Preface

Interactive curve modeling techniques and their applications are extremely useful in a number of academic and industrial settings. Specifically, curve modeling plays a significant role in multidisciplinary problem solving. It is extremely useful in various situations like font design, designing objects, CAD/CAM, medical imaging and visualization, scientific data visualization, virtual reality, object recognition, etc. In particular, various problems like iris recognition, fingerprint recognition, signature recognition, etc. can also be intelligently solved and automated using curve techniques. In addition to its critical importance more recently, the curve modeling methods have also proven to be indispensable in a variety of modern industries, including computer vision, robotics, medical imaging, visualization, and even media.

This book aims to provide a valuable source that focuses on interdisciplinary methods and to add up-to-date methodologies in the area. It aims to provide the user community with a variety of techniques, applications, and systems necessary for various real-life problems in the areas such as font design, medical visualization, scientific data visualization, archaeology, toon rendering, virtual reality, body simulation, outline capture of images, object recognition, signature recognition, industrial applications, and many others.

Book Features

It aims to collect and disseminate information in various disciplines including computer graphics, image processing, computer vision, pattern recognition, artificial intelligence, soft computing, shape analysis and description, curve and surface fitting, scientific visualization, shape abstraction and modeling, intelligent CAD systems, computational geometry, reverse engineering, and levels of details for curves and surfaces. The major goal of this book is to stimulate views and provide a source where students, researchers, and practitioners can find the latest developments in the field of interactive curve modeling and its applications. The book provides classical and up-to-date theory and practice to get the problems solved in diverse areas of science and engineering.

All the chapters of the book will contribute toward curve modeling techniques, applications, and systems. The book will have the best possible utility for students, researchers, computer scientists, practicing engineers, and many others who seek classical and state-of-the-art techniques, applications, and systems with curve modeling. It will be an extremely useful book for undergraduate senior students as well as graduate students in the areas of computer science, engineering, and other computational sciences.

Suggested Course Outlines

This book is designed to have around fifteen chapters. These chapters will contribute toward interactive curve modeling techniques, applications, systems, and tools. The book is planned to have the best possible utility for researchers, computer scientists, practicing engineers, and many others who seek classical and state-of-the-art techniques and applications for computer graphics, vision, and imaging. It will also be equally and extremely useful for undergraduate senior students as well as graduate students in the areas of computer science. It is also beneficial to students in other disciplines including computer engineering, electrical engineering, mechanical engineering, and mathematics. The book is equally beneficial to researchers and practitioners in the industry and academia.

The book has been designed as a course book for undergraduate as well as graduate students in the area of computer science in particular. The main audience of the book are the communities related to the field of computer graphics, vision, and imaging. However, it can be useful for students in other disciplines like computer engineering, electrical engineering, mechanical engineering, mathematics, etc. The book is equally beneficial to researchers and practitioners in the industry. The book can formulate at least three courses as follows:

Course I. As an undergraduate course, at senior level, Chaps. 1–3, 8, 9, 11 (any two corner detectors), 12 (any two methods), 13, and 14 (one heuristic approach) will comprise a full length three credit hours course for a semester of 15 weeks. This course can be conducted with practical projects of reasonable weight.

Course II. As a graduate course consisting of Chaps. 1–4, 6–8 (self-study), 9, and 11–14 (one heuristic approach). This course should also have heavy projects for practical applications.

Course III. As a slightly different graduate course, if the undergraduate course described in Course I is considered to be a prerequisite. This course can be designed with Chaps. 4–7, 9 (using other curve schemes in the book but different than those in Chap. 9), 11–13 (just a quick review), 14, and 15. This course design can also consist of some state-of-the-art topics together with good weighted projects.

The researchers and practitioners can utilize the manuscript as a source as well as a reference book. Depending on their needs, they can study on pick and choose basis. They are also advised to study in their leisure time as it may prove to be fruitful to them.

Required Background

As such, it is not required to possess a specific qualification as a prerequisite to any of the undergraduate Course I or graduate courses II or III mentioned above. But, the user of this book is presumed to have some knowledge of computer programming together with some basic mathematical topics including analytic geometry, linear algebra, and calculus.

Acknowledgments

This manuscript has been prepared after a lot of struggle and efforts. Many graduate students and colleagues around the globe have assisted toward its completion. It is worthwhile to mention Asif Masood, Zulfiqar Habib, M. Zawwar Hussain, S. Ali Rizvi, M. Balah, M. Riyazuddin, Humayun Baig, S. Arshad Raza, Murtaza Ali Khan, Faisal AbdulRazzak, and M.A. Siddiqui. The author is thankful to all of them for their valuable efforts and advice. A lot of credit is also due to various experts who reviewed the chapters and provided helpful feedback.

It is not possible to forget my family here without whose help and support I would not have completed this work. Their love, support, and patience were tremendous throughout. In addition to thanking, I should also apologize for having taken much of their time during the conduct of my work.

The author is happy to acknowledge the support of King Fahd University of Petroleum and Minerals (KFUPM) toward the compilation of this book, against the Book Project #ICS/GRAPHICS/306. This book project was a main source of funding to this book. A partial funded support of KFUPM, through another Research Project #ICS/REVERSE ENG./312, also contributed toward a couple of chapters.

M. Sarfraz

2
Weighted Nu Splines

Abstract. *Weighted ν-splines are the composition of two spline methods, namely, weighted splines and ν-splines. These are the generalization of cubic spline method and are highly useful for CAD/CAM and various applications in computer graphics. Both—interpolatory and freeform—schemes are available in the literature. This chapter explains interpolatory weighted ν-splines together with a construction of its B-spline-like form. The design curves, constructed through B-spline-like form, possess all the ideal geometric properties such as partition of unity, convex hull, and variation diminishing. The splines provide not only a variety of very interesting shape control such as point and interval tensions but also, as a special case, recover the cubic spline method. In addition, these weighted ν-splines also provide, as special cases, the weighted splines and the ν-splines. The method for evaluating these splines is suggested by a transformation to Bézier form.*

2.1 Introduction

Designing of curves, especially those curves that are robust and easy to control and compute, has been one of the significant problems of computer graphics and geometric modeling. Specific applications including font designing, capturing hand-drawn images on computer screens, data visualization, and computer-supported cartooning are main motivations toward curve designing. In addition, various other applications in CAD/CAM/CAGD are also a good reason to study this topic. Many authors have worked in this direction. For brevity, the reader is referred to [1–22].

A cubic spline curve method is considered to be a considerably decent approach for designing applications in the area of computer graphics and geometric modeling. However, due to its various limitations, such as lack of freedom in shape control, a designer may not have much help. In this study, the weighted ν-spline method has been reviewed. This curve design method, in addition to enjoying the good features of cubic splines, possesses interesting shape design features too. It has two families of shape parameters working in such a way that one family of parameters is associated with intervals and the other with points. These parameters provide a variety of shape control such as point and interval tension. This is an interpolatory curve scheme, which utilizes a piecewise cubic function in its description. However, it is desired to extend this idea to freeform curves, which can enjoy all the ideal properties related to B-spline theory. This work is mainly concerned with developing such a theory.

Weighted splines [7] were discovered as a cubic spline method. The method provides a C^1 computationally simpler alternative to the exponential spline-under-tension [4, 13, 20]. Regarding shape characteristics, it has shape control parameters associated with each interval, which can be used to flatten or tighten the curve locally. Nu-splines [11,12] were discovered as another cubic spline method. It provides a GC^2 computationally simpler alternative to the exponential spline-under-tension [4,13,21]. Regarding shape characteristics, it has shape control parameters associated with each point, which can be used to tighten the curve both locally and globally. The ideas of weighted splines and Nu-splines were married together to formulate another spline called weighted Nu-spline [11, 12, 19, 22]. This curve design method covers the shape features of both of its counter parts and provides a C^1 computationally economical method.

B-splines are a useful and powerful tool for computer graphics and geometric modeling. They can be found frequently in the existing CAD/CAM (computer-aided design/computer -aided manufacturing) systems. They form a basis for the space of n th degree splines of continuity class C^{n-1}. Each B-spline is a non-negative n th-degree spline that is nonzero only on $n + 1$ intervals. The B-splines form a partition of unity, that is, they sum up to one. Curves generated by summing control points multiplied by the B-splines have some very desirable shape properties, including the local convex hull property and variation diminishing property.

It is desirable to generalize the idea of B-spline-like local basis functions for the classes of splines with shape parameters considered in the description of continuity. The first local basis for GC^2 splines was developed by Lewis [10]. In 1981, Barsky [1] generalized B-splines to Beta splines. These splines preserve the geometric smoothness of the design curve while allowing the continuity conditions on the spline functions at the knots to be varied by certain parameters, thus giving greater flexibility. Later, in 1984, Bartels and Beatty [2] developed local bases for Beta spline curves that are equivalent to Boehm's [3] Gamma splines. Foley [7], in 1987, constructed a B-spline-like basis for weighted splines; different weights were built into the basis functions so that the control point curve was a C^1 piecewise cubic with local control of interval tension.

In this work, a constructive approach has been adopted to build B-spline-like basis for cubic spline curves with the same continuity constraints as those for interpolatory weighted υ-splines. These are local basis functions with local support which have the property of being positive everywhere. The design curve, constructed through these functions, possesses all the ideal geometric properties like partition of unity, convex hull, and variation diminishing. This curve method provides not only a variety of very interesting shape control such as point and interval tensions, but also, as a special case, recovers the cubic B-spline curve method. In addition, it also provides B-spline-like design curves for weighted splines, ν-splines and weighted ν-splines. The method for evaluating these splines is suggested by a transformation to Bézier form.

The approach adopted in the construction of local basis for the weighted ν-splines is quite different from those adopted for different spline methods in [1–8, 15]. The way for evaluating the weighted ν-splines representation of a curve is suggested by a transformation to piecewise defined Bézier form. This form will also expedite a proof of the variation diminishing property for the Bézier representation.

This chapter is related the weighted ν-spline method [19, 22] explained in Section 2.2. It studies, in Section 2.3, a B-spline-like local basis for the weighted ν-spline. The design curve in Section 2.4 maintains the C^1 continuity of the weighted ν-splines. This description of freeform weighted ν-spline not only provides a variety of interesting shape control such as point and interval tensions but also, as a special case, recovers the cubic B-spline curve method. In addition, it also provides B-spline like design curves for weighted splines, ν-splines and weighted ν-splines. The method has been extended for the construction of surfaces in Section 2.5. Section 2.6 summarizes the chapter.

2.2 Some Spline Methods

This section gives a brief review of the cubic spline, weighted splines, ν-splines, and weighted ν-splines. Detailed description of the weighted ν-splines is given in Sections 2.3 and 2.4. Assume that we are given knot partition as $t_1 < t_2 < \ldots . < t_n$, and set of control points $F_1, F_2, \ldots, F_n$. Let us have the Followings:

$$\left.\begin{array}{ll}\text{Point tension factors:} & \nu_i \geq 0, i = 1, 2, \ldots, n, \\ \text{Interval weights:} & w_i > 0, i = 1, 2, \ldots, n.\end{array}\right\}, \tag{2.1}$$

Consider the piecewise cubic function:

$$p(t) \equiv p_i(t) = F_i(1-\theta)^3 + 3\theta(1-\theta)^2 V_i + 3\theta^2(1-\theta)W_i + F_{i+1}\theta^3, \tag{2.2}$$

where

$$\theta = \frac{t - t_i}{h_i}, \quad h_i = t_{i+1} - t_i, \tag{2.3}$$

and

$$V_i = F_i + \frac{h_i D_i}{3}, \quad W_i = F_{i+1} - \frac{h_i D_{i+1}}{3}. \tag{2.4}$$

It is obvious to see that the piecewise cubic function (2.2) holds the following interpolatory properties:

$$\left.\begin{array}{ll} p(t_i) = F_i, & p(t_{i+1}) = F_{i+1} \\ p^{(1)}(t_i) = D_i, & p^{(1)}(t_{i+1}) = D_{i+1}\end{array}\right\}, \tag{2.5}$$

where $p^{(1)}$ denotes first derivative with respect to t and D_i denote derivative values given at the knots t_i. This leads the piecewise cubic (2.2) to the piecewise Hermite interpolant $p \in C^1[t_1, t_n]$.

2.2.1 *Cubic Splines*

The cubic spline interpolant is a C^2 piecewise cubic function $p(t)$ that minimizes

$$V(f) = \sum_{i=1}^{n-1} \int_{t_i}^{t_{i+1}} \left[f''(t)\right]^2 dt,$$

subject to the interpolation conditions $f(t_i) = F_i$ for $i = 1, 2, \ldots, n$ and one of the following end conditions:

- **Type 1:** First derivative end conditions,
- **Type 2:** Natural end conditions, or
- **Type 3:** Periodic end conditions.

Given F_i and D_i for $i = 1, 2, \ldots, n$, there exists a unique C^2 piecewise cubic function $f(t)$ that satisfies $f(t_i) = F_i$ and $f'(t_i) = D_i$ for $i = 1, 2, \ldots, n$. The unknowns are the first derivative values, D_i, $i = 1, 2, \ldots, n$, and once they are computed, the function $f(t)$ can be easily evaluated using the standard piecewise cubic Hermite form explained in (2.2). Necessary and sufficient conditions for the function $f(t)$ to be the cubic spline interpolant are that its derivatives D_i's satisfy

$$\hat{c}_{i-1} D_{i-1} + \left(2\hat{c}_{i-1} + 2\hat{c}_i\right) D_i + \hat{c}_i D_{i+1} = \hat{b}_i \left(F_{i+1} - F_i\right) + \hat{b}_{i-1} \left(F_i - F_{i-1}\right),$$

for $i = 1, 2, \ldots, n$, where $\hat{c}_i = 1/h_i$, $\hat{b}_i = 3\hat{c}_i/h_i$. The above system of equations provides $(2.n - 2)$ equations for n unknowns, $D_1, \ldots, D_n$, and the additional equations come from the given end conditions. The equations for Type I first derivative end conditions are $D_1 = f'(t_1)$ and $D_n = f'(t_n)$. For Type II natural end conditions they are

$$2\hat{c}_1 D_1 + \hat{c}_1 D_2 = \hat{b}_1 \left(F_2 - F_1\right),$$

and

$$\hat{c}_{n-1} D_{n-1} + 2\hat{c}_{n-1} D_n = \hat{b}_{n-1}(F_n - F_{n-1}).$$

For Type 3 periodic end conditions, they are

$$\left(2\hat{c}_1 + 2\hat{c}_{n-1}\right) D_1 + \hat{c}_1 D_2 + \hat{c}_{n-1} D_{n-1} = \hat{b}_1(F_2 - F_1) + \hat{b}_{n-1}(F_n - F_{n-1}),$$

and $D_1 = D_n$. The linear system of equations that occurs when Type 1 or 2 end conditions are used is tridiagonal and diagonally dominant; thus it can be solved efficiently by using a standard tridiagonal system solver. Figure 2.1 is a cubic spline curve for a data shown as bullets.

2.2.2 *Weighted Splines*

The weighted spline interpolant is a C^1 piecewise cubic function $p(t)$ that minimizes

$$V(f) = \sum_{i=1}^{n-1} \omega_i \int_{t_i}^{t_{i+1}} \left[f''(t)\right]^2 dt,$$

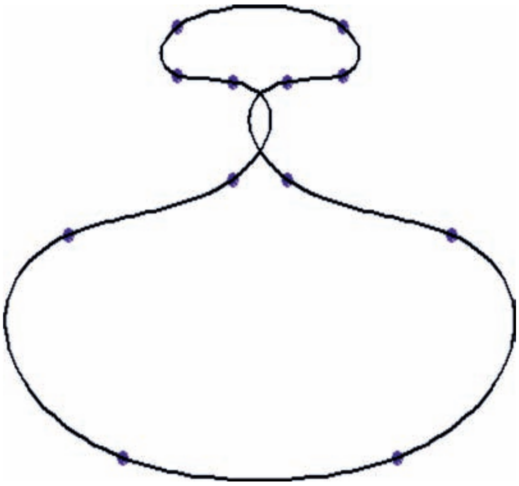

FIGURE 2.1. The default weighted ν-spline with periodic end conditions.

subject to the interpolation conditions $f(t_i) = F_i$ for $i = 1, 2, \ldots, n$ and one of the Type 1, Type 2, and Type 3 end conditions.

The ω_i's are termed as interval weights because they "tighten" the curve on the ith interval in the same way that they do for the weighted splines in [14]. If and all $\omega_i = q$, where q is some constant value, then the weighted spline equals the *cubic* spline as in Section 2.2.1.

The approach taken in [21] uses piecewise cubic Hermite basis functions to represent the weighted splines. Given F_i and D_i for $i = 1, 2, \ldots, n$, there exists a unique C^1 piecewise cubic function $f(t)$ that satisfies $f(t_i) = F_i$ and $f'(t_i) = D_i$ for $i = 1, 2, \ldots, n$. The unknowns are the first derivative values, D_i, $i = 1, 2, \ldots, n$, and once they are computed, the function $f(t)$ can be easily evaluated using the standard piecewise cubic Hermite form. Necessary and sufficient conditions for the function $p(t)$ to be the weighted spline interpolant are that its derivatives D_i satisfy

$$c_{i-1}D_{i-1} + (2c_{i-1} + 2c_i)\,D_i + c_i D_{i+1} = b_i\,(F_{i+1} - F_i) + b_{i-1}\,(F_i - F_{i-1})\,,$$

for $i = 1, 2, \ldots, n$, where $c_i = \omega_i/h_i,\ b_i = 3c_i/h_i$. The above system of equations provides $(n-2)$ equations for n unknowns, $D_1, \ldots, D_n$, and the additional equations come from the given end conditions. The equations for Type I first derivative end conditions are $D_1 = f'(t_1)$ and $D_n = f'(t_n)$. For Type II natural end conditions they are

$$2c_1 D_1 + c_1 D_2 = b_1\,(F_2 - F_1)\,,$$

and

$$c_{n-1}D_{n-1} + 2c_{n-1}D_n = b_{n-1}(F_n - F_{n-1}).$$

For Type 3 periodic end conditions, they are

$$2c_1 + 2c_{n-1}D_1 + c_1 D_2 + c_{n-1}D_{n-1} = b_1(F_2 - F_1) + b_{n-1}(F_n - F_{n-1}),$$

and $D_1 = D_n$. The linear system of equations that occurs when Type 1 or 2 end conditions are used is tridiagonal and diagonally dominant; thus it can be solved efficiently by using a standard tridiagonal system solver.

2.2.3 *Nu Splines*

The ν-spline interpolant is a GC^2 piecewise cubic function $p(t)$ that minimizes

$$V(f) = \sum_{i=1}^{n} \nu_i \left[f'(t_i) \right]^2,$$

subject to the interpolation conditions $f(t_i) = F_i$ for $i = 1, 2, \ldots, n$ and one of the Type 1, Type 2, and Type 3 end conditions.

The ν_i are termed point tension factors because they "tighten" a parametric curve at the ith point in the same way that they do for the ν-splines in [11, 12]. If $\nu_i = 0$ ν-spline equals the *cubic* spline in [11, 12].

The approach taken in [11, 12, 19, 21] uses piecewise cubic Hermite basis functions to represent the ν-splines. Given F_i and D_i for $i = 1, 2, \ldots, n$, there exists a unique GC^2 piecewise cubic function $f(t)$ that satisfies $f(t_i) = F_i$ and $f'(t_i) = D_i$ for $i = 1, 2, \ldots, n$. The unknowns are the first derivative values, D_i, $i = 1, 2, \ldots, n$, and once they are computed, the function $f(t)$ can be easily evaluated using the standard piecewise cubic Hermite form. Necessary and sufficient conditions for the function $p(t)$ to be the ν-spline interpolant are that its derivatives D_i satisfy

$$\breve{c}_{i-1} D_{i-1} + \left(\frac{1}{2} \nu_i + 2\breve{c}_{i-1} + 2\breve{c}_i \right) D_i + \breve{c}_i D_{i+1} = \breve{b}_i \left(F_{i+1} - F_i \right) + \breve{b}_{i-1} \left(F_i - F_{i-1} \right),$$

for $i = 1, 2, \ldots, n$, where $\breve{c}_i = 1/h_i$, $\breve{b}_i = 3\breve{c}_i/h_i$. The above system of equations provides $(n-2)$ equations for n unknowns, $D_1, \ldots, D_n$, and the additional equations come from the given end conditions. The equations for Type I first derivative end conditions are $D_1 = f'(t_1)$ and $D_n = f'(t_n)$. For Type II natural end conditions they are

$$\left(\frac{1}{2} \nu_1 + 2c_1 \right) D_1 + c_1 D_2 = b_1 \left(F_2 - F_1 \right),$$

and

$$c_{n-1} D_{n-1} + \left(\frac{1}{2} \nu_n + 2c_{n-1} \right) D_n = b_{n-1}(F_n - F_{n-1}).$$

For Type 3 periodic end conditions, they are

$$\left(\frac{1}{2} \nu_1 + \frac{1}{2} \nu_n + 2c_1 + 2c_{n-1} \right) D_1 + c_1 D_2 + c_{n-1} D_{n-1}$$
$$= b_1(F_2 - F_1) + b_{n-1}(F_n - F_{n-1}),$$

and $D_1 = D_n$. The linear system of equations that occurs when Type 1 or 2 end conditions are used is tridiagonal and diagonally dominant; thus it can be solved efficiently by using a standard tridiagonal system solver.

2.2.4 *Weighted Nu Splines*

The weighted ν-spline interpolant is a C^1 piecewise cubic function $p(t)$ that minimizes

$$V(f) = \sum_{i=1}^{n-1} \omega_i \int_{t_i}^{t_{i+1}} \left[f''(t)\right]^2 dt + \sum_{i=1}^{n} \nu_i \left[f'(t_i)\right]^2,$$

subject to the interpolation conditions $f(t_i) = F_i$ for $i = 1, 2, \ldots\ldots, n$ and one of the Type 1, Type 2, Type 3 end conditions. This is the marriage of Weighted splines and Nu splines which can be recovered as special cases discussed later in this chapter.

The ν_i are termed point tension factors because they 'tighten' a parametric curve at the ith point in the same way that they do for the ν-splines in [11, 12]. The w_i are termed interval weights because they 'tighten' the curve on the ith interval in the same way that they do for the weighted splines in [14]. If $\nu_i = 0$ and all $\omega_i = q$, where q is some constant value, then the weighted ν-spline equals the ν-spline in [11, 12] with tension factors ν_i/q. If all $\nu_i = 0$, then it equals the weighted spline given in [14].

The approach taken in [8] uses piecewise cubic Hermite basis functions to represent the weighted ν-splines. Given F_i and D_i for $i = 1, 2, \ldots, n$, there exists a unique C^1 piecewise cubic function $f(t)$ that satisfies $f(t_i) = F_i$ and $f'(t_i) = D_i$ for $i = 1, 2, \ldots, n$. The unknowns are the first derivative values, D_i, $i = 1, 2, \ldots, n$, and once they are computed, the function $f(t)$ can be easily evaluated using the standard piecewise cubic Hermite form. Necessary and sufficient conditions for the function $p(t)$ to be the weighted ν-spline interpolant are that its derivatives D_i satisfy

$$\begin{aligned} c_{i-1}D_{i-1} + \left(\frac{1}{2}\nu_i + 2c_{i-1} + 2c_i\right) D_i + c_i D_{i+1} \\ = b_i\left(F_{i+1} - F_i\right) + b_{i-1}\left(F_i - F_{i-1}\right), \end{aligned} \tag{2.6}$$

for $i = 1, 2, \ldots, n$. The above system of equations provides $(n-2)$ equations for n unknowns, $D_1, \ldots, D_n$, and the additional equations come from the given end conditions. The equations for Type I first derivative end conditions are $D_1 = f'(t_1)$ and $D_n = f'(t_n)$. For Type II natural end conditions they are

$$\left(\frac{1}{2}\nu_1 + 2c_1\right) D_1 + c_1 D_2 = b_1\left(F_2 - F_1\right),$$

and

$$c_{n-1}D_{n-1} + \left(\frac{1}{2}\nu_n + 2c_{n-1}\right) D_n = b_{n-1}(F_n - F_{n-1}).$$

For Type 3 periodic end conditions, they are

$$\begin{aligned} &\left(\frac{1}{2}\nu_1 + \frac{1}{2}\nu_n + 2c_1 + 2c_{n-1}\right) D_1 + c_1 D_2 + c_{n-1}D_{n-1} \\ &= b_1(F_2 - F_1) + b_{n-1}(F_n - F_{n-1}), \end{aligned}$$

and $D_1 = D_n$. The linear system of equations that occurs when Type 1 or 2 end conditions are used is tridiagonal and diagonally dominant; thus it can be solved efficiently by using a standard tridiagonal system solver.

The weighted ν-spline can be computed by solving for D_i's. This can be done by re-writing the system of equations in (2.10) as follows:

$$c_{i-1}D_{i-1} + \left(\frac{\nu_i}{2} + 2c_{i-1} + 2c_i\right) D_i + c_i D_{i+1} = 3c_i \Delta_i + 3c_{i-1}\Delta_{i-1}, \qquad (2.7)$$

where

$$\Delta_i = (F_{i+1} - F_i)/h_i.$$

for $i = 2, \ldots, n-1$. For given appropriate end conditions (Type 1, Type 2, or Type 3), this system of equations is a tridiagonal linear system. This is also diagonally dominant for the following constraints on the shape parameters as in (2.1), and hence has a unique solution for D_i's. As far as the computation method is concerned, it is much more economical to adopt the LU-decomposition method to solve the tridiagonal system. Therefore, the above discussion can be concluded in the following:

Theorem 2.1. *For the shape parameter constraints (2.1), the spline solution of the weighted ν-spline exists and is unique.*

Remark 2.1. Each component of the parametric weighted ν-spline is a C^1 function in general, but it has second-order geometric continuity at t_i if $\omega_{i-1} = \omega_i$ and the tangent vector at t_i is non zero and it is C^2 at t_i if $\omega_{i-1} = \omega_i$ and $\nu_i = 0$.

2.2.5 *Demonstration*

Figure 2.1 is the parametric weighted ν-spline interpolant to the points denoted by circles using periodic end conditions. In Figure 2.2, interval weight,ω_i, of 30 is used in the base interval, while point tension factors, ν_i of 10 are used on the four vertices defining the "neck." The rest of the parameters are taken as $\omega_i = 1$ and $\nu_i = 0$.

2.3 Freeform Weighted Nu Spline

This section is devoted to constructing the freeform weighted Nu spline which has inherent properties of B-spline curves. This formulation is possible through the construction of local support basis B_i's to compute the cubic weighted ν-spline $p(t)$ satisfying the following constraints:

$$\begin{bmatrix} p(t_{i+}) \\ p^{(1)}(t_{i+}) \\ p^{(2)}(t_{i+}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{\nu_i}{\omega_i} & \frac{\omega_{i-1}}{\omega_i} \end{bmatrix} \begin{bmatrix} p(t_{i-}) \\ p^{(1)}(t_{i-}) \\ p^{(2)}(t_{i-}) \end{bmatrix}. \qquad (2.8)$$

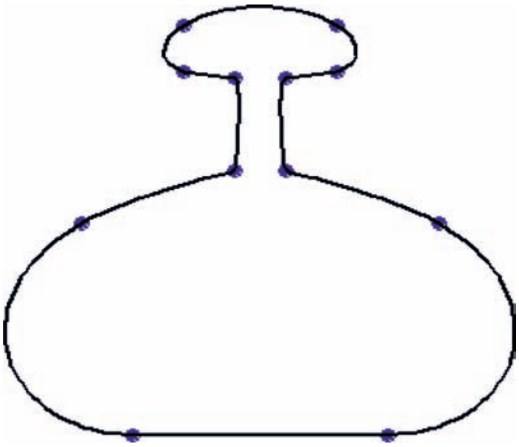

FIGURE 2.2. The weighted ν-spline with periodic end conditions using $\omega_i = 30$ on the base interval, $\omega_i = 1$ otherwise, $\nu_i = 10$ on the four vertices defining the "neck," and $\nu_i = 0$ otherwise.

2.3.1 *Local Support Basis*

For the purpose of the analysis, let additional knots be introduced outside the knot partition $t_1 < t_2 < \ldots < t_n$ of the interval $[t_1, t_n]$, defined by:

$$t_{-2} < t_{-1} < t_0 < t_1 \text{ and } t_n < t_{n+1} < t_{n+2} < t_{n+3}. \tag{2.9}$$

Let

$$a_i = 1/c_i, \tag{2.10}$$

and ϕ_i be cubic weighted ν-spline:

$$\phi_i\,(t) = \begin{cases} 0, & t \le t_{i-2}, \\ 1, & t \ge t_{i+1}. \end{cases} \tag{2.11}$$

Imposing weighted ν-spline constraints (2.8), we have:

$$\phi_i(t_{i-1}) = \frac{h_{i-2}}{3}\phi_i^{(1)}(t_{i-1}),$$

$$\phi_i(t_i) = 1 - \frac{h_i}{3}\phi_i^{(1)}(t_i),$$

$$\phi_i^{(1)}(t_{i-1}) = \frac{A_i}{C_i},$$

and

$$\phi_i^{(1)}(t_i) = \frac{B_i}{C_i},$$

where, if $d_i = \frac{1}{2}a_i a_{i-1}\nu_i + a_{i-1} + a_i$, then

$$\begin{aligned} A_i &= \frac{3a_{i-2}}{h_{i-1}} d_i, \\ B_i &= \frac{3a_i}{h_{i-1}} d_{i-1}, \\ C_i = d_i d_{i-1} + \frac{a_i}{h_{i-1}}&(h_{i-1} + h_i)d_{i-1} + \frac{a_{i-2}}{h_{i-1}}(h_{i-1} + h_{i-2})d_i. \end{aligned}$$

Let

$$\begin{gathered} D_i = h_{i-1}d_i d_{i-1} + a_i(h_{i-1} + h_i)d_{i-1} + a_{i-2}(h_{i-1} + h_{i-2})d_i, \\ \mu_i = \phi_{i+1}(t_i), \lambda_i = 1 - \phi_i(t_i), \\ \hat{\mu}_i = \phi_{i+1}^{(1)}(t_i), \hat{\lambda}_i = \phi_i^{(1)}(t_i), \end{gathered}$$

Then

$$\begin{gathered} \hat{\lambda}_i = \frac{3a_i d_{i-1}}{D_i}, \hat{\mu}_i = \frac{3a_{i-1}d_{i+1}}{D_{i+1}}, \\ \mu_i = \frac{h_{i-1}}{3}\hat{\mu}_i, \lambda_i = \frac{h_i}{3}\hat{\lambda}_i, \end{gathered}$$

and hence

$$0 \le \mu_i \le 1,\ 0 \le \lambda_i \le 1 \quad \text{and} \quad 0 \le \mu_i + \lambda_i \le 1.$$

Now define

$$B_i(t) = \phi_i(t) - \phi_{i+1}(t).$$

Then B_i has the local support (t_{i-2}, t_{i+2}) and an explicit representation of B_j on any interval (t_i, t_{i+1}) (in particular, for $i = j-2, j-1, j, j+1$) can be calculated as:

$$\begin{aligned} B_j(t) = {} & (1-\theta)^3 B_j(t_i) + \theta(1-\theta)^2(3B_j(t_i)h_i B_j^{(1)}(t_i)) \\ & + \theta^2(1-\theta)(3B_j(t_{i+1}) - h_i B_j^{(1)}(t_{i+1})) + \theta^3 B_j(t_{i+1}), \end{aligned} \tag{2.12}$$

where

$$B_j(t_i) = B_j^{(1)}(t_i) = 0 \text{ for } i \ne j-1, j, j+1,$$

and

$$\left.\begin{aligned} B_j(t_{j-1}) &= \mu_{j-1}, & B_j^{(1)}(t_{j-1}) &= \hat{\mu}_{j-1}, \\ B_j(t_j) &= 1 - \lambda_j - \mu_j, & B_j^{(1)}(t_j) &= \hat{\lambda}_j - \hat{\mu}_j, \\ B_j(t_{j+1}) &= \lambda_{j+1}, & B_j^{(1)}(t_{j+1}) &= -\hat{\lambda}_{j+1}. \end{aligned}\right\} \tag{2.13}$$

Careful examination of the Bézier vertices of $B_j(t)$ in (2.12) shows these to be non-negative for ν_i, ω_i satisfying (2.7) and thus $B_j(t) \ge 0,\ \forall t$. This leads to the following:

Proposition 2.1. The local support basis functions (2.12) are such that the following properties hold:

(i) (Local support) $B_j(t) = 0\, \textit{for}\ t \notin (t_{j-2}, t_{j+2})$,

(ii) (Partition of unity) $\sum\limits_{j=-1}^{n+1} B_j(t) = 1\, \textit{for}\ t \in [t_1, t_n]$,

(iii) (Positivity) $B_j(t) \geq 0$ for all t.

2.3.2 *Design Curve*

Now, we need a convenient method to compute the curve representation. It is desired to apply the above developed local basis functions to develop a freeform weighted ν-spline curve as follows:

$$P(t) = \sum_{j=-1}^{n+1} B_j(t) P_j, t \in [t_1, t_n], \tag{2.14}$$

where $P_j \in R^N$, $j = 0, 1, \ldots, n+1$, define the control points of the representation. By the local support property,

$$P(t) = \sum_{j=i-1}^{i+2} B_j(t) P_j,\ t \in [t_i, t_{i+1}), i = 0, \ldots, n-1.$$

Substitution of (2.12), $t \in [t_i, t_{i+1})$, then gives the piecewise defined Bézier representation

$$P(t) \equiv P_i(t) = F_i(1-\theta)^3 + 3\theta(1-\theta)^2 V_i + 3\theta^2(1-\theta) W_i + F_{i+1}\theta^3, \tag{2.15}$$

where

$$\left.\begin{aligned} F_i &= \lambda_i P_{i-1} + (1-\lambda_i-\mu_i)P_i + \mu_i P_{i+1}, \\ V_i &= (1-\alpha_i)P_i + \alpha_i P_{i+1}, \\ W_i &= \beta_i P_i + (1-\beta_i)P_{i+1}, \end{aligned}\right\} \tag{2.16}$$

with

$$\begin{aligned} \alpha_i &= \mu_i + h_i \hat{\mu}_i/3 = \frac{\hat{\mu}_i}{3}(h_{i-1} + h_i), \\ \beta_i &= \lambda_{i+1} + h_i \overset{\wedge}{\lambda_i} = \frac{\hat{\lambda}_{i+1}}{3}(h_i + h_{i+1}). \end{aligned}$$

This transformation to Bézier form is very convenient for computational purposes and also leads to the following:

Proposition 2.2. (Variation Diminishing Property) The weighted ν-spline curve $P(t)$, $t \in [t_0, t_n]$, defined by (2.14), crosses any (hyper) plane of dimension $N-1$ no more times than it crosses the "control polygon" joining the control points $P_{-1}, P_0, \ldots, P_{n+1}$.

Proof. Following the arguments of positivity in the previous proposition, it is straightforward that $0 \leq \alpha_i \leq 1$, $0 \leq \beta_i \leq 1$, and $0 \leq \alpha_i + \beta_i \leq 1$. Thus, V_i and W_i lie on the line segment joining P_i and P_{i+1}, where V_i is before W_i. It can also be simply noted that

$$F_i = (1 - \gamma_i)W_{i-1} + \gamma_i V_i, \tag{2.17}$$

where

$$0 < \gamma_i = \frac{h_{i-1}}{h_{i-1} + h_i} < 1.$$

Thus, the control polygon of the piecewise defined Bézier representation is obtained by corner cutting of the weighted ν-spline control polygon. Since the piecewise defined Bézier representation is variation diminishing, it follows that weighted ν-spline representation is variation diminishing.

2.3.3 *Shape Control*

The shape parameters, defined in (2.7), can be used to control the local or global shape of the design curve. To analyze such behaviors, the explicit form on $(2.t_i, t_{i+1})$ of the weighted ν-spline design curve (2.14) can be expressed as:

$$P(t) = l_i(t) + e_i(t), \tag{2.18}$$

where

$$l_i(t) = (1 - \theta)F_i + \theta F_{i+1}, \tag{2.19}$$

and

$$\begin{aligned} e_i(t) \, &= \theta(1-\theta)\left\{\left[(F_{i+1} - F_i) - h_i P^{(1)}(t_i)\right](\theta - 1)\right. \\ &\quad \left. + \left[(F_{i+1} - F_i) - h_i P^{(1)}(t_{i+1})\right]\theta\right\}. \end{aligned} \tag{2.20}$$

Proposition 2.3. Let $\omega_i = \omega \geq 1$, and $\nu_i = 0, \forall i$ are all bounded then the weighted ν-spline design curve is straightway the standard cubic spline.

Proof. It follows from the last constraint of relation (2.8).

Proposition 2.4. (Global Tension) Let $\omega_i \geq 1$, $\forall i$, be bounded and $\nu_i \geq \nu$ then the weighted ν-spline curve (2.14) converges uniformly to the control polygon $P_0, \ldots, P_n$ as $\nu \to \infty$.

Proof. Let $\nu_i = \nu, \forall i$ then from (2.1)

$$\lim_{\nu \to \infty} P^{(1)}(t_i) = 0. \tag{2.21}$$

Moreover

$$\lim_{\nu \to \infty} \hat{\mu}_i = 0 = \lim_{\nu \to \infty} \hat{\lambda}_i, \ \forall i.$$

This implies the following:

$$\lim_{\nu\to\infty} F_i = P_i, \ \forall i. \tag{2.22}$$

More generally, for $\nu_i \geq \nu \geq 0$, it can be shown that

$$\max_i |\hat{\lambda}_i| \leq r(\nu),$$

and

$$\max_i |\hat{\mu}_i| \leq s(\nu),$$

where

$$\lim_{\nu\to\infty} r(\nu) = 0 = \lim_{\nu\to\infty} s(\nu),$$

and again (2.21) and (2.22) hold. Hence the result.

Proposition 2.5. [Local Tension] Consider an interval $[t_k, t_{k+1}]$ for a fixed k. Then on $[t_k, t_{k+1}]$ weighted ν-spline curve converges uniformly to a line segment of the line $P_k P_{k+1}$ as $w_k \to \infty$ where ω_{k-1} and ν_k are bounded.

Proof. Careful examination shows

$$\lim_{\omega\to\infty} \mu_k = \frac{h_{k-1}}{(3h_k + h_{k-1} + h_{k+1})} = \hat{\alpha}_k(\text{say})$$
$$\lim_{\omega\to\infty} \mu_{k+1} = 0$$
$$\lim_{\omega\to\infty} \lambda_k = 0$$
$$\lim_{\omega\to\infty} \lambda_{k+1} = \frac{h_{k+1}}{(3h_k + h_{k-1} + h_{k+1})} = \hat{\beta}_k(\text{say})$$

This implies the following:

$$\lim_{\omega\to\infty} F_k = \left(1 - \hat{\alpha}_k\right) P_k + \hat{\alpha}_k P_{k+1} = \hat{F}_k(\text{say})$$

and

$$\lim_{\omega\to\infty} F_{k+1} = \hat{\beta}_k P_k + \left(1 - \hat{\beta}_k\right) P_{k+1} = \hat{F}_{k+1}(\text{say})$$

Obviously $\hat{F}_k$ and $\hat{F}_{k+1}$ lie on $\overline{P_k P_{k+1}}$ and $\hat{F}_k$ is before $\hat{F}_{k+1}$ as $\hat{\alpha}_k < (1 - \hat{\beta}_k)$. Also

$$\lim_{\omega\to\infty} (F_{k+1} - F_k) = \lim_{\omega\to\infty} h_k P^{(1)}(t_k) = \lim_{\omega\to\infty} h_k P^{(1)}(t_{k+1}) = \frac{3h_k(P_{k+1} - P_k)}{(3h_k + h_{k-1} + h_{k+1})}$$

Hence from (2.18), (2.19), (2.20) if $P(t) = P_k(t)$ for $t \in (t_k, t_{k+1})$, then

$$\lim_{\omega\to\infty} P_k(t) = (1 - \theta)\hat{F}_k + \theta \hat{F}_k.$$

Proposition 2.6. (Local Tension) Consider an interval as in Proposition 5. Then on $[t_k, t_{k+1}]$, the weighted ν-spline converges uniformly to the linear interpolant $l_i(t)$ as both $\nu_k, \nu_{k+1\rightarrow} \propto$, where $\omega_{k-1}, \omega_k, \omega_{k+1}$ are bounded.

Proof. It can be noted that

$$\begin{aligned} \lim \mu_k &= \lim \mu_{k+1} = 0, \\ \lim \lambda_k &= \lim \lambda_{k+1} = 0, \end{aligned}$$

and

$$\lim P^{(1)}(t_k) = \lim P^{(1)}(t_{k+1}) = 0.$$

This gives the desired result.

2.3.4 *Demonstration*

The tension behavior of the weighted ν-spline is illustrated by the following simple examples for data set in R^2. Unless otherwise stated, in all the figures, the parameter ν_i will be assumed as zero $\forall i$ and the parameters ω_i as 1 for all i.

Figure 2.3 is the default curve, which is a cubic spline for $\nu_i = 0$, and $\omega_i = 1$, for all i. The control polygon, together with the control points, is also shown in the figure. Figure 2.4 shows the effect of a progressive increase in the interval tension in the base of the figure. The top, middle, and bottom curves have been demonstrated for $\omega = 1$, 10, and 100, respectively. The effect of the high-tension parameters is clearly seen in the corresponding interval in the base of the figure. Figure 2.5 shows the effect of a progressive increase in point tension behavior locally at two opposite points of the figure. The top, middle, and bottom curves have been demonstrated for $\nu = 0$, 10, and 100, respectively. The effect of the high-tension parameters is clearly seen at the corresponding points in the figure.

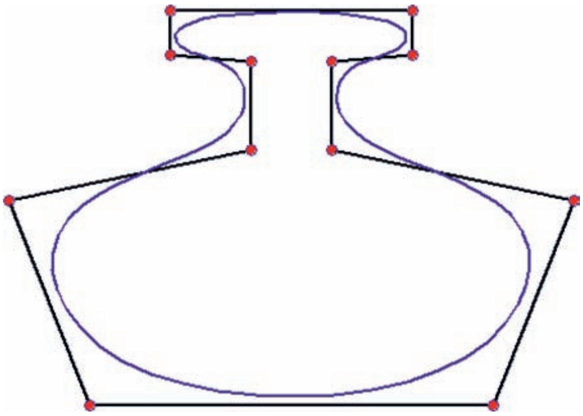

FIGURE 2.3. The default weighted Nu spline.

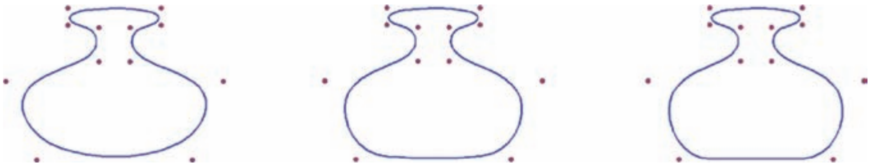

FIGURE 2.4. The weighted Nu spline with interval tension at the base with ω values as 1 (left curve), 10 (middle curve), 100 (right curve).

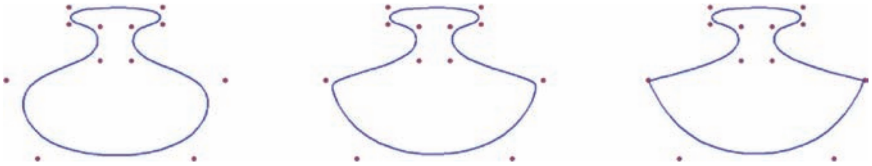

FIGURE 2.5. The weighted Nu spline with corner tension at two opposite points with ν values as 0 (left curve), 10 (middle curve), 100 (right curve).

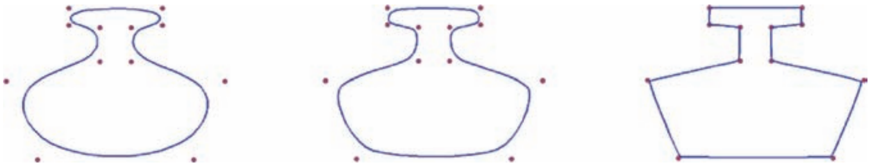

FIGURE 2.6. The weighted Nu spline with global tension $\nu = 1$ (top curve), $\nu = 5$ (middle curve), $\nu = 100$ (bottom curve).

Figures 2.6 illustrates the effect of progressively increasing the values of the point tension parameters ν_i's $= 0$, 5, and 100, for the top, middle, and bottom curves, respectively, at all the points of the figure. This is the global tension effect due to progressive increase.

Figure 2.7 demonstrates an important observation about the negative values of the shape parameters. The global values of the interval shape parameters ω's will not make any effect to the picture. However, the local values do influence the picture. The curve bulges inside for negative values $\omega = 0, -3, -4, -5, -25$, and -100. It can be noted (row-wise from left to right) that lower negative values make the curve bulge more inside, but higher negative values again start making the curve tensed in the interval.

Behavior of the negative ν values can be seen in Figure 2.8. It illustrates the effect of progressive negative increase in the values of the point tension parameters ν_i's $= 0, -1, -5, -25, -100$, and -1000. It can be seen (row-wise from left to right) that Lower negative values make the curve bulge inside so much so the curve starts looping with the negative increase. However, it again starts getting tensed after attaining certain values. Ultimately, higher negative values make the curve tensed to converge to the control polygon.

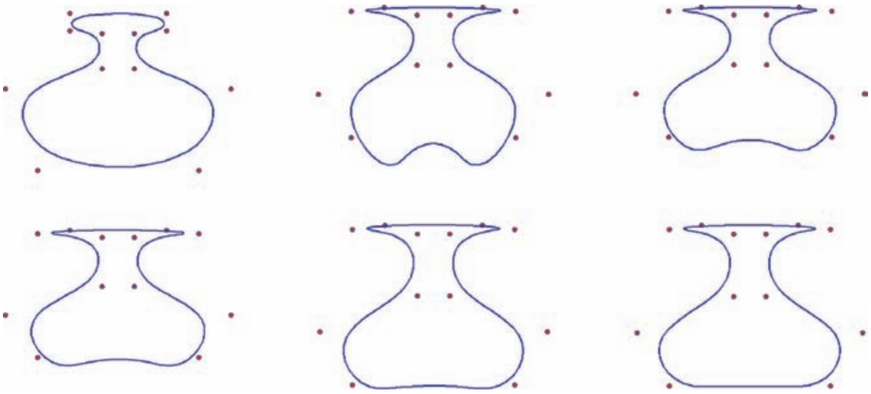

FIGURE 2.7. The weighted Nu spline curves (row vise from left to right) with negative global tension ω values 0, −3, −4, −5, −25, 100.

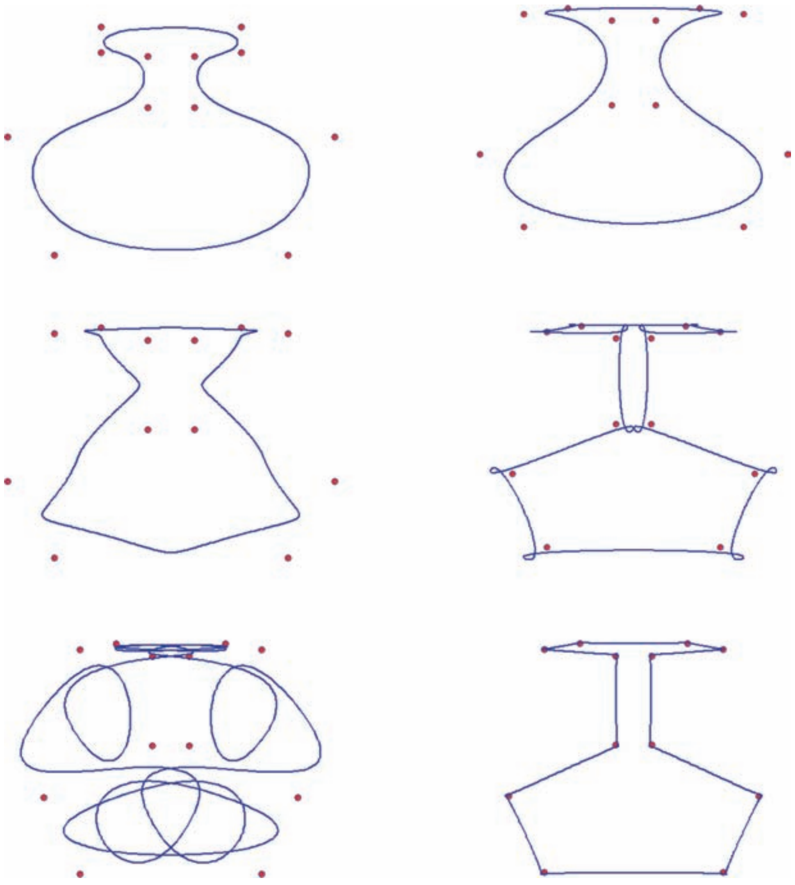

FIGURE 2.8. The weighted Nu spline curves (row vise from left to right) with negative global tension ν values 0, −1, −5, −25, 100, and 1000.

2.3.5 *Advantages and Features*

The method has various advantages and features as follows:

- It enjoys the good features of cubic splines.
- It enjoys all the standard geometric properties of B-splines.
- The method is geometrically smooth.
- It recovers the cubic B-spline method as a special case.
- It recovers the weighted spline method as a special case.
- It recovers the Nu-spline method as a special case.
- It possesses interested shape design features.
- It has two families of shape parameters working in such a way that one family of parameters is associated with intervals and the other with points. These parameters provide a variety of shape controls such as point and interval tension.
- Negative weights can also be utilized for shape design.
- It is computationally economical because it consumes the cubic function only.
- The method of evaluation is suggested by a transformation to Bézier form, which is computable by any well-known recursive method too.
- In addition to direct manipulation, the interpolation method can be computed through B-spline-like formulation too. This point will be discussed in detail somewhere else later.
- The curve method is extendable to surfaces. The direct approach using a tensor product is the simplest one.

2.4 Surfaces

The extension of the curve scheme, to tensor product surface representations:

$$P(\tilde{t}, t) = \sum_{i=-1}^{m+1} \sum_{j=-1}^{n+1} P_{i,j} \tilde{B}_i(\tilde{t}) B_j(t),$$

where $\tilde{t}_{-2} \leq \tilde{t} \leq \tilde{t}_{m+3}, t_{-2} \leq t \leq t_{n+3}$, is immediately apparent. This surface presents a bicubic weighted ν-spline surface with shape parameters as:

$$\tilde{v}_i \geq 0,\ i = 1, \ldots, m, \tilde{w}_i > 0,\ i = 1, \ldots, m-1,$$
$$\nu_j \geq 0, j = 1, \ldots, n, \omega_j > 0, j = 1, \ldots, n-1.$$

Here

$$P_{i,j} \in R^3, i = -1, \ldots, m+1,\ j = -1, \ldots, n+1.$$

are the data points and $\tilde{B}_i, i = -1, \ldots, m+1$ and $B_j, j = -1, \ldots, n+1$ are the local support bases functions for the weighted ν-spline in $\tilde{t}$ and t directions, respectively. However, this representation exhibits a problem common to all tensor product descriptions in that the shape control parameters now affect a complete row or column of the tensor product array.

Nielson [12] solves this problem for his cubic ν-spline representation by constructing a Boolean sum, spline-blended, rectangular network of parametric ν-spline curves. Another possibility is to allow the shape parameters to be variable in the orthogonal direction to, for example, the local support basis functions of the tensor product form.

We propose a tensor product like the approach in [16,17], but actually it is not a tensor product. Instead of step functions, the tension weights are introduced as C^2 continuous cubic B-splines in the description of the tensor product. This produces local control in the construction of surfaces in an independent way. The details of the proposed method are out of the scope of this paper and will be discussed elsewhere.

2.5 Summary

A freeform C^1 weighted Nu spline curve design has been developed through the construction of local support B-spline-like basis functions. This cubic spline method has been developed with a view to its application in computer graphics, geometric modeling, and CAGD. It is quite reasonable to construct a freeform cubic spline method, which involves two families of shape parameters in exactly a similar way as in interpolatory weighted ν-spline. These parameters provide a variety of local and global shape controls such as interval and point shape effects. The visual smoothness of the proposed method is also C^1, which is same as the smoothness of interpolatory weighted ν-spline. The freeform C^1 weighted Nu-spline method can be applied to tensor product surfaces, but unfortunately, in the context of interactive surface design, this tensor product surface is not that useful because any one of the tension parameters controls an entire corresponding interval strip of the surface. Thus, as an application of C^1 spline for the surfaces, a method similar to Nielson's [12] spline blended methods may be attempted. This will produce local shape control, which is quite useful regarding the computer graphics and geometric modeling applications.

2.6 Exercises

1. Write a program to implement the curve design method in Section 2.2.
2. Write a program to implement the curve design method in Section 2.3.
3. Check the difference of shape effects in your programs of Exercise 2.6.1 and 2.6.2 when the schemes are implemented in scalar form.

References

1. Barsky, B.A. (1981), *The Beta-Spline: A Local Representation Based on Shape Parameters and Fundamental Geometric Measure*, Ph.D. Thesis, University of Utah.

2. Bartels, R., and Beatty, J. (1984), Beta-splines with a difference, Technical Report CS-83-40, Computer Science Department, University of Waterloo, Waterloo, Canada.
3. Boehm, W. (1985), Curvature continuous curves and surfaces, *Comp Aided Geom Design* 2(2), 313–323.
4. Cline, A. (1974), Curve fitting in one and dimensions using splines under tension, *Comm ACM* 17, 218–223.
5. Dierckx, P., and Tytgat, B. (1989), Generating the Bézier points of β-spline curve, *Comp Aided Geom Design* 6, 279–291.
6. Farin, G.E. (1996), *Curves and Surfaces for CAGD*, Academic Press, New York.
7. Foley, T.A. (1987), Local control of interval tension using weighted splines. *Comp Aided Geom. Design* 3, 281–294.
8. Foley, T.A. (1987), Interpolation with interval and point tension controls using cubic weighted ν-splines, *ACM Trans Math Software* 13, 68–96.
9. Goodman, T.N.T., and Unsworth, K. (1985), Generation of Beta spline curves using a recursive relation. In: *Fundamental Algorithms for Computer Graphics*. R.E. Earnshaw (Ed.), Springer-Verlag, Berlin, pp. 326–357.
10. Lewis, J. (1975), *"B-spline" bases for splines under tension, Nu-splines, and fractional order splines,* Presented at the SIAM-SIGNUM Meeting, San Francisco, CA.
11. Nielson, G.M. (1974), Some piecewise polynomial alternatives to splines under tension, In: *Computer-Aided Geometric Design*, R.F. Barnhill (Ed.), Academic Press. New York.
12. Nielson, G.M. (1986), Rectangular ν-splines. *IEEE Comp Graph Appl* 6, 35–40.
13. Pruess, S. (1979), Alternatives to the exponential spline in tension, *Math Comp* 33, 1273–1281.
14. Salkauskas, K. (1984), C^1 splines for interpolation of rapidly varying data, *Rocky Mtn J Math* 14, 239–250.
15. Sarfraz, M. (1992), A C^2 rational cubic spline alternative to the NURBS, *Comp & Graph* 16(l), 69–78.
16. Sarfraz, M. (1995), Curves and surfaces for CAD using C2 rational cubic splines, *Int J Eng Comput*, Springer-Verlag, 11(2), 94–102.
17. Sarfraz, M. (1994), Freeform rational bicubic spline surfaces with tension control, Facta Universitatis (NIS), *Ser Math Informatics*, 9, 83–93.
18. Sarfraz, M. (1994), Cubic spline curves with shape control, *Int J Comput Graphics*, Elsevier Science, 18(5), 707–713.
19. Sarfraz, M, (2003), Weighted Nu splines: an alternative to NURBS, *Advances in Geometric Modeling,* Ed.: M. Sarfraz, John Wiley, pp. 81–95.
20. Schoenburg, I. J. (1946), Contributions to the problem of approximation of equidistant data by analytic functions, *Appl Math* 4, 45–99.
21. Schweikert, D.G. (1966), An interpolation curve using a spline in tension, *J Math Phys* 45, 312–317.
22. Sarfraz, M. (2004), Weighted Nu splines with local support basis functions, *Int J Comput Graphics*, Elsevier Science, 28(4), 539–549.