

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/47761241>

# Applications Of B-Spline Approximation To Geometric Problems Of Computer-Aided Design

Article · January 1973

Source: OAI

---

CITATIONS  
145

---

READS  
365

1 author:



[Richard F. Riesenfeld](#)

University of Utah

88 PUBLICATIONS 2,771 CITATIONS

SEE PROFILE

**APPLICATIONS OF B-SPLINE APPROXIMATION TO  
GEOMETRIC PROBLEMS OF COMPUTER-AIDED  
DESIGN**

by

Richard Riesenfeld

A.B., Princeton University, 1966  
M.A., Syracuse University, 1969

DISSERTATION

Submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Systems and Information Science in the Graduate School of Syracuse University, May 1973.

Copyright 1973

Richard Riesenfeld

To Bill, Robin, and Steve

## ACKNOWLEDGMENTS

During the preparation of this thesis I have become indebted to many people for their patience, advice, and support. Most of all I owe very special thanks to Dr. William J. Gordon, my principal advisor whose central idea this thesis embodies, to Professor Steven A. Coons, my resident advisor, and to Dr. A. Robin Forrest, my advisor across the sea.

The graphics group at Syracuse has been very patient and helpful in listening and contributing to the ideas as they were being developed. Mr. Lewis Knapp has offered particularly valuable assistance in connection with his implementation of a B-spline curve and surface package.

The Syracuse University Computing Center staff constantly served me with many favors and honored my numerous special requests. The data in my APL workspace would have been inextricably lost to APL/360 and O/S-360 were it not for M. Prakash Thatte's special programs to transfer and translate data files.

At the University of Utah, I wish to express my appreciation to Professor Devid C. Evans and Computer Science for the understanding and assistance extended to me while I finished this thesis. Professor Ivan E. Sutherland offered several improvements during that period. Mr. Michael Milochik provided the photographic services that the illustrations required. The final draft was typed by Ms. Jo Ann Rich. Mr. Bui Tuong Phong helped with programs for illustrations. Art work by Mr. Lance Williams appears in several figures.

Mr. Malcom Sabin of British Aircraft Corporation has my gratitude for his candid correspondence while he simultaneously conducted closely related research.

I have developed a strong admiration for the ingenuity involved in the original method that Professor Pierre Bézier of Régie Renault developed as a result of the deep insight into the practical problems surrounding computer-aided curve and surface design. His interest in this work was reassuring.

I appreciate the contributions of illustrations from Dr. A. Robin Forrest, Mr. Kenneth Versprille, Mr. Lewis Knapp, and Mr. James Clark.

Finally I must mention the important counsel and moral support of Ms. Elaine Cohen who was always willing to quite the applicable theorem at any hour.

And thanks to those those names I have failed to mention!

# CONTENTS

	PAGE
ACKNOWLEDGMENTS	iv
I. INTRODUCTION	1
II. BERNSTEIN APPROXIMATION—A REVIEW	5
III. BÉZIER CURVES—GEOMETRIC APPLICATION OF BERNSTEIN APPROXIMATION	17
IV. B-SPLINE APPROXIMATION—A REVIEW	24
V. B-SPLINE CURVES AND SURFACES—AN APPLICATION OF B-SPLINE APPROXIMATION	33
VI. CONCLUSION	45
BIBLIOGRAPHY	73

# LIST OF FIGURES

		PAGE
Figure 1	Sequence of Bézier curves approximating a hand drawn curve.	48
Figure 2	Typical Bézier surface and net.	49
Figure 3	Binomial distribution for $m = 5$	50
Figure 4	Simple example of a curve which is not the graph of a scalar-valued function.	51
Figure 5	Graphs of $X(s)$ vs. $s$ and $Y(s)$ vs. $s$ for the vector-valued cubic polynomial of expression (3.1), and the cross-plot of $Y(s)$ vs. $X(s)$ .	52
Figure 6	Example of a Bézier curve for a 9-sided polygon.	53
Figure 7	Example of a Bézier curve for a 10-sided polygon.	53
Figure 8	Geometric construction of a Bézier curve for the parameter value $s = 1/4$ .	54
Figure 9	Hodographs of Bézier curves.	55
Figure 10	Canonical B-spline basis functions for degrees 0, 1, 2, 3.	56
Figure 11	Graphs of $N_{i,M}(s)$ and $-N_{i+1,M}(s)$ for $M = 1, 2, 3$ .	57
Figure 12	Family of periodic B-spline basis functions $\{N_{1,3}\}_{i=0}^5$ of degree 2.	58
Figure 13	Family of nonperiodic B-spline basis functions $\{N_{1,3}\}_{i=0}^5$ of degree 2.	58
Figure 14	Family of quadratic B-spline basis functions with a double knot at $s = 3$ .	59
Figure 15	Open cubic ( $M = 4$ ) B-spline curve.	59
Figure 16	Open quadratic ( $M = 3$ ) B-spline curve defined by the polygon from Figure 6.	60

	PAGE
Figure 17	Open cubic ( $M = 4$ ) B-spline curve defined by the polygon from Figure 7. 60
Figure 18	Closed quadratic ( $M = 3$ ) B-spline curve. 61
Figure 19	Closed cubic ( $M = 4$ ) B-spline curve defined by 4 vertices on a square polygon. 61
Figure 20	Closed cubic ( $M = 4$ ) B-spline curve defined by 8 vertices on a square polygon. Collinearity of 3 vertices induces interpolation to the middle vertex. 62
Figure 21	Closed cubic ( $M = 4$ ) B-spline curve defined by 16 vertices on a square polygon. Collinearity of 4 vertices induces a linear span connecting the middle two vertices. 62
Figure 22	Perturbing a vertex of the polygon from Figure 21 produces a local change in the B-spline curve. 63
Figure 23	Closed B-spline curve of degree 9 ( $M = 10$ ). 63
Figure 24	Progression of convex hulls for $M = 2, 3, \dots, 10$ . 64
Figure 25	Geometric construction of the B-spline in Example 5.1. 65
Figure 26	Cubic B-spline and hodograph. 66
Figure 27	Closed cubic ( $M = 4$ ) B-spline with triple vertex that induces a cusp and interpolation. 67
Figure 28	Closed cubic ( $M = 4$ ) B-spline with successive triple vertices that induce 2 interpolating cusps joined by a linear segment of a curve. 67
Figure 29	Same cubic ( $M = 4$ ) B-spline curve defined by a 4-sided polygon and a 5-sided polygon. 68
Figure 30	Half-tone picture of a canonical biquadratic ( $L, M = 3$ ) B-spline basis function with darkened knot lines. 69

	PAGE
Figure 31 Pictures of a simple B-spline surface cut by a numerically controlled milling machine.	70
Figure 32 Pictures of a complicated B-spline surface cut by a numerically controlled miling machine.	71
Figure 33 Summary of results	72

# Chapter I

## INTRODUCTION

### Statement of Problem

The central problem that this thesis addresses is the problem of interactively designing free-form curves and surfaces on a computer graphics display. With slight variations, the same methods may be used for a broad class of data fitting and data smoothing problems.

This class of problems is part of a study that Forrest has termed *computational geometry* [21]. The goal of this thesis is to provide a mathematical system that meets Forrest's criteria [20, p.3]:

Not only must we be able to represent the required shapes accurately and in a form amenable to computation, but we must also provide a good interface between a user who does not necessarily have any mathematical skills and the mathematical representation he is controlling. This is of considerable importance, both to ensure initial acceptance of new techniques and to maintain as many [existing] design procedures as possible.

The two important and readily distinguishable aspects of the central problem were thus pointed out by Forrest. To paraphrase, first we need a satisfactory and suitably general mathematical method for describing or, more appropriately, defining very general free-form curves and "sculptured" surfaces such as those found for instance on hulls, automobile bodies, or even a marble statue. Simply stated, the first aspect of the problem is to give a mathematical description of the shape of an object.

The second and equally important aspect concerns the interface between the underlying mathematical techniques and the designer or user who may have little mathematical training. In order to be successful, a computer-aided design (CAD) system must have appeal

to the designer. It must be simple, intuitive and easy to use. Ideally, an interactive design system makes no mathematical demands on the user other than those to which he has been formerly accustomed through drafting and design experience.

## **Traditional Approach—Make a (Hard) Model**

Traditionally the design process has been implemented in a non-interactive way. The designer goes about his chore in essentially the same way that it has always been done—draw a picture or sketch, make a model, then have that model copied. There are certain modern tools and conveniences like large drafting boards, French curves or sweeps, various drawing instruments, computers with graphics terminals, and various sculpturing instruments for working in a variety of materials; but this basic procedure, the basic sequence of steps, has remained relatively unchanged through centuries of manufacturing design.

From the point of view of modern manufacturing methods there are many deficiencies in the traditional approach to design. To the extent that traditional design techniques have been automated, the automation has consisted principally of mathematically copying pre-existing curves or surfaces. Deriving a viable mathematical definition from a “hard” model, that is, an actual 2-dimensional or 3-dimensional replica, is very expensive and time consuming, and, in general, the model is imprecise. It may not be exactly symmetric where it was intended to be. Moreover, many other types of inaccuracies enter in to an entirely “free-form” process like this.

Thus, the conventional approach to computer-aided design is characterized by a stage that involved the *copying* of a physical model. The computer is used primarily in the final stages to do data management and curve fitting in a non-interactive way.

## **Bézier and Système UNISURF—Make a Mathematical (Soft) Model**

P. Bézier a director of Régie Renault in Paris, conceived Système UNISURF and guided it through its development. It is different from most CAD systems in several ways. Bézier approached the system from an instinctive, geometrical direction that drew heavily on his knowledge and background in styling. Most importantly, it is highly successful operational system—designers use it every day to design automobiles (and occasionally less practical artifacts).

The interface with the user is a large numerically controlled drafting board which is shared by the designer and the computer. The man-machine dialogue proceeds in the following way for the design of a curve. First the designer sketches on the drafting board until he finds a curve that suits his eye. Then he calls upon his previous experience with Système UNISURF as he draws an (open) polygon about the sketched curve (see Figures 1, 6, 7) which in a crude way mimics the gross shape properties of the curve. The next step is to digitize the vertices of the polygon and to compute the corresponding “Bézier curve.” This curve is then drawn by an automatic drafting machine driven by the computer. Usually there is an intolerable discrepancy between the hand-drawn curve and the machine drawn model, although the resemblance is obvious. After a few cycles of perturbing the polygon and visually comparing the associated curve with the hand drawn curve, a suitable representation is achieved. “Convergence” is purely a matter of subjective taste and aesthetic judgment. Part of the designer’s function at Renault is to exercise this judgment. Figure 1 shows such a “convergent” sequence.

This way of designing curves easily generalized to a method for designing surfaces simply by using the usual cartesian product form of the one-dimensional case [6]. The (open) Bézier polygon becomes a “Bézier net,” that is, a piecewise bilinear net. Figure 2 illustrates the relationship between a Bézier net and a Bézier surface.

There are many reasons for the success of Système UNISURF. One is that the system

manages to extract from the designer a succinct mathematical definition (in the form of an appropriate Bézier polygon or net) of each curve or surface almost at the moment of its conception. The *very early* existence of a *mathematical model* is a great advantage in placing the remainder of the manufacturing process under computer control. From the computer based mathematical model a system can provide a two-dimensional drawing or a three-dimensional physical model cut by numerically controlled machines for immediate inspection. The mathematical model is also used to produce the dies needed in the manufacturing process; for example, for sheet metal stamping of automobile panels. The philosophy at Renault is to leave all aesthetic decisions to the designer and all of the tedious data manipulation to the machine.

In Chapters II and III we will examine the mathematical basis for the method of Bézier. The essence of the success of Bézier system is that it combines modern approximation theory and geometry in a way that provides the designer with computerized analogs of his conventional design and drafting tools. For more details discussions of this method of CAD see Bézier [6, 7, 8], Forrest [18], and Gordon and Riesenfeld [25].

## **B-Spline Curves and Surfaces**

This thesis shows how it is possible to preserve the basic method of Bézier for curve and surface descriptions while generalizing the class of curves and surfaces that it encompasses from parametric polynomials to parametric spline functions. These new curves and surfaces will be based on B-spline approximation instead of the more classical Bernstein polynomial approximation theory which underlies Bézier curves and surfaces.

## Chapter II

### BERNSTEIN APPROXIMATION—A REVIEW

The purpose of this section is to describe the approximation theoretic basis of Bernstein approximation.

#### Definition of Berstein Polynomials

**Definition 2.1** The *Berstein polynomial approximation of degree  $m$*  to an arbitrary function  $f: [0, 1] \rightarrow \mathbb{R}$  is:

$$B_m[f; s] = \sum_{i=0}^m f\left(\frac{i}{m}\right) \phi_i(s) \quad (2.1)$$

where the weighting functions  $\phi_i$  are, for  $s \in [0, 1]$ , the discrete binomial probability density functions for a fixed probability  $s$

$$\phi_i(s) = \binom{m}{i} s^i (1-s)^{m-i} \quad i = 0, 1, \dots, m. \quad (2.2)$$

Graphics of the above weights as functions of  $s$  appear in Figure 3.

#### Convergence Properties

Very often Bernstein polynomials are encountered in a constructive proof of the Weierstrass Theorem. They, in fact, lead to a much stronger theorem of which Weierstrass is a special case:

**Theorem 2.1** Let  $f(s) \in C^{[n]}[0, 1]$ . Then

$$\lim_{n \rightarrow \infty} B_n^{(p)}[f; s] = f^{(p)}(s) \quad (2.3)$$

uniformly on  $[0, 1]$  for  $p = 0, 1, \dots, n$ .

For  $p = 0$ , Theorem 2.1 reduces to the ordinary Weierstrass Theorem. P. J. Davis [15] refers to the convergence behavior described in Theorem 2.1 as the “simultaneous approximation of the function and its derivatives.”

Bernstein approximation does not fare well if the role of convergence is measured in the sup norm.

**Example 2.1** The second degree Bernstein approximation to the function  $f(s) = s^2$  is

$$B_2[s^2; s] = s^2 + \frac{s(1-s)}{2} \tag{2.4}$$

Obviously,  $s^2 < B_s[s^2; s] < s$  for all  $s \in (0, 1)$ . More generally, the  $m^{\text{th}}$  degree Bernstein approximation to  $f(s) = s^2$  is

$$B_m[s^2; s] = s^2 + \frac{s(1-s)}{m}, \tag{2.5}$$

which illustrates the notably slow convergence (like  $1/m$ ) of the Bernstein approximation  $B_m[f]$  to the primitive  $f$ .

On the other hand, we do not wish to discount the possibility that there may exist some norm for which the Bernstein approximation is, in fact, a minimum norm approximation. In view of the fact that Bernstein polynomials provide simultaneous approximation of a function and its derivative, such a norm must clearly involved the derivatives of  $f$ .

## Probabilistic Interpretation

Bernstein approximation theory abounds with probabilistic interpretations. For instance, we can give the following meaning to  $B_m[f; s]$ : Let  $s$  be the probability of the occurrence of a given even in each of  $m$  Bernoulli trials. Then the probabilities of  $0, 1, \dots, m$  occurrences of this are  $\phi_0(s), \phi_1(s), \dots, \phi_m(s)$ , respectively. Moreover, if  $f(\frac{i}{m})$  represents the “value” of obtaining exactly  $i$  occurrences, then expression (2.1) is the expected “value” of the  $m + 1$  trials. Of course, for  $s = 0$  the expected value is simply  $f(0)$ ; and for  $s = 1$  it is  $f(1)$ .

Bernstein polynomials are often encountered in discussions of the Law of Large Numbers. Roughly speaking, this law states that in experiment involving  $m$  Bernoulli trials, the

ratio of the number of successes to the total number of trials approaches the true population probability as  $m$  becomes infinite. More precisely, let  $s$  be the probability of a success. Then for given  $\epsilon, \delta > 0$ , there exists an  $m$  sufficiently large such that the probability that  $i/m$  differs from  $s$  by less than  $\delta$  is greater than  $1 - \epsilon$ , that is,

$$\text{Prob} \left\{ \left| \frac{i}{m} - s \right| < \delta \right\} = \sum_{i \in I} \phi_i(s) > 1 - \epsilon, \quad \text{where } I = \left\{ i : \left| \frac{i}{m} - s \right| < \delta \right\} \quad (2.6)$$

For a fixed value  $s$ , the value of the Bernstein polynomial  $B_m[f; s]$  at  $s$  is essentially determined by the values of  $f$  in a small neighborhood of  $s$ , and

$$\lim_{m \rightarrow \infty} B_m[f; s] = f(s) \quad (2.7)$$

uniformly for all  $s \in [0, 1]$  provided  $f$  is continuous. But this last statement is nothing more than the ubiquitous Weierstrass Theorem.

In general, for any value of  $s \in [0, 1]$ ,  $B_m[f; s]$  is a convex linear combination of the values of  $f$  at the  $m + 1$  nodal points  $f(0), f\left(\frac{1}{m}\right), \dots, f(1)$ . Furthermore, in view of the above remarks, we can think of the approximation as being, in some sense, a statistically smooth convex linear combination of these nodal points.

## The Bernstein Basis

We now recall some of the elementary properties of the binomial probability densities:

$$\phi_i(s) \geq 0, \quad i = 0, 1, \dots, m; \quad \sum_{i=0}^m \phi_i(s) \equiv 1 \quad \text{for } s \in [0, 1]. \quad (2.8)$$

And,

$$i = 1, 2, \dots, m-1 \left\{ \begin{array}{ll} \phi_0(0) = 1; \phi_0^{(j)}(1) = 0 & (j = 0, 1, \dots, m-1) \\ \phi_i^{(j)}(0) = 0 & (j = 0, 1, \dots, i-1) \\ \phi_i^{(i)}(0) = \frac{m!}{(m-i)!} & \\ \phi_i^{(j)}(1) = 0 & (j = 0, 1, \dots, m-i-1) \\ \phi_i^{(m-1)}(1) = (-1)^{m-i} \frac{m!}{(m-i)!} & \\ \phi_m(1) = 1; \phi_m^{(j)}(0) = 0 & (j = 0, 1, \dots, m-1) \end{array} \right. \quad (2.9)$$

These relations serve to completely characterize the Bernstein basis  $\{\phi_i(s)\}_{i=0}^m$  for the linear space  $\mathcal{P}_m$  of polynomials of degree  $m$ .

The maximum of the function  $\phi_i(s)$  occurs at the value  $s = 1/m$  ( $i \neq 0, m$ ) and is

$$\phi_i\left(\frac{i}{m}\right) = \binom{m}{i} \frac{i^i (m-i)^{m-i}}{m^m} \quad (2.10)$$

In particular, note that for  $i \neq 0, m$ :

$$\phi_i(s) < 1 \quad \text{for } s \in [0, 1], \quad (2.11)$$

and that  $\phi_0(0) = 1$ ,  $\phi_0(1) = 0$ , and  $\phi_m(0) = 0$ ,  $\phi_m(1) = 1$ , which imply that the two endpoint values  $f(0)$  and  $f(1)$  are, in general, the only values which are interpolated by the Bernstein polynomial. Figure 3 shows the graphs of the Bernstein basis functions for  $m = 5$ .

From the above relations involving the values of the  $\phi_i(s)$  and their derivatives at the endpoints of the unit interval, it is easy to see that the endpoint derivatives of the Bernstein polynomial itself are given by

$$\begin{aligned} \text{At } s = 0: \quad \frac{d^i}{ds^i} B_m[f; s] \Big|_{s=0} &= \frac{m!}{(m-i)!} \sum_{j=0}^i (-1)^{i-j} \binom{i}{j} f\left(\frac{j}{m}\right) \\ \text{At } s = 1: \quad \frac{d^i}{ds^i} B_m[f; s] \Big|_{s=1} &= \frac{m!}{(m-i)!} \sum_{j=0}^i (-1)^j \binom{i}{j} f\left(\frac{m-j}{m}\right) \end{aligned} \quad (2.12)$$

From these last expressions, note that the  $i^{\text{th}}$  derivatives at the endpoints  $s = 0, 1$  are determined by the values of  $f(s)$  at that endpoint and at the  $i$  points nearest the endpoint. Specifically, the first derivatives are

$$\begin{aligned} B'_m[f; s] \Big|_{s=0} &= \frac{\Delta f(0)}{\frac{1}{m}} = m \left[ f\left(\frac{1}{m}\right) - f(0) \right] \\ B'_m[f; s] \Big|_{s=1} &= \frac{\Delta f\left(\frac{m-1}{m}\right)}{\frac{1}{m}} = m \left[ f(1) - f\left(\frac{m-1}{m}\right) \right] \end{aligned} \quad (2.13)$$

which means that the polynomial is tangent at the endpoints to the straight line joining the endpoint to the neighboring interior point.

## Finite Difference Characterization

It is not difficult to show [15, pp. 108–109] that  $B_m[f; s]$  can be expressed in finite difference form as follows

$$B_m[f; s] = \sum_{i=0}^m \Delta^i f(0) \binom{m}{i} s^i \quad (2.14)$$

where  $\Delta^i$  is the forward difference operator applied  $i$  times. We shall examine some of the geometric implications of these formulae in the following section.

Another expression for  $B_m[f; s]$  can be obtained by using relations (2.12) in the Taylor expansion of the polynomial, *i.e.*,

$$\begin{aligned} B_m[f; s] &= \sum_{i=0}^m B_m^{(i)}[f; s] \Big|_{s=0} \frac{s^i}{i!} \\ &= \sum_{i=0}^m \binom{m}{i} s^i \sum_{j=0}^i (-1)^{i-j} \binom{j}{i} f\left(\frac{j}{m}\right) \end{aligned} \quad (2.15)$$

## Inversion Formula

Now if  $P(s)$  is any  $m^{\text{th}}$  degree polynomial defined for  $s \in [0, 1]$ , it can be regarded as the unique Bernstein polynomial approximation of degree  $m$  to some *unique* set of values  $\{f(0), f(1/m), \dots, f(1)\}$ . This latter set of values is easily determined in terms of the

endpoint derivatives of  $P(s)$  as

$$\begin{aligned} f\left(\frac{j}{m}\right) &= \sum_{i=0}^j \binom{j}{i} \frac{(m-i)!}{m!} p^{(i)}(0) \\ &= \sum_{i=0}^{m-j} (-1)^i \binom{m-j}{k} \frac{(m-i)!}{m!} p^{(i)}(1) \end{aligned} \quad (2.16)$$

Expression (2.16) is an *inversion formula* and can be fruitfully exploited in connection with CAD applications.

## Bounded Derivatives, Monotonicity, Convexity

The remarkable characteristics of the Bernstein polynomials are the extent to which they mimic the principal features of the primitive function  $f$  and the fact that the Bernstein approximation is always at least as smooth as the primitive function. These properties are more precisely described in the following theorems.

**Theorem 2.2** *cf.* [15, p. 114]: *If the  $i^{\text{th}}$  derivative of  $f(s)$  is bounded between the limits  $\alpha_i$  and  $\beta_i$ :*

$$\alpha_i \leq f^{(i)}(s) \leq \beta_i \quad s \in [0, 1] \quad (i = 0, 1, \dots, m), \quad (2.17)$$

*then the  $i^{\text{th}}$  derivative of its  $m^{\text{th}}$  degree Bernstein approximant is bounded by*

$$\alpha_i \leq \frac{m^i}{m(m-1) \cdots (m-i+1)} B_m^{(i)}[f] \leq \beta_i \quad (2.18)$$

*for  $1 \leq i \leq m$ , and for  $i = 0$*

$$\alpha_0 \leq B_m[f] \leq \beta_0 \quad (2.19)$$

In words, for  $i = 0$ , the theorem states that the values of the Bernstein polynomials lie entirely within the range of the maximum and minimum values of  $f(s)$  for  $s \in [0, 1]$ . And, for  $i = 1$ , we conclude that if  $f$  is monotonic, so is  $B_m[f]$ ; for  $i = 2$ , if  $f$  is convex (or concave), so is  $B_m[f]$ . In general, if  $f^{(i)} \geq 0$ , then  $B_m^{(i)}[f] \geq 0$ .

The standard objection to Lagrange polynomial interpolation and minimum norm approximation (*e.g.*, least squares) is that the approximant evidences extraneous features such

as unwanted undulations. In contrast, the above theorem states that the Bernstein polynomial imitates the gross properties of  $f$  extremely well, but at the expense of closeness of fit. We have already noted, however, that  $\lim_{m \rightarrow \infty} B_m[f] = f$ .

## Variation Diminishing Property

Another important result concerning the smoothness of Bernstein polynomials is due to Schoenberg [36].

**Theorem 2.3** *Let  $f$  be the real-value function defined on the interval  $[0, 1]$ . Then*

$$Z[B_m[f]] \leq V[f] \quad (2.20)$$

where  $V[f]$  is the number of sign changes of  $f$  in  $[0, 1]$  and  $Z[B_m[f]]$  is the number of real zeros of  $B_m[f]$  in  $[0, 1]$ .

Schoenberg refers to this property of the Bernstein polynomial as the “variation diminishing property”.

**Definition 2.2** An approximation scheme has the *variation diminishing property* if:

1. It reproduces linear functions exactly, and
2. The approximation has no more zeros than the primitive itself.

By condition (1) above and the fact that  $B_m$  is a linear operator (see (2.4)) we have

$$B_m[f - a - bs] = B_m[f] - a - bs \quad (2.21)$$

Subtracting the linear function  $a + bs$  from  $f$  on both sides of (2.20) and using the above inequality admits the following useful extension

$$Z[B_m[f] - a - bs] \leq V[f - a - bs] \quad (2.22)$$

This means that the number of intersections of the graph of  $B_m[f]$  with any straight lines  $y = a + bs$  does not exceed the number of crossing of that straight line by the primitive function  $f$ .

In the same paper, Schoenberg has also shown that the variation diminishing property cited above leads directly to a proof of the Popoviciu Theorem [33, 37]

$$T(B_m[f]) \leq T(f) \quad (2.23)$$

where  $T(f)$  denotes the more traditional notion of total variation of  $f$  over the domain  $s \in [0, 1]$ . In fact, unless  $f$  is monotonic over  $[0, 1]$ , the inequality in (2.23) is strict.

## The Bernstein Operator

As is the usual practice in approximation theory, we can consider  $B_m$  to be an operator defined on  $C^{[i]}[0, 1]$  for some natural number  $i$ . Obviously,  $B_m$  is a linear operator:

$$B_m[\alpha f + \beta g] = \alpha B_m[f] + \beta B_m[g] \quad (2.24)$$

for any real numbers  $\alpha$  and  $\beta$ .

In the sense of (2.23), the Bernstein operator  $B_m$  can be said to be a *contraction operator* on the space of functions of bounded variation. On the contrary, the commonly used methods of polynomial interpolation and minimum norm approximation have no such smoothing effect.

As we noted in Example 2.1, the Bernstein approximation of degree  $m$  to a given  $m^{th}$  degree polynomial  $P_m$  is *some other*  $m^{th}$  degree polynomial; that is,

$$B_m[P - m] \neq P_m. \quad (2.25)$$

Thus, as distinguished from most linear approximation operators, the Bernstein operator is *not idempotent*.

An obvious corollary to the Schoenberg theorems on the variations diminishing properties of Bernstein polynomials is that successive iterates of the operator  $B_m$  results in

successively “smoother” approximations. In [27] Rivlin and Kelisky showed that as the number of iterations becomes unbounded, the Bernstein approximation

$$B_m B_m \cdots B_m[f] \tag{2.26}$$

converges to the straight line passing through  $f(0)$  and  $f(1)$ .

The definition of  $B_m$  also quickly verifies that it is a positive operator; that is, it produces a positive approximation to a primitive function that is positive over  $[0, 1]$ . However, we do not explore the repercussions of this observation in this thesis.

In view of the many appealing properties of the Bernstein operator, it seems surprising that these techniques have not been more widely used in applications. The explanation for this is, of course, that they converge very slowly in the uniform norm (Example 2.1). However, they do seem eminently well-suited to the geometric problems of free-form curve and surface design in which smoothness rather than convergence is the consideration of overriding importance.

## Bernstein Approximation Methods for Functions of More Than One Variable

The natural Cartesian product (cross product, tensor product) generalization of the Bernstein operator for the case of bivariate functions is the linear operator

$$\begin{aligned} B_{m,n}[f; s, t] &= B_m B_n[f; s, t] \\ &= \sum_{i=0}^m \sum_{j=0}^n \phi_i(s) \psi_j(t) f\left(\frac{i}{m}, \frac{j}{n}\right) \end{aligned} \tag{2.27}$$

where  $\phi_i$  and  $\psi_j$  are the binomial densities of (2.2) and  $f : [0, 1] \times [0, 1] \rightarrow \mathbb{R}$ . As the notation suggests,  $B_{m,n}$  is simply a product of the univariate operators  $B_m$  and  $B_n$ . It should be noted that the operators  $B_m$  and  $B_n$  *commute*, provided  $f$  is continuous:

$$B_m B_n[f] = B_n B_m[f] \tag{2.28}$$

But,  $B_{m,n}$  is *not a projector* since it is not idempotent.

Many of the properties of the bipolynomial  $B_{m,n}[f]$  are easily inferred from the univariate situation. For example,  $B_{m,n}[f]$  coincides with  $f$  in general, *only* at the four corners of the  $(s, t)$  unit square— $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$ ,  $(1, 1)$ . However, along the four boundaries of the square,  $B_{m,n}[f]$  reduces to the appropriate Bernstein polynomial approximation to the value of  $f$  on that edge. For instance

$$\begin{aligned} B_{m,n}[f; s, t] \Big|_{t=0} &= B_m[f; s, 0] \\ &= \sum_{i=0}^m \phi_i(s) f(i/m, 0) \end{aligned} \quad (2.29)$$

The values  $B_{m,n}[f]$  in the interior of the unit square are simply convex weighted combinations of the values of  $f$  at the  $(m + 1) \times (n + 1)$  points of the cartesian product partition  $(i/m, j/n)$ .

Schoenberg's results concerning the variation diminishing properties of univariate Bernstein polynomials carry over into higher dimensions. In other words,  $B_{m,n}$  is a *smoothing operator*. It is interesting to inquire about the limiting situation when the operator  $B_{m,n}$  is iterated. In a manner analogous to the univariate case, it can be shown that

$$\begin{aligned} \lim_{k \rightarrow \infty} \underbrace{B_{m,n}[\cdots B_{m,n}[f] \cdots]}_{k \text{ times}} &= (1-s)(1-t)f(0,0) \\ &+ (1-s)tf(0,1) + s(1-t)f(1,0) + stf(1,1) \end{aligned} \quad (2.30)$$

which is the bilinear function interpolating the four corner values of  $f$ .

In the extensions from one to two or more independent variables, there are always four possibilities to be considered. These are the operators  $B_m$  and  $B_n$  themselves in parametric form, the product operator  $B_m B_n$  in (2.26) and the *Boolean sum operator*

$$B_m \oplus B_n \equiv B_m + B_n - B_m B_n \quad (2.31)$$

More explicitly, this last is given by

$$\begin{aligned} (B_m \oplus B_n)[f] &= \sum_{i=0}^m \phi_i(s) f\left(\frac{i}{m}, t\right) + \sum_{j=0}^n \psi_j(t) f\left(s, \frac{j}{n}\right) \\ &- \sum_{i=0}^m \sum_{j=0}^n \phi_i(s) \psi_j(t) f\left(\frac{i}{m}, \frac{j}{n}\right). \end{aligned} \quad (2.32)$$

(For a detailed study of the extensions of univariate approximation operators to multivariate functions see [22, 23, 24] and the references therein.) It can be easily shown that (2.27) is a very special case of (2.32) and is obtained from the approximations

$$f(i/m, t) = \sum_{j=0}^n \psi_j(t) f(i/m, j/m) \quad (2.33)$$

$$f(s, j/n) = \sum_{i=0}^m \phi_i(s) f(i/m, j/n), \quad (2.34)$$

which means replacing the arbitrary univariate functions  $f(i/m, t)$  and  $f(s, j/n)$  by their Bernstein approximations of degree  $n$  and  $m$ , respectively. When the expressions on the right hand sides of (2.33) and (2.34) are used in place of the functions  $f(i/m, t)$  and  $f(s, j/n)$ , (2.32) collapses to

$$B_n[B_m[f]] + B_m[B_n[f]] - B_m[B_n[f]] \equiv B_{m,n}[f]. \quad (2.35)$$

The Boolean sum operator has some very interesting properties which the reader may readily verify. *Along the perimeter of the unit square*, it interpolates  $f$ , e.g, along the side  $t = 0$ :

$$(B_m \oplus B_n)[f; s, t] \Big|_{t=0} = f(s, 0) \quad (2.36)$$

*regardless of the specific form of the univariate function  $f(s, 0)$* . However, in the interior of the square, the surface  $(B_m \oplus B_n)[f]$  is a convex weighted average of the values of  $f$  along the lines of constant  $s = i/m$  and  $y = j/n$ . The function (2.32) is annihilated by the partial differential operator  $\partial^{m+n+2}/\partial s^{m+1}\partial t^{n+1}$ . If the operator  $B_m \oplus B_n$  is applied iteratively to  $f$ , the limiting function is the bilinearly blended function

$$\begin{aligned} U(s, t) &= (1-s)f(0, t) + sf(1, t) + (1-t)f(s, 0) \\ &\quad + tf(s, 1) - (1-s)(1-t)f(0, 0) - (1-s)tf(0, 1) \\ &\quad - s(1-t)f(1, 0) - stf(1, 1) \end{aligned} \quad (2.37)$$

This function, which is the simplest of the ‘‘Coons functions’’ [11], is the bivariate Boolean

sum analog of linear univariate interpolation. It satisfies the boundary conditions:

$$U(0, t) = f(0, t) \quad ; \quad U(1, t) = f(1, t) \tag{2.38}$$

$$U(s, 0) = f(s, 0) \quad ; \quad U(s, 1) = f(s, 1).$$

## Chapter III

# BÉZIER CURVES: GEOMETRIC APPLICATION OF BERNSTEIN APPROXIMATION

### Parametric Curves

Since an arbitrary curve in the plane or in  $\mathbb{R}^k$  generally cannot be regarded as the graph of a single-valued scalar function, classical approximation theory *per se* is inappropriate for these and many other applications. (For example, consider the curve in Figure 4.) Parametric representations of curves and surfaces handily circumvent these difficulties.

To apply the classical results of (linear) approximation theory to the description of arbitrary curves and surfaces, one typically treats each of the coordinate functions  $x$ ,  $y$ , and  $z$  independently. To illustrate this standard way of differential geometry, consider again the curve in Figure 4. Although it is globally impossible to express this locus as either  $x = x(y)$  or  $y = y(x)$  (*i.e.*, one of the coordinates as a single-valued function of the other), it has a very simple description as a vector-valued cubic

$$F(s) = \begin{pmatrix} X(s) \\ Y(s) \end{pmatrix} = \begin{pmatrix} 5s - 11.5s^2 + 7.5s^3 \\ 2s - s^2 - 0.67s^3 \end{pmatrix} \quad (3.1)$$

Figure 5 shows the graphs of the two cubics  $X(s)$  and  $Y(s)$  and the graph of the curve  $F(s)$  which is their *cross plot*.

### Vector-Values Definitions

Let  $P_i$  ( $i = 0, 1, \dots, m$ ) be  $(m + 1)$  ordered points in  $\mathbb{R}^k$ . In what follows we will refer to the (open) polygon formed by joining successive points as the Bézier polygon  $P = P_0P_1 \cdots P_m$ .

**Definition 3.1** The *Bézier curve* associated with the Bézier polygon  $\mathcal{P}$  is the *vector-valued Bernstein polynomial*  $\mathcal{B}_m[\mathcal{P}]$  given by

$$\mathcal{B}_m[\mathcal{P}] = \sum_{i=0}^m \phi_i(s) P_i \quad (3.2)$$

where the  $\phi_i(s)$  are the Bernstein basis functions of (2.2).

Some examples of Bézier curves appear in Figures 6 and 7.

The right hand side of expression (3.2) is essentially the formulation due to Forrest [18]. Note that this is an approximation to the discrete data  $\{P_i\}$  rather than to a continuous primitive function that is defined over the interval  $[0, 1]$ .

Let us now consider an alternative definition of a Bézier curve in  $k$ -space. Here we assume the existence of an underlying vector-valued ( $k$  components) primitive function  $\mathcal{F}(s)$  defined for  $s \in [0, 1]$ .

**Definition 3.2** The  $m^{\text{th}}$  degree Bernstein approximation to  $\mathcal{F}$  is the *vector-valued Bernstein polynomial*

$$\mathcal{B}_m[\mathcal{F}] = \sum_{i=0}^m \phi_i(s) \mathcal{F}(i/m) \quad (3.3)$$

where the  $\phi_i(s)$  are as in (3.2).

Implicitly we might consider Bézier's primitive function to be the polynomial (piecewise linear) function:

$$\mathcal{F}(s) = m \left\{ \left( \frac{i+1}{m} - s \right) P_i + \left( s - \frac{i}{m} \right) P_{i+1} \right\} \quad \text{for } s \in \left[ \frac{i}{m}, \frac{i+1}{m} \right] \text{ and } 0 \leq i \leq (m-1). \quad (3.4)$$

Substituting the definition of  $\mathcal{F}(s)$  given by (3.4) into (3.3) verifies that the definition reduces properly to the one given by Definition 3.1.

## Convex Hull Property

It was already noted in Chapter II that Bernstein approximation yields a function that is a convex combination of the nodal values. In the case of Bézier curves we have, again

by virtue of the non-negativity and normalization properties of the density function  $\phi_i(s)$ , that every point of the Bézier curve is a convex combination of the vertices of the polygon  $P_i$ . Hence, *the curve must lie entirely within the convex hull of the extreme points of the polygon.*

For a fixed  $s$ , expression (3.2) gives the location of the center of mass of the ensemble of points  $P_i$  having masses  $\phi_i(s)$ . That is,  $\phi_i(s)$  is the *barycentric coordinate of the base point*  $P_i$ . This observation shows that the Bézier curve is intrinsically coordinatized. In other words, the vector-valued function  $\mathcal{B}_m[\mathcal{P}]$  is invariant under euclidean transformations, one can easily show that the transformed Bézier curve is identical to the Bézier curve associated with similarly transformed vertices.

## Geometric Construction of Bézier Curves

Bézier has provided an interesting geometric construction which leads to (3.2). To find the point on the curve corresponding to the parameter value  $s$ , consider the construction illustrated in Figure 8. First, along the  $(i + 1)^{th}$  side ( $i = 0, 1, \dots, m - 1$ ) move a fractional distance  $s$  toward the point  $P_{i+1}$ . Call this new set of  $m$  points  $P_{i+1}^{(1)}$  and consider the  $(m - 1)$ -sided Bézier polygon  $P_1^{(1)} P_2^{(1)} \dots P_m^{(1)}$ . Repeat the proportioning procedure described above on the  $(m - 2)$  sides of this new polygon to obtain a set of  $(m - 1)$  points  $P_2^{(2)} P_3^{(2)} \dots P_m^{(2)}$ . After  $m$  states of reduction one is left with a single point  $P_m^{(m)}$ , which is the point on the curve corresponding to the parameter value  $s \in [0, 1]$ .

Algebraically this construction leads to the recurrence equation

$$P_{i+1}^{(j)} = P_i^{(j-1)}(s) + s \left\{ P_{i+1}^{(j-1)}(s) - P_i^{(j-1)}(s) \right\} \quad (j \leq m) \quad (3.5)$$

where  $P_i^{(0)} = P_i, i = 0, 1, \dots, m$ . The point  $P_m^{(m)}$  is given explicitly in terms of the original vertices by

$$\begin{aligned} P_m^{(m)}(s) = & (1 - s)^m P_0 + s(1 - s)^{m-1} \binom{m}{1} P_1 + s^2(1 - s)^{m-2} \binom{m}{2} P_2 \\ & + \dots + s^{m-1}(1 - s) \binom{m}{m-1} P_{m-1} + s^m P_m \end{aligned} \quad (3.6)$$

which is the same as (3.2).

The geometric construction of a Bézier curve obviously is independent of the coordinate system which respect to which the vertices  $P_i$  are measured. *This again shows that the curve defined by (3.2) is invariant under euclidean transformation of the coordinate system.*

## Parametric Variation Diminishing Property

Recently W. W. Meyer has extended the scalar-valued notion of variation diminishing to higher dimensions [32].

**Definition 3.3** A parametric approximation scheme (that is, the identical scalar-valued scheme applied to each coordinate function) is termed *parametrically variation diminishing* if it is variation diminishing as a scalar-valued approximation scheme.

Meyers uses the following lemma to prove the basic theorem for higher dimensional curves.

**Lemma 3.1** *If an approximation scheme is parametrically variation diminishing, it is invariant under euclidean transformation. That is, the transformed approximation is the same as the approximation to the primitive function transformed. Or, in other words, the order of the operations: (1) approximation, and (2) euclidean transformation, is interchangeable.*

This lemma is a direct consequence of the characteristic property that variation diminishing approximations reproduce linear functions (and constants, *a fortiori*) exactly.

**Theorem 3.1** *No (hyper)plane is pierced more often by a parametrically variation diminishing approximation curve than by the primitive curve itself.*

The proof is short. Surely the theorem is true for any principal (hyper)plane  $x_i = 0$  because it is variation diminishing in the particular coordinate function  $x_i(s)$ . But any

(hyper)plane can be designated as a principal one by a euclidean transformation. Hence the result is true in general.

The notion of variation diminishing characterizes the property that often is alluded to by heuristic phrases such as “fair”, “sweet”, and “smooth”. Bézier curves owe much of their appeal to this rare quality which, of course, is intimately related to the aesthetic characteristics of curve and surface shape.

## Bézier Surfaces

Extending the idea of Bézier curves to surfaces (see Figure 2) is simply a matter of introducing vector-valued functions for  $f(s, t)$  in the basic surface formulae for cartesian product approximation (2.27) and for the Boolean sum (2.32). In the cartesian form of the extension, the Bézier polygon is replaced with the Bézier “net”  $P = \{P_{ij}, i = 0, 1, \dots, m; j = 0, 1, \dots, n\}$ . Substituting the following function

$$f(i/m, j/n) = P_{ij} \quad (3.7)$$

in equation (2.27) gives the following *Bézier surface*  $\mathcal{B}[P]$  that corresponds to the Bézier net  $\mathcal{P}$ .

$$B_{m,n}[\mathcal{P}] = \sum_{i=0}^m \sum_{j=0}^n \phi_i(s) \psi_j(t) P_{ij} \quad (3.8)$$

where the  $\phi_i$  and  $\psi_j$  are the usual binomial density functions of (2.2).

## Hodographs

In the English edition of his book [6], Bézier discusses hodographs and their uses in connection with Bézier curves.

**Definition 3.4** Consider the polygon derived from  $\mathcal{P}$  having vertices

$$P_i^* = P_{i+1} - P_i \quad i = 0, 1, \dots, (m-1) \quad (3.9)$$

$B_{m-1}[\mathcal{P}^*]$ , the Bézier curve of degree  $(m-1)$  is said to be the hodograph of (3.2) (to within a scalar multiple). The hodograph curve defined above is the locus of the points traced out by (a constant multiple of) the derivative vector of (3.2). Simple differentiation of (3.2) establishes this fact.

Hodographs provide a convenient graphical method for detecting the following conditions: If a vector can be drawn from the origin tangent to the hodograph, then (1) there is a point of inflection at that parameter value in the original curve (see Figure 9(ii)). If the hodograph passes through the origin, then (3) there may be a cusp in the curve (see Figure 9(iii)). In practice hodographs are useful for avoiding unwanted “flats,” points of inflection and cusps.

## Previous Extensions of Bézier Curves

In an earlier attempt to enhance the fidelity of the Bézier curve approximation to the polygon, Gordon and Riesenfeld [25] explored parametric Bernstein approximations to the polygon  $\mathcal{P}$  of degree greater than the number of legs  $m$ . The two approaches taken both involved augmenting the original polynomial curves which mimicked the polygon substantially better than the original Bézier method, very high degree polynomials were required. The authors also tried using conditional binomial probability density functions in place of the unconditional binomial probability density functions in place of the unconditional densities. These latter curves were found to lack the aesthetic qualities of the Bézier curves. The tentative conclusions of Gordon and Riesenfeld were that these high degree polynomial and rational extensions seemed inferior to the B-spline generalizations that are described later. That tentative conclusion is, indeed, valid.

A generalization in a different direction that they reported was an *ad hoc* method involving a spline weighted function for generating smooth *closed curves*. The closed curves in that paper can be viewed as intermediate between the polynomial weighting functions and the polynomial spline functions that follow in the next chapters. The periodic weighting

functions used by Gordon and Riesenfeld to generate closed curves are actually polynomial splines with deficient continuity at the knots. That is, they lack the maximum possible degree of spline continuity. Nevertheless, the schemes are simple, easy to use, and practically valuable for designing closed curves.

## Chapter IV

# B-SPLINE APPROXIMATION—A REVIEW

### Splines

The modern mathematical theory of spline approximation was introduced by I. J. Schoenberg in 1946 [37]. In that paper he developed splines for use in a new approach to statistical data smoothing. A partial list of those who have promoted the applications of splines to computer-aided design include J. H. Ahlberg, G. Birkhoff, C. deBoor, S. A. Coons, J. C. Ferguson, H. L. Garabedian, W. J. Gordon, T. Johnson, E. N. Nilson, M. Sabin, and J. L. Walsh. By the early 1960's, General Motors and United Aircraft had become known as institutional proponents of spline techniques. The first applications of splines in computer-aided design were for interpolation and approximation of existing drawings, that is, copying as opposed to design. This thesis differs from these earlier applications of spline theory in that it offers a Bézier-type facility for controlling splines in a *ad initio* design situation. However, the same interactive techniques can be adapted to the fitting of pre-existing data.

A polynomial spline can be viewed as a generalized polynomial that has certain chosen points of derivative discontinuity.

**Definition 4.1** Let  $X = (x_0, x_1, \dots, x_k)$  be a vector of reals such that  $x_i \leq x_{i+1}$ . A function  $S$  is called a (polynomial) *spline function* of degree  $M - 1$  (order  $M$ ) if it satisfies the following two conditions:

1.  $S$  is a polynomial of degree  $M - 1$  on each subinterval  $(x_i, x_{i+1})$ ,
2.  $S$  and its derivatives of order  $1, 2, \dots, M - 2$  are everywhere continuous, that is  $S \in C^{[M-2]}$ .

The points  $x_i$  are called the *knots* and  $X$  is the *knot vector*. We denote the  $M + k - 1$  dimensional space of all such spline functions as  $S(M, X)$ . The restriction of  $S$  to the interval  $(x_i, x_{i+1})$ , which we denote as  $S|_{(x_i, x_{i+1})}$ , is called the  $i^{th}$  *span* of the spline.

## B-Spline Basis for $\mathcal{S}(M, X)$

Since  $\mathcal{S}(M, X)$  is a finite dimensional linear space, it is spanned by a finite set of linearly independent splines. Several bases for  $\mathcal{S}(M, X)$  are in common use. We shall be interested in the so-called B-spline basis because it is the corresponding spline extension of the Bernstein basis, the mathematical underpinning of Bézier curves. Figure 10 presents a progression, ordered by degree, of B-spline basis functions having knots at the integers. There are two important properties of B-splines that one can readily observe in Figure 10. First, for degree  $M - 1$  the basis functions have finite local support of width  $M$ . Secondly the progression of differentiability class is striking at lower degrees. Figure 12 displays the complete set of periodic B-spline basis functions  $\{N_{1,3}\}_{i=0}^5$  of degree 2 for the linear space of quadratic spline functions with period 5 and with knots at the integers. It is evident from the picture that the basis functions all are cyclic translates (modulo 5) of the canonical basis function  $N_{0,3}$  having the interval  $(0, 3)$  for support. The set of nonperiodic quadratic B-spline basis functions in Figure 13 more closely resemble the Bernstein basis function in Figure 3. At the ends of the interval  $[0, 5]$  the nonperiodic B-spline basis functions obviously are not simple translates of a canonical function.

There are several ways to define the B-spline basis mathematically. Originally Schoenberg gave a divided difference formulation [37]. Or, as Figure 10 might suggest, they can be characterized as the iterate integral of a step function. An instructive observation is that Figures 10 (ii)–(iv) are the results of integrating the sums of the functions appearing in Figures 11 (i)–(iii) respectively. An explicit formula [36, p. 271] for  $N_{0,M}$ , the canonical B-spline basis function of degree  $M - 1$  shown for  $M = 1, 2, 3, 4$  in Figure 10 is:

$$N_{0,M}(s) = \frac{1}{(M-1)!} \sum_{i=0}^M (-1)^i \binom{M}{i} (s-i)_+^{M-1} \quad (4.1)$$

where

$$(s-i)_+ = \begin{cases} s-i & \text{for } s \geq i \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

There are two parameters that specify the linear space of spline functions  $\mathcal{S}(M, X)$ : the

degree  $M - 1$ , and the knot vector  $X$ . Without essential loss of generality in the development, we will take the set of knots to be the integers  $\{0, 1, \dots, n\}$ . This is known as the *uniform B-spline basis*. Later we will relax this restriction.

**Definition 4.2** The *B-Spline basis function*  $N_{i,M}(s)$  of degree  $M - 1$  having support  $(x_i, x_{(i+M \bmod n)})$  is given by the following recursive procedure.

$$N_{i,1}(s) = \begin{cases} 1 & \text{for } x_i \leq s \leq x_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (4.3)$$

And for  $M > 1$ ,

$$N_{i,M}(s) = \frac{s - x_i}{x_{i+M-1} - x_i} N_{i,M-1}(s) + \frac{x_{i+M} - s}{x_{i+M} - x_{i+1}} N_{i+1,M-1}(s) \quad (4.4)$$

(For the periodic basis compute indices and differences modulo  $n$ . The convention  $\frac{0}{0} = 0$  is assumed above.)

This recursive relationship, which we have taken as definitive, was discovered by deBoor and Mansfield [9] and also by Cox [12].

A knot vector  $X$  may contain identical knots up to multiplicity  $M$ . One effect on the basis of a knot  $x_i$  occurring with *multiplicity*  $k_i$ , that is,

$$x_i = x_{i+1} = \dots = x_{i+k-1} \quad (4.5)$$

is to decrease the differentiability of the basis functions  $\{N_{i,M}\}$  at  $x_i$  to  $C^{[M-k_i-1]}$ . A basis that arises from interior knots having multiplicity greater than one is sometimes called a *subspline basis*. Multiplicity  $k_i = 1$  for all  $i$  generates a “full spline” (or simply, spline) basis. If  $k_i = 0$  for all  $i$ , the basis is a spline basis of degree  $M - 1$  and  $C^{[M-1]}$  differentiability everywhere including the knots. But this implies that the knots are “*pseudo-knots*” so that the spline is actually just a single polynomial!

Another effect on the basis of a knot  $x_i$  occurring with multiplicity  $k_i$  is to reduce the support of basis functions that were nonzero at  $x_i$  by as much as  $k_i$  spans. If the point

$s \neq x_i$  is in the interval over which B-spline basis is defined, then there always are  $M$  nonzero basis functions of degree  $M - 1$  at  $s$ . At the knot  $s = x_i$  of multiplicity  $k_i$ , there are  $M - k_i$  nonzero basis functions. The full spline basis ( $k_i = 1$  for all  $i$ ) thus has  $M - 1$  nonzero basis functions at each knot and  $M$  elsewhere. Figure 14 shows an illustration of a subspline basis.

From Definition 4.2 we can generate both the periodic and nonperiodic bases by appropriately specifying the multiplicity of the knots in the vector  $X$ . For the periodic basis  $\{N_{i,M}\}_{i=0}^{n-1}$  of the type in Figure 12, the knot vector is simply the string of integers

$$X = (0, 1, \dots, n) \quad (4.6)$$

or a cyclic shift of the above.

The nonperiodic basis  $\{N_{i,M}\}_{i=0}^{M+n-1}$  over the interval  $(0, n)$ , as in Figure 13 where  $n = 5$ , for example, takes the following knot vector:

$$X = (\underbrace{0, 0, \dots, 0}_M, 1, 2, \dots, n-1, \underbrace{n, n, \dots, n}_M) \quad (4.7)$$

A degenerate knot vector consisting only of end points with multiplicity  $M$

$$X = (\underbrace{0, 0, \dots, 0}_M, \underbrace{1, 1, \dots, 1}_M) \quad (4.8)$$

results in a degenerate spline basis, namely, the Bernstein basis of (2.2). To see this we observe that in (4.3) of Definition 4.2,  $N_{M-1,1}(s) \equiv 1$  and  $N_{i,1}(s) \equiv 0$  for all other values of  $i$ . At the next level  $M = 2$  formula (4.4) gives  $N_{M-2,2}(s) = 1 - s$  and  $N_{M-2,2}(s) = s$  with  $N_{i,2}(s) \equiv 0$  in every other case. As we have seen in (3.5) and (3.6), if we continue this binary tree enumeration we get the binomial density function at each level. *Thus B-splines are a proper spline extension of Bernstein polynomials.* Schoenberg provides an alternative proof of this fact [36, p. 275] based on characterizing a polynomial by its zeros.

Another basic property that we can readily prove from Definition 4.2 by induction on the degree is

$$\sum_i N_{i,M}(s) \equiv 1 \quad (4.9)$$

A proof which is essentially due to deBoor [9, p. 55] is as follows. By associating terms properly under the summation, we can easily get from (4.4) that

$$\sum_i N_{i,M}(s) = \sum_i N_{i,M-1}(s) \quad \text{for } M > 1. \quad (4.10)$$

And, since (4.9) is trivially true for  $M = 1$  (cf. (4.3)), we have the result for all  $M$ .

In [9], deBoor proves (4.9) as a corollary to his proof of the Marsden identity [30]. Arguments similar to the above show that  $N_{i,M}$  is nonnegative and has support  $(x_i, x_{(i+1) \bmod n})$ .

Here again we wish to emphasize that, although we have discussed only splines over equally spaced knots, all proofs and algorithms are equally valid for the more general case in which the knots are non-uniformly spaced, subject only to the condition  $x_1 \leq x_{i+1}$ . A contrast of the properties of the *nonuniform basis* with the uniform basis is given in the next chapter.

## B-Spline Approximation

In this section we review a method of approximation that uses the basis function developed in the previous sections.

**Definition 4.3** The *B-spline approximation of degree  $M - 1$  (order  $M$ )* to an arbitrary function  $f: [0, n] \rightarrow \mathbb{R}$  is

$$S_M[f; s] = \sum_i f(\xi_i) N_{i,M}(s) \quad (4.11)$$

where

$$\xi_i = \frac{1}{M-1} (x_{i+1} + x_{i+2} + \cdots + x_{i+M-1}) \quad (4.12)$$

(for the periodic case compute indices and sums modulo  $n$  in (4.12))

The  $\xi_i$ 's are called the *nodes*. Formula (4.12) for the nodes first appears in a supplement to Schoenberg's 1967 paper [36] and supplied by T. N. E. Greville. Note that the nodes degenerate properly to  $\xi_i = 1/(M-1)$  giving Bernstein approximation for the knot vector (4.8) that generates the Bernstein basis.

In his 1967 paper and elsewhere, Schoenberg stresses the fact that B-spline approximation enjoys the same variation diminishing property that Bernstein approximation does. There are other generalizations of Bernstein approximation that do not have this very desirable attribute.

As a consequence of the local support of a B-spline basis function, B-spline approximation is a *local approximation scheme*. The summation in (4.11) involves, as most,  $M$  nonzero terms. At any point, the approximation only takes into account the local behavior of the primitive function. This enables, for instance, B-spline approximation to reproduce locally linear portions of a primitive function. Moreover, a local perturbation in the primitive function produces only a local perturbation in the B-spline approximation. This stands in contrast to Bernstein approximation which is a *global approximation scheme*.

In a paper that presents a generalize B-spline version of the Weierstrass Approximation Theorem [30], Marsden has shown that lower degree *B-spline approximations can be made to converge considerably faster than the corresponding Bernstein approximation*. In the same paper Marsden also examines the simultaneous convergence of the derivatives of the B-spline approximation to the derivatives of the primitive function  $f$ . B-spline approximations preserve ordinary convexity, although they fall short of Bernstein approximation in simultaneous convergence to derivatives.

## Inversion Formula

Suppose we are given a set of data values  $\{y_0, y_1, \dots, y_m\}$  and we are asked to find the unique set of function values  $\{f(\xi_0), f(\xi_1), \dots, f(\xi_m)\}$  such that the B-spline approximation to  $f$  interpolates the data. That is, we wish to find  $f(\xi_i)$  to satisfy

$$S_M[f; \xi_i] = y_i \quad \text{for } i = 0, 1, \dots, m. \quad (4.13)$$

Or, in matrix form, the following relationship obtains

$$N \cdot F^t = y^t \quad (4.14)$$

where

$$N = \begin{pmatrix} N_{0,M}(\xi_0) & N_{1,M}(\xi_0) & \cdots & N_{m,M}(\xi_0) \\ N_{0,M}(\xi_1) & N_{1,M}(\xi_1) & \cdots & N_{m,M}(\xi_1) \\ \vdots & \vdots & & \vdots \\ N_{0,M}(\xi_m) & N_{1,M}(\xi_m) & \cdots & N_{m,M}(\xi_m) \end{pmatrix}$$

$$F = (f(\xi_0), f(\xi_1), \dots, f(\xi_m))$$

$$Y = (y_0, y_1, \dots, y_m)$$

$N$  is a Gram matrix that is known to be invertible [15] for a distinct set of nodes  $\{\xi_i\}$  which normally is the case. The *inversion formula* is then

$$F^t = N^{-1} \cdot y^t \quad (4.15)$$

## Computational Aspects

Recently deBoor published a study [9] of the computation problems involved in evaluating B-spline functions of the form

$$f(s) = \sum_i a_i N_{i,M}(s), \quad a_i \in \mathbb{R} \quad (4.16)$$

and their derivatives. There he provides some new algorithms that overcome the problems of numerical instability inherent in methods based on previous definitions of B-splines. One stable method is to use Definition 4.2 to calculate the basis functions  $N_{i,M}(s)$  explicitly and then apply the respective coefficients and sum. Its computation utility is the principal reason for adopting Definition 4.2 in this thesis. The above method might be used for construction tables that contain tabular values of basis functions that are used repeatedly.

The following algorithm is a stable method for evaluation the B-spline function  $f$  defined over  $[0, n]$  in (4.16) without explicitly calculating the basis functions  $N_{i,m}(s)$ . This algorithm is very fast if the basis functions are not available in lookup tables.

### DeBoor's Algorithm

*Step 1.* Find  $i$  such that

$$x_i \leq s \leq x_{i+1} \quad (4.17)$$

*Step 2.*

$$f(s) = a_i^{[M-a]}(s) \quad (4.18)$$

where

$$a_j[k](s) = \begin{cases} a_j & \text{for } k = 0 \\ \lambda a_j^{[k-1]} + (1 - \lambda) a_{j-1}^{[k-1]} & \text{otherwise} \end{cases} \quad (4.19)$$

and

$$\lambda = \frac{s - x_j}{x_{j-k+M} - x_j} \quad (j \text{ is the same as above}) \quad (4.20)$$

(For the periodic case calculate indices and differences modulo  $n$ .)

DeBoor also gives the following formula for the  $j^{\text{th}}$  derivative of  $f$  in (4.16)

$$f^{(j)}(s) = (M - 1)(M - 2) \cdots (M - j) \sum_i b_i^{[j]} N_{i, M-j}(s) \quad (4.21)$$

where

$$b_i^{[k]} = \begin{cases} a_i & \text{for } k = 0 \\ \frac{b_i^{[k-1]} - b_{i-1}^{[k-1]}}{x_{i+M-k} - x_i} & \text{otherwise} \end{cases} \quad (4.22)$$

(For the periodic case calculate indices and differences modulo  $n$ .)

Note that we still can use deBoor's algorithm for evaluating  $f^{(j)}(s)$  once the coefficients  $b_i^{[j]}$  are computed by (4.22) if the knots are successive integers. Then the denominator in (4.22) cancels the product  $(M - 1) \cdots (M - j)$  in (4.21). The reader should consult the original paper [9] for more details and variations. Also see Cox [12] who independently developed similar methods.

## Summary

In this chapter we have collected some of the basic results that make B-spline approximation a very attractive tool for use in computer-aided design. One of their most important

characteristics is their variation diminishing property. In addition, spline approximations afford the choice of *either* adding more knots *or* raising the degree of the polynomial in order to increase the number of parameters (degrees of freedom). By introducing more knots and keeping the degree low, they can be made to converge more quickly than Bernstein approximations. Moreover, the “localness” and smoothness (differentiability) are properties that can be controlled. On the practical side, we have seen that the computation algorithms involved are stable, fast, and relatively inexpensive—especially so if the basis consists of simple translates of a canonical B-spline function. The remainder of this thesis is concerned with the exploitation of these properties of B-splines in the context of computer-aided curve and surface design.

## Chapter V

# B-SPLINE CURVES AND SURFACES—AN APPLICATION OF B-SPLINE APPROXIMATION

This chapter applies the B-spline approximation theory of the previous chapter just as Chapter III on “Bézier curves” applied the Bernstein approximation results presented in Chapter II.

### Curve Definitions

Throughout this chapter  $\mathcal{P} = P_0P_1 \cdots P_M$  is a Bézier polygon, either open or closed ( $P_{m+1} \equiv P_0$ ) as the situation requires. Let  $X' = \{x'_i : x'_{i-1} < x'_i\}_{i=1}^k$  be the set of knots over which the B-splines are to be defined. We shall specify a knot vector  $X = (x_0, x_1, \dots, x_n)$ ,  $x_i \in X'$ , such that the  $N_{i,M}$  are correctly defined relative to  $X$  for formula (5.1). In (5.2) below,  $X$  is a cyclic permutation of  $X'$ ; in (5.3)  $X$  is  $X'$  augmented by multiplicities in the first and last components. The uniform basis functions, with which we will normally deal, result from  $x_i = i$ .

**Definition 5.1** The *B-spline curve of degree  $M - 1$  (order  $M$ )* associated with the polygon  $\mathcal{P}$  is

$$S_m[\mathcal{P}] = \sum_{i=0}^m P_i N_{i,M}(s) \quad 0 \leq s \leq x'_k \quad (5.1)$$

The *periodic (or closed) B-spline curve* results when the B-spline basis functions are defined by the knot vector  $X = (x_0, x_1, \dots, x_{m+1})$  where

$$x_i = x'_{(i-M/2) \bmod x'_k} \quad i = 0, 1, \dots, m+1. \quad (5.2)$$

The *nonperiodic (or open) B-spline curve* occurs if  $X = (x_0, x_1, \dots, x_{m+M+1})$  where

$$\begin{aligned} x_i &= x'_0 & i &= 0, 1, \dots, M - 1 \\ x_{i+M-1} &= x'_i & i &= 1, 2, \dots, M - 1 \\ x_{i+m+2} &= x'_{m-M+2} & i &= 0, 1, \dots, M - 1 \end{aligned} \quad (5.3)$$

## Applied B-Spline Approximations

In Chapter III it was shown how Bézier curve can be viewed as the parametric Bernstein approximation of a vector-valued, piecewise linear function  $\mathcal{F}$  whose graph is the polygon  $\mathcal{P}$  (cf. (3.4)). A *B-spline curve* is a parametric B-spline approximation to  $\mathcal{P}$  in a completely analogous fashion. Furthermore, deBoor's algorithm can be applied to compute efficiently the points on the parametric curve. Alternatively the basis set could be computed (for a sufficiently dense sample of values of  $s$ ) beforehand by Definition 4.2 and simply referenced as they are needed. Figures 15–23 are examples of B-spline curves.

In either case, practical implementation requires a routine called B-SPLINE for generating these curves with the following basic parameters:

**Procedure** B-SPLINE( $m, P, M, CLOSED, CURVE$ )

where the parameters are

$m$  = index of last vertex (zero indexing)

$P$  = a matrix of coordinates of  $P_0, P_1, \dots, P_m$

$M - 1$  = degree of B-spline curve

$CLOSED = 1$  or  $0$ , according to whether the desired curve is to be closed or open

$CURVE$  = a matrix of coordinate points that form the corresponding B-spline curve.

The essential instructions in the B-spline routine would consist of the deBoor Algorithm discussed in the previous chapter.

## Geometric Properties of B-Spline Curves

The localness of B-spline approximation and the variation diminishing property are the predominant factors in analyzing the geometric behavior of B-spline curves. By virtue of

the localness property, only the  $m$  neighboring vertices of the controlling polygon determine a point on the curve. Similarly, perturbing a single vertex of the polygon produces only a local perturbation of the curve in the vicinity of that vertex. It is a very desirable property for a designer to have the facility to make local alternation in the shape of a curve with the assurance that other areas of the curve will remain unaltered. Figures 31 and 22 demonstrate this characteristic which is sometimes referred to as “plastic behavior”.

Since the B-spline weighting functions  $N_{i,M}(s)$  is (5.1) are non-negative and sum to 1, each point on a B-spline curve is a *convex* combination of polygonal vertices. Just as with Bézier curves, the weighting functions can be regarded as barycentric coordinates with respect to the base points  $P_i$ . But there are only (at most)  $M$  vertices that determine a point on the curve. This implies a much stronger *convex hull property* than is true for Bézier curves. *Specifically, for a B-spline curve of degree  $m - 1$ , a given point lies within the convex hull of the neighboring  $M$  vertices.* In other words, the union of all points on a B-spline curve must lie within the union of all such convex hulls formed by taking  $M$  successive polygonal vertices. The shaded portions in Figures 24 (i)–(v) show how this region grows from  $M = 2, 3, 4$  and  $M = 10$  (same as Bézier), respectively.

An interesting special case occurs if  $P_i = P_{i+1} = \dots = P_{i+M-1}$ : The convex hull of that set is the point  $P_i$  itself, and so we have that the curve must pass through  $P_i$ . In fact, the entire span of the spline curve determined by these vertices is identically equal to  $P_i$ .

Another interesting special case is when the  $M$  vertices  $P_i, P_{i+1}, \dots, P_{i+M-1}$  are all collinear. Since B-spline approximation is a local scheme, this collinear set of  $M$  points fully determines one span of the B-spline curve. We also know that the variation diminishing property of B-spline approximation assures that it reproduces linear primitives. The conjunction of these two properties is that the span determined by  $M$  collinear vertices is also linear. That is to say, B-spline curves can have locally linear segments smoothly ( $C^{[M-2]}$ ) embedded in them. A simple example of this behavior is shown in Figures 21 and 22. Its usefulness in design is apparent when one considers the frequency of occurrence

of linear segments in the engineering drawings of ordinary mechanical parts. Immediate examples where avoidance of a step change in curvature ( $M \geq 4$ ) is desirable for dynamic reasons including cams, highways, railways, and aircraft fuselage surfaces.

A further consequence of the variation diminishing property of B-spline approximation is that B-spline curves enjoy the same absence of extraneous undulations that the Bézier curves do: Namely, in the sense of Meyer, no (hyper)plane is pierced more often by the curve than by its controlling polygon.

## Geometric Construction

Interpreting the deBoor Algorithm geometrically leads to a constructive method for determining a point of a B-spline curve. Formula (5.4) is a vector-valued statement of (4.19) in terms of polygonal vertices  $P_i$ .

$$P_j^{[k]}(s) = \begin{cases} P_j & \text{for } k = 0 \\ \lambda P_j^{[k-1]}(s) + (1 - \lambda)P_{j-1}^{[k-1]} & \text{for } k > 0 \end{cases} \quad (5.4)$$

where

$$\lambda = \frac{s - x_j}{x_{j-k+M} - x_j} \quad \left( = \frac{s - x_j}{M - k} \text{ for the uniform periodic case} \right) \quad (5.5)$$

**Example 5.1** Given the closed polygon  $P_0P_1 \cdots P_{12}$ , we constructively find the point on the cubic ( $M = 4$ ) B-spline curve that corresponds to  $s = 7.6$ . In Figure 25 we see the relevant vertices and the geometric interpretation to the following calculation. According to (5.2) the knot vector is  $X = (x_0, x_1, \dots, x_{13})$  where

$$x_i = (1 - 2) \bmod 13$$

Step 1 of the deBoor algorithm requiring  $x_1 \leq s < x_{i+1}$  is satisfied by  $x_i = 7$ , or  $i = 9$  according to (5.6). In the notation of (5.4) we seek the value of  $P_j^{[M-1]}(s) = P_9^{[3]}(7.6)$ . Recursively applying the algorithm yields:

$$P_9^{[3]}(7.6) = \lambda P_9^{[2]}(7.6) + (1 - \lambda)P_8^{[2]}(7.6) \quad (5.6)$$

$$\text{where } \lambda = \frac{7.6 - 7.0}{1} = 0.6$$

$$P_9^{[2]}(7.6) = \lambda P_9^{[1]}(7.6) + (1 - \lambda) P_8^{[1]}(7.6) \quad (5.7)$$

$$\text{where } \lambda = \frac{7.6 - 7.0}{4 - 2} = 0.3$$

$$P_8^{[2]}(7.6) = \lambda P_8^{[1]}(7.6) + (1 - \lambda) P_7^{[1]}(7.6) \quad (5.8)$$

$$\text{where } \lambda = \frac{7.6 - 6.0}{4 - 2} = 0.8$$

$$P_9^{[1]}(7.6) = \lambda P_9 + (1 - \lambda) P_8 \quad (5.9)$$

$$\text{where } \lambda = \frac{7.6 - 7.0}{4 - 1} = 0.2$$

$$P_8^{[1]}(7.6) = \lambda P_8 + (1 - \lambda) P_7 \quad (5.10)$$

$$\text{where } \lambda = \frac{7.6 - 6.0}{4 - 1} = 0.47$$

$$P_7^{[1]}(7.6) = \lambda P_7 + (1 - \lambda) P_6 \quad (5.11)$$

$$\text{where } \lambda = \frac{7.6 - 5.0}{4 - 1} = 0.87$$

Note that for the Bézier knot vector (4.8) the construction reduces to the constant proportioning scheme  $\lambda = s$  in all cases and every original vertex is involved in the evaluation. Then it is identical to the construction in Figure 8.

## Hodographs

Given the facility for computing B-spline curves, it is completely trivial to compute the hodograph of a B-spline curve  $\gamma$  of degree  $M - 1$  determined by the (open or closed)

polygon  $\mathcal{P}$ . First calculate the derived vertices

$$P_i^* = (P_i - P_{i-1}) / (x_{i+M-1} - x_i), \quad i = 1, 2, \dots, m \quad (5.12)$$

Let  $\gamma^*$  be the B-spline curve of degree  $M - 2$  determined by the derived polygon (open or closed, respectively)  $P_1^* P_2^* \cdots P_m^*$ . The hodograph of  $\gamma$  is  $\gamma^*$ , which differs from the derivative by a scale factor  $M - 1$ . In the uniform periodic case the scale factor cancels the denominator in (5.12), therefore simply differencing the vertices  $P_i$  gives the tangent curve exactly. By iterating this process, one can obtain  $n^{\text{th}}$  order hodographs. The hodographs of B-spline curves are useful in the same way that the hodographs of Bézier curves are [6]. Figure 26 shows how a typical span of a uniformly spaced knot, closed cubic B-spline curve gives rise to a quadratic hodograph curve.

## Nonuniform B-Spline Curves

A further generalization of B-spline curves is available by using a nonuniform B-spline basis, one in which the knots are not restricted to integer values. The mathematical development of these curves is completely analogous to what has been done so far in this chapter (*cf.* Supplement by Greville in [36]). The added freedom of assigning arbitrary real values to the knots raises the question of what values they should be given.

The author experimented with a knot vector that reflected the euclidean distances of the vertices along the polygon. The relative spacing of the knots was proportional to the respective lengths of the sides. The behavior of the uniform B-spline curves does not appear to differ dramatically from the nonuniform curves unless the relative lengths of the legs of the polygon vary greatly. But, if two polygon vertices are allowed to coalesce, the effect on the nonuniform basis is to produce a multiple knot and a subsequent lowering by one of the differentiability class at the at the multiple knot. This loss of differentiability in the basis is inherited by the curve and this fact can be exploited to great advantage in CAD applications.

Recall from (4.5) that a B-spline basis of degree  $M - 1$  having a knot of multiplicity  $k$  belongs to the differentiability class  $C^{[M-k-1]}$  at that knot. According to this, the uniform basis ( $k = 1$ ) of degree  $M - 1$  always is  $C^{[M-2]}$  differentiable. In the nonuniform case a vertex  $P_i$  that occurs with multiplicity  $k = M/2$  (assume integer division in this discussion) should have the effect of leaving only  $M/2$  nonzero basis functions of differentiability  $C^{[M/2-1]}$  at the corresponding knot  $x_j$ . If these  $M/2$  basis functions are all weighted by the vertex  $P_i = P_{i+1} = \dots = P_{i+M/2-1}$ , then at the parameter value  $x = x_j$  interpolation to  $P_i$  must take place. This is obvious from (5.1) if we factor out the identical vertex  $P_i$  and recall that the basis functions *always* sum to 1, even at the multiple knot  $x_j$ ,

Another possible variant is a scheme that allows the user to specify the multiplicity of the knots (in the basis definition) independently of the multiplicity of the vertices. Then a knot  $x_j$  would require multiplicity  $M - 1$  to induce interpolation at the vertex corresponding to  $s = x_j$ . Multiplicity  $M - 1$  means that only one nonzero basis function  $N_{j,M}$  exists at  $s = x_j$ . From (5.1) it is clear that the curve will pass through  $P_j$  in this case.

Presumably in an interactive design environment one could always arrive at the desired *shape* of curve after sufficient dialogue with the CAD system, but the *parameterization* may not be acceptable. The added advantage of these generalizations is primarily in achieving a different parameterization of the B-spline curve and a different positioning of the knots than that rendered by the uniform basis.

## Cusps

Since no curve can smoothly interpolate to a corner vertex and still remain within the convex hull of the entire polygon, requiring interpolation and containment in the convex hull must give rise to a *cusps or slope discontinuity in the curve* at the corner vertex. Construction the hodograph affirms this conclusion. Moreover, a cusp must result if we recall that the curve is tangent to the legs on either side of an interpolated vertex.

At first glance, the notion of a cusp appears to contradict the  $C^{[M-2]}$  differentiability of

the uniform basis. Indeed the *tangent vector* varies continuously, but this condition does not preclude its vanishing and a zero tangent vector defines a cusp. Other possible generalization can introduce cusps that lead to actually discontinuous tangent vectors. These considerations may be important for CAD systems that employ heavy machines for which inertial effects must be taken into account.

The above discussion indicates that with B-spline curves the designer has a convenient and tractable method for introducing and controlling cusps and this is illustrated in Figures 27 and 28. On the other hand, it is equally important to know that cusps cannot occur unintentionally.

## Knots, Nodes, and Vertices

If we consider the polygon  $\mathcal{P}$  as a piecewise linear function  $\mathcal{F}$ , where  $\mathcal{F}$  is defined so that the B-spline curve is the parametric B-spline approximation to  $\mathcal{F}$ , we recall from (4.11) and (4.12) that  $\mathcal{F}$  is characterized by

$$\mathcal{F}(\xi_i) = P_i \quad i = 0, 1, \dots, m \quad (5.13)$$

where

$$\xi_i = \frac{1}{M-1}(x_{i+1} + x_{i+2} + \dots + x_{i+M-1}).$$

Formula (5.13) relates the *knots*  $x_i$ , the *nodes*  $\xi_i$ , and the *vertices*  $P_i$ . We see immediately that for odd degree (*Meven*) uniform (knots at the integers) B-splines the nodes are also knots where the  $x_{i+1}, x_{i+2}, \dots, x_{i+M-1}$  are successive integers:  $\xi_i = x_{i+M/2}$  in this case. This says that the point on an odd degree uniform B-spline curve that corresponds to the vertex  $P_i$  is the knot on the curve corresponding to the parameter value  $x_{i+M/2}$ . Clearly this is not the case near the ends of an open curve where the knots are not successive integers but they occur with multiplicity.

**Example 5.2** Consider the open cubic ( $M = 4$ ) uniform B-spline curve determined by the polygon  $P_0P_1 \dots P_5$ . The appropriate knot vector is  $X = (0, 0, 0, 0, 1, 2, 3, 3, 3, 3)$ . The set

of nodes is  $\{0, 1/3, 1, 2, 8/3, 2\}$ . Substituting in (5.13) gives

$$\begin{aligned}\mathcal{F}(0) &= P_0, & \mathcal{F}(1/3) &= P_1, & \mathcal{F}(1) &= P_2 \\ \mathcal{F}(2) &= P_3, & \mathcal{F}(8/3) &= P_4, & \mathcal{F}(3) &= P_5\end{aligned}\tag{5.14}$$

Even degree uniform B-splines, on the other hand, approximate the vertices with the (parametric) midpoints between the knots, except for the special cases near the ends of the curve.

**Example 5.3** Consider the open quadratic ( $M = 3$ ) uniform B-spline curve determined by the polygon  $P_0P_1P_2P_3$ . The knot vector for this case is  $X = (3, \overset{0}{4}, 1, 2)$ , where we indicate the modulo equivalence of 0 and 4 by  $\overset{0}{4}$ . The nodes are  $\{1/2, 3/2, 5/2, 7/2\}$ . From (5.13) we see that

$$\begin{aligned}\mathcal{F}(1/2) &= P_0, & \mathcal{F}(3/2) &= P_1 \\ \mathcal{F}(5/2) &= P_2, & \mathcal{F}(7/2) &= P_3\end{aligned}\tag{5.15}$$

## The Inversion Problem

Suppose we pose the problem for B-spline curves of finding the unique (for the minimal number of vertices) polygon  $P_0P_1 \cdots P_m$  that corresponds to any given spline curve  $S$ . That is, can it be viewed as a B-spline curve and, if so, what is the polygon that defines it? The solution is obtained by a simple parametric application of the B-spline inversion formula (4.14). Since we are given the spline curve  $S$ , we know the vector  $X = (x_0, x_1, \dots, x_n)$  and we can easily find the nodes by (4.12). Reformulating the problem, we seek a polygon to satisfy

$$N \cdot (P_0, P_1, \dots, P_m)^t = (S(\xi_0), S(\xi_i), \dots, S(\xi_m))^t\tag{5.16}$$

The problem is solved by inverting  $N$ . The existence and uniqueness follow from the fact that B-splines do, indeed, form a basis for the linear space of splines with fixed knots. Viewed in this way, the problem of finding the vertices  $P_i$  is the problem of finding the proper coefficients to express the given spline  $S$  as a linear combination of the B-spline

basis functions  $N_{i,M}$ . This procedure shows how to construct *interpolating splines* from B-splines.

Another application of the inversion procedure is the problem of reparameterizing a B-spline curve. The situation can arise that the *shape* of a B-spline curve is completely satisfactory but the *parameterization* is not. In this case one may wish to specify a new nonuniform basis that has carefully chosen knot values to produce a particular parameterization. The inversion procedure determines a new polygon that defines the new curve which interpolates the nodes of the original one. Presumably the new curve would be close enough to the original curve so that only a slight adjustment of the new nodes would be necessary to duplicate the shape of the original. An alternative scheme is to specify points on the original curve that are to become the nodes of the new curve with respect to the uniform B-spline basis. The applicability and the relative advantages of these techniques is a matter for further exploration.

The inversion procedure can be used for smoothing an interpolating spline which has unwanted undulations or bumps. It will be obvious where vertices to move and any change will be local.

## **Adding a New Vertex**

When designing a B-spline curve  $S$ , occasionally one would like to add an extra vertex to the polygon in order to increase the *potential* flexibility in a certain region of the curve. Furthermore, it is desirable to add the new vertex in such a way that the shape of the curve is not altered until the new vertex is specifically repositioned. A way to accomplish this is to define a new “pseudo-knot” midway (parametrically) between two of the previously existing knots. Recall from (4.5) that a pseudo-knot is a knot across which derivatives of all orders are continuous, *i.e.*, where the spline is defined by the same polynomial on both sides of the knot.

From (5.13) we see that the extra knot between two original knots defines an extra

node. Applying the inversion procedure of the previous section, we find a new polygon that has one more vertex than the original one. The equivalence of the curves defined by the two polygons is assured by the uniqueness of interpolating splines. As the new vertices are repositioned the pseudo-knot is free to become a *bona fide* knot bearing no more than  $C^{[M-2]}$  differentiability. Figure 29 illustrates a single curve being determined by two different polygons. Note that this procedure leads to a nonstandard parameterization of the knots. As we have just seen, the inversion problem gives a unique polygon only for the one with the minimum number of vertices. There are an infinitude of polygons with more than the minimum number of vertices that give rise to the identical B-spline curve.

## Designing B-Spline Curves

Once a dialogue between a designer and a CAD system has begun, the effect on the curve of moving a vertex becomes immediately apparent. Altering the shape of a curve by perturbing the vertices of its associated polygon is a dynamic process with instantaneous visual feedback if the CAD system has an adequate response time. But designating an initial configuration of vertices to match either a previously sketched curve or a mental image of a curve demands an intimate familiarity with the system. Instead it seems less demanding to allow the designer initially to *specify points that lie on the curve*. The system then constructs an *interpolating spline*. Then, the above inversion procedure is used to determine the unique polygon corresponding to the interpolated spline. Thereafter, the user concentrates on the polygon and uses it to perturb the initial interpolating approximation into a more satisfactory shape.

## B-Spline Surfaces

The extension of B-spline curves to *B-spline surfaces* is exactly analogous to the development of Bézier surfaces from Bézier curves. The B-spline equivalent of the cartesian

product surface (3.7) determined by the net  $\mathcal{P} = \{P_{ij}, i = 0, 1, \dots, m; j = 0, 1, \dots, n\}$  is

$$S_{L,M}[\mathcal{P}] = \sum_{i=0}^m \sum_{j=0}^n N_{i,L}(s) N_{j,M}(t) P_{ij} \quad (5.17)$$

Figure 30 is a half-tone picture of the biquadratic basis function given by (5.17).

For certain purposes such as calculating offsets for milling machine cutters or presenting a smooth-shaded (half-tone) picture of a B-spline surface under prescribed lighting conditions, it is useful to calculate the *vector normal* to the surface. We can extend (4.21) to useful expressions for the *parametric partial derivatives* of (5.17). The *uniform basis* partials simplify to

$$\partial_s S_{L,M}[\mathcal{P}] = \sum_{i=1}^m \sum_{j=0}^n N_{i,L-1}(s) N_{j,M}(t) P_{i,j}^{i*} \quad (5.18)$$

where

$$P_{ij}^{i*} = P_{ij} - P_{i-1,j}, \quad i = 1, 2, \dots, m; j = 0, 1, \dots, n$$

and

$$\partial_t S_{L,M}[\mathcal{P}] = \sum_{i=0}^m \sum_{j=1}^n N_{i,L}(s) N_{j,M-1}(t) P_{i,j}^{j*} \quad (5.19)$$

where

$$P_{ij}^{j*} = P_{ij} - P_{i,j-1}, \quad i = 0, 1, \dots, m; j = 1, 2, \dots, n.$$

A normal to the surface results from the vector cross product of the parametric derivatives (5.18) and (5.19).

Figures 31–32 are pictures of B-spline surfaces cut on the 3-D plotter at Computer Laboratory, University of Cambridge.

## Chapter VI

# CONCLUSION

### Summary

The B-spline approach to curve and surface design developed herein provides a generalization of Bézier that is simple to implement and computationally efficient.

As we saw earlier, a simple general algorithm encompasses both the periodic and non-periodic cases, *i.e.*, both open and closed curves. Furthermore, the shape and complexity of the curve is easily controlled by the addition or deletion of knots. Because it is locally defined, the computation cost of evaluating depends upon the polynomial degree and is virtually independent of the number of knots. Interpolation and cusps are simple to induce using multiple vertices. On the other hand, unwanted cusps and “flat spots” are precluded. B-spline curves are always smooth in the variation diminishing sense which essentially means that they are free of extraneous undulations. Their differentiability is determined by the degree of the spline. Finally, the implementation of this method is entirely feasible with existing computer graphics facilities. It is numerically stable, computationally efficient, and modest in its storage requirements.

But further study is needed in many areas of theory and application. At the time of this writing, the author has been aware of the following research efforts related to B-spline curves and surfaces:

- (a) Forrest at the University of Cambridge is working on geometric properties of B-spline curves and systems for designing and milling B-spline surfaces.
- (b) Negroponte and the Architecture Machine Group at M.I.T. are investigating applications to representing sketched lines and curves.

- (c) Lewis Knapp at Syracuse University has completed a prototype interactive system for designing B-spline curves [28]. Recently, while visiting University of Cambridge, Knapp wrote a B-spline surface packages for their 3-D plotter.
- (d) Coons at Syracuse University has been doing theoretical work with variations on Boolean sum methods of surface delineation.
- (e) Gordon, Meyer and Thomas at General Motors Research Laboratories have been studying theoretical and practical properties of B-spline curves as part of their continuing research in splines and in mathematical surface description.
- (f) James Clark at the University of Utah is developing a real-time system for interactively designing B-spline surfaces using a 3-D input device and the head-mounted display.
- (g) Ronald Resch, also at the University of Utah, is exploring the use of B-spline curves as curvilinear score lines for folded paper modular forms.
- (h) David Pilcher, in connection with Resch's work, is seeking variants of B-spline curves that exhibit alternative kinds of approximation behavior while still preserving the variation diminishing property.
- (i) Mr. Malcom Sabin of British Aircraft Corporation is working on triangular surface patches based on B-splines.
- (j) Roy Keir and Johan Calu are building a hardware B-spline generator to support real-time systems that use B-spline curves and surfaces.
- (k) The author is also pursuing some of the untouched problems regarding manipulation of B-spline curves and their behavior in limiting cases. Of particular interest is the question of positioning the knots of the nonuniform basis in a way that is meaningful to applications.

## **An Overview**

This thesis joins together some of the important ideas of Bézier, deBoor, and Schoenberg to complete the diagram in Figure 33 from Bézier curves to B-spline curves. It identifies Bernstein approximation as the underlying mathematical structure of the Bézier method, appropriately substitutes the more general scheme of B-spline approximation, and then interprets that in the context of the original method.

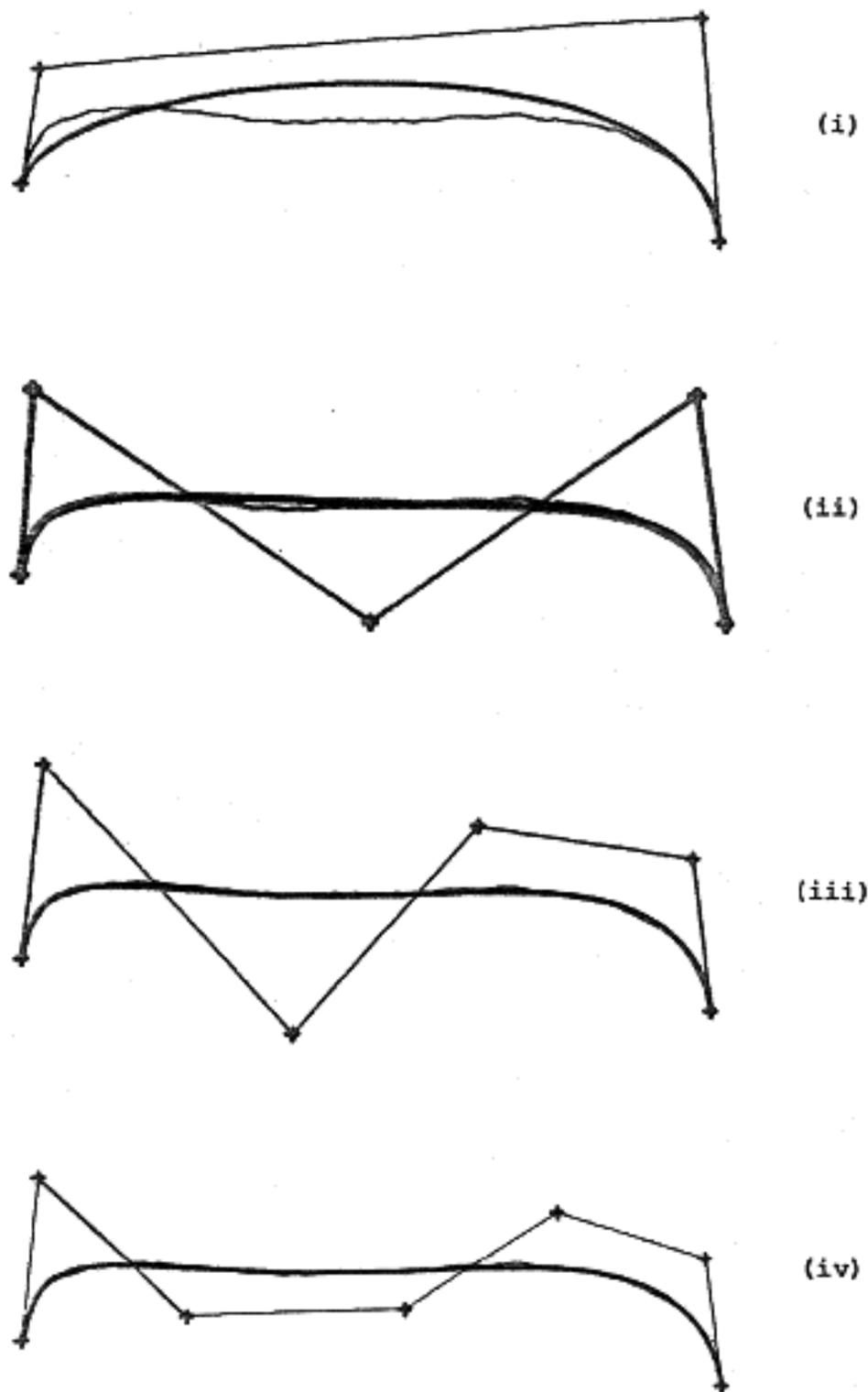


Figure 1 – Sequence of Bézier curves approximating a hand drawn curve.

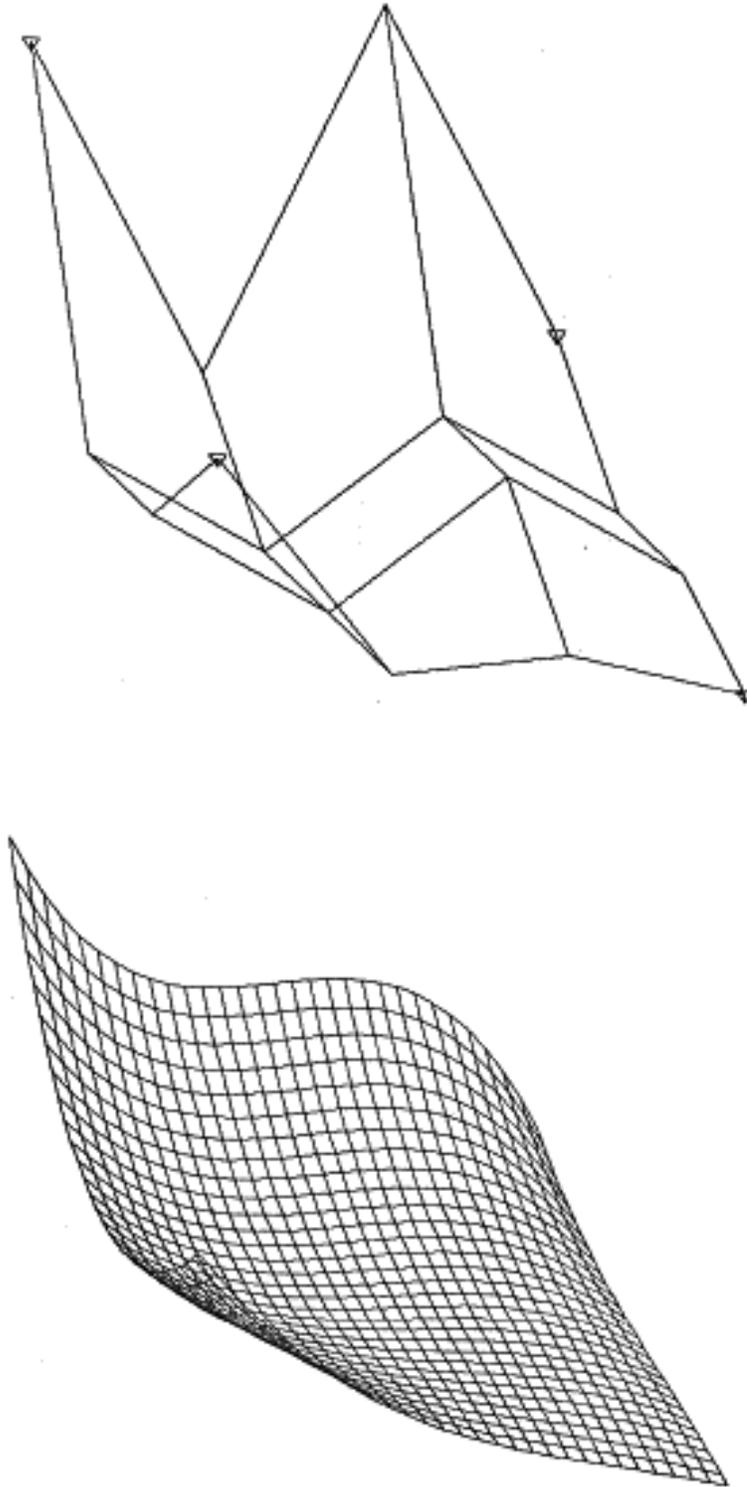


Figure 2 – Typical Bézier surface and net.

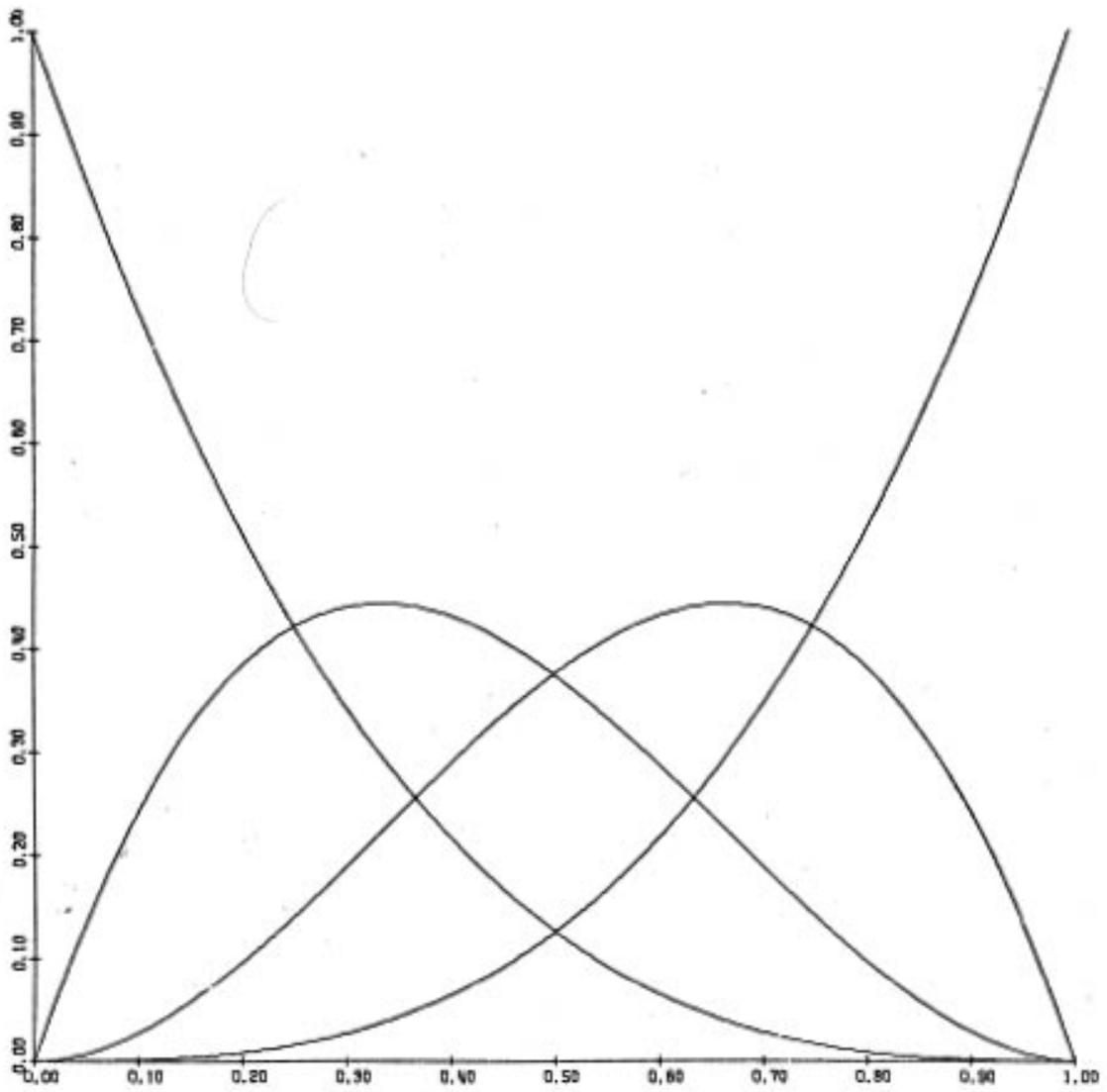


Figure 3 – Binomial distribution for  $m = 5$

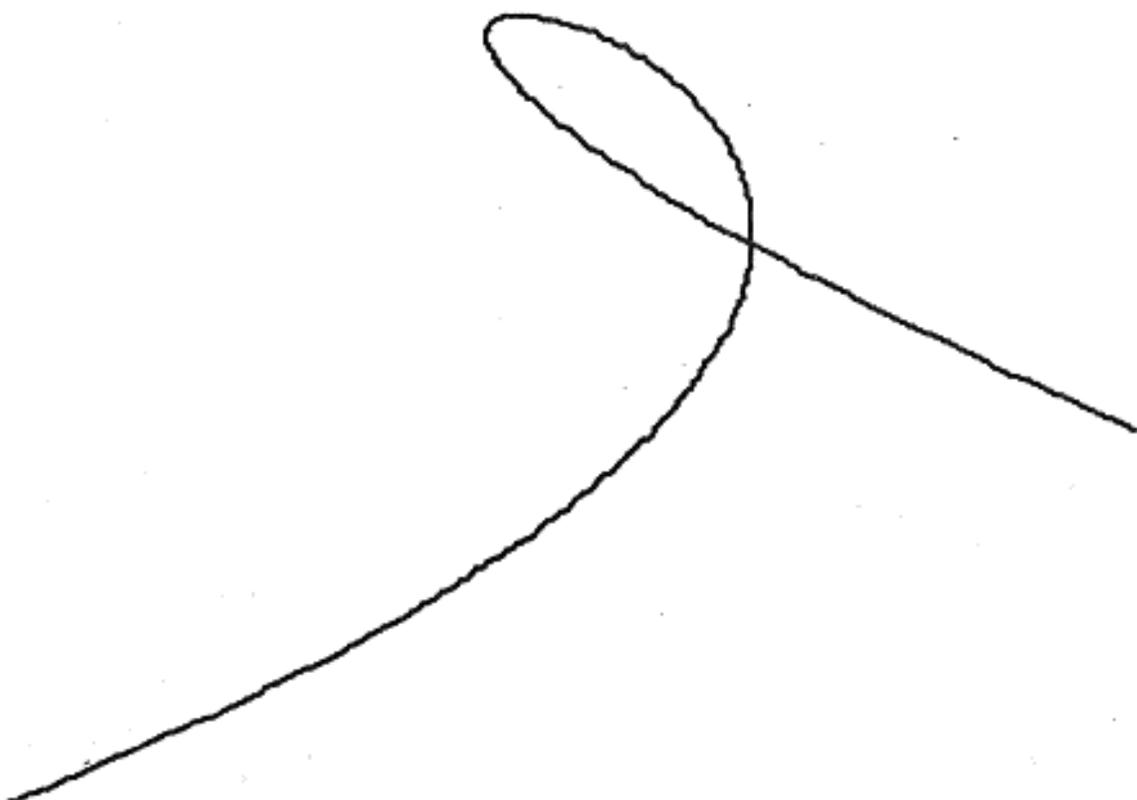


Figure 4 – Simple example of a curve which is not the graph of a scalar-valued function.

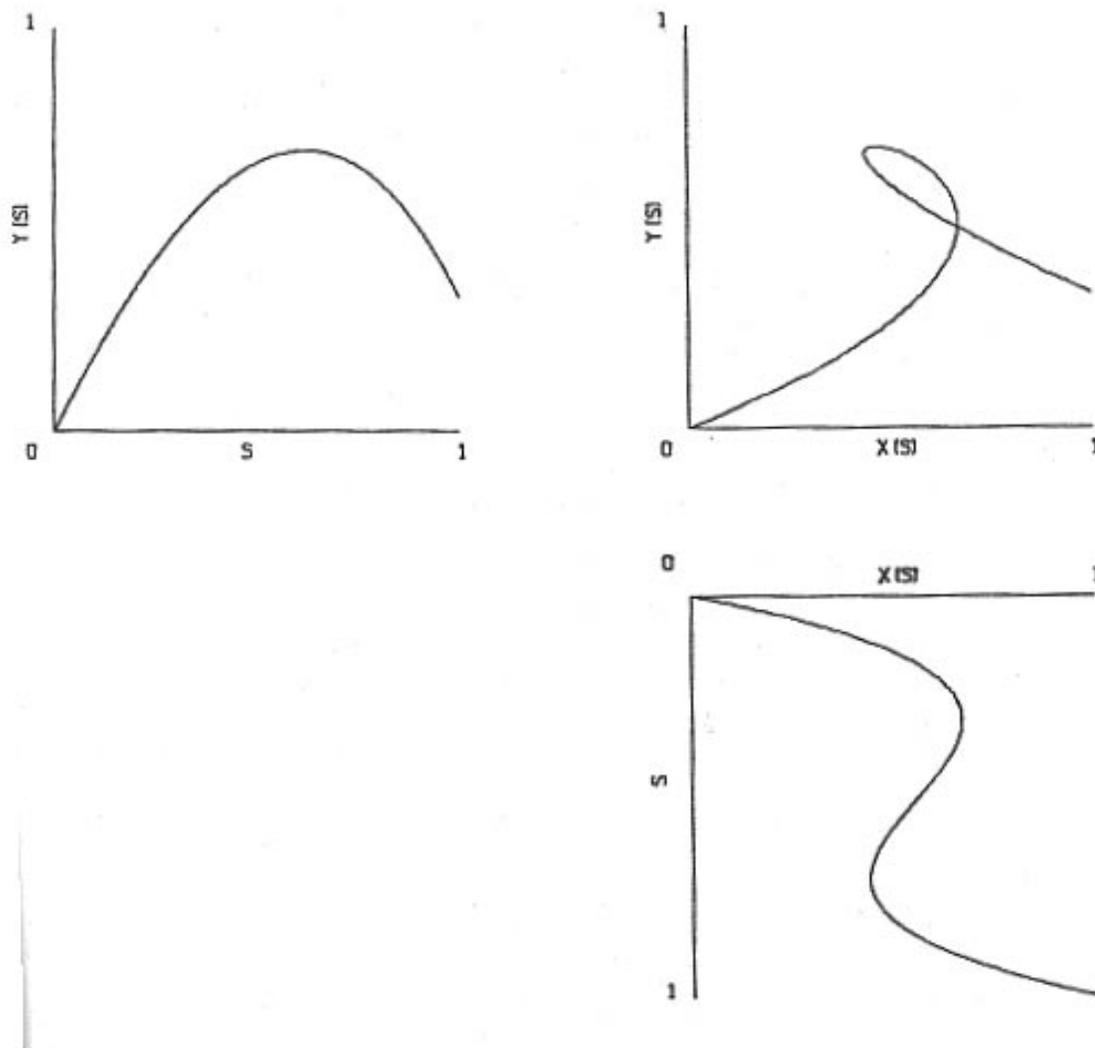


Figure 5 – Graphs of  $X(s)$  vs.  $s$  and  $Y(s)$  vs.  $s$  for the vector-valued cubic polynomial of expression (3.1), and the cross-plot of  $Y(s)$  vs.  $X(s)$ .

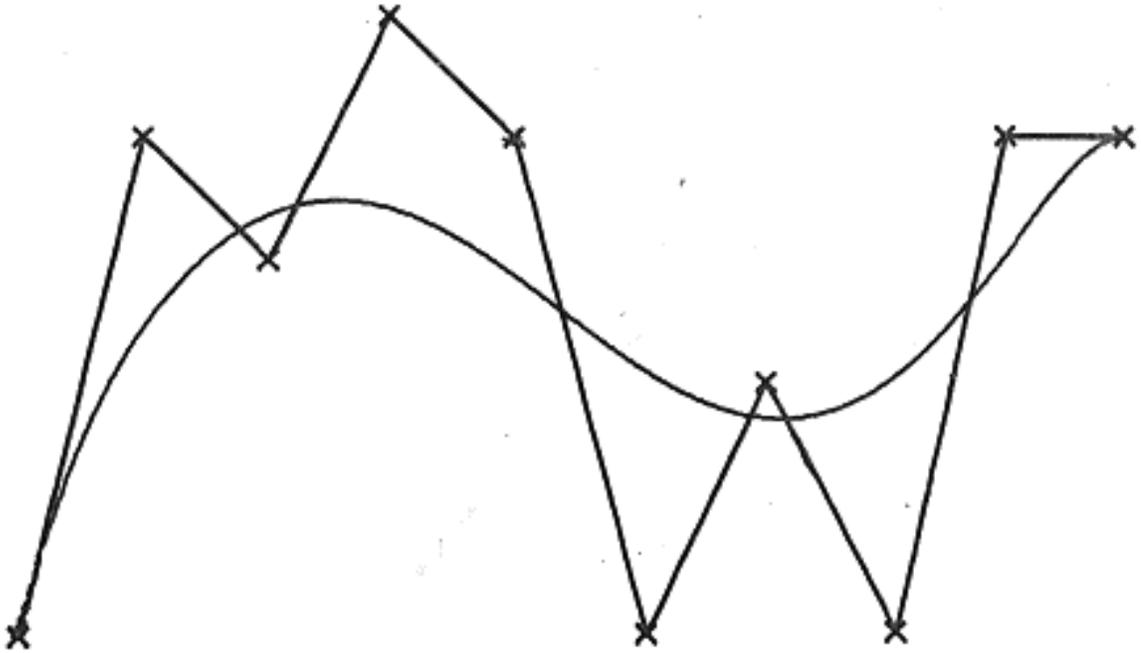


Figure 6 – Example of a Bézier curve for a 9-sided polygon.



Figure 7 – Example of a Bézier curve for a 10-sided polygon.

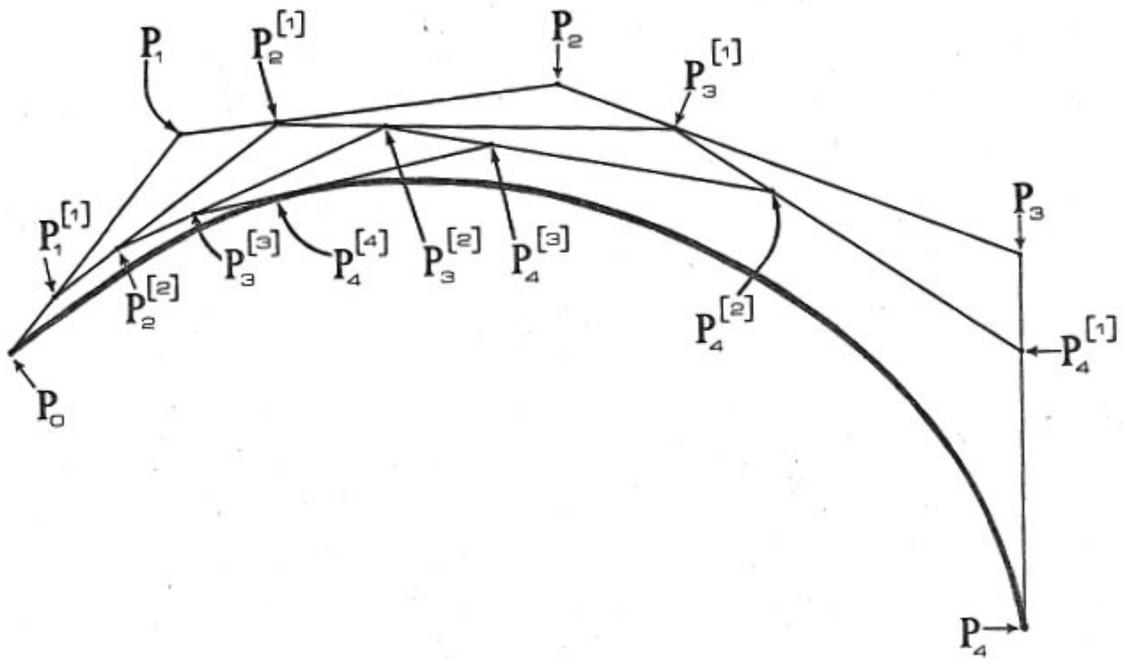


Figure 8 – Geometric construction of a Bézier curve for the parameter value  $s = 1/4$ .

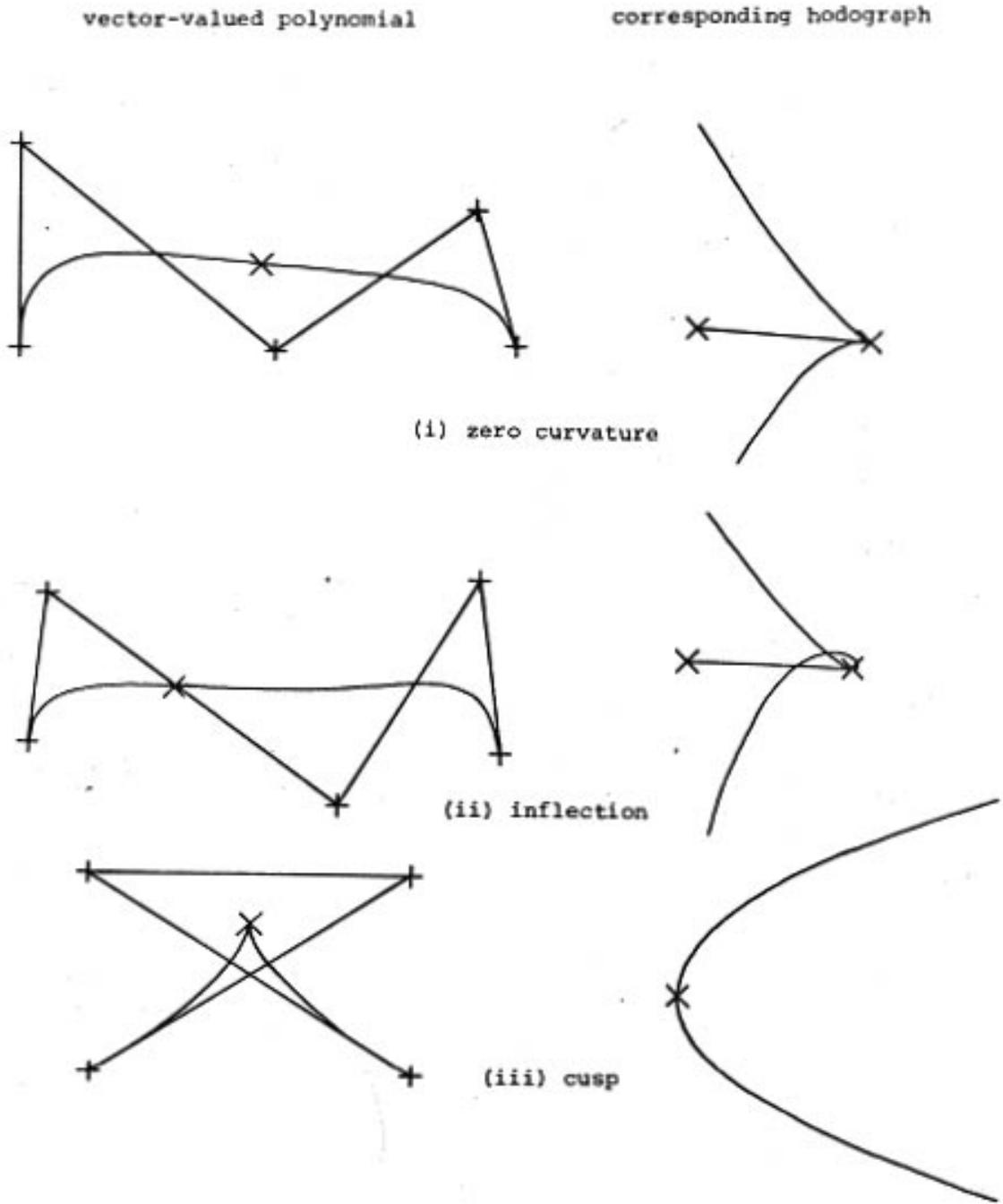


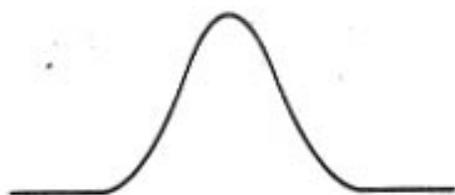
Figure 9 – Hodographs of Bézier curves.



(i) constant



(ii) linear



(iii) quadratic



(iv) cubic

Figure 10 – Canonical B-spline basis functions for degrees 0, 1, 2, 3.

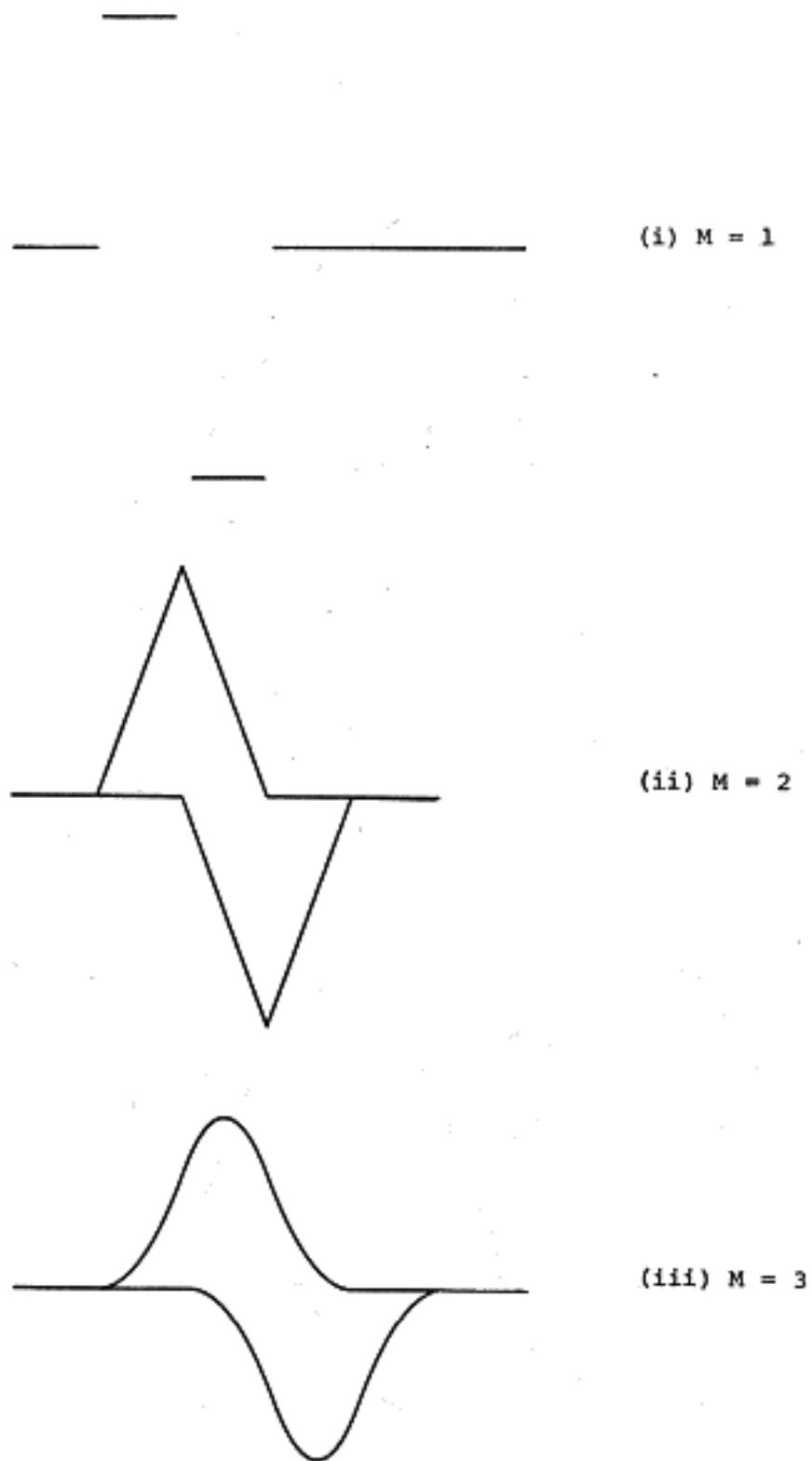


Figure 11 – Graphs of  $N_{i,M}(s)$  and  $-N_{i+1,M}(s)$  for  $M = 1, 2, 3$ .

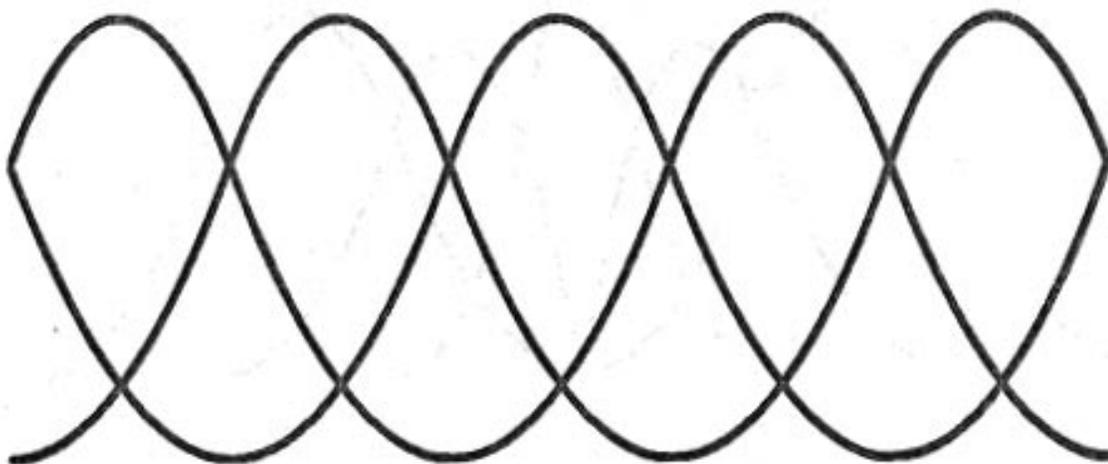


Figure 12 – Family of periodic B-spline basis functions  $\{N_{1,3}\}_{i=0}^5$  of degree 2.

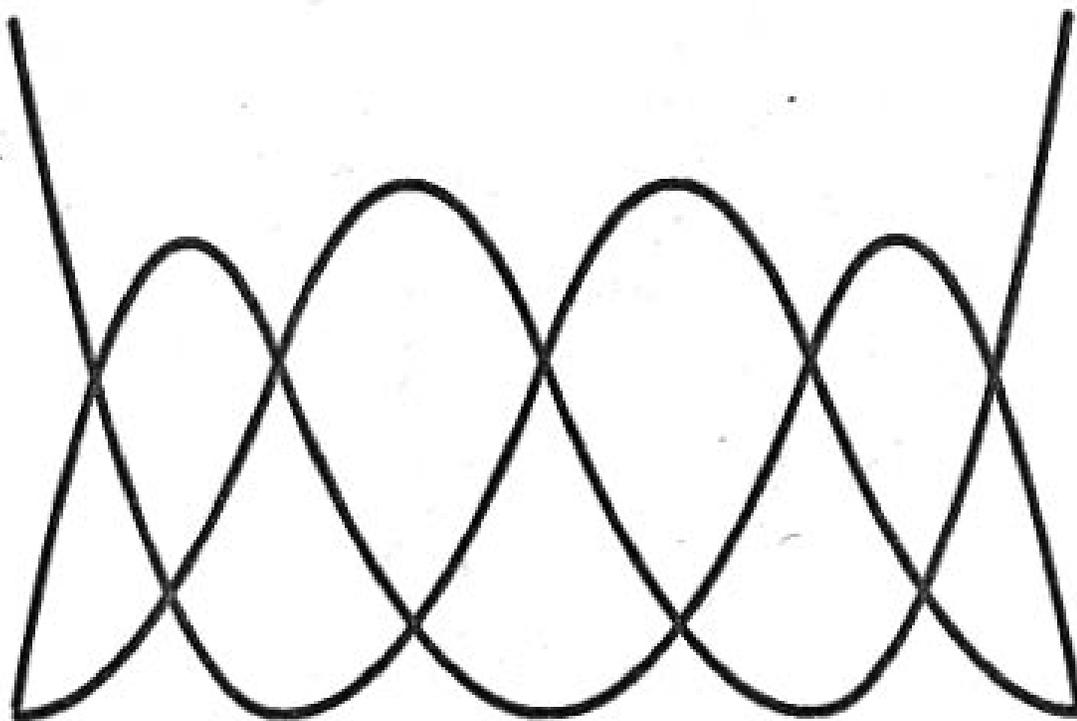


Figure 13 – Family of nonperiodic B-spline basis functions  $\{N_{1,3}\}_{i=0}^5$  of degree 2.

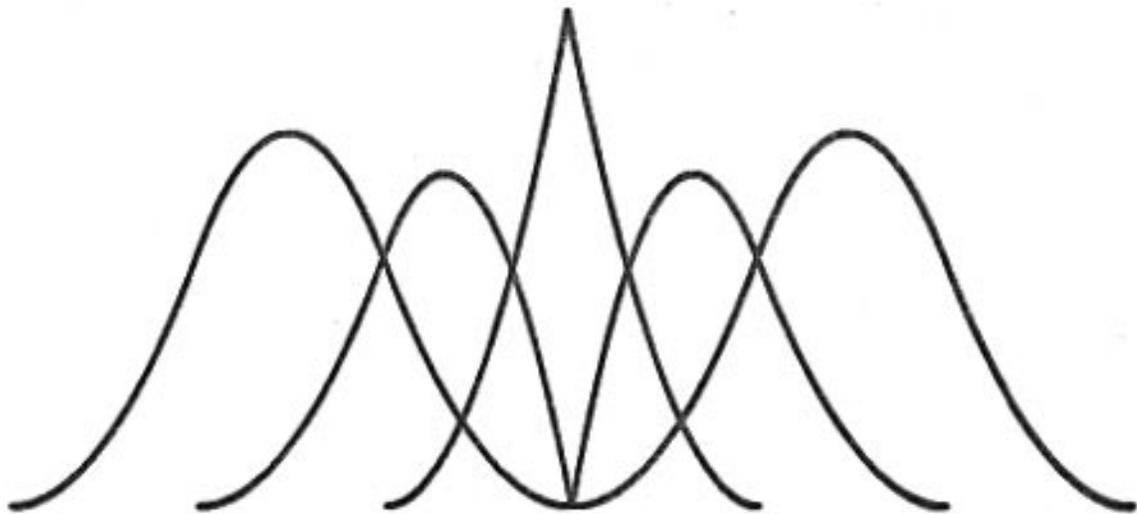


Figure 14 – Family of quadratic B-spline basis functions with a double knot at  $s = 3$ .

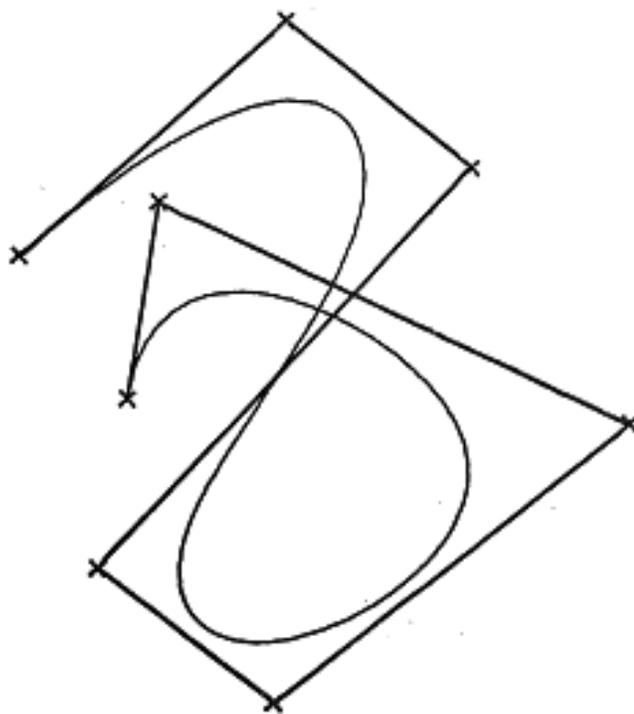


Figure 15 – Open cubic ( $M = 4$ ) B-spline curve.

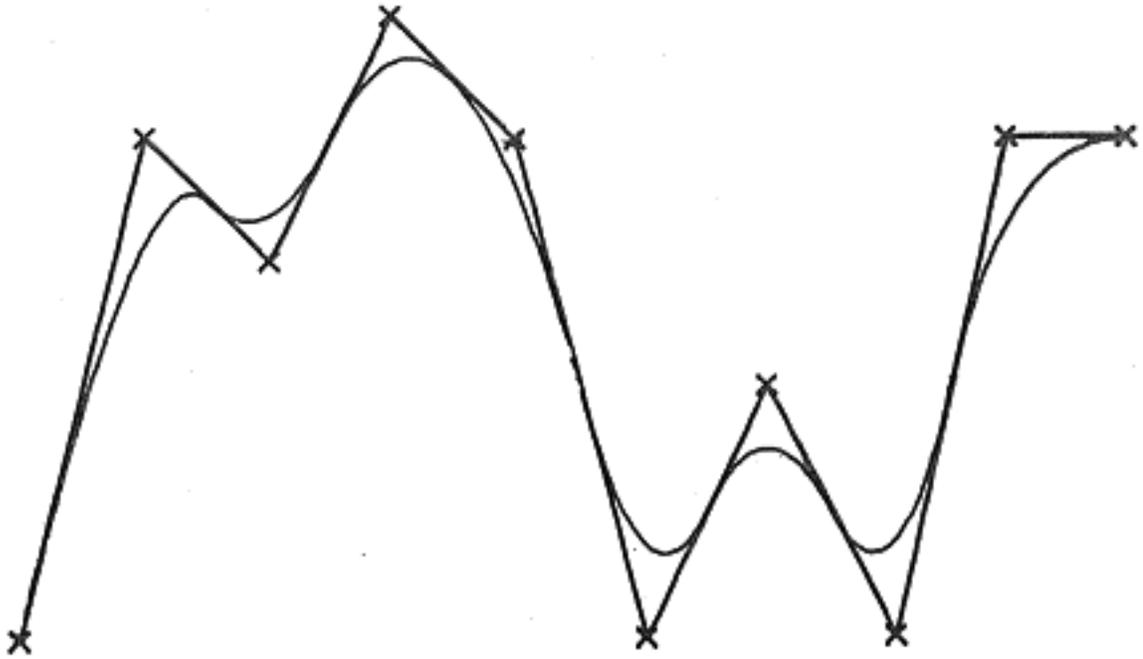


Figure 16 – Open quadratic ( $M = 3$ ) B-spline curve defined by the polygon from Figure 6.



Figure 17 – Open cubic ( $M = 4$ ) B-spline curve defined by the polygon from Figure 7.



Figure 18 – Closed quadratic ( $M = 3$ ) B-spline curve.

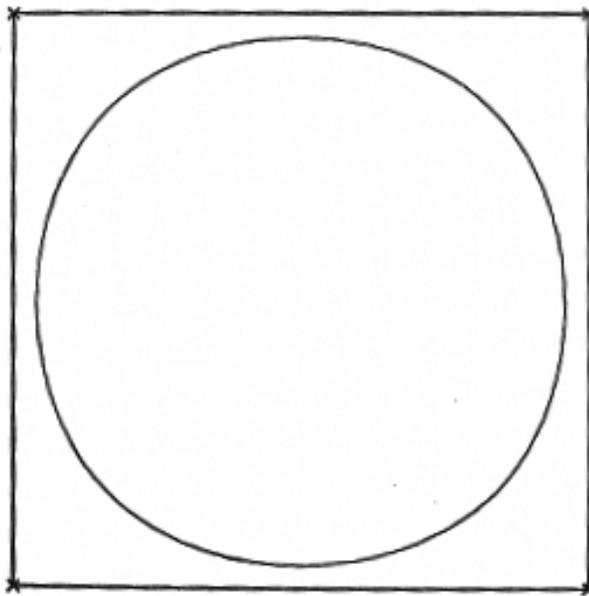


Figure 19 – Closed cubic ( $M = 4$ ) B-spline curve defined by 4 vertices on a square polygon.

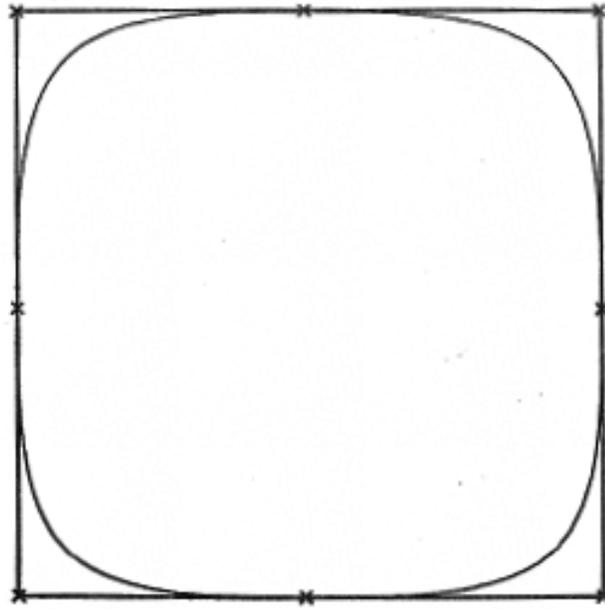


Figure 20 – Closed cubic ( $M = 4$ ) B-spline curve defined by 8 vertices on a square polygon. Collinearity of 3 vertices induces interpolation to the middle vertex.

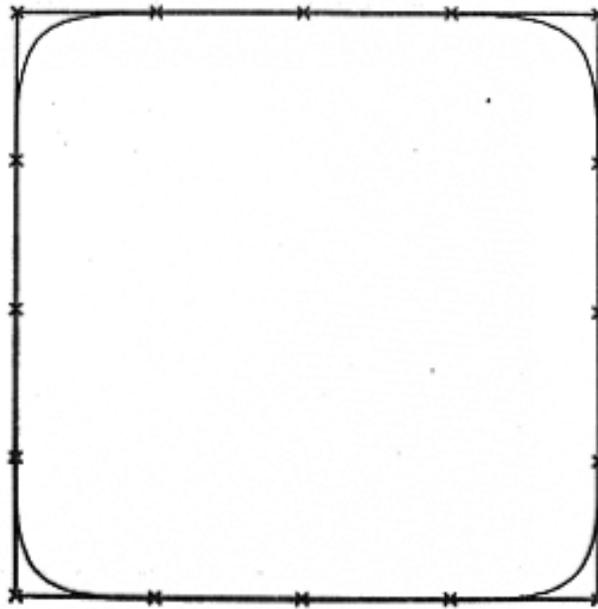


Figure 21 – Closed cubic ( $M = 4$ ) B-spline curve defined by 16 vertices on a square polygon. Collinearity of 4 vertices induces a linear span connecting the middle two vertices.

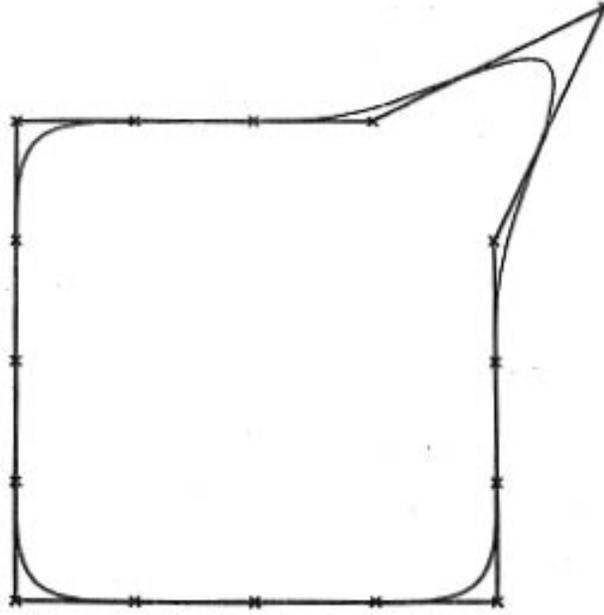


Figure 22 – Perturbing a vertex of the polygon from Figure 21 produces a local change in the B-spline curve.

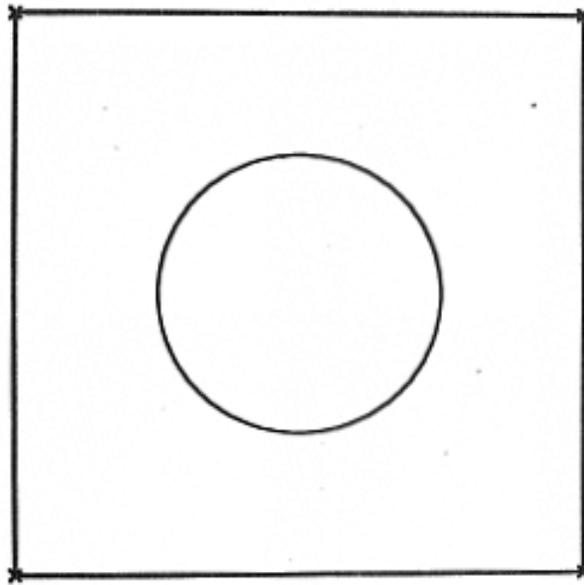


Figure 23 – Closed B-spline curve of degree 9 ( $M = 10$ ).

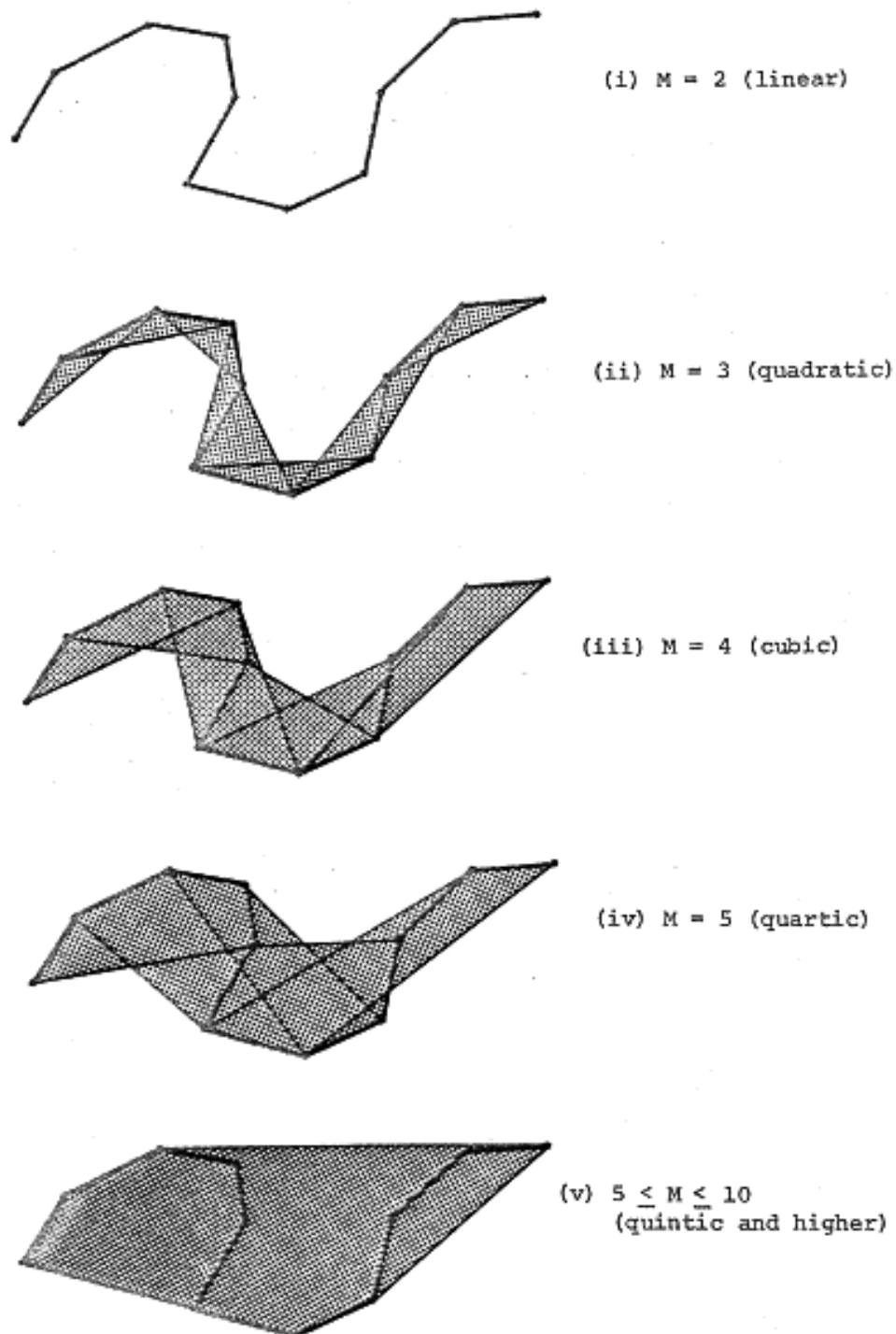


Figure 24 – Progression of convex hulls for  $M = 2, 3, \dots, 10$ .

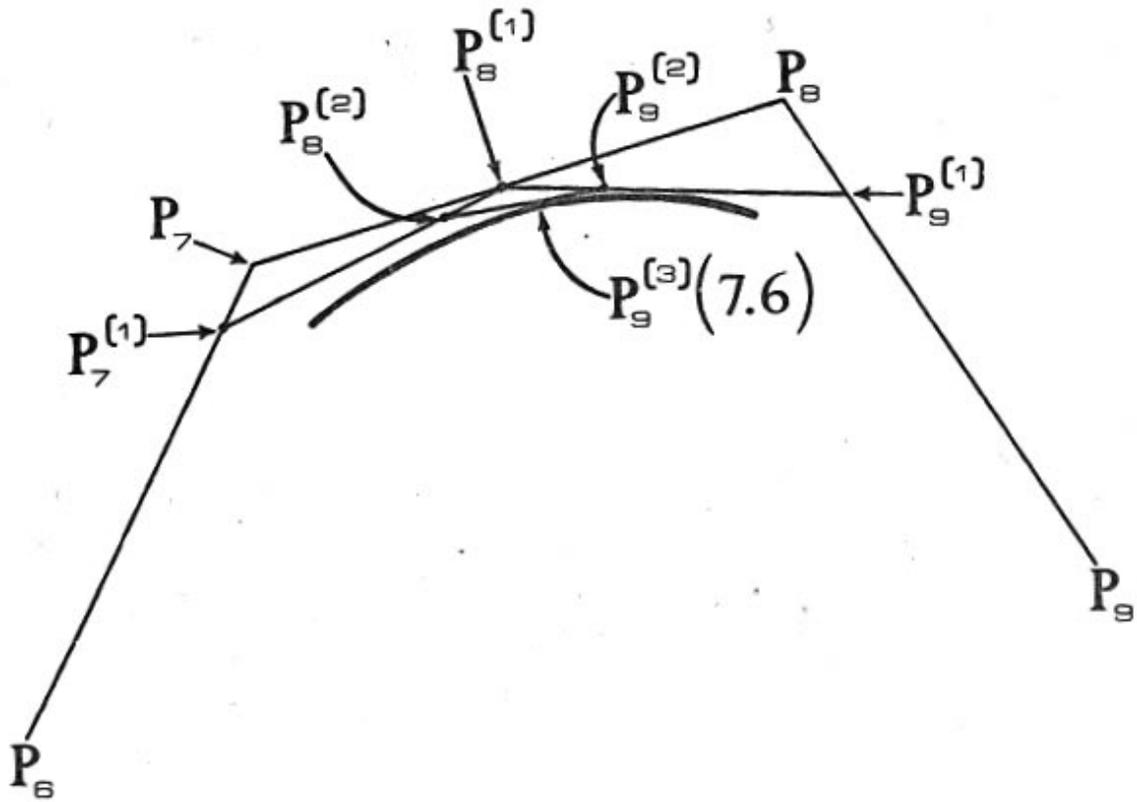


Figure 25 – Geometric construction of the B-spline in Example 5.1.

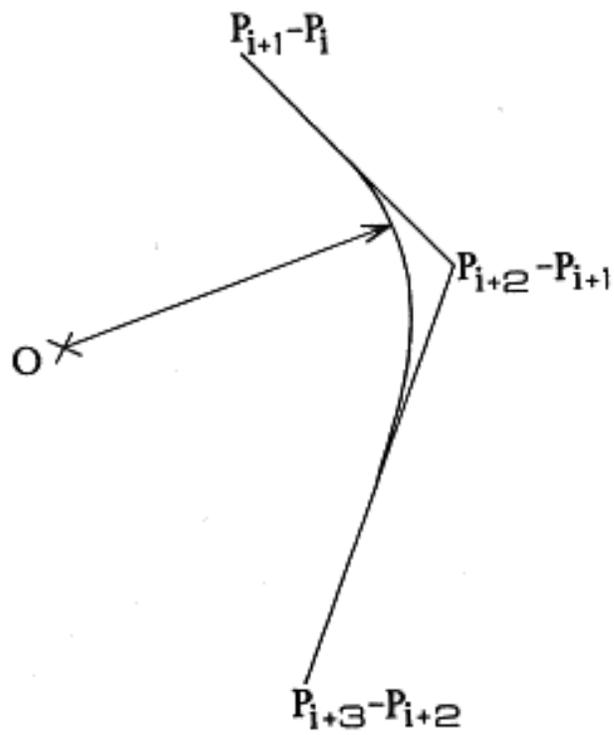
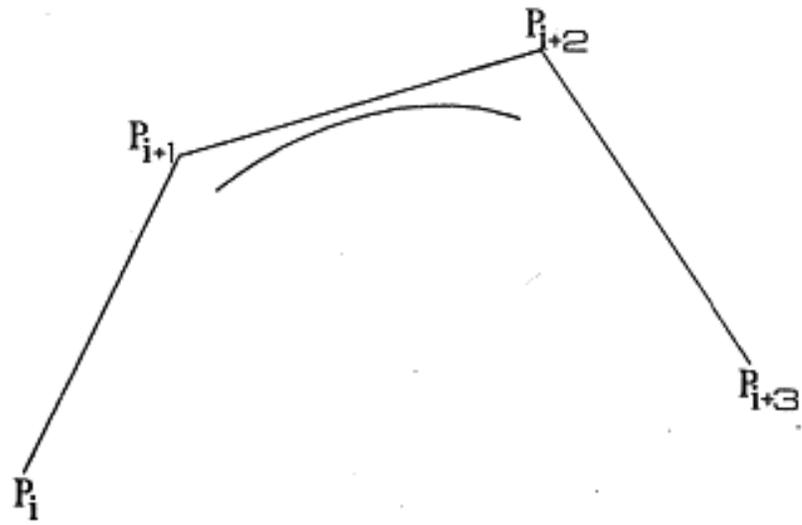


Figure 26 – Cubic B-spline and hodograph.

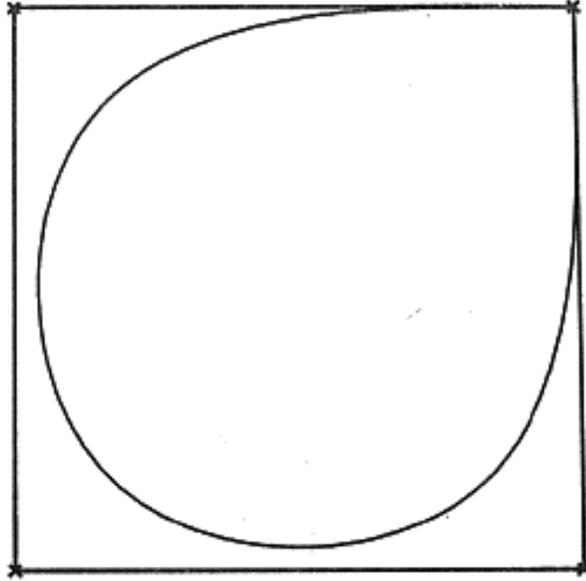


Figure 27 – Closed cubic ( $M = 4$ ) B-spline with triple vertex that induces a cusp and interpolation.

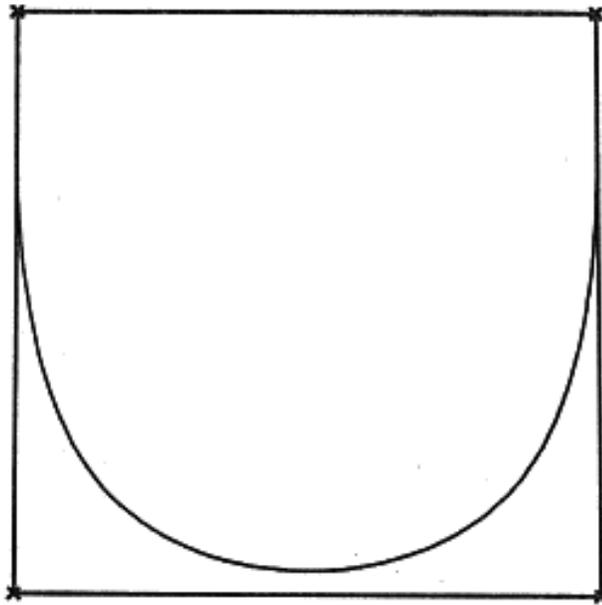


Figure 28 – Closed cubic ( $M = 4$ ) B-spline with successive triple vertices that induce 2 interpolating cusps joined by a linear segment of a curve.

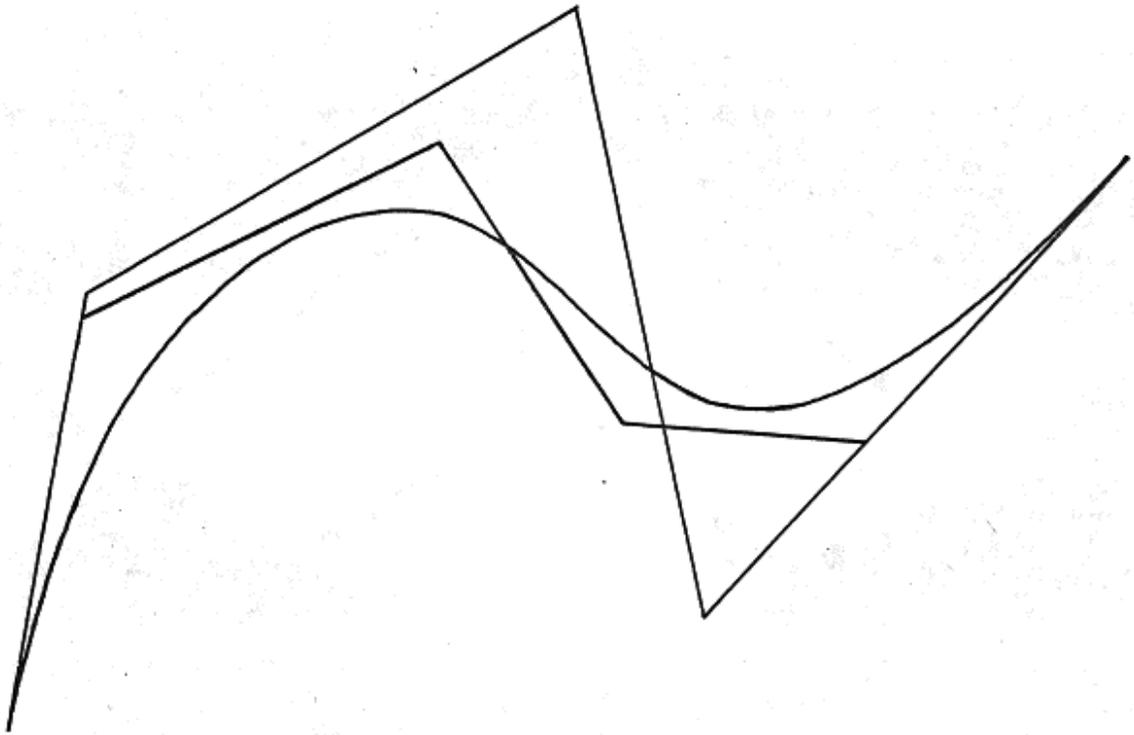


Figure 29 – Same cubic ( $M = 4$ ) B-spline curve defined by a 4-sided polygon and a 5-sided polygon.

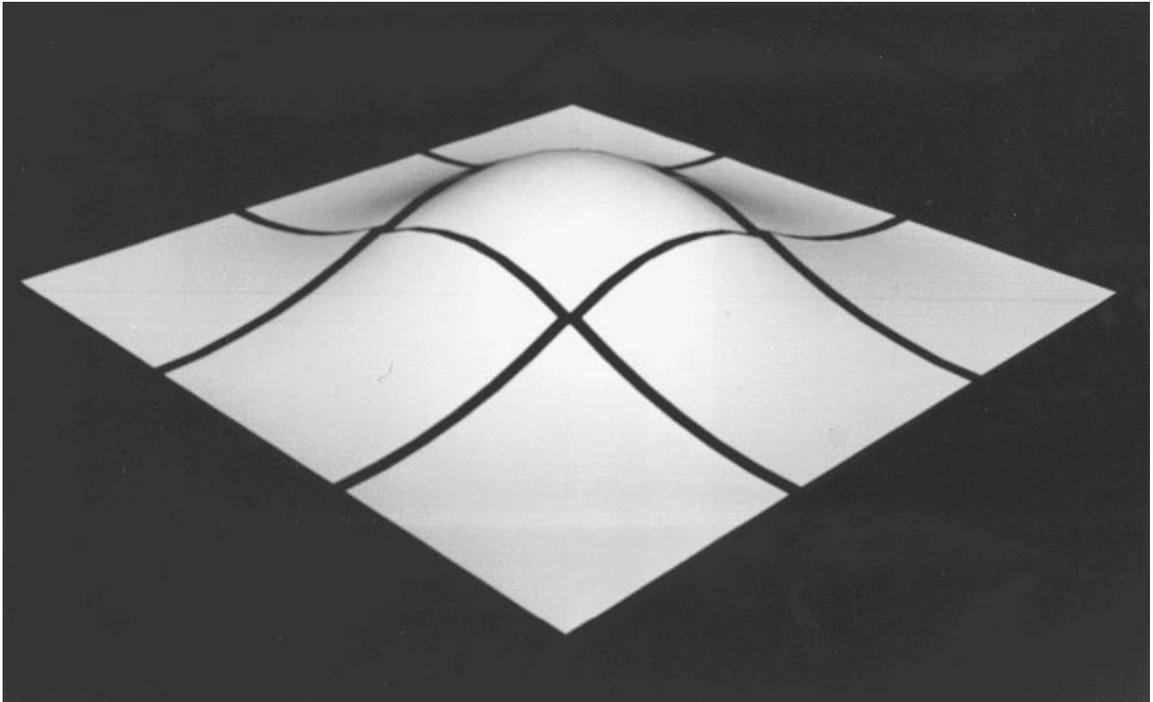


Figure 30 – Half-tone picture of a canonical biquadratic ( $L, M = 3$ ) B-spline basis function with darkened knot lines.

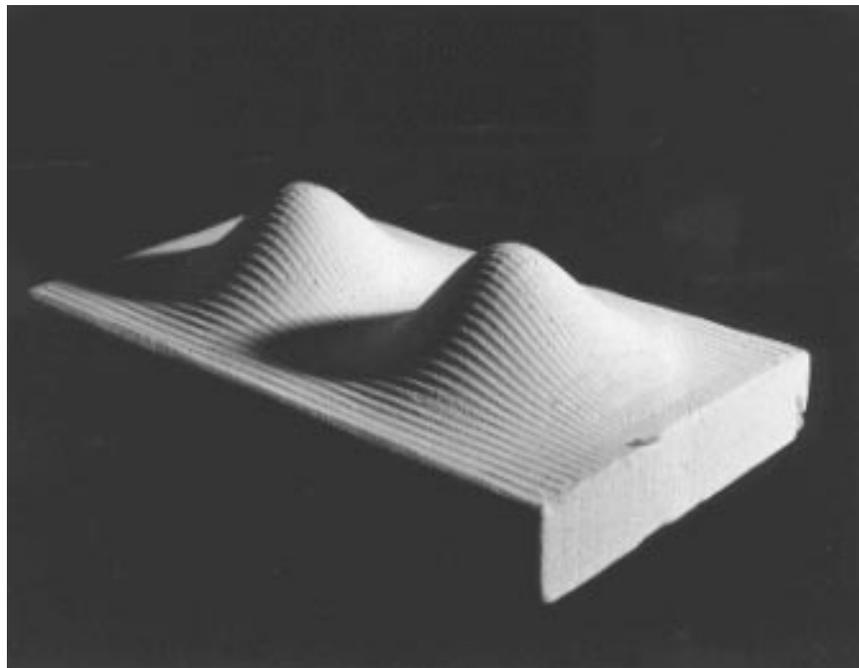
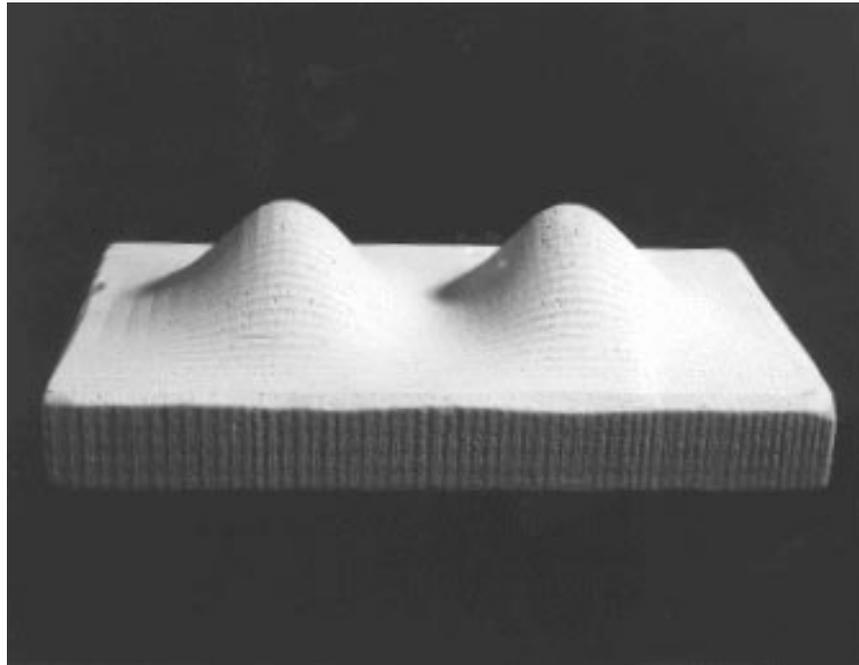


Figure 31 – Pictures of a simple B-spline surface cut by a numerically controlled milling machine.

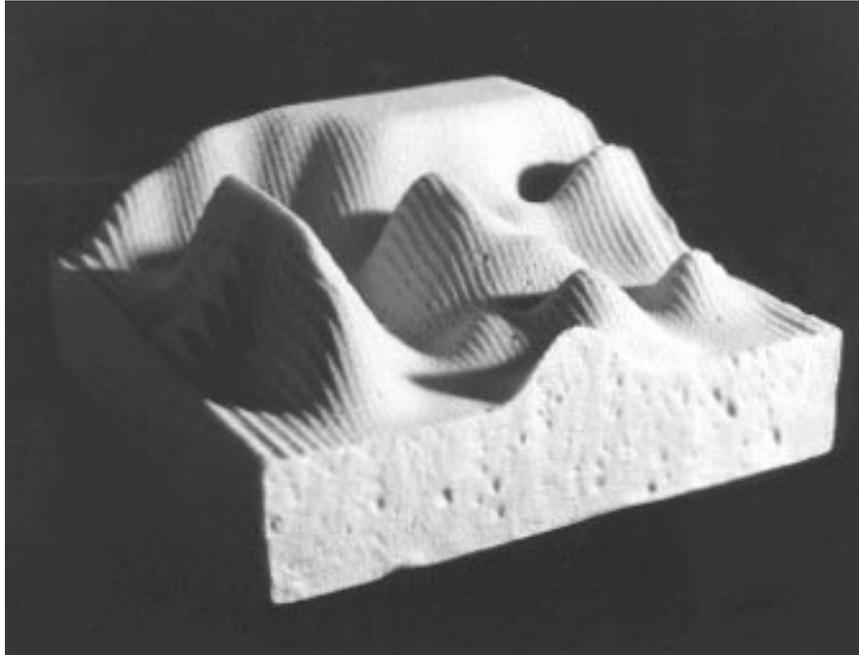


Figure 32 – Pictures of a complicated B-spline surface cut by a numerically controlled milling machine.

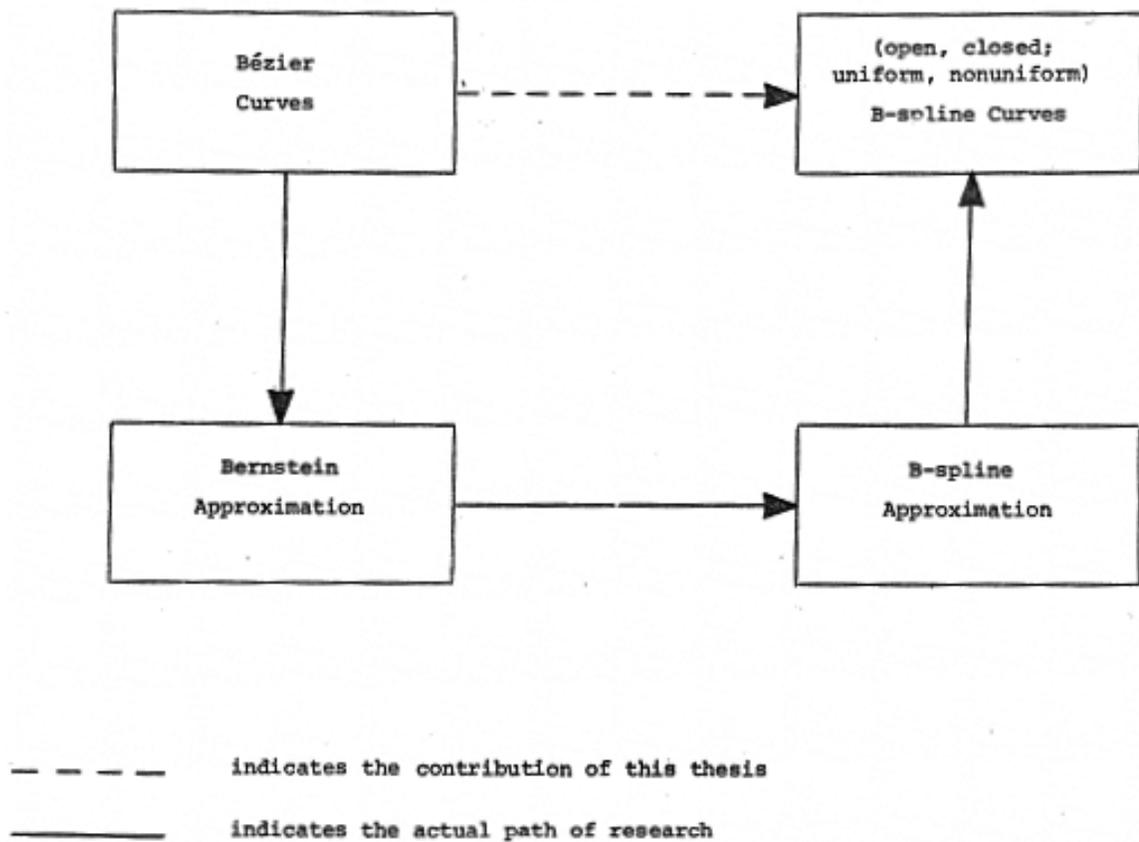


Figure 33 – Summary of results

## REFERENCES

- [1] Ahlberg, J. H. "Spline Approximation and Computer-Aided Design." *1970 Advances in Computers*. New York: Academic Press.
- [2] Ahlberg, J. H. "The Spline Approximation as an Engineering Tool." Study No. 5, Computer-Aided Engineering, Solid Mechanics Division, Univeristy of Waterloo, Waterloo, Ontario, Canada.
- [3] Ahlberg, J. H.; Nilson, E. N.; and Williams, J. N. "The Representation of Curves and Surfaces." Technical Report (unnumbered), Division of Applied Math, Brown University, Providence, Rhode Island, January 1970.
- [4] Ahlberg, J. H.; and Williams, J. N. "Cardinal Splines and Plane Curves." Technical Report (unnumbered), Division of Applied Math, Brown University, Providence, Rhode Island, August 1969.
- [5] Ahlberg, J. H.; Nilson, E. N.; and Walsh, J. L. *The Theory of Splines and Their Applications*. New York: Academic Press; 1967.
- [6] Bézier, P. *Numerical Control: Mathematics and Applications*. (translated by A. R. Forrest). London: John Wiley and Sons, 1972.
- [7] Bézier, P. "Procédé de Définition Numérique des Courbes et Surfaces Non Mathématiques; Système UNISURF." *Automatisme 13*, May 1968.
- [8] Bézier, P. "How Renault Uses Numerical Control for Car Body Design and Tooling." Paper SAE 680010, Society of Automobile Engineers, 1968
- [9] deBoor, C. On Calculating with B-splines. *J. Approx. Theory*, vol. 6 (1972), pp. 50–62.
- [10] deBoor, C. On Uniform Approximation by Splines. *J. Approx. Theory*, vol. 1 (1968), pp. 50–62.
- [11] Coons, S. A. *Surfaces for Computer-Aided Design of Space Forms*. Project MAC, MIT, 1964. Revised to MAX-TR-41, 1967.
- [12] Cox, M. G. "The Numerical Evaluation of B-splines." National Physical Laboratory (Teddington, England), DNAC 4, August 1971.
- [13] Curry, H. B.; and Schoenberg, I. J. "On Spline Distributions and Their Limits: The Pólya Distributions." Abstr., *Bull. AMS*, (1947), p. 1114.
- [14] Curry, H. B.; and Schoenberg, I. J. On Pólya Frequency Functions IV: The Fundamental Spline Functions and Their Limits. *J. D'Anal. Math.*, XVII (1966), pp.71–107.
- [15] Davis, P. J. *Interpolation and Approximation*. New York: Ginn-Blaisdell, 1963.
- [16] Feller, W. *Introduction to Probablility Theory and Its Applications*. Vol I (1957) and Vol II (1966), Mew York: John Wiley and Sons.
- [17] Ferguson, J. C. "Multi-Variable Curve Interpolation." The Boeing Company D2-22504, July 1962. Also in abbreviated from in *J. of ACM*, 11, (April, 1964).

- [18] Forrest, A. R. "Interactive Interpolation and Approximation by Bézier Polynomials." *Computer J.*, Vol. 15 (1972), pp. 71–79.
- [19] Forrest, A. R. "Coons' Surface and Multi-Variable Functional Interpolation." CAD Group Doc 38 (revised), Computer Laboratory, Cambridge University, (Dec 1971).
- [20] Forrest, A. R. Computer-aided design of three-dimensional objects: a survey. In *Proc. ACM/AICA Intl. Computing Symp.* Venice, 1972.
- [21] Forrest, A. R. "Computational geometry." *Proceedings of the Royal Society*, London, A321, pp. 187–195.
- [22] Gordon, W. J. "'Blending-Function' Methods of Bivariate and Multivariate Interpolation and Approximation." *SIAM J Numer. Anal.*, vol. 8, No. 1 (1971).
- [23] Gordon, W. J. "Free-Form Surface Interpolation Through Curve Networks." General Motors Research Publication GMR-921, October 1969.
- [24] Gordon, W. J. "Distributive Lattices and the Approximation of Multivariate Functions." *Proceedings of the Symposium on Approximation with Special Emphasis on Splines*, (I. J. Schoenberg, editor), Univ. of Wisconsin Press, 1969.
- [25] Gordon, W. J. and Riesenfeld, R. F. Bernstein-Bézier Methods for the Computer-Aided Design of Free-Form Curves and Surfaces. General Motors Research Laboratories, GMR 1176 (March 1972), to appear in *J. of ACM*.
- [26] Greville, T.N.E. "Introduction to Spline Functions", in *Theory and Applications of Spline Functions*. (Greville, ed.), Academic Press, 1969.
- [27] Keliski, R. P. and Rivlin, T. J. "Iterates of Bernstein Polynomials." *Pacific J. of Math*, vol. 21, No. 3 (1967) pp. 511–520.
- [28] Knapp, L. Master's thesis, Systems and Information Science, Syracuse University, 1972.
- [29] Lorentz, G. G. *Bernstein Polynomials*. University of Toronto Press (1953).
- [30] Marsden, M. "An Identity for Spline Functions and its Application to Variation Diminishing Spline Approximations." *J Approx. Theory*, 3:7–49, 1970.
- [31] Marsden, M. J. and Schoenberg, I. "On Variation Diminishing Spline Approximation Methods." *Mathematica* (Cluj), vol. 31 (1966)
- [32] Meyer, W. W. Private correspondence. General Motors Res. Labs., Math Dept. (1972).
- [33] Popoviciu, T. "Sur l'Approximation des Fonctions Convex d'Ordre Superieur." *Mathematica*, vol. 10 (1935), pp. 49–54.
- [34] Sabin, M. Private correspondence. British Aircraft Corp, Commercial Aircraft Div., Weybridge, England (1971).
- [35] Schoenberg, I. J. "Cardinal Interpolation and Spline Functions." *J. Approx. Theory*, vol. 2 (1969), pp. 167–206.
- [36] Schoenberg, I. J. "On Spline Functions." with Supplement by T.N.E Grevill, *Inequalities* (O. Shisha, editor), Academic Preses (1967), pp. 255–291.

- [37] Schoenberg, I. J. “On Variation Diminishing Approximation Methods” *On Numerical Approximation*. (R. E. Langer, editor). University of Wisconsin Press, 1959, pp. 249-274.
- [38] Schoenberg, I. J. “Smoothing Operators and Their Generating Functions.” In *Bull. Am. Math. Soc.*, vol. 59, (1953), pp. 199–230.
- [39] Schoenberg, I. J. “Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions.” *Quart. Appl. Math*, vol. 4 (1946), pp. 45–99; 112-141.
- [40] Schoenberg, I. J. “Notes on Spline Functions. The Limits of the Interpolating Periodic Spline Functions as Their Degree Tends to Infinity.” Mathematics Research Center, University of Wisconsin Madison, MRC Technical Summary Report #1219 (April 1972).
- [41] Schoenberg, I. J. and Whitney, A. “On Pôlya Frequency Functions III: The Positivity of Translation Determinants with an Application to the Interpolation Problem by Spline Curves.” *Trans. of AMS*, vol 74 (1953), pp. 246–259.

## BIOGRAPHICAL DATA

Name: Richard Riesenfeld

Date and Place of Birth: November 26, 1944, Milwaukee, Wisconsin

Elementary School: Switlik Elementary School, Jackson, New Jersey  
Graduated 1958

High School: Lakewood High School, Lakewood, New Jersey  
Graduated 1962

College: Princeton University, Princeton, New Jersey  
Graduated 1966, A.B.

Graduate Work: Syracuse University, Syracuse, New York  
Graduate Assistant in Department of Mathematics,  
Computing Center, Systems and Information Science  
Graduated 1969, M.A. in Mathematics

## Notes on the Preparation of This Document

Dr. Riesenfeld's thesis was originally written on a typewriter as was the custom of the time. This reformatted version, prepared to facilitate electronic distribution, was typeset with  $\text{\LaTeX}2_{\epsilon}$ . While this is a fundamentally different process than using a typewriter, I have tried to be faithful to the original document format. This section describes the general typesetting and the differences between this document and the original.

### General formatting and fonts

The formatting was based on the “book” document class with the “oneside” option. The left margin was fixed at 1.5 inches, and the right, top, and bottom margins set at 1 inch to match the original thesis. The original appears to have been typewritten using an “elite” typeface; this is a fixed-width font of 12 characters per inch. The vertical spacing was the standard typewriter spacing of 6 lines per inch, and the main body of the original thesis was double spaced. This amounts to a “baseline skip” of 12 points, and to most closely match the size of the original typeface, 12 point font was used for the main text. However, the Adobe Times font was used as the basic font and this differs significantly from the typeface in the original.

In the original, the chapter headings were written in all capitals. This has been retained in the present document, but they have also been typeset in “Large” boldface. The chapter numbers are in “Large” Roman font with the number in capital Roman numerals to adhere to the style of the original. Also, the chapter headings are centered, and the sections remain unnumbered. The front matter, including the title page, the table of contents, and the list of figures, were formatted to try to match the style and positioning of the original.

The underlining found in the original has been replaced with italics, with the exception of the names of theorems and such which have instead been typeset in boldface. Also, Latin phrases and abbreviations have been typeset in italics.

As in the original, page numbers have been left off the front matter and the first page of each chapter and the first page of the references section.

### Mathematical formulas

The mathematical formulas, including those in running text, have been formatted using  $\text{\TeX}$ 's capabilities—this, of course, was a primary reason to reformat the thesis. In particular, single variable names such as “ $s$ ” appear in italic font rather than in Roman. Also, the equation numbers, which appeared on the left in the original, have been moved to the right to conform to the modern convention. Some of the equations and equation notes have been repositioned slightly in order to improve alignment.

### Figures

All the figures were digitized in greyscale at an original resolution of 150 dpi, but the cropping process reduced the resolution on some of the images. The original sizes of the figures were maintained as close as possible but again due to cropping some size variation has been introduced. However, Figures 8, 25, 29, 30, and 33 were placed in landscape

format in the original, as were the corresponding captions, and these have been rotated and scaled down to be in portrait format in the present document. All the main figure captions were reformatted (in the style of the original) but the sub-figure captions were not.

### **Other remarks**

The final chapter of the original document ended on page 56, in this document it ends on page 47. There was therefore some compression of the text, due primarily to the slightly different font size, the fact that the Times font is proportionally spaced, and the more compact formatting of mathematical formulas. The figures, however, have the same pagination as the original.

The modern bibliography style has changed significantly since the original thesis was produced. The references in this document are as in the original. The only modification is the typeface, and the use of italics to replace underlining in book titles.

Finally, a few obvious and very minor typographical errors in the original thesis have been corrected (these are documented in the source files). Doing so seemed only fair, as the preparation of the present document inevitably has introduced other typographical errors.

M. Stark  
November 3, 2001