Curves and Surfaces In Geometric Modeling: Theory And Algorithms

Jean Gallier
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
e-mail: jean@saul.cis.upenn.edu

November 8, 2008

To my new daughter Mia, my wife Anne,

my son Philippe, and my daughter Sylvie.

Contents

Preface

This book is primarily an introduction to geometric concepts and tools needed for solving problems of a geometric nature with a computer. Our main goal is to provide an introduction to the mathematical concepts needed in tackling problems arising notably in computer graphics, geometric modeling, computer vision, and motion planning, just to mention some key areas. Many problems in the above areas require some geometric knowledge, but in our opinion, books dealing with the relevant geometric material are either too theoretical, or else rather specialized and application-oriented. This book is an attempt to fill this gap. We present a coherent view of geometric methods applicable to many engineering problems at a level that can be understood by a senior undergraduate with a good math background. Thus, this book should be of interest to a wide audience including computer scientists (both students and professionals), mathematicians, and engineers interested in geometric methods (for example, mechanical engineers). In particular, we provide an introduction to affine geometry. This material provides the foundations for the algorithmic treatment of polynomial curves and surfaces, which is a main theme of this book. We present some of the main tools used in computer aided geometric design (CAGD), but our goal is not to write another text on CAGD. In brief, we are writing about

Geometric Modeling Methods in Engineering

We refrained from using the expression "computational geometry" because it has a well established meaning which does not correspond to what we have in mind. Although we will touch some of the topics covered in computational geometry (for example, triangulations), we are more interested in dealing with curves and surfaces *from an algorithmic point of view*. In this respect, we are flirting with the intuitionist's ideal of doing mathematics from a "constructive" point of view. Such a point of view if of course very relevant to computer science.

This book consists of four parts.

- Part I provides an introduction to affine geometry. This ensures that readers are on firm grounds to proceed with the rest of the book, in particular the study of curves and surfaces. This is also useful to establish the notation and terminology. Readers proficient in geometry may omit this section, or use it *by need*. On the other hand, readers totally unfamiliar with this material will probably have a hard time with the rest of the book. These readers are advised do some extra reading in order to assimilate some basic knowledge of geometry. For example, we highly recommend Berger [5, 6], Pedoe [59], Samuel [69], Hilbert and Cohn-Vossen [42], do Carmo [26], Berger and Gostiaux [7], Boehm and Prautzsch [11], and Tisseron [83].

- Part II deals with an algorithmic treatment of polynomial curves (Bézier curves) and spline curves.

- Part III deals with an algorithmic treatment of polynomial surfaces (Bézier rectangular or triangular surfaces), and spline surfaces. We also include a section on subdivision surfaces, an exciting and active area of research in geometric modeling and animation, as attested by several papers in SIGGRAPH'98, especially the paper by DeRose et al [24] on the animated character Geri, from the short movie *Geri's game*.

- Part IV consists of appendices consisting of basics of linear algebra, certain technical proofs that were omitted earlier, complements of affine geometry, analysis, and differential calculus. This part has been included to make the material of parts I–III self-contained. Our advice is to use it *by need*!

Our goal is not to write a text on the many specialized and practical CAGD methods. Our main goal is to provide an introduction to the concepts needed in tackling problems arising in computer graphics, geometric modeling, computer vision, and motion planning, just to mention some key areas. As it turns out, one of the most spectacular application of these concepts is the treatment of curves and surfaces in terms of control points, a tool extensively used in CAGD. This is why many pages are devoted to an algorithmic treatment of curves and surfaces. However, we only provide a cursory coverage of CAGD methods. Luckily, there are excellent texts on CAGD, including Bartels, Beatty, and Barsky [4], Farin [32, 31], Fiorot and Jeannin [35, 36], Riesler [68], Hoschek and Lasser [45], and Piegl and Tiller [62]. Similarly, although we cover affine geometry in some detail, we are far from giving a comprehensive treatments of these topics. For such a treatment, we highly recommend Berger [5, 6], Pedoe [59], Tisseron [83], Samuel [69], Dieudonné [25], Sidler [76], and Veblen and Young [85, 86], a great classic. Several sections of this book are inspired by the treatment in one of several of the above texts, and we are happy to thank the authors for providing such inspiration.

Lyle Ramshaw's remarkably elegant and inspirational DEC-SRC Report, "Blossoming: A connect–the–dots approach to splines" [65], radically changed our perspective on polynomial curves and surfaces. We have happily and irrevocably adopted the view that the most transparent manner for presenting much of the theory of polynomial curves and surfaces is to stress the multilinear nature (really multiaffine) of these curves and surfaces. This is in complete agreement with de Casteljau's original spirit, but as Ramshaw, we are more explicit in our use of multilinear tools. As the reader will discover, much of the algorithmic theory of polynomial curves and surfaces is captured by the three words:

Polarize, homogenize, tensorize!

We will be dealing primarily with the following kinds of problems:

- **Approximating a shape (curve or surface)**.

  We will see how this can be done using polynomial curves or surfaces (also called Bézier curves or surfaces), spline curves or surfaces.

- **Interpolating a set of points, by a curve or a surface**.

  Again, we will see how this can be done using spline curves or spline surfaces.

- **Drawing a curve or a surface**.

  The tools and techniques developed for solving the approximation problem will be very useful for solving the other two problems.

The material presented in this book is related to the classical differential geometry of curves and surfaces, and to numerical methods in matrix analysis. In fact, it is often possible to reduce problems involving certain splines to solving systems of linear equations. Thus, it is very helpful to be aware of efficient methods for numerical matrix analysis. For further information on these topics, readers are referred to the excellent texts by Gray [39], Strang [81], and Ciarlet [19]. Strang's beautiful book on applied mathematics is also highly recommended as a general reference [80]. There are other interesting applications of geometry to computer vision, computer graphics, and solid modeling. Some great references are Koenderink [46] and Faugeras [33] for computer vision, Hoffman [43] for solid modeling, and Metaxas [53] for physics-based deformable models.

Novelties

As far as we know, there is no fully developed modern exposition integrating the basic concepts of affine geometry as well as a presentation of curves and surfaces from the algorithmic point of view in terms of control points (in the polynomial case). There is also no reasonably thorough textbook presentation of the main surface subdivision schemes (Doo-Sabin, Catmull-Clark, Loop), and a technical discussion of convergence and smoothness.

New Treatment, New Results

This books provides an introduction to affine geometry. Generally, background material or rather technical proofs are relegated to appendices.

We give an in-depth presentation of polynomial curves and surfaces from an algorithmic point of view. The approach (sometimes called *blossoming*) consists in multilinearizing everything in sight (getting *polar forms*), which leads very naturally to a presentation of polynomial curves and surfaces in terms of control points (Bézier curves and surfaces). We present many algorithms for subdividing and drawing curves and surfaces, all implemented in *Mathematica*. A clean and elegant presentation of control points is obtained by using a construction for embedding an affine space into a vector space (the so-called "hat construction", originating in Berger [5]). We even include an optional chapter (chapter 11) covering tensors and symmetric tensors to provide an in-depth understanding of the foundations of blossoming and a more conceptual view of the computational material on curves and surfaces. The continuity conditions for spline curves and spline surfaces are expressed in terms of polar forms, which yields both geometric and computational insights into the subtle interaction of knots and de Boor control points.

Subdivision surfaces are the topic of Chapter 9 (section 9.4). Subdivision surfaces form an active and promising area of research. They provide an attractive alternative to spline surfaces in modeling applications where the topology of surfaces is rather complex, and where the initial control polyhedron consists of various kinds of faces, not just triangles or rectangles. As far as we know, this is the first textbook presentation of three popular methods due to Doo and Sabin [27, 29, 28], Catmull and Clark [17], and Charles Loop [50]. We discuss Loop's convergence proof in some detail, and for this, we give a crash course on discrete Fourier transforms and (circular) discrete convolutions. A glimpse at subdivision surfaces is given in a new Section added to Farin's Fourth edition [32]. Subdivision surfaces are also briefly covered in Stollnitz, DeRose, and Salesin [79], but in the context of wavelets and multiresolution representation.

A a general rule, we try to be rigorous, but we always keep the algorithmic nature of the mathematical objects under consideration in the forefront.

Many problems and programming projects are proposed (over 200). Some are routine, some are (very) difficult.

Many algorithms and their implementation

Although one of our main concerns is to be mathematically rigorous, which implies that we give precise definitions and prove almost all of the results in this book, we are primarily interested in the repesentation and the implementation of concepts and tools used to solve geometric problems. Thus, we devote a great deal of efforts to the development and implemention of algorithms to manipulate curves, surfaces, triangulations, etc. As a matter of fact, we provide *Mathematica* code for most of the geometric algorithms presented in this book. These algorithms were used to prepare most of the illustrations of this book. We also urge the reader to write his own algorithms, and we propose many challenging programming projects.

Open Problems

Not only do we present standard material (although sometimes from a fresh point of view), but whenever possible, we state some open problems, thus taking the reader to the cutting edge of the field. For example, we describe very clearly the problem of finding an efficient way to compute control points for C^k-continuous triangular surface splines. We also discuss some of the problems with the convergence and smoothness of subdivision surface methods.

What's not covered in this book

Since this book is already quite long, we have omitted rational curves and rational surfaces, and projective geometry. A good reference on these topics is [31]. We are also writing a text covering these topics rather extensively (and more). We also have omitted solid modeling techniques, methods for rendering implicit curves and surfaces, the finite elements method, and wavelets. The first two topics are nicely covered in Hoffman [43], a remarkably clear presentation of wavelets is given in Stollnitz, DeRose, and Salesin [79], and a more mathematical presentation in Strang [82], and the finite element method is the subject of so many books that we will not even attempt to mention any references.

Acknowledgement

This book grew out of lectures notes that I wrote as I have been teaching CIS510, *Introduction to Geometric Methods in Computer Science*, for the past four years. I wish to thank some students and colleagues for their comments, including Doug DeCarlo, Jaydev Desai, Will Dickinson, Charles Erignac, Hany Farid, Steve Frye, Edith Haber, Andy Hicks, David Jelinek, Ioannis Kakadiaris, Hartmut Liefke, Dimitris Metaxas, Jeff Nimeroff, Rich Pito, Ken Shoemake, Bond-Jay Ting, Deepak Tolani, Dianna Xu, and most of all Raj Iyer, who screened half of the manuscript with a fine tooth comb. Also thanks to Norm Badler for triggering my interest in geometric modeling, and to Marcel Berger, Chris Croke, Ron Donagi, Gerald Farin, Herman Gluck, and David Harbater, for sharing some of their geometric secrets with me. Finally, many thanks to Eugenio Calabi for teaching me what I know about differential geometry (and much more!).

Chapter 1

Introduction

1.1 Geometric Methods in Engineering

Geometry, what a glorious subject! For centuries, geometry has played a crucial role in the development of many scientific and engineering disciplines such as astronomy, geodesy, mechanics, balistics, civil and mechanical engineering, ship building, architecture, and in this century, automobile and aircraft manufacturing, among others. What makes geometry a unique and particularly exciting branch of mathematics is that it is primarily *visual*. One might say that this is only true of geometry up to the end of the nineteenth century, but even when the objects are higher-dimensional and very abstract, the intuitions behind these fancy concepts almost always come from shapes that can somehow be visualized. On the other hand, it was discovered at the end of the nineteenth century that there was a danger in relying too much on visual intuition, and that this could lead to wrong results or fallacious arguments. What happened then is that mathematicians started using more algebra and analysis in geometry, in order to put it on firmer grounds and to obtain more rigorous proofs. The consequence of the strive for more rigor and the injection of more algebra in geometry is that mathematicians of the beginning of the twentieth century began suppressing geometric intuitions from their proofs. Geometry lost some of its charm and became a rather inpenetrable discipline, except for the initiated. It is interesting to observe that most College textbooks of mathematics included a fair amount of geometry up to the fourties. Beginning with the fifties, the amount of geometry decreases to basically disappear in the seventies.

Paradoxically, with the advent of faster computers, starting in the early sixties, automobile and plane manufacturers realized that it was possible to design cars and planes using computer-aided methods. These methods pioneered by de Casteljau, Bézier, and Ferguson, used geometric methods. Although not very advanced, the type of geometry used is very elegant. Basically, it is a branch of affine geometry, and it is very useful from the point of view of applications. Thus, there seems to be an interesting turn of events. After being neglected for decades, stimulated by computer science, geometry seems to be making a come-back as a fundamental tool used in manufacturing, computer graphics, computer vision, and motion planning, just to mention some key areas.

We are convinced that geometry will play an important role in computer science and engineering in the years to come. The demand for technology using $3D$ graphics, virtual reality, animation techniques, etc, is increasing fast, and it is clear that storing and processing complex images and complex geometric models of shapes (face, limbs, organs, etc) will be required. We will need to understand better how to *discretize* geometric objects such as curves, surfaces, and volumes. This book represents an attempt at presenting a coherent view of geometric methods used to tackle problems of a geometric nature with a computer. We believe that this can be a great way of learning about curves and surfaces, while having fun. Furthermore, there are plenty of opportunities for applying these methods to real-world problems.

Our main focus is on curves and surfaces, but our point of view is algorithmic. We concentrate on methods for discretizing curves and surfaces in order to store them and display them efficiently. Thus, we focus on polynomial curves defined in terms of control points, since they are the most efficient class of curves and surfaces from the point of view of design and representation. However, in order to gain a deeper understanding of this theory of curves and surfaces, we present the underlying geometric concepts in some detail, in particular, affine geometry. In turn, since this material relies on some algebra and analysis (linear algebra, directional derivatives, etc), in order to make the book entirely self-contained, we provide some appendices where this background material is presented.

In the next section, we list some problems arising in computer graphics and computer vision that can be tackled using the geometric tools and concepts presented in this book.

1.2 Examples of Problems Using Geometric Modeling

The following is a nonexhaustive listing of several different areas in which geometric methods (using curves and surfaces) play a crucial role.

- Manufacturing
- Medical imaging
- Molecular modeling
- Computational fluid dynamics
- Physical simulation in applied mechanics
- Oceanography, virtual oceans
- Shape reconstruction
- Weather analysis
- Computer graphics (rendering smooth curved shapes)
- Computer animation

- Data compression
- Architecture
- Art (sculpture, 3D images, ...)

A specific subproblem that often needs to be solved, for example in manufacturing problems or in medical imaging, is to fit a curve or a surface through a set of data points. For simplicity, let us discuss briefly a curve fitting problem.

Problem: Given $N+1$ data points $x_0, \ldots, x_N$ and a sequence of $N+1$ reals $u_0, \ldots, u_N$, with $u_i < u_{i+1}$ for all i, $0 \leq i \leq N-1$, find a C^2-continuous curve F, such that $F(u_i) = x_i$, for all i, $0 \leq i \leq N$.

As stated above, the problem is actually underdetermined. Indeed, there are many different types of curves that solve the above problem (defined by Fourier series, Lagrange interpolants, etc), and we need to be more specific as to what kind of curve we would like to use. In most cases, efficiency is the dominant factor, and it turns out that piecewise polynomial curves are usually the best choice. Even then, the problem is still underdetermined. However, the problem is no longer underdetermined if we specify some "end conditions", for instance the tangents at x_0 and x_N. In this case, it can be shown that there is a unique B-spline curve solving the above problem (see section 6.8). The next figure shows $N+1 = 8$ data points, and a C^2-continuous spline curve F passing through these points, for a uniform sequence of reals u_i.

Other points $d_{-1}, \ldots, d_8$ are also shown. What happens is that the interpolating B-spline curve is really determined by some sequence of points $d_{-1}, \ldots, d_{N+1}$ called *de Boor control points* (with $d_{-1} = x_0$ and $d_{N+1} = x_N$). Instead of specifying the tangents at x_0 and x_N, we can specify the control points d_0 and d_N. Then, it turns out that $d_1, \ldots, d_{N-1}$ can be computed from $x_0, \ldots, x_N$ (and d_0, d_N) by solving a system of linear equations of the form

$$\begin{pmatrix} 1 & & & & & & & \\ \alpha_1 & \beta_1 & \gamma_1 & & & & & \\ & \alpha_2 & \beta_2 & \gamma_2 & & 0 & & \\ & & & \ddots & & & & \\ & & 0 & & \alpha_{N-2} & \beta_{N-2} & \gamma_{N-2} & \\ & & & & & \alpha_{N-1} & \beta_{N-1} & \gamma_{N-1} \\ & & & & & & & 1 \end{pmatrix} \begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{N-2} \\ d_{N-1} \\ d_N \end{pmatrix} = \begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ \vdots \\ r_{N-2} \\ r_{N-1} \\ r_N \end{pmatrix}$$

where r_0 and r_N may be chosen arbitrarily, the coefficients $\alpha_i, \beta_i, \gamma_i$ are easily computed from the u_j, and $r_i = (u_{i+1} - u_{i-1})\, x_i$ for $1 \leq i \leq N-1$ (see section 6.8).

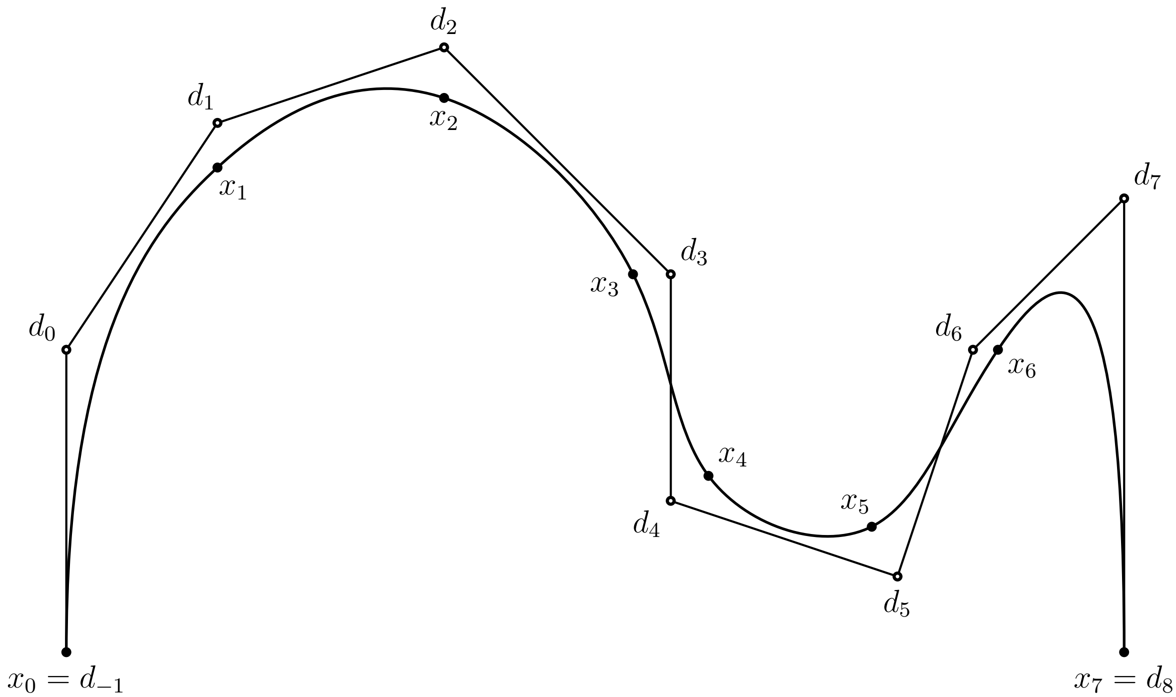

Figure 1.1: A C^2 interpolation spline curve passing through the points x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7

The previous example suggests that curves can be defined in terms of *control points*. Indeed, specifying curves and surfaces in terms of control points is one of the major techniques used in geometric design. For example, in medical imaging, one may want to find the contour of some organ, say the heart, given some discrete data. One may do this by fitting a B-spline curve through the data points. In computer animation, one may want to have a person move from one location to another, passing through some intermediate locations, in a smooth manner. Again, this problem can be solved using B-splines. Many manufacturing problems involve fitting a surface through some data points. Let us mention automobile design, plane design, (wings, fuselage, etc), engine parts, ship hulls, ski boots, etc.

We could go on and on with many other examples, but it is now time to review some basics of affine geometry!