Chapter 2
Introduction to Geometric Design

Abstract This chapter provides an introduction to geometric design. It introduces various popular mathematical methods used for shape representation in geometric design. It also discusses the role of interactive design and parametric design to enhance the processes involved in a geometric design problem. Furthermore, this chapter discusses the use of design optimization to carry out automatic design for function.

2.1 Introduction

Geometric design concerns with the mathematical description and analysis of shape. Geometric design draws upon the fields such as algebra, geometry, numerical analysis and computer programming. Let us consider the process involved in the design of a new engineering product. Often such a process starts with a definition of a template shape where the requirements in terms of the product's geometric shape and its functionality are specified. This process then proceeds through a sequence of iterative activities to seek an optimal design. Today, this process of 'automatic design for function' relies on the increased use of computers. Although geometric design based on the extensive use of computers does not automatically provide the solution to a given design problem, it can increase the efficiency of the design process. Thus, the main processes of geometric design involve the efficient description of the geometric shape and the integration of the shape with functional analysis. For this purpose, for geometric design, a mathematical method which can generate complex geometries and can relate to the functionality of the object at an early stage of the design process is desirable.

Over the past 40 years, the use of geometric design methods has grown explosively. Today, virtually all computer-based design tasks commence with the use of Computer Aided Design (CAD) systems to create detailed geometric models. These models serve as the point of departure for diverse analysis tools, such as computational fluid dynamics (CFD), stress analysis, geophysical data exploration, and computational electromagnetics or acoustics. Due to the increase in the power of computer hardware, industries such as those related to aerospace, automotive and electronics make more and more integrated use of CAD and analysis. This provides a 'virtual laboratory' for assessing performance characteristics (such as structural

H. Ugail, *Partial Differential Equations for Geometric Design*,
DOI 10.1007/978-0-85729-784-6_2,

strength or aerodynamic drag) that otherwise would require expensive and time-consuming physical experimentation.

As mentioned above as part of the process of geometric design, the functional properties of the object being created are analyzed by solving the field equations governing the physical process(es) under consideration. One major difficulty encountered here is the linking of complicated surface geometry to analysis [1, 2].

With the assimilation of CAD systems and analysis tools in the major industrial processes, an integrated approach is certainly desirable. However, the need for a systematic way of considering the relationship between geometry and the functional aspects of the geometric model becomes paramount [3].

A mathematical method which can generate complex geometries and can relate to the functionality of the object at an early stage of the design process is desirable. Furthermore, it will be an added advantage if such a method can create a parameterized representation of the object as the variation of the parameters that will then enable the creation of alternative descriptions of the geometry in question while maintaining the functional relations. Such alternative models are necessary for design optimization where the best available design candidate is chosen out of a possible range of designs.

2.2 Mathematical Methods for Shape Representation in Geometric Design

In geometric design, it is common practice to represent geometry of complex shapes in terms of polynomial functions of two parameters. The nature of the surface obtained using such polynomial-based methods usually depends on the type of polynomial chosen. Examples of such surfaces are Bézier surfaces [4], B-splines [5], rational B-splines [6] and non-uniform rational B-splines (NURBS) [7, 8].

A typical bicubic patch in its parametric form can be described as

$$\mathbf{p}(u, v) = \sum_{i=0}^{3} \sum_{j=0}^{3} \mathbf{a}_{ij} u^i v^j, \quad u, v \in [0, 1], \tag{2.1}$$

where $\mathbf{p}$ is a vector of the Cartesian coordinates of points on the surface, u and v are parametric coordinates, and the $\mathbf{a}_{ij}$ are vector coefficients that determine the shape of the surface patch.

The bicubic patch was first introduced in 1963 by Ferguson [9], where the coefficients $\mathbf{a}_{ij}$ in Eq. (2.1) can be expressed in terms of the vectors $\mathbf{p}$, $\mathbf{p}_u$, $\mathbf{p}_v$ and $\mathbf{p}_{uv}$ at the four corner points of the surface patch. The terms $\mathbf{p}_u$ and $\mathbf{p}_v$ are taken to be tangents to the surface in each parametric direction and $\mathbf{p}_{uv}$ is termed the twist vector. The effect of the twist vectors is not intuitively obvious and in his original work Ferguson set them to zero. Ferguson patches are thus expressed in terms of positional and derivative information at the patch corners and can be considered to be obtained from Hermite polynomial interpolation between the corner points. A Ferguson patch can be interpreted as a specific form of the more general Coons patch [10]. The main difference between a Coons patch and a Ferguson patch is that the former is obtained by interpolation between the boundaries of arbitrary form while the latter can be obtained by using parametric cubic boundary curves.

Another common type of surface patch is the so-called Bézier patch,

$$\mathbf{p}(u, v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \mathbf{p}_{ij} B_{i,m}(u) B_{j,n}(v), \quad u, v \in [0, 1], \tag{2.2}$$

where the $\mathbf{p}_{ij}$ are the Cartesian coordinates of the vertices or the 'control points' which form a characteristic polyhedron with an $(m+1) \times (n+1)$ rectangular array of points. The $B_{i,m}(u)$ and $B_{j,n}(v)$ are known as Bernstein basis functions and are defined by

$$B_{i,m}(u) = \frac{m!}{i!(m-i)!} u^i (1-u)^{m-i}, \tag{2.3}$$

and similarly for $B_{j,n}(v)$. A Bézier surface approximates the characteristic polyhedron, and interactive surface design is achieved by moving the control points. The bicubic Bézier patch, for which $m, n = 3$, is essentially a reformulation of the Ferguson patch [9].

An alternative to the Bézier patch is the B-spline surface, which is also defined in terms of the characteristic polyhedron [11]. B-spline surface patches permit the use of more control points in the characteristic polyhedron whilst retaining low order basis functions. They are obtained by replacing the Bernstein basis functions $B_{i,m}(u)$ and $B_{j,n}(v)$ in Eq. (2.2) by the B-spline basis functions $N_{i,k}(u)$ and $N_{j,l}(v)$. The B-spline basis functions are defined recursively by the following formulae:

$$N_{i,1}(u) = \begin{cases} 1 & \text{if } t_i \le u, v \le t_{i+1}, \\ 0 & \text{otherwise}, \end{cases} \tag{2.4}$$

$$N_{i,k}(u) = \frac{(u - t_i)}{(t_{i+k-1} + t_i)} N_{i,k-1}(u) + \frac{(t_{i+k} - u)}{(t_{i+k} - t_{i+1})} N_{i+1,k-1}(u), \tag{2.5}$$

and similarly $N_{j,l}(v)$. The parameters k and l control the degrees $(k-1)$ and $(l-1)$ of the resulting polynomials in u and v, and thus also control the continuity of these curves. The t_i and t_j are called knot values and they relate the parametric variables u and v to the $\mathbf{p}_{ij}$ control points. The functions $N_{i,1}(u)$ and $N_{j,1}(v)$ switch between the values 1 and 0 depending on the values of u and v. These B-spline basis functions are non-zero only over a given finite interval and enable the effect of a control point on the surface shape to be localized. Another advantage of the B-spline formulation is its ability to preserve arbitrarily high degrees of continuity over the complex surface patch. These characteristics make the B-spline surfaces popular for use in an interactive modeling environment.

The B-spline formulation was extended to non-uniform rational B-splines (NURBS) by Versprille [12]. The term rational refers to the ratio of the polynomials that characterizes this approach, i.e. a NURBS surface is the ratio of the two B-spline functions [13, 14].

A NURBS surface is defined as

$$S(u, v) = \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} \underline{p}_{i,j} w_{i,j} B_{i,k}(u) B_{j,l}(v)}{\sum_{i=0}^{m} \sum_{j=0}^{n} w_{i,j} B_{i,k}(u) B_{j,l}(v)}. \tag{2.6}$$

The surface has $(m+1) \times (n+1)$ control points $\underline{p}_{i,j}$ and weights $w_{i,j}$. Assuming the degrees of basis functions along u and v axes to be $k-1$ and $l-1$ respectively, the number of knots is $(m+k+1) \times (m+l+1)$. The non-decreasing knot sequence is $t_0 \le t_1 \le \cdots \le t_{m+k}$ along the u direction and $s_0 \le s_1 \le \cdots \le s_{n+l}$ along the v direction with the parameter domain in the range: $t_{k-1} \le u \le t_{m+1}$ and $s_{l-1} \le v \le s_{n+1}$. If the knots have multiplicity k and l in the u and v directions, respectively, the surface patch will interpolate the four corners of the boundary control points.

Like the rational B-splines NURBS are infinitely smooth in the interior of the knot span provided the denominator is not zero; and at a given knot NURBS are at least C^{k-1-r} continuous with knot multiplicity r, which enable them to satisfy different smoothness requirements. NURBS also share properties such as the 'convex hull' property, 'local support' and invariance under standard geometric transformations [14]. Additionally, the weights $w_{i,j}$ act as extra degrees of freedom influencing the local shape, i.e. if a particular weight is set to zero, then the corresponding rational basis function is also zero, and its control point does not effect the NURBS shape. The spline is attracted towards a control point more if the corresponding weight is increased and less if the weight is decreased. Moreover, NURBS also form a common mathematical framework for both implicit and parametric forms, i.e. in principle they can represent analytic functions such as conics and quadratics as well as free-form shapes.

The spline based definition for curves and surfaces forms the basis for many of today's geometric design systems. However, to create a given object, the chosen geometric design system may use a variety of analytic descriptions for curves and surfaces, or the system may use a combination of analytic forms and spline based functions to perform operations, such as union, difference and intersection [15]. Furthermore, some geometric design systems use variational modeling schemes in which the basic spline functions are manipulated using physically based relations, such as force and energy [16].

2.2.1 Schemes for Geometry Model Representation

Various geometry representation techniques have been developed to represent two-dimensional or three-dimensional geometric shapes. Popular representation techniques include: Boundary Representation (B-Rep), Constructive Solid Geometry (CSG), feature based representations and variational geometry.

B-Rep Approach In a B-Rep approach, a shape is represented by the boundary information such as faces, edges and vertices, i.e. B-Rep represents geometry in terms of boundaries and topological relations.

CSG Approach The CSG approach models geometric shapes using a set of 'primitives' such as cubes, cylinders or prisms. Complex shapes are built from the primitives through a set of Boolean operations (e.g. union, difference and intersection). Most CSG systems in use today offer quite a variety of primitive solids, ranging from various types of spheres and ellipses, boxes and cones, and solids defined by swept or extruded curves. The CSG modeling approach has several inherent limitations of which the most notable limitation is the non-uniqueness of a CSG representation. This non-uniqueness of representations makes recognition of shapes from their CSG representation extremely difficult.

Feature Based Approach In the feature based representation, a part is built from a set of feature 'primitives'. Examples of features include holes, slots and ribs. A feature based design approach allows a user to use features stored in a feature library. It provides a means for building a complete CAD database with the features right from the start of the design. However, this approach suffers from the difficulty of there being a limited number of available feature primitives. It is difficult to satisfy various design needs, and in the event that the features interact with one another, new features may arise that can cause complication with the analysis process. Feature based design allows a designer to bridge the gap between units of the designer's perception of forms and data in geometric models. In this scheme of representation, shapes are described in the way the designer understands them [17].

Variational Approach The concept of using variational geometry in geometric design started as early as 1981. Instead of defining a geometric model with respect to a set of characteristic points in $\mathbf{R}^3$, dimensions are treated as constraints limiting the permissible locations of these points. Many schemes for variational design have been suggested, e.g. [18–21].

Many of these schemes use a physical analogy in which a chosen functional is used to minimize the elastic energy satisfying certain interpolation constraints imposed on the mechanism by which the surface is created. The method of Partial Differential Equations [22–24] which is discussed in this book falls into this category.

It is important to note that the modeling scheme we choose forms an integral part of the geometric design process. To be useful within a given application area, the range of shapes that can be represented by a given scheme should be adequate. Moreover, the scheme should be user friendly, i.e. the model representation scheme should be well suited for 'interactive design'.

2.3 Enhancing Geometric Design Using Interactive and Parametric Design

In the early days of geometric design, design applications were carried out in 'batch', i.e. a complete task (or job) was first defined by the user and then submitted to the computer. The computer processed the complete job without further interaction from the user and then produced an output.

Most of the existing geometric design systems, if not all, make heavy use of interactive graphics techniques rather than batch techniques. Thus, the user can interact with the computer via input devices such as the mouse and keyboard.

2.3.1 Techniques for Interactive Design

As discussed above, in geometric design, it is common practice to describe the geometric models by means of spline based methods. There exist many techniques for interactive design using such methods. Perhaps the most basic case is the use of Bézier patches in which the displacement of a control point results in the change in the shape. This technique has also been applied to B-splines to control the shape of the surface patch locally. Applying the above technique to an isolated control point frequently leads to results with an unpredictable effect on the resulting shape of the surface. Designers usually face the questions of choosing which control points to move in which direction [25]. In principle, it is possible to produce large-scale changes to the shape of the surface by moving more than one control point. However, such interactive manipulations often result in undesired bumps or wiggles within the surface patch.

As far as interactive design methods using NURBS are concerned, an initial surface is created via specification of a control polygon. The initial shape is then refined into the final desired shape through interactive adjustments of control points and weights and possibly addition and deletion of knots. The knot insertion algorithm [26], the control point insertion algorithm [27] are all complementary elements for interactive shape refinements. However, such refinement processes are often considered to be tedious and very unpredictable [28]. For example, to adjust the shape of a surface should a designer move a control point, or change a weight?

Despite the recent advent of sophisticated devices for 3D interaction, the above mentioned techniques for interactive surface design and manipulation can be difficult for a designer to use effectively. To overcome this problem, techniques which allow ‘physically based’ manipulation have been introduced. Many authors have suggested the use of ‘constraint based interface’, where some of the design parameters have some form of physical relevance.

For example, Terzopoulos and Watkin describe simple interactive sculpting using viscoelastic and plastic models [29]. Celniker and Gossard [30] describe an interesting prototype system for interactive free-form design based on the finite-element optimization of energy functionals. Thingvold and Cohen [31] proposed a deformable B-spline whose control points are mass points connected by elastic springs and hinges. Celniker and Welch [30] investigated deformable B-splines with linear constraints. Furthermore, for design using NURBS, free-form deformable models were introduced by Terzopoulos et al. [32]. Such models were further developed by Pentland and Williams [33], Platt and Barr [34]. A similar technique for real time design using deformations is discussed by Borrel and Rappoport [35].

Terzopoulos and Qin, on the other hand, describe a model for interactive design in which they use a generalized form of NURBS called Dynamic NURBS or

D-NURBS. The D-NURBS model is governed by dynamic differential equations which, when integrated numerically through time, continuously evolve the control points and weights in response to applied forces [36].

Unlike models based on the direct manipulation of surfaces, the behavior of deformable models are governed by 'physical' laws. The result is that such models respond to the user interactions in a natural and somewhat predictable way. Many existing geometric design systems use these techniques. However, as far as surface manipulations in such systems are concerned, the initial surface is often provided as a pre-defined geometry model obtained from scan-data, for example, on which only small scale manipulations are allowed to be carried out.

An important point to note in the existing mathematical models which allow interactive manipulations of surfaces is the large number of design parameters often involved. In the development of effective mathematical models, for the purpose of interactive design, much effort has been put into trying to reduce the number of design parameters associated with the chosen model. Moreover, much effort has been concentrated towards choosing design parameters with a readily apparent physical meaning. Thus, the use of 'parametric design' has recently been very popular.

2.3.2 Parametric Design

One of the requirements for geometric design systems is the ability to parameterize the shape of objects. In parametric design, the basic approach is to develop a generic description of an object or class of objects, in which the shape is controlled by the values of a set of design variables or parameters. A new design, created for a particular application, is obtained from this generic template by selecting particular values for the design parameters so that the item has properties suited to that application.

The design of a wide range of manufactured products conforms to this general pattern, ranging from engine components to such objects as aircrafts. If a product's geometry is composed of standard geometric 'constructs' such as circles, ellipses, cylinders, etc., then the parameterizations of its shape is relatively straightforward. However, for most products at least some parts of their shape are composed of free-form surfaces which, although they may constitute a small fraction of the total surface area, can, nevertheless, be very important functionally.

Many geometric design systems can handle the parameterizations of 'standard' shapes, though for objects with complicated shapes, commercial systems often fail, owing to the inability of the geometry modeling package to parameterize such shapes. Particular problem areas include the generation of surfaces which do not conform to standard, limited descriptions.

The inherent problem with the mathematical models which are used to describe the geometry of a given model is the nature of their complexity. This is particularly problematic when design has to be carried out from 'scratch' in an interactive environment. Thus, a mathematical model which can model and parameterize the geometry in terms of small set of shape parameters and, also, enable a quick interaction with the geometry is desirable.

2.4 Use of Optimization Techniques in Geometric Design

A typical demand in a practical geometric design task may be to minimize or maximize an objective function without violating a set of constraints. In order to improve a design by applying methods of computational optimization, it is necessary to express the design objective and constraints of the optimization problem by an appropriate mathematical formulation. A general formulation of the optimization problem can be written as

$$\min\left\{ f(\mathbf{x}) \;\middle|\; \mathbf{x}_l \le \mathbf{x} \le \mathbf{x}_u;\ \mathbf{g}(\mathbf{x}) = \mathbf{0};\ \mathbf{h}(\mathbf{x}) \le \mathbf{0} \right\}, \quad \mathbf{x} \in \Re^n, \tag{2.7}$$

with

f	the objective function;
$\mathbf{x}$	vector of n design variables;
$\mathbf{g}$	vector of p equality constraints;
$\mathbf{h}$	vector of q inequality constraints;
$\mathbf{x}_l$ and $\mathbf{x}_u$	lower and upper bounds for the design variables.

The design variables and the constraints form the feasible design space

$$\mathbf{x} \in R^n \mid \mathbf{x}_l \le \mathbf{x} \le \mathbf{x}_u;\ \mathbf{g}(\mathbf{x}) = 0;\ \mathbf{h}(\mathbf{x}) \le 0, \tag{2.8}$$

which describes the design space.

Coming up with appropriate formulations of the design objective and constraints of the optimization problem in a geometric design application is not always a trivial task. For example, due to the complex nature of most engineering problems, the choice of the right objective function requires experience and the fundamental understanding of the design objectives. Furthermore, not all constraints can be easily formulated in a mathematically correct way for optimization [37].

There exist a wide variety of methods for numerical optimization. The choice of a particular method is problem specific and involves considerations such as the computational cost of evaluating the function to be optimized and also the behavior of the function within the design space. Generally, these methods can be divided into two categories, i.e. those that only require the evaluation of the objective function and those that require the evaluation of the objective function and its derivatives with respect to the design parameters. During the process of optimization, most of the computational effort is spent on evaluating the objective function rather than in the optimization routine itself. Therefore, it is desirable to use a design method which minimizes the number of design variables and therefore requires as few function evaluations as possible.

Generally, the optimization process requires a search to be made in the parameter space in order to find the minimum value of the objective function. (Note that, without loss of generality, we can consider minimization problems, since maximizing a function f is equivalent to minimizing $-f$.) Particular algorithms which are used for minimization in numerical analysis include the downhill simplex algorithm due to Nelder and Mead [38] and Powell's direction set algorithm [39].

Nearly all these gradient-based methods have the common feature that they perform a series of local minimizations in which the objective function f is minimized along a straight line in the parameter space. These methods are iterative and at each successive iteration they give a vector $\mathbf{x}^k = ({x_1}^k, {x_2}^k, \ldots, x_n^k)$ of the n independent design variables which is computed from the previous iterations using the expression

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha^k \mathbf{s}^k. \tag{2.9}$$

Here $\mathbf{s}^k$ is a direction of search and α^k is a scalar that minimizes the one-dimensional function $F(\alpha) \equiv f(\mathbf{x}^k + \alpha^k \mathbf{s}^k)$. Thus, given a starting point, the algorithm moves in a series of steps through points in the parameter space, giving a lower value of the objective function than previously, until it finds a lowest possible local value of the objective function. An important point to note regarding this type of methods is that they find local minima. Thus, if a global minimum is required, multiple searches by such methods have to be performed with different starting points.

Very often it is the case that the design space considered contains many local minima, and it becomes extremely difficult to search for a global minimum using local minimization methods. An alternative method, which is considered in this work, is a global optimization method which uses a stochastic process known as Simulated Annealing [40, 41]. This method probabilistically searches in every region of the design space and therefore converges to a global minimum although not necessarily in a finite time.

As mentioned before, the particular algorithm used for numerical optimization must take into account the computing time needed to evaluate the objective function. Each function evaluation must be performed and thus may be very costly in terms of computing time.

Various approaches have been taken to perform the actual optimization. A typical approach [42] is to consider the optimization in terms of successive linear programming problems. The constraints and the objective function are linearized about the current design variable values and this simplified problem is solved. The result is taken as the new design variable values and the process is repeated until no further improvements can be made. This method has the advantage of making use of the efficient linear programming algorithms that are available. Alternatively, the search algorithm can be based on the design sensitivities to solve the full non-linear problem, again in an iterative manner. In most design optimization, it is common that the design variables are effectively taken as the Cartesian coordinates of points that boundary curves of a particular form were required to pass through. However, a different approach to this was taken by Kristensen and Madsen [44] who described the boundary as a weighted sum of certain specified functions, the weights being taken as the design variables.

It should be noted that the most important aspect of shape optimization is the choice of the design variables to be used and how the boundary shape is parameterized in terms of these design variables. Choosing too many variables will considerably complicate the design problem with severe implications on the computational time required, and having too few variables may result in only trivial solutions being obtained [43, 45]. It is therefore a basic requirement that a wide range of boundary shapes (which can be defined by a relatively small number of parameters) are accessible to the method of optimization used.

2.5 Summary

In this chapter, we have given an introduction to the geometric design. The discussions have been centered around some of the popular methods for geometry representation such as splines. The key points to consider when developing a geometric design system are the ability to represent a given object in an efficient way, the ability then to create alterative designs using parametric representation, and the ability to generate an optimal design by means of careful consideration of alterative designs in a consistent fashion via the use of numerical optimization techniques.

References

1. Farouki RT (1999) Closing the gap between CAD model and downstream application. SIAM News 32:5. http://www.siam.org/siamnews/
2. Lee W (1999) Principles of CAD/CAM/CAE systems. Addison-Wesley, Reading
3. Shapiro V, Voelcker H (1989) On the role of geometry in mechanical design. Res Eng Des 1:63–73
4. Bézier P (1986) The mathematical basis of UNISURF CAD system. Butterworths, London
5. Woodward CD (1987) Blends in geometric modelling. In: Martin RR (ed) Mathematical methods of surfaces II. Oxford University Press, London, pp 255–297
6. Tiller W (1983) Rational B-splines for curve and surface representation. IEEE Comput Graph Appl 3:61–69. doi:10.1016/0010-4485(87)90234-X
7. Piegl L, Tiller W (1987) Curve and surface constructions using rational B-splines. Comput Aided Des 19:485–498. doi:10.1016/0010-4485(87)90234-X
8. Schumaker LL (1981) Spline functions: basic theory. Wiley, New York
9. Faux DI, Pratt MJ (1979) Computational geometry for design and manufacture. Ellis Horwood, Chichester
10. Coons SA (1994) Surfaces for computer-aided design of space forms. Project MAC, Report MAC-TR-41, Massachusetts Institute of Technology
11. Mortenson ME (1985) Geometric modelling. Wiley, New York
12. Vasprille KJ (1975) Computer-aided design applications of the rational B-spline approximation form. PhD thesis, Syracuse University, Syracuse, New York
13. Piegl L (1991) On NURBS: a survey. IEEE Comput Graph Appl 11(1):55–71. doi:10.1109/38.67702
14. Farin G (1990) Curves and surfaces for computer-aided design: a practical guide, 2nd edn. Academic Press, New York
15. Hsu W, Woon YI (1998) Current research in the conceptual design of mechanical products. Comput Aided Des 30(5):377–389. doi:10.1016/S0010-4485(97)00101-2
16. Nowacki H, Dingyuan L, Xinmin L (1989) Mesh fairing GC surface generation method. In: Straßer W, Seidel HP (eds) Theory and practise of geometric modelling. Springer, Berlin, pp 93–103
17. Nakajima N, Gossard D (1982) Basic study in feature descriptor. MIT CAD Technical Report, Massachusetts Institute of Technology, Cambridge

18. Nowacki H, Rees D (1983) Design and fairing of ship surfaces. In: Barnhill RE, Boehm W (eds) Surfaces in CAGD. North-Holland, Amsterdam, pp 121–134
19. Hagen H, Schulze G (1990) Variational principles in curve and surface design. In: Hagen H, Roller D (eds) Geometric modelling. Springer, Berlin, pp 161–184
20. Kallay M (1993) Constrained optimisation in surface design. In: Falcidieno B, Kunii TC (eds) Modelling in computer graphics. Springer, Berlin, pp 85–93
21. Brunnett G, Wendt J (1998) Elastic splines with tension control. In: Dæhlen M, Lyche T, Schumaker LL (eds) Mathematical methods for curves and surfaces II. Vanderbilt University Press, Nashville, pp 33–40
22. Bloor MIG, Wilson MJ (1989) Generating blend surfaces using partial differential equations. Comput Aided Des 21(3):33–39. doi:10.1016/0010-4485(89)90071-7
23. Bloor MIG, Wilson MJ (1989) Blend design as a boundary-value problem. In: Straßer W, Seidel HP (eds) Theory and practise of geometric modelling. Springer, Berlin, pp 221–234
24. Ugail H, Bloor MIG, Wilson MJ (1999) Techniques for interactive design using the PDE method. ACM Trans Graph 18(2):195–212. doi:10.1145/318009.318078
25. Farin G, Sapidis N (1989) Curvature and the fairness of curves and surfaces. IEEE Comput Graph Appl 3:25–57. doi:10.1109/38.19051
26. Boehm W (1980) Inserting new knots into B-spline curves. Comput Aided Des 12:199–201. doi:10.1016/0010-4485(80)90154-2
27. Léon JC (1991) Modélisation des courbes et des surfaces pour la CFAO. Hermés, Paris
28. Rappoport A, Helor Y, Werman M (1994) Interactive design of smooth objects with probabilistic point constraints. ACM Trans Graph 13:156–176. doi:10.1145/176579.176582
29. Terzopoulos D, Witkin A (1988) Physically based models with rigid and deformable components. IEEE Comput Graph Appl 8(6):41–51. doi:10.1109/38.20317
30. Celniker G, Gossard D (1991) Deformable curve and surface finite-elements for free-form shape design. Comput Graph 25(4):257–266. doi:10.1145/122718.122746
31. Thingvold JA, Cohen E (1990) Physical modelling with B-spline surfaces for interactive design and animation. Comput Graph 24(2):129–137. doi:10.1145/91385.91430
32. Terzopoulos D, Platt J, Barr A, Fleischer K (1987) Elastically deformable models. Comput Graph 21(4):205–214. doi:10.1145/37401.37427
33. Pentland A, Williams J (1989) Good vibrations: modal dynamics for graphics and animation. Comput Graph 23(3):215–222. doi:10.1145/74333.74355
34. Platt J, Barr A (1988) Constraints methods for flexible models. Comput Graph 22(4):279–288. doi:10.1145/54852.378524
35. Borrel P, Rappoport A (1994) Simple constrained deformations for geometric modelling and interactive design. ACM Trans Graph 13:137–155. doi:10.1145/176579.176581
36. Terzopoulos D, Qin H (1994) Dynamic NURBS with geometric constraints for interactive sculpting. ACM Trans Graph 13:103–136. doi:10.1145/176579.176580
37. Cohen MZ (1994) Theory and practise of structural optimisation. Struct Multidiscip Optim 7:20–31. doi:10.1007/BF01742500
38. Nelder JA, Mead R (1965) A simplex method for function minimisation. Comput J 7:308–313. doi:10.1093/comjnl/7.4.308
39. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (1992) Numerical recipes in C. Cambridge University Press, Cambridge
40. Kirkpatrick S, Gellat D Jr, Vecchi MP (1983) Optimisation by simulated annealing. Science 220(4598):671–680. doi:10.1126/science.220.4598.671
41. Vanderbilt D, Louie SG (1984) A Monte Carlo simulated annealing approach to optimisation over continuous variables. J Comput Phys 56:259–271. doi:10.1023/A:1004680806815
42. Zienkiewicz OCZ, Campbell JS (1973) Shape optimisation and sequential linear programming. In: Gallagher RH, Zienkiewicz OCZ (eds) Optimal structural design. Wiley, London, pp 109–126
43. Raphael TH, Ramana GV (1986) Structural shape optimization: a survey. Comput Methods Appl Mech Eng 57(1):91–106. doi:10.1016/0045-7825(86)90072-1

44. Kristensen ES, Madsen NF (1976) On the optimal shape of fillets in plates subject to multiple in-plane loading cases. Int J Numer Methods Eng 10:1007–1019. doi:10.1002/nme.1620100504
45. Imam MH (1982) Three-dimensional shape optimisation. Int J Numer Methods Eng 18:661–673. doi:10.1002/nme.1620180504

http://www.springer.com/978-0-85729-783-9

Partial Differential Equations for Geometric Design
Ugail, H.
2011, IX, 107 p., Hardcover
ISBN: 978-0-85729-783-9