Fakulta matematiky, fyzikzy a informatiky
Univerzity Komenského
Bratislava

Vybrané vlastnosti Bernsteinovej–Bézierovej bázy

Diplomová práca

Tomáš Lackó 2006

Vybrané vlastnosti Bernsteinovej–Bézierovej bázy

DIPLOMOVÁ PRÁCA

Tomáš Lackó

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA ALGEBRY, GEOMETRIE
A DIDAKTIKY MATEMATIKY

študijný odbor:
INFORMATIKA

školiteľ:
RNDr. Pavel Chalmoviansky, PhD.

BRATISLAVA 2006

Chosen properties of the Bernstein–Bézier basis

DIPLOMA WORK

Tomáš Lackó

COMENIUS UNIVERSITY IN BRATISLAVA
FACULTY OF MATHEMATICS, PHYSICS AND INFORMATICS
DEPARTMENT OF ALGEBRA, GEOMETRY
AND DIDACTICS OF MATHEMATICS

field of study:
INFORMATICS

adviser:
RNDr. Pavel Chalmovianský, PhD.

BRATISLAVA 2006

Zadanie

Názov práce:

Vybrané vlastnosti Bernsteinovej-Bézierovej bázy

Cieľ práce:

Štúdium a opis základných vlastností Bernsteinovej-Bézierovej bázy v súvislosti s prienikmi kriviek a plôch.

Tematické zaradenie:

počítačová grafika a geometrické modelovanie

Čestné vyhlásenie

Čestne vyhlasujem, že diplomovú prácu som vypracoval samostatne, iba s použitím uvedenej literatúry.

V Bratislave, júl 2006 ....

Tomáš Lackó

Abstrakt

Táto práca opisuje vybrané vlastnosti Bernsteinovej bázy polynómov v súvislosti s prienikmi parametrických a algebraických kriviek. Sú uvedené základné vlastnosti a spojenia medzi monomiálnou, Bernsteinovou a škálovanou Bernsteinovou bázou. Ukáže sa, že Bernsteinová báza má v porovnaní s inými bázami najväčšiu numerickú stabilitu. Definuje sa rezultant, spolu s výkladom konštrukčných metód a vlastností rôznych formulácií rezultantu pomocou matíc. Okrem známej Sylvestrovej matice sa odvodí matica nižšieho rádu pre Bernsteinove polynómy priamo v tejto báze, vychádzajúc zo sprievodnej matice jedného z polynómov. Prezentujú sa aj transformácie týchto rezultantov z monomiálnej bázy do Bernsteinovej. Výsledky tejto teórie sa nakoniec použijú k výpočtu bodov prieniku dvoch polynomiálnych kriviek algebraickým prístupom. Uvedený je aj numerický prístup pre problém prieniku kriviek či plôch.

kľúčové slová:

Bernsteinova báza, rezultant, matica rezultantu, prienik kriviek

Abstract

This paper describes the properties of the Bernstein polynomial basis, concerning the computation of the intersection points of parametric and algebraic curves. The basic properties and connections between the power basis, the Bernstein basis and the scaled Bernstein basis are presented, and it is shown that the Bernstein basis has the best numerical stability in comparison to other bases. The resultant expression is defined, and the construction and properties of several formulations of resultant matrices are described. In addition to the widely known Sylvester's matrix, a resultant matrix of lower order – the companion matrix resultant – for two polynomials is obtained directly in the Bernstein basis, in terms of the companion matrix of one of the polynomials. The transformation of these resultants from the power basis to the Bernstein basis is also considered. The presented results of the elimination theory are then applied to compute the intersection of two polynomial curves from an algebraic approach, and a numerical solution for this problem is also included.

keywords:

Bernstein basis, resultant, resultant matrix, intersection of curves

Preface

The purpose of this work is to examine the Bernstein basis of polynomials in order to describe its properties related to finding the intersection of algebraic curves and surfaces. As it will be shown, the problem of intersection for such curves can be reduced to finding the common roots of the corresponding polynomials. Therefore, the study is concentrated on the methods of finding the common roots of polynomials expressed in the Bernstein basis.

Significant part of the work is devoted to the comparison and reformulation of properties and resultant expressions represented in different bases. Parts of the formulations come from the literature; these were examined and combined, resulting in possible more general formulations, or proofs involving more straight derivation. In several cases, interesting properties of matrices and resultant expressions seemed to be well described in some recent publications, but were revealed to be more complex when trying to unify the commentaries from more simultaneous sources, or applying them in another basis. The formulations have become more clear, and property relations which were partially missing from the existing publications are now introduced.

Concerning algebraic curves, the technique of implicitization of parametric curves is shown, enabling the application of the presented results in elimination theory to be used for arbitrary representation of polynomial curves. In addition to the algebraic approach, the Bézier curves are described, and their convex hull property along with the Bézier clipping method is used to introduce a numerical solution for the intersection problem. In the end, a numerical method for surface intersection is briefly presented, too.

The contribution of this work is the study and description of the Bernstein polynomial basis and its relation to the power basis and the scaled Bernstein basis, concerning transformations and comparison of matrix methods expressed in these bases, aimed at the computation of the intersection points of algebraic curves, including a unified exposition of recent progress in the area of resultants.

An intergral part of this work is the demonstration software, implementing the most important methods and algorithms presented in this paper.

Contents

0 Introduction **1**

0.1 Commentary on the notation 3

0.2 Auxiliary relations 4

1 Power basis and Bernstein basis **5**

1.1 Introduction to the power basis and the Bernstein basis 5

1.1.1 Definitions 5

1.1.2 Basic properties of the Bernstein basis 6

1.2 Basis transformation 8

1.3 Numerical stability 11

1.3.1 Condition numbers 12

1.3.2 Partial ordering of non-negative bases 13

1.3.3 Optimal stability of the Bernstein basis 14

2 Resultant matrices **15**

2.1 Adaptation of Bernstein polynomials for power basis resultants 16

2.2 Sylvester's resultant matrix 17

2.2.1 Sylvester's matrix for the power basis 17

2.2.2 Sylvester's matrix for Bernstein basis 18

2.2.3 Transformation of the Sylvester's matrix 20

2.3 Companion matrix resultant 22

2.3.1 Companion matrix and resultant for the power basis 23

2.3.2 Companion matrix for Bernstein polynomials 24

2.3.3 Resultant for Bernstein polynomials 27

2.3.4 Transformation of the companion matrix resultant 29

2.4 Other popular resultant matrices 31

2.4.1 Bézout's formulation 31

2.4.2 Cayley's formulation 32

3 Intersection **34**
3.1 Definition of parametric and algebraic curves 34
3.1.1 Parametric curves 34
3.1.2 Algebraic curves 36
3.1.3 Numerical versus algebraic computations 37
3.2 Algebraic approach 37
3.2.1 Implicitizing parametric curves 38
3.2.2 Intersecting parametric curves 39
3.2.3 Intersecting algebraic curves 39
3.3 Numerical approach 40
3.3.1 Solving a non-linear equation 40
3.3.2 Bézier clipping 42
3.3.3 Intersection of a Bézier curve with a line 42
3.3.4 Intersection of curves 43
3.3.5 Intersection of surfaces 44

4 Conclusion **46**

List of Figures

1.1 The Bernstein basis and the scaled Bernstein basis 6

2.1 Parameter substitution $t = \frac{x}{1-x}$ 17

3.1 The convex hull property of Bézier curves 36
3.2 Intersection points with several multiplicity 37
3.3 Numerical approach is not robust 38
3.4 Intersection of algebraic curves: ellipses 40
3.5 Newton's method 41
3.6 Newton's method – algorithm 41
3.7 Bézier clipping 43
3.8 Intersection of curves – clipping 44
3.9 Intersecting surfaces 45
3.10 Subdividing surfaces to remove intersection loops 45

Chapter 0

Introduction

Consider two univariate polynomials and find their common roots effectively, or at least detect their existence. Many practical problems can be reduced to this task, emphasizing the importance of this problem.

Results from this area of research have been known for more than hundred years, pointing out that the resultant expression computed from the coefficients of two polynomials gives a necessary and sufficient condition for the polynomials to have a common root. However, the efforts were made to handle polynomials expressed in the power basis only. Other representations of polynomials had to be transformed. Unfortunately, it was shown by the end of the last century that for a given interval of interest, the power basis is numerically unstable and such basis transformations may be ill-conditioned [1].

Nowadays, along with the increasing importance of geometric modelling, the Bernstein basis have become much more popular. They are frequently subjects to resultant computations, but the numerical superiority of the basis is lost due to unstable polynomial transformations. The necessity of resultant formulations directly in the Bernstein basis is clear, and, according to recent publications, this area has been targeted by several researchers in the last five years. Computations with Bernstein polynomials without involving the power basis have become an intensively developing branch of algebra, geometry and computer graphics.

Once the basics for polynomials and basis transformations are defined, we start a deeper examination of different formulations of resultant matrices, which yield the necessary and sufficient condition for the existence of common roots of polynomials. Then, having satisfactory theoretical results, we take a look at geometric issues, concerning parametric and algebraic curves, and finally we formulate methods for computing the intersection of such objects both numerically and algebraically.

[1] A problem is ill-conditioned if the solution does not depend continuously on the data, and – formulated informally – a very small change in the parameters may result in an incomparably large change in the solution.

The work is divided into the following chapters:

Chapter 1.
: Definition of the polynomial bases that are subject to other chapters. Basic properties of the Bernstein basis and its optimal numerical stability within the set of polynomial bases defined on a given domain of interest.

Chapter 2.
: Definition of the resultant expression. Several formulations of resultant matrices, including different forms of the same matrix type for different polynomial bases. Detailed derivation of the Sylvester resultant matrix and the companion matrix. Transformations of the most important resultant matrices between the power basis and the Bernstein basis.

Chapter 3.
: Parametric and algebraic curves. The intersection problem and different approaches for solving it. Algorithms for computing the intersection of curves or surfaces.

0.1 Commentary on the notation

Generic symbols and abbreviations

Polynomial functions and their coefficients are defined usually in the beginning of the corresponding section or chapter. The Bernstein polynomial functions (β) and scaled Bernstein polynomial functions (α), just as the transformation matrices $T_{\Phi\Omega}$, referenced in the whole document, are defined in the first chapter.

GCD is an abbreviation for the greatest common divisor.
iff is an abbreviation for the term *"if and only if"*.

Terms of type *"polynomial expressed in the Bernstein basis"* are often simplified to the form *"Bernstein polynomial"*. It will be clear from the context whether the basis polynomial functions or an arbitrary polynomial is considered.

Geometric objects

For the d-dimensional space $\mathcal{R}^d$, the following notation is used:

- $P_x, P_y, P_z, \ldots$ are the coordinates of the point P,
  $v_x, v_y, v_z, \ldots$ are the elements of the vector $\vec{v}$.
- $f_x(t), f_y(t), f_z(t), \ldots$ are the components of the d-component polynomial $f(t)$. The vector of the i^{th} coefficients of $f_\kappa(t)$, $\kappa = x, y, z, \ldots$, makes the corresponding coefficient of $f(t)$.

The comma in the vector $P = [P_x, P_y, P_z]$ or $f(t) = [f_x(t), f_y(t)]$ is just to make the notation more transparent.

The reason for a potential different notation may be the already existing index in a sequence of points or functions. In such case, the components will be explicitly defined.

Matrices

Matrix elements are indexed with (row, column) pairs, starting with (1,1).
I_n denotes the $n \times n$ identity matrix.
diag $[\ldots]$ is the diagonal matrix as shown below:

$$\operatorname{diag}\left[\; m_1 \; m_2 \; \cdots \; m_r \;\right] = \left(\; m_{ij} \;\right)_{i,j=1,\ldots,r}, \qquad m_{ij} = \begin{cases} m_i, & i = j \\ 0, & i \neq j \end{cases}$$

Transformation matrices

In the literature, two notations are used for transformation matrices:

According to the more common notation, the matrix M_f of the linear transformation f consists of columns that contain the images of the basis vectors. The image $f(x)$ of an arbitrary vector x is therefore equal to the product $M_f x^T$. This is derived from the notation of the function composition, where the order of the symbols f, g is the same on both sides of the equation $(f \circ g)(x) = f(g(x))$. Thus, $M_{f \circ g} = M_f M_g$ is used for applying g first.

On the other hand, some scientific papers use the notation xf for mapping x by f. In this case, keeping the order of the symbols unchanged, $x(f \circ g) = xfg$. Holding $xf = xM_f$, the matrix $M_{f \circ g} = M_f M_g$ is used for applying f first. In comparison to the first case, M_f is now transposed, containing the images of the basis vectors in rows instead of columns.

In this paper, the first notation is used. Concerning the transpose, it will be clear from the context whether the vector symbols denote row of column vectors.

Proofs

In the equation $A \overset{(R)}{=} B$ of a derivation sequence, the label R references the relation which authorizes this step of the sequence. Most often, the reference is made to a numbered identity elsewhere in the document.

The $\square$ symbol signalizes the end of the proof or derivation sequence.

0.2 Auxiliary relations

In this text, the following basic combinatorial identities are referenced:

$$m \geq 0 \ \wedge \ (r < 0 \ \vee \ r > m) \implies \binom{m}{r} = 0 \, , \tag{1}$$

$$\binom{m}{r}\binom{r}{s} = \binom{m}{s}\binom{m-s}{r-s} \, , \tag{2}$$

$$\sum_{r=0}^{m} (-1)^r \binom{m}{r} = \begin{cases} 1, & m = 0 \\ 0, & m > 0 \end{cases} \, . \tag{3}$$

(1) results from the interpretation of $\binom{m}{r}$ in set theory.
This identity will be used mainly for changing sum boundaries.

(2) results from the definition of $\binom{m}{r}$ by factorials.

(3) can easily be proved by splitting $\binom{m}{r}$ into $\binom{m-1}{r-1} + \binom{m-1}{r}$.

Chapter 1

Power basis and Bernstein basis

This chapter describes the power basis, the Bernstein basis and the scaled Bernstein basis, and presents some of their important properties releated to the definition and conversion of the bases.

1.1 Introduction to the power basis and the Bernstein basis

1.1.1 Definitions

A polynomial $p(x)$ of degree n is represented in basis Φ *iff*

$$p(x) = \sum_{i=0}^{n} c_i \phi_i(x) \tag{1.1}$$

where $\Phi = \{\phi_i(x)\}_{i=0}^{n}$ is a set of linearly independent basis functions that span the space of polynomials of degree n, and c_i is the coefficient of the function $\phi_i(x)$.

Definition 1.1 (power basis)
The power basis (or monomial basis) of degree n consists of polynomials

$$x^i, \qquad 0 \leq i \leq n \ .$$

Definition 1.2 (Bernstein basis – β)
The polynomial functions of the Bernstein basis of degree n are

$$\beta_i^{[n]}(x) = \binom{n}{i}(1-x)^{n-i}x^i, \qquad 0 \leq i \leq n \ .$$

The generalized formulation of the Bernstein basis for the interval $[a, b]$ is

$$\xi_i^{[n]}(x) \ = \ \beta_i^{[n]}\left(\frac{x-a}{b-a}\right) \ = \ \binom{n}{i}\frac{(b-x)^{n-i}(x-a)^i}{(b-a)^n} \ . \tag{1.2}$$

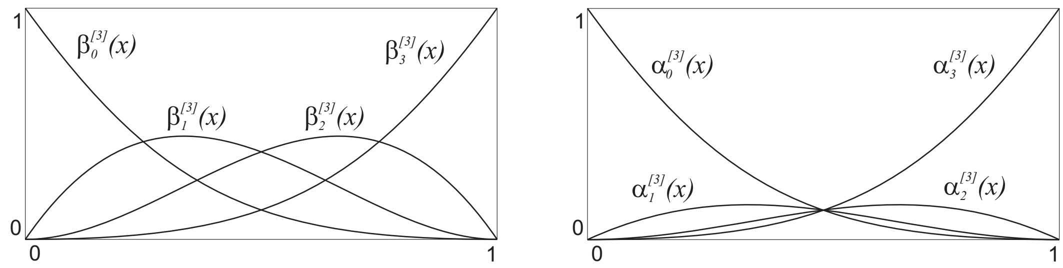
Definition 1.3 (scaled Bernstein basis – α)
The polynomial functions of the scaled Bernstein basis are expressed by removing the combinatorial factor from the Bernstein polynomial functions:

$$\alpha_i^{[n]}(x) = (1-x)^{n-i}x^i, \qquad 0 \le i \le n \ .$$

Note that

$$\alpha_i^{[n]}\alpha_j^{[m]} = \alpha_{i+j}^{[m+n]} \ . \tag{1.3}$$

Figure 1.1: The Bernstein basis and the scaled Bernstein basis



The Bernstein basis and scaled Bernstein basis polynomial funcions of degree 3 on the interval $[0,1]$.

From this point, the notation of $\alpha_i^{[n]}(x)$, $\beta_i^{[n]}(x)$ may be simplified (e.g. to α_i, β_i) if the degree n and the indeterminate x are obvious.

1.1.2 Basic properties of the Bernstein basis

The Bernstein basis was first introduced by S. Bernstein to give an especially simple proof of Weierstrass' approximation theorem (reference in [14]). Since then, it is still widely used in different areas such as approximation theory, or for the representation of curves and surfaces in computer-aided geometric design, because of its elegant geometric properties and stable algorithms that are available for processing it. One such elegant property is that the Bernstein polynomials are invariant under affine transformations [18].

The n^{th} order basis can be recursively generated from the $(n-1)^{\text{th}}$ order basis by

$$\beta_i^{[n]}(x) \ = \ (1-x)\beta_i^{[n-1]}(x) + x\beta_{i-1}^{[n-1]}(x) \ .$$

A polynomial of degree n can be represented in terms of the Bernstein basis of degree $n+1$ by *degree elevation* [18]. If $\{b_i^{[n]}\}_{i=0}^n$ are the Bernstein coefficients in the degree-n basis, then the coefficients in the next higher basis are given by:

$$b_i^{[n+1]} \ = \ \begin{cases} b_0^{[n]}, & i=0 \\ (1-k_i)b_i^{[n]} + k_i b_{i-1}^{[n]}, & 1 \le i \le n \\ b_n^{[n]}, & i = n+1 \end{cases}$$

$$k_i \ = \ \frac{i}{n+1}, \qquad i = 1, \ldots, n \ .$$

A further property of the basis is that the sum of the Bernstein polynomial functions equals 1, as stated by lemma 1.2.

Lemma 1.1
The scaled Bernstein polynomial functions can be transposed to the following sum of monomial components:

$$(1-x)^{n-i}x^i = \sum_{k=0}^{n}(-1)^{i+k}\binom{n-i}{k-i}x^k . \tag{1.4}$$

Proof.
The first step in the sequence expresses the product in powers of x.

$$\begin{aligned}(1-x)^{n-i}x^i &= \sum_{j=0}^{n-i}\binom{n-i}{j}(-x)^j x^i = \sum_{j=0}^{n-i}(-1)^j\binom{n-i}{j}x^{i+j} \overset{k:=i+j}{=} \\ &= \sum_{k=i}^{n}(-1)^{k-i}\binom{n-i}{k-i}x^k \overset{(1)}{=} \sum_{k=0}^{n}(-1)^{i+k}\binom{n-i}{k-i}x^k\end{aligned}$$

□

Lemma 1.2
The sum of the Bernstein polynomial functions equals 1:

$$\sum_{i=0}^{n}\beta_i^{[n]}(x) = 1 . \tag{1.5}$$

Proof.

$$\begin{aligned}\sum_{i=0}^{n}\beta_i^{[n]}(x) &= \sum_{i=0}^{n}\binom{n}{i}(1-x)^{n-i}x^i = \\ &\overset{(1.4)}{=} \sum_{i=0}^{n}\binom{n}{i}\sum_{k=0}^{n}(-1)^{i+k}\binom{n-i}{k-i}x^k = \\ &\overset{(2)}{=} \sum_{k=0}^{n}\binom{n}{k}x^k\sum_{i=0}^{n}(-1)^{i+k}\binom{k}{i} = \\ &\overset{(1)}{=} \sum_{k=0}^{n}\binom{n}{k}x^k(-1)^k\sum_{i=0}^{k}(-1)^i\binom{k}{i} = \\ &\overset{(3)}{=} \binom{n}{0}x^0(-1)^0\,1 + 0 = 1\end{aligned}$$

□

The facts that all the polynomial functions of the Bernstein basis are positive on the interval where they are defined, and that their sum equals 1, gives a bound on the polynomial $p(x)$ in the Bernstein basis with coefficients $\{b_i\}_{i=0}^{n}$:

$$\min_{0\le i\le n} b_i \le p(x) \le \max_{0\le i\le n} b_i .$$

A tighter bound is given by the convex hull determined by the coefficients [18].

1.2 Basis transformation

Definition 1.4
The transformation matrix between polynomial bases Φ and Ω of degree n is the $(n+1)\times(n+1)$ matrix $T_{\Phi\Omega}$ satisfying [1]

$$\Omega T_{\Phi\Omega} = \Phi \ . \tag{1.6}$$

Consider an arbitrary polynomial with vector C of coefficients in basis Φ and vector D of coefficients in basis Ω. Both representations denote the same polynomial, thus $\Phi C^T = \Omega D^T$. Using (1.6), this leads to $T_{\Phi\Omega} C^T = D^T$, which yields that (just as expected) the transformation matrix $T_{\Phi\Omega}$ maps the coefficients in C to those in D, from basis Φ to basis Ω.

Considering the power basis, the Bernstein basis and the scaled Bernstein basis, the following theorems introduce the general transformation matrices between pairs of these bases, for arbitrary degree.

Theorem 1.1 ($\beta \rightsquigarrow \alpha,\ \alpha \rightsquigarrow \beta$)
The transformation matrices between the Bernstein basis and the scaled Bernstein basis of degree $n \geq 1$ – and vice-versa – have the following form:

$$\begin{aligned} T^{[n]}_{\beta\alpha} &= \operatorname{diag}\left[\ \binom{n}{0} \ \ \binom{n}{1} \ \ \cdots \ \ \binom{n}{n} \ \right] , \\ T^{[n]}_{\alpha\beta} &= \operatorname{diag}\left[\ \frac{1}{\binom{n}{0}} \ \ \frac{1}{\binom{n}{1}} \ \ \cdots \ \ \frac{1}{\binom{n}{n}} \ \right] . \end{aligned}$$

Proof.
Trivial, both follow from the identity $\beta^{[n]}_i = \binom{n}{i}\alpha^{[n]}_i$. □

Theorem 1.2 ($\beta \rightsquigarrow x$)
The transformation matrix between the Bernstein basis and the power basis of degree $n \geq 1$ has the following form:

$$T^{[n]}_{\beta x} = \left(\ t^{[\beta,x,n]}_{ij} \ \right) ,$$

$$t^{[\beta,x,n]}_{ij} = (-1)^{i+j}\binom{n}{i}\binom{i}{j}, \quad i,j = 0,\ldots,n \ . \tag{1.7}$$

Proof.
Let $p(x)$ be an arbitrary polynomial and $\{b_i\}_{i=0}^n$ its coefficients in the Bernstein basis. The transformation is achieved using lemma 1.1:

$$\begin{aligned} p(x) &= \sum_{j=0}^{n} b_j \beta^{[n]}_j = \sum_{j=0}^{n} b_j \binom{n}{j}(1-x)^{n-j}x^j = \\ &\overset{(1.4)}{=} \sum_{j=0}^{n} b_j \binom{n}{j} \sum_{i=0}^{n}(-1)^{i+j}\binom{n-j}{i-j}x^i = \\ &= \sum_{i=0}^{n} x^i \sum_{j=0}^{n}(-1)^{i+j}\binom{n}{j}\binom{n-j}{i-j}b_j = \\ &\overset{(2)}{=} \sum_{i=0}^{n} x^i \sum_{j=0}^{n} t^{[\beta,x,n]}_{ij} b_j = \sum_{i=0}^{n} x^i a_i \ . \end{aligned}$$

[1] This time, the sets Φ and Ω are treated as row vectors of the basis polynomial functions.

The result form is the power basis representation of $p(x)$. Constants $t_{ij}^{[\beta,x,n]}$ are from (1.7) and a_i are defined by

$$\begin{pmatrix} t_{ij}^{[\beta,x,n]} \end{pmatrix} \begin{bmatrix} b_0 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} \tag{1.8}$$

This concludes that elements $t_{ij}^{[\beta,x,n]}$ define the transformation matrix $T_{\beta x}^{[n]}$ between these bases. □

Theorem 1.3 ($\alpha \rightsquigarrow x$)
The transformation matrix between the scaled Bernstein basis and the power basis of degree $n \geq 1$ has the following form:

$$T_{\alpha x}^{[n]} = \begin{pmatrix} t_{ij}^{[\alpha,x,n]} \end{pmatrix},$$

$$t_{ij}^{[\alpha,x,n]} = (-1)^{i+j}\binom{n-j}{i-j}, \quad i,j = 0,\dots,n\,.$$

Proof.
Follows from $T_{\alpha x}^{[n]} = T_{\beta x}^{[n]} T_{\alpha\beta}^{[n]}$ by theorems 1.1 and 1.2. □

Lemma 1.3
The monomial [2] x^i can be transformed to the scaled Bernstein basis as shown:

$$n > 0 \;\wedge\; 0 \leq i \leq n \quad\Longrightarrow\quad x^i = \sum_{j=i}^{n}\binom{n-i}{j-i}\alpha_j^{[n]}\,. \tag{1.9}$$

Proof.
By mathematical induction with measure $\mu = n - i$. We show that the equation holds for $\mu = 0$ and that the validity for all $\rho < \mu$ implies the validity for μ. We use the auxiliary variable $k := n - \rho$.

Step 1.
$\mu = 0, \quad i = n$ (trivial)

$$x^n = \sum_{j=n}^{n}\binom{n-n}{j-n}\alpha_j^{[n]} = \alpha_n^{[n]} = (1-x)^0 x^n$$

Step 2.
Have the induction hypothesis (IH)

$$\forall\rho: \quad \rho < \mu \quad \overset{k:=n-\rho}{\underset{k\geq i}{\Longrightarrow}} \quad x^k = \sum_{j=k}^{n}\binom{n-k}{j-k}\alpha_j^{[n]}\,.$$

[2] For univariate polynomials, monomials are the terms x^i.

Derive x^i from the definition of $\alpha_i^{[n]}$:

$$\begin{aligned}\alpha_i^{[n]} &= (1-x)^{n-i}x^i \overset{(1.4)}{=} \sum_{k=i}^{n}(-1)^{i+k}\binom{n-i}{k-i}x^k = \\ &= x^i + \sum_{k=i+1}^{n}(-1)^{i+k}\binom{n-i}{k-i}x^k \,. \end{aligned} \tag{1.10}$$

Finally, prove the equation for i using the induction hypothesis.

$$\begin{aligned} x^i &\overset{(1.10)}{=} \alpha_i^{[n]} - \sum_{k=i+1}^{n}(-1)^{i+k}\binom{n-i}{k-i}x^k = \\ &\overset{(\mathrm{IH})}{=} \alpha_i^{[n]} - \sum_{k=i+1}^{n}(-1)^{i+k}\binom{n-i}{k-i}\sum_{j=k}^{n}\binom{n-k}{j-k}\alpha_j^{[n]} = \\ &= \alpha_i^{[n]} - \sum_{\substack{j,k\\ i<k\le j\le n}}(-1)^{i+k}\binom{n-i}{k-i}\binom{n-k}{j-k}\alpha_j^{[n]} = \\ &= \alpha_i^{[n]} - \sum_{j=i+1}^{n}\alpha_j^{[n]}\sum_{k=i+1}^{j}(-1)^{i+k}\binom{n-i}{k-i}\binom{n-k}{j-k} = \\ &\overset{(2)}{=} \alpha_i^{[n]} - \sum_{j=i+1}^{n}\alpha_j^{[n]}\binom{n-i}{j-i}\sum_{k=i+1}^{j}(-1)^{i+k}\binom{j-i}{k-i} = \\ &= \alpha_i^{[n]} - \sum_{j=i+1}^{n}\alpha_j^{[n]}\binom{n-i}{j-i}\sum_{r=1}^{j-i}(-1)^{r}\binom{j-i}{r} = \\ &\overset{\substack{n>0\\(3)}}{=} \alpha_i^{[n]} - \sum_{j=i+1}^{n}\alpha_j^{[n]}\binom{n-i}{j-i}\left[0 - (-1)^0\binom{j-i}{0}\right] = \\ &= \alpha_i^{[n]} + \sum_{j=i+1}^{n}\alpha_j^{[n]}\binom{n-i}{j-i} = \sum_{j=i}^{n}\alpha_j^{[n]}\binom{n-i}{j-i}\,, \end{aligned}$$

and this is the form of x^i we requested. Hence, the equality is proved. □

Corollary 1.1
Using (1), the boundaries of the sum in (1.11) can be extended so that

$$n>0 \ \wedge\ 0\le i\le n \quad\Longrightarrow\quad x^i = \sum_{j=0}^{n}\binom{n-i}{j-i}\alpha_j^{[n]}\,. \tag{1.11}$$

Theorem 1.4 ($x \rightsquigarrow \alpha$)
The transformation matrix between the power basis and the scaled Bernstein basis of degree $n \ge 1$ has the following form:

$$T_{x\alpha}^{[n]} = \left(t_{ij}^{[x,\alpha,n]} \right),$$

$$t_{ij}^{[x,\alpha,n]} = \binom{n-j}{i-j}, \quad i,j = 0,\ldots,n\,.$$

Proof.
Let $p(x)$ be an arbitrary polynomial and $\{a_i\}_{i=0}^n$ its coefficients in the power basis. The transformation is achieved using corollary 1.1 (of lemma 1.3):

$$\begin{aligned} p(x) &= \sum_{i=0}^{n} a_i x^i \overset{(1.11)}{=} \sum_{i=0}^{n} a_i \sum_{j=0}^{n} \alpha_j^{[n]} \binom{n-i}{j-i} = \\ &= \sum_{j=0}^{n} \alpha_j^{[n]} \sum_{i=0}^{n} \binom{n-i}{j-i} a_i = \sum_{j=0}^{n} \alpha_j^{[n]} s_j \end{aligned}$$

The result form is the scaled Bernstein basis representation of $p(x)$ with coefficients $\{s_j\}_{j=0}^n$. Analogically to the proof of theorem 1.2, it concludes that factors $\binom{n-i}{j-i}$ are the elements of the transformation matrix $T_{x\alpha}^{[n]}$ between these bases. □

Theorem 1.5 ($x \rightsquigarrow \beta$)
The transformation matrix between the power basis and the Bernstein basis of degree $n \geq 1$ *has the following form:*

$$T_{x\beta}^{[n]} = \left(t_{ij}^{[x,\beta,n]} \right),$$

$$t_{ij}^{[x,\beta,n]} = \frac{\binom{i}{j}}{\binom{n}{j}}, \quad i, j = 0, \ldots, n .$$

Proof.
Follows from $T_{x\beta}^{[n]} = T_{\alpha\beta}^{[n]} T_{x\alpha}^{[n]}$ by theorems 1.1 and 1.4 and the identity (2). □

Example

Transformation of a cubic polynomial between the power basis and the Bernstein basis:

$$T_{x\beta}^{[3]} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1/3 & 0 & 0 \\ 1 & 2/3 & 1/3 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \qquad T_{\beta x}^{[3]} = \begin{pmatrix} +1 & 0 & 0 & 0 \\ -3 & +3 & 0 & 0 \\ +3 & -6 & +3 & 0 \\ -1 & +3 & -3 & +1 \end{pmatrix}$$

$$\begin{aligned} a + bx + cx^2 + dx^3 &= a\beta_0 + (a + \tfrac{1}{3}b)\beta_1 + (a + \tfrac{2}{3}b + \tfrac{1}{3}c)\beta_2 + (a + b + c + d)\beta_3 \\ a\beta_0 + b\beta_1 + c\beta_2 + d\beta_3 &= a + (-3a + 3b)x + (3a - 6b + 3c)x^2 + (-a + 3b - 3c + d)x^3 \end{aligned}$$

1.3 Numerical stability

It can be shown by backward error analysis that the cumulative effect of floating-point arithmetic errors during any computation on given polynomials is equivalent to certain perturbations on their exact coefficients [19]. Therefore, to compare polynomial bases, the point of our interest is to establish a condition number measuring the sensitivity of roots to random perturbations in the coefficients of polynomials. When such a condition number is defined, it will be desirable to employ a representation – a choice of basis – in which the condition numbers are as small as possible, in order to obtain accurate results when performing floating-point computations with polynomials.

The details of the following section are described in [19].

1.3.1 Condition numbers

Any polynomial of degree at most n can be uniquely expressed in the form

$$p(t) = \sum_{i=0}^{n} c_i \phi_i(t) \tag{1.12}$$

by a suitable choice of coefficients $c_0, \ldots, c_n$ for basis $\Phi = \{\phi_0(t), \ldots, \phi_n(t)\}$. We shall be concerned here with the stability of such representations: how senstive a value or root of p is to random perturbations of a given maximum relative magnitude ϵ in the coefficients corresponding to basis Φ.

Definition 1.5
The condition number for the value of the polynomial p defined in (1.12) is the quantity

$$C_\Phi(p(t)) = \sum_{i=0}^{n} |c_i \phi_i(t)| \ .$$

Note that $C_\Phi(p(t))$ depends as much on the adopted basis Φ as on the particular polynomial p under consideration. A sharp bound on the perturbation $\delta p(t)$, holding for arbitrary (not just infinitesimal) coefficient perturbations ϵ, may be expressed as

$$|\delta p(t)| \leq C_\Phi(p(t))\epsilon \ .$$

Suppose now that τ is a simple real root of $p(t)$, i.e. $p(\tau) = 0 \neq p'(\tau)$. The sensitivity of τ to a perturbation ϵ of the coefficients $c_0, \ldots, c_n$ in the basis Φ can also be characterized by a condition number.

Definition 1.6
The condition number for the root τ of polynomial $p(t)$ defined in (1.12) is the quantity

$$C_\Phi(\tau) = \frac{1}{|p'(\tau)|} \sum_{i=0}^{n} |c_i \phi_i(t)| \ .$$

[19] says that the displacement $\delta\tau$ of this root, strictly valid only in the limit for $\epsilon \to 0$, satisfies

$$|\delta\tau| \leq C_\Phi(\tau)\epsilon \ .$$

When comparing condition numbers in arbitrary bases without imposing suitable restrictions on the bases, no systematic inequality can be expected. We shall be concerned here only with bases which are non-negative over an interval $t \in [a, b]$. Such bases are of particular interest in the context of the following result:

Lemma 1.4
Let $\Phi = \{\phi_0(t), \ldots, \phi_n(t)\}$ *and* $\Psi = \{\psi_0(t), \ldots, \psi_n(t)\}$ *be non-negative bases for degree* n *polynomials on* $t \in [a, b]$ *such that* Ψ *can be expressed as a non-negative combination of* Φ*:*

$$\psi_j(t) = \sum_{i=0}^{n} m_{ij}\phi_i(t), \qquad j = 0, \ldots, n \ ,$$

$$m_{ij} \geq 0 \quad \textit{for all} \quad 0 \leq i, j \leq n \ .$$

Then the condition number for the value of any degree-n *polynomial* $p(t)$ *at any point* $t \in [a, b]$ *in these bases satisfies the inequality*

$$C_\Phi(p(t)) \leq C_\Psi(p(t)) \ .$$

In [19] the lemma is stated to be an immediate consequence of the triangle inequality and the non-negativity of the matrix elements m_{ij} and the bases Φ and Ψ. The inequality also holds for root condition numbers, since these differ from condition numbers for the value only by the magnitude of the derivative at the root, which is independent of the choice of basis.

1.3.2 Partial ordering of non-negative bases

Let Π_n be the space of all polynomials of degree at most n on the interval $[a, b]$. Let B_n denote the set of non-negative bases for Π_n.

Definition 1.7 ($\preceq$)
For bases Φ *and* Ψ *in* B_n*, we write* $\Phi \preceq \Psi$ *if there exist a non-negative* $(n+1)\times(n+1)$ *sized matrix* M *such that*

$$\Psi^T = M\Phi^T \ .$$

$\Phi \sim \Psi$ *holds iff* $\Phi \preceq \Psi$ *and* $\Psi \preceq \Phi$.
$\Phi \prec \Psi$ *holds iff* $\Phi \preceq \Psi$ *without* $\Phi \sim \Psi$.

The relation $\preceq$ induces a partial ordering among the members of B_n, since it is reflexive, antisymmetric and transitive [19]. The ordering is partial because if neither Φ nor Ψ can be expressed as a non-negative combination of the other, they will not be comparable.

According to the definition of minimal elements in a partially ordered set, the basis Φ is minimal if there is no basis Ψ in B_n satisfying $\Psi \prec \Phi$. Note that there may be more than one mutually incomparable minimal bases. The following theorem demonstrates that minimal bases are optimally stable, in the sence of the condition numbers from definition 1.5.

Theorem 1.6
Any two bases Φ *and* Ψ *in* B_n *satisfy*

$$\Phi \preceq \Psi \quad \Longleftrightarrow \quad \forall p \in \Pi_n \quad \forall t \in [a, b] \quad C_\Phi(p(t)) \leq C_\Psi(p(t)) \ . \tag{1.13}$$

Note that $\Phi \sim \Psi$ implies $C_\Phi(p(t)) \equiv C_\Psi(p(t))$ for each $p \in \Pi_n$ and every $t \in [a, b]$.

1.3.3 Optimal stability of the Bernstein basis

As already indicated by theorem 1.6 and the paragraph above it, being optimal means minimality in terms of the partial ordering $\preceq$. An optimal basis is the least sensitive to random perturbations in the coefficients, from within all bases which are comparable to it.

Let $\mathcal{B}$ be the generalized Bernstein basis for $x \in [a, b]$, defined in (1.2). We demonstrate that $\mathcal{B}$ has optimal stability in B_n.

Theorem 1.7
Suppose that $\Psi = \{\psi_0, \dots, \psi_n\}$ and $\Omega = \{\omega_0, \dots, \omega_n\}$ are bases in B_n satisfying [3]

$$\psi_i^{(m)}(a) = 0 \quad \textit{for} \quad i = 1, \dots, n \quad \textit{and} \quad m = 0, \dots, i-1 ,$$
$$\omega_i^{(m)}(a) = 0 \quad \textit{for} \quad i = 0, \dots, n-1 \quad \textit{and} \quad m = 0, \dots, n-i-1 .$$

Then, if $\Phi \in B_n$ satisfies both $\Phi \preceq \Psi$ and $\Phi \preceq \Omega$, we have $\Phi \sim \mathcal{B}$.

There is a quite complex proof in [19] showing that Φ may be reordered such that the matrix M from definition 1.7 is lower triangular, and that $\phi_i(t) = c_i(b-t)^{n-i}(t-a)^i$, $i = 0 \dots n$ holds for nonzero constants $c_0, \dots, c_n$.

Corollary 1.2
According to the partial ordering $\preceq$, the basis $\mathcal{B}$ is minimal.

Proof.
Suppose $\Phi \in B_n$ is such that $\Phi \preceq \mathcal{B}$. Since the basis $\mathcal{B}$ satisfies the conditions on both Ψ and Ω stipulated in theorem 1.7, this theorem implies that $\Phi \sim \mathcal{B}$. Thus, there is no basis Φ in B_n such that $\Phi \prec \mathcal{B}$. □

Corollary 1.2 establishes the optimal stability of the Bernstein basis.
We note also the following corollaries to theorem 1.7:

Corollary 1.3
If $\Phi \in B_n$ satisfies both $\Phi \preceq \{1, t-a, \dots, (t-a)^n\}$ and $\Phi \preceq \{1, b-t, \dots, (b-t)^n\}$, then $\Phi \sim \mathcal{B}$.

Corollary 1.4
Suppose that $\Phi \in B_n$ satisfies $\Phi \preceq \{1, t-a, \dots, (t-a)^n\}$ and Φ is symmetric, i.e. $\Phi(t) \sim \Phi(a+b-t)$. Then $\Phi \sim \mathcal{B}$.

[3] $f^{(m)}$ denotes the m-th derivative of f.

Chapter 2

Resultant matrices

Resultants are a classical algebraic tool for determining whether or not a system of n polynomials in $n-1$ variables have a common root without explicitly solving for the roots. Gröbner basis methods can also be used for this task [10]; however, resultants are usually more efficient than Gröbner bases in practical applications.

Definition 2.1 (resultant)
The resultant of a set of polynomials is an expression involving the coefficients of the polynomials, such that a necessary and sufficient condition for the set of polynomials to have a common root is that the resultant expression is exactly zero.

There exist several different types of resultant, for example, the Sylvester, Macaulay, Newton, Bézout, Dixon, sparse and U-resultants (and even more) [1, 14, 15], and they may be considered theoretically equivalent because they all yield necessary and sufficient conditions for polynomials to have a common root. Resultants are often represented as the determinant of a matrix whose entries are polynomials in the coefficients of the original polynomial equations. These matrices may be very large, especially in the multivariate setting (as stated in [16]).

It was realized in the 1980's that resultants can be applied to many problems in computer-aided geometric design. The Sylvester and Bézout matrices, presented in the 19^{th} century, are the oldest formulations but used up to the present. Both of them has more ways to be derived. On the other side, according to recent publications [3, 5, 8, 13], the companion matrix resultant seems to be a new, intensively developing branch of the elimination theory.

The main disadvantage of the resultant method is that in some cases the resultant of a polynomial set can become identically zero. This is due to the fact that the resultant matrix is singular (for example, due to the presence of base points of the parametrisation). To overcome these situations, different approaches have been investigated [14].

The following sections present the most known, widely used matrices used to construct the resultant expression.

2.1 Adaptation of Bernstein polynomials for power basis resultants

Resultants were originally developed for the power basis. However, the polynomial basis transformation may be ill-conditioned, thus it is not a computationally reliable solution. An alternative method involves the parameter substitution presented below [13, 11], enabling the entire theory of the resultant of power basis polynomials to be reproduced for Bernstein basis polynomials.

Let $p(x)$ be a polynomial expressed in the Bernstein basis:

$$p(x) = \sum_{i=0}^{n} b_i \binom{n}{i} (1-x)^{n-i} x^i \ . \tag{2.1}$$

The parameter substitution

$$t = \frac{x}{1-x}, \qquad x \neq 1 \tag{2.2}$$

is a Möbius transformation [2] which yields

$$p(x) \ = \ p\left(\frac{t}{1+t}\right) \ = \ \frac{1}{(1+t)^n} \sum_{i=0}^{n} b_i \binom{n}{i} t^i \ .$$

Thus, if x_0 is a root of $p(x)$, then $t_0 = \dfrac{x_0}{1-x_0}$ is a root of the polynomial

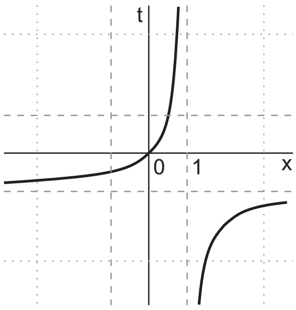
$$q(t) \ = \ (1+t)^n p\left(\frac{t}{1+t}\right) \ = \ \sum_{i=0}^{n} b_i \binom{n}{i} t^i, \qquad t \neq -1 \ .$$

The coefficients $b_i\binom{n}{i}$ of the power basis polynomial $q(t)$ are those from the representation of $p(x)$ in the scaled Bernstein basis.

The use of the Bernstein basis implies that interest is restricted to the interval $0 \leq x \leq 1$, but the parameter substitution (2.2) is not valid at $x = 1$. Assuming that $(1-x)$ is a factor of $p(x)$, this disadventage can be overcome by removing it before the parameter substitution is made. The interval $[0, 1)$ is then mapped to $[0, \infty)$ as shown in figure 2.1.

This substitution is adequate for theoretical analysis or symbolic computations, but it cannot be used in a floating-point environment because all computations are performed in the power basis, which is numerically inferior to the Bernstein basis [19]. The parameter substitution is numerically inferior even to the scaled Bernstein basis [11]. When using this or a similar transformation, one of the advantages of the Bernstein basis is lost: its enhanced numerical stability. Therefore, it is desirable to retain the Bernstein basis throughout the computations.

Figure 2.1: Parameter substitution $t = \frac{x}{1-x}$



2.2 Sylvester's resultant matrix

Sylvester's matrix, or Sylvester's dialytic expansion [1], is one of the most frequently referenced methods of constructing a resultant expression. Hereby, we present its formulation for the power basis, the Bernstein basis and the scaled Bernstein basis, and its transformation between the power basis and the Bernstein basis.

2.2.1 Sylvester's matrix for the power basis

Definition 2.2 (S_x)
Sylvester's matrix for the polynomials $f(x) = \sum_{i=0}^{m} a_i x^i$ of degree m and $g(x) = \sum_{i=0}^{n} b_i x^i$ of degree n has the form

$$S_x(f,g) = \begin{pmatrix} a_m & a_{m-1} & \cdots & a_1 & a_0 & & 0 & 0 \\ & a_m & a_{m-1} & \cdots & a_1 & a_0 & & 0 \\ 0 & & \ddots & \ddots & & \ddots & \ddots & \\ 0 & 0 & & a_m & a_{m-1} & \cdots & a_1 & a_0 \\ b_n & b_{n-1} & \cdots & b_1 & b_0 & & 0 & 0 \\ & b_n & b_{n-1} & \cdots & b_1 & b_0 & & 0 \\ 0 & & \ddots & \ddots & & \ddots & \ddots & \\ 0 & 0 & & b_n & b_{n-1} & \cdots & b_1 & b_0 \end{pmatrix} , \tag{2.3}$$

where n lines are constructed using the coefficients of $f(x)$ and m lines using the coefficients of $g(x)$.

Theorem 2.1 (S_x)
The determinant of Sylvester's matrix $S_x(f,g)$ is the resultant of the two polynomials $f(x)$ and $g(x)$.

Proof.
The matrix representation can be easily derived [6, 4] by thinking of polynomials as linear equations in powers of x. Let us start with polynomial equations

$$\begin{aligned} f(x) &= 0 , \\ g(x) &= 0 . \end{aligned} \tag{2.4}$$

[1] The naming comes from [25].

This system only has a solution when f and g have a common root. We will add more equations to come up with a matrix equation involving a square matrix. There is no harm in adding equations of the form $x^k f(x) = 0$ or $x^k g(x) = 0$ to the system (2.4) because the enlarged system will have exactly the same solutions as the original system of two equations. Consider the system

$$\begin{aligned} x^{n-1} f(x) &= 0 \,, \\ x^{n-2} f(x) &= 0 \,, \\ &\vdots \\ f(x) &= 0 \,, \\ x^{m-1} g(x) &= 0 \,, \\ x^{m-2} g(x) &= 0 \,, \\ &\vdots \\ g(x) &= 0 \,. \end{aligned}$$

This system may be written as the matrix equation

$$S_x(f,g) \begin{pmatrix} x^{m+n-1} \\ \vdots \\ x^0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

using Sylvester's matrix defined in (2.3).

An important property of matrix equations involving square matrices is that they only have non-trivial solutions when the determinant of the matrix vanishes. Hence, the determinant of Sylvester's matrix will vanish whenever f and g have a common root. □

The resultant can also be derived using a method invented by Euler. For details, see [10].

2.2.2 Sylvester's matrix for Bernstein basis

The scaled Bernstein basis is also considered, since the corresponding definitions and proofs serve as a base for their Bernstein basis equivalents.

Definition 2.3 (S_α)
Sylvester's matrix for the scaled Bernstein polynomials $p(x)$ of degree m and $q(x)$ of degree n has the same form as the matrix in (2.3), constructed using the coefficients $\{p_i\}_{i=0}^m$ and $\{q_i\}_{i=0}^n$ for the scaled Bernstein basis.

Theorem 2.2 (S_α)
The determinant of Sylvester's matrix $S_\alpha(p,q)$ is the resultant of the scaled Bernstein polynomials $p(x)$ and $q(x)$.

Proof.
Analogous to the proof of theorem 2.1.
Consider the following system of equations:

$$\begin{aligned}
\alpha_0^{[n-1]} p(x) &= 0 \, , \\
&\vdots \\
\alpha_{n-1}^{[n-1]} p(x) &= 0 \, , \\
\alpha_0^{[m-1]} q(x) &= 0 \, , \\
&\vdots \\
\alpha_{m-1}^{[m-1]} q(x) &= 0 \, .
\end{aligned}$$

According to (1.3), this system may be written as the matrix equation

$$S_\alpha(p,q) \begin{pmatrix} \alpha_0^{[m+n-1]} \\ \vdots \\ \alpha_{m+n-1}^{[m+n-1]} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

which has a solution if and only if p and q have a common root, thus, when the determinant of $S_\alpha(p,q)$ is not zero. □

Definition 2.4 (S_β)
Sylvester's matrix for the Bernstein polynomials $r(x)$ of degree m and $s(x)$ of degree n has the form

$$S_\beta(r,s) = S_\alpha(r,s)\, T_{\alpha\beta}^{[m+n-1]} \, ,$$

where

- $\{r_i\}_{i=0}^m$ *and* $\{s_i\}_{i=0}^n$ *are the coefficients of r and s in the Bernstein basis,*
- $S_\alpha(r,s)$ *has the form (2.3) for elements* $\left\{a_i := r_i\binom{m}{i}\right\}_{i=0}^m$ *and* $\left\{b_i := s_i\binom{n}{i}\right\}_{i=0}^n$,
- $T_{\alpha\beta}^{[m+n-1]}$ *defined in the theorem 1.1 is the diagonal matrix with elements* $\left\{\frac{1}{\binom{m+n-1}{i}}\right\}_{i=0}^{m+n-1}$ *on its diagonal.*

Theorem 2.3 (S_β)
The determinant of Sylvester's matrix $S_\beta(r,s)$ is the resultant of the Bernstein polynomials $r(x)$ and $s(x)$.

Proof.
The equations $\alpha_i^{[n-1]} r(x)$ for $0 \le i < n$ and $\alpha_i^{[m-1]} s(x)$ for $0 \le i < m$ form the matrix equation

$$S_\alpha(r,s) \begin{pmatrix} \alpha_0^{[m+n-1]} \\ \vdots \\ \alpha_{m+n-1}^{[m+n-1]} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \, ,$$

where the elements of $S_\alpha(r,s)$ include the combinatorial factors in addition to the Bernstein coefficients of $r(x)$ and $s(x)$. $T_{\alpha\beta}$ is diagonal, being equal to its transpose, thus (1.6) implies the basis conversion

$$\begin{pmatrix} \alpha_0^{[m+n-1]} \\ \vdots \\ \alpha_{m+n-1}^{[m+n-1]} \end{pmatrix} = T_{\alpha\beta}^{[m+n-1]} \begin{pmatrix} \beta_0^{[m+n-1]} \\ \vdots \\ \beta_{m+n-1}^{[m+n-1]} \end{pmatrix} .$$

This yields the conclusion

$$S_\alpha(r,s) \begin{pmatrix} \alpha_0^{[m+n-1]} \\ \vdots \\ \alpha_{m+n-1}^{[m+n-1]} \end{pmatrix} = \underbrace{S_\alpha(r,s)\, T_{\alpha\beta}^{[m+n-1]}}_{S_\beta(r,s)} \begin{pmatrix} \beta_0^{[m+n-1]} \\ \vdots \\ \beta_{m+n-1}^{[m+n-1]} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

proving that the matrix product $S_\alpha T_{\alpha\beta}$ satisfies the requirement for the Bernstein form of Sylvester's matrix. □

2.2.3 Transformation of the Sylvester's matrix between the power basis and the Bernstein basis

Below, the transformation of the Sylvester's matrix between the power basis and the Bernstein basis is developed, using a similar technique as presented in [6].

To make the notation more compact, define the following matrices:

$$X_x^{[k]} = \begin{pmatrix} 1 \\ x \\ \vdots \\ x^k \end{pmatrix}, \qquad X_\beta^{[k]} = \begin{pmatrix} \beta_0^{[k]} \\ \beta_1^{[k]} \\ \vdots \\ \beta_k^{[k]} \end{pmatrix},$$

$$F_x = f(x) X_x^{[n-1]} = \begin{pmatrix} f(x) \\ xf(x) \\ \vdots \\ x^{n-1} f(x) \end{pmatrix}, \qquad F_\beta = f(x) X_\beta^{[n-1]} = \begin{pmatrix} \beta_0^{[n-1]} f(x) \\ \beta_1^{[n-1]} f(x) \\ \vdots \\ \beta_{n-1}^{[n-1]} f(x) \end{pmatrix},$$

$$G_x = g(x) X_x^{[m-1]} = \begin{pmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{m-1} g(x) \end{pmatrix}, \qquad G_\beta = g(x) X_\beta^{[m-1]} = \begin{pmatrix} \beta_0^{[m-1]} g(x) \\ \beta_1^{[m-1]} g(x) \\ \vdots \\ \beta_{m-1}^{[m-1]} g(x) \end{pmatrix},$$

for which the equation (1.6) for the transformation matrix $T_{x\beta}$ implies [2]

$$X_x = T_{x\beta} X_\beta . \tag{2.5}$$

[2] For the transpose in (1.6), note that X_x and X_β are column vectors.

The definitions 2.2 and 2.4 of the representations $S_x(f,g)$ and $S_\alpha(f,g)$ of the Sylvester's matrix and the identity (1.3) yields the following equations:

$$\begin{pmatrix} F_x \\ G_x \end{pmatrix} = S_x(f,g)\, X_x^{[m+n-1]} \ , \tag{2.6}$$

and

$$\begin{pmatrix} F_\beta \\ G_\beta \end{pmatrix} = S_\beta(f,g)\, X_\beta^{[m+n-1]} \ . \tag{2.7}$$

On the other hand, from (2.5) we have

$$\begin{array}{rcccccl} F_x &=& f(x)\, X_x^{[n-1]} &=& f(x)\, T_{x\beta}^{[n-1]}\, X_\beta^{[n-1]} &=& T_{x\beta}^{[n-1]}\, F_\beta \ , \\ G_x &=& g(x)\, X_x^{[n-1]} &=& g(x)\, T_{x\beta}^{[m-1]}\, X_\beta^{[m-1]} &=& T_{x\beta}^{[m-1]}\, G_\beta \ , \end{array}$$

which is the same as the combined equation

$$\begin{pmatrix} F_x \\ G_x \end{pmatrix} = \begin{pmatrix} T_{x\beta}^{[n-1]} & 0_{n\times m} \\ 0_{m\times n} & T_{x\beta}^{[m-1]} \end{pmatrix} \begin{pmatrix} F_\beta \\ G_\beta \end{pmatrix} \ . \tag{2.8}$$

Now, (2.6) and (2.8) together lead to

$$S_x(f,g)\, X_x^{[m+n-1]} \ = \ \begin{pmatrix} T_{x\beta}^{[n-1]} & 0_{n\times m} \\ 0_{m\times n} & T_{x\beta}^{[m-1]} \end{pmatrix} \begin{pmatrix} F_\beta \\ G_\beta \end{pmatrix}$$

and according to (2.7)

$$S_\beta(f,g)\, X_\beta^{[m+n-1]} \ = \ \begin{pmatrix} T_{x\beta}^{[n-1]} & 0_{n\times m} \\ 0_{m\times n} & T_{x\beta}^{[m-1]} \end{pmatrix}^{-1} S_x(f,g)\, X_x^{[m+n-1]} \ .$$

Using (2.5), we further derive that

$$\left(S_\beta(f,g) - \begin{pmatrix} T_{x\beta}^{[n-1]} & 0_{n\times m} \\ 0_{m\times n} & T_{x\beta}^{[m-1]} \end{pmatrix}^{-1} S_x(f,g)\, T_{x\beta}^{[m+n-1]} \right) X_\beta^{[m+n-1]} \ = \ 0 \ .$$

Hence,

$$S_\beta(f,g) \ = \ \begin{pmatrix} T_{x\beta}^{[n-1]} & 0_{n\times m} \\ 0_{m\times n} & T_{x\beta}^{[m-1]} \end{pmatrix}^{-1} S_x(f,g)\, T_{x\beta}^{[m+n-1]} + Z(x) \ ,$$

where $Z(x)$ is an arbitrary matrix that satisfies $Z(x) X_\beta^{[m+n-1]} = 0$. Since the ranks of $\binom{F_\beta}{G_\beta}$ and $\binom{F_x}{G_x}$ match, the ranks of $S_\beta(f,g)$ and $S_x(f,g)$ must match too, so it follows that $Z(x) = 0$. Thus, the equation

$$S_\beta(f,g) \ = \ \begin{pmatrix} T_{x\beta}^{[n-1]} & 0_{n\times m} \\ 0_{m\times n} & T_{x\beta}^{[m-1]} \end{pmatrix}^{-1} S_x(f,g)\, T_{x\beta}^{[m+n-1]} \tag{2.9}$$

defines the transformation for Sylvester's matrix between the power basis and the Bernstein basis.

The matrix $T_{x\beta}$ is block diagonal [3], and this property guarantees that the transformation (2.9) is scale invariant. That is, for polynomials f, g and constants c, d the following is true:

$$\begin{aligned}
S_\beta(cf, dg) &= \begin{pmatrix} cI_n & 0_{n\times m} \\ 0_{m\times n} & dI_m \end{pmatrix} S_\beta(f,g) = \\
&= \begin{pmatrix} cI_n & 0_{n\times m} \\ 0_{m\times n} & dI_m \end{pmatrix} \begin{pmatrix} T_{x\beta}^{[n-1]} & 0_{n\times m} \\ 0_{m\times n} & T_{x\beta}^{[m-1]} \end{pmatrix}^{-1} S_x(f,g)\, T_{x\beta}^{[m+n-1]} = \\
&= \begin{pmatrix} T_{x\beta}^{[n-1]} & 0_{n\times m} \\ 0_{m\times n} & T_{x\beta}^{[m-1]} \end{pmatrix}^{-1} \begin{pmatrix} cI_n & 0_{n\times m} \\ 0_{m\times n} & dI_m \end{pmatrix} S_x(f,g)\, T_{x\beta}^{[m+n-1]} = \\
&= \begin{pmatrix} T_{x\beta}^{[n-1]} & 0_{n\times m} \\ 0_{m\times n} & T_{x\beta}^{[m-1]} \end{pmatrix}^{-1} S_x(cf, dg)\, T_{x\beta}^{[m+n-1]}
\end{aligned}$$

proving the scale invariance of the transformation.

2.3 Companion matrix resultant

Before the companion matrix is defined, see a few elementary definitions for matrices:

Definition 2.5 (row echelon form)
A matrix is in row echelon form if it satisfies the following requirements:

- *All non-zero rows are above any rows of all zeroes.*
- *The leading coefficient of a row is always to the right of the leading coefficient of the row above it.*

Note that the second requirement implies that all entries below a leading coefficient, if any, are zeroes. For examples, see [26].

Definition 2.6 (eigenvalues, eigenvectors)

- *Given an $n \times n$ matrix A, its eigenvalues and eigenvectors are the solutions of the equation $Ax = \lambda x$, where λ is the eigenvalue and $x \neq 0$ is the eigenvector.*
- *Given two $n \times n$ matrices A, B, the generalized eigenvalue problem corresponds to $Ax = \lambda Bx$.*

The eigenvalue λ and its corresponding eigenvector x form the eigenpair $\{\lambda, x\}$.

[3] A (square) matrix is block diagonal if it can be divided into smaller, diagonal square matrices.

The eigenvalues of a matrix are the roots of its characteristic polynomial, corresponding to $\det(A - \lambda I)$. As a result, the eigenvalues of a diagonal matrix, or upper triangular or lower triangular matrix, are the elements on its diagonal.
If B is non-singular and its condition number is low (that is, the numerical error of multiplication with B^{-1} is negligible), the generalized eigenvalue problem can be reduced to the eigenvalue problem $(B^{-1}A)x = \lambda x$.

Definition 2.7 (companion matrix)
A companion matrix C_p of a polynomial $p(\lambda)$ of degree n, expressed in an arbitrary basis $\phi(\lambda) = \{\phi_i(\lambda)\}_{i=0}^{n}$, is defined by

$$p(\lambda) = \det(C_p - \lambda I) = \sum_{i=0}^{n} a_i \phi_i(\lambda) \ ,$$

where the structure of C_p is defined for each basis.

Clearly, the eigenvalues of C_p are identically equal to the roots of $p(\lambda)$.

2.3.1 Companion matrix and resultant for the power basis

This section is meant as an introduction to the companion matrix resultant for the Bernstein basis. Therefore, only the most important definitions and theorems are presented.

According to the comment in [5], the companion matrix of a polynomial expressed in the power basis has the form

$$\begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-2} & -a_{n-1} \end{pmatrix}$$

for suitable $\{a_i\}_{i=0}^{n-1}$.

Theorem 2.4
Let $f(x)$ and $g(x)$ be two polynomials of the form

$$f(x) = \sum_{i=0}^{n} f_i x^i, \qquad g(x) = \sum_{i=0}^{m} g_i x^i, \qquad f_n = 1 \ .$$

If P is the companion matrix of the polynomial $f(x)$ and the eigenvalues of P are $\{\lambda_i\}_{i=1}^{n}$, then

$$\det(g(P)) = \prod_{i=1}^{n} g(\lambda_i)$$

and thus the determinant of $g(P)$ is equal to zero if and only if λ_i is a root of $g(x)$. Since the eigenvalues $\{\lambda_i\}_{i=1}^{n}$ are the roots of $f(x)$, it follows that $g(P)$ is a resultant matrix for the polynomials $f(x)$ and $g(x)$.

Proof.
Consider the matrix polynomial

$$g(P) = \sum_{j=0}^{m} g_j P^j$$

If the eigenpairs of P are $\{\lambda_i, x_i\}_{i=1}^n$, then the eigenpairs of P^j are $\{\lambda_i^j, x_i\}_{i=1}^n$, so

$$\begin{aligned} P^j x_i &= \lambda_i^j x_i \qquad \text{for} \quad i = 1, \ldots, n \ , \\ \sum_{j=0}^{m} g_j P^j x_i &= \sum_{j=0}^{m} g_j \lambda_i^j x_i \qquad \text{for} \quad i = 1, \ldots, n \ . \end{aligned}$$

Thus, the eigenvalues of $g(P)$ are

$$g(\lambda_i) = \sum_{j=0}^{m} g_j \lambda_i^j, \qquad \text{for} \quad i = 1, \ldots, n \ .$$

It is concluded that

$$\det(g(P)) = \prod_{i=1}^{n} g(\lambda_i) \ .$$

□

2.3.2 Companion matrix for Bernstein polynomials

The companion matrix is highly dependent on the basis to which the coefficients of its polynomial belong. Below, an expression for a companion matrix of a Bernstein polynomial is developed. For details, see [5], [8] and [13].

Theoretical development

Let $b_0, \ldots, b_{n-1}$ be arbitrary coefficients with no further meaning at this point. Consider the square matrices A, E, F of the order n as follows:

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -b_0 & -b_1 & -b_2 & \cdots & -b_{n-2} & -b_{n-1} \end{pmatrix},$$

$$E = \begin{pmatrix} \frac{\binom{n}{1}}{\binom{n}{0}} & 1 & 0 & \cdots & 0 \\ 0 & \frac{\binom{n}{2}}{\binom{n}{1}} & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \cdots & \frac{\binom{n}{n-1}}{\binom{n}{n-2}} & 1 \\ -b_0 & -b_1 & \cdots & -b_{n-2} & -b_{n-1} + \frac{\binom{n}{n}}{\binom{n}{n-1}} \end{pmatrix},$$

$$F = \operatorname{diag}\left[\ \frac{\binom{n}{1}}{\binom{n}{0}} \ \ \frac{\binom{n}{2}}{\binom{n}{1}} \ \cdots \ \frac{\binom{n}{n-1}}{\binom{n}{n-2}} \ \ \frac{\binom{n}{n}}{\binom{n}{n-1}} \ \right] .$$

$$E = A + F$$

The matrix A is in the form of a companion matrix of a power basis polynomial.

The analogous companion matrix for scaled Bernstein polynomials is obtained by replacing F by the identity matrix and redefining the coefficients b_i to include the combinatorial factor $\binom{n}{i}$. For the derivation in the scaled Bernstein basis, see [13].

For $\delta = 1 - \lambda$, consider the matrix

$$A - \lambda E = \begin{pmatrix} -\lambda\frac{\binom{n}{1}}{\binom{n}{0}} & \delta & 0 & \cdots & 0 & 0 \\ 0 & -\lambda\frac{\binom{n}{2}}{\binom{n}{1}} & \delta & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\lambda\frac{\binom{n}{n-1}}{\binom{n}{n-2}} & \delta \\ -b_0\delta & -b_1\delta & -b_2\delta & \cdots & -b_{n-2}\delta & -b_{n-1}\delta - \lambda\frac{\binom{n}{n}}{\binom{n}{n-1}} \end{pmatrix}.$$

The determinant

$$\det(A - \lambda E) = (-1)^n \sum_{i=0}^{n} b_i \binom{n}{i} (1-\lambda)^{n-i}\lambda^i, \qquad b_n = 1 \tag{2.10}$$

is obtained in [5] by a recursive definition of the determinant, involving the method of Horner for the nested multiplication of polynomials.

$$\det(-\lambda E) = \lim_{\lambda\to\infty} \det(A - \lambda E) = \lim_{\lambda\to\infty} (-1)^n \sum_{i=0}^{n} b_i \binom{n}{i} (1-\lambda)^{n-i}\lambda^i ,$$

and

$$\det E = \sum_{i=0}^{n} (-1)^{n-i} b_i \binom{n}{i}, \qquad b_n = 1 . \tag{2.11}$$

The condition $b_n = 1$ is a normalization constraint that is equivalent to the monic property of the characteristic polynomial of a companion matrix for a power basis polynomial. Thus, the more general implication of this condition is that the coefficient b_n of x^n of the polynomial in (2.1) is non-zero, which implies that $x_0 = 1$ is not a root of the polynomial. However, if $b_n = 0$, then a polynomial of degree $n-1$ is considered by removing the factor $(1-x)$. A similar situation arose in the substitution of (2.2) into the polynomial (2.1).

If E is non-singular, the eigenvalues of $E^{-1}A = (F+A)^{-1}A$ are identically equal to the roots of $p(\lambda)$, where, from (2.10),

$$p(\lambda) = (-1)^n \sum_{i=0}^{n} b_i \binom{n}{i} (1-\lambda)^{n-i}\lambda^i, \qquad b_n = 1, \tag{2.12}$$

and thus $E^{-1}A$ is the companion matrix of the polynomial $p(\lambda)$.

The Sherman–Morrison formula, as stated in [13], enables the inverse of E to be defined rather than computed numerically:

Theorem 2.5 (Sherman–Morrison formula)
For the square non-singular matrix P of rank n and vectors u, v of length n, the inverse of $P + uv^T$ equals

$$\left(P + uv^T\right)^{-1} = P^{-1} - \frac{1}{\tau} P^{-1} u v^T P^{-1}, \qquad \tau = 1 + v^T P^{-1} u \ .$$

The application of this formula requires the matrix E to be written in the requested form.

Let e_n be the n^{th} standard basis vector, b be the n-length vector of coefficients b_i and $C = E + e_n b^T$, as shown below:

$$e_n = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \qquad b = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}, \qquad c_{ij} = \begin{cases} \frac{\binom{n}{i}}{\binom{n}{i-1}}, & i = j \ , \\ 1, & i+1 = j \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

Now for $E = C - e_n b^T$, theorem 2.5 says that the inverse of E is given by

$$E^{-1} = C^{-1} + \frac{1}{\tau} C^{-1} e_n b^T C^{-1}, \qquad \tau = 1 - b^T C^{-1} e_n \ . \tag{2.13}$$

It is shown in [5] that the matrix $C^{-1} = D$ consists of elements

$$d_{i,\,k+i} = \begin{cases} \dfrac{(-1)^k}{\prod_{m=i}^{i+k} c_{mm}} & 1 \le i \le n, \quad 0 \le k \le n-i \ , \\ 0 & \text{otherwise} \end{cases}$$

and also that this formulation of D together with (2.13) and (2.11) leads to $\tau = \det E$. So finally, (2.13) can be written as

$$E^{-1} = D + \frac{1}{\det E} D e_n b^T D \ . \tag{2.14}$$

This allows the companion matrix $E^{-1}A$ of the polynomial $p(\lambda)$ to be computed.

Computational implementation

As stated in [13], a companion matrix of a power basis polynomial has the form of the matrix A. According to (2.14), the companion matrix of a Bernstein basis polynomial has the form

$$M \ = \ E^{-1} A \ = \ \left(I + \frac{1}{\det E} D e_n b^T\right) DA \ = \ HDA \ .$$

Hence, the computational cost for the Bernstein basis is higher. Since the closed form expressions for the elements A, D, b and e_n have been developed, the cost of constructing M can be reduced. In particular, D is an upper triangular matrix and

$e_n b^T$ is a matrix with only the last row being non-zero, filled with the members of b. It follows that the elements of the matrix H can be explicitly defined. Similarly, the matrix DA is the product of an upper triangular matrix and a companion matrix in the power basis form, and the elements of this product can also be defined. The computation of M can therefore be reduced to two matrix multiplications and the addition of the identity matrix.

2.3.3 Companion matrix resultant for Bernstein polynomials

The following theorem shows how the companion matrix presented in the previous section may be used to construct a resultant matrix for two Bernstein polynomials [5]. The same method, described in [13], is used to construct a resultant matrix for scaled Bernstein polynomials.

Theorem 2.6
Let $r(x)$ and $s(x)$ be two Bernstein polynomials of the form

$$r(x) = \sum_{i=0}^{n} r_i \beta_i^{[n]}, \qquad s(x) = \sum_{i=0}^{m} s_i \beta_i^{[m]}, \qquad r_n = 1 \; .$$

If M is the companion matrix of the polynomial $r(x)$ and the eigenvalues of M are $\{\lambda_i\}_{i=1}^n$, then

$$\det(s(M)) = \prod_{i=1}^{n} s(\lambda_i) \; ,$$

and thus the determinant of $s(M)$ is equal to zero if and only if λ_i is a root of $s(x)$. Since the eigenvalues $\{\lambda_i\}_{i=1}^n$ are the roots of $r(x)$, it follows that $s(M)$ is a resultant matrix for the polynomials $r(x)$ and $s(x)$.

The condition $r_n = 1$ denies only the polynomials having $r_n = 0$, because dividing the coefficients by the non-zero r_n does not change the roots. Therefore, the assumption is equivalent to that $x_0 = 1$ is not a root of r. Dealing with the root $x_0 = 1$ has already been considered in the paragraph right after (2.11).

Proof.
Consider the matrix polynomial

$$s(M) \;=\; \sum_{j=0}^{m} s_j \, \beta_j^{[m]}(M) \;=\; \sum_{j=0}^{m} s_j \binom{m}{j} (I-M)^{m-j} M^j \; .$$

If the eigenpairs of M are $\{\lambda_i, x_i\}_{i=1}^n$, then the eigenpairs of $(I-M)^{m-j}M^j$ are $\{\,(1-\lambda_i)^{m-j}\lambda_i^j,\; x_i\,\}_{i=1}^n$, so

$$\begin{aligned} (I-M)^{m-j}M^j x_i &= (1-\lambda_i)^{m-j}\lambda_i^j x_i, && \text{for} \quad i = 1,\ldots,n \; , \\ \beta_j^{[m]}(M)\, x_i &= \beta_j^{[m]}(\lambda_i)\, x_i, && \text{for} \quad i = 1,\ldots,n \; . \end{aligned}$$

It follows that

$$\sum_{j=0}^{m} s_j\, \beta_j^{[m]}(M)\, x_i \;=\; \sum_{j=0}^{m} s_j\, \beta_j^{[m]}(\lambda_i)\, x_i, \qquad \text{for} \quad i = 1, \dots, n$$

and thus the eigenvalues of $s(M)$ are

$$s(\lambda_i) = \sum_{j=0}^{m} s_j\, \beta_j^{[m]}(\lambda_i), \qquad i = 1, \dots, n\ .$$

We conclude that

$$\det(s(M)) = \prod_{i=1}^{n} s(\lambda_i)\ .$$

□

It is possible to obtain the degree and coefficients of the GCD of $r(x)$ and $s(x)$ from $s(M)$ as declared by the following theorem:

Theorem 2.7
Let $w(x)$ be the GCD of $r(x)$ and $s(x)$. Then

- *the degree of $w(x)$ is equal to n decreased by the rank of $s(M)$,*
- *the coefficients of $w(x)$ are proportional to the last non-zero row of $s(M)$ after it has been reduced to row echelon form.*

Let $\widetilde{s(M)}$ denote the row echelon form of $s(M)$.
To determine the GCD from the polynomial constructed from the last row of $\widetilde{s(M)}$, a factor of the form $(1-x)x^q$ must be deleted. This factor arises because M, which is of order $n \times n$, is a companion matrix of a polynomial of order n. This polynomial is defined by $n+1$ basis functions, and the coefficient b_n of x^n does not occur in M but arises from the term $-\lambda I$ in the expression $M - \lambda I$. It follows that M contains only the coefficients $\{b_i\}_{i=0}^{n-1}$ of the basis functions $\{\beta_i^{[n]}\}_{i=0}^{n-1}$, and thus a factor of the form $(1-x)x^q$ arises in the GCD.

For the scaled Bernstein basis, such manipulation is not required because the basis stable polynomial functions $\{\alpha_i\}_{i=0}^{n}$ do not contain combinatorial factors. Since $\alpha_i^{[n]} = (1-x)\alpha_i^{[n-1]}$, the GCD can be constructed directly from the last row of $\widetilde{s(M)}$, if, of course, M was constructed using the scaled Bernstein basis.

Below, the example of computing the GCD is taken from [5].

Example

Consider the Bernstein polynomials

$$\begin{aligned} r(x) &= 3\,\beta_0^{[3]} - \tfrac{5}{6}\,\beta_1^{[3]} - \tfrac{1}{2}\,\beta_2^{[3]} + 1\,\beta_3^{[3]} && \text{roots: } \tfrac{1}{2},\ \tfrac{2}{3},\ 3 \\ s(x) &= 2\,\beta_0^{[2]} - \tfrac{3}{2}\,\beta_1^{[2]} + 1\,\beta_2^{[2]} && \text{roots: } \tfrac{1}{2},\ \tfrac{2}{3} \end{aligned}$$

the GCD of which is $s(x)$.
The companion matrix of $r(x)$ and the resultant matrix are:

$$M = -\frac{1}{3}\begin{pmatrix} -3 & \frac{5}{6} & -\frac{1}{3} \\ 9 & -\frac{11}{2} & 1 \\ -9 & \frac{11}{2} & -4 \end{pmatrix}, \quad s(M) = \begin{pmatrix} 8 & -4 & \frac{4}{3} \\ -36 & 18 & -6 \\ 54 & -27 & 9 \end{pmatrix} \sim \begin{pmatrix} 6 & -3 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Since the rank of $\widetilde{s(M)}$ is 1, the degree of the GCD of the two polynomials is $3 - 1 = 2$. The GCD is calculated from the first row of this matrix:

$$6\beta_0^{[3]} - 3\beta_1^{[3]} + 1\beta_2^{[3]} \;=\; 3(1-x)\left(2\beta_0^{[2]} - \frac{3}{2}\beta_1^{[2]} + \beta_2^{[2]}\right) \;=\; 3(1-x)s(x)$$

The factor $(1-x)$ is ignored, and the GCD is therefore proportional to $s(x)$.

Now, start with the other polynomial. The companion matrix of $s(x)$ and the resultant matrix are:

$$N = -\frac{1}{3}\begin{pmatrix} \frac{1}{3} & \frac{1}{12} \\ -\frac{2}{3} & \frac{5}{6} \end{pmatrix}, \quad r(N) = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

The rank of $r(N)$ is zero, so the degree of the GCD is $2 - 0 = 2$. Since $s(x)$ is also of degree two, the GCD is proportional to $s(x)$.

Numerical issues

The issue of the implementation of resultants in a floating-point environment is not a trivial problem because the degree of the GCD is equal to the rank deficiency of the resultant matrix, but the rank of the matrix is not stable to compute ina floating-point environment.

The difficulties of computing resultants in a floating-point environment arise from the requirement that the determinant of a result matrix be exactly zero for the polynomials to have a non-constant common divisor. Since the coefficients of a polynomial may be multiplied by an arbitrary non-zero constant without changing its roots, the determinant of the resultant matrix may be scaled arbitrarily, and thus a non-zero determinant does not yield any information on the proximity of the roots of the polynomials.

The numerical condition of the resultant matrix $s(M)$ is investigated in [5] and [8].

2.3.4 Transformation of the companion matrix resultant between the power basis and the Bernstein basis

A simple way to transform a resultant matrix from the power basis to the Bernstein basis is by the transformation of each polynomial, and then the employment of theorem 2.6. This strategy is not satisfactory because each polynomial is treated independently, but a resultant matrix contains cross product terms of the form $r_i s_j$

and the error analysis of the transformation of each polynomial does not consider error terms of such form [12]. This section is concerned with a basis transformation of a function of the polynomials, that is, their resultant, and not a basis transformation of the polynomials.

The transformation is defined by the following theorem [3]:

Theorem 2.8
Let $f(x)$ and $g(x)$ be the power basis representations of the Bernstein polynomials $r(x)$ and $s(x)$, respectively, holding [4]

$$\begin{aligned} \sum_{i=0}^{n} f_i x^i &= f(x) = r(x) = \sum_{i=0}^{n} r_i \beta_i^{[n]} , \\ \sum_{i=0}^{m} g_i x^i &= g(x) = s(x) = \sum_{i=0}^{m} s_i \beta_i^{[m]} . \end{aligned} \tag{2.15}$$

If P is a companion matrix of $f(x)$, M is a companion matrix of $r(x)$ and

$$M = B^{-1} P B , \tag{2.16}$$

then $g(P)$ and $s(M)$ are resultant matrices that are related by

$$s(M) = B^{-1} g(P) B . \tag{2.17}$$

Proof.
The transformation (2.16) between the companion matrices M and P enables the transformation between the resultant matrices $s(M)$ and $g(P)$ to be derived:

$$\begin{aligned} I - M &= I - B^{-1}PB = B^{-1}(I-P)B \\ (I-M)^{m-i} &= B^{-1}(I-P)^{m-i}B \\ (I-M)^{m-i}M^i &= B^{-1}(I-P)^{m-i}P^iB \end{aligned}$$

Then,

$$\begin{aligned} s(M) &= \sum_{i=0}^{m} s_i \binom{m}{i} (I-M)^{m-i} M^i = \\ &= \sum_{i=0}^{m} s_i \binom{m}{i} B^{-1}(I-P)^{m-i}P^iB = \\ &= B^{-1} \left[\sum_{i=0}^{m} s_i \binom{m}{i} (I-P)^{m-i}P^i \right] B = \\ &\overset{(2.15)}{=} B^{-1} \left[\sum_{i=0}^{m} g_i P^i \right] B = B^{-1} g(P) B . \end{aligned}$$

□

The numerical condition of the transformation equation (2.17) is described in [3].

[4] Note that different symbols are used for the same polynomial in different basis to indicate which coefficients are meant.

Theorem 2.9
Using the same assumptions for f, g, r, s, P and M from theorem 2.8, we have for $M = B^{-1}PB$, the elements of $B = (b_{ij})$ and $B^{-1} = (b_{ij}')$ are

$$b_{ij} = \begin{cases} \frac{n-j+1}{n}\frac{\binom{j-1}{i-1}}{\binom{n-1}{i-1}}, & i \le j \\ 0, & i > j \end{cases}$$

$$b_{ij}' = \begin{cases} (-1)^{j-i}\binom{n}{i-1}\binom{n-i}{j-i}, & i \le j \\ 0, & i > j \end{cases}$$

$$i, j = 1, \ldots, n \,.$$

The matrices B and B^{-1} are upper triangular. Moreover, the matrix B is a totally non-negative matrix (all its minors of all orders are non-negative). The theorem is proved in [3], including a deep investigation of properties of the matrices M and B.

2.4 Other popular resultant matrices

In addition to Sylvester's matrix, the Bézout matrix [5] is also frequently referenced in many publications. To give a brief description, this section presents two of its representations for the power basis.

2.4.1 Bézout's formulation

The following is one of the possible expressions of the Bézout matrix [14]:

Definition 2.8 (Bézout's resultant)
The resultant of the polynomials

$$f(x) = \sum_{i=0}^{m} a_i x^i \quad \textit{and} \quad g(x) = \sum_{i=0}^{n} b_i x^i$$

is the determinant of matrix R of order $l = \max(m, n)$, defined as follows:

$$R = \left(r_{ij} \right), \qquad i, j = 1, \ldots, l \,,$$

$$r_{ij} = \sum_{k\,=\,\max(l-i,\,l-j)+1}^{\min(l,\,2l+1-i-j)} v_{k,\,2l+1-i-j-k} \,,$$

$$v_{i,j} = \begin{vmatrix} a_i & a_j \\ b_i & b_j \end{vmatrix} = a_i b_j - a_j b_i \,.$$

[5] also called the Bézoutian matrix

Most of the definitions in the literature define the Bézout matrix for polynomials of the same degree. According to [24], if $m = n$, the definition can be reformulated to

$$r_{ij} = \sum_{\substack{p,\,q \\ p>\max(n-i,\,n-j) \\ p+q=2n-i-j+1}} v_{pq} \; .$$

[24] also presents the Bézout resultant for Bernstein polynomials, obtained by the parameter substitution (2.2), and shows that the number of common roots and the roots themselves can be computed from the Bézout matrix.

For polynomials of roughly the same degree, the Bézout matrix is half-sized in comparison to the Sylvester matrix. Thus, the Bézout determinant is generally faster to compute. But whereas the non-zero entries of the Sylvester resultant are just the coefficients of the original two polynomials, the entries of the Bézout resultant are much more complicated expressions in these coefficients. For degree n, standard techniques based on explicit formulas require $\mathrm{O}(n^3)$ additions and $\mathrm{O}(n^3)$ multiplications to compute the entries of the Bézout matrix. A recursive algorithm for computing these entries, transforming the Sylvester matrix to the Bézoutian, requires only $\mathrm{O}(n^2)$ additions and $\mathrm{O}(n^2)$ multiplications [16].

2.4.2 Cayley's formulation

A nice derivation of Bézout's resultant is due to Cayley [20]. Having $f(x)$ of degree m and $g(x)$ of degree n, without loss of genereality, we assume that $m \geq n$. Let us consider the bivariate expression

$$p(x,w) = \frac{f(x)g(w) - f(w)g(x)}{x - w} \tag{2.18}$$

which is a polynomial of bidegree $(m-1,\, m-1)$ in x and w. This is proved by some algebraic manipulation presented in [10]. Consider the following representation:

$$p(x,w) = p_0(x) + p_1(x)w + p_2(x)w^2 + \ldots + p_{m-1}(x)w^{m-1}$$

where $p_i(x)$ is a polynomial of degree $m-1$ in x. The polynomials $p_i(x)$ can be written as follows:

$$\begin{pmatrix} p_0(x) \\ p_1(x) \\ \vdots \\ p_{m-1}(x) \end{pmatrix} = \underbrace{\begin{pmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,m-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m-1,0} & p_{m-1,1} & \cdots & p_{m-1,m-1} \end{pmatrix}}_{P} \begin{pmatrix} 1 \\ x \\ \vdots \\ x^{m-1} \end{pmatrix} . \tag{2.19}$$

Let us assume that $x = x_0$ is a common root of the two polynomials, thus, $p(x_0, w) = 0$ for all w. As a result, $p_i(x_0) = 0$ for $0 \leq i \leq m$. This condition corresponds to the fact that P is singular and $v = [1 \; x_0 \; x_0^2 \; \ldots \; x_0^{m-1}]^T$ is a vector in the kernel of the matrix P. In other words, if we substitute $x = x_0$ in (2.19), the product of P and the right-hand side vector v is the null vector:

$$\begin{pmatrix} p_{0,0} & p_{0,1} & \cdots & p_{0,m-1} \\ p_{1,0} & p_{1,1} & \cdots & p_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m-1,0} & p_{m-1,1} & \cdots & p_{m-1,m-1} \end{pmatrix} \begin{pmatrix} 1 \\ x_0 \\ \vdots \\ x_0^{m-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Therefore the determinant of P is the resultant of $f(x)$ and $g(x)$.

Cayley's formulation highlighted above is used for implicitizing parametric curves and eliminating a variable from a pair of bivariate algebraic equations representing algebraic plane curves. It can also be used to implicitize Bézier curves [24]. Moreover, this formulation of the Bézout matrix can be extended to the case of three bivariate polynomials, leading to the Dixon resultant [10].

Chapter 3

Intersection

The problems of computing the intersection of parametric and algebraic curves are fundamental to geometric and solid modeling. Parametric curves, like B-splines and Bézier curves, are extensively used in the modeling systems and algebraic plane curves are becoming popular as well.

It is universally recognized that the parametric representation is best suited for generating points along a curve or surface, whereas the implicit representation is most convenient for determining whether a given point lies on a specific curve or surface [25]. This motivates the search for a means of converting from one representation to the other.

3.1 Definition of parametric and algebraic curves

3.1.1 Parametric curves

Definition 3.1 (barycentric combination)
Let $\{P_i = [X_i, Y_i]\}_{i=0}^n$ *be points in the plane and* $\{c_i\}_{i=0}^n$ *be constants satisfying*

$$c_i \geq 0, \quad i = 0, \dots, n \; ,$$
$$\sum_{i=0}^{n} c_i = 1 \; .$$

Then the point

$$P \;=\; [X, Y] \;=\; \left[\sum_{i=0}^{n} c_i X_i, \; \sum_{i=0}^{n} c_i Y_i \right] \;=\; \sum_{i=0}^{n} c_i P_i$$

is the barycentric combination of the points $\{P_i\}_{i=0}^n$ *with barycentric coordinates* $[c_0, \dots, c_n]$*. Analogous definition holds for higher dimensions.*

The barycentric combination always lies in the convex hull of the control points. This property becomes more obvious after transforming the $n+1$ barycentric coordinates to the system with the center P_k and n basis vectors $\{\, P_i - P_k \mid 0 \leq i \leq n, \; i \neq k \,\}$.

Definition 3.2 (rational Bézier curve)
A (uniform) Bézier curve is of the form

$$P(t) \;=\; \sum_{i=0}^{n} P_i \beta_i^{[n]}(t), \qquad 0 \le t \le 1 \tag{3.1}$$

and a rational Bézier curve is of the form

$$P(t) \;=\; \frac{\sum_{i=0}^{n} w_i P_i \beta_i^{[n]}(t)}{\sum_{i=0}^{n} w_i \beta_i^{[n]}(t)}, \qquad 0 \le t \le 1 \tag{3.2}$$

where P_i are the control points of the curve, w_i is the weight of the control point P_i (such that the denominator of (3.2) is non-zero), and $\beta_i^{[n]}(t)$ are the Bernstein polynomials from definition 1.2.

Both of the two formulations can be transformed to the other, because both are represented by a polynomial of the given degree. Uniform Bézier curves correspond to rational Bézier curves with equal weights (results from theorem 1.2), and the polynomials of rational Bézier curves are implicitly in the form of a uniform curve. Thus, for a suitable transformation of the control points, the two definitions are treated equivalent.

Other rational formulations like B-splines can be converted into a series of Bézier curves by knot insertion algorithms (see the reference in [20]). Thus, the problem of intersecting parametric polynomial curves can be reduced to intersecting Bézier curves.

A Bézier curve is described by its corresponding control polygon. The curve starts at the first control point (P_0) and ends at the last one (P_n). The lines P_0P_1 and $P_{n-1}P_n$ are the tangents of the curve at the two ends. The most important property in term of the intersection problem, stated by the theorem below, is that the curve is bounded by the convex hull of the control polygon. Several other important properties are known and described in the literature of geometry and computer graphics [21].

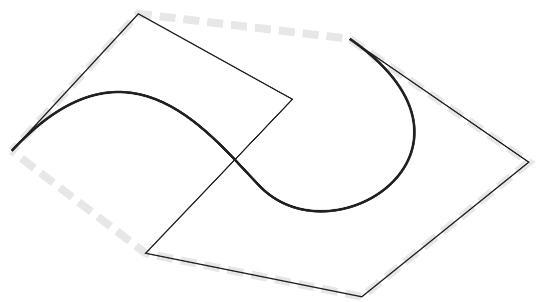
Theorem 3.1 (convex hull property)
A uniform Bézier curve, or a rational Bézier curve with positive weights, is always contained in the convex hull of its control points.

Proof.
Let $W(t) = \sum_{i=0}^{n} w_i \beta_i^{[n]}(t)$. In (3.2) the coefficients of P_i are $\frac{1}{W(t)} w_i \beta_i^{[n]}(t)$, the sum of which is

$$\sum_{i=0}^{n} \frac{1}{W(t)} w_i \beta_i^{[n]}(t) \;=\; \frac{\sum_{i=0}^{n} w_i \beta_i^{[n]}(t)}{\sum_{i=0}^{n} w_i \beta_i^{[n]}(t)} \;=\; 1\,.$$

Further, the product $w_i \beta_i^{[n]}(t)$ is non-negative, since so are the Bernstein polynomials on $[0,1]$. Therefore, according to definition 3.1, the points of the curve are

Figure 3.1: The convex hull property of Bézier curves



The Bézier curve is always contained in the convex hull of its control points.

barycentric combinations of the control points. Since barycentric combinations are always contained in the convex hull of the control points, the same is true for the Bézier curve. □

Theorem 3.1 implies that that the intersection of the convex hull of two Bézier curves is a necessary condition for the intersection of the curves.

3.1.2 Algebraic curves

Definition 3.3 (algebraic curve)
Let $f(x, y)$ be a polynomial [1]*. Let $Z(f) = \{ [x, y] \in \mathcal{R}^2 \mid f(x, y) = 0 \}$. The set $Z(f)$ is called an algebraic plane curve, the one defined by the polynomial f.*

From this point, the polynomial of the algebraic curve and the curve itself (in terms of geometric representation) are considered to be equivalent.

The problem of intersection corresponds to the computing of the common points on curves in a particular domain. The problem is equivalent to finding the roots of the system

$$\begin{aligned} f(x, y) &= 0\ , \\ g(x, y) &= 0\ . \end{aligned} \tag{3.3}$$

For determining the number of intersections, a simple version of Bézout's theorem is used [9]:

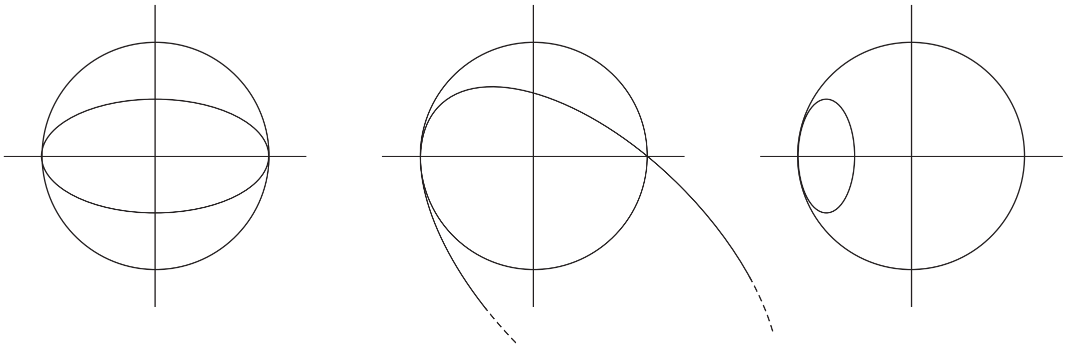
Theorem 3.2 (Bézout)
Let $f, g \in \mathcal{R}[x, y]$ be polynomials of bidegree (m, n). If f and g have no common factor of degree > 0, then there exist at most mn solutions of the system (3.3).

Written alternatively, the theorem says that if the curves have no component in common, then they intersect at mn or less points, counted properly with respect

[1] The polynomials for algebraic curves are most commonly expressed in the power basis.

to multiplicity. Figure 3.2 based on [26] shows three pairs of curves with several multiplicity of intersection points.

Figure 3.2: Intersection points with several multiplicity



Intersection of the circle $f(x,y)$ with the curves $g_i(x,y)$

$$f(x,y) = x^2 + y^2 - 1$$

left: *two intersections of multiplicity 2,* $g_2(x,y) = x^2 + 4y^2 - 1$

middle: *an intersection of multiplicity 3,* $g_3(x,y) = 6x^2 + 6xy + 5y^2 + 6y - 5$

right: *an intersection of multiplicity 4,* $g_4(x,y) = 4x^2 + y^2 + 6x + 2 = 0$

3.1.3 Numerical versus algebraic computations

Suppose we are given two parametric curves $f(u)$ and $g(v)$ and we want to find the intersection points, that is, pairs of parameter values (u_0, v_0) such that $f(u_0) = g(v_0)$. We consider 2D curves, so we can write $f(u) = [f_x(u), f_y(u)]^T$ and $g(v) = [g_x(v), g_y(v)]^T$. Then u_0 and v_0 satisfy

$$f_x(u_0) - g_x(v_0) = 0\ ,$$
$$f_y(u_0) - g_y(v_0) = 0\ .$$

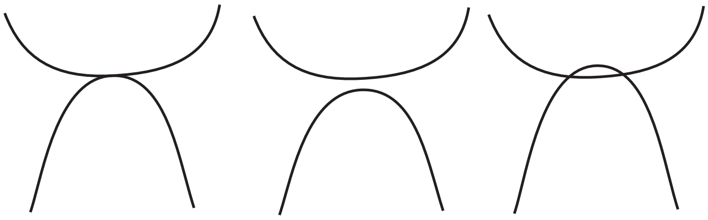
Generally, there are two approaches to solve such polynomial equations.

1. Algebraic approach using exact arithmetic. This approach is robust, but slow.
2. Numerical approach, using approximate calculations with machine accuracy. This approach is not robust, as illustrated in figure 3.3.

3.2 Algebraic approach

In this section we use elimination theory and express the resultant of the equations of intersection as a matrix determinant. The matrix itself rather than its symbolic determinant, a polynomial, is used as the representation. The problem of intersection is reduced to computing the eigenvalues and eigenvectors of a numeric matrix. The main advantage of this approach lies in its efficiency and robustness [7].

Figure 3.3: Numerical approach is not robust



left: *two curves with one intersection point*
middle, right: *after small perturbation, two or zero intersections can occur*

3.2.1 Implicitizing parametric curves

Definition 3.4 (proper parametrization)
A parametric curve has proper parametrization if the points of the curve have unique preimages, with the exception of finite number of points.

Given a parametric polynomial curve of degree m, we express it in homogeneous form:

$$P(t) \;=\; [\, x(t),\; y(t),\; w(t) \,] \;.$$

For example, a rational Bézier curve of degree n with control points $\{P_i = [X_i, Y_i]\}_{i=0}^m$ and weights $\{w_i\}_{i=0}^m$ can be expressed as

$$P(t) \;=\; \left(\sum_{i=0}^{m} w_i X_i \beta_i^{[m]}(t),\; \sum_{i=0}^{m} w_i Y_i \beta_i^{[m]}(t),\; \sum_{i=0}^{m} w_i \beta_i^{[m]}(t) \right) .$$

We assume that $P(t)$ has proper parametrization and that the GCD of $x(t), y(t), w(t)$ is a constant. To implicitize the curve we consider the following system of equations:

$$\begin{aligned} f(t):&\quad Xw(t) - x(t) = 0 \;, \\ g(t):&\quad Yw(t) - y(t) = 0 \;. \end{aligned} \tag{3.4}$$

Consider them as polynomials in t and let X, Y be indeterminates. The implicit representation corresponds to the resultant of (3.4), expressed below.

The computation of the resultant matrix involves symbolic computation. Let

$$\begin{aligned} P(t,s) \;&=\; \frac{f(t)g(s) - f(s)g(t)}{t-s} \;= \qquad\qquad (3.5) \\ &=\; X\frac{w(s)y(t) - w(t)y(s)}{t-s} + Y\frac{w(s)x(t) - w(t)x(s)}{t-s} + \frac{x(t)y(s) - y(s)x(t)}{t-s} \;. \end{aligned}$$

Just as in (2.18), each term of the form (3.5) corresponds to a polynomial and can be expressed as product of matrices and vectors, as shown in (2.19). Thus,

$$P(t,s) \;=\; \begin{pmatrix} 1 \\ s \\ \vdots \\ s^{m-1} \end{pmatrix} \Big(\underbrace{XM_1 + YM_2 + M_3}_{M} \Big) \begin{pmatrix} 1 \\ t \\ \vdots \\ t^{m-1} \end{pmatrix} ,$$

where M_1, M_2 and M_3 are $m \times m$ matrices whose entries are reals. The implicit representation of the curve is given as the determinant of the matrix M, where M is defined as

$$M = XM_1 + YM_2 + M_3 . \tag{3.6}$$

An algorithm for computation of M is presented in [20].
See [25] for another guide for implicitization, including implicitization of surfaces.

3.2.2 Intersecting parametric curves

Given the two parametric polynomial curves $P(t)$ and $Q(u)$ of degree m and n, respectively, the intersection algorithm proceeds by implicitizing $P(t)$ and obtaining an $m \times m$ matrix M of the form (3.6), whose entries are linear combinations of indeterminates X, Y. The second parametrization $Q(u) = (\overline{x}(u), \overline{y}(u), \overline{w}(u))$ is substituted into the matrix formulation (3.6) as

$$X = \frac{\overline{x}(u)}{\overline{w}(u)}, \qquad Y = \frac{\overline{y}(u)}{\overline{w}(u)} .$$

The entries of the resulting matrix are rational functions in terms of u and we multiply them by $\overline{w}(u)$ to obtain the matrix $M(u)$ with polynomial entries. It follows from the properties of the implicit representation and resultants that the intersection points correspond to the roots of

$$\det M(u) = 0 .$$

The problem of computing roots of the above equation can be reduced into an eigenvalue problem. Further details see in [20].

3.2.3 Intersecting algebraic curves

In this section we consider the intersertion of two algebraic plane curves, represented as zeros of polynomials $f(x, y)$ and $g(x, y)$ of degree m and n, respectively. Let the points of the intersection be $[x_i, y_i]$, $i = 1 \ldots mn$. To simplify the problem, we compute the projection of these points on the x-axis. Algebraically, the projection corresponds to the computing the resultant of $f(x, y)$ and $g(x, y)$ by treating them as polynomials in y and expressing the coefficients as polynomials in x. The resultant $r(x)$ is a polynomial of degree mn.

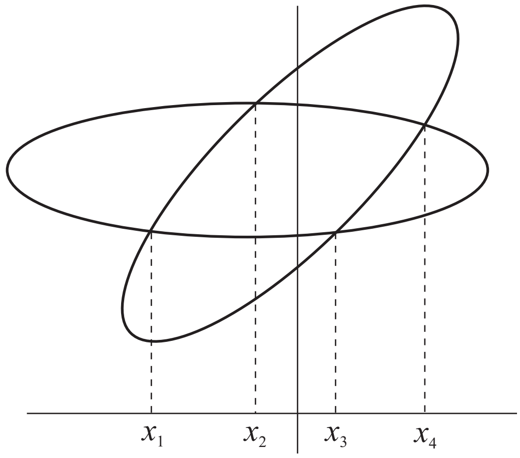
One such case corresponding to the intersection of two ellipses has been shown in figure 3.4. In this case the resultant is a polynomial $r(x)$ of degree 4 in x, such that $r(x_i) = 0$ for $i = 1, 2, 3, 4$. Thus, given $r(x)$, the problem of intersection reduces to finding its roots.

Let us express $f(x, y)$ and $g(x, y)$ as polynomials in y with their coefficients as polynomials in x. That is,

$$\begin{aligned} f(x, y) &= f_0(x) + f_1(x)y + \ldots + f_m(x)y^m , \\ g(x, y) &= g_0(x) + g_1(x)y + \ldots + g_n(x)y^n , \end{aligned}$$

where $f_i(x)$ is a polynomial of degree mi and $g_j(x)$ is a polynomial of degree $n-j$. The problem of intersection corresponds to computing the common roots of f and g by one of the available resultant methods from chapter 2.

Figure 3.4: Intersection of algebraic curves: ellipses



The projection on the x-axis corresponds to expressing the ellipses as polynomials in y with coefficients that are polynomials in x, leading to the resultant $r(x)$. The values x_1, x_2, x_3, x_4 are the roots of $r(x)$.

3.3 Numerical approach

Let us see a numerical solution for the intersection problem [7]. We start with a simple case to illustrate some general paradigms in the intersection algorithms. After computing the intersection of a Bézier curve with a line, we show how to compute the intersections of two curves. In addition, a method for surface intersection is also briefly presented.

3.3.1 Solving a non-linear equation

Consider the non-linear equation

$$f(u) = 0 \tag{3.7}$$

where $f(u)$ is a polynomial. Several geometric problems, including the curve intersection problem, can be reduced to this equation. Unfortunately, for degrees higher than five, there are no explicit formulas to express explicitly the solutions. Even for cubic equations, concerning Bézier curves of degree 3 that are used so frequently, the formula is relatively complex and not that easy to evaluate numerically in a stable way. In practice, iterative methods [17] are typically used to solve equations of degree as low as three. Newton's method is probably the most known iterative method used for solving non-linear equations.

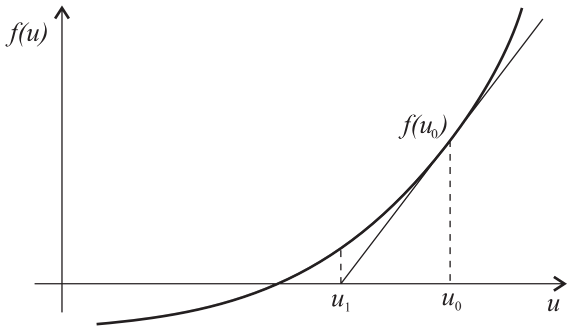
Newton's method

Newton's method of solving $f(u) = 0$ for an arbitrary continuous function $f(u)$ consists of picking up an initial value u_0 and applying the iterative formula

$$u_{n+1} = u_n - \frac{f(u_n)}{f'(u_n)}$$

unless the required precision is achieved. The precision is expressed by the means of the error tolerance $\varepsilon > 0$. When such u_n is reached for which $|u_n - u_{n-1}| < \varepsilon$, the value is considered to be the root of the equation within the tolerance. Figure 3.5 illustrates how the method works. The exact algorithm is shown in figure 3.6.

Figure 3.5: Newton's method



$$u_1 = u_0 - \frac{f(u_0)}{f'(u_0)}$$

Figure 3.6: Newton's method – algorithm

input: $f, f', u_0, \varepsilon, steps$

1. if $steps \leq 0$ then fail
2. if $f'(u_0) = 0$ then fail
3. $u_1 := u_0 - f(u_0)/f'(u_0)$
4. if $|u_1 - u_0| < \varepsilon$ then return u_1
5. $steps := steps - 1$
6. $u_0 := u_1$
7. goto 1

Newton's method is easy to implement, but it has potential problems:

- It is not guaranteed to converge, if f is not twice differentiate, or the initial value is too far from the root, or $f'(0) = 0$ at the root.
- It finds one solution, not all possible solutions.

For other methods of solving non-linear equations, as well as for a more detailed description of the Newton's method, see [17].

3.3.2 Bézier clipping

We introduce a method for solving (3.7) called *Bézier clipping* which guarantees convergence and finds all solutions under certain conditions [7]. Since $f(u)$ is a polynomial, it can be expressed in the Bernstein basis. Let $\{b_i\}_{i=0}^{n}$ be the coefficients of the Bernstein form:

$$f(u) \;=\; \sum_{i=0}^{n} b_i \beta_i^{[n]}(u) \; .$$

We will add one dimension, in which the curve will progress linearly. This leads to the following form of (3.8):

$$\sum_{i=0}^{n} \left[\frac{i}{n}, b_i\right] \beta_i^{[n]}(u) \;=\; [u, 0] \; .$$

The identity for the first component can be directly verified by recalling the definition of $\beta_i^{[n]}$ (definition 1.2). So, instead of solving the original equation, we look for the intersecting points of the newly constructed curve with u-axis.

Initially, we know that the u value of the intersection points, if it exists, is inside the interval $[0, 1]$. Using theorem 3.1 about the convex hull property of the Bézier curve, we observe that the intersection points have to be in the intersection of the convex hull of the control points and the u-axis. If the convex hull and the u-axis does not intersect, then the equation (3.7) does not have a solution within the inspected interval.

Suppose the convex hull intersects the u-axis at two points with the corresponding parameter values u_1, u_2, such that $0 \leq u_1 \leq u_2 \leq 1$. Then, all the intersection points of the Bézier curve and u-axis will be inside $[u_1, u_2]$. Now we subdivide the curve into three segments, with u ranging in $[0, u_1]$, $[u_1, u_2]$, and $[u_2, 1]$, respectively. The segments $[0, u_1]$ and $[u_2, 1]$ can be safely discarded. The process is iterated by replacing $[0, 1]$ by $[u_1, u_2]$, until the required precision $u_2 - u_1 < \varepsilon$ is achieved. Figure 3.7 shows the first step of the iteration.

For single intersection, the convergence is implied by the convex hull property. In case that the difference $u_2 - u_1$ converges to a nonzero value, the curve has multiple intersections with the line, because of multiple roots of (3.7). Then, divide the interval into two parts and search for an intersection in both of them.

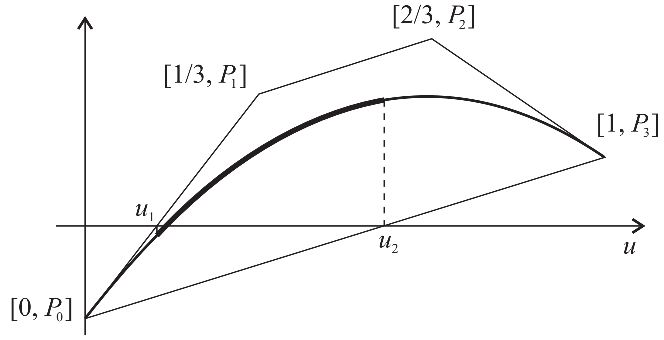
3.3.3 Intersection of a Bézier curve with a line

Let $ax + by + c = 0$ be the implicit equation of the line. In case that the line is expressed in parametric form, the constants a, b, c can be evaluated as follows. Let the parametric form of the line be

$$L(t) = Q + \vec{r}t \; .$$

Here, Q is a fixed point on the line and $\vec{r}$ is the direction vector. The normal of the line is $\vec{n} = [-r_y, r_x]$. The implicit equation of the line for the non-fixed point

Figure 3.7: Bézier clipping



The one-dimensional Bézier curve with control points $\{P_0, P_1, P_2, P_3\}$ *after extension to 2D. The u-axis is searched for intersections* u_1, u_2 *with the convex hull of the control points.*

$L = [x, y]$ can be written as $\vec{n}.(L - Q) = 0$, or

$$-r_y x + r_x y + (r_y Q_x - r_x Q_y) = 0 \ ,$$

where $a = -r_y$, $b = r_x$ and $c = r_y Q_x - r_x Q_y$.

Let the Bézier curve be expressed by the polynomial $f(u) = [f_x(u), f_y(u)]$. The coordinates x, y of the intersection point of the curve and the line will be the solution of the system

$$\begin{aligned} f_x(u) &= x \ , \\ f_y(u) &= y \ , \\ ax + by + c &= 0 \ . \end{aligned}$$

Combining these equations, we obtain a single equation for u:

$$a f_x(u) + b f_y(u) + c \ = \ 0 \ . \tag{3.8}$$

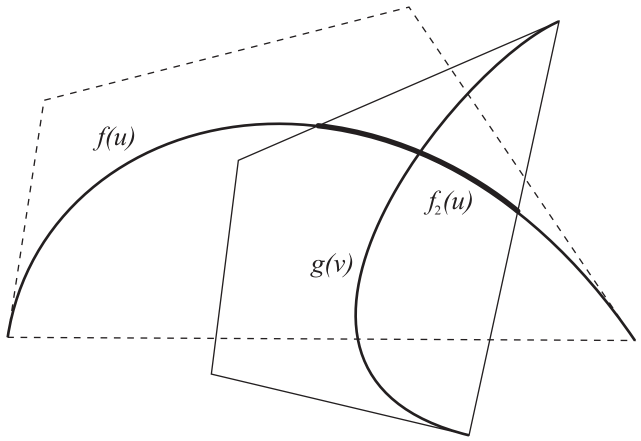
Thus, the problem of intersection of a Bézier curve with a line was reduced to this polynomial equation, which can be solved by Bézier clipping.

3.3.4 Intersection of curves

Suppose we want to intersect two Bézier curves $f(u)$ and $g(v)$. It results from the convex hull property (theorem 3.1) that if $f(u)$ intersects with $g(v)$, then $f(u)$ has to intersect with convex hull of control points of $g(v)$. Since the convex hull is bounded by a few line segments, we can use Bézier clipping for the boundary lines to compute the intersection with the convex hull. Once the minimal and maximal values u_1, u_2 within the convex hull are determined for the parameter u, we can use them to clip $f(u)$ to the shorter segment $f_2(u)$ for $u \in [u_1, u_2]$ (see figure 3.8). Now we reverse the role of f and g: we clip $g(v)$ against the convex hull of $f_2(u)$ to get $g_2(v)$.

By iteration, we get two sequences of curve segments: $f(u)$, $f_2(u)$, $f_3(u)\dots$ and $g(u)$, $g_2(v)$, $g_3(v)\dots$, both of them will converge to the intersection point.

Figure 3.8: Intersection of curves – clipping



Searching for the intersection point of two Bézier curves.
$f(u)$ is clipped against the convex hull of the control points of $g(v)$
to get a shorter segment $f_2(u)$ for the next step of iteration.

A little different Bézier clipping method, based on the same idea, is the *fat line* version described in [23]. That algorithm tests whether one curve lies in the fat line, which is a given range from the centerline of the other curve, clips the first curve against this area, reverses the roles of the two curves and continues iterating.

3.3.5 Intersection of surfaces

The situation becomes more complicated since the set of intersections of two surfaces can have a complex topology. For curves, the topology is very simple: the intersection is a finite set of isolated points, unless parts of the curves coincide. The following method is taken from [7].

Let $f(u, v)$ and $g(s, t)$ be two surface patches. We make the following assumptions:

- a boundary curve e of $f(u, v)$ intersects $g(s, t)$,
- the intersection of $f(u, v)$ and $g(s, t)$ is a simple curve h.

Then we can find the intersection curve by the following steps (see figure 3.9):

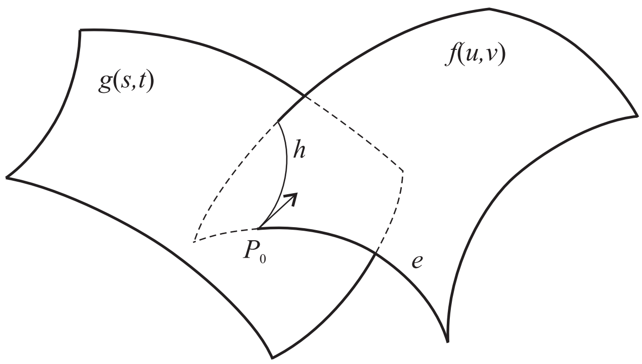
1. Intersect the boundary curve e with $g(s, t)$ to get a starting point P_0 within the intersection.
2. From P_i, step in the direction of the tangent to the intersection curve h.

3. Locate the next point P_{i+1} on h by solving the equation $f(u,v) - g(s,t) = 0$.

Comments:

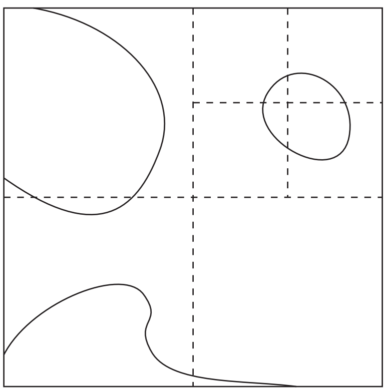
- We can use Bézier clipping to evolve a method for intersection when we try to locate P_0.
- The intersection set may not be a simple curve. For example, it can loop. This problem can be resolved by loop detection and subdivision, as it is discussed in a paper referenced by [7]. Once a loop is detected, we subdivide the surface patch, until all the loops are separated into simple curves (figure 3.10).

Figure 3.9: Intersecting surfaces



Given two surfaces f and g, the intersection curve h is detected by finding a starting point P_0 on the boundary curve e of the surface f and stepping towards the tangent of the intersection h.

Figure 3.10: Subdividing surfaces to remove intersection loops



If a loop is detected, the surface patch should be subdivided, until all the loops are separated into simple curves.

An effective surface intersection algorithm, characterizing the surfaces algebraically and using a complex tracing, is presented in [22].

Chapter 4

Conclusion

In this work, the properties of the Bernstein polynomial basis were examined, in connection with the common roots of polynomials, needed to compute the intersection of algebraic curves. The results cover the basic properties and basis transformations of polynomials, the issues concerning numerical stability, the formulation and derivation of different resultant matrices for several polynomial bases, and transformation of these matrices between bases. Moreover, algebraic and parametric curves were introduced, and a method was presented for computing the intersection of curves, both from algebraic and numerical approach. In addition, a surface intersection algorithm based on numerical approximation was briefly described.

As a conclusion, the Bernstein polynomial basis was found to be more stable on the interval of interest than other bases, therefore its usage is highly advised, along with the matrix computations involving only the Bernstein basis. Concerning the resultant matrices, the companion matrix resultant was found to be more appropriate for determining the common roots of polynomials, because of the lower order than the Sylvester and Bézout matrices have, and also because of the direct derivation of the companion matrix in the Bernstein basis. For this reason, the companion matrix resultant is considered to be more suitable for practical purposes and thus it is recommended for usage.

This study has uncovered the more advanced intersection algorithms, giving only a brief commentary on the surface intersection algorithm, and the lack of a more deeper investigation of implicitization of curves is also considered to be a weak spot.

The main contributions are the description of the relations between the power basis, the Bernstein basis and the scaled Bernstein basis, the comparison of different resultant formulations, and the unified exposition of recent progress in the area of resultants. The formulations have become more clear, and a few proofs were made more straight by introducing the requested transformation functions and the more general formulations of the properties of polynomials.

Since the assignment for this work was the study and exposition of the Bernstein polynomial basis and its properties, the main goals are considered fulfilled.

Future work should be concentrated on implicitization and surface intersection, as these were already mentioned for the less covered areas.

Bibliography

[1] Weikun Sun, Hongbo Li. On the Construction of Generic Mixed Cayley–Sylvester Resultant Matrix. Academy of Mathematics and Systems Science, Beijing, China. 2005.

[2] Joyce, D. Modern Geometry. Clark University, 2005.
http://aleph0.clarku.edu/∼djoyce/ma128/notes9.pdf

[3] Winkler, J.R. The transformation of the companion matrix resultant between the power and Bernstein polynomial bases. *Applied Numerical Mathematics*, 48, 1, pp. 113–126, 2004.

[4] Puzio, R. Derivation of Sylvester's matrix for the resultant.
PlanetMath.org – on-line encyclopedia. Article timestamp: 2004.
http://planetmath.org/encyclopedia/DerivationOfSylvestersMatrixForTheResultant.html

[5] Winkler, J.R. A companion matrix resultant for Bernstein polynomials. *Linear Algebra and Its Applications*, 362, pp. 153–175, 2003.

[6] Winkler, J.R. – Goldman, R.N. The Sylvester resultant matrix for Bernstein polynomials. *Curve and Surface Design*: Saint-Malo 2002, eds. Tom Lyche, Marie-Laurence Mazure, Larry L. Schumaker, Nashboro Predd, Brentwood, Tennessee, USA, 2003, 407–416.

[7] Zorin – Ning, F. Operations on parametric curves and surfaces: Intersection and Trimming. Article for lecture. G22.3033.002: Topics in Computer Graphics: Geometric Modeling. New York University, 2002. [1]

[8] Winkler, J.R. Properties of the companion matrix resultant for Bernstein polynomials. *Uncertainty in Geometric Computations*, eds. J. R. Winkler and M. Niranjan, Kluwer Academic Publishers, Massachusetts, USA, 2002, 185–198.

[9] Gladki, P. Resultants and the Bezout theorem. Unpublished material, written between 2002–2004. http://math.usask.ca/∼gladki/inedita/referat.pdf

[10] Sederberg, T. – Zheng, J. Algebraic Methods for Computer Aided Geometric Design. Handbook of Computer Aided Geometric Design, 2001.

[11] Winkler, J.R. Computational experiments with resultants for scaled Bernstein polynomials. *Mathematical Methods for Curves and Surfaces*: Oslo 2002, eds. T. Lyche, L. L. Schumaker, Vanderbilt University Press, Nashaville, Tennessee, USA, 2001, 535–544.

[1] The year was marked as the date for lecture #6.

[12] Zhang, H. Numerical condition of polynomials in different forms. *Electronic Transactions on Numerical Analysis*, 12, pp. 66–87, 2001.

[13] Winkler, J.R. A resultant matrix for scaled Bernstein polynomials. *Linear Algebra and Its Applications*, 319, pp. 179-191, 2000.

[14] Berchtold, J. The Bernstein basis in set-theoretic geometric modelling. PhD thesis, University of Bath, 2000.

[15] Emiris, I.Z. – Mourrain, B. Matrices in Elimination Theory. 1999.
http://www.inria.fr/saga/Bernard.Mourrain/
http://www.inria.fr/saga/Ioannis.Emiris/

[16] Chionh, E.W. – Zhang, M. – Goldman, R.N. Fast Computation of the Bezout and Dixon Resultant Matrices. In preparation, 1999a.

[17] Babušíková, J. – Slodička, M. – Weisz, J. Numerické metódy. Bratislava, 1999.

[18] Brechtold, J. – Voiculescu, I. – Bowyer, A. Multivariate Bernstein-form polynomials. Technical report No 31/98, School of Mechanical Engineering, University of Bath.

[19] Farouki, R.T. – Goodman, T.N.T. On the optimal stability of the Bernstein basis. *Mathematics of Computation*, 65(216):1553-1566, 1996.

[20] Manocha, D. – Demmel, J. Algorithms for Intersecting Parametric and Algebraic Curves I: Simple Intersections. *ACM Transactions on Graphics*. 1995.

[21] Ružický, E. – Ferko, A. Počítačová grafika a spracovanie obrazu. Bratislava: Sapienta, 1995. ISBN 80-967180-2-9

[22] Krishnan, S. – Manocha, D. An Efficient Surface Intersection Algorithm based on Lower Dimensional Formulation. *ACM Transactions on Graphics*, 1997, written in 1994.

[23] Sederberg, T.W. – Nishita, T. Curve intersection using Bézier clipping. *Computer-Aided Design*, 22, 9, 538–549, 1990.

[24] Goldman, R.N. – Sederberg, T.W. – Anderson, D.C. Vector Elimination: A technique for the implicitization, inversion, and intersection of planar parametric rational polynomial curves. *Computer Aided Geometric Design*, 1, 327–356, 1984.

[25] Sederberg, T.W. – Anderson, D.C. – Goldman, R.N. Implicit Representation of Parametric Curves and Surfaces. *Computer Vision, Graphics and Image Processing*, 28, 72–84, November, 1984.

[26] Wikipedia – free on-line encyclopedia. [2]
http://wikipedia.org

[2] Used to compare and synchronize terminology and to mine examples.

Súhrn

Táto práca zahŕňa výklad a odvodenie najmä týchto objektov a vlastností:

Bernsteinova báza

Bernsteinovu bázu polynómov stupňa n tvoria bázové funkcie $\beta_i^{[n]}(x) = (1-x)^{n-i}x^i$. Pre interval záujmu $[a, b]$ platí, že Bernsteinova báza má na tomto intervale väčšiu numerickú stabilitu ako iné bázy, ktoré sú s ňou porovnateľné. K porovnávaniu stability sa definuje čiastočné usporiadanie, podľa ktorej jedna báza predchádza druhú, ak jej korene sú menej citlivé na náhodné perturbácie v koeficientoch.

Rezultant polynómov

Rezultant množiny polynómov je výraz obsahujúci koeficienty polynómov, ktorý je nulový práve vtedy, ak polynómy majú spoločný koreň. Rezultant je často vyjadrený v tvare determinantu nejakej matice. Najznámejšie formulácie rezultantu pochádzajú z konca 19.storočia. Sú to: Sylvestrova matica, Bézoutova matica, Cayleyho formulácia Bézoutovej matice a ďalšie. Rozšírené sú hlavne tvary týchto matíc pre monomiálnu bázu, tvary pre iné bázy sa začali skúmať iba nedávno. V poslednej dobe sa intenzívne skúmajú sprievodné matice a k nim prislúchajúce rezultanty.

Sprievodná matica

Sprievodná matica *(companion matrix)* polynómu $p(x)$ je matica C_p, pre ktorú platí $(C_p - \lambda I) = p(x)$. Ak $r(x)$ a $s(x)$ sú polynómy, $r(x)$ s vedúcim koeficientom 1, a C_r je sprievodná matica polynómu $r(x)$, potom determinant matice $s(C_r)$ je rezultant polynómov r a s. Rezultant prislúchajúci k sprievodnej matici *(companion matrix resultant)* je nižšieho stupňa ako Sylvestrova matica a môže sa konstruovať priamo v Bernsteinovej báze, preto pre praktické účely sa považuje za vhodnejšiu ako iné popísané matice.

Z upravenej matice rezultantu je možné vypočítať najväčší spoločný deliteľ polynómov až na konštantný násobok. Najväčší spoločný deliteľ priamo súvisí so spoločnými koreňmi týchto polynómov.

Parametrické krivky

Každá polynomiálna parametrická krivka v rovine môže byť vyjadrená v tvare $P(t) = \sum_{i=0}^{n} P_i \beta_i^{[n]}(t)$. Krivka tohoto tvaru je Bézierova krivka s riadiacimi vrcholmi P_i. Najdôležitejšou vlastnosťou Bézierovej krivky je, že sa celá nachádza v konvexnom obale svojich riadiacich vrcholov. Táto vlastnosť môže byť využitá pri hľadaní prieniku dvoch kriviek, pretože všetky body prieniku sú obsiahnuté v prieniku prislúchajúcich konvexných obalov.

Algebraické krivky

Algebraická krivka v rovine je vyjadrená polynomiálnou funkciou $f(x, y)$. Krivku tvoria body $[x, y]$, pre ktorých $f(x, y) = 0$. Parametrické polynomiálne krivky sa dajú previesť do tohto implicitného tvaru. Z polynómov dvoch algebraických kriviek je možné odvodiť maticu, ktorej determinant je ich rezultantom. Krivky majú spoločný bod práve vtedy, ak tento determinant je nulový.

Bézier clipping

Bézier clipping je metóda aproximačného riešenia polynomiálnej rovnice $f(u) = 0$. Polynóm $f(u)$ sa vyjadrí v Bernsteinovej báze a rozšíri sa o ďalšiu dimenziu s lineárne rastúcou súradnicou. Takto sa získa krivka v rovine s bodmi $[u, f(u)]$, $u \in [0, 1]$, pričom korene rovnice zodpovedajú u-súradniciam prieniku krivky s osou u. Zistia sa hraničné hodnoty u_1, u_2 prieniku osi u a konvexného obalu prislúchajúceho krivke. Takto sa interval $[0, 1]$ rozdelí na tri časti a prienik krivky s osou u sa ďalej hľadá na intervale $[u_1, u_2]$.

Problém prieniku Bézierovej krivky s priamkou môže byť redukovaný na polynomiálnu rovnicu a riešený metódou Bézier clipping. Metóda tým pádom môže byť využitá pri hľadaní prieniku dvoch kriviek pomocou postupného orezávania kriviek príslušnými konvexnými obalmi. Metóda môže byť využitá dokonca aj pri riešení prieniku plôch.