p}_1 + 6w_2\mathbf{p}_2u - 2w_2\mathbf{p}_2u^2 - 3w_2\mathbf{p}_2 + w_3\mathbf{p}_3 - 2w_3\mathbf{p}_3u + w_3\mathbf{p}_3u^2}{-3w_2 + w_3 + 4w_1 - 4uw_1 + u^2w_1 + 6w_2u - 2w_2u^2 - 2w_3u + w_3u^2} & 1 \leq u < 2 \\ \frac{9w_2\mathbf{p}_2 - 6w_2\mathbf{p}_2u + w_2\mathbf{p}_2u^2 - 11w_3\mathbf{p}_3 + 10w_3\mathbf{p}_3u - 2w_3\mathbf{p}_3u^2 + 4w_4\mathbf{p}_4 - 4w_4\mathbf{p}_4u + w_4\mathbf{p}_4u^2}{-11w_3 + 4w_4 + 9w_2 + w_2u^2 - 6w_2u + 10w_3u - 2w_3u^2 - 4w_4u + w_4u^2} & 2 \leq u < 3 \\ \frac{16w_3\mathbf{p}_3 - 8w_3\mathbf{p}_3u + w_3\mathbf{p}_3u^2 - 32w_4\mathbf{p}_4 + 20w_4\mathbf{p}_4u - 3w_4\mathbf{p}_4u^2 + 18w_5\mathbf{p}_5 - 12w_5\mathbf{p}_5u + 2w_5\mathbf{p}_5u^2}{18w_5 + 16w_3 - 32w_4 - 8w_3u + w_3u^2 + 20w_4u - 3w_4u^2 - 12w_5u + 2w_5u^2} & 3 \leq u < 4 \\ \text{undefined} & 4 \leq u \end{cases} \quad (\text{A.21})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.22})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 2w_0(w_1 + w_2)\mathbf{p}_0 - w_1(2w_0 - w_3 + 7w_2)\mathbf{p}_1 - w_2(-7w_1 + 2w_0 - w_3)\mathbf{p}_2 - w_3(w_1 + w_2)\mathbf{p}_3 = 0 \quad (\text{A.23})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.24})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies (w_2 + w_3)w_1\mathbf{p}_1 - w_2(w_1 - w_4 + 6w_3)\mathbf{p}_2 - w_3(-6w_2 + w_1 - w_4)\mathbf{p}_3 - w_4(w_2 + w_3)\mathbf{p}_4 = 0 \quad (\text{A.25})$$

$$\mathcal{C}^1 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.26})$$

$$\mathcal{C}^2 \text{ at } u = 3 \implies w_2(w_3 + w_4)\mathbf{p}_2 - w_3(w_2 + 7w_4 - 2w_5)\mathbf{p}_3 - w_4(-7w_3 + w_2 - 2w_5)\mathbf{p}_4 - 2w_5(w_3 + w_4)\mathbf{p}_5 = 0 \quad (\text{A.27})$$

A.6 2nd order, 6 control points curve with knot multiplicity

knotvector=[0,0,0,1,1,2,3,3,3]

Figure A.6: Basis functions of 2nd order, 6 control points curve with knot multiplicity (knotvector = [0,0,0,1,1,2,3,3,3])

Figure A.7: NURBS structure: composition of other NURBS

$$\mathbf{knotvector} = [0, 0, 0, 1, 1, 2, 3, 3, 3]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{w_0 \mathbf{p}_0 - 2 w_0 \mathbf{p}_0 u + w_0 \mathbf{p}_0 u^2 + 2 u w_1 \mathbf{p}_1 - 2 u^2 w_1 \mathbf{p}_1 + w_2 \mathbf{p}_2 u^2}{w_0 - 2 w_0 u + w_0 u^2 + 2 u w_1 - 2 u^2 w_1 + w_2 u^2} & 0 \leq u < 1 \\ \frac{8 w_2 \mathbf{p}_2 - 8 w_2 \mathbf{p}_2 u + 2 w_2 \mathbf{p}_2 u^2 - 7 w_3 \mathbf{p}_3 + 10 w_3 \mathbf{p}_3 u - 3 w_3 \mathbf{p}_3 u^2 + w_4 \mathbf{p}_4 - 2 w_4 \mathbf{p}_4 u + w_4 \mathbf{p}_4 u^2}{8 w_2 - 8 w_2 u + 2 w_2 u^2 - 7 w_3 + 10 w_3 u - 3 w_3 u^2 + w_4 - 2 w_4 u + w_4 u^2} & 1 \leq u < 2 \\ \frac{9 w_3 \mathbf{p}_3 - 6 w_3 \mathbf{p}_3 u + w_3 \mathbf{p}_3 u^2 - 15 w_4 \mathbf{p}_4 + 14 w_4 \mathbf{p}_4 u - 3 w_4 \mathbf{p}_4 u^2 + 8 w_5 \mathbf{p}_5 - 8 w_5 \mathbf{p}_5 u + 2 w_5 \mathbf{p}_5 u^2}{9 w_3 - 6 w_3 u + w_3 u^2 - 15 w_4 + 14 w_4 u - 3 w_4 u^2 + 8 w_5 - 8 w_5 u + 2 w_5 u^2} & 2 \leq u < 3 \\ \text{undefined} & 3 \leq u \end{cases} \quad (\text{A.28})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies -w_1 \mathbf{p}_1 + (w_1 + w_3) \mathbf{p}_2 - w_3 \mathbf{p}_3 = 0 \quad (\text{A.29})$$

$$\begin{aligned} \mathcal{C}^2 \text{ at } u = 1 \implies & 2 w_0 \mathbf{p}_0 w_2 - 4 w_1 (2 w_1 - w_2) \mathbf{p}_1 + (-8 w_3^2 + 8 w_1^2 - 4 w_1 w_2 + w_2 w_4 + 5 w_2 w_3 - 2 w_0 w_2) \mathbf{p}_2 \\ & - w_3 (-8 w_3 + 5 w_2) \mathbf{p}_3 - w_4 \mathbf{p}_4 w_2 = 0 \end{aligned} \quad (\text{A.30})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.31})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies w_2 (w_3 + w_4) \mathbf{p}_2 - w_3 (4 w_4 - w_5 + w_2) \mathbf{p}_3 - w_4 (-4 w_3 + w_2 - w_5) \mathbf{p}_4 - w_5 (w_3 + w_4) \mathbf{p}_5 = 0 \quad (\text{A.32})$$

A.7 2nd order, 7 control points curve without knot multiplicity

knotvector=[0,0,0,1,2,3,4,5,5,5]

Figure A.8: Basis functions of 2nd order, 7 control points curve without knot multiplicity (knotvector = [0,0,0,1,2,3,4,5,5,5])

$$\begin{aligned}
& \mathbf{knotvector} = [0, 0, 0, 1, 2, 3, 4, 5, 5, 5] \\
C(u) = & \begin{cases}
\text{undefined} & u < 0 \\
\frac{2w_0\mathbf{p}_0 - 4w_0\mathbf{p}_0u + 2w_0\mathbf{p}_0u^2 + 4w_1\mathbf{p}_1u - 3w_1\mathbf{p}_1u^2 + w_2\mathbf{p}_2u^2}{2w_0 - 4w_0u + 2w_0u^2 + 4w_1u - 3w_1u^2 + w_2u^2} & 0 \leq u < 1 \\
\frac{4w_1\mathbf{p}_1 - 4w_1\mathbf{p}_1u + w_1\mathbf{p}_1u^2 + 6w_2\mathbf{p}_2u - 2w_2\mathbf{p}_2u^2 - 3w_2\mathbf{p}_2 + w_3\mathbf{p}_3 - 2w_3\mathbf{p}_3u + w_3\mathbf{p}_3u^2}{4w_1 - 4w_1u + w_1u^2 + 6w_2u - 2w_2u^2 - 3w_2 + w_3 - 2w_3u + w_3u^2} & 1 \leq u < 2 \\
\frac{9w_2\mathbf{p}_2 - 6w_2\mathbf{p}_2u + w_2\mathbf{p}_2u^2 - 11w_3\mathbf{p}_3 + 10w_3\mathbf{p}_3u - 2w_3\mathbf{p}_3u^2 + 4w_4\mathbf{p}_4 - 4w_4\mathbf{p}_4u + w_4\mathbf{p}_4u^2}{-6w_2u + w_2u^2 + 10w_3u - 2w_3u^2 - 4w_4u + w_4u^2 + 9w_2 - 11w_3 + 4w_4} & 2 \leq u < 3 \\
\frac{16w_3\mathbf{p}_3 - 8w_3\mathbf{p}_3u + w_3\mathbf{p}_3u^2 - 23w_4\mathbf{p}_4 + 14w_4\mathbf{p}_4u - 2w_4\mathbf{p}_4u^2 + 9w_5\mathbf{p}_5 - 6w_5\mathbf{p}_5u + w_5\mathbf{p}_5u^2}{-6w_5u + w_5u^2 - 8w_3u + w_3u^2 + 14w_4u - 2w_4u^2 - 23w_4 + 16w_3 + 9w_5} & 3 \leq u < 4 \\
\frac{25w_4\mathbf{p}_4 - 10w_4\mathbf{p}_4u + w_4\mathbf{p}_4u^2 - 55w_5\mathbf{p}_5 + 26w_5\mathbf{p}_5u - 3w_5\mathbf{p}_5u^2 + 32w_6\mathbf{p}_6 - 16w_6\mathbf{p}_6u + 2w_6\mathbf{p}_6u^2}{-10w_4u + w_4u^2 + 26w_5u - 3w_5u^2 - 16w_6u + 2w_6u^2 - 55w_5 + 25w_4 + 32w_6} & 4 \leq u < 5 \\
\text{undefined} & 5 \leq u
\end{cases} \quad (\text{A.33})
\end{aligned}$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.34})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 2w_0(w_1 + w_2)\mathbf{p}_0 - w_1(2w_0 - w_3 + 7w_2)\mathbf{p}_1 - w_2(-7w_1 + 2w_0 - w_3)\mathbf{p}_2 - w_3(w_1 + w_2)\mathbf{p}_3 = 0 \quad (\text{A.35})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.36})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies w_1(w_2 + w_3)\mathbf{p}_1 - w_2(w_1 + 6w_3 - w_4)\mathbf{p}_2 - w_3(-6w_2 + w_1 - w_4)\mathbf{p}_3 - w_4(w_2 + w_3)\mathbf{p}_4 = 0 \quad (\text{A.37})$$

$$\mathcal{C}^1 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.38})$$

$$\mathcal{C}^2 \text{ at } u = 3 \implies w_2(w_3 + w_4)\mathbf{p}_2 - w_3(w_2 + 6w_4 - w_5)\mathbf{p}_3 - w_4(-6w_3 + w_2 - w_5)\mathbf{p}_4 - w_5(w_3 + w_4)\mathbf{p}_5 = 0 \quad (\text{A.39})$$

$$\mathcal{C}^1 \text{ at } u = 4 \implies 0 = 0 \quad (\text{A.40})$$

$$\mathcal{C}^2 \text{ at } u = 4 \implies w_3(w_4 + w_5)\mathbf{p}_3 - w_4(7w_5 - 2w_6 + w_3)\mathbf{p}_4 - w_5(-7w_4 + w_3 - 2w_6)\mathbf{p}_5 - 2w_6(w_4 + w_5)\mathbf{p}_6 = 0 \quad (\text{A.41})$$

A.8 2nd order, 7 control points curve with knot multiplicity

knotvector=[0,0,0,1,1,2,3,4,4,4]

Figure A.9: Basis functions of 2nd order, 7 control points curve with knot multiplicity (knotvector = [0,0,0,1,1,2,3,4,4,4])

$$\mathbf{knotvector} = [0, 0, 0, 1, 1, 2, 3, 4, 4, 4]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{w_0 \mathbf{p}_0 - 2 w_0 \mathbf{p}_0 u + w_0 \mathbf{p}_0 u^2 + 2 w_1 \mathbf{p}_1 u - 2 w_1 \mathbf{p}_1 u^2 + w_2 \mathbf{p}_2 u^2}{w_0 - 2 w_0 u + w_0 u^2 + 2 w_1 u - 2 w_1 u^2 + w_2 u^2} & 0 \leq u < 1 \\ \frac{8 w_2 \mathbf{p}_2 - 8 w_2 \mathbf{p}_2 u + 2 w_2 \mathbf{p}_2 u^2 - 7 w_3 \mathbf{p}_3 + 10 w_3 \mathbf{p}_3 u - 3 w_3 \mathbf{p}_3 u^2 + w_4 \mathbf{p}_4 - 2 w_4 \mathbf{p}_4 u + w_4 \mathbf{p}_4 u^2}{8 w_2 - 8 w_2 u + 2 w_2 u^2 - 7 w_3 + 10 w_3 u - 3 w_3 u^2 + w_4 - 2 w_4 u + w_4 u^2} & 1 \leq u < 2 \\ \frac{9 w_3 \mathbf{p}_3 - 6 w_3 \mathbf{p}_3 u + w_3 \mathbf{p}_3 u^2 - 11 w_4 \mathbf{p}_4 + 10 w_4 \mathbf{p}_4 u - 2 w_4 \mathbf{p}_4 u^2 + 4 w_5 \mathbf{p}_5 - 4 w_5 \mathbf{p}_5 u + w_5 \mathbf{p}_5 u^2}{-4 w_5 u + w_5 u^2 - 6 w_3 u + w_3 u^2 + 10 w_4 u - 2 w_4 u^2 + 9 w_3 - 11 w_4 + 4 w_5} & 2 \leq u < 3 \\ \frac{16 w_4 \mathbf{p}_4 - 8 w_4 \mathbf{p}_4 u + w_4 \mathbf{p}_4 u^2 - 32 w_5 \mathbf{p}_5 + 20 w_5 \mathbf{p}_5 u - 3 w_5 \mathbf{p}_5 u^2 + 18 w_6 \mathbf{p}_6 - 12 w_6 \mathbf{p}_6 u + 2 w_6 \mathbf{p}_6 u^2}{-32 w_5 + 20 w_5 u - 3 w_5 u^2 - 12 w_6 u + 2 w_6 u^2 - 8 w_4 u + w_4 u^2 + 16 w_4 + 18 w_6} & 3 \leq u < 4 \\ \text{undefined} & 4 \leq u \end{cases} \quad (\text{A.42})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies -w_1 \mathbf{p}_1 + (w_1 + w_3) \mathbf{p}_2 - w_3 \mathbf{p}_3 = 0 \quad (\text{A.43})$$

$$\begin{aligned} \mathcal{C}^2 \text{ at } u = 1 \implies & 2 w_0 \mathbf{p}_0 w_2 - 4 w_1 (-w_2 + 2 w_1) \mathbf{p}_1 + (-2 w_0 w_2 - 4 w_1 w_2 + 5 w_2 w_3 + 8 w_1^2 + w_2 w_4 - 8 w_3^2) \mathbf{p}_2 \\ & - w_3 (-8 w_3 + 5 w_2) \mathbf{p}_3 - w_4 \mathbf{p}_4 w_2 = 0 \end{aligned} \quad (\text{A.44})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.45})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies 2 w_2 (w_3 + w_4) \mathbf{p}_2 - w_3 (7 w_4 - w_5 + 2 w_2) \mathbf{p}_3 - w_4 (-7 w_3 - w_5 + 2 w_2) \mathbf{p}_4 - w_5 (w_3 + w_4) \mathbf{p}_5 = 0 \quad (\text{A.46})$$

$$\mathcal{C}^1 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.47})$$

$$\mathcal{C}^2 \text{ at } u = 3 \implies w_3 (w_4 + w_5) \mathbf{p}_3 - w_4 (7 w_5 - 2 w_6 + w_3) \mathbf{p}_4 - w_5 (-7 w_4 + w_3 - 2 w_6) \mathbf{p}_5 - 2 w_6 (w_4 + w_5) \mathbf{p}_6 = 0 \quad (\text{A.48})$$

A.9 2nd order, 7 control points curve with knot multiplicity

knotvector=[0,0,0,1,2,2,3,4,4,4]

Figure A.10: Basis functions of 2nd order, 7 control points curve with knot multiplicity (knotvector = [0,0,0,1,2,2,3,4,4,4])

$$\mathbf{knotvector} = [0, 0, 0, 1, 2, 2, 3, 4, 4, 4]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{2w_0\mathbf{p}_0 - 4w_0\mathbf{p}_0u + 2w_0\mathbf{p}_0u^2 + 4w_1\mathbf{p}_1u - 3w_1\mathbf{p}_1u^2 + w_2\mathbf{p}_2u^2}{2w_0 - 4w_0u + 2w_0u^2 + 4w_1u - 3w_1u^2 + w_2u^2} & 0 \leq u < 1 \\ \frac{4w_1\mathbf{p}_1 - 4w_1\mathbf{p}_1u + w_1\mathbf{p}_1u^2 + 8w_2\mathbf{p}_2u - 3w_2\mathbf{p}_2u^2 - 4w_2\mathbf{p}_2 + 2w_3\mathbf{p}_3 - 4w_3\mathbf{p}_3u + 2w_3\mathbf{p}_3u^2}{4w_1 - 4w_1u + w_1u^2 + 8w_2u - 3w_2u^2 - 4w_2 + 2w_3 - 4w_3u + 2w_3u^2} & 1 \leq u < 2 \\ \frac{18w_3\mathbf{p}_3 - 12w_3\mathbf{p}_3u + 2w_3\mathbf{p}_3u^2 - 20w_4\mathbf{p}_4 + 16w_4\mathbf{p}_4u - 3w_4\mathbf{p}_4u^2 + 4w_5\mathbf{p}_5 - 4w_5\mathbf{p}_5u + w_5\mathbf{p}_5u^2}{4w_5 - 4w_5u + w_5u^2 - 12w_3u + 2w_3u^2 + 16w_4u - 3w_4u^2 + 18w_3 - 20w_4} & 2 \leq u < 3 \\ \frac{16w_4\mathbf{p}_4 - 8w_4\mathbf{p}_4u + w_4\mathbf{p}_4u^2 - 32w_5\mathbf{p}_5 + 20w_5\mathbf{p}_5u - 3w_5\mathbf{p}_5u^2 + 18w_6\mathbf{p}_6 - 12w_6\mathbf{p}_6u + 2w_6\mathbf{p}_6u^2}{-32w_5 + 20w_5u - 3w_5u^2 - 12w_6u + 2w_6u^2 - 8w_4u + w_4u^2 + 16w_4 + 18w_6} & 3 \leq u < 4 \\ \text{undefined} & 4 \leq u \end{cases} \quad (\text{A.49})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.50})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies (w_1 + w_2)w_0\mathbf{p}_0 - w_1(w_0 - w_3 + 4w_2)\mathbf{p}_1 - w_2(-4w_1 - w_3 + w_0)\mathbf{p}_2 - w_3(w_1 + w_2)\mathbf{p}_3 = 0 \quad (\text{A.51})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies -w_2\mathbf{p}_2 + (w_2 + w_4)\mathbf{p}_3 - w_4\mathbf{p}_4 = 0 \quad (\text{A.52})$$

$$\begin{aligned} \mathcal{C}^2 \text{ at } u = 2 \implies w_1\mathbf{p}_1w_3 - w_2(-5w_3 + 8w_2)\mathbf{p}_2 + (-5w_2w_3 + 8w_2^2 - 8w_4^2 + 5w_3w_4 + w_3w_5 - w_1w_3)\mathbf{p}_3 \\ - w_4(-8w_4 + 5w_3)\mathbf{p}_4 - w_5\mathbf{p}_5w_3 = 0 \end{aligned} \quad (\text{A.53})$$

$$\mathcal{C}^1 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.54})$$

$$\mathcal{C}^2 \text{ at } u = 3 \implies w_3(w_4 + w_5)\mathbf{p}_3 - w_4(4w_5 + w_3 - w_6)\mathbf{p}_4 - w_5(-4w_4 + w_3 - w_6)\mathbf{p}_5 - w_6(w_4 + w_5)\mathbf{p}_6 = 0 \quad (\text{A.55})$$

A.10 2nd order, 7 control points curve with knot multiplicity

knotvector=[0,0,0,1,1,2,2,3,3,3]

Figure A.11: Basis functions of 2nd order, 7 control points curve with knot multiplicity (knotvector = [0,0,0,1,1,2,2,3,3,3])

$$\mathbf{knotvector} = [0, 0, 0, 1, 1, 2, 2, 3, 3, 3]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{w_0 \mathbf{p}_0 - 2w_0 \mathbf{p}_0 u + w_0 \mathbf{p}_0 u^2 + 2w_1 \mathbf{p}_1 u - 2w_1 \mathbf{p}_1 u^2 + w_2 \mathbf{p}_2 u^2}{w_0 - 2w_0 u + w_0 u^2 + 2w_1 u - 2w_1 u^2 + w_2 u^2} & 0 \leq u < 1 \\ \frac{4w_2 \mathbf{p}_2 - 4w_2 \mathbf{p}_2 u + w_2 \mathbf{p}_2 u^2 - 4w_3 \mathbf{p}_3 + 6w_3 \mathbf{p}_3 u - 2w_3 \mathbf{p}_3 u^2 + w_4 \mathbf{p}_4 - 2w_4 \mathbf{p}_4 u + w_4 \mathbf{p}_4 u^2}{4w_2 - 4w_2 u + w_2 u^2 - 4w_3 + 6w_3 u - 2w_3 u^2 + w_4 - 2w_4 u + w_4 u^2} & 1 \leq u < 2 \\ \frac{9w_4 \mathbf{p}_4 - 6w_4 \mathbf{p}_4 u + w_4 \mathbf{p}_4 u^2 - 12w_5 \mathbf{p}_5 + 10w_5 \mathbf{p}_5 u - 2w_5 \mathbf{p}_5 u^2 + 4w_6 \mathbf{p}_6 - 4w_6 \mathbf{p}_6 u + w_6 \mathbf{p}_6 u^2}{9w_4 - 6w_4 u + w_4 u^2 - 12w_5 + 10w_5 u - 2w_5 u^2 + 4w_6 - 4w_6 u + w_6 u^2} & 2 \leq u < 3 \\ \text{undefined} & 3 \leq u \end{cases} \quad (\text{A.56})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies -w_1 \mathbf{p}_1 + (w_1 + w_3) \mathbf{p}_2 - w_3 \mathbf{p}_3 = 0 \quad (\text{A.57})$$

$$\begin{aligned} \mathcal{C}^2 \text{ at } u = 1 \implies & w_0 \mathbf{p}_0 w_2 - 2w_1 (2w_1 - w_2) \mathbf{p}_1 + (4w_1^2 - 2w_1 w_2 - 4w_3^2 - w_0 w_2 + w_2 w_4 + 2w_2 w_3) \mathbf{p}_2 \\ & - 2w_3 (-2w_3 + w_2) \mathbf{p}_3 - w_4 \mathbf{p}_4 w_2 = 0 \end{aligned} \quad (\text{A.58})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies -w_3 \mathbf{p}_3 + (w_3 + w_5) \mathbf{p}_4 - w_5 \mathbf{p}_5 = 0 \quad (\text{A.59})$$

$$\begin{aligned} \mathcal{C}^2 \text{ at } u = 2 \implies & w_2 \mathbf{p}_2 w_4 - 2w_3 (2w_3 - w_4) \mathbf{p}_3 + (4w_3^2 - w_2 w_4 - 2w_3 w_4 - 4w_5^2 + 2w_4 w_5 + w_4 w_6) \mathbf{p}_4 \\ & - 2w_5 (-2w_5 + w_4) \mathbf{p}_5 - w_6 \mathbf{p}_6 w_4 = 0 \end{aligned} \quad (\text{A.60})$$

A.11 3rd order, 5 control points curve without knot multiplicity

knotvector=[0,0,0,0,1,2,2,2,2]

Figure A.12: Basis functions of 3rd order, 5 control points curve without knot multiplicity (knotvector = [0,0,0,0,1,2,2,2,2])

$$\mathbf{knotvector} = [0, 0, 0, 0, 1, 2, 2, 2, 2]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{-4w_0\mathbf{p}_0 + 12w_0\mathbf{p}_0u - 12w_0\mathbf{p}_0u^2 + 4w_0\mathbf{p}_0u^3 - 12w_1\mathbf{p}_1u + 18w_1\mathbf{p}_1u^2 - 7w_1\mathbf{p}_1u^3 - 6w_2\mathbf{p}_2u^2 + 4w_2\mathbf{p}_2u^3 - w_3\mathbf{p}_3u^3}{-12w_1u + 18w_1u^2 - 7w_1u^3 + 4w_2u^3 + 12w_0u - 12w_0u^2 + 4w_0u^3 - 6w_2u^2 - w_3u^3 - 4w_0} & 0 \leq u < 1 \\ \frac{-8w_1\mathbf{p}_1 + 12w_1\mathbf{p}_1u - 6w_1\mathbf{p}_1u^2 + w_1\mathbf{p}_1u^3 - 24w_2\mathbf{p}_2u + 18w_2\mathbf{p}_2u^2 - 4w_2\mathbf{p}_2u^3 + 8w_2\mathbf{p}_2 - 24w_3\mathbf{p}_3u^2 + 7w_3\mathbf{p}_3u^3 + 24w_3\mathbf{p}_3u - 8w_3\mathbf{p}_3 + 4w_4\mathbf{p}_4 - 12w_4\mathbf{p}_4u + 12w_4\mathbf{p}_4u^2 - 4w_4\mathbf{p}_4u^3}{4w_4 + 24w_3u - 12w_4u + 12w_4u^2 - 4w_4u^3 + 12w_1u - 6w_1u^2 + w_1u^3 - 24w_2u + 18w_2u^2 - 4w_2u^3 - 24w_3u^2 + 7w_3u^3 - 8w_1 + 8w_2 - 8w_3} & 1 \leq u < 2 \\ \text{undefined} & 2 \leq u \end{cases} \quad (\text{A.61})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.62})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.63})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 1 \implies & -(w_3 + w_1 + 2w_2)w_0\mathbf{p}_0 + w_1(w_4 + 6w_2 + w_0)\mathbf{p}_1 + 2w_2(w_0 - 3w_3 + w_4 - 3w_1)\mathbf{p}_2 \\ & + w_3(w_4 + 6w_2 + w_0)\mathbf{p}_3 - w_4(w_3 + w_1 + 2w_2)\mathbf{p}_4 = 0 \end{aligned} \quad (\text{A.64})$$

A.12 3rd order, 6 control points curve without knot multiplicity

knotvector=[0,0,0,0,1,2,3,3,3,3]

Figure A.13: Basis functions of 3rd order, 6 control points curve without knot multiplicity (knotvector = [0,0,0,0,1,2,3,3,3,3])

$$\mathbf{knotvector} = [0, 0, 0, 0, 1, 2, 3, 3, 3, 3]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{-12 w_0 \mathbf{p}_0 + 36 w_0 \mathbf{p}_0 u - 36 w_0 \mathbf{p}_0 u^2 + 12 w_0 \mathbf{p}_0 u^3 - 36 w_1 \mathbf{p}_1 + 54 w_1 \mathbf{p}_1 u - 21 w_1 \mathbf{p}_1 u^2 + 18 w_2 \mathbf{p}_2 u^2 + 11 w_2 \mathbf{p}_2 u^3 - 2 w_3 \mathbf{p}_3 u^3}{-36 w_1 + 36 w_0 u - 36 w_0 u^2 + 12 w_0 u^3 + 54 w_1 u - 21 w_1 u^2 + 18 w_2 u^2 + 11 w_2 u^3 - 2 w_3 u^3 - 12 w_0} & 0 \leq u < 1 \\ \frac{-24 w_1 \mathbf{p}_1 + 36 w_1 \mathbf{p}_1 u - 18 w_1 \mathbf{p}_1 u^2 + 3 w_1 \mathbf{p}_1 u^3 - 54 w_2 \mathbf{p}_2 u + 36 w_2 \mathbf{p}_2 u^2 - 7 w_2 \mathbf{p}_2 u^3 + 18 w_2 \mathbf{p}_2 u^3 + 27 w_3 \mathbf{p}_3 u^2 + 7 w_3 \mathbf{p}_3 u^3 + 27 w_3 \mathbf{p}_3 u - 9 w_3 \mathbf{p}_3 + 3 w_4 \mathbf{p}_4 - 9 w_4 \mathbf{p}_4 u + 9 w_4 \mathbf{p}_4 u^2 - 3 w_4 \mathbf{p}_4 u^3}{-24 w_1 - 54 w_2 u + 36 w_2 u^2 - 7 w_2 u^3 + 27 w_3 u - 27 w_3 u^2 + 7 w_3 u^3 - 9 w_4 u + 9 w_4 u^2 - 3 w_4 u^3 - 18 w_1 + 3 w_1 u^3 + 36 w_1 u + 3 w_4 + 18 w_2 - 9 w_3} & 1 \leq u < 2 \\ \frac{-54 w_2 \mathbf{p}_2 + 54 w_2 \mathbf{p}_2 u - 18 w_2 \mathbf{p}_2 u^2 + 2 w_2 \mathbf{p}_2 u^3 - 189 w_3 \mathbf{p}_3 u + 81 w_3 \mathbf{p}_3 u^2 - 11 w_3 \mathbf{p}_3 u^3 + 135 w_3 \mathbf{p}_3 - 189 w_4 \mathbf{p}_4 + 279 w_4 \mathbf{p}_4 u - 135 w_4 \mathbf{p}_4 u^2 + 21 w_4 \mathbf{p}_4 u^3 + 96 w_5 \mathbf{p}_5 - 144 w_5 \mathbf{p}_5 u + 72 w_5 \mathbf{p}_5 u^2 - 12 w_5 \mathbf{p}_5 u^3}{135 w_3 - 189 w_4 + 96 w_5 - 54 w_2 - 144 w_5 u + 72 w_5 u^2 - 12 w_5 u^3 - 135 w_4 u^2 + 21 w_4 u^3 + 54 w_2 u - 18 w_2 u^2 + 2 w_2 u^3 + 279 w_4 u - 189 w_3 u + 81 w_3 u^2 - 11 w_3 u^3} & 2 \leq u < 3 \\ \text{undefined} & 3 \leq u \end{cases} \quad (\text{A.65})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.66})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.67})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 1 \implies & -4 w_0 (2 w_3 + 3 w_1 + 7 w_2) \mathbf{p}_0 + w_1 (3 w_4 + 7 w_3 + 74 w_2 + 12 w_0) \mathbf{p}_1 \\ & + w_2 (28 w_0 - 33 w_3 - 74 w_1 + 7 w_4) \mathbf{p}_2 + w_3 (-7 w_1 + 2 w_4 + 8 w_0 + 33 w_2) \mathbf{p}_3 \\ & - w_4 (2 w_3 + 3 w_1 + 7 w_2) \mathbf{p}_4 = 0 \end{aligned} \quad (\text{A.68})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.69})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.70})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 2 \implies & -w_1 (2 w_2 + 7 w_3 + 3 w_4) \mathbf{p}_1 + w_2 (2 w_1 + 33 w_3 + 8 w_5 - 7 w_4) \mathbf{p}_2 \\ & + w_3 (28 w_5 - 33 w_2 - 74 w_4 + 7 w_1) \mathbf{p}_3 + w_4 (7 w_2 + 12 w_5 + 74 w_3 + 3 w_1) \mathbf{p}_4 \\ & - 4 w_5 (2 w_2 + 7 w_3 + 3 w_4) \mathbf{p}_5 = 0 \end{aligned} \quad (\text{A.71})$$

A.13 3rd order, 6 control points curve with knot multiplicity

knotvector=[0,0,0,0,1,1,2,2,2,2]

Figure A.14: Basis functions of 3rd order, 6 control points curve with knot multiplicity (knotvector = [0,0,0,0,1,1,2,2,2,2])

$$\mathbf{knotvector} = [0, 0, 0, 0, 1, 1, 2, 2, 2, 2]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{-2w_0\mathbf{p}_0 + 6w_0\mathbf{p}_0u - 6w_0\mathbf{p}_0u^2 + 2w_0\mathbf{p}_0u^3 - 6uw_1\mathbf{p}_1 + 12u^2w_1\mathbf{p}_1 - 6u^3w_1\mathbf{p}_1 - 6w_2\mathbf{p}_2u^2 + 5w_2\mathbf{p}_2u^3 - w_3\mathbf{p}_3u^3}{-6uw_1 + 6w_0u - 6w_0u^2 + 2w_0u^3 + 12u^2w_1 - 6u^3w_1 - 6w_2u^2 + 5w_2u^3 - w_3u^3 - 2w_0} & 0 \leq u < 1 \\ \frac{-8w_2\mathbf{p}_2 + 12w_2\mathbf{p}_2u - 6w_2\mathbf{p}_2u^2 + w_2\mathbf{p}_2u^3 - 36w_3\mathbf{p}_3u + 24w_3\mathbf{p}_3u^2 - 5w_3\mathbf{p}_3u^3 + 16w_3\mathbf{p}_3 - 12w_4\mathbf{p}_4 + 30w_4\mathbf{p}_4u - 24w_4\mathbf{p}_4u^2 + 6w_4\mathbf{p}_4u^3 + 2w_5\mathbf{p}_5 - 6w_5\mathbf{p}_5u + 6w_5\mathbf{p}_5u^2 - 2w_5\mathbf{p}_5u^3}{12w_2u - 6w_2u^2 + w_2u^3 - 36w_3u + 24w_3u^2 - 5w_3u^3 + 30w_4u - 24w_4u^2 + 6w_4u^3 - 6w_5u + 6w_5u^2 - 2w_5u^3 + 2w_5 - 8w_2 + 16w_3 - 12w_4} & 1 \leq u < 2 \\ \text{undefined} & 2 \leq u \end{cases} \quad (\text{A.72})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.73})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies (w_2 + w_3)w_1\mathbf{p}_1 - w_2(-w_4 + 4w_3 + w_1)\mathbf{p}_2 - w_3(-4w_2 + w_1 - w_4)\mathbf{p}_3 - w_4(w_2 + w_3)\mathbf{p}_4 = 0 \quad (\text{A.74})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 1 \implies & -w_0(w_2 + w_3)^2\mathbf{p}_0 + 6w_1(w_2 + w_3)(2w_2 - w_3)\mathbf{p}_1 \\ & + w_2(-72w_2w_3 - 12w_1w_2 + w_0w_2 + 6w_4w_2 + w_2w_5 + w_3w_5 - 30w_4w_3 + w_0w_3 + 72w_3^2 + 24w_3w_1)\mathbf{p}_2 \\ & + w_3(w_0w_3 - 12w_4w_3 + 6w_3w_1 + w_3w_5 - 72w_2w_3 - 30w_1w_2 + 72w_2^2 + 24w_4w_2 + w_0w_2 + w_2w_5)\mathbf{p}_3 \\ & - 6w_4(w_2 + w_3)(w_2 - 2w_3)\mathbf{p}_4 - w_5(w_2 + w_3)^2\mathbf{p}_5 = 0 \end{aligned} \quad (\text{A.75})$$

A.14 3rd order, 7 control points curve without knot multiplicity

knotvector=[0,0,0,0,1,2,3,4,4,4,4]

Figure A.15: Basis functions of 3rd order, 7 control points curve without knot multiplicity (knotvector = [0,0,0,0,1,2,3,4,4,4,4])

$$\mathbf{knotvector} = [0, 0, 0, 0, 1, 2, 3, 4, 4, 4]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{12 w_0 \mathbf{p}_0 u^3 - 36 w_0 \mathbf{p}_0 u^2 + 36 w_0 \mathbf{p}_0 u - 12 w_0 \mathbf{p}_0 - 21 u^3 w_1 \mathbf{p}_1 + 54 u^2 w_1 \mathbf{p}_1 - 36 u w_1 \mathbf{p}_1 - 18 u^2 w_2 \mathbf{p}_2 + 11 u^3 w_2 \mathbf{p}_2 - 2 w_3 \mathbf{p}_3 u^3}{-12 w_0 + 12 w_0 u^3 - 36 w_0 u^2 + 36 w_0 u - 21 u^3 w_1 + 54 u^2 w_1 + 11 u^3 w_2 - 2 w_3 u^3 - 36 u w_1 - 18 u^2 w_2} & 0 \leq u < 1 \\ \frac{-24 w_1 \mathbf{p}_1 + 36 u w_1 \mathbf{p}_1 - 18 u^2 w_1 \mathbf{p}_1 + 3 u^3 w_1 \mathbf{p}_1 - 54 u w_2 \mathbf{p}_2 + 36 u^2 w_2 \mathbf{p}_2 - 7 u^3 w_2 \mathbf{p}_2 + 18 w_2 \mathbf{p}_2 - 24 w_3 \mathbf{p}_3 u^2 + 6 w_3 \mathbf{p}_3 u^3 + 24 w_3 \mathbf{p}_3 u - 8 w_3 \mathbf{p}_3 - 2 w_4 \mathbf{p}_4 u^3 + 6 w_4 \mathbf{p}_4 u^2 - 6 w_4 \mathbf{p}_4 u + 2 w_4 \mathbf{p}_4}{-8 w_3 + 2 w_4 + 18 w_2 - 24 w_1 + 24 w_3 u - 24 w_3 u^2 + 6 w_3 u^3 - 6 w_4 u + 6 w_4 u^2 - 2 w_4 u^3 - 54 u w_2 + 36 u^2 w_2 - 7 u^3 w_2 + 36 u w_1 - 18 u^2 w_1 + 3 u^3 w_1} & 1 \leq u < 2 \\ \frac{-54 w_2 \mathbf{p}_2 + 54 u w_2 \mathbf{p}_2 - 18 u^2 w_2 \mathbf{p}_2 + 2 u^3 w_2 \mathbf{p}_2 - 120 w_3 \mathbf{p}_3 u + 48 w_3 \mathbf{p}_3 u^2 - 6 w_3 \mathbf{p}_3 u^3 + 88 w_3 \mathbf{p}_3 - 48 w_4 \mathbf{p}_4 u^2 + 7 w_4 \mathbf{p}_4 u^3 + 102 w_4 \mathbf{p}_4 u - 70 w_4 \mathbf{p}_4 + 24 w_5 \mathbf{p}_5 - 36 w_5 \mathbf{p}_5 u + 18 w_5 \mathbf{p}_5 u^2 - 3 w_5 \mathbf{p}_5 u^3}{88 w_3 - 70 w_4 + 24 w_5 - 54 w_2 + 54 u w_2 - 18 u^2 w_2 + 2 u^3 w_2 - 120 w_3 u + 48 w_3 u^2 - 6 w_3 u^3 + 102 w_4 u - 48 w_4 u^2 + 7 w_4 u^3 + 18 w_5 u^2 - 36 w_5 u - 3 w_5 u^3} & 2 \leq u < 3 \\ \frac{-128 w_3 \mathbf{p}_3 + 96 w_3 \mathbf{p}_3 u - 24 w_3 \mathbf{p}_3 u^2 + 2 w_3 \mathbf{p}_3 u^3 - 384 w_4 \mathbf{p}_4 u + 114 w_4 \mathbf{p}_4 u^2 - 11 w_4 \mathbf{p}_4 u^3 + 416 w_4 \mathbf{p}_4 - 624 w_5 \mathbf{p}_5 + 612 w_5 \mathbf{p}_5 u - 198 w_5 \mathbf{p}_5 u^2 + 21 w_5 \mathbf{p}_5 u^3 + 324 w_6 \mathbf{p}_6 - 324 w_6 \mathbf{p}_6 u + 108 w_6 \mathbf{p}_6 u^2 - 12 w_6 \mathbf{p}_6 u^3}{-128 w_3 + 416 w_4 - 624 w_5 + 324 w_6 + 96 w_3 u - 24 w_3 u^2 + 2 w_3 u^3 - 11 w_4 u^3 + 108 w_6 u^2 - 12 w_6 u^3 + 612 w_5 u - 324 w_6 u - 384 w_4 u + 114 w_4 u^2 - 198 w_5 u^2 + 21 w_5 u^3} & 3 \leq u < 4 \\ \text{undefined} & 4 \leq u \end{cases} \quad (\text{A.76})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.77})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.78})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 1 \implies & -6 w_0 (2 w_3 + 7 w_2 + 3 w_1) \mathbf{p}_0 + 3 w_1 (6 w_0 + 4 w_3 + 37 w_2 + w_4) \mathbf{p}_1 + w_2 (-111 w_1 - 46 w_3 + 42 w_0 + 7 w_4) \mathbf{p}_2 \\ & + 2 w_3 (23 w_2 + w_4 + 6 w_0 - 6 w_1) \mathbf{p}_3 - w_4 (2 w_3 + 7 w_2 + 3 w_1) \mathbf{p}_4 = 0 \end{aligned} \quad (\text{A.79})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.80})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.81})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 2 \implies & -w_1 (w_4 + 4 w_3 + w_2) \mathbf{p}_1 + w_2 (16 w_3 + w_1 + w_5) \mathbf{p}_2 + 4 w_3 (w_5 - 4 w_2 - 4 w_4 + w_1) \mathbf{p}_3 \\ & + w_4 (16 w_3 + w_1 + w_5) \mathbf{p}_4 - w_5 (w_4 + 4 w_3 + w_2) \mathbf{p}_5 = 0 \end{aligned} \quad (\text{A.82})$$

$$\mathcal{C}^1 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.83})$$

$$\mathcal{C}^2 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.84})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 3 \implies & -w_2 (2 w_3 + 7 w_4 + 3 w_5) \mathbf{p}_2 + 2 w_3 (6 w_6 + w_2 - 6 w_5 + 23 w_4) \mathbf{p}_3 + w_4 (7 w_2 + 42 w_6 - 46 w_3 - 111 w_5) \mathbf{p}_4 \\ & + 3 w_5 (4 w_3 + 6 w_6 + w_2 + 37 w_4) \mathbf{p}_5 - 6 w_6 (2 w_3 + 7 w_4 + 3 w_5) \mathbf{p}_6 = 0 \end{aligned} \quad (\text{A.85})$$

A.15 3rd order, 7 control points curve with knot multiplicity

knotvector=[0,0,0,0,1,1,2,3,3,3,3]

Figure A.16: Basis functions of 3rd order, 7 control points curve with knot multiplicity (knotvector = [0,0,0,0,1,1,2,3,3,3,3])

$$\mathbf{knotvector} = [0, 0, 0, 0, 1, 1, 2, 3, 3, 3, 3]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{2w_0\mathbf{p}_0u^3 - 6w_0\mathbf{p}_0u^2 + 6w_0\mathbf{p}_0u - 2w_0\mathbf{p}_0 - 6u^3w_1\mathbf{p}_1 + 12u^2w_1\mathbf{p}_1 - 6uw_1\mathbf{p}_1 - 6u^2w_2\mathbf{p}_2 + 5u^3w_2\mathbf{p}_2 - w_3\mathbf{p}_3u^3}{2w_0u^3 - 6w_0u^2 + 6w_0u - 6u^3w_1 + 12u^2w_1 + 5u^3w_2 - w_3u^3 - 6uw_1 - 6u^2w_2 - 2w_0} & 0 \leq u < 1 \\ \frac{-16w_2\mathbf{p}_2 + 24uw_2\mathbf{p}_2 - 12u^2w_2\mathbf{p}_2 + 2u^3w_2\mathbf{p}_2 - 45w_3\mathbf{p}_3u + 27w_3\mathbf{p}_3u^2 - 5w_3\mathbf{p}_3u^3 + 21w_3\mathbf{p}_3 - 18w_4\mathbf{p}_4u^2 + 24w_4\mathbf{p}_4u + 4w_4\mathbf{p}_4u^3 - 10w_4\mathbf{p}_4 - w_5\mathbf{p}_5u^3 + 3w_5\mathbf{p}_5u^2 - 3w_5\mathbf{p}_5u + w_5\mathbf{p}_5}{-45w_3u + 27w_3u^2 - 5w_3u^3 + 24w_4u - 18w_4u^2 + 4w_4u^3 + 3w_5u^2 - 3w_5u - w_5u^3 + 24uw_2 - 12u^2w_2 + 2u^3w_2 + 21w_3 - 10w_4 + w_5 - 16w_2} & 1 \leq u < 2 \\ \frac{-27w_3\mathbf{p}_3 + 27w_3\mathbf{p}_3u - 9w_3\mathbf{p}_3u^2 + w_3\mathbf{p}_3u^3 - 72w_4\mathbf{p}_4u + 30w_4\mathbf{p}_4u^2 - 4w_4\mathbf{p}_4u^3 + 54w_4\mathbf{p}_4 - 45w_5\mathbf{p}_5u^2 + 7w_5\mathbf{p}_5u^3 + 93w_5\mathbf{p}_5u - 63w_5\mathbf{p}_5 + 32w_6\mathbf{p}_6 - 48w_6\mathbf{p}_6u + 24w_6\mathbf{p}_6u^2 - 4w_6\mathbf{p}_6u^3}{27w_3u - 9w_3u^2 + w_3u^3 - 72w_4u + 30w_4u^2 - 4w_4u^3 - 45w_5u^2 + 93w_5u + 7w_5u^3 - 48w_6u + 24w_6u^2 - 4w_6u^3 + 32w_6 - 27w_3 + 54w_4 - 63w_5} & 2 \leq u < 3 \\ \text{undefined} & 3 \leq u \end{cases} \quad (\text{A.86})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.87})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 2w_1(w_3 + w_2)\mathbf{p}_1 - w_2(2w_1 + 7w_3 - w_4)\mathbf{p}_2 - w_3(-7w_2 + 2w_1 - w_4)\mathbf{p}_3 - w_4(w_3 + w_2)\mathbf{p}_4 = 0 \quad (\text{A.88})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 1 \implies & -4w_0(w_3 + w_2)^2\mathbf{p}_0 + 24w_1(w_3 + w_2)(2w_2 - w_3)\mathbf{p}_1 \\ & + w_2(4w_0w_2 + 14w_2w_4 - 48w_1w_2 - 275w_2w_3 + w_2w_5 - 58w_3w_4 + w_3w_5 + 229w_3^2 + 4w_0w_3 + 96w_1w_3)\mathbf{p}_2 \\ & + w_3(w_3w_5 + 24w_1w_3 - 229w_2w_3 - 22w_3w_4 + 4w_0w_3 + 50w_2w_4 + 4w_0w_2 + w_2w_5 + 275w_2^2 - 120w_1w_2)\mathbf{p}_3 \\ & - 2w_4(w_3 + w_2)(7w_2 - 11w_3)\mathbf{p}_4 - w_5(w_3 + w_2)^2\mathbf{p}_5 = 0 \end{aligned} \quad (\text{A.89})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.90})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.91})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 2 \implies & -w_2(w_3 + 2w_4 + w_5)\mathbf{p}_2 + w_3(w_2 + 10w_4 - w_5 + 2w_6)\mathbf{p}_3 + 2w_4(2w_6 - 5w_3 - 6w_5 + w_2)\mathbf{p}_4 \\ & + w_5(w_3 + 2w_6 + w_2 + 12w_4)\mathbf{p}_5 - 2w_6(w_3 + 2w_4 + w_5)\mathbf{p}_6 = 0 \end{aligned} \quad (\text{A.92})$$

A.16 3rd order, 7 control points curve with knot multiplicity

knotvector=[0,0,0,0,1,1,1,2,2,2,2]

Figure A.17: Basis functions of 3rd order, 7 control points curve with knot multiplicity (knotvector = [0,0,0,0,1,1,1,2,2,2,2])

$$\mathbf{knotvector} = [0, 0, 0, 0, 1, 1, 1, 2, 2, 2, 2]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{w_0 \mathbf{p}_0 u^3 - 3 w_0 \mathbf{p}_0 u^2 + 3 w_0 \mathbf{p}_0 u - w_0 \mathbf{p}_0 - 3 u^3 w_1 \mathbf{p}_1 + 6 u^2 w_1 \mathbf{p}_1 - 3 u w_1 \mathbf{p}_1 - 3 u^2 w_2 \mathbf{p}_2 + 3 u^3 w_2 \mathbf{p}_2 - w_3 \mathbf{p}_3 u^3}{-w_0 + w_0 u^3 - 3 w_0 u^2 + 3 w_0 u - 3 u^3 w_1 + 6 u^2 w_1 + 3 u^3 w_2 - w_3 u^3 - 3 u w_1 - 3 u^2 w_2} & 0 \leq u < 1 \\ \frac{-8 w_3 \mathbf{p}_3 + 12 w_3 \mathbf{p}_3 u - 6 w_3 \mathbf{p}_3 u^2 + w_3 \mathbf{p}_3 u^3 - 24 w_4 \mathbf{p}_4 u + 12 w_4 \mathbf{p}_4 + 15 w_4 \mathbf{p}_4 u^2 - 3 w_4 \mathbf{p}_4 u^3 - 12 w_5 \mathbf{p}_5 u^2 + 15 w_5 \mathbf{p}_5 u - 6 w_5 \mathbf{p}_5 + 3 w_5 \mathbf{p}_5 u^3 - w_6 \mathbf{p}_6 u^3 + 3 w_6 \mathbf{p}_6 u^2 - 3 w_6 \mathbf{p}_6 u + w_6 \mathbf{p}_6}{-8 w_3 + w_6 + 12 w_4 - 6 w_5 + 12 w_3 u - 6 w_3 u^2 + w_3 u^3 - 24 w_4 u + 15 w_4 u^2 - 3 w_4 u^3 - 12 w_5 u^2 + 15 w_5 u + 3 w_5 u^3 - w_6 u^3 + 3 w_6 u^2 - 3 w_6 u} & 1 \leq u < 2 \\ \text{undefined} & 2 \leq u \end{cases} \quad (\text{A.93})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies -w_2 \mathbf{p}_2 + (w_2 + w_4) \mathbf{p}_3 - w_4 \mathbf{p}_4 = 0 \quad (\text{A.94})$$

$$\begin{aligned} \mathcal{C}^2 \text{ at } u = 1 \implies w_1 \mathbf{p}_1 w_3 - w_2 (-w_3 + 3 w_2) \mathbf{p}_2 + (-w_2 w_3 - w_1 w_3 + w_3 w_4 + w_3 w_5 + 3 w_2^2 - 3 w_4^2) \mathbf{p}_3 \\ - w_4 (-3 w_4 + w_3) \mathbf{p}_4 - w_5 \mathbf{p}_5 w_3 = 0 \end{aligned} \quad (\text{A.95})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 1 \implies -w_0 \mathbf{p}_0 w_3^2 + 3 w_1 w_3 (-2 w_3 + 3 w_2) \mathbf{p}_1 + 3 w_2 (-9 w_2^2 + 6 w_2 w_3 + 3 w_1 w_3 - w_3^2) \mathbf{p}_2 \\ + (3 w_2 w_3^2 - 18 w_3 w_4 w_5 + 27 w_4^3 + 6 w_1 w_3^2 - 18 w_2^2 w_3 - 18 w_1 w_2 w_3 + 6 w_3^2 w_5) \mathbf{p}_3 \\ + (3 w_3^2 w_4 + w_3^2 w_6 - 18 w_3 w_4^2 + 27 w_2^3 + w_0 w_3^2) \mathbf{p}_3 \\ - 3 w_4 (9 w_4^2 + w_3^2 - 3 w_3 w_5 - 6 w_3 w_4) \mathbf{p}_4 - 3 w_3 w_5 (-3 w_4 + 2 w_3) \mathbf{p}_5 - w_6 \mathbf{p}_6 w_3^2 = 0 \end{aligned} \quad (\text{A.96})$$

A.17 3rd order, 8 control points curve without knot multiplicity

knotvector=[0,0,0,0,1,2,3,4,5,5,5,5]

Figure A.18: Basis functions of 3rd order, 8 control points curve without knot multiplicity (knotvector = [0,0,0,0,1,2,3,4,5,5,5,5])

$$\mathbf{knotvector} = [0, 0, 0, 0, 1, 2, 3, 4, 5, 5, 5, 5]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{-12 w_0 \mathbf{p}_0 + 36 w_0 \mathbf{p}_0 u - 36 w_0 \mathbf{p}_0 u^2 + 12 w_0 \mathbf{p}_0 u^3 - 36 u w_1 \mathbf{p}_1 + 54 u^2 w_1 \mathbf{p}_1 - 21 u^3 w_1 \mathbf{p}_1 - 18 u^2 w_2 \mathbf{p}_2 + 11 u^3 w_2 \mathbf{p}_2 - 2 w_3 \mathbf{p}_3 u^3}{-2 w_3 u^3 - 36 u w_1 - 18 u^2 w_2 + 36 w_0 u - 36 w_0 u^2 + 12 w_0 u^3 + 54 u^2 w_1 - 21 u^3 w_1 + 11 u^3 w_2 - 12 w_0} & 0 \leq u < 1 \\ \frac{-24 w_1 \mathbf{p}_1 + 36 u w_1 \mathbf{p}_1 - 18 u^2 w_1 \mathbf{p}_1 + 3 u^3 w_1 \mathbf{p}_1 - 54 u w_2 \mathbf{p}_2 + 36 u^2 w_2 \mathbf{p}_2 - 7 u^3 w_2 \mathbf{p}_2 + 18 w_2 \mathbf{p}_2 - 24 w_3 \mathbf{p}_3 u^2 + 6 w_3 \mathbf{p}_3 u^3 + 24 w_3 \mathbf{p}_3 u - 8 w_3 \mathbf{p}_3 + 2 w_4 \mathbf{p}_4 - 6 w_4 \mathbf{p}_4 u + 6 w_4 \mathbf{p}_4 u^2 - 2 w_4 \mathbf{p}_4 u^3}{6 w_4 u^2 + 24 w_3 u - 24 w_3 u^2 + 6 w_3 u^3 - 2 w_4 u^3 - 6 w_4 u + 2 w_4 - 8 w_3 - 24 w_1 + 36 u w_1 - 18 u^2 w_1 + 3 u^3 w_1 - 54 u w_2 + 36 u^2 w_2 - 7 u^3 w_2 + 18 w_2} & 1 \leq u < 2 \\ \frac{-27 w_2 \mathbf{p}_2 + 27 u w_2 \mathbf{p}_2 - 9 u^2 w_2 \mathbf{p}_2 + u^3 w_2 \mathbf{p}_2 - 60 w_3 \mathbf{p}_3 u + 24 w_3 \mathbf{p}_3 u^2 - 3 w_3 \mathbf{p}_3 u^3 + 44 w_3 \mathbf{p}_3 - 31 w_4 \mathbf{p}_4 + 45 w_4 \mathbf{p}_4 u - 21 w_4 \mathbf{p}_4 u^2 + 3 w_4 \mathbf{p}_4 u^3 + 8 w_5 \mathbf{p}_5 - 12 w_5 \mathbf{p}_5 u + 6 w_5 \mathbf{p}_5 u^2 - w_5 \mathbf{p}_5 u^3}{-60 w_3 u + 24 w_3 u^2 - 9 u^2 w_2 + u^3 w_2 + 45 w_4 u - 21 w_4 u^2 + 3 w_4 u^3 - 12 w_5 u + 6 w_5 u^2 - w_5 u^3 + 27 u w_2 + 44 w_3 - 27 w_2 - 3 w_3 u^3 - 31 w_4 + 8 w_5} & 2 \leq u < 3 \\ \frac{-128 w_3 \mathbf{p}_3 + 96 w_3 \mathbf{p}_3 u - 24 w_3 \mathbf{p}_3 u^2 + 2 w_3 \mathbf{p}_3 u^3 + 262 w_4 \mathbf{p}_4 - 234 w_4 \mathbf{p}_4 u + 66 w_4 \mathbf{p}_4 u^2 - 6 w_4 \mathbf{p}_4 u^3 - 227 w_5 \mathbf{p}_5 + 219 w_5 \mathbf{p}_5 u - 69 w_5 \mathbf{p}_5 u^2 + 7 w_5 \mathbf{p}_5 u^3 + 81 w_6 \mathbf{p}_6 - 81 w_6 \mathbf{p}_6 u + 27 w_6 \mathbf{p}_6 u^2 - 3 w_6 \mathbf{p}_6 u^3}{-81 w_6 u + 27 w_6 u^2 - 3 w_6 u^3 + 219 w_5 u - 69 w_5 u^2 - 128 w_3 - 234 w_4 u + 66 w_4 u^2 - 6 w_4 u^3 + 96 w_3 u + 7 w_5 u^3 + 262 w_4 - 227 w_5 + 81 w_6 - 24 w_3 u^2 + 2 w_3 u^3} & 3 \leq u < 4 \\ \frac{-250 w_4 \mathbf{p}_4 + 150 w_4 \mathbf{p}_4 u - 30 w_4 \mathbf{p}_4 u^2 + 2 w_4 \mathbf{p}_4 u^3 + 925 w_5 \mathbf{p}_5 - 645 w_5 \mathbf{p}_5 u + 147 w_5 \mathbf{p}_5 u^2 - 11 w_5 \mathbf{p}_5 u^3 - 1455 w_6 \mathbf{p}_6 + 1071 w_6 \mathbf{p}_6 u - 261 w_6 \mathbf{p}_6 u^2 + 21 w_6 \mathbf{p}_6 u^3 + 768 w_7 \mathbf{p}_7 - 576 w_7 \mathbf{p}_7 u + 144 w_7 \mathbf{p}_7 u^2 - 12 w_7 \mathbf{p}_7 u^3}{768 w_7 + 925 w_5 - 250 w_4 - 1455 w_6 + 21 w_6 u^3 + 150 w_4 u + 2 w_4 u^3 - 30 w_4 u^2 - 11 w_5 u^3 + 147 w_5 u^2 - 645 w_5 u - 261 w_6 u^2 + 1071 w_6 u - 12 w_7 u^3 + 144 w_7 u^2 - 576 w_7 u} & 4 \leq u < 5 \\ \text{undefined} & 5 \leq u \end{cases} \quad (\text{A.97})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.98})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.99})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 1 \implies & -6w_0(2w_3 + 7w_2 + 3w_1)\mathbf{p}_0 + 3w_1(6w_0 + 4w_3 + 37w_2 + w_4)\mathbf{p}_1 + w_2(-111w_1 - 46w_3 + 7w_4 + 42w_0)\mathbf{p}_2 \\ & + 2w_3(6w_0 + w_4 + 23w_2 - 6w_1)\mathbf{p}_3 - w_4(2w_3 + 7w_2 + 3w_1)\mathbf{p}_4 = 0 \end{aligned} \quad (\text{A.100})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.101})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.102})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 2 \implies & -3w_1(w_4 + 4w_3 + w_2)\mathbf{p}_1 + w_2(3w_1 + 48w_3 + 2w_5 + w_4)\mathbf{p}_2 + 4w_3(-12w_2 + 3w_1 + 2w_5 - 11w_4)\mathbf{p}_3 \\ & + w_4(2w_5 - w_2 + 3w_1 + 44w_3)\mathbf{p}_4 - 2w_5(w_4 + 4w_3 + w_2)\mathbf{p}_5 = 0 \end{aligned} \quad (\text{A.103})$$

$$\mathcal{C}^1 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.104})$$

$$\mathcal{C}^2 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.105})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 3 \implies & -2w_2(w_3 + w_5 + 4w_4)\mathbf{p}_2 + w_3(3w_6 + 2w_2 - w_5 + 44w_4)\mathbf{p}_3 + 4w_4(3w_6 + 2w_2 - 12w_5 - 11w_3)\mathbf{p}_4 \\ & + w_5(w_3 + 3w_6 + 48w_4 + 2w_2)\mathbf{p}_5 - 3w_6(w_3 + w_5 + 4w_4)\mathbf{p}_6 = 0 \end{aligned} \quad (\text{A.106})$$

$$\mathcal{C}^1 \text{ at } u = 4 \implies 0 = 0 \quad (\text{A.107})$$

$$\mathcal{C}^2 \text{ at } u = 4 \implies 0 = 0 \quad (\text{A.108})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 4 \implies & -w_3(3w_6 + 2w_4 + 7w_5)\mathbf{p}_3 + 2w_4(23w_5 + w_3 + 6w_7 - 6w_6)\mathbf{p}_4 + w_5(-46w_4 + 42w_7 + 7w_3 - 111w_6)\mathbf{p}_5 \\ & + 3w_6(37w_5 + 4w_4 + w_3 + 6w_7)\mathbf{p}_6 - 6w_7(3w_6 + 2w_4 + 7w_5)\mathbf{p}_7 = 0 \end{aligned} \quad (\text{A.109})$$

A.18 3rd order, 8 control points curve without knot multiplicity

knotvector=[0,0,0,0,1,1,2,3,4,4,4,4]

Figure A.19: Basis functions of 3rd order, 8 control points curve without knot multiplicity (knotvector = [0,0,0,0,1,1,2,3,4,4,4,4])

$$\text{knotvector} = [0, 0, 0, 0, 1, 1, 2, 3, 4, 4, 4, 4]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{-2w_0\mathbf{p}_0 + 6w_0\mathbf{p}_0u - 6w_0\mathbf{p}_0u^2 + 2w_0\mathbf{p}_0u^3 - 6uw_1\mathbf{p}_1 + 12u^2w_1\mathbf{p}_1 - 6u^3w_1\mathbf{p}_1 - 6u^2w_2\mathbf{p}_2 + 5u^3w_2\mathbf{p}_2 - w_3\mathbf{p}_3u^3}{-w_3u^3 - 6uw_1 - 6u^2w_2 + 6w_0u - 6w_0u^2 + 2w_0u^3 + 12u^2w_1 - 6u^3w_1 + 5u^3w_2 - 2w_0} & 0 \leq u < 1 \\ \frac{-48w_2\mathbf{p}_2 + 72uw_2\mathbf{p}_2 - 36u^2w_2\mathbf{p}_2 + 6u^3w_2\mathbf{p}_2 - 135w_3\mathbf{p}_3u + 81w_3\mathbf{p}_3u^2 - 15w_3\mathbf{p}_3u^3 + 63w_3\mathbf{p}_3 - 29w_4\mathbf{p}_4 + 69w_4\mathbf{p}_4u - 51w_4\mathbf{p}_4u^2 + 11w_4\mathbf{p}_4u^3 + 2w_5\mathbf{p}_5 - 6w_5\mathbf{p}_5u + 6w_5\mathbf{p}_5u^2 - 2w_5\mathbf{p}_5u^3}{2w_5 + 72uw_2 - 36u^2w_2 + 6u^3w_2 - 51w_4u^2 - 135w_3u + 81w_3u^2 - 15w_3u^3 + 6w_5u^2 - 2w_5u^3 + 11w_4u^3 + 69w_4u - 6w_5u + 63w_3 - 29w_4 - 48w_2} & 1 \leq u < 2 \\ \frac{-81w_3\mathbf{p}_3 + 81w_3\mathbf{p}_3u - 27w_3\mathbf{p}_3u^2 + 3w_3\mathbf{p}_3u^3 + 115w_4\mathbf{p}_4 - 147w_4\mathbf{p}_4u + 57w_4\mathbf{p}_4u^2 - 7w_4\mathbf{p}_4u^3 - 70w_5\mathbf{p}_5 + 102w_5\mathbf{p}_5u - 48w_5\mathbf{p}_5u^2 + 7w_5\mathbf{p}_5u^3 + 24w_6\mathbf{p}_6 - 36w_6\mathbf{p}_6u + 18w_6\mathbf{p}_6u^2 - 3w_6\mathbf{p}_6u^3}{115w_4 + 3w_3u^3 - 147w_4u + 57w_4u^2 - 7w_4u^3 + 102w_5u - 48w_5u^2 + 7w_5u^3 - 36w_6u + 18w_6u^2 - 3w_6u^3 + 81w_3u - 27w_3u^2 - 70w_5 + 24w_6 - 81w_3} & 2 \leq u < 3 \\ \frac{-128w_4\mathbf{p}_4 + 96w_4\mathbf{p}_4u - 24w_4\mathbf{p}_4u^2 + 2w_4\mathbf{p}_4u^3 + 416w_5\mathbf{p}_5 - 384w_5\mathbf{p}_5u + 114w_5\mathbf{p}_5u^2 - 11w_5\mathbf{p}_5u^3 - 624w_6\mathbf{p}_6 + 612w_6\mathbf{p}_6u - 198w_6\mathbf{p}_6u^2 + 21w_6\mathbf{p}_6u^3 + 324w_7\mathbf{p}_7 - 324w_7\mathbf{p}_7u + 108w_7\mathbf{p}_7u^2 - 12w_7\mathbf{p}_7u^3}{416w_5 - 624w_6 - 11w_5u^3 + 96w_4u - 24w_4u^2 + 2w_4u^3 + 612w_6u - 198w_6u^2 + 21w_6u^3 - 324w_7u + 108w_7u^2 - 12w_7u^3 - 384w_5u + 114w_5u^2 + 324w_7 - 128w_4} & 3 \leq u < 4 \\ \text{undefined} & 4 \leq u \end{cases} \quad (\text{A.110})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.111})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 2 w_1 (w_3 + w_2) \mathbf{p}_1 - w_2 (7 w_3 + 2 w_1 - w_4) \mathbf{p}_2 - w_3 (-7 w_2 + 2 w_1 - w_4) \mathbf{p}_3 - w_4 (w_3 + w_2) \mathbf{p}_4 = 0 \quad (\text{A.112})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 1 \implies & -12 w_0 (w_3 + w_2)^2 \mathbf{p}_0 + 72 w_1 (w_3 + w_2) (2 w_2 - w_3) \mathbf{p}_1 \\ & + w_2 (43 w_2 w_4 + 12 w_0 w_2 - 144 w_1 w_2 + 2 w_2 w_5 - 825 w_2 w_3 - 173 w_3 w_4 + 687 w_3^2 + 288 w_1 w_3 + 12 w_0 w_3) \mathbf{p}_2 \\ & + w_2 (+2 w_3 w_5) \mathbf{p}_2 + w_3 (12 w_0 w_3 + 72 w_1 w_3 + 2 w_3 w_5 - 687 w_2 w_3 - 65 w_3 w_4 + 151 w_2 w_4 + 825 w_2^2) \mathbf{p}_3 \\ & + w_3 (-360 w_1 w_2 + 2 w_2 w_5 + 12 w_0 w_2) \mathbf{p}_3 - w_4 (w_3 + w_2) (43 w_2 - 65 w_3) \mathbf{p}_4 - 2 w_5 (w_3 + w_2)^2 \mathbf{p}_5 = 0 \end{aligned} \quad (\text{A.113})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.114})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.115})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 2 \implies & -2 w_2 (7 w_4 + 2 w_5 + 3 w_3) \mathbf{p}_2 + 3 w_3 (w_6 + 2 w_2 + w_5 + 20 w_4) \mathbf{p}_3 + w_4 (14 w_2 - 33 w_5 + 7 w_6 - 60 w_3) \mathbf{p}_4 \\ & + w_5 (-3 w_3 + 2 w_6 + 33 w_4 + 4 w_2) \mathbf{p}_5 - w_6 (7 w_4 + 2 w_5 + 3 w_3) \mathbf{p}_6 = 0 \end{aligned} \quad (\text{A.116})$$

$$\mathcal{C}^1 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.117})$$

$$\mathcal{C}^2 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.118})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 3 \implies & -w_3 (3 w_6 + 2 w_4 + 7 w_5) \mathbf{p}_3 + w_4 (33 w_5 + 2 w_3 + 8 w_7 - 7 w_6) \mathbf{p}_4 + w_5 (28 w_7 - 74 w_6 - 33 w_4 + 7 w_3) \mathbf{p}_5 \\ & + w_6 (7 w_4 + 12 w_7 + 74 w_5 + 3 w_3) \mathbf{p}_6 - 4 w_7 (3 w_6 + 2 w_4 + 7 w_5) \mathbf{p}_7 = 0 \end{aligned} \quad (\text{A.119})$$

A.19 3rd order, 8 control points curve without knot multiplicity

knotvector=[0,0,0,0,1,1,1,2,3,3,3,3]

Figure A.20: Basis functions of 3rd order, 8 control points curve without knot multiplicity (knotvector = [0,0,0,0,1,1,1,2,3,3,3,3])

$$\mathbf{knotvector} = [0, 0, 0, 0, 1, 1, 1, 2, 3, 3, 3, 3]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{-w_0\mathbf{p}_0 + 3w_0\mathbf{p}_0u - 3w_0\mathbf{p}_0u^2 + w_0\mathbf{p}_0u^3 - 3uw_1\mathbf{p}_1 + 6u^2w_1\mathbf{p}_1 - 3u^3w_1\mathbf{p}_1 - 3u^2w_2\mathbf{p}_2 + 3u^3w_2\mathbf{p}_2 - w_3\mathbf{p}_3u^3}{-w_3u^3 - 3uw_1 - 3u^2w_2 + 3w_0u - 3w_0u^2 + w_0u^3 + 6u^2w_1 - 3u^3w_1 + 3u^3w_2 - w_0} & 0 \leq u < 1 \\ \frac{-32w_3\mathbf{p}_3 + 48w_3\mathbf{p}_3u - 24w_3\mathbf{p}_3u^2 + 4w_3\mathbf{p}_3u^3 + 37w_4\mathbf{p}_4 - 69w_4\mathbf{p}_4u + 39w_4\mathbf{p}_4u^2 - 7w_4\mathbf{p}_4u^3 - 10w_5\mathbf{p}_5 + 24w_5\mathbf{p}_5u - 18w_5\mathbf{p}_5u^2 + 4w_5\mathbf{p}_5u^3 + w_6\mathbf{p}_6 - 3w_6\mathbf{p}_6u + 3w_6\mathbf{p}_6u^2 - w_6\mathbf{p}_6u^3}{-10w_5 + 39w_4u^2 + 48w_3u - 24w_3u^2 + 4w_3u^3 - 18w_5u^2 + 4w_5u^3 - 3w_6u + 3w_6u^2 - w_6u^3 - 7w_4u^3 - 69w_4u + 24w_5u + w_6 - 32w_3 + 37w_4} & 1 \leq u < 2 \\ \frac{-27w_4\mathbf{p}_4 + 27w_4\mathbf{p}_4u - 9w_4\mathbf{p}_4u^2 + w_4\mathbf{p}_4u^3 + 54w_5\mathbf{p}_5 - 72w_5\mathbf{p}_5u + 30w_5\mathbf{p}_5u^2 - 4w_5\mathbf{p}_5u^3 - 63w_6\mathbf{p}_6 + 93w_6\mathbf{p}_6u - 45w_6\mathbf{p}_6u^2 + 7w_6\mathbf{p}_6u^3 + 32w_7\mathbf{p}_7 - 48w_7\mathbf{p}_7u + 24w_7\mathbf{p}_7u^2 - 4w_7\mathbf{p}_7u^3}{-27w_4 + 27w_4u - 9w_4u^2 + w_4u^3 - 72w_5u + 30w_5u^2 - 4w_5u^3 + 93w_6u - 45w_6u^2 + 7w_6u^3 - 48w_7u + 24w_7u^2 - 4w_7u^3 + 54w_5 + 32w_7 - 63w_6} & 2 \leq u < 3 \\ \text{undefined} & 3 \leq u \end{cases} \quad (\text{A.120})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies -w_2\mathbf{p}_2 + (w_2 + w_4)\mathbf{p}_3 - w_4\mathbf{p}_4 = 0 \quad (\text{A.121})$$

$$\begin{aligned} \mathcal{C}^2 \text{ at } u = 1 \implies & 2w_1\mathbf{p}_1w_3 - 2w_2(3w_2 - w_3)\mathbf{p}_2 + (-2w_3w_2 + 6w_2^2 + 3w_3w_4 + w_3w_5 - 2w_1w_3 - 6w_4^2)\mathbf{p}_3 \\ & - 3w_4(-2w_4 + w_3)\mathbf{p}_4 - w_5\mathbf{p}_5w_3 = 0 \end{aligned} \quad (\text{A.122})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 1 \implies & -4w_0\mathbf{p}_0w_3^2 + 12w_1w_3(-2w_3 + 3w_2)\mathbf{p}_1 + 12w_2(-9w_2^2 + 6w_3w_2 + 3w_1w_3 - w_3^2)\mathbf{p}_2 \\ & + (-36w_3w_4w_5 - 72w_2^2w_3 + 14w_3^2w_5 + 108w_4^3 - 72w_1w_3w_2 + 12w_2w_3^2 - 108w_3w_4^2 + 108w_2^3)\mathbf{p}_3 \\ & + (4w_0w_3^2 + w_3^2w_6 + 24w_1w_3^2 + 25w_3^2w_4)\mathbf{p}_3 - w_4(108w_4^2 - 18w_3w_5 - 108w_3w_4 + 25w_3^2)\mathbf{p}_4 \\ & - 2w_3w_5(7w_3 - 9w_4)\mathbf{p}_5 - w_6\mathbf{p}_6w_3^2 = 0 \end{aligned} \quad (\text{A.123})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.124})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.125})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 2 \implies & -w_3(w_4 + w_6 + 2w_5)\mathbf{p}_3 + w_4(w_7 + w_3 + 6w_5)\mathbf{p}_4 + 2w_5(w_7 + w_3 - 3w_6 - 3w_4)\mathbf{p}_5 \\ & + w_6(w_7 + w_3 + 6w_5)\mathbf{p}_6 - w_7(w_4 + w_6 + 2w_5)\mathbf{p}_7 = 0 \end{aligned} \quad (\text{A.126})$$

A.20 3rd order, 8 control points curve without knot multiplicity

knotvector=[0,0,0,0,1,1,2,2,3,3,3,3]

Figure A.21: Basis functions of 3rd order, 8 control points curve without knot multiplicity (knotvector = [0,0,0,0,1,1,2,2,3,3,3,3])

$$\mathbf{knotvector} = [0, 0, 0, 0, 1, 1, 2, 2, 3, 3, 3, 3]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{-2w_0\mathbf{p}_0 + 6w_0\mathbf{p}_0u - 6w_0\mathbf{p}_0u^2 + 2w_0\mathbf{p}_0u^3 - 6uw_1\mathbf{p}_1 + 12u^2w_1\mathbf{p}_1 - 6u^3w_1\mathbf{p}_1 - 6u^2w_2\mathbf{p}_2 + 5u^3w_2\mathbf{p}_2 - w_3\mathbf{p}_3u^3}{-w_3u^3 - 6uw_1 - 6u^2w_2 + 6w_0u - 6w_0u^2 + 2w_0u^3 + 12u^2w_1 - 6u^3w_1 + 5u^3w_2 - 2w_0} & 0 \leq u < 1 \\ \frac{-8w_2\mathbf{p}_2 + 12uw_2\mathbf{p}_2 - 6u^2w_2\mathbf{p}_2 + u^3w_2\mathbf{p}_2 - 36w_3\mathbf{p}_3u + 24w_3\mathbf{p}_3u^2 - 5w_3\mathbf{p}_3u^3 + 16w_3\mathbf{p}_3 - 11w_4\mathbf{p}_4 + 27w_4\mathbf{p}_4u - 21w_4\mathbf{p}_4u^2 + 5w_4\mathbf{p}_4u^3 + w_5\mathbf{p}_5 - 3w_5\mathbf{p}_5u + 3w_5\mathbf{p}_5u^2 - w_5\mathbf{p}_5u^3}{-8w_2 + 16w_3 - 11w_4 + 3w_5u^2 - w_5u^3 + 5w_4u^3 + 27w_4u - 3w_5u + 12uw_2 - 6u^2w_2 + u^3w_2 - 21w_4u^2 - 36w_3u + 24w_3u^2 - 5w_3u^3 + w_5} & 1 \leq u < 2 \\ \frac{-27w_4\mathbf{p}_4 + 27w_4\mathbf{p}_4u - 9w_4\mathbf{p}_4u^2 + w_4\mathbf{p}_4u^3 + 81w_5\mathbf{p}_5 - 99w_5\mathbf{p}_5u + 39w_5\mathbf{p}_5u^2 - 5w_5\mathbf{p}_5u^3 - 72w_6\mathbf{p}_6 + 96w_6\mathbf{p}_6u - 42w_6\mathbf{p}_6u^2 + 6w_6\mathbf{p}_6u^3 + 16w_7\mathbf{p}_7 - 24w_7\mathbf{p}_7u + 12w_7\mathbf{p}_7u^2 - 2w_7\mathbf{p}_7u^3}{16w_7 - 72w_6 - 9w_4u^2 + w_4u^3 - 99w_5u + 39w_5u^2 - 5w_5u^3 + 96w_6u - 42w_6u^2 + 6w_6u^3 - 24w_7u + 12w_7u^2 - 2w_7u^3 + 27w_4u - 27w_4 + 81w_5} & 2 \leq u < 3 \\ \text{undefined} & 3 \leq u \end{cases} \quad (\text{A.127})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.128})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies (w_3 + w_2)w_1\mathbf{p}_1 - w_2(4w_3 - w_4 + w_1)\mathbf{p}_2 - w_3(-4w_2 + w_1 - w_4)\mathbf{p}_3 - w_4(w_3 + w_2)\mathbf{p}_4 = 0 \quad (\text{A.129})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 1 \implies & -2w_0(w_3 + w_2)^2\mathbf{p}_0 + 12w_1(w_3 + w_2)(2w_2 - w_3)\mathbf{p}_1 \\ & + w_2(-144w_2w_3 - 24w_1w_2 + 13w_2w_4 + w_2w_5 + 2w_0w_2 - 59w_3w_4 + w_3w_5 + 48w_1w_3 + 144w_3^2 + 2w_0w_3)\mathbf{p}_2 \\ & + w_3(-144w_2w_3 + 12w_1w_3 + w_3w_5 + 2w_0w_3 - 23w_3w_4 - 60w_1w_2 + w_2w_5 + 144w_2^2 + 2w_0w_2 + 49w_2w_4)\mathbf{p}_3 \\ & - w_4(w_3 + w_2)(13w_2 - 23w_3)\mathbf{p}_4 - w_5(w_3 + w_2)^2\mathbf{p}_5 = 0 \end{aligned} \quad (\text{A.130})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.131})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies w_3(w_4 + w_5)\mathbf{p}_3 - w_4(-w_6 + w_3 + 4w_5)\mathbf{p}_4 - w_5(-4w_4 + w_3 - w_6)\mathbf{p}_5 - w_6(w_4 + w_5)\mathbf{p}_6 = 0 \quad (\text{A.132})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 2 \implies & -w_2(w_4 + w_5)^2\mathbf{p}_2 + w_3(w_4 + w_5)(23w_4 - 13w_5)\mathbf{p}_3 \\ & + w_4(w_2w_4 + 12w_4w_6 + 2w_7w_4 - 23w_3w_4 - 144w_4w_5 + w_2w_5 - 60w_5w_6 + 144w_5^2 + 49w_3w_5 + 2w_7w_5)\mathbf{p}_4 \\ & + w_5(-24w_5w_6 - 144w_4w_5 + w_2w_5 + 13w_3w_5 + 2w_7w_5 + 48w_4w_6 - 59w_3w_4 + w_2w_4 + 144w_4^2 + 2w_7w_4)\mathbf{p}_5 \\ & - 12w_6(w_4 + w_5)(w_4 - 2w_5)\mathbf{p}_6 - 2w_7(w_4 + w_5)^2\mathbf{p}_7 = 0 \end{aligned} \quad (\text{A.133})$$

A.21 3rd order, 8 control points curve without knot multiplicity

knotvector=[0,0,0,0,1,2,2,3,4,4,4,4]

Figure A.22: Basis functions of 3rd order, 8 control points curve without knot multiplicity (knotvector = [0,0,0,0,1,2,2,3,4,4,4,4])

$$\mathbf{knotvector} = [0, 0, 0, 0, 1, 2, 2, 3, 4, 4, 4, 4]$$

$$C(u) = \begin{cases} \text{undefined} & u < 0 \\ \frac{-4w_0\mathbf{p}_0 + 12w_0\mathbf{p}_0u - 12w_0\mathbf{p}_0u^2 + 4w_0\mathbf{p}_0u^3 - 12uw_1\mathbf{p}_1 + 18u^2w_1\mathbf{p}_1 - 7u^3w_1\mathbf{p}_1 - 6u^2w_2\mathbf{p}_2 + 4u^3w_2\mathbf{p}_2 - w_3\mathbf{p}_3u^3}{-4w_0 - w_3u^3 - 12uw_1 - 6u^2w_2 + 12w_0u - 12w_0u^2 + 4w_0u^3 + 18u^2w_1 - 7u^3w_1 + 4u^3w_2} & 0 \leq u < 1 \\ \frac{-8w_1\mathbf{p}_1 + 12uw_1\mathbf{p}_1 - 6u^2w_1\mathbf{p}_1 + u^3w_1\mathbf{p}_1 - 24uw_2\mathbf{p}_2 + 18u^2w_2\mathbf{p}_2 - 4u^3w_2\mathbf{p}_2 + 8w_2\mathbf{p}_2 - 18w_3\mathbf{p}_3u^2 + 5w_3\mathbf{p}_3u^3 + 18w_3\mathbf{p}_3u - 6w_3\mathbf{p}_3 + 2w_4\mathbf{p}_4 - 6w_4\mathbf{p}_4u + 6w_4\mathbf{p}_4u^2 - 2w_4\mathbf{p}_4u^3}{-8w_1 - 6w_3 + 2w_4 + 8w_2 + 12uw_1 - 6u^2w_1 + u^3w_1 - 24uw_2 + 18u^2w_2 - 4u^3w_2 + 6w_4u^2 + 18w_3u - 18w_3u^2 + 5w_3u^3 - 2w_4u^3 - 6w_4u} & 1 \leq u < 2 \\ \frac{-54w_3\mathbf{p}_3 + 54w_3\mathbf{p}_3u - 18w_3\mathbf{p}_3u^2 + 2w_3\mathbf{p}_3u^3 + 98w_4\mathbf{p}_4 - 114w_4\mathbf{p}_4u + 42w_4\mathbf{p}_4u^2 - 5w_4\mathbf{p}_4u^3 - 56w_5\mathbf{p}_5 + 72w_5\mathbf{p}_5u - 30w_5\mathbf{p}_5u^2 + 4w_5\mathbf{p}_5u^3 + 8w_6\mathbf{p}_6 - 12w_6\mathbf{p}_6u + 6w_6\mathbf{p}_6u^2 - w_6\mathbf{p}_6u^3}{-54w_3 + 98w_4 - 56w_5 + 8w_6 + 2w_3u^3 - 114w_4u + 42w_4u^2 - 5w_4u^3 + 72w_5u - 30w_5u^2 + 4w_5u^3 - 12w_6u + 6w_6u^2 - w_6u^3 + 54w_3u - 18w_3u^2} & 2 \leq u < 3 \\ \frac{-64w_4\mathbf{p}_4 + 48w_4\mathbf{p}_4u - 12w_4\mathbf{p}_4u^2 + w_4\mathbf{p}_4u^3 + 160w_5\mathbf{p}_5 - 144w_5\mathbf{p}_5u + 42w_5\mathbf{p}_5u^2 - 4w_5\mathbf{p}_5u^3 - 208w_6\mathbf{p}_6 + 204w_6\mathbf{p}_6u - 66w_6\mathbf{p}_6u^2 + 7w_6\mathbf{p}_6u^3 + 108w_7\mathbf{p}_7 - 108w_7\mathbf{p}_7u + 36w_7\mathbf{p}_7u^2 - 4w_7\mathbf{p}_7u^3}{108w_7 - 64w_4 + 160w_5 - 208w_6 - 4w_5u^3 + 48w_4u - 12w_4u^2 + w_4u^3 + 204w_6u - 66w_6u^2 + 7w_6u^3 - 108w_7u + 36w_7u^2 - 4w_7u^3 - 144w_5u + 42w_5u^2} & 3 \leq u < 4 \\ \text{undefined} & 4 \leq u \end{cases} \quad (\text{A.134})$$

$$\mathcal{C}^1 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.135})$$

$$\mathcal{C}^2 \text{ at } u = 1 \implies 0 = 0 \quad (\text{A.136})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 1 \implies & -2w_0(w_3 + w_1 + 2w_2)\mathbf{p}_0 + w_1(2w_0 + w_3 + 12w_2 + w_4)\mathbf{p}_1 + 2w_2(-6w_1 + w_4 + 2w_0 - 5w_3)\mathbf{p}_2 \\ & + w_3(10w_2 - w_1 + 2w_0 + w_4)\mathbf{p}_3 - w_4(w_3 + w_1 + 2w_2)\mathbf{p}_4 = 0 \end{aligned} \quad (\text{A.137})$$

$$\mathcal{C}^1 \text{ at } u = 2 \implies 0 = 0 \quad (\text{A.138})$$

$$\mathcal{C}^2 \text{ at } u = 2 \implies w_2(w_4 + w_3)\mathbf{p}_2 - w_3(w_2 + 6w_4 - w_5)\mathbf{p}_3 - w_4(-6w_3 + w_2 - w_5)\mathbf{p}_4 - w_5(w_4 + w_3)\mathbf{p}_5 = 0 \quad (\text{A.139})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 2 \implies & -w_1(w_4 + w_3)^2\mathbf{p}_1 + 2w_2(w_4 + w_3)(11w_3 - 7w_4)\mathbf{p}_2 \\ & + w_3(w_3w_6 - 216w_4w_3 - 22w_3w_2 + 14w_3w_5 + w_1w_3 - 58w_4w_5 + w_1w_4 + w_6w_4 + 50w_4w_2 + 216w_4^2)\mathbf{p}_3 \\ & + w_4(-22w_4w_5 - 216w_4w_3 + 14w_4w_2 + w_1w_4 + w_6w_4 + 50w_3w_5 + w_1w_3 - 58w_3w_2 + 216w_3^2 + w_3w_6)\mathbf{p}_4 \\ & - 2w_5(w_4 + w_3)(7w_3 - 11w_4)\mathbf{p}_5 - w_6(w_4 + w_3)^2\mathbf{p}_6 = 0 \end{aligned} \quad (\text{A.140})$$

$$\mathcal{C}^1 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.141})$$

$$\mathcal{C}^2 \text{ at } u = 3 \implies 0 = 0 \quad (\text{A.142})$$

$$\begin{aligned} \mathcal{C}^3 \text{ at } u = 3 \implies & -w_3(w_4 + w_6 + 2w_5)\mathbf{p}_3 + w_4(10w_5 + w_3 + 2w_7 - w_6)\mathbf{p}_4 + 2w_5(2w_7 - 6w_6 - 5w_4 + w_3)\mathbf{p}_5 \\ & + w_6(w_4 + w_3 + 2w_7 + 12w_5)\mathbf{p}_6 - 2w_7(w_4 + w_6 + 2w_5)\mathbf{p}_7 = 0 \end{aligned} \quad (\text{A.143})$$

Bibliography

- [1] Introduction to the mathematical concepts of catia v5. Technical report, Dassault Systemes, 2009.
- [2] C.K. Au and M.M.F. Yuen. Unified approach to NURBS curve shape modification. *Computer-Aided Design*, 27(2):85–93, feb 1995.
- [3] Per Bergström and Inge Söderkvist. Fitting NURBS using separable least squares techniques. *International Journal of Mathematical Modelling and Numerical Optimisation*, 3(4):319, 2012.
- [4] Carole Blanc and Christophe Schlick. Accurate parametrization of conics by nurbs. *IEEE Computer Graphics Applications*, 16(6):64–71, Nov 1996.
- [5] Djordje Brujic, Iain Ainsworth, and Mihailo Ristic. Fast and accurate NURBS fitting for reverse engineering. *The International Journal of Advanced Manufacturing Technology*, 54(5-8):691–700, sep 2010.
- [6] Carlos Castro, Carlos Lozano, Francisco Palacios, and Enrique Zuazua. Systematic continuous adjoint approach to viscous aerodynamic design on unstructured grids. *AIAA Journal*, 45(9):2125–2139, sep 2007.
- [7] Joao L. Comba. Continuity aspects of spline curves. Technical report, Stanford University, 1993.
- [8] Juan de Burgos. *Cálculo infinitesimal de una variable*. Mac Graw Hill, 1 edition, 1994.
- [9] Juan de Burgos. *Cálculo infinitesimal de varias variables*. Mac Graw Hill, 1 edition, 1995.
- [10] Jonathan Elliott and Jaime Peraire. Practical 3d aerodynamic design and optimization using unstructured meshes. *AIAA Journal*, 35:1479–1485, 1997.

- [11] Gerald Farin. From conics to nurbs: A tutorial and survey. *IEEE Computer Graphics and Applications*, 12(5):78–86, Sep 1992.
- [12] Gerald Farin. *Curves and surfaces for computer-aided geometric design: A Practical Guide*. Morgan Kaufmann Publishers, 5 edition, 2002.
- [13] Hans Hagen, editor. *Curve and Surface Design*. Society for Industrial & Applied Mathematics (SIAM), jan 1992.
- [14] Philip Hewitt and Simao P. Marques. Aerofoil optimisation using cst parameterisation in su2. In *Royal Aeronautical Society Biennial Applied Aerodynamics Research Conference*, 2014.
- [15] Raymond M. Hicks and Preston A. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412, jul 1978.
- [16] Yang Yu Hong Qian. Solving high-dimensional multi-objective optimization problems with low effective dimensions. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 2017.
- [17] Babak S. Hosseini, Matthias Möller, and Stefan Turek. Isogeometric analysis of the navier–stokes equations with taylor–hood b-spline elements. *Applied Mathematics and Computation*, 267:264–281, sep 2015.
- [18] Shi-Min Hu and Johannes Wallner. A second order algorithm for orthogonal projection onto curves and surfaces. *Computer Aided Geometric Design*, 22:251–260, 2005.
- [19] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4135–4195, oct 2005.
- [20] Andres Iglesias. B-splines and nurbs curves and surfaces. Technical report, Universidad de Cantabria, 2001.
- [21] S. Jakobsson and O. Amoignon. Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Computers & Fluids*, 36(6):1119–1136, jul 2007.
- [22] Antony Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, sep 1988.

- [23] Armen Jaworski and Jens-Dominik Müller. Toward modular multigrid design optimisation. In *Advances in Automatic Differentiation*, pages 281–291. Springer Nature, 2008.
- [24] J. Jiménez-Varona and J. Ponsin. Improvement of the cruise performances of a wing by means of aerodynamic optimization. validation with a far-field method. In D. Knight, I. Lipatov, and P. Reijasse, editors, *Progress in Flight Physics – Volume 7*, pages 67–88. EDP Sciences, 2015.
- [25] I. Juhász and M. Hoffmann. Constrained shape modification of cubic b-spline curves by means of knots. *Computer-Aided Design*, 36(5):437–445, apr 2004.
- [26] Imre Juhász. Weight-based shape modification of nurbs curves. *Computer Aided Geometric Design*, 16:377–383, 1999.
- [27] Imre Juhász and Miklós Hoffmann. Modifying a knot of b-spline curves. *Computer Aided Geometric Design*, 20(5):243–245, aug 2003.
- [28] Bert Jüttler and Alf Felis. Least-squares fitting of algebraic spline surfaces. *Advances in Computational Mathematics*, 17(1/2):135–152, 2002.
- [29] D.S. Lee, G. Bugeda, J. Periaux, and E. Onate. Robust active shock control bump design optimisation using hybrid parallel MOGA. *Computers & Fluids*, 80:214–224, jul 2013.
- [30] Carlo Lovadina, Alessandro Reali, and Giancarlo Sangalli. What is isogeometric analysis? Technical report, University of Pavia and IMATI-CNR of Pavia, 2014.
- [31] Carlos Lozano. Discrete surprises in the computation of sensitivities from boundary integrals in the continuous adjoint approach to inviscid aerodynamic shape optimization. *Computers & Fluids*, 56:118–127, mar 2012.
- [32] Carlos Lozano, Esther Andrés, Mario Martín, and Pablo Bitrián. Domain versus boundary computation of flow sensitivities with the continuous adjoint method for aerodynamic shape optimization problems. *International Journal for Numerical Methods in Fluids*, 70(10):1305–1323, dec 2011.
- [33] Ying Liang Ma and W.T. Hewitt. Point inversion and projection for nurbs curve and surface: Control polygon approach. *Computer Aided Geometric Design*, 20:79–99, 2003.

- [34] Jan Maes and Adhemar Bultheel. Powell-sabin spline based multilevel preconditioners for 4th order elliptic equations on the sphere. Technical report, Deutsches Zentrum für Luft- und Raumfahrt, 2005.
- [35] M. J. Martin, E. Andres, M. Widhalm, P. Bitrian, and C. Lozano. Non-uniform rational b-splines-based aerodynamic shape design optimization with the DLR TAU code. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 226(10):1225–1242, dec 2011.
- [36] Mario J. Martin, Esther Andres, Carlos Lozano, and Eusebio Valero. Volumetric b-splines shape parametrization for aerodynamic shape design. *Aerospace Science and Technology*, 37:26–36, aug 2014.
- [37] Mario J. Martin, Mariola Gomez, and Esther Andres. Aerodynamic shape optimization of a 3d wing via volumetric b-splines. In *ECCOMAS 2014*, 2014.
- [38] Massachusetts Institute of Technology. *NLopt*. <http://ab-initio.mit.edu/wiki/index.php/NLopt>.
- [39] Dominic A. Masters, Nigel J. Taylor, T. Rendall, Christian B. Allen, and Daniel J. Poole. Review of aerofoil parameterisation methods for aerodynamic shape optimisation. In *53rd AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics (AIAA), jan 2015.
- [40] Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, 13:271–369, may 2004.
- [41] Knut Nordanger, Runar Holdahl, Trond Kvamsdal, Arne Morten Kvarving, and Adil Rasheed. Simulation of airflow past a 2d NACA0015 airfoil using an isogeometric incompressible navier–stokes solver with the spalart–allmaras turbulence model. *Computer Methods in Applied Mechanics and Engineering*, 290:183–208, jun 2015.
- [42] Ye Peiqing, Zhang Hui, Kaiyun Chen, and Jinsong Wang. The knot factor method and its applications in blade measurement. *Aerospace Science and Technology*, 10(5):359–363, jul 2006.
- [43] F. Perez-Arribas and I. Castaeda-Sabadell. Automatic modelling of airfoil data points. *Aerospace Science and Technology*, 55:449–457, aug 2016.

- [44] Jacques E.V. Peter and Richard P. Dwight. Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. *Computers & Fluids*, 39(3):373–391, mar 2010.
- [45] Les Piegl and Wayne Tiller. *The NURBS Book*. Springer-Verlag, 2 edition, 1997.
- [46] J. Pons-Prats, G. Bugada, F. Zárate, and E. Oñate. Optimización robusta en aplicaciones aeronáuticas con la combinación de cálculo estocástico y algoritmos evolutivos. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, 28(1):18–32, jan 2012.
- [47] J. Ponsin, F. Fraysse, M. Gómez, and M. Cordero-Gracia. An adjoint-truncation error based approach for goal-oriented mesh adaptation. *Aerospace Science and Technology*, 41:229–240, feb 2015.
- [48] M. J. D. Powell. *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation*, pages 51–67. Springer Netherlands, Dordrecht, 1994.
- [49] M. J. D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, 7:287–336, 001 1998.
- [50] Manuel Prieto Alberca. *Fundamentos geométricos del diseño en ingeniería*. Aula Documental de Investigación, 1 edition, 1992.
- [51] Daniel Redondo García and Javier de Bona Piedrabuena. Patent: Methods and systems for optimizing the design of aerodynamic surfaces, 11 2009.
- [52] Daniel Redondo García and Juan Ramón Pérez Garraleta. Patent: Method and system to optimize surface design, 12 2007.
- [53] Claudio Comis Da Ronco, Rita Ponza, and Ernesto Benini. Aerodynamic shape optimization of aircraft components using an advanced multi-objective evolutionary approach. *Computer Methods in Applied Mechanics and Engineering*, 285:255–290, mar 2015.
- [54] Arno Ronzheimer. Prospects of geometry parameterization based on freeform deformation in mdo. In ERCOFTAC, editor, *ERCOFTAC 2006 International Conference*. ERCOFTAC, ERCOFTAC, 2006. ISBN: 84-85650-12-3.

- [55] Jamshid A. Samareh. A survey of shape parameterization techniques. Technical report, NASA Langley Research Center, 1999.
- [56] Jamshid A. Samareh. Aerodynamic shape optimization based on free-form deformation. *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2004.
- [57] J. Sanchez-Reyes. A simple technique for NURBS shape modification. *IEEE Computer Graphics and Applications*, 17(1):52–59, 1997.
- [58] Claudia Schillings, Stephan Schmidt, and Volker Schulz. Efficient shape optimization for certain and uncertain aerodynamic design. *Computers & Fluids*, 46(1):78–87, jul 2011.
- [59] Stephan Schmidt, Caslav Ilic, Nicolas Gauger, and Volker Schulz. Shape gradients and their smoothness for practical aerodynamic design optimization. *Deutsche Forschungsgemeinschaft*, 2008.
- [60] Thomas W. Sederberg, David L. Cardon, G. Thomas Finnigan, Nicholas S. North, Jianmin Zheng, and Tom Lyche. T-spline simplification and local refinement. *ACM Transactions on Graphics*, 23(3):276, aug 2004.
- [61] Thomas W. Sederberg, Jianmin Zheng, Almaz Bakenov, and Ahmad Nasri. T-splines and t-NURCCs. *ACM Transactions on Graphics*, 22(3):477, jul 2003.
- [62] Songqing Shan and G. Gary Wang. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, 41(2):219–241, aug 2009.
- [63] Ching-Kuang Shene. Introduction to computing with geometry notes. Technical report, Department of Computer Science (Michigan Technological University), 2014.
- [64] Bharat K. Soni and Shaochen Yang. NURBS-based surface grid redistribution and remapping algorithms. *Computer Aided Geometric Design*, 12(7):675–692, nov 1995.
- [65] Hendrik Speleersa, Carla Mannib, and Francesca Pelosib. From nurbs to nurps geometries. *Computer Methods in Applied Mechanics and Engineering*, 255:238–2549, 2013.
- [66] Zhili Tang, Jacques Périaux, and Jun Dong. Constraints handling in nash/adjoint optimization methods for multi-objective aerodynamic design. *Computer Methods in Applied Mechanics and Engineering*, 271:130–143, apr 2014.

- [67] University Corporation for Atmospheric Research. *netCDF*. <http://www.unidata.ucar.edu/packages/netcdf/>.
- [68] Lijue Wang. *A NURBS-based Computational Tool for Hydrodynamic Optimization of Ship Hull Forms*. PhD thesis, George Mason University, 9 2015.
- [69] Shenren Xu, Sebastian Timme, Orest Mykhaskiv, and Jens-Dominik M $\tilde{A}$ $\frac{1}{4}$ ller. Wing-body junction optimisation with CAD-based parametrisation including a moving intersection. *Aerospace Science and Technology*, 68:543–551, sep 2017.
- [70] Guangxu Yu, Jens-Dominik M $\tilde{A}$ $\frac{1}{4}$ ller, Dominic Jones, and Faidon Christakopoulos. CAD-based shape optimisation using adjoint sensitivities. *Computers & Fluids*, 46(1):512–516, jul 2011.
- [71] Q. Zhang and S. Cen. Computational fluid dynamics in aerospace field and CFD-based multidisciplinary simulations. In *Multiphysics Modeling*, pages 295–328. Elsevier BV, 2016.