SVEUČILIŠTE U SPLITU

FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

POSLIJEDIPLOMSKI STUDIJ STROJARSTVA

KVALIFIKACIJSKI ISPIT

PARAMETRIZACIJA SKENIRANIH PLOHA ZA IZOGEOMETRIJSKU ANALIZU I OPTIMIZACIJU OBLIKA

Luka Nikolić

Split, rujan 2026.

UNIVERSITY OF SPLIT

FACULTY OF ELECTRICAL ENGINEERING, MECHANICAL ENGINEERING AND NAVAL ARCHITECTURE

DOCTORAL STUDIES IN MECHANICAL ENGINEERING

QUALIFYING EXAM

PARAMETRIZATION OF SCANNED SURFACES FOR ISOGEOMETRIC ANALYSIS AND SHAPE OPTIMIZATION

Luka Nikolić

Split, September 2026

CONTENTS

1. INTRODUCTION

Geometric modelling plays an important part in computational engineering, in computer-aided design, in computer-aided manufacturing and in computer-aided engineering alike. A geometric model is either created from nothing in modelling software, where a designer defines a new object, or recovered from an object that already exists. Whichever way it is obtained, the model is what everything downstream consumes. It is what a machine tool is driven from, what an analysis is run on and what an optimizer is permitted to change, so the choices made while it is built are felt long after the modelling itself is finished.

Geometric modelling is the branch of engineering computation concerned with how such shapes are represented, constructed and manipulated inside a computer. It grew out of the needs of the automotive, aircraft and shipbuilding industries, where the surfaces to be described were free-form and no catalogue of planes, cylinders and spheres could cover them, and where the drawings on paper carried shape information that no analytic formula reproduced [1, 2]. What the field settled on is the parametric representation. Chapter 2 gives an introduction to the standard parametric models used to describe such free-form curves and surfaces.

Reverse engineering is the process of obtaining a parametric description of an object that already exists [3]. Ship hulls, propeller blades, suspension components and worn or hand-finished tooling are digitized by optical or laser scanners, and what arrives is a dense triangulation of measured points, in some cases noisy and in most cases far larger than any description an engineer would write by hand. Such data record the shape but carry none of the structure a parametric model has, and no numerical procedure of any sophistication can be run on them directly. They have to be given a set of shape parameters that is compact and yet sufficient to stand for the measurement, which is what Chapter 3 covers, from the parametrization of the triangulated surface through the fitting of a spline surface to the routes by which such a fit is improved.

Following these descriptions of surfaces, isogeometric analysis was established. Finite element analysis in the traditional sense is the industry standard for the structural analysis of any component, a beam, a shell or a solid body. It works by meshing the structure into elements, and that step separates the analysis from the geometry, since the mesh is a second approximation of a shape the model already describes exactly. Isogeometric analysis, which is comparatively recent, works instead on the principle of integrating shape and analysis, and it uses the parametric description of the surface itself as the basis of the analysis [4]. Chapter 4 covers the basics of isogeometric analysis together with its benefits and its shortcomings, and the reasons why it has not displaced the finite element method as the industry standard to date.

Once a model can be analysed without being converted into anything else, it can also be improved. Chapter 5 treats shape optimization, where the control points that describe the shape become the variables an optimizer moves, and where the analysis of one iterate decides the next. Chapter 6 closes the review with the directions that follow from it. The five areas and the order in which they depend on one another are mapped in Figure 1.1.

Taken together the chapters describe one object rather than five subjects. The same spline surface runs through all of them. It is written in the language of Chapter 2, it is built in Chapter 3 out of a measurement that has none of its structure, it becomes the discretization of the analysis in Chapter 4 without being converted into anything else, and its control points become the design variables in Chapter 5. The chapters are therefore stages in the life of a single model, and what is decided in one of them is inherited by all that come after. That is also where the interest of the subject lies, since the control points a designer moves are the degrees of freedom the analysis

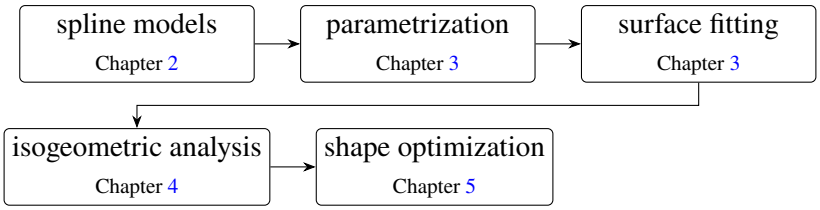


Figure 1.1. Map of the reviewed research areas and their dependencies.

solves for and the variables an optimizer steps along, and the three roles cannot be separated once the model is shared. This review is meant to give an insight into each of these areas, from the classical spline models through the parametrization and fitting of scanned surfaces to isogeometric analysis and shape optimization, at the depth needed to follow how one stage conditions the next. Alongside the established theory it reports what is new in each of them, the adaptive spline models that extend the tensor-product patch, the nonlinear routes by which a fitted surface is improved, the isogeometric treatment of thin shells and the growing use of isogeometric analysis in stress-based, topological and multidisciplinary optimization.

2. PARAMETRIC CURVES AND SURFACES

Free-form geometry that cannot be described by elementary analytic primitives is represented in CAD by piecewise polynomial or rational parametric models. This chapter surveys the standard models (Bézier, B-spline and NURBS) together with the fundamental algorithms for their manipulation. The three are nested rather than independent, as Figure 2.1 shows, since a Bézier curve is a B-spline with no interior knots and a B-spline is a NURBS whose weights are all equal to one. The exposition follows the standard references of Piegl and Tiller [5], Farin [2], de Boor [6] and Rogers [1]. Beyond their role as geometric containers, these models and their refinement algorithms are the substrate of everything that follows in this review.

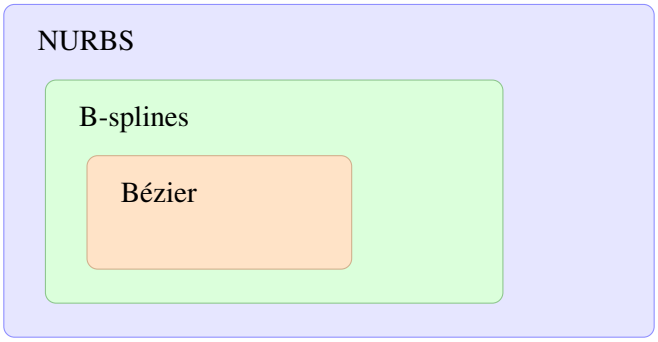


Figure 2.1. Relation between the standard parametric models.

2.1. Bézier Curves and Surfaces

The building blocks of the Bézier model are the *Bernstein polynomials* of degree p,

$$B_{i,p}(u) \;=\; \binom{p}{i} u^{i}(1-u)^{p-i}, \qquad i = 0, \dots, p, \quad u \in [0,1], \tag{2.1}$$

shown in Figure 2.2 for degrees $p = 2$ and $p = 3$. Their properties carry over to everything built on them [2, 5]. They are non-negative on $[0, 1]$ and form a partition of unity, $\sum_i B_{i,p}(u) = 1$ for every u, which follows from the binomial expansion of $[\,u + (1-u)\,]^p = 1$. Each $B_{i,p}$ attains exactly one maximum, at $u = i/p$, so the functions are ordered along the interval like the data they will weight. At the ends of the interval only one function survives, $B_{0,p}(0) = 1$ and $B_{p,p}(1) = 1$, with all others vanishing there. The basis is symmetric, $B_{i,p}(u) = B_{p-i,p}(1-u)$, and satisfies the recursion $B_{i,p}(u) = (1-u)\,B_{i,p-1}(u) + u\,B_{i-1,p-1}(u)$, a degree-raising analogue of Pascal's triangle that already hints at the evaluation algorithm below.

2.1.1. The Bézier Curve and its Properties

A Bézier curve of degree p blends $p+1$ control points $\boldsymbol{P}_i \in \mathbb{R}^3$ with these polynomials [2, 5],

$$\boldsymbol{C}(u) \;=\; \sum_{i=0}^{p} B_{i,p}(u)\,\boldsymbol{P}_i, \qquad u \in [0,1]. \tag{2.2}$$

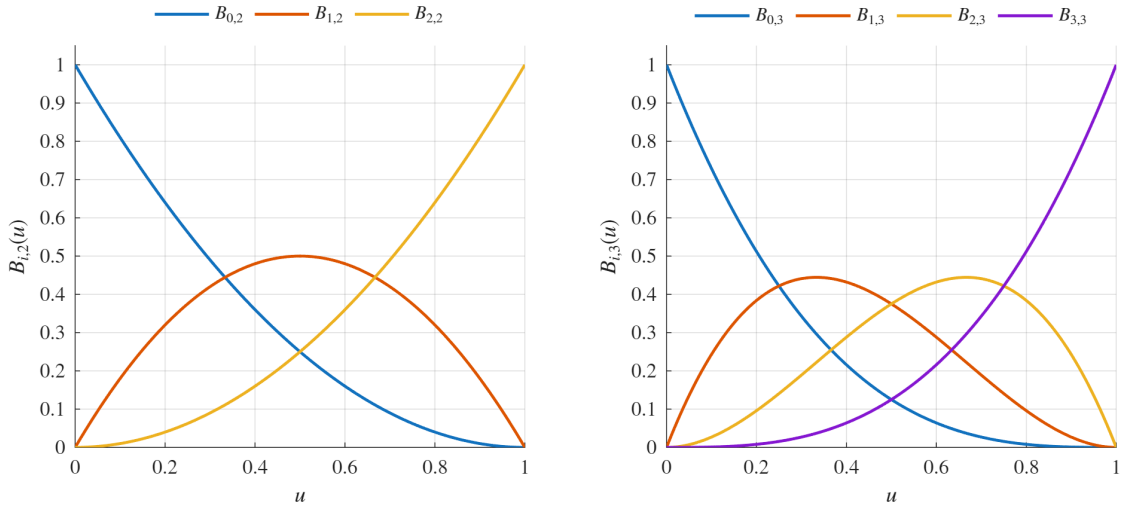


Figure 2.2. Bernstein polynomials of degree $p = 2$ and $p = 3$.

Figure 2.3 (left) shows a concrete cubic example with control points $\boldsymbol{P}_0 = (0, 0)$, $\boldsymbol{P}_1 = (1, 2.2)$, $\boldsymbol{P}_2 = (3, 2.6)$ and $\boldsymbol{P}_3 = (4, 0.4)$. The curve starts exactly at $\boldsymbol{P}_0$ and ends exactly at $\boldsymbol{P}_3$, leaves the first point along the polygon leg $\boldsymbol{P}_0\boldsymbol{P}_1$, arrives at the last point along the leg $\boldsymbol{P}_2\boldsymbol{P}_3$, and never exits the shaded convex hull of the four points. The interior control points act as magnets that the curve approaches but, in general, does not touch. The right panel of the same figure shows the flip side of the construction. Since every Bernstein polynomial is non-zero on all of $(0, 1)$, moving the single control point $\boldsymbol{P}_2$ changes the curve everywhere, not only near the moved point.

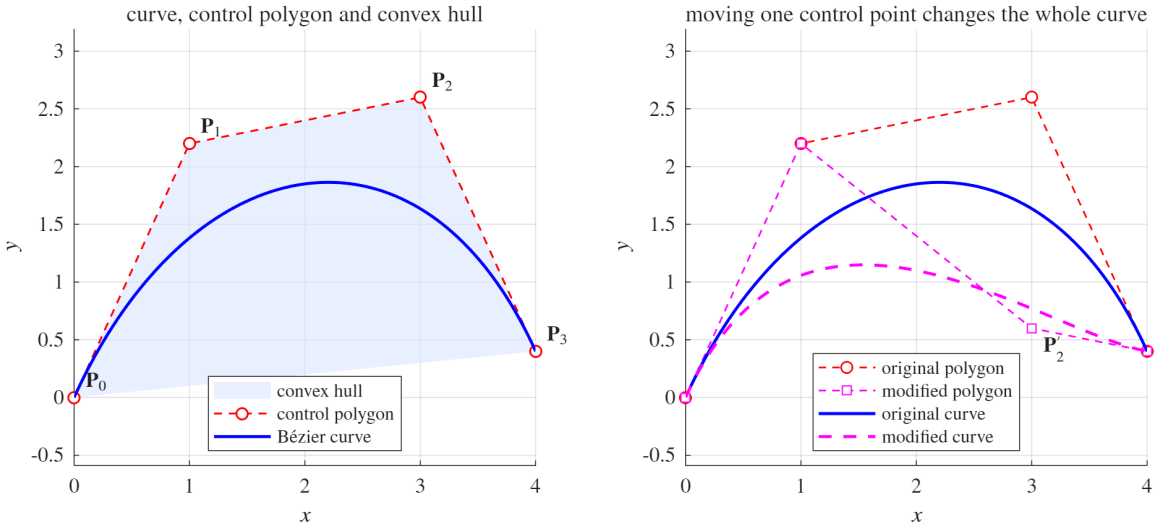


Figure 2.3. A cubic Bézier curve and the effect of moving one control point.

The behaviour visible in the example is general, and the basic properties of the Bézier curve are worth collecting explicitly [2, 5].

1. Endpoint interpolation. $\boldsymbol{C}(0) = \boldsymbol{P}_0$ and $\boldsymbol{C}(1) = \boldsymbol{P}_p$ hold because the Bernstein basis collapses to a single function at each end. Interior control points are approximated, not interpolated.

2. End tangency. $\boldsymbol{C}'(0) = p\,(\boldsymbol{P}_1 - \boldsymbol{P}_0)$ and $\boldsymbol{C}'(1) = p\,(\boldsymbol{P}_p - \boldsymbol{P}_{p-1})$, so the end legs of the control polygon are tangent to the curve. More generally, the derivatives at an endpoint

up to order r depend only on the $r+1$ control points nearest to it, the key to joining Bézier segments with prescribed smoothness.

3. Convex hull. Every curve point is a convex combination of the control points, so the curve lies in the convex hull of its control polygon. In practice this yields cheap and rigorous bounding volumes for intersection and collision queries, and it guarantees that a fitted curve cannot stray arbitrarily far from a well-placed control polygon.

4. Affine invariance. Applying an affine map to the control points and evaluating gives the same result as evaluating and then mapping, because affine maps commute with convex combinations.

5. Variation diminishing. No straight line (plane, in $\mathbb{R}^3$) intersects the curve more often than it intersects the control polygon [2]. The curve is never more "wiggly" than its control polygon. A monotone polygon produces a monotone curve, and a convex polygon a convex curve. This is the fundamental reason why control-point models behave so well in interactive design, and why spurious oscillations in a fit always betray themselves in the control net first. This diagnostic role of the control net recurs in the fitting experiments of Chapter 3, where Gibbs-like oscillations of an overstrained single-patch fit are visible in the control net before they are measurable on the surface.

6. Symmetry. Reversing the order of the control points produces the same curve traversed in the opposite direction, a consequence of the symmetry of the Bernstein basis.

7. Global control. Every basis function is supported on the whole domain, so every control point influences every curve point (Figure 2.3, right). This is the structural limitation that motivates the B-spline generalization below.

It is worth noting that a Bézier curve can also be evaluated geometrically, through repeated linear interpolation between neighbouring control points. Each round produces a polygon with one vertex fewer, and after p rounds a single point remains, which is the curve point $\boldsymbol{C}(u)$. The construction is known as the de Casteljau algorithm, and its derivation and variants are treated in detail in the standard references [2, 5]. Two of its properties are worth retaining here. Every intermediate quantity is a convex combination of control points, which makes the evaluation numerically stable, and the intermediate points obtained at a parameter value $\bar{u}$ are themselves the control points of the two curve segments lying on either side of $\bar{u}$. The curve itself is not altered by this splitting, only its description is refined, which is the same mechanism that reappears in a more general setting as knot insertion in Section 2.4.1.

2.1.2. Bézier Surfaces

A curve is a map of a single parameter into space, whereas a surface requires two. The standard way of building a bivariate model out of univariate ingredients is the *tensor product*, and since it underlies every surface model in this review, it is worth stating once in the simplest possible setting before it is reused with more elaborate bases.

The construction rests on one observation. If $\{B_{i,p}(u)\}_{i=0}^{p}$ is a basis of univariate functions in the first parameter and $\{B_{j,q}(v)\}_{j=0}^{q}$ a basis in the second, then the pairwise products

$$B_{ij,pq}(u,v) \;=\; B_{i,p}(u)\,B_{j,q}(v), \qquad i=0,\dots,p, \quad j=0,\dots,q, \tag{2.3}$$

form a basis of bivariate functions on the rectangle $[0,1]^2$ [2, 5]. There are $(p+1)(q+1)$ of them, one for every pair of indices, and each is a bump that rises and falls in both parameter directions at once. The two directions are treated entirely independently, and in particular the degrees p and q need not be equal. Figure 2.4 shows the construction with a cubic factor in the first direction and a quadratic factor in the second. The two univariate polynomials are drawn on the two vertical side planes and their product lies over the unit square between them, a single smooth bump whose peak sits where the two factors peak at the same time. Because both factors are bounded by one, the product never rises above either of them, which the common vertical scale of the figure makes immediate.

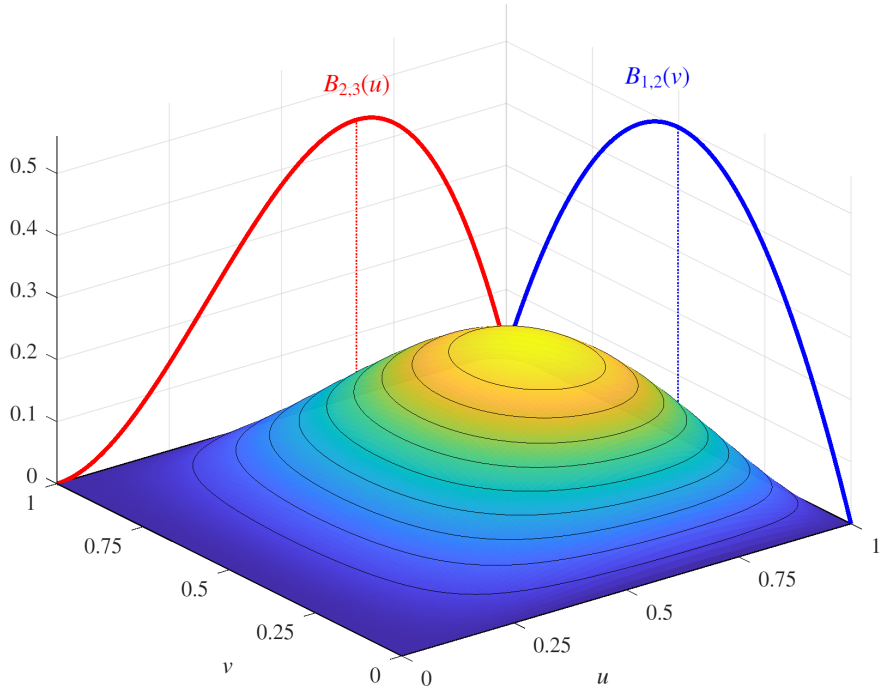


Figure 2.4. A cubic and a quadratic Bernstein polynomial and the bivariate basis function formed by their product.

What makes the construction useful is that the properties of the factors survive the multiplication. A product of non-negative functions is non-negative, and the sum over all products factorizes into a product of sums,

$$\sum_{i=0}^{p}\sum_{j=0}^{q} B_{i,p}(u)\,B_{j,q}(v) \;=\; \Big(\sum_{i=0}^{p} B_{i,p}(u)\Big)\Big(\sum_{j=0}^{q} B_{j,q}(v)\Big) \;=\; 1, \tag{2.4}$$

so a bivariate basis assembled in this way is again non-negative and again a partition of unity. Every point of the resulting surface is therefore a convex combination of its coefficients, which carries the convex hull property, affine invariance and the whole geometric intuition of the curve case over to surfaces with no further argument.

Since there is one basis function for every index pair, there is also one coefficient for every index pair. The coefficients are consequently no longer arranged along a polygon but on a rectangular grid, the *control net* $\boldsymbol{P}_{ij}$ [5]. Blending the net with the product basis gives the Bézier

surface of bi-degree (p, q),

$$\boldsymbol{S}(u, v) \;=\; \sum_{i=0}^{p} \sum_{j=0}^{q} B_{i,p}(u)\, B_{j,q}(v)\, \boldsymbol{P}_{ij}, \qquad (u, v) \in [0, 1]^2. \tag{2.5}$$

Figure 2.5 shows a bicubic example with its 4×4 control net.

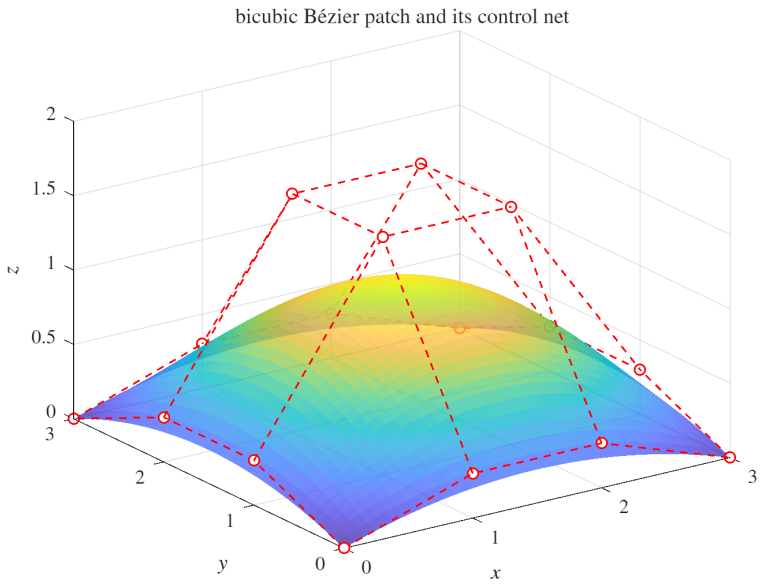


Figure 2.5. A bicubic Bézier surface patch and its control net.

The double sum becomes easier to read if one parameter is held fixed. Freezing v at a value v_0 and regrouping the terms,

$$\boldsymbol{S}(u, v_0) \;=\; \sum_{i=0}^{p} B_{i,p}(u)\, \boldsymbol{Q}_i, \qquad \boldsymbol{Q}_i \;=\; \sum_{j=0}^{q} B_{j,q}(v_0)\, \boldsymbol{P}_{ij}, \tag{2.6}$$

shows that the points $\boldsymbol{P}_{i0}, \ldots, \boldsymbol{P}_{iq}$ sharing the index i define an ordinary Bézier curve in v, that each of these $p + 1$ curves is evaluated at v_0 to produce a point $\boldsymbol{Q}_i$, and that those points then serve as the control points of an ordinary Bézier curve in u. A tensor-product surface is in this sense a curve whose control points themselves travel along curves, and sweeping v_0 across its range sweeps that curve across the surface. The same reading holds with the roles of the two parameters exchanged. The curves obtained by fixing one parameter are the *isoparametric curves* of the surface, and they form the natural grid along which such a surface is drawn and evaluated [2].

The properties of the Bézier curve transfer to the patch almost without exception. The four corner control points are interpolated. The four boundary curves of the patch are themselves Bézier curves, defined by the boundary rows and columns of the net, so two patches can be joined by matching their boundary rows. The convex hull property and affine invariance hold verbatim. Only the variation diminishing property fails to generalize in any straightforward form, a known limitation of the tensor-product construction [2].

One structural consequence of the construction should be recorded already at this point, because it shapes a large part of the literature reviewed in the later chapters. A tensor-product surface is defined on a rectangular parameter domain and carries a rectangular grid of control

points, so as a piece of geometry it is always a deformed rectangle. Shapes that are not topologically rectangular, and surfaces whose feature lines do not run along the parametric directions, cannot be accommodated by a single patch of this kind [3, 7]. The two available remedies are to divide the shape into several patches or to relax the rigidity of the grid itself, and Section 3.3 outlines both.

The practical limitation of the Bézier model, visible already in Figure 2.3, is global control. The limitation is removed by the B-spline model, which pieces polynomial segments together with prescribed smoothness while keeping every basis function locally supported.

2.2. B-Spline Curves and Surfaces

2.2.1. B-Spline Basis Functions

A B-spline is a piecewise polynomial function. Its parameter domain is divided into intervals, on each of which the function coincides with a single polynomial of degree p, and the individual polynomial pieces are then joined at prescribed parameter values so that the assembled function attains a required degree of smoothness. Those joining values are the *knots* [5, 6]. A knot is therefore not a point in space and carries no direct geometric meaning of its own. It is a location in the parameter domain at which one polynomial segment ends and the next one begins. The non-decreasing sequence that collects them,

$$U = \{u_0, u_1, \ldots, u_m\}, \qquad u_i \leq u_{i+1}, \tag{2.7}$$

is the *knot vector*, and it is the combinatorial backbone of the entire construction [1, 5]. It fixes how many polynomial segments the curve consists of, where in the parameter domain those segments meet, how smoothly they are joined and how many control points the curve requires. These quantities are not independent of one another. A knot vector of $m+1$ knots together with a degree p determines exactly $n+1 = m-p$ basis functions, so that $m = n+p+1$, and fixing any two of the degree, the number of knots and the number of control points determines the third.

The knots are values of the curve parameter, so their absolute magnitudes are immaterial and only their relative spacing has any effect. Shifting all knots by a common constant, or scaling them all by a common positive factor, leaves the basis functions unchanged up to the corresponding renaming of the parameter. For this reason it is customary to normalize the knot vector so that the parameter runs over $[0, 1]$, which is the convention adopted throughout this work, but the definition itself admits any non-decreasing real sequence. Knot vectors spanning larger intervals arise naturally whenever the knots are derived from measured data, for instance when they are spaced in proportion to accumulated chord length so that the parametrization follows the distribution of the points being approximated (Chapter 3). A knot vector whose interior knots are equally spaced is called *uniform* and any other one *non-uniform* [5]. The freedom to distribute the interior knots unevenly is what allows a spline to concentrate its resolution in those regions where the geometry actually demands it, and it is the origin of the first letter of the NURBS acronym.

Knots may also repeat, and the number of times a value occurs in the knot vector is its *multiplicity* [6]. Each repetition removes one degree of freedom from the corresponding join, so multiplicity is the mechanism that governs the smoothness of the transition between neighbouring segments. A single sequence of numbers thus encodes both the subdivision of the parameter

domain and the continuity of the curve across it.

A knot vector whose first and last knots are repeated $p + 1$ times,

$$U = \{\underbrace{u_0, \ldots, u_p}_{0},\ u_{p+1}, \ldots, u_{m-p-1},\ \underbrace{u_{m-p}, \ldots, u_m}_{1}\}, \tag{2.8}$$

is called *clamped* (open) [5]. It makes the curve interpolate its end control points and its end tangents align with the end polygon legs, which is essential for the patch-boundary constructions used in multi-patch fitting (Chapter 3). The end basis functions of a clamped knot vector are interpolatory, which is what forces the curve through its end control points.

Given a knot vector, the i-th B-spline basis function of degree p is defined by the Cox-de Boor recursion [8, 9]

$$N_{i,0}(u) = \begin{cases} 1, & u_i \le u < u_{i+1}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.9}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) \ + \ \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u), \tag{2.10}$$

with the convention $0/0 = 0$. The recursion begins with the piecewise constant functions of (2.9), which merely record which knot span a parameter value belongs to, and then builds each function of degree p as a convex-combination blend of two functions of degree $p - 1$ defined on overlapping supports. Every application of the recursion widens the support by one knot span and raises the smoothness by one order, so degree and locality are traded against each other in a controlled manner. Figure 2.6 shows the resulting bases for degrees $p = 1, 2, 3$, each on a clamped knot vector carrying three unevenly spaced interior knots. Raising the degree widens the supports and smooths the functions, the uneven spacing makes the individual functions differ in width from one another, and the partition of unity is preserved throughout. The basis functions possess the properties that make B-splines a core foundation of geometric modelling and, later, of isogeometric analysis [5, 6].

1. Local support. $N_{i,p}(u) = 0$ outside $[u_i, u_{i+p+1}]$, so each function lives on exactly $p + 1$ knot spans. Conversely, on any given knot span $[u_j, u_{j+1})$ at most the $p + 1$ functions $N_{j-p,p}, \ldots, N_{j,p}$ are non-zero. Local support means that moving one control point edits only $p+1$ spans of the curve, and, in the analysis setting, that the stiffness matrix is sparse with bandwidth governed by p.

2. Non-negativity and partition of unity. $N_{i,p}(u) \ge 0$ and $\sum_i N_{i,p}(u) = 1$, so the convex hull and variation diminishing arguments of the Bézier case carry over locally, and each curve segment lies in the convex hull of its $p + 1$ active control points.

3. Controlled continuity. Across a knot of multiplicity k the basis is C^{p-k} continuous. A simple interior knot of a quadratic basis carries C^1 continuity and doubling it drops the continuity to C^0, a cubic basis passes through C^2, C^1 and C^0 in the same way, and multiplicity $p + 1$ splits the curve entirely. Figure 2.7 illustrates this mechanism on a quadratic basis. In the left panel the knot $u = 0.5$ is absent, so a single polynomial covers that value. In the middle panel it is present once, which subdivides the span and brings in one further function, and the basis stays smooth to the eye because C^1 continuity survives. In the right panel the knot is doubled, one function reaches unity with a visible corner, and the curve

is pulled onto the corresponding control point. Interior knot multiplicities therefore provide a direct, local mechanism for embedding tangent or curvature discontinuities (chines, ridges, sharp feature curves) in an otherwise smooth model, a mechanism exploited both in single-patch feature modelling and, at the multi-patch level, along segment boundaries (Section 3.3).

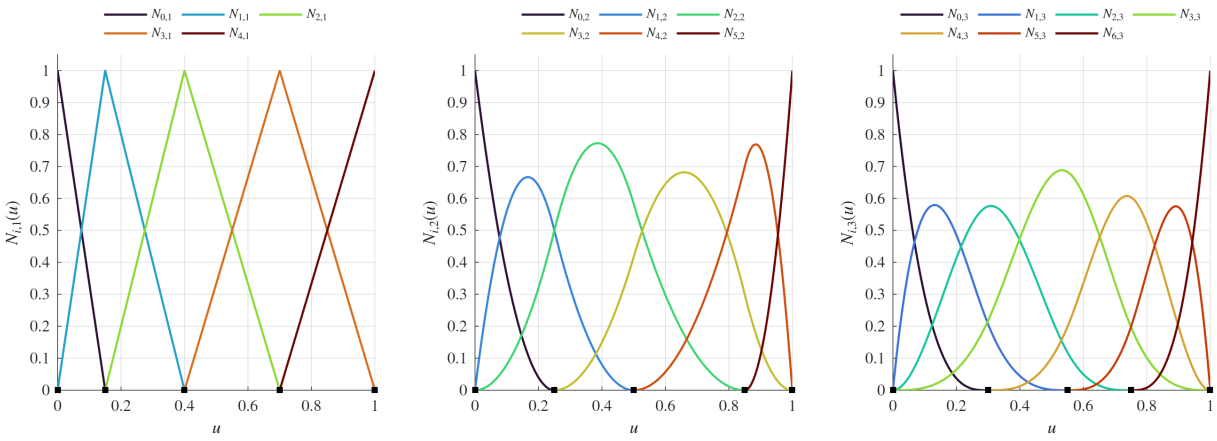


Figure 2.6. B-spline basis functions of degrees $p = 1, 2, 3$ on clamped non-uniform knot vectors.

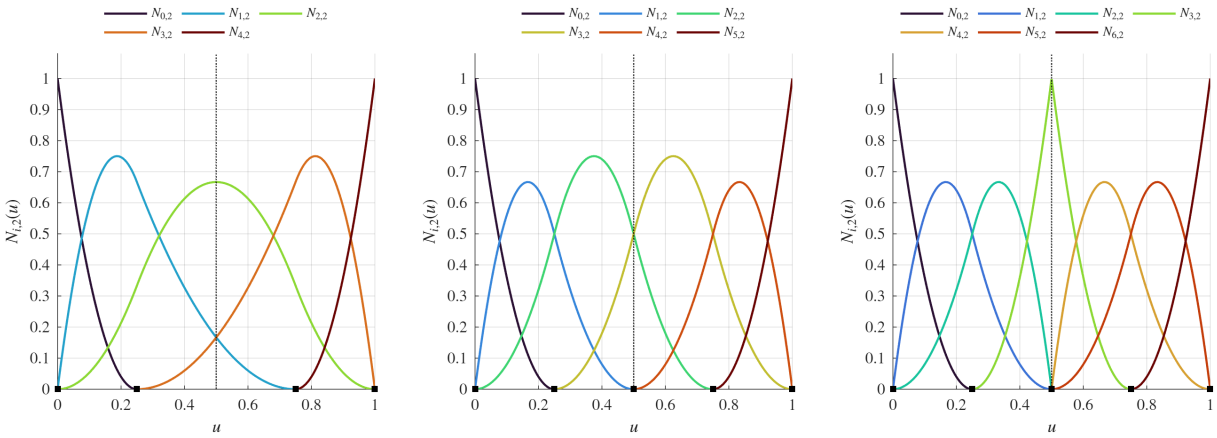


Figure 2.7. Quadratic basis functions with the knot $u = 0.5$ absent, simple and doubled.

The clamped knot vector also makes the relationship to the previous section precise. Consider a clamped knot vector containing no interior knots at all, so that the only values present are the two ends, each repeated $p + 1$ times. Such a vector holds $m + 1 = 2p + 2$ knots, the relation $n = m - p - 1$ leaves exactly $p + 1$ basis functions, and the parameter domain consists of a single knot span. Under these conditions the Cox-de Boor recursion returns nothing other than the Bernstein polynomials of (2.1), and the B-spline curve reduces to the Bézier curve of (2.2). A Bézier curve is therefore not a separate model but the special case of a B-spline curve whose parameter domain has never been subdivided, and the Bernstein basis is simply the B-spline basis of a single polynomial segment [2].

Read in the opposite direction, the same statement explains what the B-spline construction actually adds. Each interior knot introduced into the vector splits the parameter domain into one additional segment and, through the relation $m = n + p + 1$, brings in one additional control

point. The global control that limits the Bézier form is a direct consequence of having only one segment available, because with a single knot span every basis function is necessarily active across the whole domain. Introducing interior knots confines each function to $p + 1$ spans and produces the local support listed above. The step from Bézier to B-spline is accordingly not a change of representation but a refinement of the same one, carried out entirely through the knot vector and at a fixed polynomial degree, which is the property that knot insertion in Section 2.4.1 exploits systematically.

The step to surfaces needs no new machinery. The tensor-product construction of Section 2.1.2 carries over unchanged once the Bernstein polynomials are replaced by B-spline basis functions. Each parameter direction brings its own degree and its own clamped knot vector, so the two univariate families are $N_{i,p}(u)$ built on U and $M_{j,q}(v)$ built on V, and the bivariate basis functions are again their pairwise products,

$$N_{ij,pq}(u, v) \;=\; N_{i,p}(u)\, M_{j,q}(v). \tag{2.11}$$

Figure 2.8 shows one such function, drawn in the same way as the Bernstein pair of Figure 2.4 so that the two constructions can be compared directly. The two factors lie on the side walls, their product forms the bump on the base, and all three share one vertical scale. The difference from the polynomial case is immediate. A Bernstein factor is non-zero across the whole interval, so its product covers the entire square, whereas a B-spline factor of degree p vanishes outside $p + 1$ knot spans and the product is therefore confined to the rectangle

$$[u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}] \tag{2.12}$$

of knot spans, outlined on the base of the figure. Everywhere else that basis function contributes nothing at all. The two knot vectors jointly partition the parametric domain into a grid of rectangular cells, and over any single cell exactly $(p+1)(q+1)$ of the products are non-zero, so the surface is a separate polynomial patch on every cell.

2.2.2. Derivatives of the Basis Functions

Describing geometry requires the basis functions alone. A curve or a surface is completely determined by its control points and the values $N_{i,p}(u)$, and no derivative is needed to store, evaluate or render it. Derivatives enter only when the model is put to work beyond geometry, and there they are unavoidable. Tangents and surface metrics in parametrization and fitting, membrane strains and curvature changes in shell analysis, and design sensitivities in shape optimization all reduce to derivatives of the same basis (Chapters 3, 4 and 5). Since the differentiation rule is a property of the basis rather than of any particular application, it belongs here rather than in the analysis chapters, and it is stated once in full.

Differentiating the Cox-de Boor recursion (2.10) shows that the derivative of a degree-p function is a scaled difference of two functions of degree $p - 1$ on the *same* knot vector,

$$N'_{i,p}(u) \;=\; \frac{p}{u_{i+p} - u_i}\, N_{i,p-1}(u) \;-\; \frac{p}{u_{i+p+1} - u_{i+1}}\, N_{i+1,p-1}(u). \tag{2.13}$$

Applying that rule repeatedly gives the derivative of any order as a recursion that lowers the

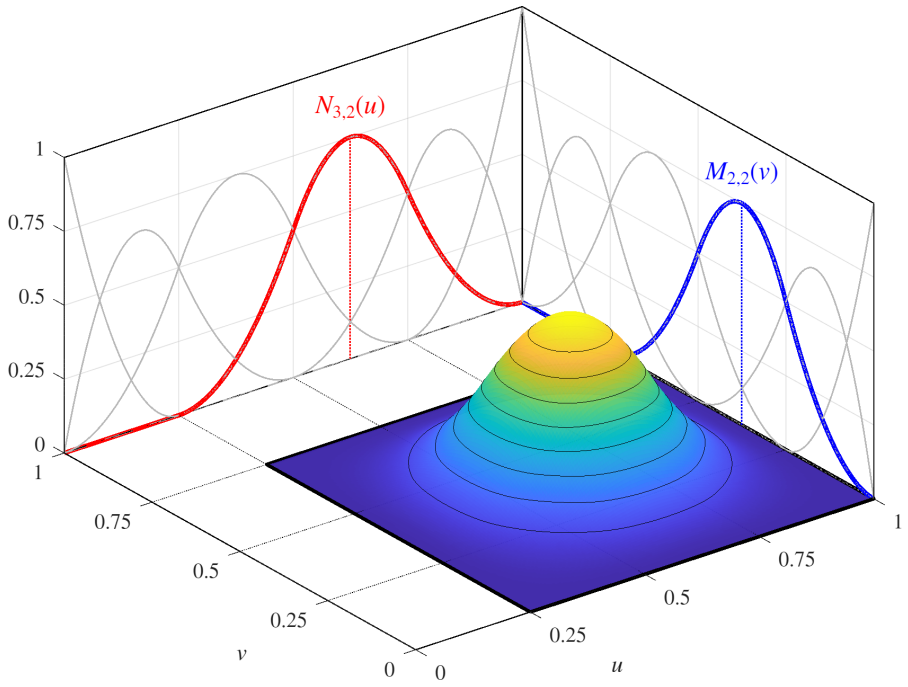


Figure 2.8. Two quadratic B-spline basis functions and the bivariate basis function formed by their product.

degree by one at each step,

$$N_{i,p}^{(r)}(u) \;=\; p\left(\frac{N_{i,p-1}^{(r-1)}(u)}{u_{i+p}-u_i} \;-\; \frac{N_{i+1,p-1}^{(r-1)}(u)}{u_{i+p+1}-u_{i+1}}\right), \tag{2.14}$$

with the convention that a term with a vanishing denominator is omitted [5]. Two consequences are worth noting. Each step of (2.14) combines only functions supported inside $[u_i, u_{i+p+1}]$, so a derivative of any order vanishes outside that interval exactly as the function itself does, and differentiation therefore preserves local support and the sparsity of every matrix assembled from these functions. And because the recursion terminates at degree zero, $N_{i,p}^{(r)} \equiv 0$ for $r > p$, so a degree-p basis carries exactly p non-trivial derivatives. This is the algebraic reason why a formulation requiring second derivatives, such as the Kirchhoff-Love shell of Chapter 4, needs $p \geq 2$.

Figure 2.9 shows a cubic basis together with its first three derivatives, on a clamped knot vector whose interior knot at $u = 0.5$ is doubled while those at 0.25 and 0.75 are simple. The sequence makes the continuity statement of the property list visible, and the two kinds of knot behave differently. The functions themselves are continuous everywhere and so are the first derivatives. The second derivatives are piecewise linear and merely bend at the simple knots, but at the doubled knot they jump, because only C^1 continuity survives there and the loss of smoothness arrives one derivative earlier than at a simple knot. The third derivatives are piecewise constant and jump from span to span at every knot. Each differentiation costs exactly one order of continuity, so a function that is C^{p-k} at a knot of multiplicity k has an r-th derivative that is only C^{p-k-r} there [5]. Two further features of the plots follow from the identities above. The derivative curves come in balanced positive and negative pairs, because differentiating the partition of unity gives $\sum_i N'_{i,p}(u) = 0$ at every parameter value, so a spline built on this basis reproduces constants with zero derivative exactly. The vertical scale also grows sharply with the

order, by roughly the reciprocal of the knot spacing per differentiation, which is the origin of the familiar conditioning penalty of higher-order derivative terms in stiffness matrices for thin structures.

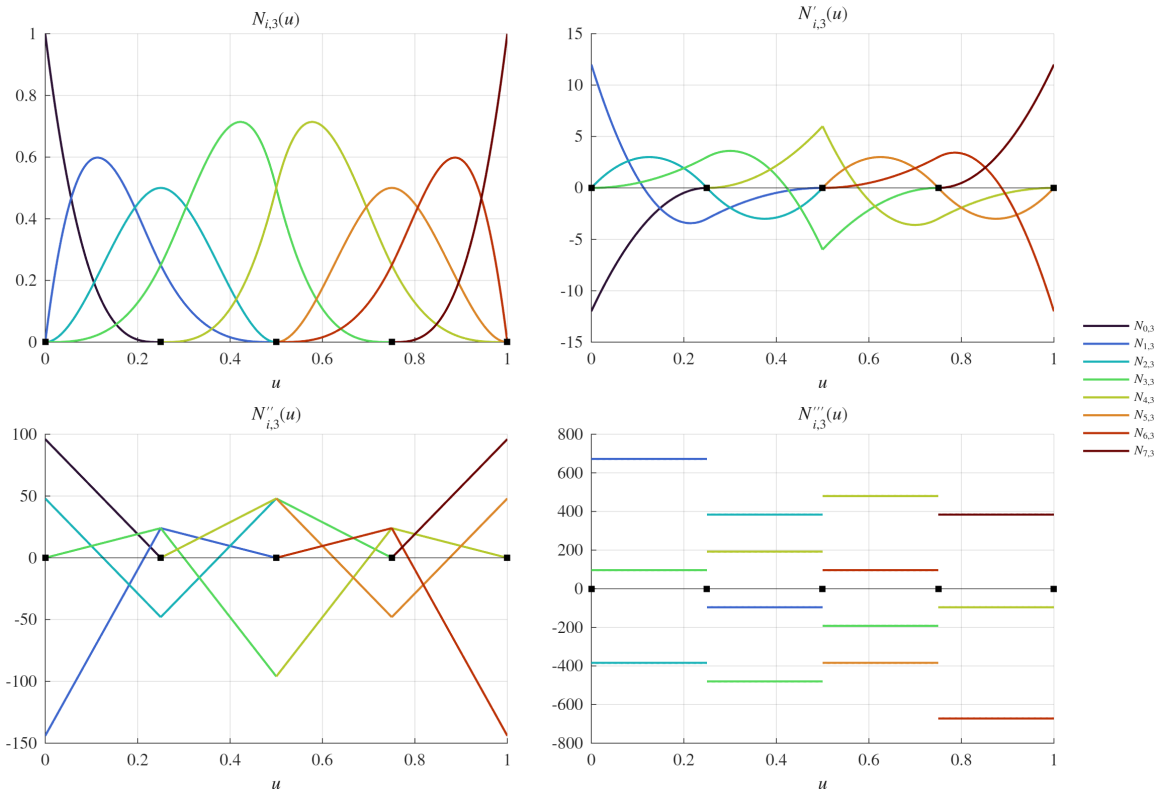


Figure 2.9. Cubic basis functions and their first three derivatives on a knot vector with the interior knot $u = 0.5$ doubled.

The bivariate basis needs no new machinery at all. A tensor-product function (2.11) is a product of one factor in each parameter, and each factor depends on its own parameter alone, so differentiating with respect to one parameter differentiates that factor and leaves the other standing as a constant multiplier,

$$\frac{\partial^{r+s} N_{ij,pq}}{\partial u^r \, \partial v^s}(u, v) \;=\; N^{(r)}_{i,p}(u)\; M^{(s)}_{j,q}(v). \tag{2.15}$$

Mixed partial derivatives of any order are therefore plain products of the univariate derivatives already delivered by (2.14), and no separate rule is required. This complete separation is a property of the polynomial basis alone. It does not survive the passage to the rational basis, where a denominator shared by all functions couples the two parameter directions and the derivatives have to be built by a recursion instead, as Section 2.3.2 sets out.

Evaluating the recursion naively over all indices is wasteful, because at any parameter value only $p + 1$ of the functions are non-zero. The standard references therefore provide dedicated procedures that first locate the relevant knot span and then compute only the non-vanishing functions and their derivatives, given as algorithms A2.1 to A2.3 by Piegl and Tiller [5] and in equivalent form by de Boor [6]. They are stable, inexpensive and standard practice in CAD applications, and they are not treated in further detail in this review.

2.2.3. B-Spline Curves and Surfaces

A B-spline curve of degree p with $n+1$ control points is

$$\boldsymbol{C}(u) = \sum_{i=0}^{n} N_{i,p}(u)\, \boldsymbol{P}_i, \tag{2.16}$$

a direct generalization of the Bézier curve in which the Bernstein polynomials are replaced by the locally supported B-spline basis.

Figure 2.10 gives an overview of this piecewise construction for a cubic curve on the knot vector $U = \{0, 0, 0, 0, 1/4, 1/2, 1/2, 3/4, 1, 1, 1, 1\}$, which carries eight control points and four knot spans. Each panel shades one span, marks the four basis functions that are non-zero on it, and marks the four control points those functions weight together with the segment they produce. Across the simple knots at $1/4$ and $3/4$ the active set shifts by a single index, so two neighbouring segments share p of their $p+1$ control points. That overlap is what makes the curve continuous across the knot without any condition being imposed after the fact, since the segments are not glued together but built from partly the same data.

The interior knot at $1/2$ is deliberately doubled. There the active set shifts by two indices instead of one, so the two segments meeting at that knot share only two control points rather than three, and the curve is correspondingly less smooth across it. A knot of multiplicity k leaves C^{p-k} continuity, so this junction is C^1 where the simple knots are C^2. Because the multiplicity is still below the degree, no basis function reaches one there and the curve is not pulled onto a control point.

Blending a rectangular control net $\boldsymbol{P}_{ij}$, with $i = 0, \ldots, n$ and $j = 0, \ldots, m$, gives the tensor-product B-spline surface of bi-degree (p, q),

$$\boldsymbol{S}(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u)\, M_{j,q}(v)\, \boldsymbol{P}_{ij}. \tag{2.17}$$

Everything established for the Bézier patch holds here for exactly the same reasons. The product of two partitions of unity is again a partition of unity, so every surface point is a convex combination of the control points, the curve-of-curves reading of (2.6) applies verbatim with the B-spline basis in place of the Bernstein one, and the isoparametric curves are again obtained by fixing one parameter. With clamped knot vectors in both directions the surface interpolates the four corner control points, and each boundary curve of the patch is the B-spline curve defined by the corresponding boundary row or column of the net [5].

Figure 2.11 shows three arbitrary biquadratic B-spline surfaces together with their control nets and with the images of the knot lines drawn on the surfaces, which make the cell structure visible.

The partition occurs in both directions at once. The two knot vectors divide the parametric square into a grid of rectangular cells, and over each cell the surface coincides with a single polynomial patch. On one cell exactly $(p+1)(q+1)$ of the product functions are non-zero, so that patch is governed by a rectangular block of the same number of control points, and neighbouring cells share all but one row or column of that block. Figure 2.12 shows one cell of a biquadratic surface, the image of that cell bounded by four isoparametric curves, and the nine control points that determine it. This structure is not merely descriptive. In isogeometric

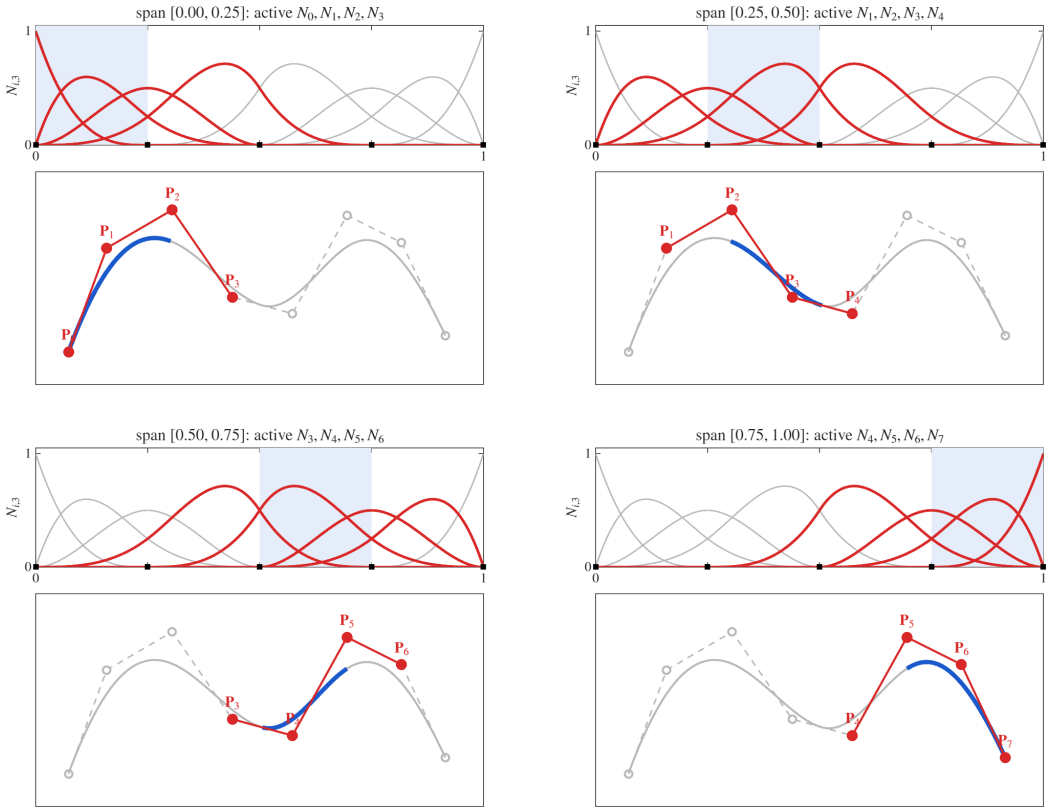


Figure 2.10. The four knot spans of a cubic curve, each with the basis functions and control points that govern it.

analysis the cells are the elements of the discretization and the control points governing a cell are the degrees of freedom of that element, which is the reading developed in Section 4.4.1.

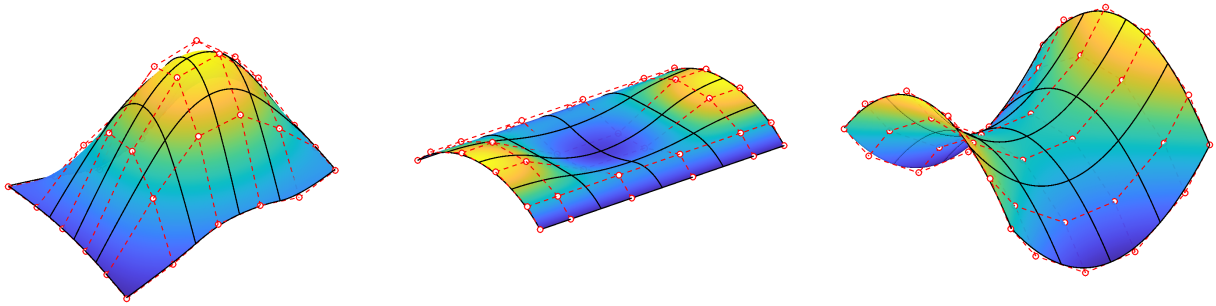


Figure 2.11. Three biquadratic B-spline surfaces with their control nets and knot lines.

The single most important practical consequence of the local support of the basis is *local control*. Figure 2.13 repeats, for a cubic B-spline curve with eight control points, the experiment of Figure 2.3, in which one control point is moved and the two curves are overlaid. This time the change is confined to the $p + 1$ knot spans on which the basis function of the moved point is supported, and the curve is unchanged elsewhere. A designer, a fitting algorithm or a shape optimizer can therefore adjust one region of the model without disturbing the rest [2, 6], and this is the property that makes B-splines, rather than high-degree Bézier curves, the working representation of interactive design and of the fitting methods reviewed in Chapter 3.

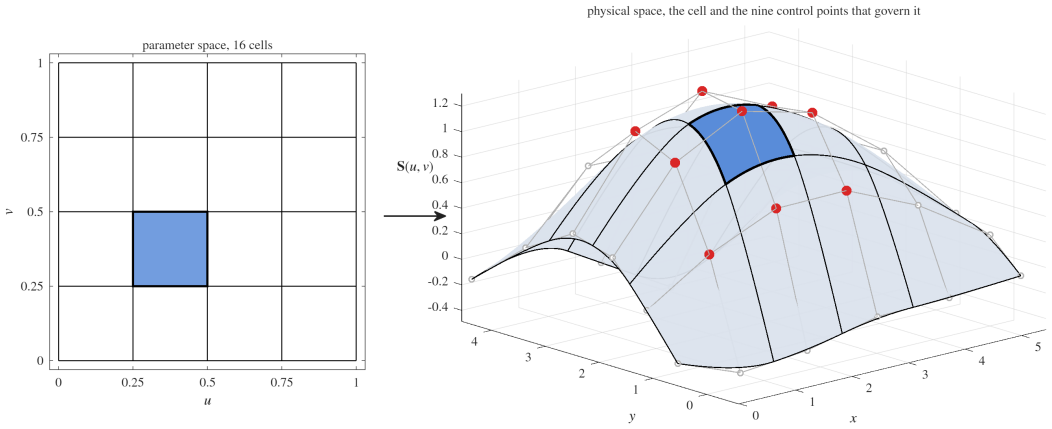


Figure 2.12. One cell of the parametric domain, its image on the same surface, and the nine control points that govern it.

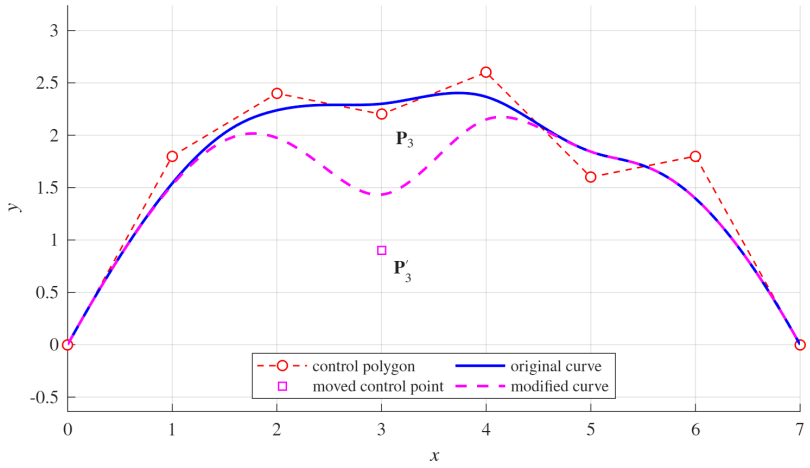


Figure 2.13. Local control of a cubic B-spline curve when one control point is moved.

2.3. NURBS

2.3.1. NURBS Basis Functions

B-splines are polynomial, and a polynomial cannot describe a conic section exactly. A circle can only be approximated, with a radius error that may be made small but never removed. The remedy is to associate a positive weight w_i with every control point and to divide by their weighted sum, which turns the polynomial basis into a *rational* one,

$$R_{i,p}(u) \;=\; \frac{w_i\, N_{i,p}(u)}{\displaystyle\sum_j w_j\, N_{j,p}(u)}, \tag{2.18}$$

and in two parameters, with the tensor-product functions of Section 2.2,

$$R_{ij,pq}(u,v) \;=\; \frac{w_{ij}\, N_{i,p}(u)\, M_{j,q}(v)}{\displaystyle\sum_k \sum_l w_{kl}\, N_{k,p}(u)\, M_{l,q}(v)}. \tag{2.19}$$

The summation indices carry a fixed meaning throughout. For a curve j runs over the control points, and for a surface k and l run over the rows and the columns of the control net. The rational basis keeps every property that mattered in the polynomial case [1, 5]. It is non-negative because the weights are positive, it sums to one at every parameter value by construction rather than as a theorem, it vanishes wherever the underlying B-spline function vanishes and therefore inherits local support unchanged, and it loses one order of continuity at a knot of each additional multiplicity exactly as before. If all weights are equal the quotient cancels and (2.18) returns the B-spline basis, so the polynomial model is the special case $w_i = \text{const}$ of the rational one.

What the weights add is a second family of shape parameters, independent of the control points. Because the denominator is shared, raising one weight inflates its own function and forces every other function to give way, and the effect is confined to the support of the function whose weight was changed. Figure 2.14 shows this on the basis that will describe a quarter circle below, a quadratic basis on the knot vector $U = \{0, 0, 0, 1, 1, 1\}$, which has a single knot span and therefore only three functions, and which reduces to the Bernstein polynomials of degree two when all weights are equal. Only the middle weight w_1 is varied. In the middle panel it carries the value $\sqrt{2}/2$, the one that will turn out to be exact, at the left it is reduced and at the right increased. The highlighted function shrinks and grows accordingly, and its two neighbours move in the opposite direction by exactly the amount needed to keep the three summing to one.

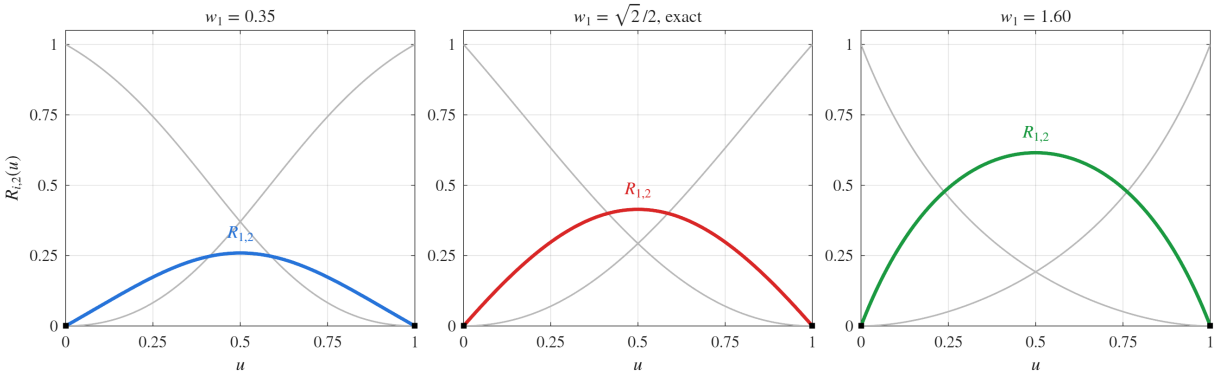


Figure 2.14. Quadratic rational basis functions for three values of the weight w_1.

2.3.2. Rational Basis Derivatives

Analysis needs the derivatives of the basis functions themselves, at a given parameter value, because these are the quantities from which strains, metrics and stiffness integrands are assembled. The standard references state the differentiation rule for the curve or the surface rather than for the individual function, so it is worth writing it in the form that is actually consumed.

Write the weight function of the denominator as $W(u) = \sum_j w_j N_{j,p}(u)$. The quotient rule applied to (2.18) gives the first derivative,

$$R'_{i,p}(u) \;=\; w_i \,\frac{N'_{i,p}(u)\,W(u) \;-\; N_{i,p}(u)\,W'(u)}{W(u)^2}, \qquad W'(u) = \sum_j w_j\, N'_{j,p}(u), \tag{2.20}$$

in which the polynomial derivatives $N'_{i,p}$ are those of Section 2.2.2. Differentiating the quotient repeatedly is impractical for higher orders. It is far cheaper to clear the denominator first, so that $R_{i,p}\,W = w_i N_{i,p}$, and to differentiate that product r times by the Leibniz rule, writing a for

the working index of the resulting sum. Solving the result for the highest derivative leaves a recursion in the order,

$$R_{i,p}^{(r)}(u) \;=\; \frac{1}{W(u)}\left(w_i\, N_{i,p}^{(r)}(u) \;-\; \sum_{a=1}^{r}\binom{r}{a} W^{(a)}(u)\, R_{i,p}^{(r-a)}(u)\right), \qquad W^{(a)} = \sum_j w_j\, N_{j,p}^{(a)}, \tag{2.21}$$

in which every order is obtained from the polynomial derivatives of the same order and from the rational derivatives already computed [5].

Three properties of (2.21) deserve notice, and each is a standard consequence of the rational form [5, 10]. The first is that the rational basis does not inherit the termination of the polynomial one. A degree-p B-spline function has only p non-trivial derivatives, whereas $R_{i,p}$ is a quotient of polynomials whose denominator stays positive, hence infinitely differentiable inside a knot span, so derivatives of every order exist and are generally non-zero [5]. The second is that local support survives differentiation, since $N_{i,p}^{(r)}$ vanishes outside $[u_i, u_{i+p+1}]$ and the recursion introduces no new support, so the sparsity of every matrix assembled from these functions is unaffected, which is the property the analysis literature relies on for the bandwidth of the discrete operators [10]. The third is a free correctness check. Differentiating the partition of unity gives

$$\sum_i R_{i,p}^{(r)}(u) \;=\; 0 \qquad \text{for every } r \geq 1, \tag{2.22}$$

so a rational basis reproduces constants exactly and any violation of (2.22) signals an error in the derivatives [5].

So far the functions have carried a single parameter. A surface needs the partial derivatives of the bivariate basis, and here the two directions no longer separate. The control net and the two knot vectors keep their tensor-product structure, and so does the numerator, but the denominator is shared and couples the parameters, so a rational basis function is not the product of two univariate rational functions. Writing the denominator of (2.19) as $W(u,v) = \sum_k \sum_l w_{kl} N_{k,p}(u) M_{l,q}(v)$ and clearing it in the same way, so that $R_{ij,pq} W = w_{ij} N_{i,p} M_{j,q}$, the Leibniz rule in two variables gives

$$R_{ij}^{(r,s)} \;=\; \frac{1}{W}\left(w_{ij}\, N_{i,p}^{(r)}(u)\, M_{j,q}^{(s)}(v) \;-\; \sum_{\substack{0\leq a\leq r,\ 0\leq b\leq s\\ (a,b)\neq(0,0)}} \binom{r}{a}\binom{s}{b} W^{(a,b)}\, R_{ij}^{(r-a,\,s-b)}\right), \tag{2.23}$$

with $W^{(a,b)} = \sum_k \sum_l w_{kl} N_{k,p}^{(a)}(u) M_{l,q}^{(b)}(v)$, in which r and s are the orders of differentiation in u and in v and a and b the working indices of the two Leibniz sums. For the first derivatives this reduces to

$$\frac{\partial R_{ij,pq}}{\partial u} \;=\; w_{ij}\,\frac{N_{i,p}'(u)\, M_{j,q}(v)\, W \;-\; N_{i,p}(u)\, M_{j,q}(v)\, W_{,u}}{W^2}, \tag{2.24}$$

and correspondingly in v.

The structure of (2.23) is worth reading carefully, because it is easy to expect more separability than is there. The numerator does separate. Differentiating with respect to u acts on the u factor alone and leaves the v factor untouched, exactly as in the polynomial case, where $\partial(N_{i,p} M_{j,q})/\partial u = N_{i,p}' M_{j,q}$ holds without qualification. The quotient does not. The denominator W depends on both parameters and does not in general factor into a function of u times

a function of v, so the bivariate rational function is not the product of two univariate rational functions, and its partial derivative is not the derivative of a univariate rational function multiplied by another one. The coupling between the two directions enters entirely through W and its derivatives. The exception is the case of weights that themselves factor into one contribution per parametric direction, for which W separates and the rational basis becomes a genuine tensor product again, but this is a special configuration rather than the rule.

Nothing further is needed for the geometry. Once the derivatives of the basis are available, those of a curve or a surface follow by differentiating (2.25) term by term, since the control points are constants, so that $\partial^{r+s}\boldsymbol{S}/\partial u^r \partial v^s = \sum_i \sum_j R_{ij}^{(r,s)} \boldsymbol{P}_{ij}$.

The polynomial derivatives that feed all of these expressions are produced by the triangular scheme of algorithm A2.3 of Piegl and Tiller [5], which returns the basis functions and their derivatives up to a prescribed order at one parameter value from a single table. The rational derivatives in the form required by analysis, that is per basis function rather than for the assembled geometry, are set out by Cottrell, Hughes and Bazilevs [10] and, with attention to the operation count, by Nguyen et al. [11].

2.3.3. NURBS Curves and Surfaces

Blending control points with the rational basis gives the NURBS curve and, with the bivariate basis and a rectangular control net, the NURBS surface,

$$\boldsymbol{C}(u) \;=\; \sum_i R_{i,p}(u)\,\boldsymbol{P}_i, \qquad \boldsymbol{S}(u,v) \;=\; \sum_i \sum_j R_{ij,pq}(u,v)\,\boldsymbol{P}_{ij}. \tag{2.25}$$

Since the basis is a partition of unity of non-negative functions, every point of the curve or surface remains a convex combination of the control points, so the convex hull property and affine invariance survive the passage to the rational form.

The reason the rational form exists is the exact representation of conics. A circular arc of opening angle $2\theta < \pi$ is reproduced exactly by a single quadratic rational segment whose middle control point sits at the intersection of the end tangents and carries the weight $\cos\theta$, the end weights being one [2, 5]. The quarter circle is the simplest case and needs nothing beyond a single segment. The data are three control points at the corners of the square the arc inscribes, the knot vector $U = \{0,0,0,1,1,1\}$ of one span, end weights of one and the middle weight $\cos 45^\circ = \sqrt{2}/2$. Larger conics are assembled from such segments, a full circle from four of them joined at doubled knots.

Figure 2.15 draws the curves produced by the three bases of Figure 2.14. All three share the same three control points and the same knot vector and differ only in the middle weight. Only one of them is a circular arc. Measured against the exact quarter circle, the curve with $w_1 = \sqrt{2}/2$ reproduces the radius to machine precision, while reducing the weight to 0.35 or raising it to 1.60 leaves radius errors of roughly 0.11 and 0.14 on a unit circle. The exactness is carried by the weight, not by the control polygon, which is the same in all three cases. The figure also makes the geometric meaning of a weight plain. As it grows the curve is drawn toward the middle control point and in the limit would pass through it, and as it falls the curve is pushed away, toward the chord joining the two end points.

The same construction carries over to surfaces of revolution. Sweeping a rational quarter circle through a quarter turn about an axis gives a biquadratic patch on the knot vectors $U = V =$

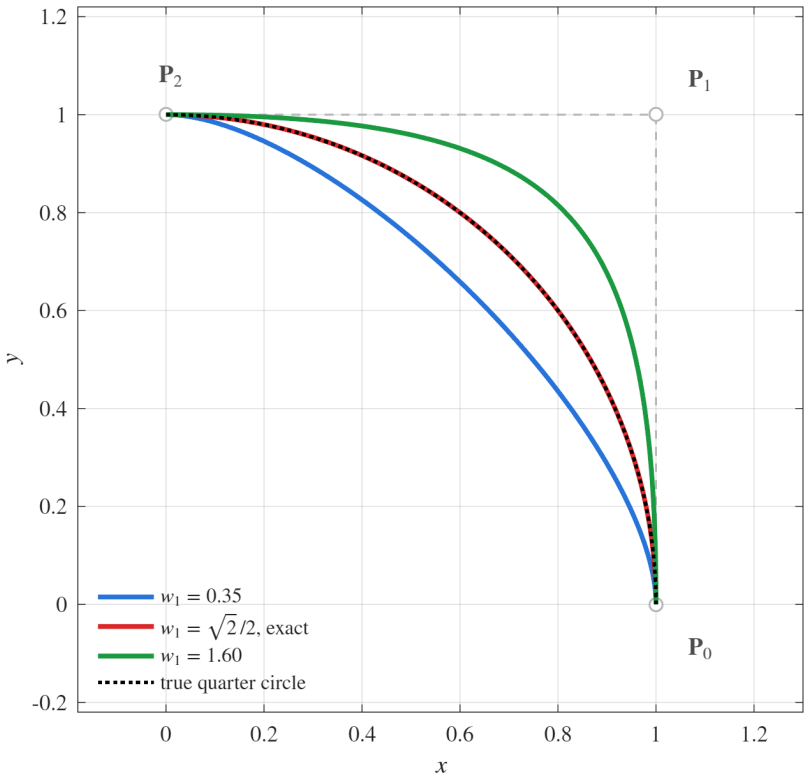


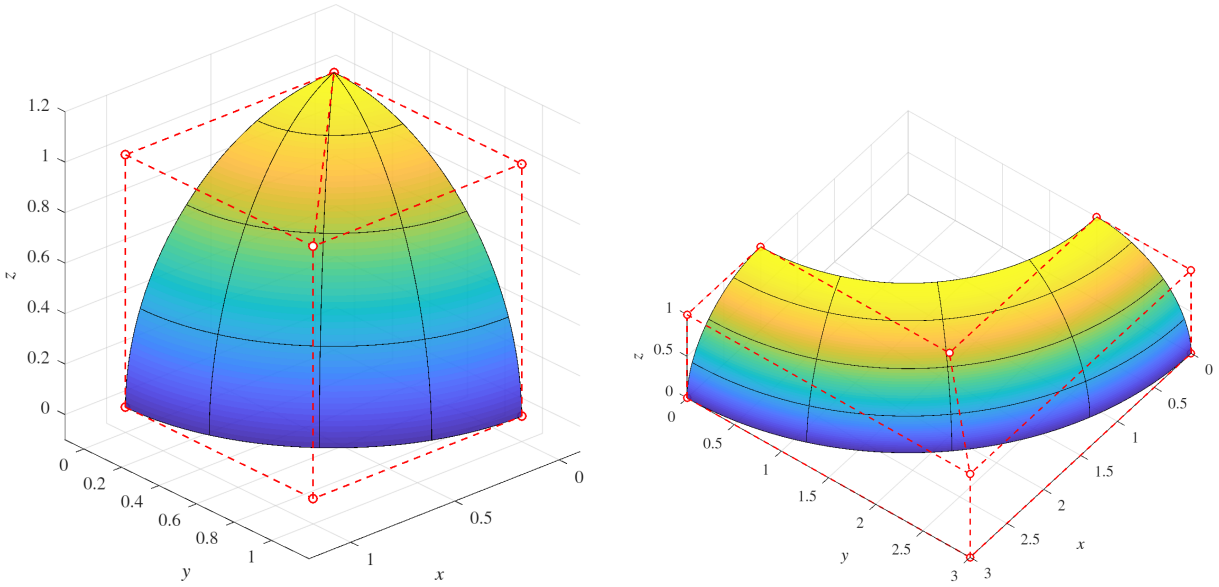
Figure 2.15. Quarter circle described by one control net and three values of the weight w_1.

$\{0,0,0,1,1,1\}$, in which the weight of every control point is the product of the corresponding generatrix and sweep weights [1, 5]. Figure 2.16 shows two patches built this way and Table 2.1 lists their complete data. The first is an octant of the unit sphere, swept from a meridian arc, whose last row of control points collapses onto the pole and leaves the patch degenerate along that edge. The second is a quadrant of a torus of major radius 2 and minor radius 1, swept from a quarter of the tube. Both are exact rather than approximate. The radius of the sphere patch and the tube radius of the torus patch reproduce their nominal values to machine precision over the whole parameter domain.

Table 2.1. Control points and weights of the two patches of Figure 2.16, after the surface of revolution construction [1, 5].

Both patches share $p = q = 2$, $U = \{0,0,0,1,1,1\}$, $V = \{0,0,0,1,1,1\}$			
$\boldsymbol{P}_{ij}, w_{ij}$	$j = 0$	$j = 1$	$j = 2$
(a) octant of the unit sphere			
$i = 0$	$(1,0,0), 1$	$(1,1,0), \sqrt{2}/2$	$(0,1,0), 1$
$i = 1$	$(1,0,1), \sqrt{2}/2$	$(1,1,1), 1/2$	$(0,1,1), \sqrt{2}/2$
$i = 2$	$(0,0,1), 1$	$(0,0,1), \sqrt{2}/2$	$(0,0,1), 1$
(b) quadrant of the torus, major radius 2, minor radius 1			
$i = 0$	$(3,0,0), 1$	$(3,3,0), \sqrt{2}/2$	$(0,3,0), 1$
$i = 1$	$(3,0,1), \sqrt{2}/2$	$(3,3,1), 1/2$	$(0,3,1), \sqrt{2}/2$
$i = 2$	$(2,0,1), 1$	$(2,2,1), \sqrt{2}/2$	$(0,2,1), 1$

This is why NURBS, and not plain B-splines, became the exchange standard of computer-aided design. Engineering geometry is full of exact circles, cylinders, cones and surfaces of



(a) Sphere octant.

(b) Torus quadrant.

Figure 2.16. Two NURBS patches with their control nets, an octant of a sphere (a) and a quadrant of a torus (b).

revolution, and any analysis that consumes design geometry directly, which is the central promise of isogeometric analysis [4, 10], has to speak the rational language.

2.4. Fundamental Refinement Algorithms

The refinement algorithms of the B-spline model manipulate the representation without changing the geometry, and acquire a second life in isogeometric analysis, where they implement h-, p- and k-refinement of the approximation space [4, 10]. They act linearly on the control points, so they carry over to NURBS unchanged provided each weight is carried along with the point it belongs to.

2.4.1. Knot Insertion

Inserting a knot $\bar{u}$ into the knot vector U enriches the basis and increases the space of curves that can be defined. The refined knot vector carries one basis function more, and therefore one control point more, while the spline space built on the coarse knot vector is contained in the one built on the refined vector [5]. The same curve is simply written down a second time in the larger basis, and only its control points have to be recomputed.

Let the inserted value $\bar{u}$ lie in the knot span $[u_k, u_{k+1})$. The refined control points $\boldsymbol{Q}_i$ are convex combinations of the coarse ones,

$$\boldsymbol{Q}_i \;=\; \alpha_i \, \boldsymbol{P}_i \;+\; (1 - \alpha_i) \, \boldsymbol{P}_{i-1}, \tag{2.26}$$

in which

$$\alpha_i = \begin{cases} 1, & i \le k-p, \\ \dfrac{\bar{u}-u_i}{u_{i+p}-u_i}, & k-p+1 \le i \le k, \\ 0, & i \ge k+1, \end{cases} \tag{2.27}$$

an algorithm due to Boehm [12]. The three cases carry the whole content of the formula. Where the coefficient is one the coarse point is taken over unchanged, where it is zero the coarse point reappears with its index shifted by one, and a genuine blend is formed only in the middle range. A single insertion therefore recomputes exactly p control points and leaves every other one untouched.

Figure 2.17 shows this on a cubic curve built on $U = \{0,0,0,0,0.2,0.4,0.6,0.8,1,1,1,1\}$ when the knot $\bar{u} = 0.5$ is inserted. Two of the eight control points are replaced by three new ones, the remaining six are carried over, and the curve is unchanged. The control polygon moves towards the curve, and repeated insertion drives it onto the curve, another echo of the variation diminishing property. The right panel compares the two bases. When several knots have to be inserted at once, the refined control points are produced in a single pass by the Oslo algorithm of Cohen, Lyche and Riesenfeld [13].

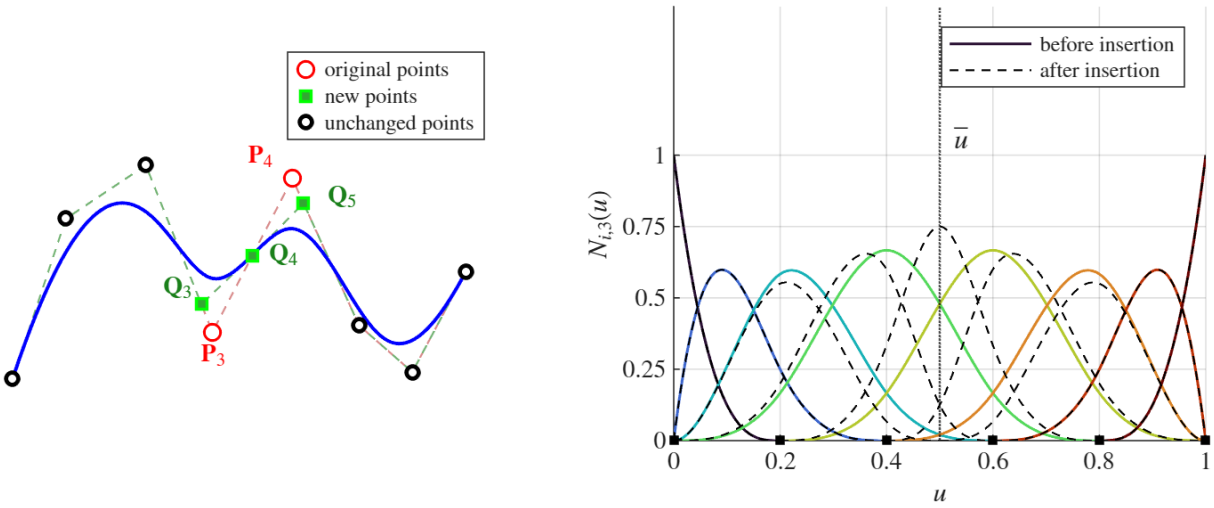


Figure 2.17. Cubic B-spline curve and its basis before and after inserting the knot $\bar{u} = 0.5$.

2.4.2. Degree Elevation

Degree elevation raises the polynomial degree by one and leaves the curve exactly where it is. The basis is enriched once more, this time along the degree rather than along the knot vector, and as in the previous section only the control points are recomputed.

The knot vector has to change as well, and the reason is continuity. At a knot of multiplicity k a spline of degree p is C^{p-k} continuous, so a spline of degree $p+1$ carrying that same multiplicity would be C^{p+1-k} there, one order smoother. Continuity of that kind is a property of the space and not of the coefficients, so *every* curve in such a space would be too smooth and no placement of control points could recover the original behaviour at the knot. Every distinct knot value therefore receives one further copy, the two ends included, which returns the continuity to $C^{(p+1)-(k+1)} = C^{p-k}$, exactly what the original curve had [5].

That single rule also fixes the size of the elevated model. Writing n_b for the number of distinct values in the knot vector, that is for the breakpoints of the piecewise polynomial, the elevated vector holds n_b knots more, so its last index becomes $\bar{m} = m + n_b$. Reading the relation $m = n + p + 1$ of Section 2.2.1 at the new degree then gives

$$\bar{n} \;=\; n + n_b - 1 \tag{2.28}$$

for the index of the last control point. The elevated space is genuinely larger than the original one, and the added freedom is spent on describing the same curve.

The elevated control points can be obtained without any dedicated elevation formula. The curve is the same object before and after, so

$$\sum_{i=0}^{n} N_{i,p}(u)\, \boldsymbol{P}_i \;=\; \sum_{j=0}^{\bar{n}} \bar{N}_{j,p+1}(u)\, \boldsymbol{Q}_j, \tag{2.29}$$

in which $\bar{N}_{j,p+1}$ are the basis functions built on the elevated knot vector. Both families of functions are known, since both knot vectors are known, and the control points $\boldsymbol{P}_i$ are known as well, so the only unknowns are the $\boldsymbol{Q}_j$. Evaluating (2.29) at as many parameter values as there are unknowns, chosen so that the resulting matrix is non-singular, turns the identity into a square linear system which is solved once for the elevated control points [5].

Dedicated procedures reach the same result without solving a system at all, among them the algorithm A5.9 of Piegl and Tiller [5] and the direct method of Prautzsch [14]. For NURBS the same procedures apply, but the rational curve or surface has first to be written as a non-rational B-spline in one dimension more, in homogeneous coordinates, and the elevated result projected back afterwards. The construction is set out in full by Piegl and Tiller [5] and is not covered in detail in this review.

Figure 2.18 shows a cubic curve elevated twice and Table 2.2 lists the knot vector at each step. Every elevation adds one copy of each distinct value, so the two interior multiplicities run $1, 2, 3$ and the two end multiplicities $4, 5, 6$, while the six control points become nine and then twelve, as (2.28) predicts. The polygon is redistributed and moves closer to the curve at every step, and the curve itself is preserved exactly. Raising the degree by more than one is nothing but a repetition of the same step, and the procedures cited above carry the repetitions out in a single sweep.

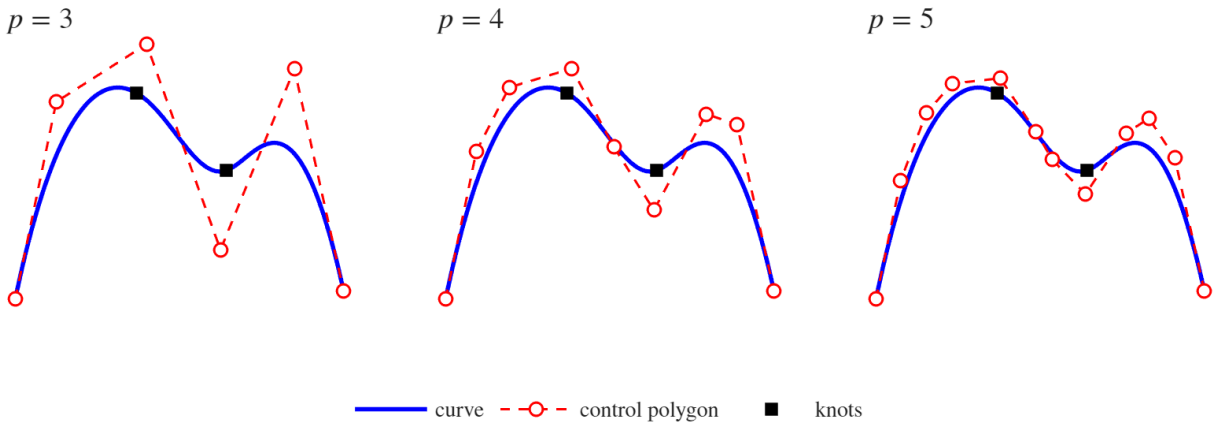


Figure 2.18. Cubic curve elevated to degree four and then to degree five.

Table 2.2. Knot vectors and control point counts of the three curves of Figure 2.18.

Degree	Knot vector	Control points
$p = 3$	$\{0, 0, 0, 0,\ 1/3,\ 2/3,\ 1, 1, 1, 1\}$	6
$p = 4$	$\{0, 0, 0, 0, 0,\ 1/3, 1/3,\ 2/3, 2/3,\ 1, 1, 1, 1, 1\}$	9
$p = 5$	$\{0, 0, 0, 0, 0, 0,\ 1/3, 1/3, 1/3,\ 2/3, 2/3, 2/3,\ 1, 1, 1, 1, 1, 1\}$	12

2.4.3. Knot Removal and Degree Reduction

Knot removal is the inverse of knot insertion. A knot is taken out of the vector, the space loses one function and the curve is written again with one control point fewer. Unlike insertion and elevation, this direction can fail. Those two enlarge the space and can therefore always reproduce the curve, whereas removal makes the space smaller, so the curve survives only if it is smoother across that knot than its multiplicity requires [5]. A knot that carries a genuine kink cannot be dropped without changing the shape.

In practice the test is made against a tolerance rather than exactly. A candidate knot is removed provisionally, the resulting curve is compared with the original, and the removal is kept when the deviation stays below a prescribed geometric tolerance. The knots are visited cheapest first, and the process stops when nothing further can be dropped within the tolerance. How the candidate control points are recomputed, and how the deviation is bounded without sampling the curve everywhere, is standard material which the references treat in full [5], so it is not set out in further detail here.

Degree reduction is the inverse of degree elevation and behaves in the same way. The degree drops from p to $p - 1$ and every knot multiplicity drops by one, which is the construction of Section 2.4.2 read backwards. Here too the target space is smaller than the one the curve lives in, so an exact result is the exception rather than the rule. A curve of degree p is also a curve of degree $p - 1$ only when its segments happen to be of the lower degree already, and in every other case the most that can be asked for is an approximation that stays within a tolerance. The procedure accordingly reports failure when the curve cannot be reduced within the tolerance allowed, and knot removal is one of its ingredients, since the surplus knot copies have to be taken out once the degree has been lowered. It is given as algorithm A5.11 of Piegl and Tiller [5], which lowers the degree by one, and a larger reduction is a repetition of the same step.

Insertion and elevation always succeed and make the model larger, whereas removal and reduction are the direction in which it is made smaller, and neither of them is exact. That is also where they earn their place. Removal reduces data after fitting, where an automatic knot placement is deliberately conservative and leaves more knots than the geometry needs [15], and the two together are the mechanism by which reverse-engineered models are brought down to a size that CAD systems handle comfortably.

2.5. Spline Models Beyond the Tensor Product

Everything so far rests on two assumptions that the tensor product carries with it. The first is that the domain of a patch is a rectangle, and the second is that refinement is global, because a knot inserted in one direction produces a whole row of control points across the patch. The consequence was already visible in the refinement algorithms of Section 2.4. Resolution spent on

one feature is paid for over the entire surface, and an object whose detail is unevenly distributed is described either too coarsely where it matters or too finely everywhere else. The literature that answers this divides into two lines of work. One keeps the rectangular patch and makes refinement local within it, the other gives up the rectangle and builds a smooth basis on a control mesh of arbitrary layout. Both are surveyed here because the fitting of Chapter 3 and the analysis of Chapter 4 draw on them.

The oldest of the local constructions is the hierarchy. Forsey and Bartels [16] overlaid a fine tensor-product patch on a coarse one and let the fine level carry a local displacement of the coarse surface, which gives detail where it is placed and leaves the rest of the model untouched. The construction is simple to state and awkward to use, since the overlaid levels are not linearly independent and the same shape has many representations. Kraft [17] removed that defect with a selection rule that keeps, from every level, only the functions whose support lies in the refined region, producing a basis that is linearly independent by construction. The truncated hierarchical B-splines of Giannelli, Jüttler and Speleers [18] go one step further and diminish the coarse functions over the refined region, which restores the partition of unity, reduces the overlap between levels and improves the conditioning of the resulting systems. This family is the best developed of the three, with adaptive refinement strategies, error estimators and convergence results collected in an overview by Bracco and co-workers [19], and it is the space on which the alternating and joint fitting methods of Section 3.3 are formulated [20].

The second local construction changes the control grid itself. A T-spline [21] admits T-junctions in the control net, so a row of control points may terminate in the interior of the surface instead of running across it, and refinement stays confined to the neighbourhood of the feature that asked for it. The price is that the blending functions of a general T-mesh need not be linearly independent, which is harmless in modelling but fatal in analysis, where a dependent basis produces a singular system. The subclass of analysis-suitable T-splines restricts the layout of the T-junctions so that linear independence and the partition of unity are guaranteed, and it comes with a local refinement algorithm that terminates and does not propagate refinement across the whole model [22]. Locally refined B-splines [23] arrive at a similar freedom from the other direction, by refining a box partition of the domain through the insertion of mesh line segments, with later work on the conditions under which the resulting functions stay locally linearly independent and on how to grade the refinement [24].

The constructions that abandon the rectangle answer a different difficulty, the topology of the object rather than the distribution of its detail. Subdivision surfaces define the geometry as the limit of repeated refinement of a control mesh of arbitrary connectivity, which admits extraordinary vertices where a number of faces other than four meet, and Cirak, Ortiz and Schröder [25] showed that the limit surface is smooth enough to serve as the basis of a thin-shell analysis. U-splines [26] pursue the same generality within spline technology, constructing a basis directly on an unstructured mesh whose cells may differ in degree and in the smoothness with which they are joined. A third route keeps ordinary tensor-product patches and repairs only what happens between them, by choosing the patch parameterizations so that functions defined patchwise join with geometric continuity across the interfaces, which yields a globally smooth space on a multi-patch domain [27, 28]. That construction belongs to the same discussion as the couplings of Chapter 4, with the difference that smoothness is built into the space rather than imposed on the solution afterwards.

None of these models has replaced NURBS in industrial practice. The trimmed tensor-product patch remains the format in which geometry is exchanged between systems, so the con-

structions above enter a review of this kind as technology that a reconstruction or an analysis may adopt, not as a standard either of them must follow.

3. SURFACE FITTING OF TRIANGULATED SURFACES

Many engineering components exist as physical objects whose digital description is missing, incomplete or lost, and their shape can be captured by a three-dimensional scanner. *Reverse engineering* is the task of recovering a structured parametric model from such a scan [3]. In this work the target of that recovery is a model of Chapter 2, a B-spline or NURBS surface, a map from a rectangular parametric domain into space in which every point is reached through a coordinate pair (u, v).

The recovery falls into two steps that make sense only together. A scan is a set of points in space, while the model of Chapter 2 is a map from a square, so the measured data must first be given a parametric structure and every point must receive its own pair (u, v). Once that chart exists the surface follows by approximation, because the control points that reproduce the data most closely are the solution of a linear system. The order is imposed by the fit, which cannot begin before its data carry parameters, and the dependence runs the other way as well, since the chart fixes how much of the parametric domain each part of the object occupies and therefore how many control points will fall on it. The two steps are taken in turn below.

3.1. Parametrization of the Mesh

The scanner delivers the object in a form far removed from that model. Its output is a triangulated surface mesh $\mathcal{M} = (V, F)$ with vertices $\boldsymbol{x}_i \in \mathbb{R}^3$ and triangles F, a faithful record of the shape that carries no parametric structure at all. To describe the object with a parametric model, every vertex must first be assigned planar coordinates $\boldsymbol{u}_i = (u_i, v_i)$ in a parametric domain $\Omega \subset \mathbb{R}^2$, so that each measured point is tied to the point of the parametric surface that is meant to reproduce it. This assignment is a *parametrization* of the mesh. The parametrization problem has been studied intensively in geometry processing since the mid-1990s, and the survey of Floater and Hormann [29] remains the standard entry point.

3.1.1. The Parametrization Problem

A parametrization is a piecewise linear map $\varphi : \mathcal{M} \to \Omega$, determined completely by the images $\boldsymbol{u}_i = \varphi(\boldsymbol{x}_i)$ of the vertices. On each triangle it reduces to the affine map that carries the triangle in space onto its planar image, so the whole map is known as soon as the vertices have been placed in the plane. Its inverse leads from the parametric domain back onto the surface, and that is precisely the map a fitted spline surface is meant to reproduce. Whatever the parametrization does to the geometry is therefore inherited by everything built on it.

Flattening a curved surface always costs something. The affine map on a single triangle is described by the two singular values $\sigma_1 \geq \sigma_2 > 0$ of its Jacobian, which carry a unit circle in the plane of the triangle into an ellipse with those semi-axes, and the classical ideals are stated through them [29].

1. *Isometric* maps have $\sigma_1 = \sigma_2 = 1$ and preserve both lengths and angles. They exist only for developable surfaces such as a cylinder or a cone, whose Gaussian curvature vanishes.

2. *Conformal* maps have $\sigma_1 = \sigma_2$ and preserve angles and the local shape of each triangle, but not its size.

3. *Equiareal* maps have $\sigma_1\sigma_2 = 1$ and preserve areas, but not angles.

A map is isometric exactly when it is both conformal and equiareal, and by Gauss's Theorema Egregium no such map exists wherever the Gaussian curvature differs from zero. Anything with genuine curvature is therefore distorted the moment it is flattened, which is familiar from cartography, where a world map either keeps angles or keeps areas but never both. The practical question is not how to avoid distortion but where to place it, and each of the methods below answers that question by minimizing an energy that penalizes one of the deviations above.

Figure 3.1 places the four situations side by side on one and the same surface. The small square in each panel is the parametric domain, which carries a uniform grid and a circle, and the arrow leads onto the surface through the inverse of the parametrization, where the lines drawn on the surface are the images of that grid, the isoparametric lines. A general map spaces those lines unevenly and carries the circle into an ellipse that covers a different area, an isometric map keeps the cells square and of unchanged size and returns the circle as it was, a conformal map keeps the cells square but enlarges them, so that the circle stays a circle and only grows, and an equiareal map stretches the cells in one direction and compresses them in the other, so that the circle becomes an ellipse which still covers the same area.

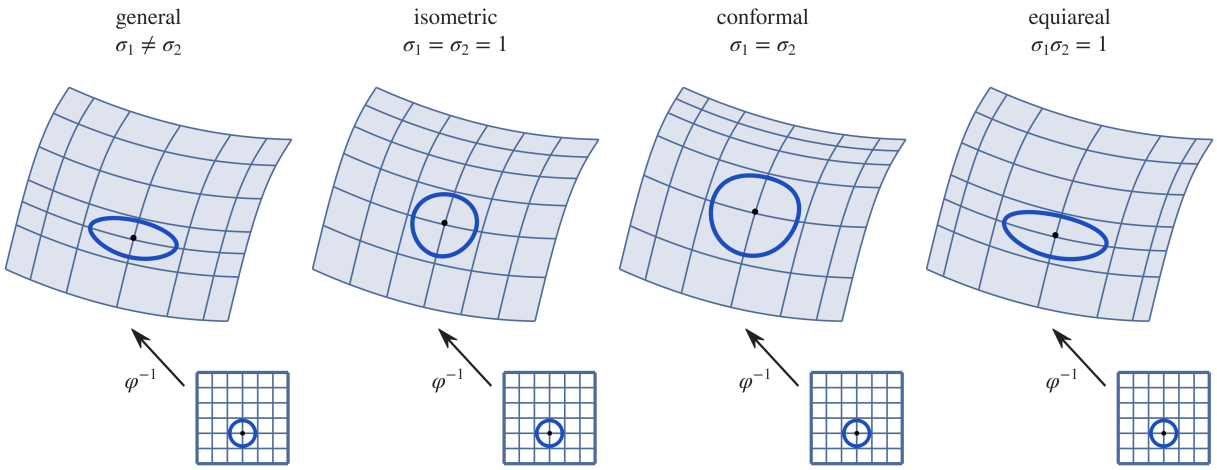


Figure 3.1. One surface carrying four different parametrizations, with the image of a circle drawn in the parametric domain.

For spline fitting the answer matters in a very concrete way. A tensor-product surface distributes its control points according to the *parametric* density of the data, so a uniform knot grid spends one knot span per equal interval of u and of v, no matter how much physical surface that interval covers. If the map compresses a strongly curved region such as a fillet or a leading edge into a small part of the parametric domain, that region is left with few control points and the fit is weakest exactly where the geometry is most demanding, while flat regions inflated by the map receive resolution they do not need. The parametrization thus acts as an implicit refinement strategy, decided before any fitting takes place, and methods that redistribute parametric area deliberately are built on this observation [30].

3.1.2. Convex Combination Maps

The oldest and most robust construction fixes the boundary of the mesh to a convex polygon and asks every interior vertex to be the weighted average of its neighbours,

$$\boldsymbol{u}_i \;=\; \sum_{j \in N(i)} \lambda_{ij}\, \boldsymbol{u}_j, \qquad \lambda_{ij} > 0, \qquad \sum_{j \in N(i)} \lambda_{ij} = 1, \tag{3.1}$$

where $N(i)$ collects the vertices adjacent to i. Written for all interior vertices, with the known boundary positions moved to the right-hand side, (3.1) becomes a pair of sparse linear systems, one for the u coordinates and one for the v coordinates. The parametrization is obtained by a single solve, nothing iterative is involved, and the whole method is characterized by the choice of the weights λ_{ij}.

The foundational result is due to Tutte [31], who proved in the context of graph drawing that a planar and 3-connected graph whose boundary is mapped to a convex polygon admits a valid straight-line embedding with uniform weights $\lambda_{ij} = 1/|N(i)|$, one in which no triangle degenerates or flips. Floater [32] generalized the theorem to arbitrary positive convex weights and introduced shape-preserving weights that reproduce the local geometry of the mesh, which established convex combination maps as the standard tool whenever the map must be valid by construction. His motivation was the approximation of triangulated surfaces by smooth spline surfaces, that is the reconstruction problem of this review. The guarantee is unconditional, but uniform weights ignore the geometry of the mesh entirely. Long edges and short edges pull equally, and the resulting map can compress fine detail severely.

A geometry-aware choice within the same linear framework follows from the minimization of the Dirichlet energy of the map, whose stationarity condition is again a system of the form (3.1), now carrying the *cotangent weights*

$$\lambda_{ij} \;\propto\; \cot \alpha_{ij} + \cot \beta_{ij}, \tag{3.2}$$

where α_{ij} and β_{ij} are the angles opposite the edge (i, j) in the two triangles that share it [33]. The map obtained this way is the discrete harmonic map, and its angular distortion is close to the smallest attainable with a fixed boundary. One caveat matters in practice. Cotangent weights stay positive only on triangulations close to Delaunay, and obtuse triangles can turn them negative, which takes the injectivity guarantee with them [29].

Figure 3.2 shows the three choices on a scanned boat hull. The four corners are pinned, the rest of the boundary slides along its side, and the three maps differ in nothing but the weights. The uniform weights ignore the geometry of the mesh, so parametric area follows the density of the triangulation rather than the area of the surface, and the chines of the hull wander through the domain as ragged bands. The shape-preserving and the harmonic weights follow the geometry and let the chines run as clean curves, and of the three the harmonic map keeps the angles of the triangles closest to their values on the surface. It pays for that with the guarantee. About one cotangent weight in five is negative on this mesh, and three triangles do reverse their orientation, all of them slivers at the boundary that together hold some six square millimetres of a hull almost four metres long. Small as that is, it is a genuine loss of injectivity, because the images of neighbouring triangles overlap there and a point of the square no longer corresponds to a single point of the surface, so the piece of geometry that the fold covers cannot be recovered from the map [29]. The two maps built on positive weights stay fold-free.

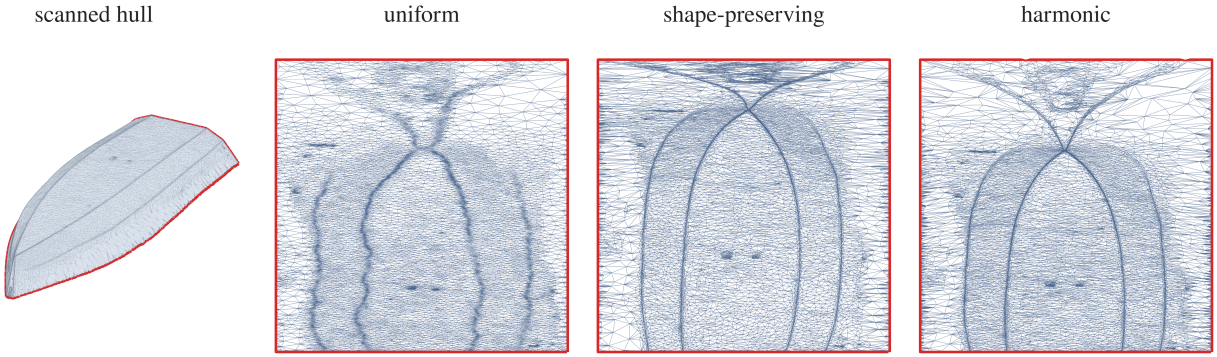


Figure 3.2. A scanned boat hull and its images in the unit square obtained with uniform, shape-preserving and harmonic weights.

3.1.3. Parametrization as Part of the Model

The methods above choose a map by criteria internal to geometry, validity and low distortion, and none of them asks what the map will be used for. That question returns one level higher. Once a spline surface has been fitted it carries a parametrization of its own, the map from the parametric square onto the geometry, and that map is not a record of how the surface was obtained but an ingredient of every computation performed on it afterwards, since it enters each integral through its Jacobian and its metric. Cohen et al. [34] named this concern *analysis-aware modeling*, and Xu et al. [35] showed on model problems that two parametrizations of one and the same geometry differ in discretization error by large factors. Figure 3.3 carries one quarter disk by two patches of the same kind. The first collapses a whole side of the parametric square onto the centre, so a row of elements degenerates there and the Jacobian of the map vanishes along it, which leaves the integrals of an analysis without meaning at that end of the domain. The second places the four corners of the square on the boundary of the disk and keeps every element a proper quadrilateral. The consequence for reconstruction is plain. The weights and the corners chosen in this chapter are inherited by the fitted surface and are therefore part of the model rather than preparation for it, yet they are settled before anything is known about what will be asked of the model, and no accepted measure predicts from the reconstruction alone what the choice will cost.

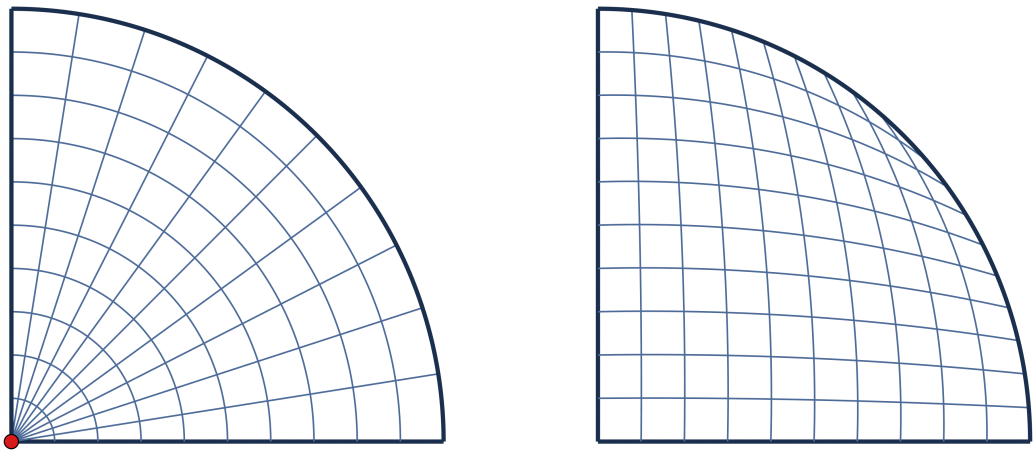


Figure 3.3. One quarter disk parametrized with a collapsed side (left) and with the four corners of the square on its boundary (right).

Fixed-boundary convex combination maps are settled theory, with unconditional injectivity

from Tutte and Floater [31, 32] and a geometry-aware weight choice in the discrete harmonic map [33]. The parametrization of triangulated surfaces has been studied extensively for three decades, and the literature holds many further constructions, among them free-boundary conformal maps [36, 37], area-preserving flattening [38] and feature-aligned metrics [39, 40], each of them answering in its own way the question of where the distortion should be placed. This review keeps to the basic idea of parametrization and to the simple constructions that carry it, because these already suffice to turn a scan into the parametric model of Chapter 2, and it is that model which makes a geometry recovered from a real object accessible to analysis.

3.2. Fitting a Spline Surface to the Data

With every measured point carrying a parameter pair, the reconstruction problem reduces to determining the control points of a spline surface that approximates the data, and possibly also its knots. The classical treatment is Piegl and Tiller [5], while Dierckx [15] and the fitting chapters of Farin [2] supply the approximation-theoretic background. The account below keeps to a single tensor-product patch, which is the setting in which the ingredients are clearest, and takes in turn how the data for the fit is produced, how the control points follow from a linear system and what is gained by releasing the unknowns that the linear system holds fixed.

3.2.1. The Topological Point Cloud

The fit does not see the mesh. It sees points, each carrying a parameter pair, and the map of the previous section is what produces them. Every vertex already has its planar coordinates, so the plainest choice is to fit the vertices themselves at those coordinates. Reported reconstruction practice takes one step more and resamples. A regular grid is laid over the parametric square, each grid point is located in the triangle whose planar image contains it, and the position of the corresponding point of the object follows by barycentric interpolation of the three vertices of that triangle. The result is a structured array $\boldsymbol{x}_{kl} = \boldsymbol{x}(\bar{u}_k, \bar{v}_l)$, the topological point cloud of representative mesh-based methods [41, 42], which turns scattered scan data into gridded data and makes tensor-product fitting natural. For curves the classical alternatives to a uniform assignment are chord-length and centripetal parametrization, both of which adapt the parameter speed to the geometry [5]. Closely related is the choice of *which* data to fit. Adaptive and curvature-aware sampling of dense meshes [43, 44] and feature-recognizing point selection [45] reduce the fitting problem to the points that carry information, which matters at scan resolutions of 10^5-10^7 points. In the gridded setting the corresponding choice is the resolution $n_u \times n_v$ of the sample grid, which has to oversample the control net enough to keep the linear system of the fit well conditioned, in practice by a factor of three to five. Recent work on engineering-scale B-spline surface reconstruction [46] and on sharp-feature-preserving reconstruction [47] documents the state of practice.

Figure 3.4 shows the construction on three scanned parts. Each row carries one object, on the left its triangulated mesh, in the middle the harmonic parametrization obtained as in Section 3.1 and on the right the topological point cloud that follows from sampling that square on a regular grid. The hull is the benign case, since the grid lands on it almost evenly. The propeller shows the coupling of the previous section in its plainest form, because the blade receives little parametric area and the same grid therefore samples it far more thinly than the hub.

Two properties of this step decide much of what follows. The first is that resampling inherits

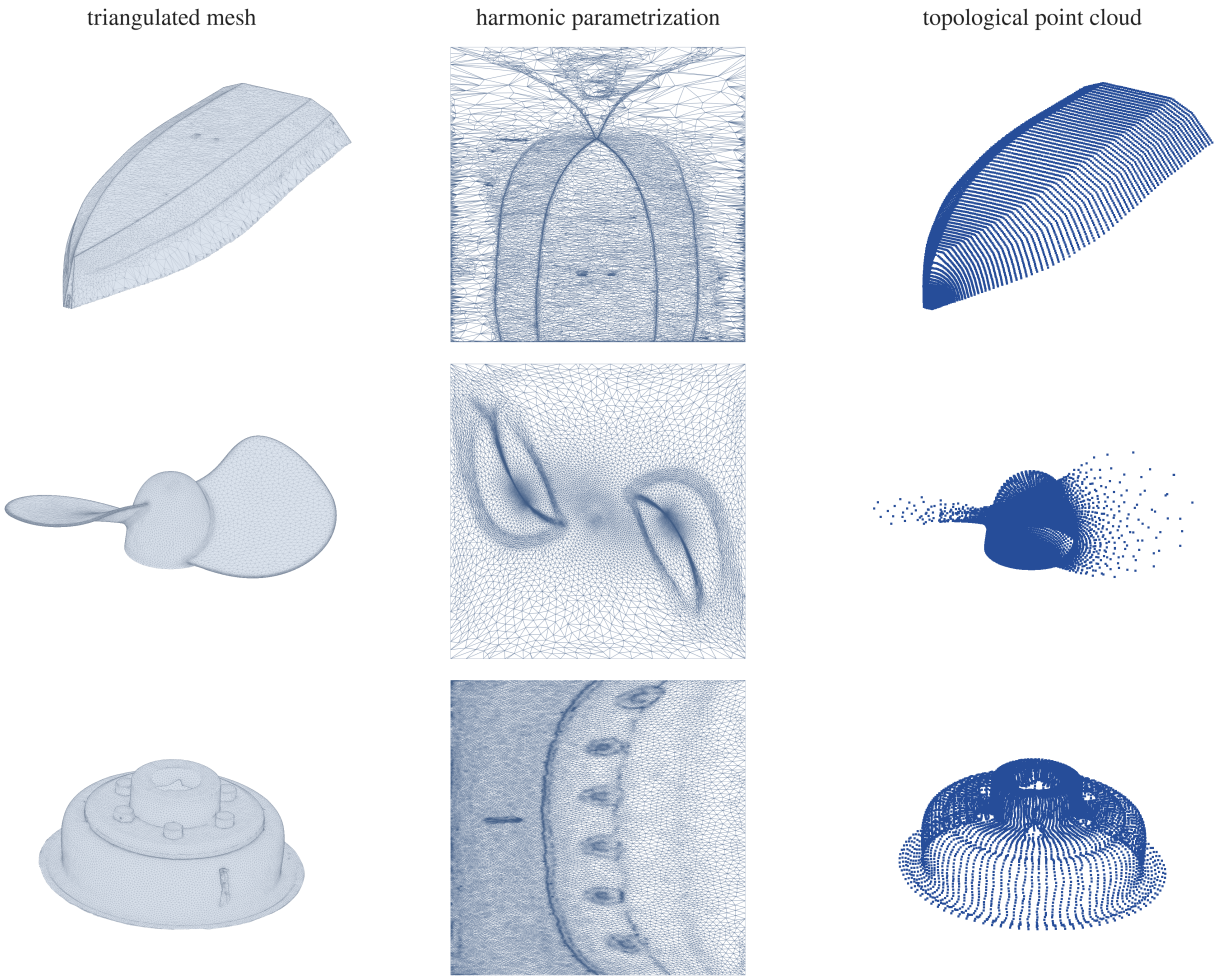


Figure 3.4. A scanned hull, propeller and clutch housing, their harmonic parametrizations and the topological point clouds sampled from them.

every decision the parametrization made, because a region that received little parametric area also receives few grid points, so the data itself and not only the basis is thinner there. The second is that to a linear fit an error in the parameter of a point is indistinguishable from an error in its position, which is why global reparametrization and the parameter correction of Section 3.3 improve fits that no adjustment of the control points alone could reach.

3.2.2. Linear Least-Squares Fitting and Regularization

Fitting means choosing the control points so that the surface passes as closely as possible to the data. Closeness is measured by the sum of the squared distances of the data points from the surface,

$$E(\bar{u}, \bar{v}, \boldsymbol{P}) \;=\; \sum_{k} \left\| \boldsymbol{S}(\bar{u}_k, \bar{v}_k) - \boldsymbol{x}_k \right\|^2 . \tag{3.3}$$

In this generality the error depends on two sets of unknowns, the parameters of the data and the control points, and it is nonlinear in the first of them because the parameters enter through the basis functions [5, 48]. The parametrization of Section 3.1 has already assigned the parameters, and once they are fixed the surface is linear in the control points, so (3.3) is a quadratic function of the control points alone and its minimization is a linear problem.

The steps are clearest on a curve. A B-spline curve of degree p with control points $\boldsymbol{P}_i$ is fitted to the data $\boldsymbol{x}_k$ at the parameters $\bar{u}_k$, with $i = 0, \ldots, n$ and $k = 1, \ldots, n_u$, and the error is

$$E(\boldsymbol{P}) \;=\; \sum_{k=1}^{n_u} \Big\| \sum_{i=0}^{n} N_{i,p}(\bar{u}_k)\, \boldsymbol{P}_i - \boldsymbol{x}_k \Big\|^2 . \tag{3.4}$$

The function is quadratic and bounded below by zero, so its minimum lies where the derivative with respect to every control point vanishes. Each term of (3.4) is the squared length of a residual, the difference between the curve at $\bar{u}_k$ and the data point $\boldsymbol{x}_k$, so the chain rule leaves twice that residual multiplied by its own derivative with respect to $\boldsymbol{P}_j$. Of the sum over the control points only the term $i = j$ contains $\boldsymbol{P}_j$, and its derivative is the number $N_{j,p}(\bar{u}_k)$, which is therefore the weight the residual of the k-th data point enters with,

$$\frac{\partial E}{\partial \boldsymbol{P}_j} \;=\; 2 \sum_{k=1}^{n_u} N_{j,p}(\bar{u}_k) \Big(\sum_{i=0}^{n} N_{i,p}(\bar{u}_k)\, \boldsymbol{P}_i - \boldsymbol{x}_k \Big) \;=\; \boldsymbol{0}, \qquad j = 0, \ldots, n. \tag{3.5}$$

Separating the unknown control points from the data gives one equation per control point,

$$\sum_{i=0}^{n} \Big(\sum_{k=1}^{n_u} N_{j,p}(\bar{u}_k)\, N_{i,p}(\bar{u}_k) \Big) \boldsymbol{P}_i \;=\; \sum_{k=1}^{n_u} N_{j,p}(\bar{u}_k)\, \boldsymbol{x}_k, \qquad j = 0, \ldots, n, \tag{3.6}$$

in which the bracket contains only basis values at the parameters of the data. Those values are gathered into the collocation matrix $\boldsymbol{A}$, whose row k holds the values of all basis functions at the parameter $\bar{u}_k$ and whose column i belongs to the control point $\boldsymbol{P}_i$,

$$\boldsymbol{A} \;=\; \begin{bmatrix} N_{0,p}(\bar{u}_1) & N_{1,p}(\bar{u}_1) & \cdots & N_{n,p}(\bar{u}_1) \\ N_{0,p}(\bar{u}_2) & N_{1,p}(\bar{u}_2) & \cdots & N_{n,p}(\bar{u}_2) \\ \vdots & \vdots & \ddots & \vdots \\ N_{0,p}(\bar{u}_{n_u}) & N_{1,p}(\bar{u}_{n_u}) & \cdots & N_{n,p}(\bar{u}_{n_u}) \end{bmatrix} \;\in\; \mathbb{R}^{\,n_u \times (n+1)} . \tag{3.7}$$

The bracket is then the entry (j, i) of $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$ and the right-hand side is the entry j of $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{x}$, so (3.6) is the system of normal equations and the fitted control points follow from it,

$$\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\, \boldsymbol{P} \;=\; \boldsymbol{A}^{\mathsf{T}}\boldsymbol{x}, \qquad \boldsymbol{P} \;=\; \big(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} \big)^{-1} \boldsymbol{A}^{\mathsf{T}}\boldsymbol{x}. \tag{3.8}$$

The control points and the data points carry three coordinates while the basis functions carry none, so (3.8) is one system solved three times, once for each coordinate, and a single factorization of $\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}$ serves all three. That matrix is symmetric and positive semidefinite by construction, and it is positive definite, so the fit is unique, exactly when $\boldsymbol{A}$ has full column rank.

A surface is fitted in the same way, and on the gridded data of Section 3.2.1 the two parameter directions stay separate. The data are the array $\boldsymbol{X}$ of the $n_u \times n_v$ points $\boldsymbol{x}_{kl}$, the control points are the net $\boldsymbol{P}$ of $(n+1) \times (m+1)$ points $\boldsymbol{P}_{ij}$, the matrix $\boldsymbol{A}$ of (3.7) holds the basis values of the first direction at the parameters $\bar{u}_k$, and the second direction brings its own collocation matrix, built

in the same way from the basis functions of that direction at the parameters $\bar{v}_l$,

$$\boldsymbol{B} \;=\; \begin{bmatrix} M_{0,q}(\bar{v}_1) & M_{1,q}(\bar{v}_1) & \cdots & M_{m,q}(\bar{v}_1) \\ M_{0,q}(\bar{v}_2) & M_{1,q}(\bar{v}_2) & \cdots & M_{m,q}(\bar{v}_2) \\ \vdots & \vdots & \ddots & \vdots \\ M_{0,q}(\bar{v}_{n_v}) & M_{1,q}(\bar{v}_{n_v}) & \cdots & M_{m,q}(\bar{v}_{n_v}) \end{bmatrix} \;\in\; \mathbb{R}^{\,n_v \times (m+1)}. \tag{3.9}$$

The value of the surface at the grid point (k, l) is then the entry (k, l) of the product $\boldsymbol{A}\boldsymbol{P}\boldsymbol{B}^{\mathsf{T}}$, the error is

$$E(\boldsymbol{P}) \;=\; \left\| \boldsymbol{A}\,\boldsymbol{P}\,\boldsymbol{B}^{\mathsf{T}} - \boldsymbol{X} \right\|^2, \tag{3.10}$$

with the norm summing the squared distances over all grid points, and the same differentiation as above gives the normal equations of the surface fit and their solution,

$$\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\,\boldsymbol{P}\,\boldsymbol{B}^{\mathsf{T}}\boldsymbol{B} \;=\; \boldsymbol{A}^{\mathsf{T}}\boldsymbol{X}\boldsymbol{B}, \qquad \boldsymbol{P} \;=\; \left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathsf{T}}\,\boldsymbol{X}\,\boldsymbol{B}\left(\boldsymbol{B}^{\mathsf{T}}\boldsymbol{B}\right)^{-1}. \tag{3.11}$$

The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ contain no coordinate index, so here too the three coordinates are treated one at a time. Read from the right, the solution is a fit of curves. The factor $\boldsymbol{X}\boldsymbol{B}(\boldsymbol{B}^{\mathsf{T}}\boldsymbol{B})^{-1}$ fits a curve of the second direction to each row of the data and returns the control points of those row curves, and the factor on the left then fits a curve of the first direction through the control points of the row curves. This is the classical surface-through-curves scheme of Piegl and Tiller [5], and it is the reason the fit of a surface costs two small factorizations, of sizes $n + 1$ and $m + 1$, instead of one factorization of the whole net.

The linearity belongs to the polynomial form. In the rational surface (2.25) the weights sit in the shared denominator of the basis functions, so the surface is not linear in them. With the weights prescribed in advance the derivation above carries over unchanged, with the rational basis functions in place of the polynomial ones, but a fit that treats the weights as unknowns leaves the linear setting altogether and requires the kind of iterative solution that Section 3.3 gives to the parameters [5]. The reconstruction literature therefore fits the B-spline form, and the rational one is reserved for the exact conics of Chapter 2, where the weights are known beforehand.

A fit can also follow the data too closely. Where the control net is dense relative to the data, the fit has more freedom than the data pin down, and a part of the object that should come out flat comes out wavy instead, the surface bending back and forth between the points while the residual stays small. Noise in the scan makes the effect worse, but it is not needed for it to appear. The remedy is to add a fairing term to the error and to minimize $E + \lambda\, E_{\text{smooth}}$ with a small weight λ, most commonly with the thin-plate energy

$$E_{\text{smooth}}(\boldsymbol{S}) \;=\; \int_{\Omega} \left(\|\boldsymbol{S}_{uu}\|^2 + 2\|\boldsymbol{S}_{uv}\|^2 + \|\boldsymbol{S}_{vv}\|^2 \right) \mathrm{d}u\, \mathrm{d}v, \tag{3.12}$$

which measures how strongly the surface bends and therefore penalizes exactly that waviness [15].

The term keeps the form of the fit. A derivative of the surface is again the product $\boldsymbol{A}\boldsymbol{P}\boldsymbol{B}^{\mathsf{T}}$, with the derivatives of the basis functions in place of their values, and only the matrix of the direction being differentiated changes, since neither the net nor the matrix of the other direction depends on that parameter. Writing $\boldsymbol{A}_u$ and $\boldsymbol{A}_{uu}$ for the matrices built from the first and the second derivatives of the basis functions of the first direction at the parameters $\bar{u}_k$, and $\boldsymbol{B}_v$ and

$\boldsymbol{B}_{vv}$ for those of the second direction at $\bar{v}_l$, the second derivatives of the surface at the grid points are the entries of $\boldsymbol{A}_{uu}\boldsymbol{P}\boldsymbol{B}^{\top}$, of $\boldsymbol{A}_{u}\boldsymbol{P}\boldsymbol{B}_{v}^{\top}$ and of $\boldsymbol{A}\boldsymbol{P}\boldsymbol{B}_{vv}^{\top}$. Evaluating (3.12) on the same grid, with the constant cell area absorbed into λ, therefore gives

$$E_{\text{smooth}} \;=\; \left\|\boldsymbol{A}_{uu}\boldsymbol{P}\boldsymbol{B}^{\top}\right\|^2 \;+\; 2\left\|\boldsymbol{A}_{u}\boldsymbol{P}\boldsymbol{B}_{v}^{\top}\right\|^2 \;+\; \left\|\boldsymbol{A}\boldsymbol{P}\boldsymbol{B}_{vv}^{\top}\right\|^2. \tag{3.13}$$

Every term has the shape of (3.10) without the data, so differentiating them one by one as before and adding the result to (3.11) gives the regularized fit,

$$\boldsymbol{A}^{\top}\boldsymbol{A}\,\boldsymbol{P}\,\boldsymbol{B}^{\top}\boldsymbol{B} \;+\; \lambda\Big(\boldsymbol{A}_{uu}^{\top}\boldsymbol{A}_{uu}\,\boldsymbol{P}\,\boldsymbol{B}^{\top}\boldsymbol{B} \;+\; 2\,\boldsymbol{A}_{u}^{\top}\boldsymbol{A}_{u}\,\boldsymbol{P}\,\boldsymbol{B}_{v}^{\top}\boldsymbol{B}_{v} \;+\; \boldsymbol{A}^{\top}\boldsymbol{A}\,\boldsymbol{P}\,\boldsymbol{B}_{vv}^{\top}\boldsymbol{B}_{vv}\Big) \;=\; \boldsymbol{A}^{\top}\boldsymbol{X}\boldsymbol{B}. \tag{3.14}$$

The two directions no longer separate, so the regularized net is solved for at once instead of by the two one-directional solves of the unregularized fit, and that is the price of the fairing term.

The weight λ trades fidelity for fairness. As $\lambda \to 0$ the fit returns to the pure least-squares solution, while for large λ it approaches the thin-plate interpolant of the boundary data, and in practice it is chosen small enough to remove the waviness without pulling the surface away from the data.

The three point clouds of Figure 3.4 are fitted in exactly this way, each with a single bicubic patch, so $p = q = 3$, and with a clamped uniform knot vector in each direction, whose interior knots follow from the degree and the number of control points. Uniform knots are the standard choice in reconstruction practice, since the parametrization has already decided where the surface carries resolution, and the reparametrization school takes that route deliberately by moving the parametrization and leaving the knots alone [30]. Figure 3.5 shows the surfaces obtained on a net of 25×25 control points, with the knot lines drawn on them, and the data are the same throughout, a fixed sample grid of 150×150 points.

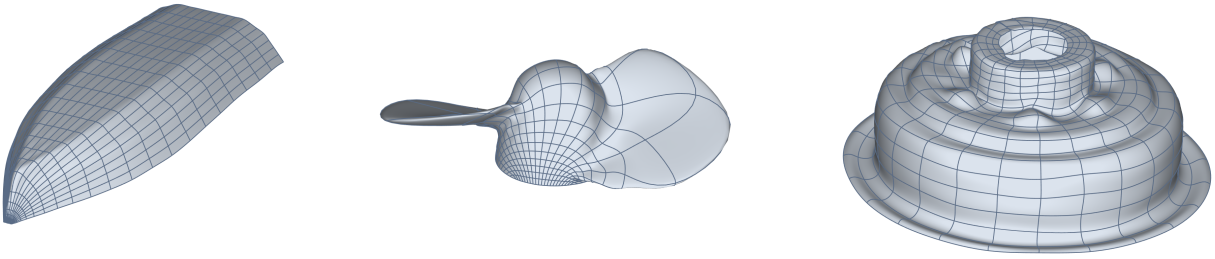


Figure 3.5. Bicubic single-patch fits of the hull, the propeller and the clutch housing on a net of 25×25 control points.

Figure 3.6 shows the geometric error of those fits, drawn on the parametric square in the top row and on the fitted surface in the bottom row. Over most of the domain the surface follows the data closely, and what remains gathers in narrow bands that trace the geometric features of each object, the ridge running to the point where the two sides of the hull meet, the edges of the propeller blade, the small raised features ringing the hub of the clutch housing. Beside every such band the error spreads sideways along the parameter lines in a ripple, the response of a smooth basis asked to reproduce a break in the surface or in its slope.

Refining the net acts on the two parts of that picture differently, which Figure 3.7 follows as the net grows from 6×6 to 40×40. The root mean square error falls steadily, since every control point that is added takes up part of the misfit spread over the smooth regions. The maximum error falls far more slowly and then stops falling, and further refinement no longer moves it. The

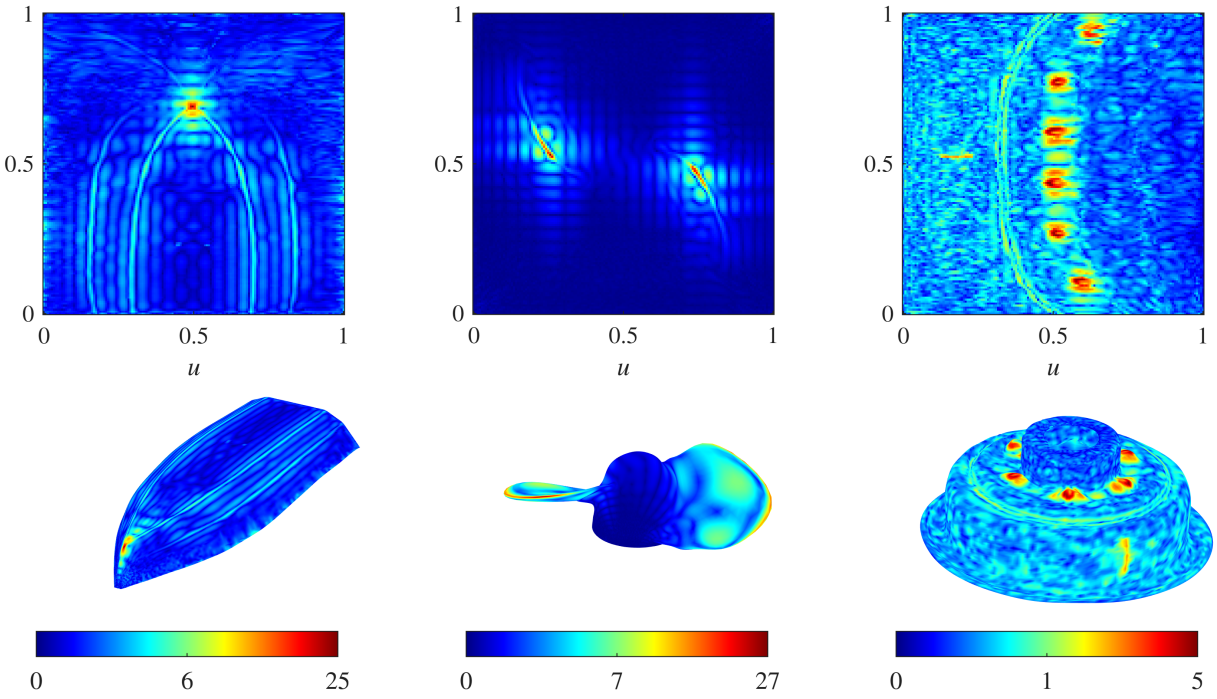


Figure 3.6. Geometric error of the fits in millimetres, on the parametric square (top row) and on the fitted surface (bottom row), on a square-root colour scale.

average quality of the fit therefore keeps improving with resolution, while its worst part settles at a level that resolution alone does not pass. A geometric feature of that kind cannot be described exactly by a globally smooth patch at any resolution, because the obstruction is the structure of the model rather than the number of its control points, and refinement compresses the band and its ripple into a narrower strip without removing either. This is what motivates the domain decomposition of Section 3.3.

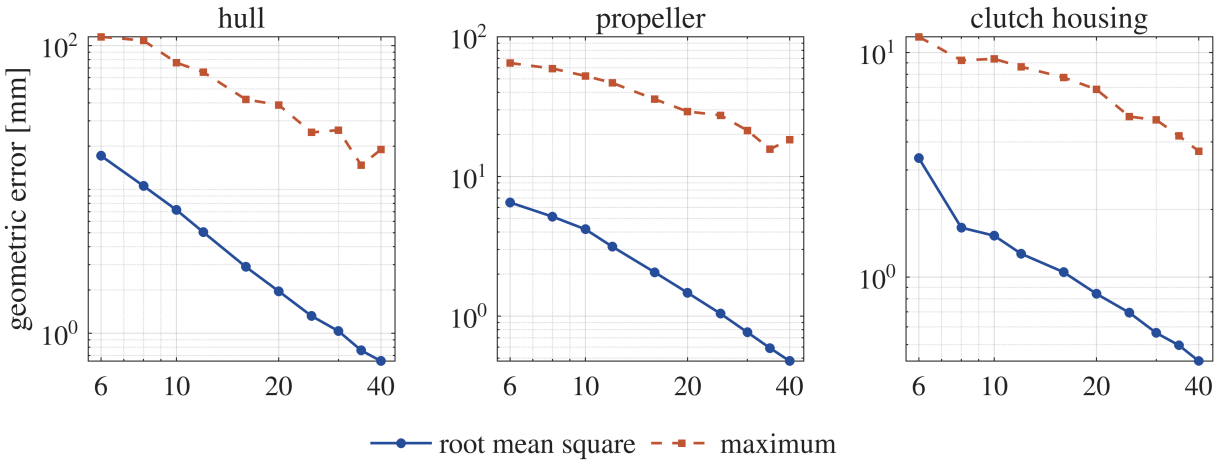


Figure 3.7. Geometric error of the single-patch fits against the number of control points per direction.

The linear core of spline fitting is settled theory. Existence, uniqueness and stability of the least-squares and smoothing formulations have been understood since de Boor [6] and Dierckx [15]. For the reconstruction chain of this work the practical consequence is the one the previous section already stated, that the fit executes a budget which the parametrization has written before

it.

3.3. Nonlinear Improvement of the Fit

The fit of Section 3.2 takes three things as given, the parametrization that assigns the data their parameters, the knot vector of each direction, and the single tensor-product patch that carries the result. Those three are what keeps the problem linear, and they are also what decides the level at which its error settles. Improving a fit beyond that level means releasing one of them, and every release either makes the problem nonlinear or exchanges the model for a richer one. The literature offers several routes, and they are set out below as directions rather than as procedures, since each of them opens a subject of its own.

The first route keeps the model and releases the unknowns that the linear system holds fixed. The parameters carried by the data may be corrected, either by projecting each point onto the current surface and refitting at the new parameters, which is the parameter correction of Hoschek, or by treating parameters and control points as one unknown vector and applying a damped Gauss-Newton method to the point-to-surface distances [5, 49, 50]. Recent work formulates both the alternating and the joint variant for adaptive spline surfaces and reports that the joint update converges considerably faster on industrial data [20]. In the rational form the weights join the same list of unknowns, which is the reason a NURBS fit that determines them is nonlinear.

The second route changes the spline space. The models that make refinement local, the hierarchical and truncated hierarchical B-splines, the T-splines and the locally refined B-splines, were reviewed in Section 2.5, and they are the natural model for an object whose features are unevenly distributed. What the fitting literature adds to that review is the requirement that the refined model remain usable by an analysis, which has produced fitting schemes that place the detail and keep the space analysis-suitable at the same time [51].

The third route leaves the model and the knots alone and moves the data instead. The parametrization decides how much parametric area each part of the object receives, so redistributing that area redistributes the resolution of the fit. It may be redistributed adaptively, by driving the map with a scalar field built from the fitting error or from a curvature indicator [30, 39, 52, 53], or by optimization, in which the parametric positions of the topological point cloud are themselves the design variables [42, 48]. Both keep every individual fit linear and place the nonlinearity in the map.

The fourth route optimizes the knot vectors. For a fixed degree the knots decide the approximation space, and moving a knot changes the basis functions themselves, so the problem is nonlinear in the same sense as the first route. The published attempts remain heuristic, from adaptive insertion until a prescribed residual tolerance is met [15] to knot-based gradient fitting [54], feature-driven dynamic adjustment [55] and learned prediction of the knot vector by neural models [56].

The fifth route gives up the single patch. A scanned component is in general not a deformed rectangle, and a sharp feature that crosses the parameter lines cannot be reproduced by a globally smooth basis at any resolution, which is what Section 3.2.2 showed. Dividing the object into several patches along its own features removes that obstruction. The division may follow a segmentation of the object, obtained with methods borrowed from image analysis [57, 58], or a quad layout laid out over the surface [7, 59, 60], and the classical reconstruction of arbitrary topological type takes the same road [61]. The price is the continuity across the seams, which

every later use of the model has to respect.

None of the five routes is a finished procedure, and they are not exclusive of one another, since a reparametrized model may still be fitted on a hierarchical space and split along its features. What they share is the observation of the previous section, that the level at which the error of a fit settles is decided before the linear system is ever assembled, and that improving it means changing one of the decisions that came first.

4. ISOGEOMETRIC ANALYSIS

Isogeometric analysis was introduced by Hughes, Cottrell and Bazilevs [4] as a computational framework designed to close the gap between computer-aided design and finite element analysis. Its core idea is to adopt the basis functions that represent the geometry exactly, the NURBS basis of Chapter 2, as the approximation functions of the analysis as well. Geometric modelling and numerical simulation are thereby unified within a single mathematical framework, the analysis consumes the design model directly, and the re-meshing that separates the two in conventional practice disappears together with the geometric error it introduces. The high continuity of the NURBS basis provides a smoothness that conventional finite elements cannot offer, which positions isogeometric analysis as a powerful alternative to the standard finite element method [4, 10].

The chapter presents the method in the order in which it is assembled. The conceptual difference from the finite element method comes first, then the principle of virtual work as the single starting point of every formulation, then one-dimensional problems in which the weak form, its discretization and the evaluation of the integrals through the geometric map are carried out in full, and then the Kirchhoff-Love shell along the same path.

4.1. From Finite Elements to Isogeometric Analysis

The finite element method approximates the geometry by its mesh. The element edges trace a piecewise polynomial curve of low degree, so a curved boundary is replaced by a polygon and the computational domain differs from the design model at every level of refinement. Isogeometric analysis works on the design model itself, and the domain stays exact at every level of refinement because knot insertion and degree elevation reproduce the geometry without change (Section 2.4). Figure 4.1 shows the difference on the classical example of a plate with a circular hole, of which one quarter is modelled. The finite element mesh replaces the circular edge of the hole by a polygon, and the hole becomes a circle again only in the limit of refinement, while the isogeometric model is a single quadratic NURBS patch that describes the geometry exactly [4]. Table 4.1 collects other conceptual differences between the two approaches. These differences are explained in detail in various literature and will be more discussed in the rest of the chapter.

The advantages listed there have not been enough to displace the finite element method from industrial practice, and the reasons lie on the geometric side rather than in the numerics. A production computer-aided design model is assembled from trimmed patches whose visible boundaries are curves in the parameter domain, not the boundaries of the patches themselves, and such a model is watertight only to display tolerance, so it is not directly analysis-suitable [34]. Analysis on such a model is possible once the cut cells are integrated and stabilized with care [62, 63], but that is machinery the mesh-based route never needs. A different response removes the trimming rather than accommodating it, by reparameterizing the boundary representation into untrimmed patches under the guidance of a frame field, which returns a model the analysis can use as it stands [64, 65]. A model built from several patches needs its continuity restored across the interfaces before a fourth-order formulation can be used on it, and the couplings that do so are a research field of their own [66, 67, 68, 69]. The parametrization of the domain, which the designer chose for convenience, decides the conditioning of the discretization and there is no established procedure for repairing a bad one [35, 70]. Around all of this stands an ecosystem of solvers, element libraries and pre-processing and post-processing tools built for meshes,

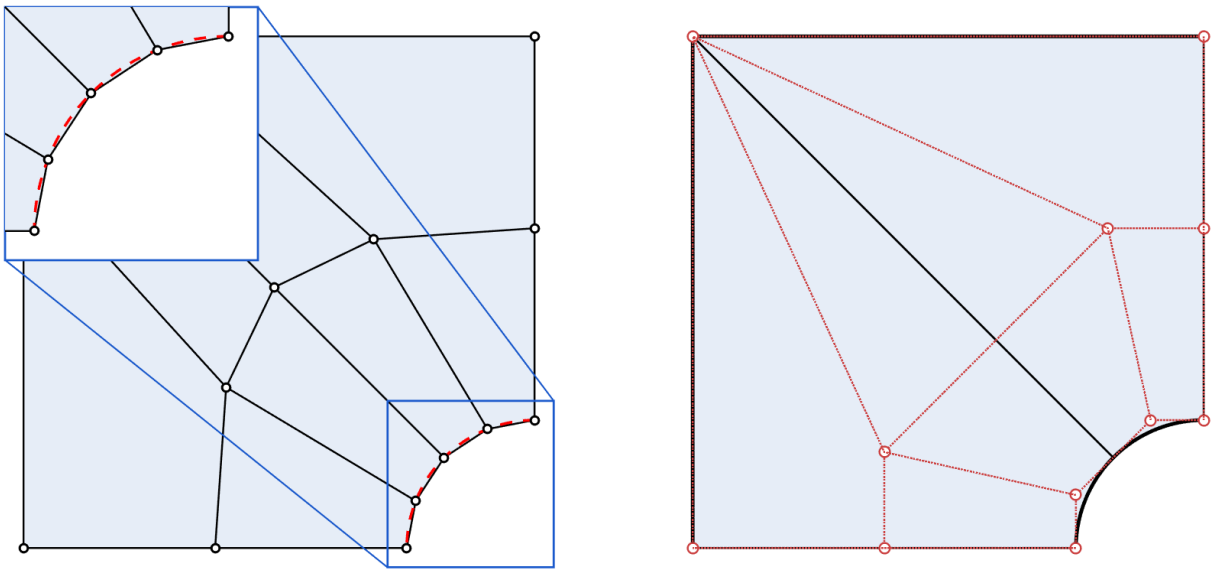


Figure 4.1. Quarter of a plate with a circular hole as a finite element mesh (left) and as one NURBS patch (right).

Table 4.1. Conceptual differences between the finite element method and isogeometric analysis, after Cottrell, Hughes and Bazilevs [10].

	Finite element method	Isogeometric analysis
Geometry	approximated by the mesh and re-approximated at every refinement	exact at every level of refinement
Basis functions	Lagrange polynomials, interpolatory	B-splines or NURBS, non-interpolatory
Element	cell of the mesh	non-empty knot span
Continuity between elements	C^0 for any degree	C^{p-k} at a knot of multiplicity k, C^{p-1} by default
Degrees of freedom	values of the field at the nodes	control variables, without pointwise meaning
Support of a basis function	one layer of elements around a node	$p+1$ knot spans in each direction
Refinement	h and p, each with a new mesh	h, p and k, exact and nested
Boundary conditions	imposed on nodal values	imposed on control variables, weakly or by projection

together with mesh generation that has itself become largely automatic. Isogeometric analysis is therefore established as a research technology and used in industry in special cases, while the mesh remains the default [10].

4.2. Principle of Virtual Work

Every formulation in this chapter descends from one statement. A body is in equilibrium when, for every admissible virtual displacement $\delta \boldsymbol{u}$, the virtual work of the internal forces equals the virtual work of the external forces [71, 72],

$$\int_{\Omega} \delta \boldsymbol{\varepsilon}^{\top} \boldsymbol{\sigma} \, \mathrm{d}\Omega \;=\; \int_{\Omega} \delta \boldsymbol{u}^{\top} \boldsymbol{b} \, \mathrm{d}\Omega \;+\; \int_{\Gamma} \delta \boldsymbol{u}^{\top} \boldsymbol{t} \, \mathrm{d}\Gamma, \tag{4.1}$$

in which $\boldsymbol{\sigma}$ is the stress, $\delta \boldsymbol{\varepsilon}$ the virtual strain produced by $\delta \boldsymbol{u}$, $\boldsymbol{b}$ the body force over the domain Ω and $\boldsymbol{t}$ the traction on the loaded part Γ of its boundary. A concentrated force enters the right-hand side as the product of the force and the virtual displacement of its point of application. For a linearly elastic body the same statement is the condition that the total potential energy $\Pi = U+V$, the strain energy plus the potential of the loads, is stationary, $\delta\Pi = \delta U + \delta V = 0$. This is the form in which the statement is most convenient to specialize to beams and shells, because the strain energy of each structural model is a textbook expression [71].

What (4.1) asks of the displacement field is less than what the differential equation of equilibrium asks. The derivatives are shared between the real and the virtual field, so each carries only half the order of the strong form. A beam whose equilibrium equation is of fourth order needs a displacement with square-integrable second derivatives and nothing more, and this is the property that decides which basis is admissible for which problem. The requirement is stated precisely for each structure below, and it is the point at which the continuity of the spline basis, established in Chapter 2, becomes a structural asset.

A word on notation is needed before the fields are discretized, because the analysis literature and the literature on parametric curves and surfaces do not write the parameters in the same way. In Chapter 2 the parametric coordinates were u and v, as is customary in geometric modelling. In the analysis literature those letters are taken, since u, v and w denote the components of the displacement, and the parametric coordinates are written ξ for a curve and ξ^1 and ξ^2 for a surface [10]. This chapter follows the analysis convention. The basis functions are the same functions as in Chapter 2, only their argument is renamed.

Isogeometric analysis changes nothing in (4.1). The difference from the finite element method appears when the statement is discretized. The displacement and the virtual displacement are expanded in the basis R_i that describes the geometry, the rational basis of Chapter 2, which reduces to the B-spline basis when all weights are equal,

$$\boldsymbol{u}(\xi) \;=\; \sum_i R_i(\xi)\, \boldsymbol{d}_i, \qquad \delta \boldsymbol{u}(\xi) \;=\; \sum_i R_i(\xi)\, \delta \boldsymbol{d}_i, \tag{4.2}$$

with control variables $\boldsymbol{d}_i$ attached to the control points, and the integrals of (4.1) are evaluated on the geometric model itself, through its parametrization. Two steps therefore remain. The statement has to be specialized to a structural model, and its integrals have to be evaluated through the geometric map. Both are carried out in full for one-dimensional problems in Section 4.3 and for surfaces in Section 4.4. Worked implementations of the same steps are given by Nguyen

et al. [11], in the GeoPDEs framework of de Falco, Reali and Vázquez [73], and in the lecture notes of Hughes, Sangalli and Tani [74].

4.3. One-Dimensional Problems

The simplest setting in which the whole method can be seen at work is a beam. Its axis is a NURBS curve, written here as $\boldsymbol{C}(\xi)$ in the analysis coordinate $\xi \in [0, 1]$, and the knot spans of its knot vector are the elements. Here it is important to distinguish the three domains involved in isogeometric analysis. Unlike FEM, where basis functions are defined over an element in physical space, IGA uses basis functions defined in the parametric domain, over parametric knot spans. The integration is done in quadrature points on the parent element, which are then mapped onto the parametric space and again onto the physical space where the geometry actually exists. The three spaces are explained in mostly all literature regarding isogeometric analysis, but the detailed review is given in [10] Figure 4.2 shows the three domains involved and the two maps between them. The affine map of the parent element onto the parametric domain is given as

$$\xi \;=\; \tfrac{1}{2}\big[(\xi_{i+1} - \xi_i)\,\tilde{\xi} + (\xi_{i+1} + \xi_i)\big] \tag{4.3}$$

The geometric map $\boldsymbol{C}(\xi)$ carries the quadrature points onto the beam axis, where the integrand is evaluated.

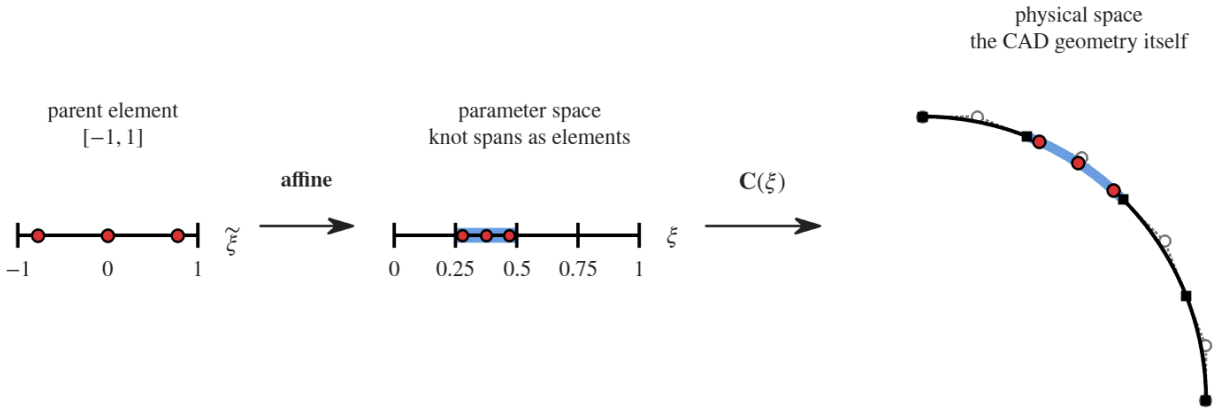


Figure 4.2. The parent interval, one knot span of the parameter line and its image on the beam axis, with the quadrature points carried through both maps.

The geometric map also carries the derivatives and the measure. With s the arc length along the axis, $J(\xi) = \|\mathrm{d}\boldsymbol{C}/\mathrm{d}\xi\|$ is the length of axis per unit of parameter, so [2, 10]

$$\mathrm{d}s = J\,\mathrm{d}\xi, \qquad \frac{\mathrm{d}(\cdot)}{\mathrm{d}s} = \frac{1}{J}\frac{\mathrm{d}(\cdot)}{\mathrm{d}\xi}, \qquad \frac{\mathrm{d}^2(\cdot)}{\mathrm{d}s^2} = \frac{1}{J^2}\frac{\mathrm{d}^2(\cdot)}{\mathrm{d}\xi^2} \;-\; \frac{1}{J^4}\left(\frac{\mathrm{d}\boldsymbol{C}}{\mathrm{d}\xi}\cdot\frac{\mathrm{d}^2\boldsymbol{C}}{\mathrm{d}\xi^2}\right)\frac{\mathrm{d}(\cdot)}{\mathrm{d}\xi}. \tag{4.4}$$

An integral along the beam is therefore assembled span by span [10, 11],

$$\int_0^L f(s)\,\mathrm{d}s \;=\; \sum_e \int_{-1}^{1} f\big(s(\xi)\big)\,J\,\tfrac{1}{2}(\xi_{i+1} - \xi_i)\,\mathrm{d}\tilde{\xi}, \tag{4.5}$$

and each span integral is evaluated by a Gauss-Legendre rule with $p + 1$ points, exact for the polynomial part of the integrand. Every stiffness matrix and load vector of the chapter is computed through this chain, and it reappears unchanged, one dimension up, for surfaces. The rule

itself is inherited from the finite element method, where it integrates functions that meet with C^0 continuity at the element boundaries. On a spline basis of higher continuity it evaluates many more points than the accuracy of the result requires, so the cost of assembly has become a subject of its own, addressed by quadrature rules built for the smoothness of the space and by assembly schemes written for locally refined bases [75, 76].

4.3.1. Curved Beam in the Plane

The beam is treated in its general form from the start, with an axis that is an arbitrary NURBS curve in the plane. Straight beams and the circular arches appear as special cases rather than as separate theories. Spatial curves, which bring torsion and the theory of spatial rods with them, are outside the scope of this review. The shape of the axis enters the formulation through its initial curvature,

$$\kappa_0 \;=\; \frac{x'\,y'' - y'\,x''}{J^3}, \qquad \boldsymbol{C}(\xi) = \big(x(\xi),\, y(\xi)\big), \tag{4.6}$$

primes denoting derivatives with respect to ξ, whose reciprocal $\rho = 1/\kappa_0$ is the local radius of curvature [2]. For a straight axis $\kappa_0 = 0$, for a circular arc of radius R it is the constant $1/R$, and for a general spline it varies along the axis. Since the axis is the exact geometric model, (4.6) is evaluated from the derivatives of Section 2.3.2 without any approximation, whereas on a faceted mesh the curvature of the axis is not even defined.

A point of the axis moves tangentially by u and normally by w, both functions of the arc length s. The membrane strain and the change of curvature of the axis are [77, 78]

$$\varepsilon \;=\; \frac{\mathrm{d}u}{\mathrm{d}s} - \kappa_0\, w, \qquad \kappa \;=\; \frac{\mathrm{d}^2 w}{\mathrm{d}s^2} + \frac{\mathrm{d}(\kappa_0\, u)}{\mathrm{d}s}, \tag{4.7}$$

the signs following the convention for the positive directions on the differential element. For constant curvature the second term of κ reduces to $\kappa_0\,\mathrm{d}u/\mathrm{d}s$, which is the classical arch theory [71], and for $\kappa_0 = 0$ both coupling terms vanish, $\varepsilon = \mathrm{d}u/\mathrm{d}s$ and $\kappa = \mathrm{d}^2 w/\mathrm{d}s^2$, which is a straight Euler-Bernoulli beam with its membrane and bending parts decoupled. The strain energy and the potential of tangential and normal distributed loads f and q and of concentrated forces F_k and Q_k at the points s_k are

$$\begin{aligned} U \;&=\; \frac{1}{2}\int_0^L \big(EA\,\varepsilon^2 + EI\,\kappa^2\big)\mathrm{d}s, \\ V \;&=\; -\int_0^L \big(fu + qw\big)\mathrm{d}s \;-\; \sum_k \big(F_k\,u(s_k) + Q_k\,w(s_k)\big), \end{aligned} \tag{4.8}$$

and $\delta U + \delta V = 0$ gives the weak form

$$\int_0^L \big(EA\,\varepsilon\,\delta\varepsilon + EI\,\kappa\,\delta\kappa\big)\mathrm{d}s \;=\; \int_0^L \big(f\,\delta u + q\,\delta w\big)\mathrm{d}s \;+\; \sum_k \big(F_k\,\delta u(s_k) + Q_k\,\delta w(s_k)\big). \tag{4.9}$$

Concentrated moments enter in the same way, through the rotation of the cross section at their point of application.

The highest derivative in (4.9) is the second one, in κ, whereas the equilibrium equations of the beam are of higher order. The basis in which w is expanded must therefore have square-

integrable second derivatives, which means that its first derivative has to be continuous across element boundaries. Classical finite elements meet this with Hermite polynomials and pay with a rotational unknown at every node. A spline basis of degree $p \geq 2$ with simple interior knots is C^{p-1} and meets it without any addition, so the isogeometric beam carries two displacements per control point and no rotations [10, 11]. Both fields are expanded in the same basis, so u inherits the same continuity although C^0 would suffice for it alone.

Both displacements are discretized in the rational basis of the axis,

$$u \;=\; \boldsymbol{R}\,\boldsymbol{d}_u, \qquad w \;=\; \boldsymbol{R}\,\boldsymbol{d}_w, \qquad \boldsymbol{R} = \begin{bmatrix} R_1 & R_2 & \cdots & R_n \end{bmatrix}, \tag{4.10}$$

with the derivatives with respect to s supplied by (4.4). Inserting this into (4.7) makes both strain measures linear in the control variables. Ordering the unknowns control point by control point as $\boldsymbol{d} = [u_1, w_1, u_2, w_2, \dots]^\top$, the strains become $[\varepsilon, \kappa]^\top = \boldsymbol{B}\,\boldsymbol{d}$, in which the block of $\boldsymbol{B}$ belonging to control point i has the two rows

$$\boldsymbol{B}_i \;=\; \begin{bmatrix} R_i' & -\kappa_0\, R_i \\ (\kappa_0\, R_i)' & R_i'' \end{bmatrix}, \tag{4.11}$$

primes now denoting derivatives with respect to arc length, and $(\kappa_0 R_i)' = \kappa_0 R_i' + \kappa_0' R_i$ carrying the variation of the curvature along the axis. The virtual strains are $\boldsymbol{B}\,\delta\boldsymbol{d}$ with the same matrix. The two stiffnesses are collected in $\boldsymbol{D} = \operatorname{diag}(EA,\, EI)$, and substituting into (4.9) and demanding that the result hold for every $\delta\boldsymbol{d}$ gives the linear system

$$\boldsymbol{K}\,\boldsymbol{d} \;=\; \boldsymbol{F}, \qquad \boldsymbol{K} \;=\; \int_0^L \boldsymbol{B}^\top \boldsymbol{D}\, \boldsymbol{B}\, \mathrm{d}s \;=\; \int_0^1 \boldsymbol{B}^\top \boldsymbol{D}\, \boldsymbol{B}\, J\, \mathrm{d}\xi, \tag{4.12}$$

in which the load vector collects, for every control point, the integrals $\int_0^L R_i f \,\mathrm{d}s$ and $\int_0^L R_i q \,\mathrm{d}s$ of the distributed loads and the products $R_i(s_k)F_k$ and $R_i(s_k)Q_k$ of the concentrated ones. The integrals are evaluated span by span through (4.5). The matrix $\boldsymbol{K}$ is symmetric and banded, with a bandwidth set by the degree, because two basis functions interact only where their supports of $p + 1$ spans overlap.

Boundary conditions are imposed on the control variables, and the non-interpolatory basis makes this slightly less direct than in the finite element method. A clamped end requires both displacements and the rotation of the cross section to vanish. Only the first basis function is non-zero at the end of a clamped knot vector, and the tangent there is set by the first two control points, so the displacements are fixed through the control variables of the end control point and the rotation through those of its neighbour, which are tied to them. The rotation is prescribed through the position of a control point rather than through a rotational unknown, and this is the mechanism that carries over to the boundary conditions of shells.

It should be clear that imposing boundary conditions elsewhere than at the ends of the beam is considerably more difficult than in the standard finite element method. It is necessary to establish relations among the control point displacements in order to ensure that the boundary condition is met. This can be done by the penalty method or by Lagrange multipliers, which is not covered in this review but is contained in the literature [10, 11].

4.3.2. Examples

The first example is the straight cantilever of Figure 4.3, clamped at the left end and loaded by a force at the right end, with the data of Table 4.2. The axis is a B-spline curve of degree one with two control points. Although a simple problem, it serves to demonstrate the difference in solutions between isogeometric analysis and finite element analysis. It also shows how refinement, covered in Section 2.4, changes the solution. With degree two and a single element the deflection at the tip is 25 per cent too small, because a quadratic can only bend with constant curvature, and splitting the beam into two elements halves the error to 6 per cent, so refinement converges towards the solution in the familiar finite element manner. Degree three with a single element gives the deflection to machine precision at once, since the exact deflection of a cantilever under a tip force is a cubic and the discrete space contains it. Refinement approaches the solution, whereas a basis smooth enough to contain it delivers it outright. Under a uniformly distributed load the exact deflection is a quartic, and the same exactness on a single element would require degree four. On a straight axis the geometry is exact in both methods, so this example shows the gain of the smooth basis alone. The advantage of isogeometric analysis shows fully on curved members, where a mesh also loses the geometry, and that is the second example.

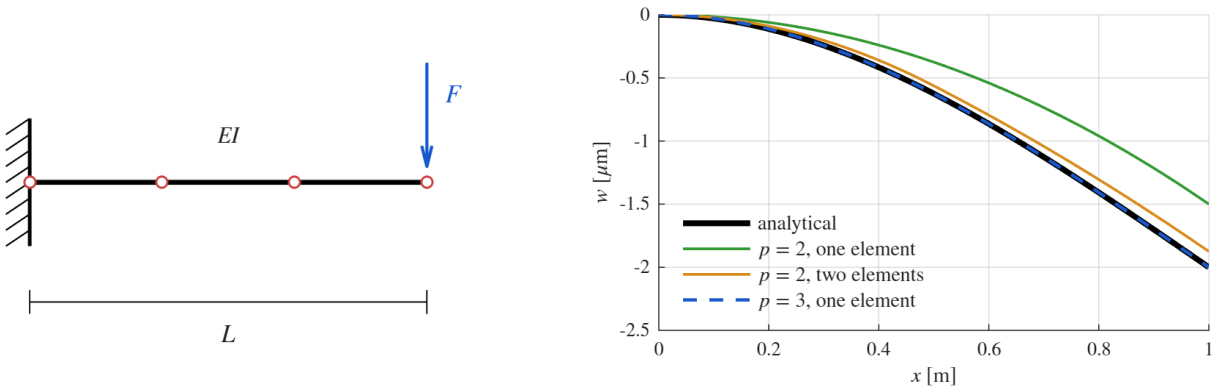


Figure 4.3. Straight cantilever with its control points (left) and its deflection for degree two with one and two elements and for degree three with one element (right).

Table 4.2. Data of the straight cantilever example.

Length	$L = 1$ m
Young's modulus	$E = 200$ GPa
Second moment of area	$I = 8.333 \times 10^{-6}$ m^4
Tip force	$F = 10$ N
Initial geometry	$p = 1$, $U = \{0, 0, 1, 1\}$, control points $(0, 0)$ and $(L, 0)$
Discretizations	$p = 2$, $U = \{0, 0, 0, 1, 1, 1\}$, one element
	$p = 2$, $U = \{0, 0, 0, 1/2, 1, 1, 1\}$, two elements
	$p = 3$, $U = \{0, 0, 0, 0, 1, 1, 1, 1\}$, one element

The second example is the classical quarter-circle cantilever of Figure 4.4, clamped at one end and loaded by a radial force F at the free end, with the data of Table 4.3. Its axis is the quarter circle of Section 2.3.3. The analysis needs a richer space than the three control points of the geometry, so the segment is elevated to degree four and split by one inserted knot into two

elements, which gives six control points, while the arc remains a circle to machine precision. The clamped end is imposed through the first two control points.

Table 4.3. Data of the curved cantilever example.

Radius	$R = 1$ m
Young's modulus	$E = 200$ GPa
Cross-section area	$A = 0.01$ m^2
Second moment of area	$I = 8.333 \times 10^{-6}$ m^4
Tip force	$F = 10$ N, radial, towards the centre
Geometry	$p = 2$, $U = \{0, 0, 0, 1, 1, 1\}$, control points $(0, R)$, (R, R) and $(R, 0)$, weights 1, $\sqrt{2}/2$ and 1
Analysis discretization	$p = 4$, $U = \{0, 0, 0, 0, 0, 1/2, 1, 1, 1, 1, 1\}$, six control points

Figures 4.4 and 4.5 show the solution, the deformed axis and the recovered internal forces, the axial force, the shear force and the bending moment, against the statically determinate analytical solution. While the displacement fields produce an exact solution, it is clear that the recovered internal forces have significant error, as the numerical solution oscillates around the exact solution. This is not a defect of the formulation but a phenomenon with a name and a literature of its own, and the next section is devoted to it.

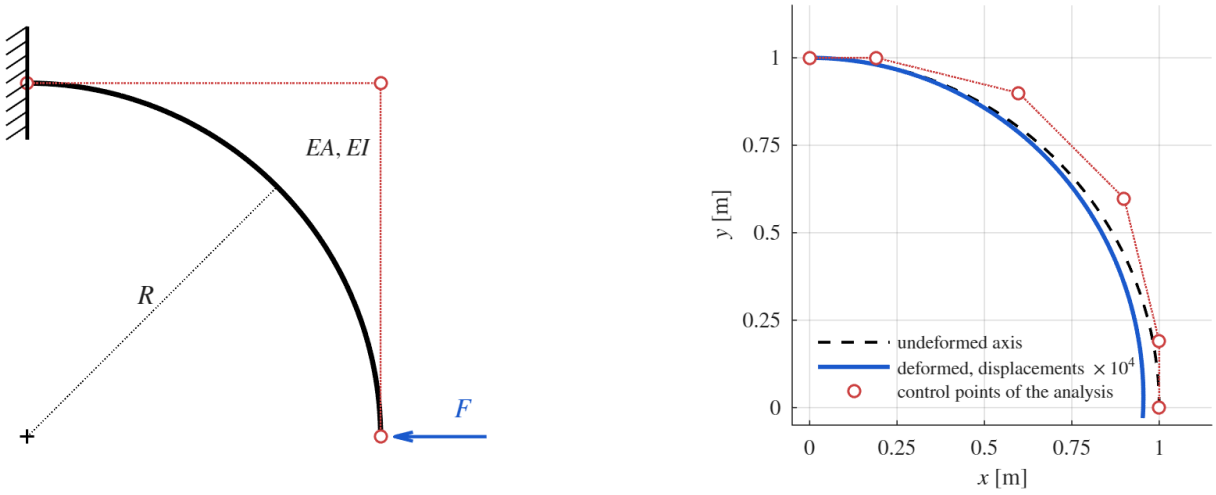


Figure 4.4. Quarter-circle cantilever with the three control points of its geometry (left) and its deformed axis with the control points of the analysis (right).

4.3.3. Locking

In the slender regime the quarter circle carries its load almost entirely by bending, and the exact solution has a nearly inextensible axis, $\varepsilon \approx 0$. The discrete solution has to satisfy $\mathrm{d}u/\mathrm{d}s - \kappa_0 w \approx 0$ with u and w expanded in the same basis, and since the derivative of a basis function is not a combination of the basis functions themselves, the constraint cannot be met term by term. It is satisfied only in the mean, and the residual is multiplied by the large membrane stiffness EA and appears as the spurious, oscillating axial force of Figure 4.5. As the slenderness grows the same mechanism stiffens the whole response, because the discrete space cannot bend without

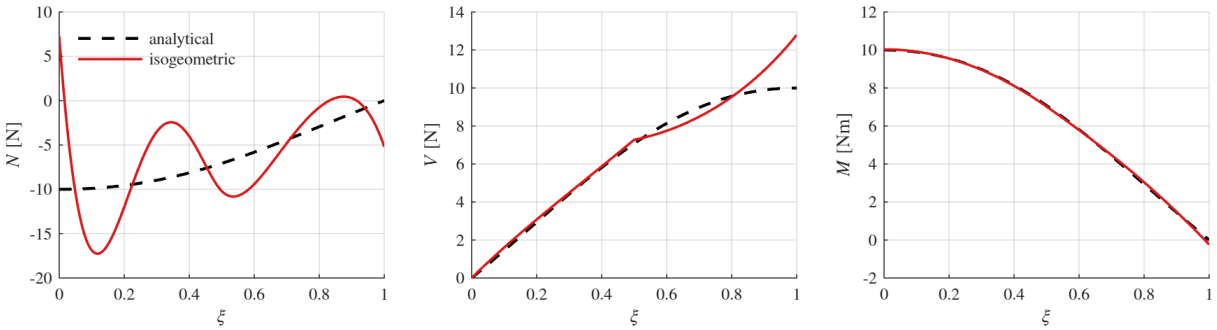


Figure 4.5. Axial force, shear force and bending moment along the quarter-circle cantilever, compared with the analytical solution.

stretching, and the error grows with the ratio of radius to thickness. This is *membrane locking*, and it is a phenomenon of curved geometry. On a straight axis the membrane strain is $\mathrm{d}u/\mathrm{d}s$ alone, the constraint $\varepsilon = 0$ is met by $u = 0$ without any demand on w, and bending and stretching are decoupled, which is why the straight cantilever of Section 4.3.2 shows nothing of the kind. As $\kappa_0 \to 0$ the coupling in (4.7) disappears and the locking with it, so the straight beam is the limit in which the curved beam is free of locking, and the severity of the effect grows with the slenderness of the curved member. The mechanism was analysed for classical curved elements by Stolarski and Belytschko [79] and revisited for spline discretizations by Echter and Bischoff [80], who showed that the higher continuity of the spline basis reduces the effect but does not remove it.

Raising the degree is the simplest and most effective way of reducing the effect. Figure 4.6 repeats Figure 4.5 with the degree raised from four to six on the same two elements, showing reduced oscillations. Refining the knot vector has the same effect, and in practice the two are combined, the degree first and the knot vector afterwards, which is the k-refinement of the next section. Different and more complex methods that remove locking more systematically are given in literature [80, 81]. In shells the same locking appears in the membrane strains of bending-dominated responses, shear locking joins it in shear-deformable formulations. The remedies are the same, and the Kirchhoff-Love shells of Section 4.4 inherit them.

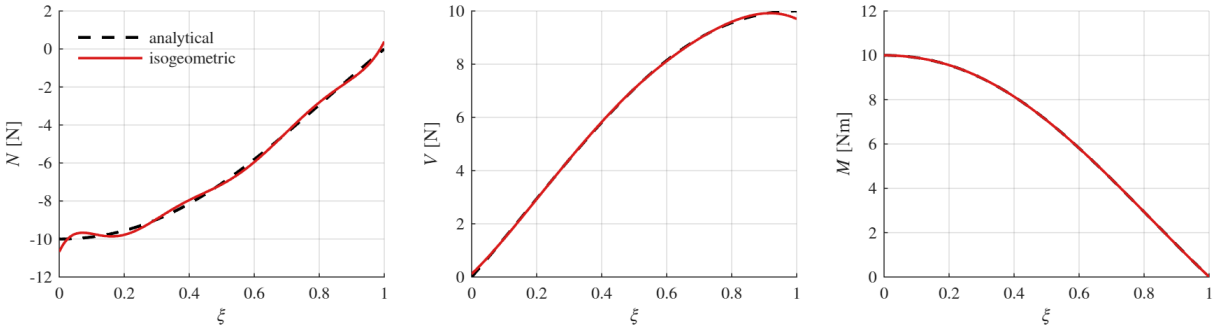


Figure 4.6. Axial force, shear force and bending moment of the quarter-circle cantilever with the degree raised to six on the same two elements.

4.3.4. Refinement of the Discretization

Refining the discretization requires no new machinery. The solution space is the space of the geometry, so enlarging it means refining the geometric model by the operators of Section 2.4, and both of them leave the model itself untouched. Knot insertion subdivides the elements and is the counterpart of h-refinement, degree elevation raises the order and is the counterpart of p-refinement, and because each coarse space is contained in the fine one, the sequence of spaces is nested and nothing computed on a coarse level has to be re-approximated on a fine one [4, 10]. What the finite element method cannot offer is the third possibility that arises because the two operators do not commute. If knots are inserted first and the degree is elevated afterwards, the elevation has to raise the multiplicity of every knot, the inserted ones included, and the refined space is only as smooth across the new knots as the coarse one was. If the degree is elevated first, on the coarsest representation, and knots are inserted afterwards, the new knots enter with multiplicity one and the space is C^{p-1} across every one of them, with fewer degrees of freedom for the same mesh and degree. This second order of operations is k-refinement [4, 10], and it is the procedure by which the spaces used in this chapter are produced, the fitted geometry of degree two or three being elevated first and the knot vector refined afterwards. Beyond the tensor-product patch, local refinement is the subject of the hierarchical and T-spline constructions of Section 3.3.

4.4. Two-Dimensional Problems

The step to surfaces follows the same path. The domain is now a NURBS surface, the parameter space carries two knot vectors, and the elements are the cells of the tensor-product grid. Of the two-dimensional structural models only the Kirchhoff-Love shell is treated, because it is the most general member of the family. The membrane is the shell without bending stiffness and the plate is the shell without curvature, so both are recovered from it by simplification. The chain of maps is set out first, the differential geometry that the shell requires second, and the shell itself last.

4.4.1. Domains and Mappings

An isogeometric discretization is defined on three domains, and every quantity the method computes is obtained by transporting an integral through the maps that connect them. The chain is the one of Section 4.3 with one parameter more, and it is worth setting out in full because every shell integral is evaluated through it [4, 10]. Figure 4.7 shows the three domains and the two maps for a single element of a bivariate patch.

The *parameter space* is the rectangle spanned by the knot vectors, taken here as the unit square $\hat{\Omega} = [0,1]^2$ with coordinates (ξ^1, ξ^2). The knot lines partition it into rectangular cells. A cell of non-zero area,

$$\hat{\Omega}^e = [\xi_i^1, \xi_{i+1}^1) \times [\xi_j^2, \xi_{j+1}^2), \qquad \xi_i^1 < \xi_{i+1}^1, \quad \xi_j^2 < \xi_{j+1}^2, \tag{4.13}$$

is what plays the role of an element [11]. This is the cell structure introduced on curves and surfaces in Section 2.2.3, now read as a mesh. Repeated knots produce empty cells, which carry no quadrature and are simply skipped, so the number of elements is set by the number of *distinct*

knots and not by the length of the knot vector. Over one such cell the surface is a single polynomial or rational patch, and exactly $(p+1)(q+1)$ basis functions are non-zero on it. Because each univariate function is supported on $p+1$ consecutive spans, two neighbouring elements do not merely share the functions attached to a common interface, as in a C^0 discretization, but share p entire rows of control variables. This overlap is the combinatorial expression of the higher continuity recorded in Table 4.1, and it is the reason the resulting system matrices are wider than their finite element counterparts.

The *physical space* is the image of the parameter space under the geometric map, which for the models of this review is the NURBS surface (2.25) itself,

$$\boldsymbol{S} : \hat{\Omega} \to \mathbb{R}^3, \qquad \boldsymbol{S}(\xi^1, \xi^2) \;=\; \sum_{i,j} R_{ij,pq}(\xi^1, \xi^2)\, \boldsymbol{P}_{ij}. \tag{4.14}$$

This is the decisive point of the whole method. The map is not constructed by the analysis and is not an interpolant of anything. It is the design model, taken over unchanged, and it remains the same map under every refinement of the basis, because knot insertion and degree elevation reproduce the geometry exactly (Section 2.4).

The third domain is the *parent element* $\tilde{\Omega} = [-1,1]^2$ with coordinates $(\tilde{\xi}^1, \tilde{\xi}^2)$, on which quadrature rules are tabulated once and for all. It is connected to a knot span by the affine map

$$\xi^1 \;=\; \tfrac{1}{2}\big[(\xi^1_{i+1} - \xi^1_i)\,\tilde{\xi}^1 + (\xi^1_{i+1} + \xi^1_i)\big], \qquad \xi^2 \;=\; \tfrac{1}{2}\big[(\xi^2_{j+1} - \xi^2_j)\,\tilde{\xi}^2 + (\xi^2_{j+1} + \xi^2_j)\big], \tag{4.15}$$

whose Jacobian determinant is the constant $\tfrac{1}{4}(\xi^1_{i+1} - \xi^1_i)(\xi^2_{j+1} - \xi^2_j)$, one quarter of the area of the knot span.

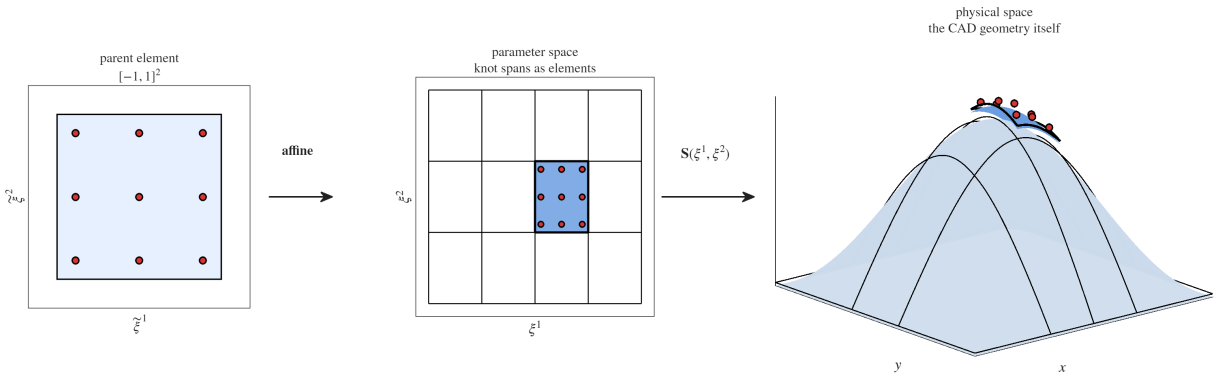


Figure 4.7. The parent element, one knot span of the parameter space and its image in physical space, with the quadrature points carried through both maps.

The two maps compose. A point of the parent element is carried by (4.15) into a knot span and by (4.14) onto the surface, and Figure 4.7 follows a set of quadrature points along that chain. They are tabulated on the parent square, distributed over the knot span by the affine map, and land on the curved image of the element in physical space, where the integrand is actually evaluated. An integral over a physical element then becomes an integral over the parent square,

$$\int_{\Omega^e} f \,\mathrm{d}\Omega \;=\; \int_{\tilde{\Omega}} f\big(\boldsymbol{S}(\xi^1(\tilde{\xi}^1), \xi^2(\tilde{\xi}^2))\big)\, \sqrt{\det a}\, \tfrac{1}{4}(\xi^1_{i+1} - \xi^1_i)(\xi^2_{j+1} - \xi^2_j)\, \mathrm{d}\tilde{\Omega}, \tag{4.16}$$

in which $\sqrt{\det a}$ is the area measure of the surface, built in Section 4.4.2 from the first fundamental form of the geometric map [82], and the constant factor is the Jacobian of the affine map

(4.15). The integral is then approximated by a Gauss-Legendre rule. The default choice is the full tensor-product rule with $(p+1)\times(q+1)$ points per element, exact for the polynomial part of the integrand and sufficiently accurate for the rational and geometric factors. Rules that exploit the inter-element smoothness of the basis to use fewer points exist, but the full rule remains the reference practice [10, 74].

One requirement closes this section. Everything above presupposes that the geometric map is regular, that is that the tangent vectors it induces stay linearly independent and the area measure $\sqrt{\det a}$ keeps away from zero throughout the patch. A map that folds or degenerates carries no meaningful derivatives at the offending points and poisons the quadrature around them.

4.4.2. Curvilinear Geometry of Surfaces

The Kirchhoff-Love shell theory of the next subsection is written in the language of the differential geometry of surfaces. This subsection condenses that apparatus, following the classical expositions in the CAGD and shell literature [2, 82, 83], in exactly the amount that the discretization consumes. The shell formulation itself is based on the work of Bletzinger and his co-workers [82, 84].

The apparatus is written in index notation, so its conventions are stated first. Greek indices take the values 1 and 2 and refer to the parametric coordinates ξ^1 and ξ^2, while Latin indices take the values 1 to 3 and refer to the global Cartesian coordinates. A comma in a subscript denotes a partial derivative with respect to a parametric coordinate, so $\boldsymbol{r}_{,\alpha}$ stands for $\partial \boldsymbol{r}/\partial \xi^\alpha$. Summation follows the Einstein convention. An index that appears twice within a term, once as a subscript and once as a superscript, is summed over its range, so $v^\alpha \boldsymbol{g}_\alpha$ abbreviates $v^1\boldsymbol{g}_1 + v^2\boldsymbol{g}_2$. An index that appears only once is free, and the equation holds for each of its values [83].

The reason curvilinear coordinates are introduced at all is mechanical. The strain of a shell lives on its midsurface. The membrane strain measures the change of lengths and angles within the surface and the bending strain the change of its curvature. Both are computed in a local coordinate system that the parametrization attaches to every point of the surface, and this subsection supplies exactly that local system together with the two fundamental forms measured in it.

The central object is the mapping $\boldsymbol{r}(\xi^1, \xi^2)$ that carries a point of the flat parametric domain to a point of the surface in space. For the reconstructed models of this review this is precisely the NURBS map (2.25). Its partial derivatives define the *covariant basis*

$$\boldsymbol{g}_\alpha \;=\; \frac{\partial \boldsymbol{r}}{\partial \xi^\alpha}, \qquad \alpha = 1, 2, \tag{4.17}$$

the tangents to the coordinate curves. They span the tangent plane but are in general neither unit nor mutually orthogonal. Figure 4.8 shows the arrangement. The position vector $\boldsymbol{r}$ points from the origin of the global Cartesian frame to a point of the surface, the coordinate curves of ξ^1 and ξ^2 cross at that point, and the covariant base vectors follow them while the unit normal completes the local frame.

The scalar products of the basis vectors form the surface metric, the *first fundamental form*

$$a_{\alpha\beta} \;=\; \boldsymbol{g}_\alpha \cdot \boldsymbol{g}_\beta, \qquad \mathrm{d}s^2 = a_{\alpha\beta}\,\mathrm{d}\xi^\alpha \mathrm{d}\xi^\beta, \tag{4.18}$$

which measures all lengths and angles on the surface. Its determinant supplies the area element

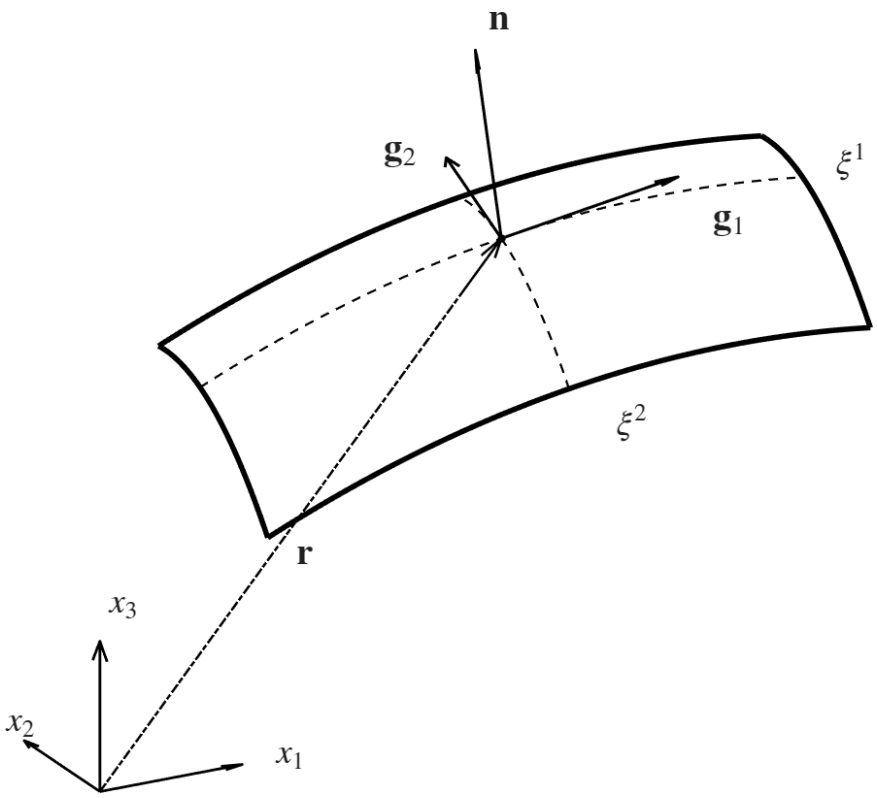


Figure 4.8. A surface patch in the global Cartesian frame, with the position vector, the covariant base vectors and the unit normal at one point.

$\mathrm{d}S = \sqrt{\det a}\ \mathrm{d}\xi^1\mathrm{d}\xi^2$. Alongside the covariant basis lives the *dual (contravariant) basis* $\boldsymbol{g}^\alpha$, defined by $\boldsymbol{g}^\alpha \cdot \boldsymbol{g}_\beta = \delta^\alpha_\beta$, with $a^{\alpha\beta} = \boldsymbol{g}^\alpha \cdot \boldsymbol{g}^\beta$ the inverse of $a_{\alpha\beta}$. The metric raises and lowers indices, $\boldsymbol{g}_\alpha = a_{\alpha\beta}\,\boldsymbol{g}^\beta$ and $\boldsymbol{g}^\alpha = a^{\alpha\beta}\,\boldsymbol{g}_\beta$. A vector $\boldsymbol{g}$ is an invariant geometric object, neither covariant nor contravariant in itself, but on a non-orthogonal basis it admits two genuinely different decompositions. Vectors and tensors and their decomposition are a demanding topic and thus will not be covered entirely in this review, but further reading is contained in [85].

The unit normal $\boldsymbol{n} = (\boldsymbol{g}_1 \times \boldsymbol{g}_2)/\|\boldsymbol{g}_1 \times \boldsymbol{g}_2\|$ completes $\{\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{n}\}$ to a (generally non-orthogonal) spatial basis at each surface point. The first fundamental form knows nothing of how the surface curves in space, and a flat and a rolled sheet of paper share it. Curvature is measured by the change of the normal along the surface. Differentiating the identity $\boldsymbol{g}_\alpha \cdot \boldsymbol{n} = 0$ gives the two equivalent faces of the *second fundamental form*,

$$b_{\alpha\beta} = \boldsymbol{g}_{\alpha,\beta} \cdot \boldsymbol{n} = -\,\boldsymbol{g}_\alpha \cdot \boldsymbol{n}_{,\beta}, \tag{4.19}$$

where $\boldsymbol{g}_{\alpha,\beta} = \boldsymbol{r}_{,\alpha\beta}$. The left expression projects the curvature vector of the coordinate curves onto the normal (how fast the surface leaves its tangent plane), the right one measures the tilting of the normal along the surface, and symmetry of mixed partials makes $b_{\alpha\beta}$ symmetric.

On a non-orthogonal basis the position of indices is physics, not typography. Strain measures arise as changes of the metric, $\varepsilon_{\alpha\beta} = \frac{1}{2}(\bar{a}_{\alpha\beta} - a_{\alpha\beta})$, and are therefore *naturally covariant*. Stress resultants are the energy conjugates of strain and are *naturally contravariant*, so that the energy density $W = \frac{1}{2}\, n^{\alpha\beta}\varepsilon_{\alpha\beta}$ pairs upper against lower indices and is an invariant scalar. The constitutive tensor consequently carries four upper indices, $n^{\alpha\beta} = C^{\alpha\beta\gamma\delta}\varepsilon_{\gamma\delta}$, absorbing the index-lowering into itself. This bookkeeping, mechanical as it looks, is exactly what makes the

Voigt matrices of Section 4.4.3 metric-dependent [82, 83].

4.4.3. Kirchhoff-Love Shells

Thin-walled scanned components such as hulls, blades, housings and stamped parts are shells, and for thickness-to-radius ratios below $\sim 1/20$ the appropriate model is the Kirchhoff-Love (KL) theory, which constrains the director to remain normal to the deformed midsurface and eliminates transverse shear as an independent kinematic field [83, 84]. Let the midsurface be the NURBS map (2.25) with covariant basis $\boldsymbol{g}_\alpha$, unit normal $\boldsymbol{n}$ and fundamental forms (4.18) and (4.19) (in much of the shell literature the midsurface basis is written $\boldsymbol{a}_\alpha$, $\boldsymbol{a}_3$). The shell continuum is the normal extrusion

$$\boldsymbol{x}(\xi^1,\xi^2,\xi^3) \;=\; \boldsymbol{r}(\xi^1,\xi^2) + \xi^3\,\boldsymbol{n}(\xi^1,\xi^2), \qquad \xi^3 \in [-t/2,\, t/2]. \tag{4.20}$$

The KL hypotheses reduce the entire kinematics to the midsurface displacement $\boldsymbol{u}(\xi^1,\xi^2)$ and make the Green-Lagrange strain of the continuum linear in ξ^3,

$$E_{\alpha\beta}(\xi^3) \;=\; \varepsilon_{\alpha\beta} + \xi^3\,\kappa_{\alpha\beta}, \tag{4.21}$$

with the membrane strain and bending (curvature change) tensors

$$\varepsilon_{\alpha\beta} \;=\; \tfrac{1}{2}\big(\bar{a}_{\alpha\beta} - a_{\alpha\beta}\big), \qquad \kappa_{\alpha\beta} \;=\; b_{\alpha\beta} - \bar{b}_{\alpha\beta}, \tag{4.22}$$

that is, the changes of the first and second fundamental forms between the reference and the deformed (barred) configuration [83, 84]. Pre-integration through the thickness yields the internal energy of an isotropic shell,

$$\Pi_{int} \;=\; \tfrac{1}{2}\int_S \Big(t\;\boldsymbol{\varepsilon} : \mathbb{C} : \boldsymbol{\varepsilon} \;+\; \tfrac{t^3}{12}\;\boldsymbol{\kappa} : \mathbb{C} : \boldsymbol{\kappa} \Big)\,\mathrm{d}S, \tag{4.23}$$

with the plane-stress constitutive tensor $\mathbb{C}$ expressed in the contravariant surface basis. Membrane stiffness scales with t and bending stiffness with $t^3/12$, and the ratio of the two energies will serve below as a diagnostic of the load-carrying mode.

For the linear analyses of this review, both measures are linearized about the reference state. Writing the deformed basis as $\bar{\boldsymbol{g}}_\alpha = \boldsymbol{g}_\alpha + \boldsymbol{u}_{,\alpha}$ and dropping the quadratic term $\boldsymbol{u}_{,\alpha}\cdot\boldsymbol{u}_{,\beta}$, the membrane strain becomes the symmetric surface gradient

$$\varepsilon_{\alpha\beta} \;=\; \tfrac{1}{2}\big(\boldsymbol{g}_\alpha\cdot\boldsymbol{u}_{,\beta} + \boldsymbol{g}_\beta\cdot\boldsymbol{u}_{,\alpha}\big), \tag{4.24}$$

dependent only on first derivatives of the displacement. The bending measure is harder. The deformed second fundamental form $\bar{b}_{\alpha\beta} = \bar{\boldsymbol{r}}_{,\alpha\beta}\cdot\bar{\boldsymbol{n}}$ involves the deformed unit normal, whose variation

$$\delta\boldsymbol{n} \;=\; \frac{\delta(\boldsymbol{g}_1\times\boldsymbol{g}_2)}{\|\boldsymbol{g}_1\times\boldsymbol{g}_2\|} \;-\; \boldsymbol{n}\left(\boldsymbol{n}\cdot\frac{\delta(\boldsymbol{g}_1\times\boldsymbol{g}_2)}{\|\boldsymbol{g}_1\times\boldsymbol{g}_2\|}\right) \tag{4.25}$$

is a tangential rotation, since the normal cannot stretch, and historically dominates the implementation effort of any KL code [82, 84]. Carrying the linearization of (4.25) through $\bar{b}_{\alpha\beta}$ and collecting terms by $\boldsymbol{u}_{,1}$, $\boldsymbol{u}_{,2}$ and $\boldsymbol{u}_{,\alpha\beta}$, however, yields a closed-form expression that is *explicitly*

linear in the displacement derivatives,

$$\kappa_{\alpha\beta} \;=\; \boldsymbol{u}_{,\alpha\beta}\cdot\boldsymbol{n}+\frac{1}{\sqrt{\det a}}\Big\{\boldsymbol{u}_{,1}\cdot\big[\boldsymbol{g}_2\times\boldsymbol{r}_{,\alpha\beta}-b_{\alpha\beta}\,(\boldsymbol{g}_2\times\boldsymbol{n})\big]+\boldsymbol{u}_{,2}\cdot\big[\boldsymbol{r}_{,\alpha\beta}\times\boldsymbol{g}_1-b_{\alpha\beta}\,(\boldsymbol{n}\times\boldsymbol{g}_1)\big]\Big\}, \tag{4.26}$$

first derived in this form in the subdivision-shell setting by Cirak, Ortiz and Schröder [25]. The two terms of (4.26) have clean mechanical readings. The term $\boldsymbol{u}_{,\alpha\beta}\cdot\boldsymbol{n}$ is the change of the curvature vector projected on the fixed normal, while the braced term is the change of reading caused by the rotation of the normal itself (the director rotation), at frozen geometry.

For a thin, isotropic, linearly elastic shell in plane stress, the shell elasticity tensor is built from the inverse first fundamental form,

$$C^{\alpha\beta\gamma\delta} \;=\; \frac{E}{1-\nu^2}\Big[\nu\,a^{\alpha\beta}a^{\gamma\delta}+\tfrac{1-\nu}{2}\big(a^{\alpha\gamma}a^{\beta\delta}+a^{\alpha\delta}a^{\beta\gamma}\big)\Big], \tag{4.27}$$

which is Hooke's law of plane stress transported to the curvilinear basis, with the Cartesian metric δ^{ij} replaced by the surface metric $a^{\alpha\beta}$. Thickness integration gives the membrane force and bending moment resultants

$$n^{\alpha\beta}=t\;C^{\alpha\beta\gamma\delta}\,\varepsilon_{\gamma\delta}, \qquad m^{\alpha\beta}=\frac{t^3}{12}\;C^{\alpha\beta\gamma\delta}\,\kappa_{\gamma\delta}, \tag{4.28}$$

contravariant, as energy conjugates of the covariant strains must be (Section 4.4.2). For computation, $C^{\alpha\beta\gamma\delta}$ is arranged as a symmetric Voigt matrix in the ordering $[\,11,22,12\,]$, acting on $[\varepsilon_{11},\varepsilon_{22},2\varepsilon_{12}]^{\top}$ and $[\kappa_{11},\kappa_{22},2\kappa_{12}]^{\top}$,

$$\boldsymbol{D} \;=\; \frac{E}{1-\nu^2}\begin{bmatrix} (a^{11})^2 & \nu\,a^{11}a^{22}+(1-\nu)(a^{12})^2 & a^{11}a^{12} \\ \nu\,a^{11}a^{22}+(1-\nu)(a^{12})^2 & (a^{22})^2 & a^{22}a^{12} \\ a^{11}a^{12} & a^{22}a^{12} & \frac{1-\nu}{2}\,a^{11}a^{22}+\frac{1+\nu}{2}\,(a^{12})^2 \end{bmatrix}, \tag{4.29}$$

with $\boldsymbol{D}^m = t\,\boldsymbol{D}$ and $\boldsymbol{D}^b = \frac{t^3}{12}\,\boldsymbol{D}$. In an orthonormal basis ($a^{11}=a^{22}=1$, $a^{12}=0$) the matrix (4.29) reduces to the classical plane-stress matrix, and its 13 and 23 entries, which couple normal strains to in-plane shear, are nonzero *only* when $a^{12}\neq 0$, precisely on the non-orthogonal parametrizations that reconstructed geometry routinely produces. A solver validated only on axis-aligned patches never exercises these entries.

The curvature change $\kappa_{\alpha\beta}$ contains second derivatives of the displacement, so a conforming primal discretization requires C^1 continuity, which is unavailable to standard finite elements and historically the reason KL theory ceded ground to Reissner-Mindlin elements despite its leaner kinematics. The isogeometric KL shell of Kiendl et al. [84] resolved this impasse. NURBS of degree $p\geq 2$ are globally C^{p-1} within a patch, enabling a rotation-free discretization with three displacement degrees of freedom per control point and none of the rotational machinery of classical shell elements. Geometry and displacement share the basis,

$$\boldsymbol{r}(\xi)=\sum_i R_i(\xi)\,\boldsymbol{P}_i, \qquad \boldsymbol{u}(\xi)=\sum_i R_i(\xi)\,\boldsymbol{u}_i, \tag{4.30}$$

so that $\boldsymbol{u}_{,\alpha}=\sum_i R_{i,\alpha}\boldsymbol{u}_i$ and $\boldsymbol{u}_{,\alpha\beta}=\sum_i R_{i,\alpha\beta}\boldsymbol{u}_i$ substitute directly into (4.24) and (4.26), making both strain vectors linear in the control-point displacements, $\boldsymbol{\varepsilon}=\boldsymbol{B}^m\hat{\boldsymbol{u}}_e$ and $\boldsymbol{\kappa}=\boldsymbol{B}^b\hat{\boldsymbol{u}}_e$, where $\hat{\boldsymbol{u}}_e$ stacks the displacement vectors of the $(p+1)(q+1)$ control points supported on the element.

The membrane block of control point i (three rows, each a 1×3 Cartesian row) reads

$$\boldsymbol{B}_i^m \;=\; \begin{bmatrix} R_{i,1}\,\boldsymbol{g}_1^\top \\ R_{i,2}\,\boldsymbol{g}_2^\top \\ R_{i,1}\,\boldsymbol{g}_2^\top + R_{i,2}\,\boldsymbol{g}_1^\top \end{bmatrix}, \tag{4.31}$$

and the bending row for component $(\alpha\beta)$ follows verbatim from (4.26),

$$\boldsymbol{B}^b_{(\alpha\beta),i} \;=\; \Big(R_{i,\alpha\beta}\,\boldsymbol{n} + \tfrac{1}{\sqrt{\det a}} \big[R_{i,1}\big(\boldsymbol{g}_2 \times \boldsymbol{r}_{,\alpha\beta} - b_{\alpha\beta}(\boldsymbol{g}_2 \times \boldsymbol{n})\big) + R_{i,2}\big(\boldsymbol{r}_{,\alpha\beta} \times \boldsymbol{g}_1 - b_{\alpha\beta}(\boldsymbol{n} \times \boldsymbol{g}_1)\big) \big] \Big)^\top, \tag{4.32}$$

stacked in Voigt order (11), (22), $2 \times$ (12). The element (knot-span) stiffness matrix is then

$$\boldsymbol{K}_e \;=\; \int_{\hat{\Omega}_e} \Big[(\boldsymbol{B}^m)^\top \boldsymbol{D}^m\, \boldsymbol{B}^m + (\boldsymbol{B}^b)^\top \boldsymbol{D}^b\, \boldsymbol{B}^b \Big] \sqrt{\det a}\; \mathrm{d}\xi^1 \mathrm{d}\xi^2, \tag{4.33}$$

evaluated by Gauss quadrature per knot span. At each Gauss point the chain of maps of Section 4.4.1 is traversed in its surface form. One evaluates R_i, $R_{i,\alpha}$ and $R_{i,\alpha\beta}$, forms $\boldsymbol{g}_\alpha$ and $\boldsymbol{r}_{,\alpha\beta}$ from the control points, assembles and inverts $a_{\alpha\beta}$, computes $\boldsymbol{n}$, $b_{\alpha\beta}$ and $\sqrt{\det a}$, builds $\boldsymbol{D}$ from $a^{\alpha\beta}$ by (4.29) and the blocks (4.31)-(4.32), and accumulates with the quadrature weight.

The load vector completes the discrete system. A distributed surface load $\boldsymbol{f}$, a pressure or a weight per unit area, contributes to the three-component block of control point i

$$\boldsymbol{F}_i \;=\; \int_{\hat{\Omega}} R_i\, \boldsymbol{f}\, \sqrt{\det a}\; \mathrm{d}\xi^1 \mathrm{d}\xi^2, \tag{4.34}$$

evaluated by the same quadrature as the stiffness, and a concentrated force $\boldsymbol{F}_k$ acting at the point (ξ_k^1, ξ_k^2) adds $R_i(\xi_k^1, \xi_k^2)\,\boldsymbol{F}_k$, exactly as for the beam. With the supports imposed through the control-point displacements, the system $\boldsymbol{K}\boldsymbol{d} = \boldsymbol{F}$ is solved for the control-point displacements.

The formulation is developed fully in Kiendl's dissertation [82]. Extensions cover general hyperelastic materials and large deformations [86], explicit dynamic membrane counterparts [87], and inverse shell analysis (iFEM) for shape sensing of thin structures [88]. The line remains open. Recent formulations enrich the kinematics hierarchically so that the warping the classical theory discards is recovered, which matters for layups alternating stiff and soft layers [89], and add large deformation frictional contact to the rotation-free element [90].

5. SHAPE OPTIMIZATION ON PARAMETRIC MODELS

Shape optimization is where the isogeometric model is put to work. The parametric surface of Chapter 3 and the analysis of Chapter 4 are joined in a loop that changes the shape, analyses it and changes it again, and in that loop the fact that the geometric model and the analysis model are one and the same object stops being a matter of principle and becomes a matter of cost. Like Chapter 3 with the routes for improving a fit, this chapter is an overview of what isogeometric analysis is used for rather than a theory of optimization. Optimization of any kind would fill a theoretical chapter of its own, so what follows keeps to the basic ideas of isogeometric shape optimization, to what the isogeometric setting contributes to them and to the difficulties one meets along the way.

The advantage over the classical route is structural rather than incidental. In a finite element treatment the geometry is meshed and the mesh carries the analysis, so the design variables belong to a separate description of the shape and every design step has to pass through meshing again. The correspondence between the two models is rebuilt at every iterate, the discretization error changes together with the shape, and the derivatives of the response with respect to the design have to be transferred across that gap. In isogeometric analysis the control points that describe the surface are the degrees of freedom of the analysis, so they are already design variables. Moving one of them changes the geometry and the discretization in the same stroke, nothing is re-meshed, nothing is transferred, and the sensitivities follow from the same basis functions that produced the analysis [10, 91]. This chapter reviews what the literature makes of that, the choice of design variables, the sensitivities that a gradient method needs, and the strategies that take over when gradients are unavailable or the landscape is multimodal.

5.1. Isogeometric Shape Optimization

Shape optimization on CAD-derived parametric models predates IGA, but IGA removes its classical bottleneck. With the analysis model and the design model sharing one NURBS description, every design update is analysis-ready without remeshing, and design velocities are available analytically from the basis functions. The foundational formulation is due to Wall, Frenzel and Cyron [91], who optimized structural shapes with NURBS control points as design variables and demonstrated the seamless design-analysis loop. Hassani et al. [92] and the shape optimization chapters of Kiendl's dissertation [82] extended the framework to shell structures, and Bandara and Cirak [93] generalized the geometric layer to multiresolution subdivision surfaces, covering unstructured layouts with extraordinary points. The recent literature carries the same formulation in several directions at once, to shells assembled from patches that do not match along their interfaces [94], to multi-patch free-form surfaces under buckling constraints [95], to problems with two competing objectives on a Kirchhoff-Love shell [96] and to beams and lattice structures at large deformations [97]. This chapter keeps to shape optimization, in which the topology of the object is fixed and only its boundary moves, but the isogeometric model is increasingly used for topology optimization as well, where material is distributed inside a fixed domain and the same spline description serves as the design field and as the analysis basis. That branch has produced formulations which treat shape, thickness and topology as coupled unknowns on free-form shells [98] and frameworks in which a trained network carries part of

the optimization [99]. A standard problem statement minimizes compliance or a stress measure,

$$\min_{\boldsymbol{s}} \; J\big(\boldsymbol{s}, \boldsymbol{u}(\boldsymbol{s})\big) \quad \text{s.t.} \quad K(\boldsymbol{s})\,\boldsymbol{u} = \boldsymbol{F}(\boldsymbol{s}), \qquad g_i(\boldsymbol{s}) \le 0, \tag{5.1}$$

where $\boldsymbol{s}$ collects the design variables. The constraints g_i enforce manufacturing or physics-driven bounds such as area and volume budgets. On reconstructed geometry they must also enforce *element quality*, through lower bounds on the surface Jacobian that prevent the optimizer from degenerating the parametrization while exploiting it, so the analysis-suitability theme of Section 3.1.3 reappears as an optimization constraint [100].

The choice of design variables deserves elaboration, because it governs both the conditioning of the problem and the cost of its derivatives. Letting every control point move in all three Cartesian directions triples the dimension of the design space and hands the optimizer many directions that change the shape very little, so the design space is restricted instead. The standard remedy, adopted throughout the isogeometric shape optimization literature [82, 91, 93], gives each designed control point $\boldsymbol{P}_k$ a single scalar offset α_k along one fixed direction $\boldsymbol{n}_k$, evaluated once on the *initial* geometry and thereafter frozen. For the examples of this chapter that direction is the surface normal,

$$\boldsymbol{P}_k(\alpha_k) \;=\; \boldsymbol{P}_k^0 + \alpha_k\,\boldsymbol{n}_k, \qquad k = 1, \dots, m, \tag{5.2}$$

where $\boldsymbol{P}_k^0$ is the control point of the initial model and m the number of design variables. Freezing the directions makes the design map (5.2) *linear*. The geometry depends affinely on the design variables, the design velocity of a control point is the constant vector $\partial \boldsymbol{P}_k / \partial \alpha_k = \boldsymbol{n}_k$, and the chain rule from Cartesian to design sensitivities is an exact, once-assembled sparse multiplication rather than a re-linearization at every iterate. The price is that for very large shape changes the frozen directions cease to be normal to the current surface. In typical applications the offsets remain a modest fraction of the local feature size, the linear map is retained throughout, and its exactness is what allows the adjoint gradients of Section 5.2 to be validated against finite differences to near machine-limited agreement. Alternative design parametrizations from the literature, such as free-form deformation boxes wrapped around the model or coarse "design" splines controlling a finer "analysis" spline through refinement operators, fit the same template, a linear (or explicitly differentiable) map from few design variables to many control points, with the geometry primitives of Chapter 2 supplying the map [10, 93].

5.2. Sensitivity Analysis

Gradient-based solution of (5.1) at realistic DOF counts requires adjoint sensitivities. For compliance $J = \boldsymbol{F}^{\mathsf{T}}\boldsymbol{u} = \boldsymbol{u}^{\mathsf{T}} K \boldsymbol{u}$ the derivation takes three lines. Differentiating the state equation $K\boldsymbol{u} = \boldsymbol{F}$ with respect to a design variable s_k gives

$$K \frac{\partial \boldsymbol{u}}{\partial s_k} \;=\; \frac{\partial \boldsymbol{F}}{\partial s_k} - \frac{\partial K}{\partial s_k}\,\boldsymbol{u}, \tag{5.3}$$

while the chain rule applied to $J = \boldsymbol{F}^{\mathsf{T}}\boldsymbol{u}$ yields $\mathrm{d}J/\mathrm{d}s_k = (\partial \boldsymbol{F}/\partial s_k)^{\mathsf{T}}\boldsymbol{u} + \boldsymbol{F}^{\mathsf{T}}(\partial \boldsymbol{u}/\partial s_k)$. Compliance is *self-adjoint*, its adjoint vector coinciding with the state $\boldsymbol{u}$ itself. Substituting $\boldsymbol{F}^{\mathsf{T}} = \boldsymbol{u}^{\mathsf{T}} K$

(symmetry of K) and eliminating $\partial \boldsymbol{u}/\partial s_k$ with (5.3) produces the classical identity

$$\frac{\mathrm{d}J}{\mathrm{d}s_k} \;=\; 2\,\boldsymbol{u}^{\top}\frac{\partial \boldsymbol{F}}{\partial s_k} \;-\; \boldsymbol{u}^{\top}\frac{\partial K}{\partial s_k}\,\boldsymbol{u}, \tag{5.4}$$

which requires *no additional linear solves*, since one factorization of K serves all m design derivatives, each reduced to derivatives of element matrices [50, 91, 93]. Because the design map (5.2) is linear, $\partial K/\partial s_k$ and $\partial \boldsymbol{F}/\partial s_k$ are derivatives along fixed, known perturbation directions, and (5.4) is exact up to how those element derivatives are computed. Fully analytic differentiation of the Kirchhoff-Love element matrices through the curvature terms of Section 4.4.3 is laborious. The pragmatic and widely used alternative is the *semi-analytic* scheme, in which $\partial K_e/\partial s_k$ is approximated by finite differences of the element matrices. Only the control points of element e are perturbed, its stiffness is re-evaluated, and the difference is taken, while the global assembly and the adjoint identity remain analytic. The order of the element-level differences matters in practice, and the semi-analytic scheme has a known pathology for bending-dominated structures. Rigid-body rotation errors of the perturbed elements pollute one-sided differences, which makes central differences the default recommendation of the shape optimization literature [91, 93]. On scanned hull geometry the size of the effect is easy to see. One-sided forward differences of the shell element matrices give total design gradients that agree with a global finite-difference check only to a relative error of order 10^{-3}, which is marginal for a line-search method and fatal for quasi-Newton curvature estimates. Central differences at the same step size recover agreement of order 10^{-7}, effectively exact gradients at twice the element-evaluation cost. A third route sidesteps the choice between the analytic and the semi-analytic scheme by deriving the element derivatives automatically from the analysis code itself, which removes the step size from the discussion at the price of a dependence on the tooling [101].

Stress-based objectives are a different matter, and the obstacle is differentiability. The quantity an engineer designs against is the largest von Mises stress in the structure, and that maximum is not a differentiable function of the design, because the place where it occurs jumps from one element to another as the shape changes and the gradient jumps with it. A gradient method has nothing dependable to descend on in such a case. The way out is either a smooth substitute for the maximum, which buys differentiability at the price of conservatism and of a parameter to tune, or a method that needs no gradients at all, of the kind Section 5.3 describes, which costs a full analysis for every candidate design and is therefore far more demanding in computation and in time. Neither route is settled practice, least of all on reconstructed geometry, where the stress field carries the error of the fit before any constraint is imposed. The isogeometric literature has taken the question up, although almost entirely in topology optimization rather than in shape optimization, either by aggregating the local stresses into one differentiable measure or by imposing them as many separate constraints [102, 103, 104, 105, 106]. The study that comes closest to the setting of this chapter combines a stress-based objective with a level-set description on trimmed multi-patch shells [107]. Since stress is what decides whether a design is admissible at all, its treatment in the shape optimization of reconstructed geometry remains a natural subject for further research.

As an illustration of the reviewed formulation, Figure 5.1 shows three scanned parts, a boat hull, a propeller and a suspension control arm. The upper row holds the fitted models and the lower row the shapes that compliance minimization returns on them, with the control points of each model serving as the design variables [91]. The offsets stay within a modest fraction of the local feature size, so what the optimizer produces is a redistribution of curvature over the same object rather than a different one.

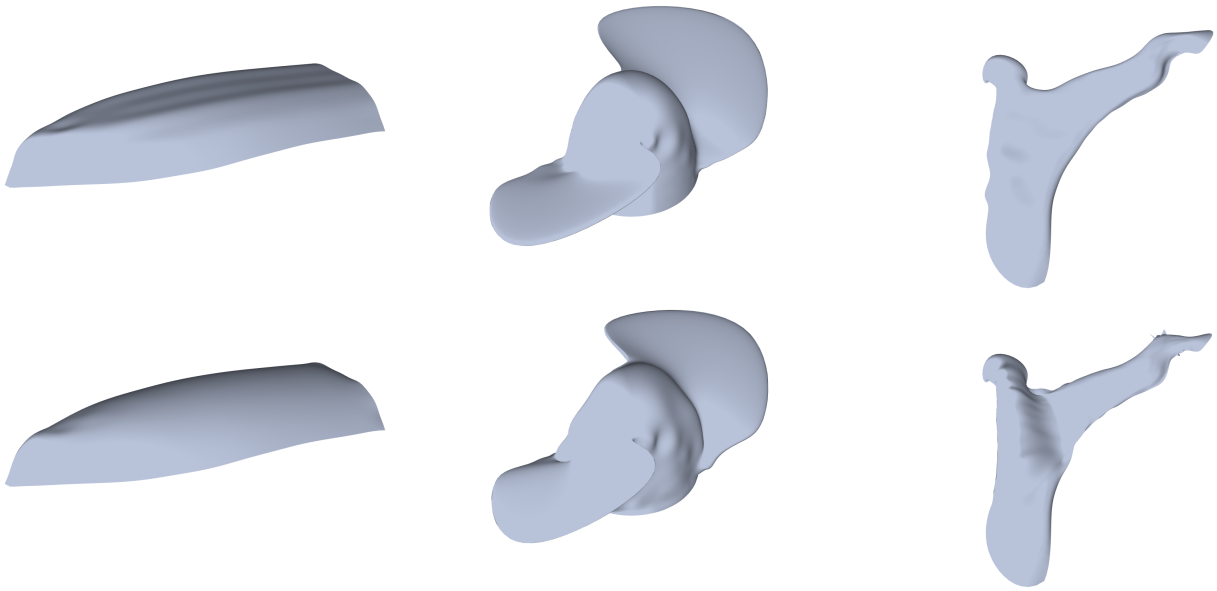


Figure 5.1. Fitted (top) and compliance-optimized (bottom) geometry of a scanned hull, propeller and control arm.

5.3. Multiresolution, Evolutionary and Surrogate Strategies

Three further ideas shape how such an optimization is run, and each answers a different difficulty.

The first is the resolution of the design space, which is itself a design decision. Optimizing on the full control net from the start exposes the optimizer to a large set of oscillatory modes that change the shape very little and the objective even less, which slows convergence and, on a reconstructed model, erodes element quality. Multiresolution strategies therefore start on a coarse net and refine it in stages through the knot insertion of Section 2.4, which carries a coarse design to the finer level without changing the geometry. The staged design spaces are nested, so nothing attained at one stage is lost at the next, and the idea is developed in that form by Bandara and Cirak [93].

The second is what to do when gradients are unavailable or unreliable, or the landscape is multimodal. Evolutionary methods work directly on populations of designs, encoding each candidate as its control points or their normal offsets, evaluating it with the analysis model and evolving the population by selection, crossover and mutation, with constraints handled by penalties [108]. They need no derivatives and they search globally, and they pay for it in analyses, since every individual of every generation costs a full solve, which confines them to models cheap enough to afford thousands of them.

The third idea addresses that cost. A *surrogate* replaces the solver inside the loop with a neural network that is trained to predict the outcome of the analysis, and the use of such networks on structural problems is by now a subject of its own [109, 110, 111]. A network trained on a modest sample of the design space reproduces a smooth response such as compliance closely enough to drive the search and answers in microseconds instead of in a full analysis, so a budget of thousands of solves collapses to the sampling of the design space and the training of the network. A rougher response such as a stress measure needs far more samples for the same accuracy, and the count grows quickly with the dimension of the design space, so a surrogate pays off where many evaluations of a smooth response are needed and not where a single gradient

descent with cheap adjoints (5.4) would do.

Table 5.1 condenses the three method families reviewed in this chapter into their defining trade-offs. Adjoint gradient methods dominate whenever derivatives are available, because one solve and one adjoint evaluation price an entire gradient regardless of the number of design variables, at the cost of local convergence and the implementation burden of sensitivities. Evolutionary algorithms invert this profile. They explore globally and require no derivatives, but every individual in every generation costs a full analysis, which confines them to models cheap enough to afford thousands of solves. Surrogate-assisted search occupies the middle ground by converting the evaluation cost into an up-front sampling budget, and it pays off precisely when many evaluations of a smooth response are needed.

Table 5.1. Comparison of optimization strategies for shape design on parametric models.

Strategy	Cost per design evaluation	Search scope	Derivative requirement	Typical use
Adjoint gradient methods [91, 93]	one solve plus one adjoint per iterate	local	analytic or semi-analytic sensitivities	large design spaces, compliance objectives
Evolutionary algorithms [108]	one full solve per individual	global	none	multimodal landscapes, small models, mixed variables
Surrogate-assisted search [109, 112]	near zero after sampling and training	global or local	none for the surrogate	population methods, multi-objective and robust design

Isogeometric shape optimization on designed geometry is well established. For compliance-type objectives with adjoint sensitivities, the formulations of Wall et al., Kiendl, and Bandara and Cirak constitute a mature technology in which design and analysis share one spline model and gradients are exact and cheap [82, 91, 93]. The open questions concentrate at the transfer to reconstructed geometry. There the control points are not placed by a designer but inherited from the fit, so the parametrization decides where the design space has any resolution at all, and the quality of the elements decides how far the optimizer may push before the analysis degrades. On such models it is the parametrization rather than the optimizer that binds [30, 100]. The shapes of Figure 5.1 were obtained on models fitted with a uniform net, and the route that suggests itself is to combine the optimization with the reparametrization of Section 3.3, so that the control points are redistributed toward the regions the design has to move before the optimizer is let loose. Whether that yields a more faithful optimum on reconstructed geometry, and how the two should be interleaved, is open. Stress-based objectives remain markedly harder than compliance for the reason given above, and the work that exists on them belongs to topology optimization rather than to shape optimization on a fitted surface [104, 107]. A formulation that treats shape and parametrization as coupled unknowns, subject to fitting fidelity and element-quality constraints, has not emerged in the literature, and it stands as the principal open problem at this end of the reviewed field.

6. CONCLUSION

The first conclusion of this review is that the five areas it covers are not five subjects but one object seen at five stages of its life. A spline surface is written in the language of Chapter 2, built in Chapter 3 out of a measurement that has none of its structure, made the discretization of the analysis in Chapter 4 without being converted into anything else, and handed to the optimizer of Chapter 5 with its control points as the design variables. Because it is one object, every decision made at an early stage is inherited by all that come after. The map chosen for the mesh fixes where the control points of the fitted surface sit, the control points fix the resolution of the analysis and the freedom of the optimizer, and neither the analysis nor the optimizer can recover what the earlier stages did not give them. The literature of each stage is mature on its own terms. Convex combination maps come with guarantees of injectivity, regularized least-squares fitting is well understood, single-patch isogeometric analysis and the Kirchhoff-Love shell are textbook material, and adjoint sensitivities on the shared model are standard. What the literature does not offer is a way of judging one stage by the needs of the next, since each area measures its own quality with its own instrument, distortion for the map, geometric error for the fit, convergence for the analysis and the objective for the optimizer.

The second conclusion comes from the application examples that accompanied the reviewed methods, and it is that this separation is visible on real scanned parts as soon as the whole chain is run on them. The discrete harmonic map parametrized every part without failure, and in doing so compressed the parametric area at the bow of the hull and along the edges of the blade, which is exactly where the geometry demands the most resolution. The single-patch fit reached a small mean error on all three parts, while its largest error settled in narrow bands along the features of each object and stayed there under every refinement of the control net, so the level of that error was decided by the map and by the single patch before the linear system was assembled. The shell and the beam were solved on the reconstructed geometry with the accuracy the isogeometric literature promises, but that accuracy is bounded by the quality of the Jacobian of the very map that the fit had inherited. Compliance minimization then moved the shape only where the fitted model had control points to move, which is to say within a design space that the parametrization had settled before the first iteration. Each stage therefore behaved as its own literature describes, and each was blind to what it cost the stage after it. The present state of the work is a chain of single-patch links that has been assembled end to end but has not yet been judged as a whole, and that is the finding on which what follows is built.

The direction in which the work continues follows from that finding. The quality of a reconstructed model has so far been assessed where it is produced, in the geometric error of the fit, whereas the model is consumed elsewhere, in the analysis and in the optimization. The next step is therefore to carry the assessment downstream, and to ask of every decision made while the model is built, the choice of the map, the distribution of its parametric area, the placement of the knots and the division of the object into patches, what it does to the accuracy of the analysis and to the design space of the optimizer. The reviewed literature shows that the tools for this exist. Reparametrization can redistribute parametric area toward the regions the design has to reach while keeping the map injective by construction, the division of an object into several patches along its own features removes the obstruction that a globally smooth basis meets at a sharp edge, and the couplings that restore continuity across the seams of a multi-patch model are available for the analysis. None of these has been applied to a scanned part with the downstream analysis as the measure of success, and doing so, first on the parts already used in this review, is the concrete task the work turns to next. Which of the two routes, the reparametrization of a single

patch or the division into several, proves the more useful will be decided by that comparison rather than in advance.

Beyond that task the review leaves open the wider question that gives the subject its interest. Because one spline model carries the geometry, the analysis and the design at the same time, the same surface can be analysed against more than one kind of physics and optimized against more than one criterion, a propeller against the flow that loads it as much as against the stresses it carries. The isogeometric setting removes the step that usually makes such loops impractical, the preparation of a separate mesh for each discipline at every design iterate, and leaves the cost of the analyses themselves, which is where surrogate models belong. Whether the doctoral research reaches that far will depend on what the first step shows. The review has been written to lay out the ground on which the next steps are taken.

REFERENCES

[1] Rogers, D. F.: “An Introduction to NURBS: With Historical Perspective”, Morgan Kaufmann, 2001.

[2] Farin, G.: “Curves and Surfaces for CAGD: A Practical Guide”, Morgan Kaufmann, 2002.

[3] Várady, T.; Martin, R. R.; Cox, J.: “Reverse engineering of geometric models-an introduction”, Computer-Aided Design, vol. 29, no. 4, pp. 255-268, 1997.

[4] Hughes, T. J. R.; Cottrell, J. A.; Bazilevs, Y.: “Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement”, Computer Methods in Applied Mechanics and Engineering, vol. 194, no. 39-41, pp. 4135-4195, 2005.

[5] Piegl, L.; Tiller, W.: “The NURBS Book”, Springer, Berlin, Heidelberg, 1997.

[6] de Boor, C.: “A Practical Guide to Splines”, Springer, New York, 1978.

[7] Campen, M.: “Partitioning surfaces into quadrilateral patches: A survey”, Computer Graphics Forum, vol. 36, no. 8, pp. 567-588, 2017.

[8] Cox, M. G.: “The numerical evaluation of B-splines”, IMA Journal of Applied Mathematics, vol. 10, no. 2, pp. 134-149, 1972.

[9] de Boor, C.: “On calculating with B-splines”, Journal of Approximation Theory, vol. 6, no. 1, pp. 50-62, 1972.

[10] Cottrell, J. A.; Hughes, T. J. R.; Bazilevs, Y.: “Isogeometric Analysis: Toward Integration of CAD and FEA”, John Wiley & Sons, 2009.

[11] Nguyen, V. P. et al.: “Isogeometric analysis: An overview and computer implementation aspects”, Mathematics and Computers in Simulation, vol. 117, pp. 89-116, 2015.

[12] Boehm, W.: “Inserting new knots into B-spline curves”, Computer-Aided Design, vol. 12, no. 4, pp. 199-201, 1980.

[13] Cohen, E.; Lyche, T.; Riesenfeld, R.: “Discrete B-splines and subdivision techniques in computer-aided geometric design and computer graphics”, Computer Graphics and Image Processing, vol. 14, no. 2, pp. 87-111, 1980.

[14] Prautzsch, H.: “Degree elevation of B-spline curves”, Computer Aided Geometric Design, vol. 1, no. 2, pp. 193-198, 1984.

[15] Dierckx, P.: “Curve and Surface Fitting with Splines”, Oxford University Press, 1993.

[16] Forsey, D. R.; Bartels, R. H.: “Hierarchical B-spline refinement”, ACM SIGGRAPH Computer Graphics, vol. 22, no. 4, pp. 205-212, 1988.

[17] Kraft, R.: “Adaptive and linearly independent multilevel B-splines”, Surface Fitting and Multiresolution Methods, Vanderbilt University Press, pp. 209-218, 1997.

[18] Giannelli, C.; Jüttler, B.; Speleers, H.: “THB-splines: The truncated basis for hierarchical splines”, Computer Aided Geometric Design, vol. 29, no. 7, pp. 485-498, 2012.

[19] Bracco, C. et al.: “Adaptive isogeometric methods with hierarchical splines: An overview”, Discrete and Continuous Dynamical Systems, vol. 39, no. 1, pp. 241-261, 2019.

[20] Giannelli, C. et al.: “Efficient alternating and joint distance minimization methods for adaptive spline surface fitting”, Graphical Models, vol. 137, pp. 101251, 2024.

[21] Sederberg, T. W. et al.: “T-splines and T-NURCCs”, ACM Transactions on Graphics, vol. 22, no. 3, pp. 477-484, 2003.

[22] Scott, M. A. et al.: “Local refinement of analysis-suitable T-splines”, Computer Methods in Applied Mechanics and Engineering, vol. 213-216, pp. 206-222, 2012.

[23] Dokken, T.; Lyche, T.; Pettersen, K. F.: “Polynomial splines over locally refined box-partitions”, Computer Aided Geometric Design, vol. 30, no. 3, pp. 331-356, 2013.

[24] Patrizi, F. et al.: “Adaptive refinement with locally linearly independent LR B-splines: Theory and applications”, Computer Methods in Applied Mechanics and Engineering, vol. 369, pp. 113230, 2020.

[25] Cirak, F.; Ortiz, M.; Schröder, P.: “Subdivision surfaces: a new paradigm for thin-shell finite-element analysis”, International Journal for Numerical Methods in Engineering, vol. 47, no. 12, pp. 2039-2072, 2000.

[26] Thomas, D. C. et al.: “U-splines: Splines over unstructured meshes”, Computer Methods in Applied Mechanics and Engineering, vol. 401, pp. 115515, 2022.

[27] Kapl, M.; Sangalli, G.; Takacs, T.: “Construction of analysis-suitable G^1 planar multi-patch parameterizations”, Computer-Aided Design, vol. 97, pp. 41-55, 2018.

[28] Farahat, A. et al.: “A locally based construction of analysis-suitable G^1 multi-patch spline surfaces”, Computers and Mathematics with Applications, vol. 168, pp. 46-57, 2024.

[29] Floater, M. S.; Hormann, K.: “Surface parameterization: a tutorial and survey”, Advances in Multiresolution for Geometric Modelling, Springer, pp. 157-186, 2005.

[30] Marinić-Kragić, I.; Vučina, D.; Ćurković, M.: “Adaptive re-parameterization based on arbitrary scalar fields for shape optimization and surface fitting”, Engineering Applications of Artificial Intelligence, vol. 67, pp. 39-51, 2018.

[31] Tutte, W. T.: “How to draw a graph”, Proceedings of the London Mathematical Society, vol. s3-13, no. 1, pp. 743-767, 1963.

[32] Floater, M. S.: “Parametrization and smooth approximation of surface triangulations”, Computer Aided Geometric Design, vol. 14, no. 3, pp. 231-250, 1997.

[33] Pinkall, U.; Polthier, K.: “Computing discrete minimal surfaces and their conjugates”, Experimental Mathematics, vol. 2, no. 1, pp. 15-36, 1993.

[34] Cohen, E. et al.: “Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis”, Computer Methods in Applied Mechanics and Engineering, vol. 199, no. 5-8, pp. 334-356, 2010.

[35] Xu, G. et al.: “Parameterization of computational domain in isogeometric analysis: Methods and comparison”, Computer Methods in Applied Mechanics and Engineering, vol. 200, no. 23-24, pp. 2021-2031, 2011.

[36] Desbrun, M.; Meyer, M.; Alliez, P.: “Intrinsic parameterizations of surface meshes”, Computer Graphics Forum, vol. 21, no. 3, pp. 209-218, 2002.

[37] Lévy, B. et al.: “Least squares conformal maps for automatic texture atlas generation”, ACM Transactions on Graphics, vol. 21, no. 3, pp. 362-371, 2002.

[38] Choi, G. P. T.; Giri, A.; Kumar, L.: “Adaptive area-preserving parameterization of open and closed anatomical surfaces”, Computers in Biology and Medicine, vol. 148, pp. 105715, 2022.

[39] Lai, Y.; Hu, S.; Pottmann, H.: “Surface fitting based on a feature sensitive parametriza-

tion", Computer-Aided Design, vol. 38, no. 7, pp. 800-807, 2006.

[40] Hu, K. et al.: "Feature-aligned surface parameterization using secondary Laplace operator and Loop subdivision", Procedia Engineering, vol. 163, pp. 186-198, 2016.

[41] Ćurković, M.; Vučina, D.: "3D shape acquisition and integral compact representation using optical scanning and enhanced shape parameterization", Advanced Engineering Informatics, vol. 28, no. 2, pp. 111-126, 2014.

[42] Ćurković, M.; Vučina, D.; Ćurković, A.: "Enhanced 3D parameterization for integrated shape synthesis by fitting parameter values to point sets", Integrated Computer-Aided Engineering, vol. 24, no. 3, pp. 241-260, 2017.

[43] Miao, Y.; Pajarola, R.; Feng, J.: "Curvature-aware adaptive re-sampling for point-sampled geometry", Computer-Aided Design, vol. 41, no. 6, pp. 395-403, 2009.

[44] Chen, C. et al.: "An adaptive hybrid sampling method for free-form surfaces based on geodesic distance", Sensors, vol. 23, no. 6, pp. 3224, 2023.

[45] Lu, L.; Zhao, S.: "High-quality point sampling for B-spline fitting of parametric curves with feature recognition", Journal of Computational and Applied Mathematics, vol. 345, pp. 286-294, 2018.

[46] Liu, M.; Sun, C.; Ge, S.: "Engineering-scale B-spline surface reconstruction using a hungry predation algorithm, with validation on ship hulls", Applied Sciences, vol. 15, no. 21, pp. 11471, 2025.

[47] Lee, D. et al.: "Optimizing B-spline surface reconstruction for sharp feature preservation", Proceedings of the 10th Annual Computing and Communication Workshop and Conference (CCWC), pp. 359-364, 2020.

[48] Samardžić, D.; Ćurković, M.: "Shape synthesis with optimization of parametric values grid and THB-spline parameterization model", Proceedings of the 10th International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, 2025.

[49] Gavin, H. P.: "The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems", Department of Civil and Environmental Engineering, Duke University, 2024.

[50] Nocedal, J.; Wright, S. J.: "Numerical Optimization", Springer, New York, 2006.

[51] Wang, J. et al.: "Efficient analysis-suitable T-spline fitting for freeform surface reconstruction and intelligent sampling", Precision Engineering, vol. 66, pp. 417-428, 2020.

[52] Marinić-Kragić, I.; Vučina, D.; Ćurković, M.: "Efficient shape parameterization method for multidisciplinary global optimization and application to integrated ship hull shape optimization workflow", Computer-Aided Design, vol. 80, pp. 61-75, 2016.

[53] Ji, Y. et al.: "Curvature-based r-adaptive planar NURBS parameterization method for isogeometric analysis using bi-level approach", Computer-Aided Design, vol. 150, pp. 103305, 2022.

[54] Ćurković, M.; Ćurković, A.; Vučina, D.: "Novel re-parameterization for shape optimization and comparison with knot-based gradient fitting method", Computer Methods in Applied Mechanics and Engineering, vol. 336, pp. 304-332, 2018.

[55] Chen, X. et al.: "B-spline curve fitting based on dynamic adjustment of knot vector using feature points", PLoS ONE, vol. 20, no. 6, pp. e0325458, 2025.

[56] Bilucan, F.; Ergun, S.: "A novel explainable AI driven framework for parametric knot

vector estimation in NURBS surfaces", Applied Sciences, vol. 16, no. 3, pp. 1667, 2026.

[57] Soille, P.: "Morphological Image Analysis: Principles and Applications", Springer, Berlin, Heidelberg, 1999.

[58] Ćurković, M. et al.: "Image segmentation in shape synthesis, shape optimization and reverse engineering", Computer Science & Information Technology (CS & IT), pp. 279-284, 2022.

[59] Bommes, D.; Zimmer, H.; Kobbelt, L.: "Mixed-integer quadrangulation", ACM Transactions on Graphics, vol. 28, no. 3, pp. 77:1-77:10, 2009.

[60] Pietroni, N. et al.: "Reliable feature-line driven quad-remeshing", ACM Transactions on Graphics, vol. 40, no. 4, pp. 155:1-155:17, 2021.

[61] Eck, M.; Hoppe, H.: "Automatic reconstruction of B-spline surfaces of arbitrary topological type", Proceedings of SIGGRAPH '96, pp. 325-334, 1996.

[62] Coradello, L. et al.: "Hierarchically refined isogeometric analysis of trimmed shells", Computational Mechanics, vol. 66, pp. 431-447, 2020.

[63] Buffa, A.; Puppi, R.; Vázquez, R.: "A minimal stabilization procedure for isogeometric methods on trimmed geometries", SIAM Journal on Numerical Analysis, vol. 58, no. 5, pp. 2711-2735, 2020.

[64] Hiemstra, R. R. et al.: "Towards untrimmed NURBS: CAD embedded reparameterization of trimmed B-rep geometry using frame-field guided global parameterization", Computer Methods in Applied Mechanics and Engineering, vol. 369, pp. 113227, 2020.

[65] Wei, Z.; Wei, X.: "Scalable field-aligned reparameterization for trimmed NURBS", Engineering with Computers, vol. 41, pp. 2043-2067, 2025.

[66] Farahat, A. et al.: "Isogeometric analysis for multi-patch structured Kirchhoff-Love shells", Computer Methods in Applied Mechanics and Engineering, vol. 411, pp. 116060, 2023.

[67] Proserpio, D.; Kiendl, J.: "Penalty coupling of trimmed isogeometric Kirchhoff-Love shell patches", Journal of Mechanics, vol. 38, pp. 156-165, 2022.

[68] Du, X. et al.: "Nitsche's method for non-conforming multipatch coupling in hyperelastic isogeometric analysis", Computational Mechanics, vol. 65, pp. 687-710, 2020.

[69] Guarino, G. et al.: "An interior penalty coupling strategy for isogeometric non-conformal Kirchhoff-Love shell patches", Engineering with Computers, vol. 40, pp. 3031-3057, 2024.

[70] Gravesen, J. et al.: "Planar parametrization in isogeometric analysis", Mathematical Methods for Curves and Surfaces (MMCS 2012), Springer, pp. 189-212, 2014.

[71] Bathe, K.: "Finite Element Procedures", Prentice Hall, 1996.

[72] Fish, J.; Belytschko, T.: "A First Course in Finite Elements", John Wiley & Sons, 2007.

[73] de Falco, C.; Reali, A.; Vázquez, R.: "GeoPDEs: A research tool for isogeometric analysis of PDEs", Advances in Engineering Software, vol. 42, no. 12, pp. 1020-1034, 2011.

[74] Hughes, T. J. R.; Sangalli, G.; Tani, M.: "Isogeometric analysis: Mathematical and implementational aspects, with applications", Splines and PDEs: From Approximation Theory to Numerical Linear Algebra, Springer, pp. 237-315, 2018.

[75] Pan, M.; Jüttler, B.; Mantzaflaris, A.: "Efficient matrix assembly in isogeometric analysis with hierarchical B-splines", Journal of Computational and Applied Mathematics, vol.

390, pp. 113278, 2021.

[76] Teijeiro, T. et al.: "Machine learning discovery of optimal quadrature rules for isogeometric analysis", Computer Methods in Applied Mechanics and Engineering, vol. 416, pp. 116310, 2023.

[77] Shames, I. H.; Dym, C. L.: "Energy and Finite Element Methods in Structural Mechanics", Hemisphere Publishing Corporation, New York, 1985.

[78] Safari Hesari, P.; Almstedt, S.: "Isogeometric analysis of curved beams and thin shells", master's thesis, Chalmers University of Technology, 2017.

[79] Stolarski, H.; Belytschko, T.: "Membrane locking and reduced integration for curved elements", Journal of Applied Mechanics, vol. 49, no. 1, pp. 172-176, 1982.

[80] Echter, R.; Bischoff, M.: "Numerical efficiency, locking and unlocking of NURBS finite elements", Computer Methods in Applied Mechanics and Engineering, vol. 199, no. 5-8, pp. 374-382, 2010.

[81] Echter, R.: "Isogeometric analysis of shells", doctoral dissertation, Universität Stuttgart, 2013.

[82] Kiendl, J.: "Isogeometric analysis and shape optimal design of shell structures", doctoral dissertation, Technische Universität München, 2011.

[83] Chapelle, D.; Bathe, K.: "The Finite Element Analysis of Shells-Fundamentals", Springer, Berlin, Heidelberg, 2003.

[84] Kiendl, J. et al.: "Isogeometric shell analysis with Kirchhoff-Love elements", Computer Methods in Applied Mechanics and Engineering, vol. 198, no. 49-52, pp. 3902-3914, 2009.

[85] Fleisch, D. A.: "A Student's Guide to Vectors and Tensors", Cambridge University Press, Cambridge, 2011.

[86] Kiendl, J. et al.: "Isogeometric Kirchhoff-Love shell formulations for general hyperelastic materials", Computer Methods in Applied Mechanics and Engineering, vol. 291, pp. 280-303, 2015.

[87] Chen, L. et al.: "Explicit finite deformation analysis of isogeometric membranes", Computer Methods in Applied Mechanics and Engineering, vol. 277, pp. 104-130, 2014.

[88] Kefal, A.; Oterkus, E.: "Isogeometric iFEM analysis of thin shell structures", Sensors, vol. 20, no. 9, pp. 2685, 2020.

[89] Magisano, D. et al.: "Large deformation Kirchhoff-Love shell hierarchically enriched with warping: Isogeometric formulation and modeling of alternating stiff/soft layups", Computer Methods in Applied Mechanics and Engineering, vol. 418, pp. 116556, 2024.

[90] Zhang, R. et al.: "Large deformation frictional contact formulations for isogeometric Kirchhoff-Love shell", International Journal of Mechanical Sciences, vol. 249, pp. 108253, 2023.

[91] Wall, W. A.; Frenzel, M. A.; Cyron, C.: "Isogeometric structural shape optimization", Computer Methods in Applied Mechanics and Engineering, vol. 197, no. 33-40, pp. 2976-2988, 2008.

[92] Hassani, B.; Tavakkoli, S. M.; Moghadam, N. Z.: "Application of isogeometric analysis in structural shape optimization", Scientia Iranica, vol. 18, no. 4, pp. 846-852, 2011.

[93] Bandara, K.; Cirak, F.: "Isogeometric shape optimisation of shell structures using mul-

tiresolution subdivision surfaces", Computer-Aided Design, vol. 95, pp. 62-71, 2018.

[94] Zhao, H. et al.: "Automated shape and thickness optimization for non-matching isogeometric shells using free-form deformation", Engineering with Computers, vol. 40, pp. 3495-3518, 2024.

[95] Song, Z. et al.: "Multi-patch isogeometric shape optimization of complex free-form surfaces with buckling constraints", Computers and Structures, vol. 321, pp. 108093, 2026.

[96] Yang, F. et al.: "Isogeometric double-objective shape optimization of free-form surface structures with Kirchhoff-Love shell theory", Finite Elements in Analysis and Design, vol. 223, pp. 103989, 2023.

[97] Weeger, O.: "Isogeometric sizing and shape optimization of 3D beams and lattice structures at large deformations", Structural and Multidisciplinary Optimization, vol. 65, pp. 62, 2022.

[98] Meng, X. et al.: "Shape-thickness-topology coupled optimization of free-form shells", Automation in Construction, vol. 142, pp. 104476, 2022.

[99] Xia, Z. et al.: "A machine-learning framework for isogeometric topology optimization", Structural and Multidisciplinary Optimization, vol. 66, pp. 83, 2023.

[100] Hinz, J. et al.: "The role of PDE-based parameterization techniques in gradient-based IGA shape optimization applications", Computer Methods in Applied Mechanics and Engineering, vol. 378, pp. 113685, 2021.

[101] López, J.; Anitescu, C.; Rabczuk, T.: "Isogeometric structural shape optimization using automatic sensitivity analysis", Applied Mathematical Modelling, vol. 89, pp. 1004-1024, 2021.

[102] Duysinx, P.; Bendsøe, M. P.: "Topology optimization of continuum structures with local stress constraints", International Journal for Numerical Methods in Engineering, vol. 43, no. 8, pp. 1453-1478, 1998.

[103] Kennedy, G. J.; Hicken, J. E.: "Improved constraint-aggregation methods", Computer Methods in Applied Mechanics and Engineering, vol. 289, pp. 332-354, 2015.

[104] Fan, Z.; Gao, L.; Li, H.: "Isogeometric topology optimization method for design with local stress constraints", Computers and Structures, vol. 305, pp. 107564, 2024.

[105] Zhai, X.: "Alternating optimization method for isogeometric topology optimization with stress constraints", Journal of Computational Mathematics, vol. 42, no. 1, pp. 134-155, 2024.

[106] Huang, Y.; Chen, X.; Yao, S.: "Strength-constrained topology optimization considering static stress and fatigue using isogeometric analysis", Structural and Multidisciplinary Optimization, vol. 68, pp. 109, 2025.

[107] Hübner Scherer, F. et al.: "Stress-based topological shape optimization for thick shells using the level set method and trimmed non-conforming multi-patch isogeometric analysis", Structural and Multidisciplinary Optimization, vol. 67, pp. 26, 2024.

[108] Eiben, A. E.; Smith, J. E.: "Introduction to Evolutionary Computing", Springer, Berlin, Heidelberg, 2015.

[109] Haykin, S.: "Neural Networks and Learning Machines", Pearson, 2009.

[110] Nielsen, M. A.: "Neural Networks and Deep Learning", Determination Press, 2015.

[111] Lee, S. et al.: "Background information of deep learning for structural engineering",

Archives of Computational Methods in Engineering, vol. 25, no. 1, pp. 121-129, 2018.

[112] McKay, M. D.; Beckman, R. J.; Conover, W. J.: "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code", Technometrics, vol. 21, no. 2, pp. 239-245, 1979.

LIST OF SYMBOLS AND ABBREVIATIONS

Latin symbols

a, b	working indices of the Leibniz rule for rational derivatives
$a_{\alpha\beta}$	covariant components of the first fundamental form of a surface
$a^{\alpha\beta}$	contravariant components of the first fundamental form
A, I	cross-section area and second moment of area of a beam
$\boldsymbol{A}$, $\boldsymbol{B}$	collocation matrices of the two parametric directions of a fit, and $\boldsymbol{A}_u$, $\boldsymbol{A}_{uu}$, $\boldsymbol{B}_v$, $\boldsymbol{B}_{vv}$ the same matrices built from the derivatives of the basis functions
A_t	area of a triangle
$b_{\alpha\beta}$	covariant components of the second fundamental form
$B_{i,p}$	Bernstein polynomial of degree p
$B_{ij,pq}$	bivariate Bernstein basis function of bi-degree (p, q)
$\mathbb{C}$	elasticity tensor
$\boldsymbol{b}$	body force
$\boldsymbol{B}$	strain-displacement matrix
$\boldsymbol{C}(u)$, $\boldsymbol{C}(\xi)$	parametric curve, and the axis of a beam
$\boldsymbol{d}$, $\boldsymbol{d}_i$	vector of control variables of the discrete field and its entries
$\boldsymbol{D}$	matrix of stiffnesses of a structural model
E	Young's modulus, the sum of squared residuals of a fit, and the signed fitting error of a reparametrization
E_D	Dirichlet energy of a map
E_{smooth}	fairing energy added to the error of a fit
f, q, $\boldsymbol{f}$	distributed tangential and transverse loads on a beam, and the surface load of a shell
F	tip force of the cantilever examples
$\boldsymbol{F}$	load vector
F_k, Q_k	concentrated forces
$\boldsymbol{g}_\alpha$	covariant basis vectors of a surface
h	knot span size, and the depth threshold of the h-minima transform
i, j	indices of control points
J	Jacobian of the geometric map of a beam axis
$\boldsymbol{J}$	Jacobian matrix of the geometric map, and of the residuals of a fit
J_t	Jacobian of the affine map on a triangle
k, l	indices of the rows and the columns of a control net and of a grid of data, k also the multiplicity of a knot and the index of the span receiving an inserted knot
$\boldsymbol{K}$	stiffness matrix
L	graph Laplacian matrix of a triangulation, and the length of a beam
m	index of the last knot of a knot vector
$\mathcal{M}$	triangulated surface mesh
$M_{j,q}$	B-spline basis functions of the second parametric direction
n	index of the last control point
n_b	number of distinct values in a knot vector
n_u, n_v	numbers of data points of a grid in the two parametric directions

$\boldsymbol{n}$	unit normal vector of a surface
$N_{i,p}$	B-spline basis function of degree p
$N_{ij,pq}$	bivariate B-spline basis function of bi-degree (p, q)
$\boldsymbol{N}$, $\boldsymbol{R}$	rows of basis functions of a discretization
p, q	polynomial degrees in the two parametric directions
$\boldsymbol{P}_i$, $\boldsymbol{P}_{ij}$	control points and control net
$\boldsymbol{P}^w_{ij}$	control point in homogeneous coordinates
$\boldsymbol{Q}_i$	control points after knot insertion or degree elevation
r, s	orders of differentiation in u and in v, and s also the arc length along a beam axis
R	radius of the circular arc of the curved cantilever example
$R_{i,p}$, $R_{ij,pq}$	rational basis functions
$\boldsymbol{r}(\xi^1, \xi^2)$	mapping of the parametric domain onto a surface
S	refinement matrix of knot insertion
$\boldsymbol{S}(u, v)$	parametric surface
t	shell thickness
$\boldsymbol{t}$	traction on the loaded part of the boundary
u, v	parametric coordinates, and u, w the tangential and normal displacements of a beam axis
u_i	i-th knot of a knot vector
$\bar{u}$	inserted knot
$\boldsymbol{u}^h$	discrete displacement field
U, V	knot vectors of the two parametric directions, and the strain energy and the potential of the loads in Chapter 4
W	weight function of a rational form
w_i, w_{ij}	weights associated with control points
$\boldsymbol{x}_i$, $\boldsymbol{x}_{kl}$	vertex of a triangulated surface, and a point of a grid of data
$\boldsymbol{X}$	array of the data points of a fit

Greek symbols

α, β	indices of the curvilinear surface coordinates
α_i	coefficients of the knot insertion formula
Γ	loaded part of the boundary
δ	variation, prefix of a virtual quantity
Δu, Δv	spacing of a regular parametric grid
ε	scalar indicator field driving adaptive reparametrization, and the membrane strain of a beam
$\varepsilon_{\alpha\beta}$	membrane strain of a shell
κ	discrete curvature indicator, and the change of curvature of a beam axis
κ_0	initial curvature of a beam axis
$\kappa_{\alpha\beta}$	change of curvature of a shell
λ_{ij}	weights of a convex combination map
ν	Poisson's ratio
ρ	radius of curvature of a beam axis
ξ, ξ^1, ξ^2	parametric coordinates in the analysis chapters, ξ on a curve and ξ^1, ξ^2 on a surface

$\tilde{\xi}, \tilde{\xi}^1, \tilde{\xi}^2$	coordinates of the parent element
Π	total potential energy
$\boldsymbol{\sigma}$	stress
σ_1, σ_2	singular values of the Jacobian of a parametrization
φ	parametrization map of a triangulated surface
Ω	physical domain
$\hat{\Omega}$	parameter space
$\tilde{\Omega}$	parent element

Abbreviations

CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing
CST	Constant Strain Triangle
DKT	Discrete Kirchhoff Triangle
DOF	Degree(s) of Freedom
FEM	Finite Element Method
GA	Genetic Algorithm
HB	Hierarchical B-Spline
IGA	Isogeometric Analysis
KL	Kirchhoff-Love (shell theory)
KS	Kreisselmeier-Steinhauser (aggregation function)
LHS	Latin Hypercube Sampling
LR	Locally Refined (B-splines)
LSCM	Least-Squares Conformal Maps
LSQ	Least Squares
ML	Machine Learning
NN	Neural Network
NURBS	Non-Uniform Rational B-Spline
PC	Point Cloud
RMSE	Root Mean Square Error
4-RoSy	4-fold Rotational Symmetry (field)
STL	Stereolithography (triangulated mesh format)
THB	Truncated Hierarchical B-Spline
UV	Parametric coordinates (u, v)

ABSTRACT

A scanned engineering component arrives as a dense triangulated mesh, which records the shape but carries none of the smooth structured description that computer-aided design, higher-order numerical analysis and parametric shape optimization are built on. This qualification exam reviews the literature of the conversion from the one to the other and of what the resulting model is then used for, organized into five connected research areas presented in the order of their mutual dependence. The parametric models are summarized first, from Bernstein and Bézier representations through B-spline basis functions and NURBS to the refinement algorithms of knot insertion, degree elevation and knot removal. The recovery of such a model from measured data follows, covering the parametrization of a triangulated surface by convex combination and discrete harmonic maps, the distortion those maps introduce and what it costs the model, the gridded data the fit consumes, regularized least-squares approximation, and the routes by which a fit is improved once the limits of a single uniform patch are reached. Isogeometric analysis is then introduced as the bridge that integrates analysis with design by using the parametric surface representations established before it, and its formulation is developed from the foundations, from its relationship to the finite element method and the domains and mappings on which the discretization is defined, through the Galerkin formulation, to one-dimensional structural members, the curvilinear geometry of surfaces, Kirchhoff-Love shells and locking. Shape optimization on the shared model closes the review, with its choice of design variables, adjoint sensitivities and the multiresolution, evolutionary and surrogate strategies that surround them. The concluding chapter sets out three directions in which the reviewed areas can be joined, the coupling of reparametrization with shape optimization, multi-patch fitting together with multi-patch analysis, and optimization against several criteria and several kinds of physics on a single model.

PROŠIRENI SAŽETAK

Geometrijsko modeliranje bavi se time kako se oblici prikazuju, grade i mijenjaju u računalu, a izraslo je iz potreba automobilske, zrakoplovne i brodograđevne industrije. Kada model treba dobiti iz predmeta koji već postoji, predmet se skenira, a skener isporučuje gustu trianguliranu mrežu koja vjerno bilježi oblik, ali ne nosi strukturu na kojoj počivaju računalno potpomognuto projektiranje, numerička analiza višeg reda i parametarska optimizacija oblika. Ovaj kvalifikacijski ispit daje pregled literature o pretvorbi takvog skena u parametarski model plohe i o tome čemu dobiveni model dalje služi. Pregled je organiziran u pet povezanih područja, parametarske modele, parametrizaciju i fitanje trianguliranih ploha, izogeometrijsku analizu, optimizaciju oblika te zaključak sa smjerovima daljnjeg istraživanja. Kroz sva poglavlja provlači se isti objekt, spline ploha, koja je zapisana jezikom drugog poglavlja, izgrađena u trećem iz mjerenja koje nema njezinu strukturu, u četvrtom postaje diskretizacija analize, a u petom njezine kontrolne točke postaju projektne varijable.

Drugo poglavlje sažima teoriju parametarskih krivulja i ploha. Polazi od Bernsteinovih polinoma i Bézierovih krivulja, na kojima se uvode kontrolni poligon, svojstva konveksne ljuske i particije jedinice te tenzorski produkt kojim se od dviju jednodimenzijskih baza gradi baza plohe. Slijede B-spline bazne funkcije definirane Cox-de Boorovom rekurzijom na vektoru čvorova, njihov lokalni nosač, kontinuitet određen višestrukošću čvorova, derivacije i podjela parametarske domene na ćelije, od kojih je svaka određena malim blokom kontrolnih točaka. Na toj osnovi uvode se B-spline krivulje i plohe te NURBS, s težinama koje omogućuju egzaktan prikaz kružnica i ploha rotacije i derivacijama racionalne baze. Poglavlje zatim izlaže algoritme profinjavanja, umetanje čvorova, podizanje stupnja polinoma te uklanjanje čvorova, a zatvara ga pregled spline modela izvan tenzorskog produkta, hijerarhijskih, T-spline i LR konstrukcija, koji dopuštaju lokalno profinjavanje.

Treće poglavlje bavi se obrnutim inženjerstvom, dakle izgradnjom takvog modela iz skenirane triangulirane plohe, u dva koraka koji imaju smisla samo zajedno. U prvom se svakom vrhu mreže dodjeljuje par parametarskih koordinata u ravnoj domeni, a u drugom se na tako parametrizirane podatke prilagođava spline ploha. Parametrizacija se izlaže kroz problem preslikavanja zakrivljene plohe u ravninu, koje ne može istodobno očuvati duljine, kutove i površine, pa svako preslikavanje unosi distorziju. Pregled se usredotočuje na konveksne kombinacijske mape s fiksnim rubom, od Tutteove mape s uniformnim težinama, koja jamči injektivnost, preko Floaterovih težina koje čuvaju oblik, do diskretne harmonijske mape, koja kutove čuva najbolje, ali može stvoriti preklope. Parametrizacija ujedno određuje gdje će kontrolne točke plohe sjesti i kolika je razlučivost fita. Fitanje polazi od topološkog oblaka točaka uzorkovanog na pravilnoj mreži parametarske domene, a kontrolne točke dobivaju se linearnom aproksimacijom najmanjim kvadratima u matričnom obliku, dopunjenom regularizacijom koja prigušuje oscilacije u područjima s malo podataka. Na trupu brodice, propeleru i kućištu spojke srednja pogreška pada s brojem kontrolnih točaka, dok se najveća pogreška zaustavlja na razini koju profinjavanje jedne uniformne zakrpe ne prelazi, skupljena uz značajke predmeta. Poglavlje zatvara pregled nelinearnih putova kojima se fit dalje poboljšava, od nelinearne optimizacije fita i lokalno profinjivih spline prostora do reparametrizacije, optimizacije vektora čvorova i dekompozicije predmeta na više zakrpa.

Četvrto poglavlje uvodi izogeometrijsku analizu, koju su Hughes, Cottrell i Bazilevs predložili kako bi spojili geometriju i analizu. Iste bazne funkcije koje egzaktno opisuju geometriju koriste se i kao funkcije aproksimacije rješenja, pa geometrijski i računski model postaju je-

dan objekt. Poglavlje najprije uspoređuje obje metode po konceptualnim razlikama, a zatim iz principa virtualnog rada izvodi slabu formulaciju kao zajedničko polazište. Jednodimenzijski problemi obrađuju se u cijelosti, od ravne konzole, na kojoj visoka glatkoća baze daje egzaktan progib već s jednim elementom, do zakrivljenog nosača u ravnini, za koji se uzdužna i savojna deformacija izražavaju preko parametarske koordinate i geometrijskog preslikavanja. Na istom se primjeru prikazuje membranski locking, koji se s podizanjem stupnja polinoma smanjuje, a profinjavanje diskretizacije povezuje se s algoritmima drugog poglavlja i s k-profinjavanjem. Dvodimenzijski problemi započinju domenama i preslikavanjima i Jakobijanom čija kvaliteta odlučuje može li se rekonstruirani model uopće analizirati. Slijedi krivocrtna geometrija ploha, kovarijantna baza, prva i druga fundamentalna forma i njihove promjene kao mjere deformacije, na čemu se gradi Kirchhoff-Loveova ljuska u Kiendlovoj formulaciji, bez rotacijskih stupnjeva slobode i s rubnim uvjetima koji se nameću preko položaja kontrolnih točaka.

Peto poglavlje razmatra optimizaciju oblika na tako dijeljenom modelu, gdje se ploha i analiza spajaju u jednu petlju. Kontrolne točke koje opisuju plohu ujedno su nepoznanice analize i projektne varijable, pa se veza između geometrijskog i računskog modela ne mora ponovno graditi u svakoj iteraciji. Pregled obrađuje izbor projektnih varijabli, gdje se kontrolne točke pomiču duž normale plohe, adjungiranu metodu kojom se gradijent podatljivosti dobiva uz cijenu jedne dodatne analize, te teškoću s naprezanjem kao ciljem, jer najveće naprezanje nije derivabilna funkcija projekta. Zatim se izlažu višerezolucijski pristup, u kojem se oblik najprije mijenja na gruboj razini kontrolnih točaka, evolucijske metode bez gradijenata i surogatni modeli u kojima neuronska mreža uči predviđati ishod analize. Poglavlje bilježi i novije radove o izogeometrijskoj optimizaciji po naprezanju i topološkoj optimizaciji.

Zaključno poglavlje sažima kako se pregledana područja podupiru nesimetrično. Literatura o obradi geometrije jamči injektivnost parametrizacije, ali njezinu kvalitetu mjeri kutovima i površinama, koji su s potrebama analize tek posredno povezani. Literatura o izogeometrijskoj analizi točno određuje kakvu parametrizaciju treba, ali geometriju uzima kao unaprijed zadanu. Obrnuto inženjerstvo povezuje jedno s drugim, ali vlastite rezultate ocjenjuje samo geometrijskom pogreškom, pa razlika koja se pokazuje tek u analizi ostaje nevidljiva. Iz toga slijede tri smjera predloženog istraživanja. Prvi je spajanje reparametrizacije i optimizacije oblika u jedan problem, u kojem se oblik i preslikavanje mijenjaju zajedno, uz ograničenje vjernosti fita i injektivnost ugrađenu u konstrukciju. Drugi je multi-patch lanac od početka do kraja, fitanje predmeta kao više zakrpa duž njegovih značajki i analiza sastavljenog modela sa spojevima koji vraćaju kontinuitet preko šavova. Treći je optimizacija prema više kriterija i više vrsta fizike na jednom modelu, primjerice propelera koji se istodobno optimira strukturno i prema strujanju. Sva tri smjera postaju mogući samo zato što jedan spline model istodobno nosi geometriju, analizu i projekt.