8-19-2015 12:00 AM

Determining Critical Points of Handwritten Mathematical Symbols Represented as Parametric Curves

Aoesha G. Alsobhe, *The University of Western Ontario*

Supervisor: Dr. Stephen Watt, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Determining Critical Points of Handwritten Mathematical Symbols Represented as Parametric Curves

(Thesis format: Monograph)

by

Aoesha Alsobhe

Graduate Program in Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

The School of Graduate and Postdoctoral Studies
The University of Western Ontario
London, Ontario, Canada

Abstract

We consider the problem of computing critical points of plane curves represented in a finite orthogonal polynomial basis. This is motivated by an approach to the recognition of hand-written mathematical symbols in which the initial data is in such an orthogonal basis and it is desired to avoid ill-conditioned basis conversions. Our main contribution is to assemble the relevant mathematical tools to perform all the necessary operations in the orthogonal polynomial basis. These include implicitization, differentiation, root finding and resultant computation.

Keywords: Handwriting recognition, parametric curve, implicit curve, orthogonal bases, critical points, Chebyshev basis, Legendre basis, resultants, resultants in orthogonal bases, Sylvester matrix, companion matrix.

Acknowledgments

In The Name Of ALLAH, The Most Beneficent, The Most Merciful

First and foremost, praise and gratitude be to ALLAH, the Almighty, who has blessed me and given me the strength, health and patience to complete my study.

I would like to express my deep gratitude to my supervisor, Dr. Stephen Watt for his patient direction, enthusiastic encouragement, and useful suggestions, without which this thesis would not be complete. I owe much gratitude and sincere thanks to him for giving me this unique opportunity to be his student. His academic, generous and moral support during these past two years has been unceasing. I will forever be grateful to him. Thank you Dr. Stephen for your generous spirit. I am humbled and grateful that you give me the chance to be your student.

I would like to acknowledge Saudi Arabia and the Saudi bureau for granting and providing the scholarship for my Master program at Western University.

I especially thank my father and my lovely mother for their care, their prayers, and their concern for my progress throughout my period of living abroad. I will forever be thankful to my brothers and sisters for their moral support and strong encouragement to complete this program.

Lastly, but by no means the least, I would also like to extend a special thanks to my lovely husband for his love, support, patience, and advice, and my sweet girls, Jana and Jwan, for their encouraging smiles and warm hugs every day.

Glossary of Terms

Condition number: The measure of how much a function's results change (at a given point) relative to a change in the argument. Functions with a small condition number are called "well-conditioned". Functions with a large condition number are called "ill conditioned". When calculated with finite precision, functions with a high condition number cannot be computed to full precision.

Implicitization: Implicitization is the process of conversion from the parametric form of a curve to its implicit form.

Resultant: A resultant matrix of two polynomials is a matrix whose entries are functions of their coefficients, such that a necessary and sufficient condition for the polynomials to have a common root is that the determinant of this matrix, called the *resultant* of the polynomials, is exactly zero.

Inner Product: A function with certain properties that takes two elements of a vector space and gives a scalar. In the context of this thesis, we use an integral inner product on the space of real functions. The integral inner product with weight w on the interval $[a, b]$ of two functions, $f(x)$, and $g(x)$ is

$$\langle f, g \rangle = \int_a^b f(x)g(x)w(x)d\,x$$

Orthogonal Functions: Two functions f and g are orthogonal if $\langle f, g \rangle = 0$. A set of functions $\{f_i\}$ is orthogonal if $\langle f_i, f_j \rangle = 0$ when $i \neq j$.

Chebyshev polynomials: Chebychev polynomials are orthogonal on $[-1, 1]$ for weight $w(t) = 1/\sqrt{1-t^2}$: $\langle T_i, T_j \rangle = \int_{-1}^{1} T_i(t) T_j(t) \left(\frac{1}{\sqrt{1-t^2}}\right) dt = 0$ when $i \neq j$.

Legendre polynomials: Legendre Polynomials are orthogonal on $[-1, 1]$ with respect to the weight function $w(t) = 1$: $\langle P_i, P_j \rangle = \int_{-1}^{1} P_i(\mathrm{t})\, P_j(t) dt = 0$ $if\ i \neq j$

Abbreviations and Notation

$\frac{df}{dx}$ $\frac{\partial f}{\partial x}$, f_x	The partial first derivative of function f with respect to X coordinate
$\frac{d^2f}{dx^2}$, $\frac{\partial^2 f}{\partial x^2}$, f_{xx}	The partial second derivative of function f with respect to X coordinate
$Res(f,g)$	Resultant Matrix for Polynomials $f(x)$ and $g(x)$
$Syl(f,g)$	The Sylvester resultant matrix for the Monomial polynomials f and g
$B(f,g)$	The Bezout resultant matrix for the Monomial polynomials f and g
$Det(Syl(f,g))$	The determinant of the Sylvester matrix for polynomials f and g
$CSyl(f,g)$	The Sylvester resultant matrix for the Chebyshev polynomials f and g
$LSyl(f,g)$	The Sylvester resultant matrix for the Legendre polynomials f and g
$T_n(x)$	Chebyshev Polynomial of degree n
$P_n(x)$	Legendre Polynomial of degree n
$\langle f,g \rangle$	The Inner product of two functions f and g
GCD	Greatest Common Divisor
$C(f)$	Companion matrix of the polynomial $f(x)$
$eig\ (A)$	Eigenvalues of matrix (A)

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Handwriting recognition has been studied for several decades. The field has flourished due to its widespread application in economic activities such as cheque processing and mail sorting. In recent years, with the development of Personal Digital Assistants (PDAs) and smart phones with styluses, mathematical handwriting recognition has received increasing attention and has become a problem of particular interest.

Recognition of handwritten mathematical expressions is more difficult than recognition of the handwritten characters of Western languages. There are more mathematics symbols than Western characters: mathematical expressions include different alphabets, digits, operators, and special characters. There are many similar characters which are written with just a few strokes and must be distinguished. Figure 1.1 shows symbols with similar features which are written with one stroke. In mathematical handwriting, there is no fixed dictionary of words to distinguish between similar symbols. Moreover, the two-dimensional nature of the input as well as placement information is important. Symbols are typically of several sizes, leading to ambiguous juxtapositions as shown in Figure 1.2. For all of these reasons, recognizing mathematical characters accurately is more challenging. New methods are required for the accurate recognition of handwritten mathematical symbols, owing to their complicated characteristics [1, 3].

Figure 1.1: Similar single-stroke mathematical characters.
Figure from [1]. Used with permission.

Figure 1.2: Juxtaposition ambiguity. Are the first two symbols $a^p or\ aP$?
Figure from [1]. Used with permission.

There is a geometry theory that addresses these issues and is thus useful for recognizing mathematical symbols [4, 5, 6, 7]. According to this theory, handwritten mathematical symbols should be treated as parametric curves rather than sets of points. This approach, used in both earlier and later work [1, 4, 5, 6, 7], proposes to represent characters as parametric curves approximated by truncated orthogonal series such as Chebyshev polynomials, Legendre polynomials, or Legendre -Sobolev polynomials [4, 5 6]. Rather than considering character traces as collections of discrete points, symbols are mapped to low-degree polynomials. This approach has been found to yield a high recognition rate [1, 4, 5, 6]. By choosing an appropriate family of basis polynomials of high enough degree, the approximating curve can be made arbitrarily close to the original trace [1, 4, 5]. The trace is represented using the coefficients x_i and y_i from:

$$X(t) \approx \sum_{i=0}^{d} x_i B_i(t), \qquad Y(t) \approx \sum_{i=0}^{d} y_i B_i(t)$$

This representation is compact, device independent, and allows algebraic treatment of the curves .The beauty of this representation is that finding the critical points (those that are used as features for recognition, such as self-intersection points, local maxima and minima, cusps, and loops) from polynomial approximations is robust against changes in device resolution [1, 2]. Finding the critical points from the polynomial representation can be done by solving the differential equation system $X'(\lambda) = 0, Y'(\lambda) = 0$ by finding univariate polynomial roots [1]. Roots of univariate polynomials are the eigenvalues of the companion matrix.

Implicit representation is best suited for some operations on curves such as computing critical points of curves [21, 22]. This motivates converting from parametric representations to implicit representations. Motivation is provided by singular points problems on curves. If curves can be expressed implicitly as $f(x, y) = 0$, then singular points can be obtained by solving for the roots of the partial differential system of bivariate polynomial equations given by $f = f_x = f_y = 0$ where f_x *and* f_y are the x and y partial derivatives of f, respectively. The common solutions of the differential system can be obtained by finding the roots of the resultant matrices of f_x and f_y [25]:

$$Res_x(f_x, f_y) = 0 \; and \; Res_y(f_x, f_y) = 0 .$$

Implicitization of a parametric curve can be obtained directly based on the resultant method [15, 21-23] as follows:

$$Res\ (X - x(\lambda), Y - y(\lambda))$$

On the other hand, the problem with this representation is that the transformation between orthogonal bases and the monomial basis is ill conditioned (change the representation of a polynomial from one basis to another can amplify numerical errors. Error bounds can grow exponentially with the degree of the polynomial, and the relative errors can be infinitely larger in one basis than in another) [1, 34, 40, 41]. This requires the development of derivatives and resultant matrices for orthogonal bases. We need to compute the first derivative of truncated orthogonal series and resultants in the right basis without transforming to the monomial basis. Changing the representation of a truncated series from one basis to another can amplify numerical errors. We want to avoid this error. In this thesis, our goal is to determine whether the mathematical tools exist to perform all the necessary operations in the orthogonal polynomial basis without converting to monomial bases. We are interested in computing the derivative of orthogonal series in orthogonal bases rather than doing ill-conditioned conversions. We are also interested in computing roots of derivative orthogonal series based on resultant matrices computed in orthogonal bases in order to find the critical points of parametric and algebraic curves in orthogonal bases. Figure 1.3 shows the critical points of a handwritten character represented as a parametric curve in an orthogonal basis.

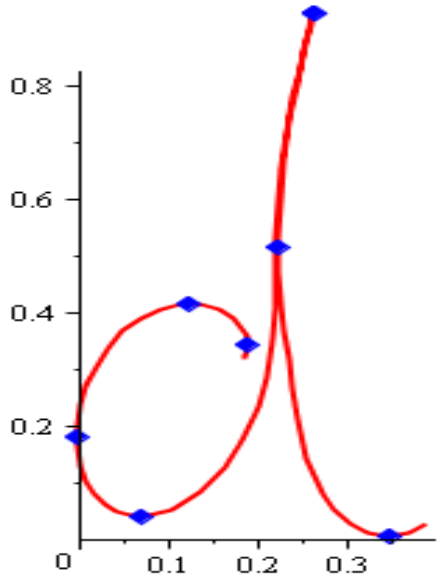


Figure 1.3: Critical points computed from parametric approximation.
Figure from [1]. Used with permission.

1.1Motivation and Related Work

The original motivation behind this thesis comes from a problem arising in handwriting recognition: handwriting represented as plane curves in an orthogonal function basis. Orthogonal polynomials arise in many applications: linear control theory, interpolation problems, rational interpolation problems, least-squares approximation, and approximation theory. This motivates us to compute resultant matrices, derivatives, and roots of polynomials in orthogonal bases rather than transforming to the monomial basis.

Most approaches to the manipulation of polynomials assume that the polynomials are expressed in the monomial basis, for simplicity. When polynomials are represented in alternative bases such as orthogonal bases, we can convert them into polynomials in the monomial basis, do some polynomial manipulations such as resultants, GCD, derivative, etc., and then convert them back to the right orthogonal basis. In this thesis we avoid such

conversions and describe the arithmetic computations in the original bases [41]. There are reasons to avoid converting to the monomial basis: the conversion can convert a polynomial in a simple computational domain into one in a complex mathematic domain. Also, coefficients can grow large, the computation cost may increase, and an orthogonal polynomial basis is a better choice for numerical stability [41].

There has been considerable work done on the manipulation of polynomials represented in alternative bases. The Bezout matrix for Chebyshev polynomials is derived in [39]. Nakatsukasa, Noferini and Townsend develop practical strategies for bivariate root-finding including the Chebyshev Bezout matrix in [27]. Boyd computes roots for polynomials in Chebyshev and Legendre bases [53, 58, 64, 67]. The resultant of Chebyshev polynomials is investigated in [31]. The Sylvester resultant matrix in the Chebyshev basis is computed in [38].

1.2 Contributions

This thesis demonstrates an approach of computing critical points, maxima, and minima of an orthogonal truncated series based on resultants in orthogonal basis. In particular, we compute these points in Legendre basis based on resultant matrices built in Legendre basis. First, we compute the derivative of truncated Legendre series, and then we compute the roots of derivative series based on resultants. We develop an algorithm for computing first derivative in Legendre basis. We also compute resultant Sylvester matrix in Legendre basis. Usual methods for finding the critical points of a truncated orthogonal series *p(x)* is to express *p (x)* as a sum of monomials, and then to calculate the derivative and the roots of the monomial series. On the other hand, change the representation of $p(x)$ from one basis to another can amplify numerical errors. Error bounds can grow exponentially with the degree of the polynomial, and the relative errors can be infinitely larger in one basis than in another. Our

goal of this thesis is to avoid this conversion, and do computations in the right basis.

1.3 Organization of This Thesis

This thesis is organized as follows. Chapter 2 introduces orthogonal polynomial approximation in handwriting recognition. It describes some of the approaches in [1-10] of representing handwritten characters as parametric curves approximated by truncated orthogonal series such as Chebyshev polynomials, Legendre polynomials, and Legendre-Sobolev polynomials. Moreover, this chapter presents some of the benefits of this representation.

Chapter 3 describes the critical points of curves in general, and how to find maxima, minima, cusps, self-intersection points, etc., on curves, which help to identify algebraic features of curves. An algorithm to find the critical points and maxima and minima is introduced. The chapter also introduces computing the partial first and second derivatives with respect to x and y coordinates at the critical points, and provides some possible types of singular points of curves.

In chapter 4, we introduce our approach of computing resultant Sylvester matrix in orthogonal bases. First we give a brief overview of the most common formulas for resultant matrices in the monomial basis. Then, we introduce computing resultants in orthogonal bases based on a basis conversion between orthogonal polynomials and monomial polynomials. Moreover, this chapter gives the basic ideas and theories of orthogonal polynomials and their properties. We also introduce our method to find the resultant matrix (Sylvester matrix) of an orthogonal basis (in particular, the Legendre basis).

Chapter 5 presents some important applications of resultants in geometric fields. It introduces

a method for root-finding of bivariate polynomial systems based on resultants. In this chapter, we analysis our approach of finding singular points of parametric and implicit curves based on resultants in the monomial basis. Moreover, we show how to use resultants to convert from parametric representations of curves to implicit representations.

In chapter 6, the critical points of an orthogonal truncated series are computed. First, we develop an algorithm for computing the first derivative of a truncated Legendre series. Second, the roots of the derivative series are computed based on the Legendre-Companion matrix. We also do some numerical examples to demonstrate our approach for computing critical points in orthogonal bases.

Chapter 7 concludes the thesis and outlines future work.

Chapter 2

Orthogonal Polynomial Representation in Handwriting Recognition

Traditional handwriting recognition methods are typically based on representing characters by sequences of discrete points, each of which contain x and y values in a rectangular coordinate system. Figure 2.3 shows the representation of a handwritten character using a sequence of sample points. This method of representing characters leads to a slow recognition rate, however, for character sets containing many similar characters (like mathematical symbols) since comparison against all possible symbol models is then required. Furthermore, determining features such as the number of cusps, number of loops, and self-intersections from a set of points requires many ad hoc treatments such as smoothing, resampling, and various numerical heuristics. It is impractical to develop hand-tuned heuristics to recognize specific features for each symbol. To avoid all of these difficulties, handwritten characters should be treated as curves [1, 2, 3].

Handwritten symbols can be represented in the space of coefficients of a functional approximation. This approach has been used in earlier work [4, 5, 6]. Mathematical handwritten characters can be represented as parametric curves approximated by truncated orthogonal polynomial series. There are many possible parameterizations of a curve, including parameterization by time (as the curve was traced), and parameterization by arc length [1]. It has been shown in [10] that arc length is the most robust parameterization in

most cases since it provides independence from variations in speed of writing, leading to curves that look the same regardless of their original parameterization.

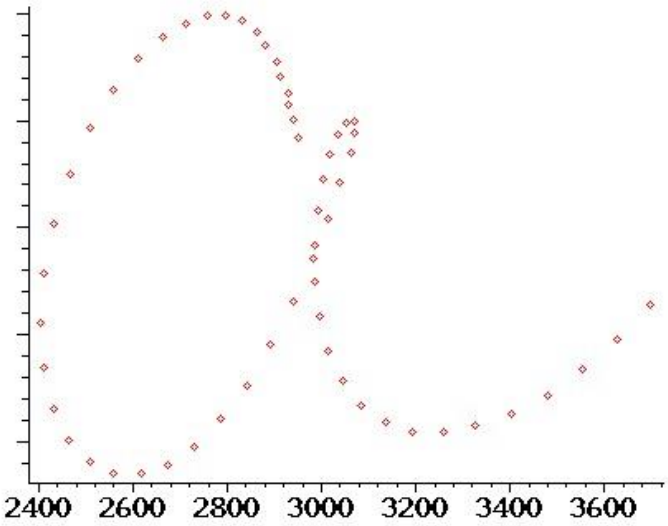


Figure 2.1: Sequence of sample points for character "a".
Figure from [10]. Used with permission.

This approach has many useful properties. It allows high recognition rates for small data sizes. It maps symbols from two dimensions to a low dimensional vector space of orthogonal series coefficients. The Euclidean distance in this space is related to the integral between two curves and can be used to find similar symbols very efficiently [1, 2, 3]. By choosing the functional basis appropriately, the series coefficients can be computed in real-time, as the symbol is being written [5, 6]. Figure 2.4 shows a parametric curve approximation for the symbol 'G'.

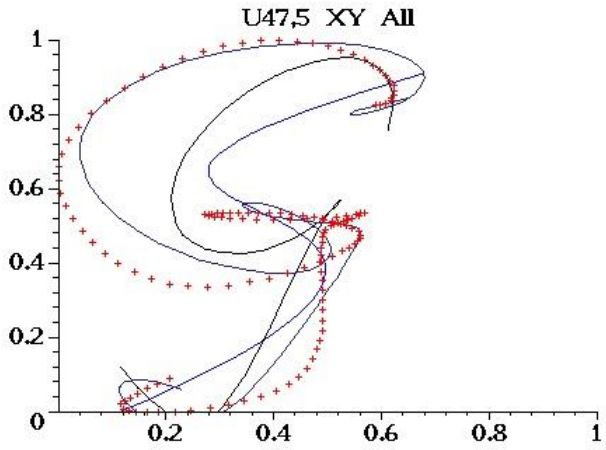


Figure 2.2: Approximated representation for symbol 'G'.
Figure from [1]. Used with permission.

2.1 Series of Orthogonal Functions

Truncated orthogonal series are an important subject in geometric symbolic-numeric polynomial algorithms. A set of functions $\{f_i\}$ is orthogonal with respect to their inner product with weight $w(t)$ on domain [a, b] if

$$\langle f_i, f_j \rangle = \int_a^b f_i(t) f_j(t)\, w(t) dt = 0 \;\; if\; i \neq j. \qquad 2.1$$

A well-known technique of approximation of a function $f: R \rightarrow R$ is finding a linear combination of functions from the truncated basis $P = \{P_i : R \rightarrow R, i = 0, 1, \dots, d\}$:

$$f(t) \approx \sum_{i=0}^{d} c_i P_i(t), \qquad c_i \in R,\; P_i \in P \qquad 2.2$$

In orthogonal bases, the coefficients can be easily computed by the inner product of the orthogonal functions [6].

$$\langle f, h_i \rangle = \langle \sum_{j=0}^{\infty} \alpha_j h_j(t), h_i(t) \rangle$$

$$= \sum_{j=0}^{\infty} \alpha_j \langle h_j, h_i \rangle \; = \; \alpha_i \langle h_i, h_i \rangle$$

$$\alpha_i = \langle f, h_i \rangle / \langle h_i, h_i \rangle \qquad 2.3$$

2.2 Bases for Approximation

Handwritten symbols may be represented as the coefficients of an approximating basis in a finite function space [1].

$$X(t) \approx \sum_{i=0}^{d} x_i B_i(t), \qquad Y(t) \approx \sum_{i=0}^{d} y_i B_i(t) \qquad 2.4$$

By choosing appropriate basis functions $B_i \;, i = \; 0, \ldots\ldots, d,$ the approximating curve can be close to the original trace [2]. If the basis functions are orthogonal with respect to the function inner product, it is easy to compute the coefficients $(x_i \; y_i)$ by numeric integration [6, 7].

It is more accurate to choose the function basis to be orthogonal polynomials, e.g. Chebyshev [4], Legendre [6] or Legendre-Sobolev [5]. Char and Watt propose to represent a character as a vector of coefficients of the approximation of the curve coordinates with truncated orthogonal series in [4]. The approximating orthogonal curve will be close to the curve of a character if we choose an appropriate orthogonal basis of polynomials [6, 7]. In the next sections of this chapter, we discuss in detail some approaches to handwritten character representation using orthogonal polynomial approximation.

2.2.1 Chebyshev Representation

In early work [4], Char and Watt show that the coordinate function (X, Y) of handwritten characters can be represented as truncated series of Chebyshev polynomials of the first kind, $T_n(t) = cos\ (n\ arcos\ t)$. Chebychev polynomials are orthogonal on $[-1, 1]$ for weight $w(t) = 1/\sqrt{1-t^2}$.

$$\langle T_i, T_j \rangle = \int_{-1}^{1} T_i(t) T_j(t) \left(\frac{1}{\sqrt{1-t^2}}\right) dt = 0 \qquad \text{i} \neq j \tag{2.5}$$

For series truncated at order *d*, the approximation is given

$$X(t) \approx \sum_{i=0}^{d} \alpha_i T_i(t), \qquad Y(t) \approx \sum_{i=0}^{d} \beta_i T_i(t) \tag{2.6}$$

The Chebyshev series for X and Y are developed in [4] along with a technique for computing the coefficients of a Chebyshev series for a function. An example of using Chebyshev polynomial approximation in handwriting recognition is shown in Figure 2.5.

Although Chebyshev polynomials are easy to compute and allow the accurate approximation of a curve by low-degree series, the form of the corresponding weight function is not appropriate for the online computation of the approximation [6]. The weight function of Chebyshev polynomials is non-linear $\left(1/\sqrt{1-t^2}\right)$, requiring the capture of the entire curve before the series coefficients can be computed [6]. Clearly, the problem with Chebyshev approximation is that we cannot compute the Chebyshev series directly in an online manner; this is related to the algebraic form of the weight function $w(t) = 1/\sqrt{1-t^2}$. Instead of using Chebyshev polynomials, a family of orthogonal polynomials with the simplest weight function can be considered [6].

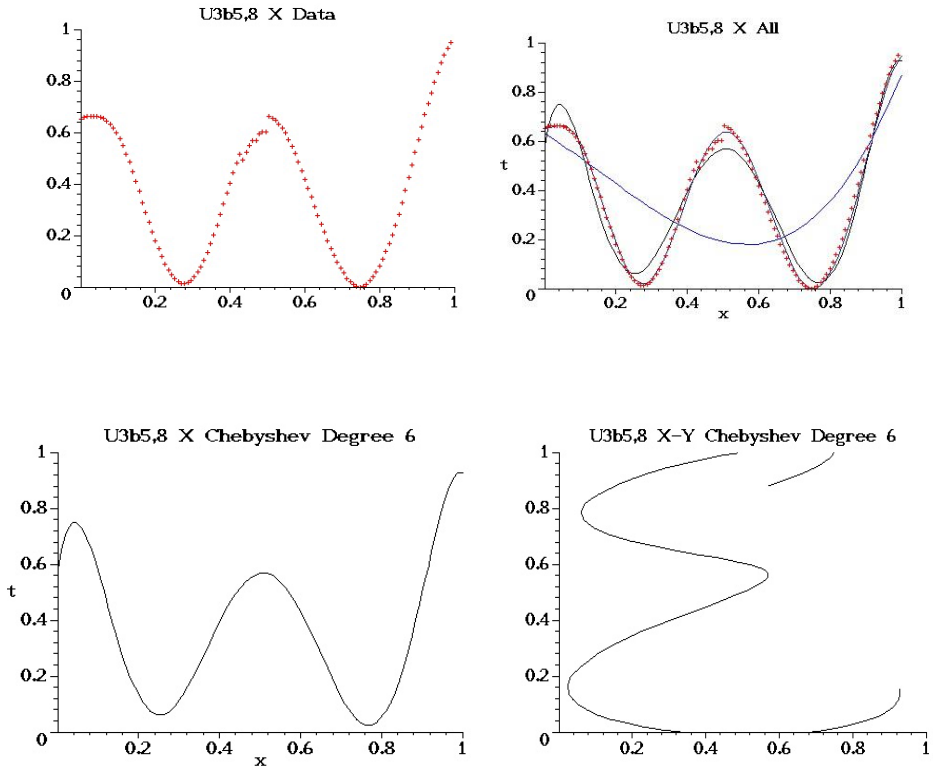


Figure 2.3: The x, y traces of ϵ: Data Degree 3, 6, 10 Chebyshev fits.
Figure from [4]. Used with permission.

2.2.2 Legendre Representation

Most handwriting recognition systems require characters to be completed before any recognition takes place. Another approach is for real-time recognition [6]; that is, a curve can be computed as the curve is being written and can be classified when the pen is lifted by using a different functional basis, allowing useful computation as the curve data is received. Golubitsky and Watt in [6] propose to use Legendre polynomials. The Legendre polynomials can be defined as

$$P_n(t) = \frac{1}{2^n n!} \frac{d^n}{dt^n} (t^2 - 1)^n. \qquad 2.7$$

These are orthogonal on $[-1, 1]$ with respect to the weight function $w(t) = 1$:

$$\langle P_i, P_j \rangle = \int_{-1}^{1} P_i(\mathrm{t})\, P_j(t)dt = \frac{2}{2n+1}\delta_{ij}, \qquad 2.8$$

$$\delta_{ij} = \begin{cases} 1 & i = j \\ o & otherwise \end{cases}$$

By using a Legendre polynomial basis, the coefficients can be computed from the moments of $x\ (\lambda)$ and $y\ (\lambda)$ integrated as the curve is written [6]. The main idea is to calculate moments of the coordinate curves in real time as the character is written and then compute the coefficients of the Legendre series representation when the pen is lifted [6]. Figure 2.6 shows the approximations of the character '3' with a Legendre series.

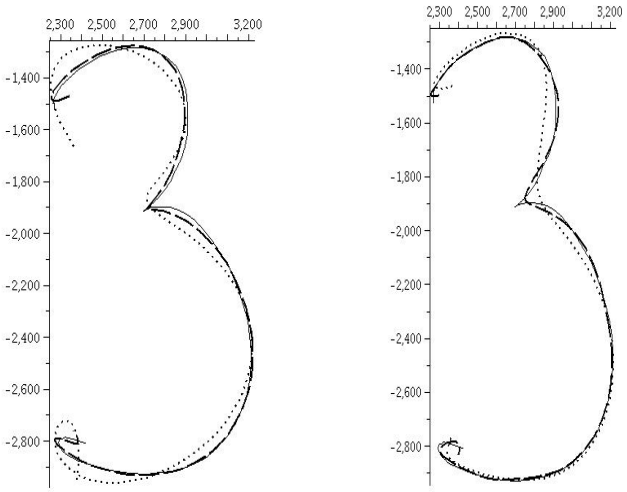


Figure 2.4: Approximations of character '3' with Legendre series. Left: time parameterization. Right: arc length parameterization.
Figure from [8]. Used with permission.

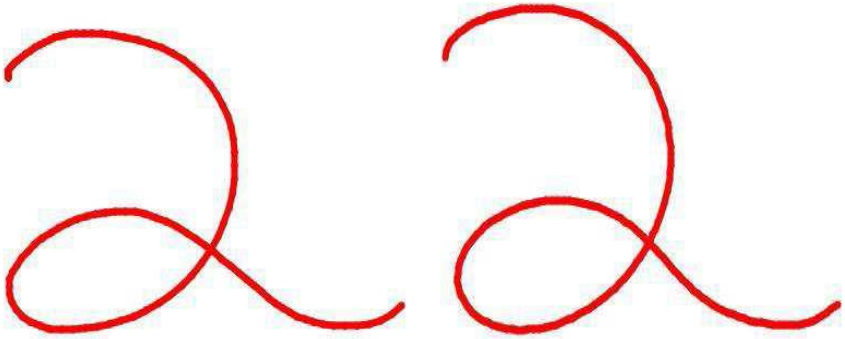
It has been found [6, 8] that the Legendre series representation is the same as the Chebyshev representation; however, the Legendre series representation has the advantage of only requiring a small, fixed number of arithmetic operations at the end of a stroke, making it more appropriate for online computation.

2.2.3 Legendre-Sobolev Representation

In later work [5], it is shown that the Legendre-Sobolev basis allows curves to be approximated close to the original curve and with low degree.. It is shown that the real-time property is preserved using the Legendre-Sobolev basis [5, 7, 8], which is orthogonal with respect to the inner product:

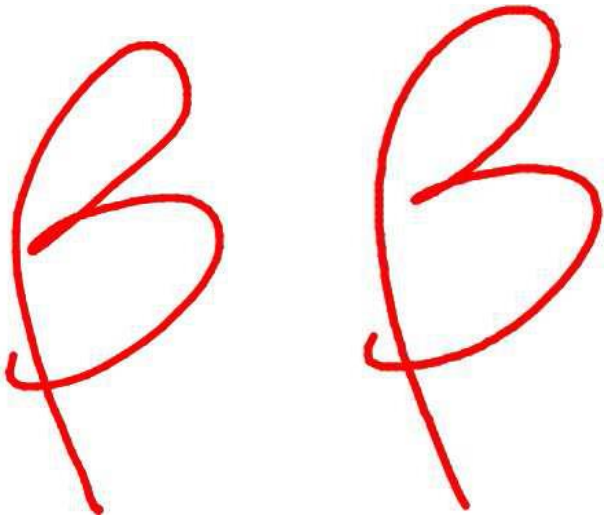
$$\langle f, g \rangle = \int_a^b f(\lambda) g(\lambda)\, d\lambda + \mu \int_a^b f`(\lambda) g`(\lambda)\, d\lambda. \qquad 2.9$$

Legendre-Sobolev polynomials can be computed by Gram-Schmidt orthogonalization of the monomial basis $\{1, \lambda, \lambda^2, \ldots \ldots\}$ [1, 5, 7]. Figure 2.7 and Figure 2.8 show approximations of the characters '2' and 'B' with the Legendre-Sobolev series of degrees $0\ to\ 12$ for $a = 0, b = 0, \mu = 1/8$.



(a) (b)

Figure 2.5: Approximation using Legendre-Sobolev series.
(a) Original. (b) Approximated using series of degree 12 with μ = 1/8.
Figure from [2]. Used with permission.



(a) (b)

Figure 2.6: Another example of approximation using Legendre-Sobolev series. (a) Original. (b) Approximated using series of degree 12 with $\mu = 1/8$. Figure from [11]. Used with permission.

To summarize, Chebyshev, Legendre, and Legendre-Sobolev series approximations give the same convergence rates. However, Legendre and Legendre-Sobolev are the most convenient approaches to computing the coefficients online as the curve is written, unlike the other inner products with non-linear weight functions such as Chebyshev series [8]. Figure 2.9 shows Legendre-Sobolev on [0, 1].

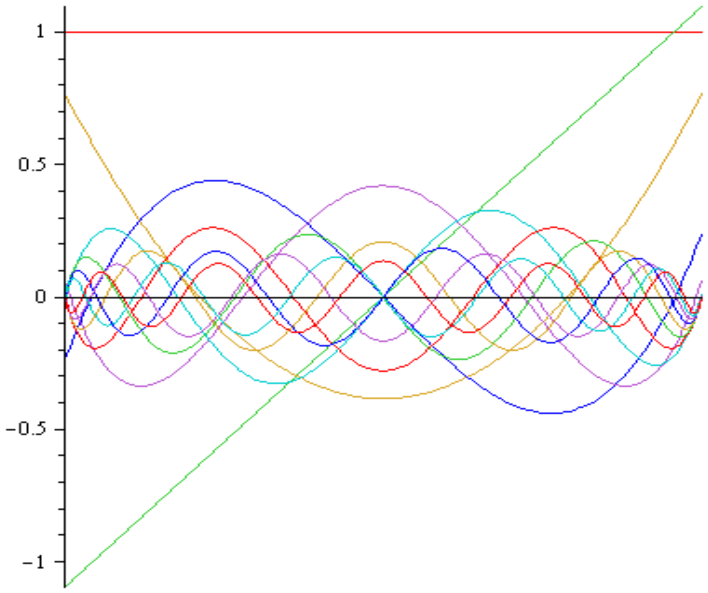


Figure 2.7: L-S polynomials on [0, 1] for μ = 1/8.
Figure from [1]. Used with permission.

2.3 Computing Distances Between Curves

Representing handwritten mathematical symbols as parametric curves in orthogonal spaces has many benefits [1]. One of these benefits is that distances in orthogonal functional bases are Euclidean distances in the coefficient space [1, 8]:

$$||f - g|| \approx \sqrt{\sum_{i=0}^{d}(f_i - g_i)^2} \qquad 2.10$$

The inner product of the function bases $B_i B_j \quad i \neq j$ is zero by orthogonality [1, 8]. This allows the computation of the integrals approximated by elastic matching very quickly. Polynomial parameterization allows computing distances between curves. The types of distance depend on the choice of the orthogonal polynomial basis. The Chebyshev basis maximal distance between points on curves is [10]:

$$\max_{t\epsilon[-1,1]}\left((x_1(t) - x_2(t)\right)^2 + \left((y_1(t) - y_2(t)\right)^2 \qquad 2.11$$

The Legendre basis to $L2$ distance is given by:

$$\int_{-1}^{1}\big(x_1(t)-x_2(t)\big)^2+\big(y_1(t)-y_2(t)\big)^2dt \qquad 2.12$$

The Legendre-Sobolev to the distance in the Sobolev space is:

$$\int_{-1}^{1}\big(x_1(t)-x_2(t)\big)^2+\big(y_1(t)-y_2(t)\big)^2dt+\int_{-1}^{1}\big(x`_1(t)-x`_2(t)\big)^2+\big(y`_1(t)-y`_2(t)\big)^2dt$$

2.4 Feature Extraction

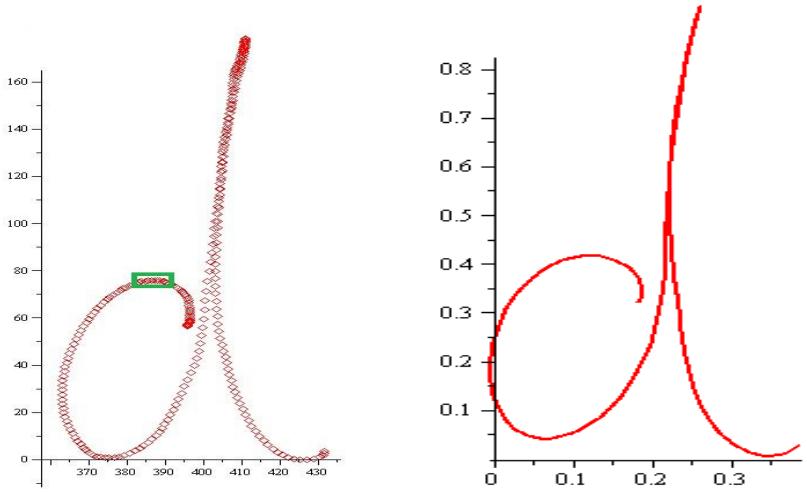
Another benefit of using orthogonal polynomial approximation in handwriting recognition is the convenience in determining the critical points such as self-intersections, number of local maxima, number of local minima, loops, cusps and all other points which are used in features detection for recognition [1]. The properties of curves can be computed algebraically by finding these critical points. The usual methods of feature extraction depend on device resolution (which means new algorithms are very important with rapidly changing technology), whereas determining the critical points from the polynomial representation is robust against changes in device resolution [1]. Figure 2.8 (a) shows the trace data of letter d.

Finding the critical points from the polynomial approximation in parametric form, $(x(\lambda), y(\lambda))$, can be achieved by solving the derivative equations for parametric curves: $x'(\lambda)=0,\ \text{and}\ \ y'(\lambda)=0$. This is obtained by univariate polynomial root-finding. Figure 2.8 (b) shows an approximation in parametric form, $(x(\lambda),\ y(\lambda))$, with *x*, and *y* Legendre-Sobolev series. Figure 2.8 (c) shows the critical points found by solving $x'(\lambda)=0\ \ \text{and}\ \ y'(\lambda)=0$.

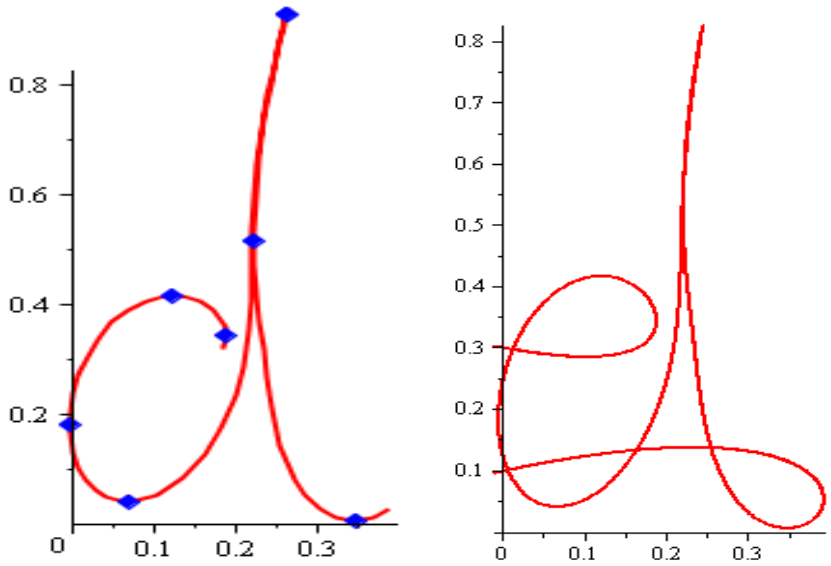
The implicit representation, $(X, Y) = 0$, is most convenient for some operations. This can be obtained directly from the parametric representation as $Resultant(X - x(\lambda),\ Y - y(\lambda), \lambda)$. Figure 2.8 (d) shows the implicit polynomial obtained by resultants. It is easy to obtain critical points of an implicit curve $P(X, Y) = 0$ by solving for the roots of the partial differential system of polynomial equations, $P = P_x = P_y = 0$ where P_x and P_y are the x and y partial derivatives of P, respectively. One way of obtaining the common solutions is to find those roots of the resultant Sylvester matrix of P_x and P_y :

$$Res_x(P_x, P_y) = 0 \ and \ Res_y(P_x, P_y) = 0 \qquad 2.14$$

Transformation between orthogonal bases and the monomial basis can leads to numeric error [1, 34, 40, 41]. This requires computing resultants in orthogonal bases without transforming to the monomial basis. In this thesis, we are interested in finding the resultant matrices in some orthogonal bases such as the Chebyshev basis and the Legendre basis.



(a) (b)



(c) (d)

Figure 2.8: (a) Trace data of letter d, (b) Parametric approximation $(x(\lambda), y(\lambda))$. (c) Critical points computed from Parametric approximation $(x(\lambda), y(\lambda))$. (d) Implicit approximation obtained by $Resultant(X - x(\lambda), Y - y(\lambda), \lambda)$. Figure from [1]. Used with permission.

2.5 Summary

This chapter discusses the geometric theory of handwriting recognition, which is stated as treating handwritten characters as parametric curves approximated in orthogonal bases such as the Chebyshev basis, the Legendre basis, or the Legendre-Sobolev basis. Moreover, the chapter introduces the benefits of this representation. The most important benefit of using orthogonal polynomial approximation in handwriting recognition is that it is easy to determine algebraic features of curves such as loops, cusps, maxima, and so on.

The next chapter discusses critical points of algebraic curves. It presents the use of the first-derivative test to determine the critical points of the curves, and the use of the second-derivative test to determine minima and maxima of the curves. There are three types of

critical points: minima, maxima, and inflection points. We also introduce the algebraic properties of curves such as loops, cusps, self-intersections and so on. Computing the singular points of curves is discussed in subsequent chapters.

Chapter Three

Critical Points on Curves

The problem of finding the critical points of parametric and algebraic curves arises in many applications of computer graphics, computer vision, image processing, pattern recognition, and artificial intelligence [18]. We are concerned with the problem of finding critical points within the realm of handwriting recognition. An essential stage of handwriting recognition is finding the critical points (such as maxima, minima, cusps, loops, and points of intersection) of parametric curves representing handwritten characters, in order to detect their basic features. There are three types of critical points: maxima, minima, and inflection points. In this chapter, we give an overview on how to find the critical points on curves and how to classify them as maxima, minima, cusps, loop, and self-intersections.

3.1 Finding Critical Points of a Function

In mathematics, a critical point of a single variable function is a value in its domain where its derivative is zero or the derivative doesn't exist. For a function of several variables, the critical points are the values in its domain where all partial derivatives are zero or don’t exist [13, 14]. Example 3.1 illustrates how to find the critical points of a single-variable function and Example 3.2 illustrates how to find the critical points of a function of two variables.

Definition 3.1 A critical point on $f(x)$ occurs at x_0 if and only if either $f'(x_0)$ is zero or the derivative doesn't exist.

Definition 3.2 Critical points on $f(x, y)$ occur at (x, y) if and only if either the partial derivatives $\frac{df}{dx}, \frac{df}{dy} = 0$ or $\frac{df}{dx}, \frac{df}{dy}$ do not exist.

Example 3.1 Consider the function of $(x) = \sqrt[3]{x^3 - 3x}$. The critical points of this function can be found and classified in three steps [13]. First, we need to compute $f'(x) = \frac{x^2-1}{\sqrt[3]{(x^3-3x)^2}}$. Second, we need to find the value of x for which $f'(x) = \frac{x^2-1}{\sqrt[3]{(x^3-3x)^2}} = 0$. The function $f(x)$ has critical points at $x = \pm 1$. Third, we need to classify the critical points. We plug the values of x into the function f(x): $f(1) = \sqrt[3]{1^3 - 3(1)} = \sqrt[3]{-2} < 0$ $(minimum)$ and $f(-1) = \sqrt[3]{(-1)^3 - 3(-1)} = \sqrt[3]{2} > 0$ $(maximum)$. Figure 3.1 shows the graph of $f(x)$ together with the maxima and minima of the graph.

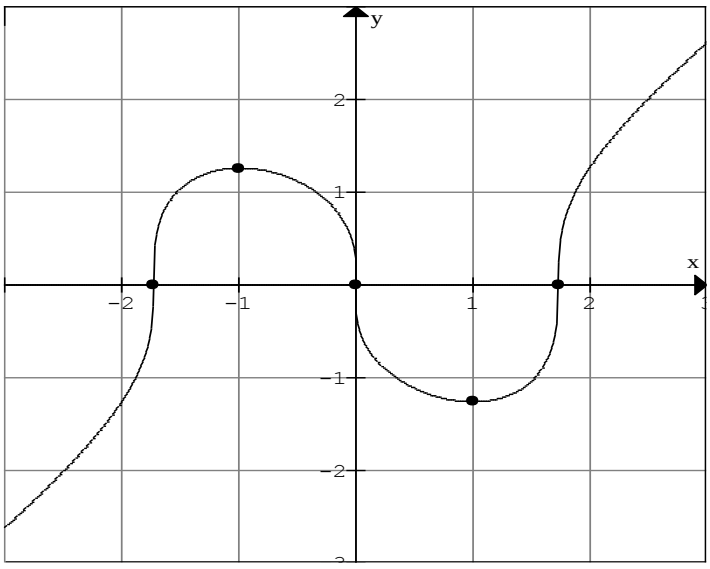


Figure 3.1: The graph of $f(x) = \sqrt[3]{X^3 - 3X}$ shows critical points of the function at $x = \pm 1$.

Example 3.2 Consider the function of two variables $f(x, y) = 4 + X^3 + y^3 - 3xy$. The critical points are the solutions to the system of equations:

$$f'_x = 3x^2 - 3y = 0 \qquad (1)$$

$$f'_y = 3y^2 - 3x = 0 \qquad (2)$$

Solving the first equation, we get $y = x^2$. Plugging this into the second equation gives:

$$3x^4 - 3x = 0 \Rightarrow x(3x^3 - 3) = 0 \Rightarrow$$

$$x = 0 \text{ or } x = 1 \Rightarrow y = 0 \text{ or } y = 1$$

The critical points of this function are $(0, 0), (1, 1)$.

3.2 Absolute, Local Maximum and Minimum Values

The interest in critical points lies in the fact that the point where the function has an extremum (maximum or minimum) is a critical point [13, 14]. The maximum and minimum of a function are the largest and smallest value that the function takes at a point. They are quite distinctive on the graph of a function, and useful in understanding the shape of the graph [12]. In many applied problems we want to find the largest or smallest value that a function achieves and so identifying maximum and minimum points is useful for applied problems [14]. There are two kinds of extrema: global and local (sometimes referred to as "absolute" and "relative", respectively). A global maximum is a point that takes the largest value on the entire range of the function, while a global minimum is the point that takes the smallest value on the range of the function. On the other hand, local extrema are the largest or smallest values of the function on a determined interval [14]. Figure 3.2 shows the absolute local extrema and end points of the given graph.

Definition 3.3 $f\ (x)$ has an absolute maximum value $f\ (c)$ at the point $x = c$ in its domain if $f\ (x)\ \leq f\ (c)$ holds for every x in the domain of $f\ (x)$.

Definition 3.4 $f\ (x)$ has an absolute minimum value $f\ (c)$ at the point $x\ =\ c$ in its domain if $f\ (x)\ \geq f\ (c)$ holds for every x in the domain of $f\ (x\ )$.

Definition 3.5 $f\ (x)$ has a local maximum value $f\ (c)$ at the point $x\ =\ c$ in its domain if $f\ (x)\ \leq f\ (c)$ holds for every x in an interval around c. $f\ (x)$ has a local minimum value $f\ (c)$ at the point $x\ =\ c$ in its domain if $f\ (x)\ \geq f\ (c)$ holds for every x in an interval around c.

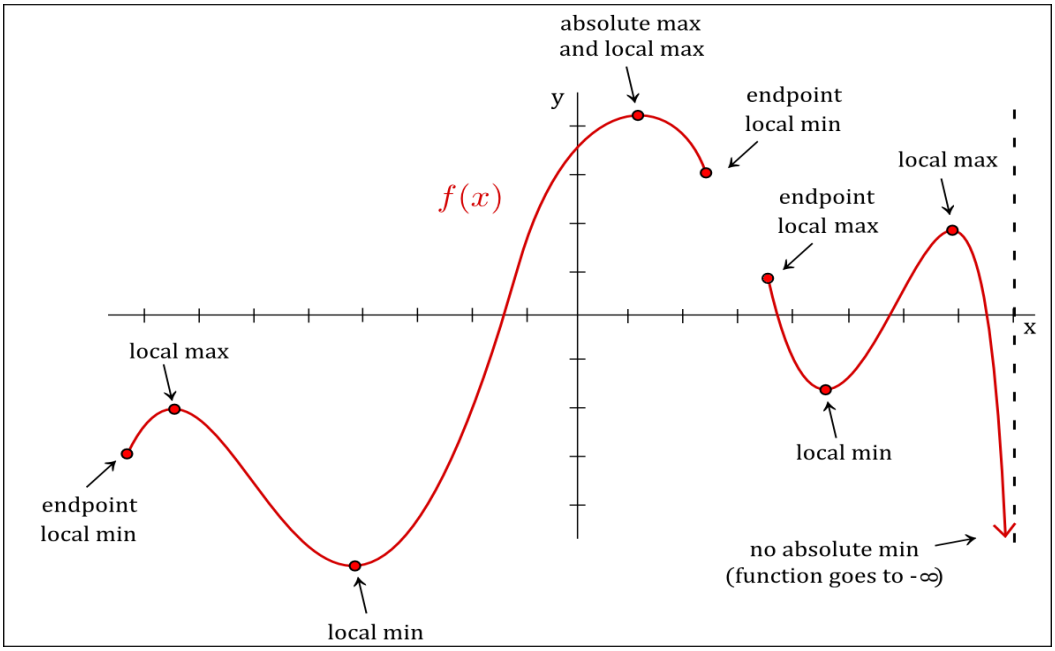


Figure 3.2: Absolute maxima and minima, local maxima and minima, end points of the graph.
Figure from [14].

3.3 First Derivative Test of a Function

In this section, we discuss the information given by the first derivative about the shape of the

graph of a function. The first derivative of the function $f(x)$, $\frac{df}{dx}$, tells us whether a function is increasing or decreasing [13, 15]:

- If $\frac{df}{dx}(p) > 0$, then $f(x)$ is an increasing function at $x = p$.
- If $\frac{df}{dx}(p) < 0$, then $f(x)$ is a decreasing function at $x = p$.
- If $\frac{df}{dx}(p) = 0$ then $x = p$ is called a critical point of $f(x)$, and we do not know anything new about the behavior of $f(x)\ at\ x = p$.

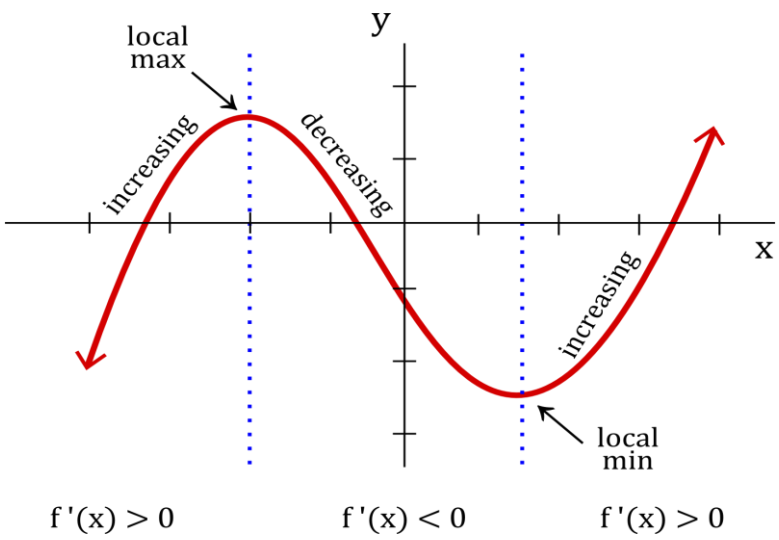


Figure 3.3: The first derivative of the function shows the shape of the curve. Figure from [15].

3.4. Second Derivative Test of a Function

When x is a critical point of a function $f\ (x\ )$, we do not learn anything new about the function at that point. It may be increasing, decreasing, or a local maximum, a local minimum, or an inflection point. We can use the second derivative of the function to find out when *x* is a local maximum, a local minimum, or an inflection point [13, 14].

Suppose that x is a critical point of a function and the second derivative of the function at that point is positive. The positive second derivative at x tells us that the derivative of $f\ (x\ )$ is increasing at that point and, graphically, that the curve of the graph is concave up at that point; then, *x* is a local minimum of $f\ (x\ )$. If x is a critical point of $f\ (x\ )$ and the second derivative of $f\ (x\ )$ is negative, then the curve of the graph is concave down. The only way to draw the corresponding graph is to make the point x a local maximum of the function and the function is decreasing. When x is a critical point of $f\ (x\ )$ and the second derivative of $f\ (x\ )$ is zero, we learn no new information about the point. The point *x* may be a local maximum or a local minimum, or may be an inflection point [13, 14]. The three cases above, (that is, when the second derivative is positive, negative, or zero) are collectively called the *second derivative test* for critical points. The second derivative test gives us a way to classify critical points and, in particular, to find local maxima and local minima (see Figure 3.5). To summarize the second derivative test [13, 14]:

- If $\frac{df}{dx}\ (p) = 0\ and\ \frac{d^2f}{dx^2}\ (p) > 0$, then $f\ (x\ )$ has a local minimum at $x\ =\ p$.

- If $\frac{df}{dx}\ (p) = 0\ and\ \frac{d^2f}{dx^2}\ (p) < 0$, then $f\ (x\ )$ has a local maximum at $x\ =\ p$.

- If $\frac{\mathrm{df}}{\mathrm{dx}}(p) = 0$ and $\frac{d^2f}{dx^2}(p) = 0$, then we learn no new information about the behavior of $f(x)$ at $x = p$.

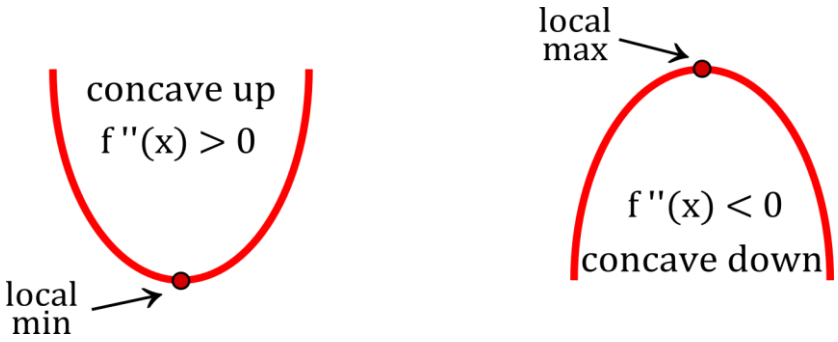


Figure 3.4: The second derivative of the graph classifies critical points to find maxima and minima. For positive second derivative, the function is concave up; for negative second derivative, the function is concave down.
Figure from [14].

Example 3.3 Consider the function $f(x) = x^3 - 9x^2 + 15x - 7$**.** The first derivative of *f(x) is* $\frac{df}{dx} = 3x^2 - 18x + 15$, and the critical points of $f(x)$ are when the first derivative of $f(x)$ equals zero: $\frac{df}{dx} = 3x^2 - 18x + 15 = 0 \Longrightarrow x = 1 \; or \; x = 5$. The critical points of $f(x)$ are 1 and 5. Now, we need to classify the critical points as maxima or minima. We will use the second derivative test *of* $f(x)$*:* $\frac{d^2f}{dx^2} = 6x - 18$. The second derivative test at $x = 1$ gives $\frac{d^2f}{dx^2}(1) = 6(1) - 18 = -12 < 0$ $(local \; maximum)$, and the second derivative test at $x = 5$ gives $\frac{d^2f}{dx^2}(5) = 6(5) - 18 = 12 > 0$ *(local minimum).* We can see from the second derivative that $f(x)$ has a local maximum at $x = 1$, and a local minima at $x = 5$. Now, if the second derivative is zero, we have a problem. It could be a point of inflection, or it could still be an extremum.

3.5 Inflection Points

It should be noticed that not every critical point corresponds to a local maximum or local minimum. A slope of zero does not guarantee a maximum or minimum; there is a third class of stationary point called a *point of inflection* [13]. If we are trying to understand the shape of the graph of a function, knowing where it is concave up and concave down helps us to get more information. The graph of a function $f\ (x\ )$ has an inflection point at x if the graph of the function goes from concave up to concave down, or from concave down to concave up at that point [13].

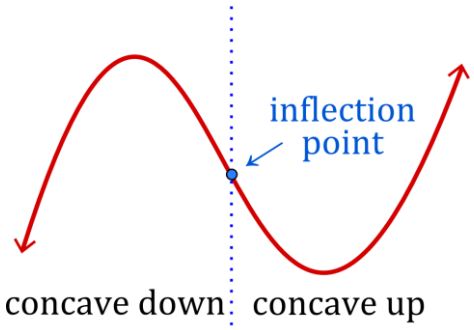


Figure 3.5 : Inflection points when the graph goes from concave up to concave down.
Figure from [13].

Example 3.4 Consider the function $f\ (x\ )\ =\ 3x^3\ +\ 1$. The critical point of the function is $f'(x) = 9x^2 = 0 \Rightarrow x = 0$**.** To classify the critical point, we find the second derivative of the function at the critical point $x = 0$: $f''(x) = 18x = 18(0) = 0$. We learn no new information about the behavior of $f\ (x\ )$ at $x\ =\ 0$. If we plug the value of the point $x = 0$ into the function, we get $3x^2 + 1\ = 1 > 0$, meaning it is a maximum; but, that is not true. Figure 3.7 shows the graph of the function, and there is no maximum at $x = 0$. The second derivative of function $f(x)$ at the critical points is zero, so we do not have any information to help us to classify the point. It may be a maximum, minimum, or inflection point. We can pick a

number on either side of $x = 0$ and check what the concavity is at those locations. Let's use $x = -1$ and $x = 1$ to check. At $x = -1$, the second derivative gives: $f''(-1) = -18 < 0$, so the function is concave down at $x = -1$. If we check $x = 1$ we get $f''(1) = 18 > 0$ which means the function is concave up at $x = 1$. We can see that the function has different concavities on either side of $x = 0$ and the inflection point is at $x = 0$. As we can see, the inflection point is where the concavity actually changes.

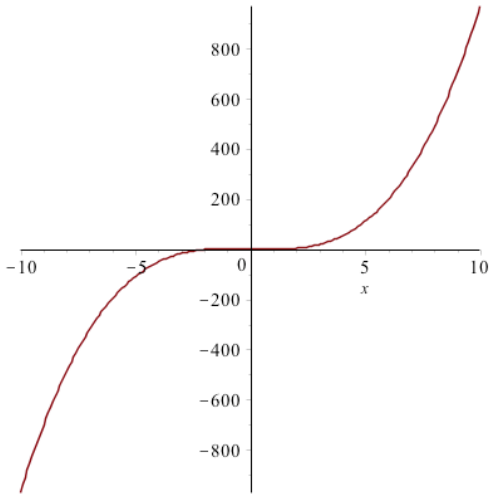


Figure 3.6: The graph of function $f(x) = 3x^3 + 1$.

Theorem 3.1. **(Fermat's Theorem or Local Extreme Point Theorem)** If a function $f(x)$ has a local minimum or maximum at the point c and $f'(c)$ exists, then $f'(c) = 0$.

This means that any time there is a local extremum, the derivative should be zero there. This is not always the case. Consider the function $f(x) = |x|$. Notice that this function is not differentiable at $x = 0$ but since $f(x) = |x| \geq 0 = f(0)$, we see that it has a local minimum at 0 (and in fact, this is a global minimum).

Just having a zero derivative at a local extremum is not the whole picture. It turns out that the only

case where this fails is when the derivative does not exist.

Algorithm 3.1. (Algorithm for finding global minima and global maxima) Let f be a continuous function on a closed interval $[a;\ b]$ (so that our algorithm satisfies the conditions of the extreme value theorem).

- Find all the critical points of (a, b), that is, the points $x \in (a;\ b)$ where $f'(x)$ is not defined or where $f'(x) = 0$. Call these points $x_1, \ldots, x_n$.
- Evaluate $f(x_1), \ldots\ldots, f(x_n),\ f(a), f(b)$, that is, evaluate the function at all the critical points found from the previous step and at the two end point values.
- The largest and the smallest values found in the previous step are the global minimum and global maximum values.

3.6 Singular Points of Algebraic and Parametric Curves

A singular point of a curve is a point on the curve where the tangent line of the curve is not uniquely defined [16]. A point (a, b) on a curve $f(x, y) = 0$ is singular if the x and y partial derivatives of f are both zeros at the point (a, b). Singular points play an essential role in the analysis of geometric curves. They help to define algebraic properties and features of curves. Moreover, the singularities of a curve represent shape features known as cusps, loops, and self-intersections [19]. Thus, detection of singularities helps to determine the geometric shape and topology of real curves, which has wide-ranging applications in computer-aided geometric design [19]. To find the singular points of the implicit curve $f(x, y)$, we compute $\frac{\delta f}{\delta x}$ and $\frac{\delta f}{\delta y}$ and set both to zero. This gives us two equations for the two unknowns x and y. We need to solve these equations for x and y in order to find the singular points of the curve. Suppose we have a curve C defined by equation: $f(x, y) = 0$. We are interested in finding the singular points with respect to the coordinates x and y; that is, the system of equations:

$$f_x(x,y) = 0 \quad , \quad f_y(x,y) = 0$$

Example 3.5 Consider the implicit curve $f(x,y) = y^2 - xy + x^2 - 2y + x$**.** The singular points of the curve are the solutions of the system of equations:

$$\begin{cases} f_x = -y + 2x + 1 = 0 \\ f_y = 2y - x - 2 = 0 \end{cases}$$

By solving the system, we obtain the singular point $(0,1)$.

3.7 Some of Possible Types of Singular Points

Cusp

A cusp is a type of singular point of a curve that is not formed by a self-intersection point of the curve. Consider the curve $F(x,y) = x^3 - y^2 = 0$, and the partial derivatives with respect to x and y: $\frac{\partial f}{\partial x}(x,y) = 3x^2 = 0$, $\frac{\partial f}{\partial y}(x,y) = 2y = 0$. The singular point is $(0,0)$. Figure 3.7 shows the graph of the curve, which has a cusp at the origin [16, 20].

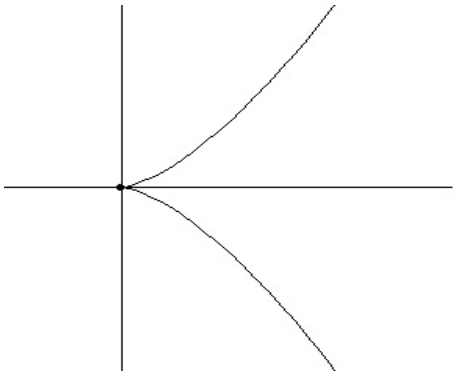


Figure 3.7: An ordinary cusp on the curve $x^3 + y^2 = 0$.
Figure from [16, 20].

Node

A node is a point where a curve intersects itself such that two branches of the curve have distinct tangent lines. It is also called an ordinary double point of a plane curve. Consider the curve $f(x,y) = x^3 - x^2 + y^2$. The singular points are given by $f(x,y) = \frac{\partial f}{\partial x}(x,y) = \frac{\partial f}{\partial y}(x,y) = 0$. Figure 3.8 shows the origin is an ordinary double point of the curve [16, 20].

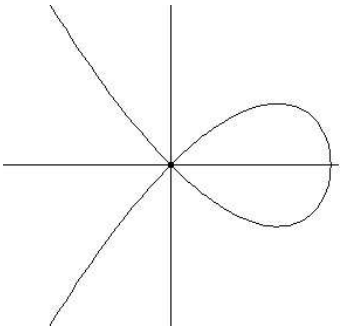


Figure 3.8 : Singular point on the curve $f(x,y) = x^3 - x^2 + y^2$ with a loop and self-intersection.
Figure from [16, 20].

Self-intersection

For an implicitly defined curve q, we refer to points where $q(x,y) = 0$ and $q'(x,y) = (0,0)$ as singularities of the curve. All self-intersections of algebraic curves are singularities, but the converse is not necessarily true. Singularities can also occur as cusps and isolated points known as acnodes [53].

Acnode

An acnode is an isolated point not on a curve, but it satisfies the equation of the curve. Consider the curve $x^3 + x^2 + y^2 = 0$. The origin is an isolated point of the real curve. Figure 3.9 shows the curve [16, 20].

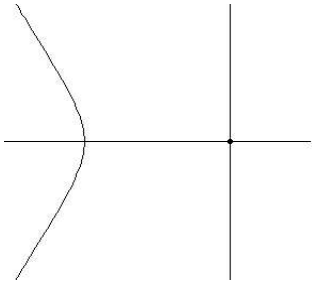
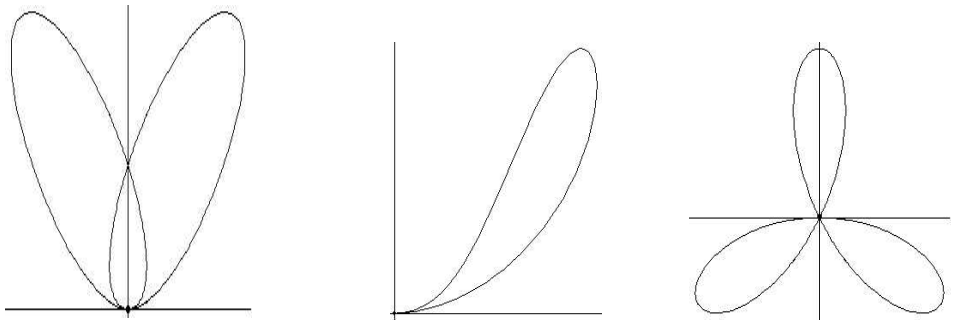


Figure 3.9: Acnode on the curve $x^3 + y^2 = 0$.
Figure from [16, 20].



(a) (b) (c)

$2x^4 - 3x^2y + y^2 - 2y^3 + y^4$ $x^4 + x^2y^2 - 2x^2y - xy^2 + y^2$ $(x^2 + y^2)^2 + 3x^2y - y^3$

(d) (e)

$((x^2 + y^2)^3 - 4x^2y^2 = 0$ $x^6 - x^2y^3 - y^5 = 0$

Figure 3.10: Some algebraic curves with singular points at the origin.
Figure from [16, 19, 20].

Computing singular points based on implicitization is well studied in the literature [19, 22, 23, 25]. One standard way is to convert the parametric equation of the curve into an implicit equation $f(x,\ y) = 0$, and then find the singular points by solving the system of equations: $f = f_x = f_y = 0$, where f_x and f_y are the x and y partial derivatives of f, respectively [22, 23, 25]. There are many existing numerical algorithms for bivariate root-finding based on resultants [23, 24, 27]. One way of obtaining the common solutions is to find those roots of the resultant matrices: $Res_x(f_x\ , f_y) = 0$, $Res_y(f_x, f_y) = 0$ [25]. There are many different types of resultant matrices such as Sylvester [38], Bézout[27], and others [24, 34, 49]. These resultant matrices are usually constructed for polynomials expressed in the monomial basis [28-30], but they can also be constructed for polynomials expressed in other bases [31, 38, 39, 50]. The next chapter presents the different resultant matrix formulations and how they are constructed in the monomial basis and other orthognal bases such as the Chebyshev and the Legendre basis. Theoretical properties of resultants are also discussed.

Chapter 4

Resultants in Orthogonal Bases

A fundamental problem in computational algebra that requires symbolic-numeric computations and manipulations is the computation of resultants. A resultant matrix of two polynomials is a matrix whose entries are functions of their coefficients, such that a necessary and sufficient condition for the polynomials to have a common root is that the determinant of this matrix, called the *resultant* of the polynomials, is exactly zero [28-30]. There is a long history of research into the properties of resultants due to their widespread application in algebraic geometry, computer graphics, computer vision, robotics, and computer-aided geometric design (CAGD) [15-28]. There has been a lot of research and investigation into their theoretical properties.

In computer algebra, the resultant is a tool that can be used to analyze the greatest common divisor (GCD) of polynomials. Two polynomials have common roots if and only if their resultant is zero [29]. Moreover, resultants can be used in algebraic geometry to determine intersections [15, 20]. An important problem in computational geometry is to find the intersection of algebraic and parametric curves, which reduces to the case of determining whether two polynomials have a common root [15, 20]. In fact, there are many applications of resultants, but we will be concerned only with the problem of finding the critical points of algebraic and parametric curves based on resultants [15, 22-25].

The monomial basis is the standard basis in the literature due to its simplicity and flexibility for algebraic manipulations. Usually, resultant matrices are constructed from polynomials

expressed in the monomial basis; however, they can be derived when using any other bases. If polynomials are expressed in other bases, we can transfer the polynomials to the monomial basis, and then compute their resultant matrices [41]. We plan to apply resultant matrices in the same basis (non-monomial basis) to avoid change of basis and the ill-conditioned conversion [41]. This leads us to the problem of studying the generalization of resultant matrices to non-monomial bases like orthogonal bases such as the Chebyshev basis, Legendre basis, and so on [31, 38, 39].

In this chapter, we investigate the problem of computing the resultants of univariate polynomials expressed in orthogonal bases. In the first section, we discuss resultant matrix formulas and their theoretical properties (the most common in the literature, which are the Sylvester resultant matrix, Bezout resultant matrices, and companion resultant matrices) in the monomial basis [28-30]. Orthogonal polynomials (such as Legendre polynomials and Chebyshev polynomials) and their properties are introduced in the second section. In the third section, we investigate the resultants in orthogonal bases, and we plan to find the resultant Sylvester matrix in the Legendre basis.

4.1 Resultants in the Monomial Basis

Many mathematical problems require the computation of the resultant of two polynomials and most approaches to compute resultant matrices assume that the polynomials are expressed in the power basis. There are many different resultant matrices such as the Sylvester matrix, the Bezout matrix, the companion matrix, and others [23, 24]. This section contains a review of the Sylvester [21], Bezout [21, 27], and companion resultant matrices for power basis polynomials, and examples are included in order to show the form of each of the matrices.

4.1.1 Sylvester Matrix

In the literature, the most common method of computing resultants is the Sylvester resultant matrix due to its ease of construction. In this section, we consider Sylvester resultant matrices defined in the power basis.

Definition 4.1 Let $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0$ and $g(x) = b_m x^m + b_{m-1}x^{m-1} + \cdots \ldots + b_0$ be two polynomials of degree *n* and *m*, respectively, with the coefficients $\{a_n, a_{n-1}, \ldots a_0\}$ and $\{b_m, b_{m-1}, \ldots\ldots b_0\}$ respectively. The resultant $Res(f, g)$ is the determinant of the $(m + n)$ by $(m + n)$ *Sylvester matrix*, $Syl(f, g)$, given by [21, 28-30]

$$Syl(f,g) = \begin{bmatrix} a_0 & a_1 & \ldots & a_n & 0 & \ldots & 0 \\ 0 & a_0 & \ldots & a_{n-1} & a_n & \ldots & 0 \\ : & : & : & : & : & : & : \\ 0 & 0 & \ldots & a_0 & a_1 & \ldots & a_n \\ b_0 & b_1 & \ldots & b_m & 0 & \ldots & 0 \\ 0 & b_0 & \ldots & b_m & 0 & \ldots & 0 \\ : & : & : & : & : & : & : \\ 0 & 0 & \ldots & b_0 & \ldots & b_{m-1} & b_m \end{bmatrix} \tag{4.1}$$

where the m first rows contain the coefficients of the polynomial $f(x)$, $\{a_n, a_{n-1}, \ldots . a_0\}$ shifted $0, 1, \ldots . m - 1$ steps and the n last rows contain the coefficients of the polynomial $g(x), \{b_m, b_{m-1}, \ldots \ldots b_0\}$ shifted $0, 1, \ldots n - 1$ steps, and the remaining elements are zero.

Example 4.1: Consider the polynomials $f(x) = 4x^4 + 3x^3 + x^2 + 4x - 7$, and $g(x) = 3x^3 - 6x^2 + x + 1$, expressed in the power basis. The resultant of the two polynomials $f(x)$ and $g(x)$ is the determinant of the Sylvester matrix : $Res(f, g) = Det(Syl(f, g))$:

$$\text{Det}\left(\begin{bmatrix} 4 & 3 & 1 & 4 & -7 & 0 & 0 \\ 0 & 4 & 3 & 1 & 4 & -7 & 0 \\ 0 & 0 & 4 & 3 & 1 & 4 & -7 \\ 3 & -6 & 1 & 1 & 0 & 0 & 0 \\ 0 & 3 & -6 & 1 & 1 & 0 & 0 \\ 0 & 0 & 3 & -6 & 1 & 1 & 0 \\ 0 & 0 & 0 & 3 & -6 & 1 & 1 \end{bmatrix}\right)$$

It is easily seen that $Syl\ (f, g)$ is a square matrix and of order $n + m$ .The entries of the Sylvester matrix of two polynomials are coefficients of the polynomials, and the determinant of the Sylvester matrix of two polynomials is their resultant. A well-known theorem states that $f(x)$ and $g(x)$ are relatively prime if and only if the determinant is non-zero or, alternatively, if and only if $Syl(f, g)$ has full rank. Further, the rank of $Syl(f, g)$ is equal to the difference between the degree of the Sylvester matrix $(m + n)$ and the degree of the greatest common divisor of $f(x)$ and $g(x)$. The degree of the greatest common divisor of $f(x)$ and $g(x)$ is determined by the rank of the Sylvester matrix [29].

Although the resultant doesn't directly reveal a common root, it can be found from the resultant matrix [22]. To illustrate, the resultant of $f(x) = x^2 - 4x - 5$ and $g(x) = x^2 - 7x + 10$ is 0, which means they do have a common root. To find the common root, first we need to discard any row of the resultant matrix. By discarding the fourth row, we get:

$$\begin{bmatrix} 1 & -4 & -5 & 0 \\ 0 & 1 & -4 & -5 \\ 1 & -7 & 10 & 0 \end{bmatrix}$$

The common root is computed by taking the ratio of the determinant of the coefficient matrix with columns i and j, respectively, deleted and multiplying by $(-1)^{i+j}$:

$$x = \frac{x^i}{x^j} = (-1)^{i+j}\ \frac{|A_i|}{|A_j|} \qquad 4.2$$

The common root x can be found as either x^3/x^2, x^2/x, or $x/1$:

$$x = \frac{x^1}{x^2} = -\begin{vmatrix} -4 & -5 & 0 \\ 1 & -4 & -5 \\ -7 & 10 & 0 \end{vmatrix} \Big/ \begin{vmatrix} 1 & -5 & 0 \\ 0 & -4 & -5 \\ 1 & 10 & 0 \end{vmatrix} = -\frac{-375}{75} = 5$$

Theorem 4.1. There exist polynomials A and B, of degree at most $m-1$ and $n-1$, such that $Res = Af + Bg$.

Proof. We have:

$$f = a_0 + a_1 x + \cdots + a_n x^n$$

$$xf = a_0 x + \cdots + a_{n-1} x^n + a_n x^{n+1}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$x^{m-1} f = a_0 x^{m-1} + \cdots + a_n x^{m+n+1}$$

$$g = b_0 + b_1 x + \cdots + b_m x^m$$

$$xg = b_0 x + \cdots + b_{m-1} x^m + a_m x^{m+n-1}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$x^{n-1} g = b_0 x^{n-1} + \cdots + b_m x^{m+n+1}$$

Let $A_1, \ldots, A_{m+n}$ be the cofactors of the elements of the first column of Res. If we multiply the i^{th} equation above by A_i and add corresponding term for $i = 1, \ldots, m+n$, we obtain:

$$(A_0 + A_1 x + \cdots + A_m x^{m-1}) f + (A_{m+1} + A_{m+2} x + \cdots A_{m+n} x^{n-1}) g = Res,$$

which proves the theorem [16, 20].

Another matrix, called the Bezout matrix [21, 27, 50], can be used to compute the resultant of two polynomials. It is discussed in the next section.

4.1.2 Bezout Matrix

Definition 4.3 For two polynomials in one variable $p(x)$ and $q(x)$ of degree n and m, respectively, expressed in the monomial basis, there exists a uniquely determined $n \times n$ symmetric matrix, since it is assumed that $n \geq m$ [21, 27]

$$B_n(p,q) = \left(b_{ij}\right)_{ij=1}^{n} \tag{4.3}$$

such that

$$\frac{P(x)q(y)-q(x)p(y)}{x-y} = \sum_{i,j=1}^{n} b_{ij} X^{i-1} y^{j-1}. \tag{4.4}$$

The matrix $B(p,q)$ is called the Bezout matrix of polynomials p and q. The Bezout matrix is smaller than the Sylvester matrix, but has more complicated entries, and its determinant still equals the resultant of the polynomials.

Example 4.2 Consider two polynomials, expressed in the monomial basis, $p(x) = 3x + 3$, $q(x) = x^3 - x^2 + 2$. Their Bezout matrix is:

$$B_3(p,q) = \begin{bmatrix} 6 & 0 & -6 \\ 3 & 3 & 0 \\ 0 & 3 & 3 \end{bmatrix}$$

Let us now recall some essential properties of the Bezout matrix:

1. $B(p,q)$ is an $N \times N$ *symmetric matrix*, where $N = max\{n,m\}$

2. $B(p,q) = -B(p,q)$

3. $B(p,q)$ is linear in p and q, that is:

$$B(\alpha p, \beta f, v) = \alpha B(p,q) + \beta B(f,q),$$

$$B(p, \alpha q, \beta g) = \alpha B(p,q) + \beta B(q,g),$$

for any polynomials $f(x), g(x)$ and scalars α, β.

4. $B(p,q)$is invertible if and only if $p(x)$and $q(x)$are coprime:

$$dim(kerB(p,q)) \text{ is equal to the degree of } GCD(p,q).$$

All of these properties can be found in computer algebra texts [28-30].

4.1.3 Companion Matrix

Finding the roots of a polynomial is one of the main applications of the resultant. We now show how to use the resultant to find the eigenvalues of polynomials. Suppose polynomials $A(x) = \sum_{i=0}^{n} a_i x^i$ and $x - \alpha$ have common root, meaning $A(\alpha) = 0$. The resultant matrix of these polynomials can be written as [42]:

$$M = \begin{bmatrix} a_0 & a_1 & a_2 & \cdot\cdot & a_{n-1} & a_n \\ 1 & -\alpha & 0 & 0 & 0 & 0 \\ 0 & 1 & -\alpha & 0 & 0 & 0 \\ 0 & 0 & 1 & -\alpha & 0 & 0 \\ \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot & \cdot\cdot \\ 0 & 0 & \cdot\cdot & \cdot\cdot & 1 & -\alpha \end{bmatrix} \quad 4.5$$

If $a_0 = 1$, we can subtract the first row from the second row to eliminate the first element in the second row and we end up with a matrix N. The determinant of matrix N can be obtained by crossing off the first row and column of N, and taking the determinant of the resulting matrix. This determinant is the characteristic polynomial of the matrix [42]:

$$C = \begin{bmatrix} -a_1 & -a_2 & \dots & \dots & -a_{n-1} & -a_n \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ : & : & : & : & : & : \\ 0 & 0 & : & : & 1 & 0 \end{bmatrix} \quad 4.6$$

The roots of the polynomial $A(x)$ are the eigenvalues of the matrix C. The matrix C is called the *companion matrix*.

4.1.4 Properties of Resultants

This section reviews the basic properties of resultants. If $f(x)$ and $g(x)$ are two polynomials of degree $m \; and \; n$ respectively, and they are expressed in term of their zeros such that

$$f(x) = \; a_m \prod_{k=1}^{m}(x - \alpha_k)$$

and

$$g(x) = b_n \prod_{j=1}^{n}(x - \beta_j),$$

then the resultant can be expressed by the products:

$$Res(f, g) = a_m^n \prod_{i=1}^{m} g(\alpha_i) = b_n^m \prod_{j=1}^{n} f(\beta_j) \qquad 4.7$$

$$Res(f, g) = \; a_m^n b_n^m \; \prod_{i=1}^{m} \prod_{j=1}^{n}(\alpha_i - \beta_j) \qquad 4.8$$

Other well-known properties of resultants are:

$$Res(f, g) = (-1)^{mn} Res(g, f) \qquad 4.9$$

$$Res(f, qg) = \; Res(f, q) Res(f, g) \qquad 4.10$$

If $a \neq 0$ is a constant, then

$$Res(f, a) = \; Res(a, f) = \; a^m. \qquad 4.11$$

The fundamental property of the resultant is stated in the following theorem:

Theorem 4.2 Let f and g be polynomials. Then $Res\;(f, g) \; = \; 0$ if and only if f and g have common roots.

Lemma 4.1 Let f and g be polynomials

1- If we can write $f(x) = q(x)g(x) + r(x)$, with polynomials $q, r, and\ \delta =$ $\deg(r)$, then $res(f,g) = b_n^{m-\delta}\, res(g,r)$.

2- $If\ deg(qf,g) = \deg g$ for a polynomial q, then $res(f,qf)+g) = res(f,\ g)$.

All of these properties and proofs of these theorems are well known and can be found in most linear algebra texts [28-30].

The resultant matrices reviewed in this section are the most familiar methods of computing resultants between polynomials written in the monomial basis $\{1, x, x^2, \cdots, x^n\}$. However, resultant matrices can be written relative to polynomial bases other than the monomial basis.

Bases other than the monomial basis find many applications. Orthogonal bases such as the Legendre polynomials and Chebyshev polynomials are often the most useful in many applications including interpolation and least-squares approximation, and approximation theory. They are more stable for numerical analysis than the monomial basis [44]. This motivates us to compute resultant matrices in orthogonal bases rather than perform a transformation to the monomial basis. In the next subsection, we give an overview of the most important classical orthogonal polynomials and their applications.

4.2 Orthogonal Polynomials

A set of orthogonal polynomials is an infinite set of sequence of polynomials $B_0(x), B_1(x), B_2(x),\ B_3(x), \ldots\ldots B_n(x)$, such that any family of polynomials orthogonal on [a, b] with respect to an inner product of the form 4.12 [44]. Table 4.3 shows the common classical orthogonal polynomails.

$$\langle B_i, B_j \rangle = \int_a^b B_i(x) B_j(x) w(x)\, dx \qquad 4.12$$

$$\langle B_i, B_j \rangle = 0 \quad if \quad i \neq j$$

Orthogonal polynomials can be obtained by applying the Gram-Schmidt orthogonalization process to the monomial basis $\{1, x, x^2, \ldots\}$ [47]:

$$B_0(x) = 1,$$

$$B_1(x) = x - \frac{\langle x, B_o \rangle}{\langle B_0, B_0 \rangle} P_0(x),$$

$$B_1(x) = x^2 - \frac{\langle x^2, B_0 \rangle}{\langle B_0, B_0 \rangle} P_0(x) - B_1(x) = x - \frac{\langle x^2, B_1 \rangle}{\langle B_1, B_1 \rangle} P_1(x),$$

$$\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$$

$$B_n(x) = x^n - \frac{\langle x^n, B_o \rangle}{\langle B_0, P_0 \rangle} B_0(x) - \cdots - \frac{\langle x^n, B_{n-1} \rangle}{\langle B_{n-1}, B_{n-1} \rangle} B_{n-1}(x),$$

$B_0, B_1, B_2, \ldots \ldots .. B_n$ are orthogonal polynomials.

4.2.1 Three-Term Recurrence Relation

Any orthogonal polynomial has a recurrence formula given by [45]:

$$B_{-1} = 0$$

$$B_0 = 1$$

$$B_{(n+1)} = (a_n x + b_n) B_n - c_n B_{n-1} \qquad n = 0, 1, 2, \ldots \ldots$$

The three-term recurrence relation satisfied by orthogonal polynomials is very important information for the constructive and computational use of the orthogonal polynomials. It is

used to generate the values of orthogonal polynomials and their derivatives. It also allows the zeros of orthogonal polynomials to be computed, and it allows an efficient evaluation of expansion in orthogonal polynomials [45, 46]. The most widely used orthogonal polynomials are the classical orthogonal polynomials consisting of the Hermite polynomials, the Laguerre polynomials, and the Jacobi polynomials, together with their special cases the Chebyshev polynomials and the Legendre polynomials [44].

4.2.2 Example of Orthogonal Polynomials

In this subsection, we briefly discuss properties of Legendre, Jacobi, and Chebyshev polynomials, which are classical examples of orthogonal families of polynomials.

Legendre Polynomials

Legendre polynomials are usually used in numerical analysis such as approximation theory. They are orthogonal in the interval $[-1, 1]$, and satisfy the 3-term recurrence relation [46]:

$$(n+1)P_{n+1}(x) = (2n+1)x\,P_n(x) - nP_{n-1}(x) \tag{4.13}$$

$$P_0(x) = 1$$

$$P_1(x) = x$$

They may be expressed using Rodrigues's formula

$$P_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}(x^2-1)^n. \tag{4.14}$$

Legendre's differential equation is:

$$\frac{d}{dx}\left[(1-x^2)\frac{d}{dx}P_n(x)\right] + n(n+1)\,P_n(x) = 0 \tag{4.15}$$

Table 4.1: Legendre Polynomials [46].

$P_0 = 1$
$P_1 = x$
$P_2 = \frac{3}{2}x^2 - \frac{1}{2}$
$P_3 = \frac{5}{2}x^3 - \frac{3}{2}x$
$P_4 = \frac{35}{8}x^4 - \frac{30}{8}x^2 + \frac{3}{8}$
$P_5 = \frac{63}{8}x^5 - \frac{70}{8}x^3 + \frac{15}{8}x$
$P_6 = \frac{231}{16}x^6 - \frac{315}{16}x^4 + \frac{105}{16}x^2 - \frac{5}{16}$

Figure 4.1 shows Legendre polynomials up to degree 5. Table 4.1 shows the Legendre polynomials up to degree 6.

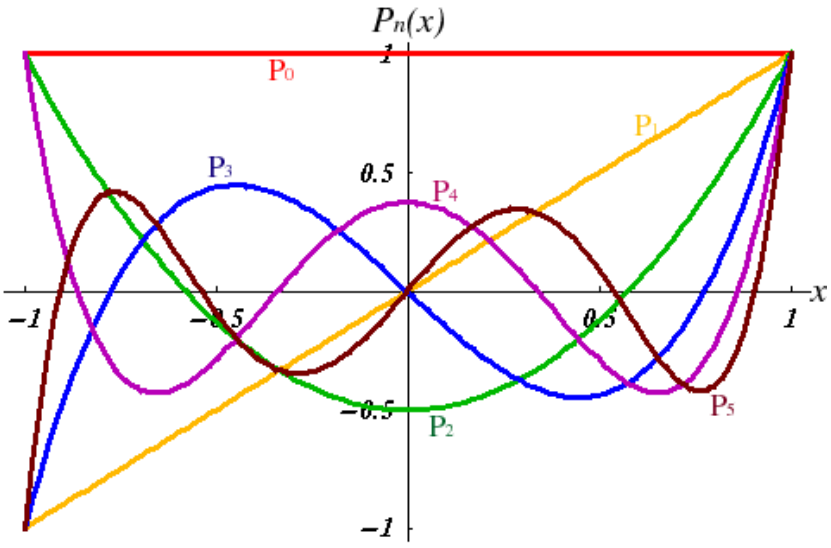


Figure 4.1: The graphs of the Legendre polynomials (up to n = 5).
Figure from [46].

Chebyshev Polynomials

Chebyshev polynomials are important in approximation theory.They are orthogonal with respect to the weight $\frac{1}{\sqrt{(1-x^2)}}$ on the interval $[-1, 1]$. The classical Chebyshev polynomial of the first kind is defined by the three following recurrence formula [44]:

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \tag{4.16}$$

$$T_1(x) = x$$

$$T_0(x) = 1$$

The Chebyshev differential equation is:

$$(1 - x^2)y'' - xy + n^2y = 0 \tag{4.17}$$

The product formula of Chebyshev polynomials of the first kind is [45]:

$$T_i(x)T_j(x) = \frac{1}{2}(T_{i+j}(x) + T_{|i-j|}(x)) \tag{4.18}$$

Figure 4.2 shows the graphs of Chebyshev polynomials of the first kind up to degree 5. Table 4.2 shows the values of Chebyshev polynomials up to degree 5.

Table 4.2: Chebyshev Polynomials [45]

$T_0 = 1$
$T_1 = x$
$T_2 = 2x^2 - 1$
$T_3 = 4x^3 - 3x$
$T_4 = 8x^4 - 8x^2 + 1$
$T_5 = 16x^5 - 20x^3 + 5x$
$T_6 = 32x^6 - 48x^4 + 18x^2 - 1$

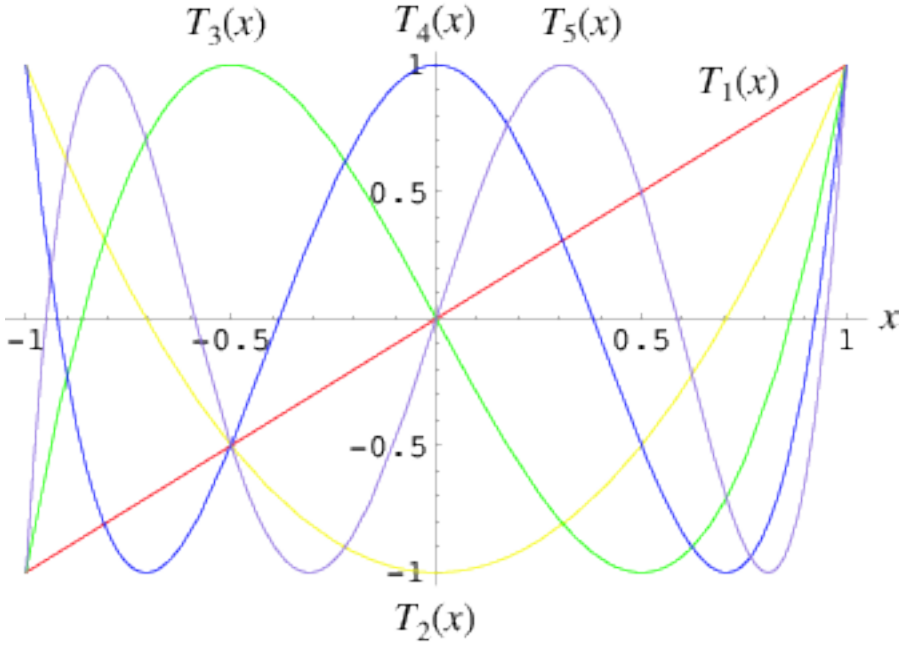


Figure 4.2: The graphs of the Chebyshev polynomials of the first kind in the interval $-1 \leq x \leq 1$ up to $n = 5$.
Figure from [45].

Jacobi Polynomials

Jacobi polynomials $P_n^{\alpha,\beta}(x)$ are a class of classical orthogonal polynomials [44]. They are orthogonal with respect to the weight $(1 - x)^{\alpha}(1 + x)^{\beta}$ on the interval $[-1, 1]$:

$$\int_{-1}^{1} (1-x)^{\alpha}(1+x)^{\beta} P_n^{\alpha,\beta}(x) P_m^{\alpha,\beta}(x) dx$$

Legendre and Chebyshev polynomials are special cases of Jacobi polynomials. For $\alpha = \beta = 0$, it is called a Legendre polynomial with weight function 1, and for $\alpha = \beta = \pm 1/2$, we obtain the Chebyshev polynomial [44].

Table 4.3 Most Common Classical Orthogonal Polynomials [45].

Polynomial Name	Notation	Interval	Weight function $\omega_i(x)$
Chebyshev of the first kind	$T_n(x)$	[-1,1]	$\frac{1}{\sqrt{(1-x^2)}}$
Chebyshev of the second kind	$U_n x$	[-1,1]	$\sqrt{(1-x^2)}$
Legendre polynomial	$P_n(x)$	[-1, 1]	1
Jacobi polynomial	$P_n^{(\alpha,\beta)}(x)$	(-1, 1)	$(1-x)^{\sigma}(1+x)^{\beta}$
Hermite polynomial	$H_n(x)$	$(-\infty, \infty)$	e^{-x^2}

4.3 Resultant in Orthogonal Bases

Polynomials can be expressed in many different bases such as the monomial, Chebyshev, Legendre, and any orthogonal bases. Each basis has its advantages and disadvantages, and by choosing the appropriate bases, many problems can be solved. For example, orthogonal basis polynomials such as Chebyshev, Legendre, and Legendre-Sobolev are very useful for polynomial approximation in handwriting recognition. They have the advantage of providing a high recognition rate. Since symbolic methods are too slow for practical applications of resultants such as computer-aided geometric design, the numerical computation of resultants must be considered. In this section, we compute resultant matrices in orthogonal bases.

4.3.1 Basis Conversion among Power Basis Polynomial Representations and Orthogonal Basis Polynomial Representation

When polynomials are expressed in orthogonal bases, we are able to convert them into polynomials in the power basis, compute resultant matrices, and then convert back to the right orthogonal basis. In many theoretical and practical applications, several representations of polynomials are used, and they can be converted between each other. In the literature, there exist several conversion algorithms [32-37] for changing from any univariate basis (monomial and orthogonal polynomials: Legendre, Chebyshev, and Jacobi) into another one. In this section, we discuss a basis conversion from an orthogonal basis to the monomial basis.

As we mentioned above (in the orthogonal polynomials section) the sequence of orthogonal polynomials F_i can be defined by:

$$F_{-1} = 0$$

$$F_0 = 1$$

$$F_i = (a_i x + b_i)F_{i-1} + c_i F_{i-2}$$

In order to perform the basis change between the orthogonal basis (F_i) and the monomial basis (x^i), we need to study the following problems [35].

Expansion Problem (Expand): Given the coefficients $\{\alpha_0, \dots \alpha_{n-1}\}$ in an orthogonal basis, compute the coefficients on the monomial basis of the polynomial A defined by the map

$$[\alpha_0, \dots \alpha_{n-1}] \rightarrow A = \sum_{i=0}^{n-1} \alpha_i F_i \qquad 4.19$$

Decomposition Problem (Decomp): Conversely, given the coefficients of A in the monomial basis, recover the coefficients $\{\alpha_0, \dots \alpha_{n-1}\}$.

Let F_n be $n \times n$ translation matrix; then, multiply the matrix F_n by the vector $[\alpha_0 \dots \dots \alpha_{n-1}]^t$ for problem Expand. F_n is an upper triangular matrix whose diagonal entry is non-zero; so, it is linearly independent and its matrix inverse can be computed [35]. Next two sections, we introduce Legendre-Power and Chebyshev-Power basis transformation.

4.3.1.1 Legendre-Power Basis Transformation

A polynomial $Q(x)$ of degree n can be written in the form of the power basis and the Legendre basis as[34]:

$$Q(x) = \sum_{i=0}^{n} a_i X^i \ = \sum_{j=0}^{n} l_j P_j(x).$$

In this section, we try to find the entries of the translation matrix *M* which transforms the coefficients of the power basis $\{a_0, a_1, a_2, \ldots\ldots a_n\}$ to the coefficients of the Legendre basis $\{L_0, L_1, L_2, \ldots\ldots L_n\}$:

$$L_j = \sum_{i=0}^{n} a_i M_{ij} \ . \qquad 4.20$$

Introducing the vectors $a^t = [a_0, a_1, a_2, \ldots. a_n]$ and $l^t = [l_0, l_1, l_2, \ldots\ldots\ldots l_n]$, we can write (4.20) in matrix-vector form as

$$l = aM. \qquad 4.21$$

By computing the inverse of M, we can get the monomial coefficients from the orthogonal polynomial [34]:

$$a = lM^{-1}. \qquad 4.22$$

The transition matrix M from the monomial basis to the Legendre basis can be computed from the recursive relation of the Legendre polynomials.

$$\begin{bmatrix} 1 \\ X \\ X^2 \\ \ldots\ldots \\ X^n \end{bmatrix} = M * \begin{bmatrix} P_0(x) \\ P_1(x) \\ P_2(x) \\ \ldots\ldots \\ P_n(x) \end{bmatrix} \qquad 4.23$$

$$M = \begin{bmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ \frac{1}{3} & 0 & \frac{2}{3} & & & & & \\ 0 & \frac{3}{5} & 0 & \frac{2}{5} & & & & \\ \frac{7}{35} & 0 & \frac{20}{35} & 0 & \frac{8}{35} & & & \\ 0 & \frac{27}{63} & 0 & \frac{28}{63} & 0 & \frac{8}{63} & 0 & \frac{8}{63} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

By using Equation (4.23), a polynomial expressed by the power basis

$$\alpha(x) = \alpha_n x^n + \alpha_{n-1} x^{n-1} + \cdots . \alpha_1 x + \alpha_0$$

can be expressed by the Legendre polynomial basis in the following form:

$$\alpha(x) = [\alpha_0 \ \alpha_1 \ \alpha_2 \ldots\ldots \alpha_n] M \begin{bmatrix} P_0(x) \\ P_1(x) \\ \cdots \cdots \\ P_n(x) \end{bmatrix}$$

M is linearly independent and it has full rank, so it is invertible. We have M^{-1} given by:

$$M^{-1} = \begin{bmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ -\frac{1}{2} & 0 & \frac{3}{2} & & & & & \\ 0 & -\frac{3}{2} & 0 & \frac{5}{2} & & & & \\ \frac{3}{8} & 0 & -\frac{15}{4} & 0 & \frac{35}{8} & & & \\ 0 & \frac{15}{8} & 0 & -\frac{35}{4} & 0 & \frac{63}{8} & 0 & \frac{8}{63} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

A polynomial expressed by the Legendre basis,

$$\beta(x) = \beta_n P_n(x) + \beta_{n-1} P_{n-1}(x) + \cdots . \beta_1 P_1 + \beta_0,$$

can be expressed by the power polynomial basis in the following form [34]:

$$\beta(x) = [\beta_0 \ \beta_1 \ \beta_2 \ldots\ldots \beta_n] \, M^{-1} \begin{bmatrix} x^0 \\ x^1 \\ \ldots\ldots \\ x^n \end{bmatrix} \tag{4.24}$$

Example 4.5 Consider two polynomials expressed in the Legendre basis, $A(x) = 3P_3(x) - 2P_2(x) - 3P_1(x) + 10P_0(x)$ and $B(x) = P_3(x) + 5P_2(x) - 7P_1(x) + 11P_0(x)$**.** Using equation 4.20, we can obtain the polynomials in the power basis:

$$A(x) = [10 \quad -3 \quad -2 \quad 3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{3}{2} & 0 \\ 0 & -\frac{3}{2} & 0 & \frac{5}{2} \end{bmatrix} \begin{bmatrix} x^0 \\ x^1 \\ x^2 \\ x^3 \end{bmatrix} = 11 - \frac{15}{2}x - 3x^2 + \frac{15}{2}x^3$$

$$B(x) = [11 \quad -7 \quad 5 \quad 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{1}{2} & 0 & \frac{3}{2} & 0 \\ 0 & -\frac{3}{2} & 0 & \frac{5}{2} \end{bmatrix} \begin{bmatrix} x^0 \\ x^1 \\ x^2 \\ x^3 \end{bmatrix} = \frac{17}{2} - \frac{17}{2}x + \frac{15}{2}x^2 + \frac{5}{2}x^3$$

An algorithm for converting a polynomial in the power basis to a polynomial in the Legendre basis can be deduced as follows [40].

Let $P(x) = \sum_{k=0}^{n} a_k x^k$ and $P(x) = \sum_{k=0}^{n} b_k P_k$ be the polynomials in the power basis and the Legendre basis, respectively. Then:

$$P_{n+1} = A_n x P_n + C_n P_{n-1}$$

$$\Longrightarrow xP_n = \frac{1}{A_n}P_{n+1} - \frac{C_n}{A_n}\,P_{n-1} \qquad (n > 1),$$

$$P_1 = A_0 x P_0 \Longrightarrow xP_o = \frac{1}{A_0}\,P_1$$

we have

$$P_n = \sum_{K=0}^{j-1} a_k x^k \; + x^j \sum_{K=0}^{n-j} b_k P_k \;(x)$$

$$\Rightarrow \sum_{k=0}^{j} a_k x^k + x^j \sum_{k=1}^{n-j-1} b_k^{j+1} \left(\frac{1}{A_k}\,P_{k+1} - \frac{C_k}{A_k}P_{k-1}\right) + x^{j+1} b_0^{j+1}\,P_0$$

4.3.1.2 Chebyshev-Power Basis Transformation

A polynomial $Q(x)$ of degree n can be written in the form of the power basis and the Chebyshev basis as

$$Q(x) = \sum_{i=0}^{n} a_i x^i \;= \sum_{j=0}^{n} c_j T_j(x).$$

The translation matrix N transforms the coefficients in the power basis $\{a_0, a_1, a_2, \ldots\ldots a_n\}$ to the coefficients in the Chebyshev basis $\{c_0, c_1, c_2, \ldots\ldots c_n\}$ where

$$c_j = \sum_{i=0}^{n} a_i N_{ij}\,. \qquad 4.25$$

Introducing the vectors $a^t = [a_0, a_1, a_2, \ldots. a_n]$ and $c^t = [c_0, c_1, c_2, \ldots\ldots\ldots c_n]$, we can write (4.25) in matrix- vector form as

$$c = aN.$$

By computing the inverse of N, we can get the monomial coefficients from the Chebyshev polynomial

$$a = cN^{-1}.$$

The transition matrix N from the monomial basis to the Chebyshev basis can be computed from the recursive relation of the Chebyshev polynomial.

$$\begin{bmatrix} 1 \\ x \\ x^2 \\ \cdots \\ x^n \end{bmatrix} = N \begin{bmatrix} T_0(x) \\ T_1(x) \\ T_2(x) \\ \cdots \\ T_n(x) \end{bmatrix} \tag{4.26}$$

$$N = \begin{bmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & & & & \\ 0 & \frac{3}{4} & 0 & \frac{1}{4} & & & & \\ \frac{3}{8} & 0 & \frac{1}{2} & 0 & \frac{1}{8} & & & \\ 0 & \frac{5}{8} & 0 & \frac{5}{16} & 0 & \frac{1}{16} & & \\ \frac{5}{16} & 0 & \frac{15}{32} & 0 & \frac{3}{16} & 0 & \frac{1}{32} & \\ 0 & \frac{35}{64} & 0 & \frac{21}{64} & 0 & \frac{7}{64} & 0 & \frac{1}{64} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

N is linearly independent and has full rank, so it is invertible. We have:

$$N^{-1} = \begin{bmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ -1 & 0 & 2 & & & & & \\ 0 & -3 & 0 & 4 & & & & \\ 1 & 0 & -8 & 0 & 8 & & & \\ 0 & 5 & 0 & -20 & 0 & 16 & & \\ -1 & 0 & 18 & 0 & -48 & 0 & 32 & \\ 0 & -7 & 0 & 56 & 0 & -112 & 0 & 64 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

4.3.2 Transformation of the Sylvester Matrix Resultant between the Power Basis and Orthogonal bases

A simple way to compute a resultant matrix in the Legendre basis is to transform the polynomials to the power basis, compute their Sylvester matrix in the power basis, and then convert it back to the orthogonal basis. The transformation of the Sylvester resultant matrix between the power and orthogonal bases is computed in this section.

Consider two polynomials of degree n and m respectively, expressed in the Legendre basis: $f(x) = \sum_{i=0}^{n} a_i P_i \ and \ g(x) = \sum_{j=0}^{m} b_j P_j$ with$n \geq m$. To compute their Sylvester matrix, consider the equation:

$$\begin{bmatrix} a_0 & a_1 & \dots & a_n & 0 & \dots & 0 \\ 0 & a_0 & \dots & a_{n-1} & a_n & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_0 & a_1 & \dots & a_n \\ b_0 & b_1 & \dots & b_m & 0 & \dots & 0 \\ 0 & b_0 & \dots & b_m & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & b_0 & \dots & b_{m-1} & b_m \end{bmatrix} \cdot \begin{bmatrix} x^0 \\ x^1 \\ \dots \\ x^m \\ \dots \\ x^n \\ x^{n+1} \\ \dots \\ x^{m+n-1} \end{bmatrix} = \begin{bmatrix} f(x)x^0 \\ f(x)x^1 \\ f(x)x^2 \\ \dots \dots \\ f(x)x^{m-1} \\ g(x)x^0 \\ g(x)x^1 \\ g(x)x^2 \\ \dots \dots \\ g(x)x^{n-1} \end{bmatrix} \Longrightarrow [LSyl \quad (f,g)] \cdot \begin{bmatrix} P_0(x) \\ P_1(x) \\ \dots \\ P_m(X) \\ \dots \\ P_n(x) \\ P_{n+1}(x) \\ \dots \\ P_{m+n-1}(x) \end{bmatrix}$$

$$= \begin{bmatrix} P_0(x)f(x) \\ P_1(x)f(x) \\ P_2(x)f(x) \\ \dots \\ P_{m-1}(x)f(x) \\ P_0(x)g(x) \\ P_1(x)g(x) \\ P_2(x)g(x) \\ \dots \\ P_{n-1}(x)g(x) \end{bmatrix}$$

Multiplying the Sylvester matrix in the power basis by the translation matrix *M*, we get the

Sylvester matrix in the Legendre basis:

$$LSyl_x(f,g) = Syl(f,g) * M \qquad 4.27$$

$$\begin{bmatrix} a_0 & a_1 & \dots & a_n & 0 & \dots & 0 \\ 0 & a_0 & \dots & a_{n-1} & a_n & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_0 & a_1 & \dots & a_n \\ b_0 & b_1 & \dots & b_n & 0 & \dots & 0 \\ 0 & b_0 & \dots & b_n & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & b_0 & \dots & b_{m-1} & b_m \end{bmatrix} * \begin{bmatrix} 1 & & & & & & & \\ 0 & 1 & & & & & & \\ \frac{1}{3} & 0 & \frac{2}{3} & & & & & \\ 0 & \frac{3}{5} & 0 & \frac{2}{5} & & & & \\ \frac{7}{35} & 0 & \frac{20}{35} & 0 & \frac{8}{35} & & & \\ 0 & \frac{27}{63} & 0 & \frac{28}{63} & 0 & \frac{8}{63} & 0 & \frac{8}{63} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \Longrightarrow LSyl_x(f,g)$$

Consider two polynomials in the power basis: $A(x) = 11 - 6x - 3x^2 + 8x^3$ and $B(x) = 3 - 9x + 2x^2 + 5x^3$. Their Sylvester matrix in the power basis is:

$$\begin{bmatrix} 11 & -6 & -3 & 8 & 0 & 0 \\ 0 & 11 & -6 & -3 & 8 & 0 \\ 0 & 0 & 11 & -6 & -3 & 8 \\ 3 & -9 & 2 & 5 & 0 & 0 \\ 0 & 3 & -9 & 2 & 5 & 0 \\ 0 & 0 & 3 & -9 & 2 & 5 \end{bmatrix}$$

By using equation 4.27, we obtain the Sylvester matrix in the Legendre basis:

$$LSyl(f,g) = \begin{bmatrix} 10 & -\frac{6}{5} & -2 & \frac{16}{5} & 0 & 0 \\ -\frac{2}{5} & \frac{46}{5} & \frac{4}{7} & -\frac{6}{5} & \frac{64}{35} & 0 \\ \frac{46}{15} & -\frac{6}{35} & \frac{118}{21} & \frac{52}{45} & -\frac{24}{35} & \frac{64}{63} \\ \frac{11}{3} & -6 & \frac{4}{3} & 2 & 0 & 0 \\ -2 & \frac{21}{5} & -\frac{22}{7} & \frac{4}{5} & \frac{8}{7} & 0 \\ \frac{7}{5} & -\frac{114}{35} & \frac{22}{7} & -\frac{62}{45} & \frac{16}{35} & \frac{40}{63} \end{bmatrix}$$

It has been shown that this change of basis is ill conditioned [34]. It is concluded that the Sylvester matrix resultant should be constructed and computed in the Legendre basis, such that the power basis is not used. In the next section, computation of the Sylvester matrix in the Legendre basis is considered.

4.3.3 Computation of the Sylvester Matrix Resultant in Orthogonal Bases

In order to compute the Sylvester matrix resultant in an orthogonal basis, we begin with the familiar construction of the Sylvester matrix in the monomial basis. Consider two polynomials of degree n and m respectively: $f(x) = \sum_{i=0}^{n} a_i x^i$ and $g(x) = \sum_{j=0}^{m} b_j x^j$. For the m first columns, for the first column of the matrix, we have x^0 times the coefficients of $f(x)$. For the second column, we have x^1 times the coefficients of $f(x)$, and so on until the m^{th} column, where we have x^{m-1} times the coefficients of $f(x)$. For the n last columns, for the first column of the matrix, we have x^0 times the coefficients of $g(x)$, for the second column, we have x^2 times the coefficients of $g(x)$ etc. until the n^{th} column, where we have x^{n-1} times the coefficients of $f(x)$. The columns look like:

$$\begin{array}{c} \\ x^0 \\ x^1 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ x^{m+n-1} \end{array} \begin{array}{c} f(x)x^0 \quad f(x)x^1 \ldots f(x)x^{m-1} \quad g(x)x^0 \quad g(x)x^1 \ldots g(x)x^{n-1} \\ \left[\begin{array}{ccccccc} a_0 & 0 & \ldots & 0 & b_0 & 0 & \vdots \quad 0 \\ a_1 & a_0 & \ldots & 0 & b_1 & b_0 & \vdots \quad 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \quad \vdots \\ a_n & \vdots & \ldots & a_0 & b_m & \vdots & \vdots \quad b_0 \\ 0 & a_{n-1} & \ldots & a_1 & 0 & b_m & \vdots \quad \vdots \\ \vdots & a_n & \ldots & \vdots & \vdots & \vdots & \vdots \quad \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \quad b_{m-1} \\ 0 & 0 & \ldots & a_n & 0 & 0 & \vdots \quad b_m \end{array}\right] \end{array}$$

As long as we have the product formula for bases, it is possible to have the formula for the resultant matrices. Consider a polynomial $A(x) = \sum_{i=0}^{m} a_i x^i$ in the monomial basis. We

know that the product formula for the monomial basis is $x^k x^1 = x^{k+1}$ and the recursive relation is $x^k = x^1 x^{k-1}$. Clearly, we have $x^1\ A(X) = \sum_{i=0}^{m+1} a_i x^{i+1}$ and more generally we can compute the Sylvester matrix based on the formula:

$$x^j * A(x) = \sum_{i=0}^{n} a_i x^{i+j} \tag{4.28}$$

Consider the two polynomials $A(x) = a_3x^3 + a_2x^2 + a_1x + a_0, B(x) = b_3x^3 + b_2x^2 + b_1x + b_0$, based on the multiplication formula, we can compute the Sylvester matrix as:

$$\begin{array}{c} \\ x^0A(x) \\ x^1A(x) \\ x^2A(x) \\ x^0B(x) \\ x^1B(x) \\ x^2B(x) \end{array} \begin{array}{c} \begin{array}{cccccc} x^0 & x^1 & x^2 & x^3 & x^4 & x^5 \end{array} \\ \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & 0 & 0 \\ 0 & a_0 & a_1 & a_2 & a_3 & 0 \\ 0 & 0 & a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 & 0 & 0 \\ 0 & b_0 & b_1 & b_2 & b_3 & 0 \\ 0 & 0 & b_0 & b_1 & b_2 & b_3 \end{bmatrix} \end{array}$$

For non-monomial bases such as orthogonal bases (e.g. Chebyshev and Legendre bases), we can apply the same process as long as we have the multiplication formula; however, in general it is not always possible to have a simple product formula as for monomial basis polynomials.

4.3.3.1 Resultant Sylvester Matrix in Chebyshev Basis

This section is based on the work in [38]. The Sylvester matrix can be generalized for the Chebyshev basis using the multiplication formula in the Chebyshev basis. Let $f(z) = \sum_{i=0}^{d} f_i T_i(z)$ be a polynomial of degree *d* expressed in the Chebyshev basis, and let $T_j(z)$ be a Chebyshev polynomial. Then, the product $T_j(z)f(z)$ is given by

$$T_j(z)f(z) = \frac{1}{2}\sum_{i=0}^{d} f_i T_{i+j}(z) + \frac{1}{2}\sum_{i=j}^{d} f_i T_{i-j}(z) + \frac{1}{2}\sum_{i=0}^{j-1} f_i T_{j-i}(z). \tag{4.29}$$

The formula will be proven in the next section.

Assume that $f(z) = \sum_{k=0}^{m} a_k T_k(z)$, $g(z) = \sum_{k=0}^{n} b_k T_k(z)$, and we want to compute their Sylvester matrix in the Chebyshev basis, $CSyl(f, g)$: We have to compute:

$$T_0(z)f(z)$$

$$T_1(z)f(z)$$

$$\vdots$$

$$\vdots$$

$$T_{n-1}(z)f(z)$$

$$T_0(z)g(z)$$

$$T_1(z)g(z)$$

$$\vdots$$

$$\vdots$$

$$T_{m-1}(z)g(z)$$

By the previous formula (4.29):

$$T_i(z)f(z) = \frac{1}{2}\sum_{k=0}^{m} a_k T_{k+i}(z) + \frac{1}{2}\sum_{k=i}^{m} a_k T_{k-i}(z) + \frac{1}{2}\sum_{k=0}^{i-1} a_i T_{i-k}(z)$$

$$= \frac{1}{2}\left(B_{i,0}^{f}(z) + B_{i,1}^{f}(z) + B_{i,2}^{f}(z)\right),$$

where

$$B_{i,0}^{f}(z) = \sum_{k=0}^{m} a_k T_{k+i}(z)$$

$$B_{i,1}^{f}(z) = \sum_{k=i}^{m} a_k T_{k-i}(z),$$

and $B_{i,2}^{f}(z) = \sum_{k=0}^{i-1} a_i T_{i-k}(z).$

For the product, we have the three matrices:

$$S_0 = \begin{array}{c} \\ T_0 \\ T_1 \\ \vdots \\ \\ \\ T_{n+m-1} \end{array} \begin{array}{c} \begin{array}{cccccccc} B_{0,0}^{f} & B_{1,0}^{f} & \dots & B_{n-1,0}^{f} & B_{0,0}^{g} & B_{1,0}^{g} & \dots & B_{m-1,0}^{g} \end{array} \\ \begin{pmatrix} a_0 & & & & b_0 & & & \\ a_1 & a_0 & & & b_1 & b_0 & & \\ \vdots & \vdots & \vdots & & \vdots & \vdots & b_0 & \\ a_m & a_{m-1} & \vdots & a_0 & b_n & b_{n-1} & \vdots & b_0 \\ & a_m & a_{m-1} & \vdots & & b_n & b_{n-1} & \vdots \\ & & a_m & a_{m-1} & & & b_n & b_{n-1} \\ & & & a_m & & & & b_n \end{pmatrix} \end{array}$$

$$S_1 = \begin{array}{c} \\ T_0 \\ T_1 \\ \vdots \\ \\ \\ T_{n+m-1} \end{array} \begin{array}{c} \begin{array}{cccccccc} B_{0,1}^{f} & B_{1,1}^{f} & \dots & B_{n-1,1}^{f} & B_{0,1}^{g} & B_{1,1}^{g} & \dots & B_{m-1,1}^{g} \end{array} \\ \begin{pmatrix} a_0 & a_1 & \dots & a_{n-1} & b_0 & b_1 & \dots & b_n \\ a_1 & a_2 & \dots & \vdots & b_1 & \vdots & \dots & \\ \vdots & \vdots & \vdots & a_m & \vdots & \vdots & \vdots & \\ a_{m-1} & a_m & \vdots & & \vdots & b_n & \vdots & \\ a_m & & & & b_n & & & \\ 0 & & & & & & & \\ \vdots & & & & & & & \end{pmatrix} \end{array}$$

$$S_2 = \begin{pmatrix} 0 & 0 & & & & 0 & 0 & & 0 & 0 \\ 0 & a_0 & \dots & \dots & a_{n-1} & 0 & b_0 & \dots & b_m & 0 \\ \vdots & & & & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & & & & a_0 & 0 & & & b_0 & 0 \\ 0 & & & & 0 & 0 & & & 0 & 0 \end{pmatrix}$$

The resultant of $f(x)$ and $g(x)$ is the determinant of the sum of the matrices:

$$Res(f,g) = \frac{1}{2} Det(S_0 + S_1 + S_2)$$

Example 4.7 Let $f(x) = a_0T_0(x) + a_1T_1(x) + a_2T_2(x)$, and $g(x) = b_0T_0(x) + b_1T_1(x) + b_2T_2(x)$ be polynomials expressed in the Chebyshev basis. Suppose we wish to compute their Sylvester resultant matrix in the same basis to avoid basis conversion such that the power basis is not used. We have four columns containing $(T_0f(x), T_1f(x), T_0g(x), T_1g(x))$:

$$T_0f(x) = f(x) = \frac{1}{2}(a_0T_0 + a_1T_1 + a_2T_2) + \frac{1}{2}(a_0T_0 + a_1T_1 + a_2T_2)$$

$$T_1f(x) = \frac{1}{2}a_0T_1 + \frac{1}{2}a_0T_1 + \frac{1}{2}(a_1T_0 + a_1T_2) + \frac{1}{2}(a_2T_3 + a_2T_1)$$

We have the same multiplication for $g(x)$:

$$T_0g(x) = g(x) = \frac{1}{2}(b_0T_0 + b_1T_1 + b_2T_2) + \frac{1}{2}(b_0T_0 + b_1T_1 + b_2T_2)$$

$$T_1g(x) = \frac{1}{2}b_0T_1 + \frac{1}{2}b_0T_1 + \frac{1}{2}(b_1T_0 + b_1T_2) + \frac{1}{2}(b_2T_3 + b_2T_1)$$

The resultant is given by:

$$Res(f,g) = \left(\frac{1}{2}\right) \quad Det\left(\begin{pmatrix} a_0 & 0 & b_0 & 0 \\ a_1 & a_0 & b_1 & b_0 \\ a_2 & a_1 & b_2 & b_1 \\ 0 & a_2 & 0 & b_2 \end{pmatrix} + \begin{pmatrix} a_0 & a_1 & b_0 & b_1 \\ a_1 & a_2 & b_1 & b_2 \\ a_2 & 0 & b_2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & a_0 & 0 & b_0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}\right) =$$

$$= \left(\frac{1}{2}\right) Det\left(\begin{bmatrix} 2a_0 & a_1 & 2b_0 & b_1 \\ 2a_1 & 2a_0 + a_2 & 2b_1 & 2b_0 + b_2 \\ 2a_2 & a_1 & 2b_2 & b_1 \\ 0 & a_2 & 0 & b_2 \end{bmatrix}\right)$$

$$= Det\left(\begin{bmatrix} a_0 & \frac{1}{2}a_1 & b_0 & \frac{1}{2}b_1 \\ a_1 & a_0 + \frac{1}{2}a_2 & b_1 & b_0 + \frac{1}{2}b_2 \\ a_2 & \frac{1}{2}a_1 & b_2 & \frac{1}{2}b_1 \\ 0 & \frac{1}{2}a_2 & 0 & \frac{1}{2}b_2 \end{bmatrix}\right)$$

$$= Det\left(\begin{pmatrix} a_0 & a_1 & a_2 & 0 \\ \frac{a_1}{2} & a_0 + \frac{a_2}{2} & \frac{a_1}{2} & \frac{a_2}{2} \\ b_0 & b_1 & b_2 & 0 \\ \frac{b_1}{2} & b_0 + \frac{b_2}{2} & \frac{b_1}{2} & \frac{b_2}{2} \end{pmatrix}\right)$$

If we do the same example using the basis change method (where we convert the polynomials in the Chebyshev basis into polynomials in the power basis, compute the resultant Sylvester matrix, and then convert back to the Chebyshev basis) then we get the same answer.

We need these steps:

- Convert polynomials into polynomials in the power basis:

$$f(x) = [a_0\ a_1\ a_2] * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix} \begin{bmatrix} x^0 \\ x^1 \\ x^2 \end{bmatrix}$$

$$f(x) = (a_0 - a_2) + a_1 x + 2a_2 x^2$$

$$g(x) = [b_0\ b_1\ b_2] * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 2 \end{bmatrix} \begin{bmatrix} x^0 \\ x^1 \\ x^2 \end{bmatrix}$$

$$g(x) = (b_0 - b_2) + \ b_1 x + 2b_2 x^2$$

- Compute their Sylvester matrix resultant in the power basis:

$$Syl(f, g, x) = \begin{bmatrix} a_0 - a_2 & a_1 & 2a_2 & 0 \\ 0 & a_0 + a_2 & a_1 & 2a_2 \\ b_0 - b_2 & b_1 & 2b_2 & 0 \\ 0 & b_0 - b_2 & b_1 & 2b_2 \end{bmatrix}$$

- Convert the Sylvester matrix back to the Chebyshev basis:

$$\begin{bmatrix} a_0 - a_2 & a_1 & 2a_2 & 0 \\ 0 & a_0 + a_2 & a_1 & 2a_2 \\ b_0 - b_2 & b_1 & 2b_2 & 0 \\ 0 & b_0 - b_2 & b_1 & 2b_2 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & -\frac{3}{4} & 0 & \frac{1}{4} \end{bmatrix}$$

$$= \begin{bmatrix} a_0 & a_1 & a_2 & 0 \\ \frac{a_1}{2} & a_0 + \frac{1}{2}a_2 & \frac{1}{2}a_1 & \frac{1}{2}a_2 \\ b_0 & b_1 & b_2 & 0 \\ \frac{1}{2}b_1 & b_0 + \frac{1}{2}b_2 & \frac{1}{2}b_1 & \frac{1}{2}b_2 \end{bmatrix}$$

As you can see, we get the same result. The resultant matrix in the Chebyshev basis can be

computed in the following way. On the first matrix, we shift the first column down to get the second column. On the second matrix, we shift the first column up, but a_0 will go out of the matrix and it is pushed on the third matrix.

4.3.3.2 Resultant in Legendre Basis

In the case of the Legendre basis, we can apply the same process as with the Chebyshev basis to compute the Sylvester matrix resultant in the same basis as long as we have the multiplication formula. The Neumann-Adams formula gives the expansion of a product of two Legendre polynomials in a finite series of such polynomials. The multiplication formula for Legendre polynomials is [55]:

$$P_n(x)P_m(x) = \sum_{r=0}^{m} \frac{A_r A_{n-r} A_{m-r}}{A_{m+n-r}} \frac{2n+2m-4r+1}{2n+2m-2r+1} P_{n+m-2r}(x) \tag{4.31}$$

where $m \geq n$, $A_r = \frac{\left(\frac{1}{2}\right)_r}{r!}$ $(a)_r = a(a+1)(a+2)\ldots(a+r-1)$, and $(a)_0 = 1$.

For most orthogonal basis polynomials, a simple product formula is not always possible (like in the Chebyshev or monomial basis). We assume that we have a special element (z) that acts on orthogonal polynomials via a special multiplication rule. The special rule allows us to define a multiplication $P_n(x) * A(x)$ for any orthogonal polynomials in orthogonal bases. Table 4.4 summaries the special element (z) with corresponding coefficients for the Chebyshev and Legendre bases. This approach is used in [41, 43]. For monomial polynomials, the special multiplication rule is simply the standard multiplication by x:

$$x * x^i = x^{i+1}$$
$$x^j * x^i = x^{i+j} \tag{4.32}$$

Now assume $f(x) = \sum_{i=0}^{n} a_i x^i$ be a polynomial of degree d expressed in the monomial

basis and let x^j be a power basis polynomial, then the product $x^j\ f(x)$ is given by:

$$x^j * f(x) = \sum_{i=0}^{n} a_i x^i * x^j$$

$$= \sum_{i=0}^{n} a_i x^{i+j} \qquad 4.33$$

For Chebyshev polynomials $T_i(x)$, the special element is $z = 2x$ given by the rule:

$$z.T_i(x) = T_{i+1} + T_{i-1}(x)$$

From the recursive definition of Chebyshev polynomials, we have:

$$T_{i+1} = 2xT_i(x) - T_{i-1}(x)$$

$$T_1(x) = x \Rightarrow$$

$$T_1(x) * T_i(x) = \frac{1}{2}(T_{i+1}(x) + T_{i-1}(x))$$

We then get the multiplication formula:

$$T_j(x)T_i(x) = \tfrac{1}{2}\left(T_{i+j}(x) + T_{i-j}(x)\right) \qquad 4.34$$

Now assume $f(z) = \sum_{i=0}^{n} a_i T_i$ be a polynomial of degree d expressed in the Chebyshev basis and let $T_j(z)$ be a Chebyshev polynomial, then the product $Tj(z)\ f(z)$ can be expressed in the Chebyshev basis using $O(d)$ arithmetic operations:

$$f(z) * T_j(z) = \sum_{i=0}^{n} a_i T_i * T_j(z)$$

$$= \sum_{i=0}^{n} a_i \left(\frac{1}{2} \left(T_{i+j}(z) + T_{i-j}(z) \right) \right)$$

$$= \frac{1}{2} \sum_{i=0}^{n} a_i T_{i+j} + \frac{1}{2} \sum_{i=0}^{n} a_i T_{i-j}(z)$$

and we decompose it in the following way:

$$\frac{1}{2} \sum_{i=0}^{n} a_i T_{i+j} + \frac{1}{2} \sum_{i=0}^{j-1} a_i T_{j-i}(z) + \frac{1}{2} \sum_{i=j}^{n} a_i T_{i-j}(z)$$

which is the formula in (4.29).

Table 4.4 : Orthogonal Polynomial Bases with Special Element *z* [41]

Basis	Special element (z)	$c_{i,i+1}$	$c_{i,i-1}$
Standard	*X*	0	1
Chebyshev	*2x*	1	1
Legendre	*X*	$\frac{i+1}{2i+3}$	$\frac{i}{2i-1}$

In the case of Legendre polynomials, it is observed that the formula is very complicated. To make it simple, we assume we have a special element *z* that acts on Legendre polynomials via a special multiplication rule. As we mentioned before, the basic recursive formula for Legendre polynomials is:

$$(n+1)P_{n+1}(x) = (2n+1)x\, P_n(x) - nP_{n-1}(x)$$

$$P_0(x) = 1$$

$$P_1(x) = x$$

We can obtain a simple product formula based on the special element z = x:

$$x * P_n(x) = \frac{(n+1)}{(2n+1)} P_{n+1}(x) + \frac{n}{(2n+1)} P_{n-1}(x)$$

$$P_1(x) = x \Rightarrow$$

$$P_1(x) * P_n(x) = \frac{(n+1)}{(2n+1)} P_{n+1}(x) + \frac{n}{(2n+1)} P_{n-1}(x) \qquad 4.35$$

Let $f(x) = \sum_{i=0}^{d} a_i P_i(x)$ be a polynomial of degree *d* expressed in the Legendre basis and let $P_1(x)$ be a Legendre polynomial. Then the product $P_1(x) f(x)$ is:

$$P_1(x) * f(x) = \sum_{i=0}^{n} a_i P_1(x) P_i(x)$$

$$= a_0 P_1(x) + \sum_{i=1}^{n} a_i \left(\frac{i+1}{2i+1} P_{i+1}(x) + \frac{i}{2i+1} P_{i-1}(x) \right)$$

$$a_0 P_1(x) + \sum_{i=1}^{n} a_i \frac{i+1}{2i+1} P_{i+1}(x) + \sum_{i=1}^{n} a_i \frac{i}{2i+1} P_{i-1}(x)$$

$$\sum_{i=0}^{n} a_i \frac{i+1}{2i+1} P_{i+1}(x) + \sum_{i=1}^{n} a_i \frac{i}{2i+1} P_{i-1}(x)$$

$$\sum_{i=1}^{n+1} a_{i-1}\frac{(i-1)+1}{2(i-1)+1}P_{i+1-1}(x) + \sum_{i=0}^{n-1} a_{i+1}\frac{i+1}{2(i+1)+1}P_{i-1+1}(x)$$

$$= \sum_{i=1}^{n+1} a_{i-1}\ \frac{i}{2i-1}P_i(x) + \sum_{i=o}^{n-1} a_{i+1}\ \frac{i+1}{2i+3}\ P_i(x)$$

More generally,

$$P_1(x)f(x) \equiv \textstyle\sum_{n=0}^{\infty} b_n P_n(x), \qquad 4.36$$

where

$$b_n = \tfrac{n}{2n-1}a_{n-1} + \tfrac{n+1}{2n+3}a_{n+1}, \qquad n \geq 1 \qquad [b_0 = 0].$$

The Sylvester matrix can be generalized for the Legendre basis based on the multiplication formula in (4.31). Let $f(x) = \sum_{i=0}^{n} a_i P_i(x)$ be a polynomial of degree *n* expressed in the Legendre basis and let $P_i(x)$ be a Legendre polynomial. Then, the product $P_m(x)f(x)$ is:

$$P_m(x)f(x) = P_m(x)\sum_{i=0}^{n} a_i P_i(x)$$

$$= \sum_{i=0}^{n} a_i P_i(x)P_m(x)$$

$$= \sum_{i=0}^{n} a_i \left(\sum_{r=0}^{m} \frac{A_r A_{i-r} A_{m-r}}{A_{m+i-r}} \cdot \frac{2i+2m-4r+1}{2i+2m-2r+1} P_{i+m-2r}(x) \right)$$

$$= \sum_{i=0}^{n} a_i \left(\sum_{r=0}^{m} A_{m,i}^{r}\, P_{i+m-2r}(x) \right)$$

$$= \textstyle\sum_{i=0}^{m} a_i \left(\sum_{r=0}^{i} A_{m,i}^{r}\, P_{i+m-2r}(x) \right) + \sum_{i=m+1}^{n} a_i \left(\sum_{r=0}^{n} A_{m,i}^{r}\, P_{i+m-2r}(x) \right) \qquad 4.37$$

where, for convenience,

$$A_{m,i}^{r} = \frac{A_r A_{i-r} A_{m-r}}{A_{m+i-r}} \left(\frac{2i+2m-4r+1}{2i+2m-2r+1}\right)$$

and

$$A_m = \frac{1.3.5 \ldots (2m-1)}{m!}$$

where $m, i \in N; n \geq m > 1$.

Even though the formula seems complicated, it is enough for the algorithmic process. The point is that once we have the product formula in (4. 31), it is possible to have almost all the classical formulations of matrices for resultants. Consider two polynomials of degree *n* and *m* respectively, expressed in the Legendre basis: $f(x) = \sum_{i=0}^{n} a_i P_i \ and \ g(x) = \sum_{j=0}^{m} b_j P_j$. To compute their Sylvester matrix, we have the first *m* columns containing $f(x)'s$ coefficients multiplied by $(P_o, P_1, \ldots \ldots P_{m-1})$ respectively, and the last *n* columns containing $g(x)'s$ coefficients multiplied by $(P_o, P_1, \ldots \ldots P_{n-1})$, respectively:

$$f(x)P_0(x) \quad f(x)P_1(x) \quad \ldots. \quad f(x)P_{m-1}(x) \quad g(x)P_0(x) \quad g(x)P_1(x) \quad \ldots \quad g(x)P_{n-1}(x)$$

$$\begin{array}{c} P_0 \\ P_1 \\ \vdots \\ P_m \\ \vdots \\ P_n \\ P_{n+1} \\ \vdots \\ \vdots \\ P_{n+m-1} \end{array} \begin{bmatrix} a_0 & \frac{a_1}{3} & \ldots & \gamma_{m-1,0} & b_0 & \frac{b_1}{3} & \ldots & \delta_{n-1,0} \\ a_1 & a_0 + \frac{2}{5}a_2 & \ldots & \gamma_{m-1,1} & b_1 & b_0 + \frac{2}{5}b_2 & \ldots & \delta_{n-1,1} \\ : & : & \ldots & : & : & : & \ldots & : \\ a_m & \frac{m}{2m-1}a_m + \frac{m+1}{2m+3}a_{m+1} & \ldots & \gamma_{m-1,m} & b_m & b_{m-1}\frac{m}{2m+1} & \ldots & \delta_{n-1,m} \\ : & : & \ldots & : & : & : & \ldots & : \\ a_n & a_{n-1}\frac{n}{2n+1} & \ldots & \gamma_{m-1,n} & 0 & 0 & \ldots & \delta_{n-1,n} \\ 0 & a_n\frac{n+1}{2n-1} & \ldots & \gamma_{m-1,n+1} & 0 & 0 & \ldots & \delta_{n-1,n+1} \\ : & : & \ldots & : & : & : & \ldots & : \\ 0 & 0 & \ldots & \gamma_{m-1,m+n-1} & 0 & 0 & \ldots & \delta_{n-1,m+n-1} \end{bmatrix}$$

Example 4.7 Let $f(x) = a_0 + a_1P_1(x) + a_2P_2(x)$ and $g(x) = b_0 + b_1P_1(x) + b_2P_2(x)$ be two polynomial expressed in the Legendre basis. We need to compute their Sylvester resultant matrix in the Legendre basis, $Res(f,g) = \text{Det}(LSyl(f,g))$, such that the power basis is not used. We have four columns: $(P_0f(x),\ P_1f(x),\ P_0g(x),\ P_1g(x))$. Then:

$$f(x)P_0(x) = f(x)$$

$$f(x)P_1(x) = a_0P_1(x) + \frac{2}{3}a_1P_2(x) + \frac{3}{5}a_2P_3(x) + \frac{1}{3}a_1P_0(x) + \frac{2}{5}a_2P_1(x)$$

$$= \frac{1}{3}a_1P_0(x) + \left(a_0 + \frac{2}{5}a_2\right)P_1(x) + \frac{2}{3}a_1P_2(x) + \frac{3}{5}a_2P_3(x)$$

The same for $g(x)$:

$$g(x)P_0(x) = g(x)$$

$$g(x)P_1(x) = b_0P_1(x) + \frac{2}{3}b_1P_2(x) + \frac{3}{5}b_2P_3(x) + \frac{1}{3}b_1P_0(x) + \frac{2}{5}b_2P_1(x)$$

$$= \frac{1}{3}b_1P_0(x) + \left(b_0 + \frac{2}{5}b_2\right)P_1(x) + \frac{2}{3}b_1P_2(x) + \frac{3}{5}b_2P_3(x)$$

The resultant is the determinant of the matrix:

$$LSylv(f,g) = \begin{bmatrix} a_0 & a_1 & a_2 & 0 \\ \frac{1}{3}a_1 & a_0 + \frac{2}{5}a_2 & \frac{2}{3}a_1 & \frac{3}{5}a_2 \\ b_0 & b_1 & b_2 & 0 \\ \frac{1}{3}b_1 & b_0 + \frac{2}{5}b_2 & \frac{2}{3}b_1 & \frac{3}{5}b_2 \end{bmatrix} \quad 4.38$$

If we do the same example using the basis change method (convert polynomials in Legendre basis into polynomials in the power basis, compute the resultant Sylvester matrix, and then convert back to the Legendre basis) then we get the same answer.

We need these steps:

- Convert polynomials into polynomials in the power basis:

$$f(x) = \left(a_0 - \frac{1}{2}a_2\right) + a_1 x + \frac{3}{2}a_2 x^2$$

and

$$g(x) = \left(b_0 - \tfrac{1}{2}b_2\right) + b_1 x + \tfrac{3}{2}b_2 x^2$$

- Compute their Sylvester matrix resultant in the power basis:

$$\begin{bmatrix} a_0 - \frac{1}{2}a_2 & a_1 & \frac{3}{2}a_2 & 0 \\ 0 & a_0 - \frac{1}{2}a_2 & a_1 & \frac{3}{2}a_2 \\ b_0 - \frac{1}{2}b_2 & b_1 & \frac{3}{2}b_2 & 0 \\ 0 & b_0 - \frac{1}{2}b_2 & b_1 & \frac{3}{2}b_2 \end{bmatrix}$$

- Convert the Sylvester matrix back to the Legendre basis:

$$\begin{bmatrix} a_0 - \frac{1}{2}a_2 & a_1 & \frac{3}{2}a_2 & 0 \\ 0 & a_0 - \frac{1}{2}a_2 & a_1 & \frac{3}{2}a_2 \\ b_0 - \frac{1}{2}b_2 & b_1 & \frac{3}{2}b_2 & 0 \\ 0 & b_0 - \frac{1}{2}b_2 & b_1 & \frac{3}{2}b_2 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 \\ 0 & \frac{3}{5} & 0 & \frac{2}{5} \end{bmatrix}$$

$$\begin{bmatrix} a_0 & a_1 & a_2 & 0 \\ \frac{1}{3}a_1 & a_0 + \frac{2}{5}a_2 & \frac{2}{3}a_1 & \frac{3}{5}a_2 \\ b_0 & b_1 & b_2 & 0 \\ \frac{1}{3}b_1 & b_0 + \frac{2}{5}b_2 & \frac{2}{3}b_1 & \frac{3}{5}b_2 \end{bmatrix}$$

As you can see, we get the same result in (4.38). To sum up, for bases other than monomial bases, it is possible to follow the same scheme to find Sylvester matrix as soon the basis is graduated by the degree. This is the case for classical orthogonal polynomials since they are given by a recurrence. In the monomial case, the product formula is $x^i x^k = x^{i+k}$, and in a classical orthogonal polynomials basis, one needs a product formula expressing $P_i(x)P_k(x)$. In the case of the Chebyshev polynomials of first kind the product formula is simple: $T_i\, T_j = T_{i+j} + T_{|i-j|}$, and leads to a simple structure of the Sylvester matrix. However, for general orthogonal polynomials basis, it is not always possible to have a simple product formula like for monomials or Chebyshev polynomials. For instance, the product formula for Legendre basis is complicated. Even though the formula seems to be very complicated, it is possible to have almost all the classical formulations of matrices for resultant.

4.4 Summary

This chapter introduces the basic ideas and theories of resultants. It introduces a review of the most common resultant matrices for monomial basis polynomials. Moreover, the conversion between power basis polynomial representations and orthogonal polynomial representations is considered. Furthermore, resultant matrices for orthogonal basis polynomials are investigated and the Sylvester matrix of Legendre basis polynomials is computed. It is noted that resultant matrices are widely applied in computer-aided geometric design (CAGD). As stated earlier, resultants are used for implicitization and bivariate-polynomial root finding. An important problem in CAGD is to compute the common zeros of two bivariate polynomial equations, which can be solved using resultant matrices. This problem is discussed in the next chapter.

Chapter 5

Resultant-Based Methods for Critical Points of Plane Curves Problems

In Chapter 4, we presented the resultant tool for determining whether or not two polynomials have a common root. In this chapter, we apply this tool to find the critical points of parametric curves. In order to find the critical points of parametric curves, we first convert the parametric equation of a curve into an implicit equation of the form $f(x, y) = 0$. This operation can be obtained by resultants [15]. Then, we can find the critical points by finding the roots of the bivariate polynomial system : $f = f_x = f_y = 0$. The common zeros of polynomials can be also computed by resultants [24, 27].

5.1 Computing the Common Zeros of a Two Bivariate Polynomial Equations System Via Resultants

Finding the solutions of a system of nonlinear polynomial equations is a classical and fundamental problem in the computational literature including computer algebra, robotics, computer graphics, geometry, and computer vision [51]. For example, in geometric modeling, the solutions of a system of polynomial equations are used to find the intersection of parametric and algebric curves. Moreover, this problem arises for example when detecting the critical points of a curve $C(x,\ y) = 0$, since the critical points satisfy $f(x, y) = \frac{\partial C}{\partial x} = 0$, $g(x, y) = \frac{\delta C}{\delta y} = 0$. Solutions of a system of partial derivatives of polynomial equations are used to find the critical points of parametric and algebric curves .There are many existing

approaches for computing the roots of a system of polynomial equations based on resultants [23, 24, 27, 50]. In this section, we discuss a method for computing the roots of a polynomial equations system based on the resultants of a system of polynomial equations. The method described in this chapter is based on the hidden variable resultant method for polynomials expressed in the power basis [27, 50].

5.2 Hidden Variable Resultant Method

The hidden variable resultant method is based on selecting one variable, say y, and rewriting bivariate polynomials F and G of degrees m and n as polynomials in x with coefficients that are polynomials in y [50, 51, 53]. Suppose we have a system of two polynomial equations $F(x, y)$ and $G(x, y)$:

$$F(x,y) = \sum_{i=0,j=0}^{m} a_i b_j x^i y^j , \qquad G(x,y) = \sum_{i=0,j=0}^{n} a_i b_j x^i y^j \tag{5.1}$$

By selecting the variable y and rewriting bivariate polynomials fx and fy of degrees m and n as polynomials in x with coefficients in y we get:

$$F(x,y) = F_y(x) = \sum_{i=0}^{m} a_i(y) x^i , \qquad G(x,y) = G_y(x) = \sum_{i=0}^{n} b_i(y) x^i \tag{5.2}$$

It is well known that two univariate polynomials have a common root if and only if a resultant is zero [59]. In particular, the two polynomials $Fy(x)$ and $Gy(x)$ in (5.2), thought of as univariate in x, have a common zero if and only if a resultant matrix is singular. Therefore, the y-values of the solutions to $F = G = 0$ can be computed by finding the y-values such that a resultant matrix is singular [50, 51].

$$Res\big(F_y(x),\ G_y(x), x\big) = \sum_{i=0}^{m} a_i y^i = 0 \qquad 5.3$$

Once we have found the y-values of the solutions we then fnd the x −values by a univariate root-finding algorithm based on computing the eigenvalues of the companion matrix.

$$C = \begin{bmatrix} 0 & 0 & \ldots & 0 & -a_0/a_i \\ 1 & 0 & \ldots & 0 & -a_1/a_i \\ 0 & 1 & \cdots & 0 & -a_2/a_i \\ \ldots & \ldots & \cdots & \ldots & \ldots \\ 0 & 0 & \cdots & 1 & -a_{i-1}/a_i \end{bmatrix}$$

5.3 Resultant Methods with Sylvester Resultants

In the case of two equations in two variables, the Sylvester resultant, described in a previous chapter, can be used to eliminate one variable. Then, the system can be solved for the other variable. Once the roots for one variable are known, those values can be substituted into either polynomial to solve for the other variable. Although our focus is on Sylvester resultants, other resultants can also be applied to solve polynomial systems such as Cayley's formulation [23], the Macaulay resultant [24], or the Bezout resultant matrix [27, 50]. Example 5.1 illustrates which variable the resultant method should hide.

Example 5.1 Consider a system of two polynomials:

$$\begin{cases} f_1(x,y) = x^2 + 6x + 3y - 4 \\ f_2(x,y) = 2x^2 + 3y^2 - 7x + 3y + 5 \end{cases}$$

If we treat them as polynomials in x, then the Sylvester matrix is a 4 by 4 square matrix, and the size of eigenvalues problem will be a 4 by 4 companion matrix.

$$\begin{cases} f_1(x) = x^2 + 6x + (3y - 4) \\ f_2(x) = 2x^2 - 7x + (3y^2 + 3y + 5) \end{cases}$$

$$Syl_x(f_1(x), f_2(x)) = \begin{bmatrix} 1 & 6 & 3y-4 & 0 \\ 0 & 1 & 6 & 3y-4 \\ 2 & -7 & 3y^2+3y+5 & 0 \\ 0 & 2 & -7 & 3y^2+3y+5 \end{bmatrix}$$

$$Res(f_1(x), f_2(x), x) = 9y^4 - 18y^3 + 429y^2 + 663y + 207 = 0$$

$$C = \begin{bmatrix} 2 & -47.6667 & -73.6667 & -23 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

If we treat them as polynomials in y, then the Sylvester matrix is a 3 by 3 square matrix, and the size of eigenvalues problem will be a 4 by 4 companion matrix.

5.4 Finding Critical Points of Parametric Curves Based on Resultants in the Monomial Basis

Consider the parametric curve : $x(t) = a_0 + a_1 t + a_2 t^2 + \cdots a_{n-1} t^{n-1}$ and $y(t) = b_0 + b_1 t + b_2 t^2 + \cdots b_{m-1} t^{m-1}$. To find the critical points we compute $x'(t) = a_1 + 2a_2 t + 3a_3 t^2 + \cdots (n-1) a_{n-1} t^{n-2}$ and $y'(t) = b_1 + 2b_2 t + 3b_3 t^2 + \cdots + (m-1) b_{m-1} t^{m-2}$. Then, we compute the roots of $x'(t) = 0$ and $y'(t) = 0$. For finding the roots of a polynomial, the companion matrix can be used. The roots of a polynomial are the eigenvalues

of the companion matrix. The default root solver in Matlab, for example, via the "roots" command, is a companion-matrix method.

Example 5.2 Consider the parametric curve $x(t) = t^3 - 3t^2$ and $y(t) = \frac{1}{3}t^3 - 4t$. Then,

$$x'(t) = 3t^2 - 6t \; and \; y'(t) = t^2 - 4.$$

The roots of derivatives are the eigenvalues of the companion matrices:

$$C\big(x'(t)\big) = \begin{bmatrix} 2 & 0 \\ 1 & 0 \end{bmatrix} \quad , \quad eig(C(x'(t))) = 0, 2$$

$$C\big(y'(t)\big) = \begin{bmatrix} 0 & 4 \\ 1 & 0 \end{bmatrix} \quad , \quad eig(C\big(y'(t)\big) = 2, -2$$

Every parametric curve can be implicitized into an algebraic curve of the form $f(x, y)$, where $f(x, y)$ is a bivariate polynomial. There are several methods for implicitization (finding an implicit representation of an algebraic curve given by its parametric equations): the classical implicitization using Groebner bases, the implicitization using resultants, polynomial interpolation, moving curves and surfaces, and the direct implicitization method. All of these methods are used for finding the implicit equation of curves, which are common objects in geometric modeling. The problem of computing implicit representations from parametric representations based on resultants is well studied in the literature [15, 21-23]. The resultant method is efficient and known to be better than the other methods. It can be applied in numerical computations, while the other three methods cannot be used to do this task.

5.5 Implicitization

There are two standard ways of representing algebraic curves: the implicit representation and the parametric representation. The choice of either an implicit or parametric representation of an algebraic curve depends on the operations to be performed on the curve. The parametric representation is appropriate for generating points of the curve and for plotting it with the computer. The implicit representation is convenient for checking whether a given point lies on the curve. That's why the transition from one representation to the other is important. Some operations can be more natural on implicit curve models. Implicitization is the process of conversion from the parametric form of a curve to its implicit form. This section discusses how to use resultants to obtain the implicit representation from the parametric representation [61]. Two implicitization methods, by direct substitution and by resultant, are discussed in this section.

Direct substitution can be used to convert some curves expressed parametrically to their implicit form. For example, given a curve represented by two parametric equations

$$x(t) = t + 4 \quad \text{and} \quad y(t) = t^2 + 2t - 3,$$

we can solve *t* in term of *x* to obtain $t = x - 4$, and substituting into $y = t^2 + 2t - 3$ gives its implicit equation $x^2 - 6x - y + 5 = 0$.

This method is suitable for the implicit forms of linear and quadratic curves. However, it can not be applied to curves of higher degree [52]. A more general approach is to use the resultant of two polynomials [57, 60, 61]. Implicit representation of curves can obtained directly from the parametric representation as $Res(X - x(t),\ Y - y(t),\ t)$. Consider a curve defined by two parametric equations:

$$x(t) = a_m t^m + a_{m-1} t^{m-1} + \cdots \ldots . a_1 t + a_0$$

$$y(t) = b_n t^n + b_{(n-1)} t^{n-1} + \cdots . . + b_1 t + b_0$$

To implicitize this curve, two auxiliary polynomials need to be created:

$$x(t) = a_m t^m + a_{m-1} t^{m-1} + \cdots \ldots . a_1 t + (a_0 - x)$$

$$y(t) = b_n t^n + b_{(n-1)} t^{n-1} + \cdots . . + b_1 t + (b_0 - y)$$

If the point (x, y) lies on the curve, the polynomials $x(t)$ and $y(t)$ have at least one common root. Det $(Syl(\mathrm{x(t)}, \mathrm{y(t)}, \mathrm{t})) = 0$ and Det $(Syl\big(x(t), y(t)\big))$ is the resultant of $x(t)$, and $y(t)$ in t. Det($Syl(\mathrm{x(t)}, \mathrm{y(t)}, \mathrm{t})$) is a function of x and y, and thus it is the implicit equation for the curve. We give an example to illustrate the implicitization of a curve.

Example 5.3. Consider a curve defined by two parametric equations $x(t) = 2t^2 + t + 3$ and $y(t) = t^2 + 3t + 1$. Creat two polynomials: $x(t) = 2t^2 + t + (3 - x)$ and $y(t) = t^2 + 3t + (1 - y)$. The Sylvester resultant matrix of $x(t), y(t)$ is:

$$Syl(x(t), y(t), t) = \begin{bmatrix} 2 & 1 & 3-x & 0 \\ 0 & 2 & 1 & 3-x \\ 1 & 3 & 1-y & 0 \\ 0 & 1 & 3 & 1-y \end{bmatrix}$$

The resultant of $x(t)$, and $y(t)$ is:

$$\mathrm{Det}(\, Syl(x(t), y(t), t)) = x^2 - 4xy + 4y^2 - 17x + 9y + 41$$

the implicit form of the curve is:

$$x^2 + 4y^2 - 4xy - 17x + 9y + 41 = 0$$

5.6 Finding Critical Points of Implicit Curves Based on Resultants in the Monomial Basis

Critical points of implicit and algebraic curves $C = f(x, y)$ can be obtained by finding the roots of the system of bivariate polynomial equations $f = f_x = f_y = 0$, where f_x and f_y are the *x* and *y* partial derivatives of f, respectively. The roots of a polynomial system can be computed based on the resultant of the system of polynomial equations. In the literature, there are many methods for bivariate polynomial root finding based on resultants [22, 23, 24, 27]. The method used in this thesis is based on the hidden variable resultant method for polynomials as described in previous section [50]. Figure 5.1 shows the approach to finding the critical points of an implicit curve in the monomial basis.

Example 5.4 Consider the curve $x^3 + x^2y - y^2 - 4y$. In order to find the critical points, we need to compute :

$$\frac{df}{dx} = 3x^2 + 2xy$$

$$\frac{df}{dy} = x^2 - 2y - 4$$

$$Syl\left(\frac{df}{dx}, \frac{df}{dy}, x\right) = \begin{bmatrix} 3 & 2y & 0 & 0 \\ 0 & 3 & 2y & 0 \\ 1 & 0 & -2y-4 & 0 \\ 0 & 1 & 0 & -2y-4 \end{bmatrix}$$

$$Det\left(Syl\left(\frac{df}{dx}, \frac{df}{dy}, x\right)\right) = -8y^3 + 20y^2 + 144y + 144$$

$$C\left(Det\left(Syl\left(\frac{df}{dx},\frac{df}{dy},x\right)\right)\right)=\begin{bmatrix}2.5 & 18 & 18\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}$$

$$eig\left(C\left(Det\left(Syl\left(\frac{df}{dx},\frac{df}{dy},x\right)\right)\right)\right)=\begin{bmatrix}6\\ -2\\ -1.5\end{bmatrix}$$

Table 5.1 Maple Code for Example 5.3

```
f := x^3+x^2*y-y^2-4*y;

with(LinearAlgebra);

diff(f, x);

diff(f, y);

SylvesterMatrix(diff(f, x), diff(f, y), x);

Determinant(SylvesterMatrix(diff(f, x), diff(f, y), x));
CompanionMatrix(Determinant(SylvesterMatrix(diff(f, x), diff(f,
y), x)));
Eigenvalues(CompanionMatrix(Determinant(SylvesterMatrix(diff(f,
x), diff(f, y), x))))
```

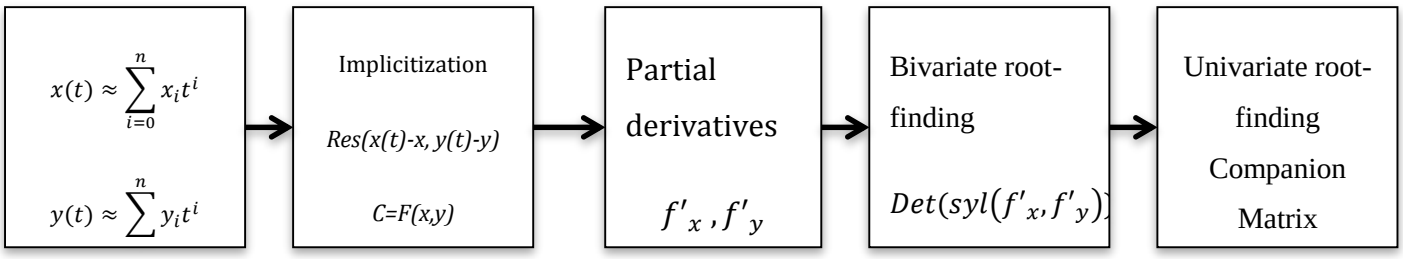


Figure 5.1: Flowchart of our approach for computing critical points of implicit curves in the power basis. Finding roots of bivariate derivative polynomials by computing the determinant of the Sylvester matrix. After that, we find the roots of univariate polynomials by computing the eigenvalues of the companion matrix.

The analysis of the approach to computing singular points presented in this section is restricted to the monomial basis, but it is well known that the conversion to the power basis is ill conditioned. One needs orthogonal basis equivalents of the Sylvester matrix and companion matrix resultants. In the case that the curve $f(x,y) = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{ij}P_i(x)P_j(y)$ is expressed in an orthogonal basis, we can follow this general algorithm to find the critical points:

1. Compute the partial derivative series $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ in orthogonal bases:

$$f_x(x,y) = \sum_{j=0}^{q}\sum_{i=0}^{r} a_{ij}P_i(x)P_j(y), \quad f_y(x,y) = \sum_{j=0}^{q}\sum_{i=0}^{r} b_{ij}P_i(x)P_j(y),$$

2. Rewrite the polynomials $f_x(x,y)\ and\ f_y(x,y)$ as univariate polynomials in $P_i(x)$ with coefficients that depend on $P_j(y)$ to find roots based on the hidden variable method:

$$f_x(x,y) = f_x(x) = \sum_{i=0}^{r} \alpha_j(y)P_i(x), \quad f_y(x,y) = f_y(x) = \sum_{i=0}^{r} \beta_j(y)P_i(x)$$

where $\alpha_j(y),\ \beta_j(y)$ are polynomials in y :

$$\alpha_j(y) = \sum_{j=0}^{q} a_j P_j(y) \qquad \beta_j(y) = \sum_{j=0}^{q} b_j P_j(y)$$

3. Form a Sylvester matrix for the orthogonal series:

$$Syl(f_x(x),\ f_y(x),\ x)$$

4. Compute the determinant of the Sylvester matrix:

$$Res(f_x(x),\ f_y(x)) = 0 = \sum_{i=0}^{n} a_i P_i(y) = 0$$

5. Form the companion matrix for the orthogonal series.

6. Compute the eigenvalues of the orthogonal companion matrix.

We need to compute derivatives, roots, and resultants in orthognal bases to avoid ill-conditioned conversion. In the next chapter, we compute the critical points of curves expressed in Legendre and Chebyshev bases. We compute the first derivative of Legendre series. Then, we compute the roots of Legendre and Chebyshev series based on the eignvalues of companion matrices expressed in orthogonal bases.

Chapter 6

Computing Critical Points of Orthogonal Truncated Series: Chebyshev and Legendre Truncated series

Finding the maxima, minima, and inflection points of a truncated Chebyshev or Legendre series is also a problem of finding the zeros of a polynomial when written in truncated orthogonal form. In fact, computing maxima, minima, and inflection points can be computed by first finding first derivatives in an orthogonal basis, and then computing the roots of the derivative series. The Chebyshev or Legendre coefficients of the derivatives of a function $f\ (x)$ can be computed by trivial recurrences from those of the function itself.

6.1 Computing Coefficients of a General-Order Derivative of an Orthogonal Series

Suppose we are given a function $f(x)$ which is infinitely differentiable in the closed interval $[-1,1]$. Then, we can write

$$f(x) = \sum_{n=0}^{\infty} a_n P_n(x)$$

and for the q^{th} derivative of $f(x)$ we have

$$f^q(x) = \sum_{n=0}^{\infty} a_n^q P_n(x). \qquad 6.1$$

so that the superscript "q" denotes the coefficients of the q^{th} derivative. Phillips in [65]

proved that the Legendre coefficients, a_n^q, of the qth derivative of $f(x)$ are related to the Legendre coefficients, a_n, of $f(x)$ by:

$$a_n^q = \left(\frac{(2n+1)}{2^{q-2}(q-1)!}\right)\sum_{i=1}^{\infty}\frac{(i+q-2)!\,(2n+2i+2q-3)!\,(n+i)!}{(i-1)!\,(2n+2i)!\,(n+i+q-2)!}a_{n+2i+q-2}$$

Now, we compute first derivative from this formula:

$$a_n^1 = \left(\frac{(2n+1)}{2^{1-2}(1-1)!}\right)\sum_{i=1}^{\infty}\frac{(i+1-2)!\,(2n+2i+2\times 1-3)!\,(n+i)!}{(i-1)!\,(2n+2i)!\,(n+i+1-2)!}a_{n+2i+1-2}$$

$$a_n^1 = \left(\frac{(2n+1)}{2^{-1}(0)!}\right)\sum_{i=1}^{\infty}\frac{(i-1)!\,(2n+2i-1)!\,(n+i)!}{(i-1)!\,(2n+2i)!\,(n+i-1)!}a_{n+2i-1}$$

$$a_n^1 = \left(\frac{(2n+1)}{2^{-1}}\right)\sum_{i=1}^{\infty}\frac{(i-1)!\,(2n+2i-1)!\,(n+i)(n+i-1)!}{(i-1)!\,(2n+2i)(2n+2i-1)!\,(n+i-1)!}a_{n+2i-1}$$

$$a_n^1 = 2(2n+1)\sum_{i=1}^{\infty}\frac{(n+i)}{(2n+2i)}a_{n+2i-1}$$

$$a_n^1 = 2(2n+1)\sum_{i=1}^{\infty}\frac{(n+i)}{2(n+i)}a_{n+2i-1}$$

$$a_n^1 = 2(2n+1)\sum_{i=1}^{\infty}\left(\frac{1}{2}\right)a_{n+2i-1}$$

$$a_n^1 = (2n+1)\sum_{i=1}^{\infty}2\left(\frac{1}{2}\right)a_{n+2i-1}$$

$$a_n^1 = (2n+1)\sum_{i=1}^{\infty}a_{n+2i-1}$$

$$a_n^1 = (2n+1)\sum_{p=n+1,p+n\ odd}^{\infty}a_p \qquad 6.2$$

If $f(x)$ is a polynomial of degree less than or equal to N, then the expansion coefficents of its first derivative can be computed only with $O(N)$ opreations.

Table 6.1 Maple Code for the Legendre Polynomial First Derivative

```
LegendreRep := proc (coeffs, x) local i, lRep;

    lRep := 0;

  for i from 1 to nops(coeffs) do lRep :=lRep+coeffs[i]*P[i-
1](x) od;

lRep

end:

ll := LegendreRep([7, 23, 55, 0, 12, 0, 0, 405, 0, 598, 0, 0,
0, 444, 0, 0, 0, 663], xi);

LegendreCoeff := proc (lRep, x, i)

   coeff(lRep, P[i](x), 1)

end:

LegendreCoeff(ll, xi, 2);

dll := diff(ll, xi);

LegendreDegree := proc (e)

   local lRep, i, d;

   lRep := e;

    d := -infinity;

    for i from 0 while has(lRep, P) do lRep := eval(subs(P[i]
= (x->0), lRep)); d := i od;

    d

end:

coeffOfDiff := proc (lRep, x, n)

local i, s, d;

 d := LegendreDegree(lRep);

s := 0;
```

```
 for i from 1 to d-n by 2 do
s := s+LegendreCoeff(lRep, x, n+i)
od;
 (2*n+1)*s
end:
for i from 0 to LegendreDegree(ll) do
       coeffOfDiff(ll, xi, i)
od;
```

6.2 Computing Roots of Orthogonal Series

Frobenius showed that the roots of a polynomial are the eigenvalues of the "Frobenius companion matrix" whose elements are the coefficients of the polynomial. One way to find the roots of an orthogonal series $P(x)$ is to express $P(x)$ as a sum of monomials, and then to calculate the roots as the eigenvalues of the standard companion matrix. However, expressing a polynomial by its monomial coefficients is not as well conditioned as expressions in terms of orthogonal polynomials. It is not necessary to convert the orthogonal series into the power basis. It has been shown that a polynomial equation in the form of a truncated orthogonal series can be solved directly: the roots are the eigenvalues of the orthogonal Frobenius matrix [53, 58, 62-64, 67].

6.2.1 Computing Roots of Legendre Truncated Series

For general orthogonal polynomials, the companion matrix was first discovered by Specht, and then independently rediscovered several times for specific orthogonal polynomials. Let $f_N(x)$ be a polynomial of degree N expressed in the Legendre basis as

$$f_N(x) = \sum_{i=0}^{N} a_i P_i(x).$$

All roots of $f_N(x)$ are eigenvalues of the $N \times N$ matrix C whose elements are:

$$\begin{cases} \delta_{2,k} & i = 1, k = 1, 2, \ldots . N, \\ \frac{i-1}{2i-1}\delta_{i,k+1} + \frac{i}{2i-1}\delta_{i,k-1} & , \ i = 1 \ldots (N-1), k = 1 \ldots N, \\ (-1)\frac{N}{2N-1}\frac{a_{i-1}}{a_N} + \frac{N-1}{2N-1}\delta_{k,N-1} \ , & i = N, k = 1, \ldots, N, \end{cases} \quad 6.3$$

where δ_{ik} is the Kronecker delta function such that $\delta_{ik} = 0 \ if \ i \neq k$, while $\delta_{ii} = 1$ for all i.

Table 6.2 Matlab code for the Legendre polynomial companion matrix

```
clear all

N=16;

s=[2133 201 10550 84 18990 0 27430 0 28985 0 23247 0 27675 0 19227 0
21879]

P=zeros(N,N);

P(1,2)=1;

P(N,1)=-(s(1)/s(N+1))*(N/(2*N-1));

for j=2:(N-1), P(j,j-1)=(j-1)/(2*j-1); P(j,j+1)=j/(2*j-1); P(N,j)=-
(s(j)/s(N+1))/(N/(2*N-1)); end

P(N, N-1)=P(N,N-1)+(N-1)/(2*N-1); P(N,N)=-(N*s(N)/(s(N+1)*(2*N-1)));

x=eig(P)
```

6.2.2 Computing Roots of Chebyshev Truncated Series

Day and Romero [63] have shown that a polynomial equation in the form of a truncated Chebyshev series can be solved directly: the N roots are the eigenvalues of the Chebyshev–Frobenius matrix. Let $f_N(x)$ a polynomial of degree N expressed in the basis as:

$$f_N(x) = \sum_{i=0}^{N} a_j T_j(x)$$

All roots of $f_N(x)$ are eigenvalues of the $NXN\ matrix\ C$ whose elements are:

$$a_{jk} = \begin{cases} \delta_{2,k} \quad , & j = 1, k = 1, 2, \ldots . N, \\ \frac{1}{2}\{\delta_{j,k+1} + \delta_{j,k-1}\}, & , \ j = 2 \ldots (N-1), k = 1 \ldots N, \\ (-1)\ \frac{a_{j-1}}{a_N} + (\frac{1}{2})\delta_{k,N-1}, & j = N, k = 1, \ldots, N, \end{cases} \qquad 6.4$$

where δ_{jk} is the Kronecker delta function such that $\delta_{jk} = 0\ if\ j \neq k$, while $\delta_{jj} = 1$ for all j.

Table 6.3 Matlab Code for the Chebyshev Polynomial Companion Matrix

```
n=18
c=[7 23 55 0 12 0 0 405 0 598 0 0 0 444 0 0 0 663]
% Define A as the Frobenius-Chebyshev companion matrix.
A=zeros(n-1);
A(1,2)=1;
for j=2:n-2
    for k=1:n-1
        if j==k+1 || j==k-1
            A(j,k)=0.5;
        end
    end
end
for k=1:n-1
    A(n-1,k)=-c(k)/(2*c(n));
end
A(n-1,n-2)=A(n-1,n-2)+0.5;
eigvals=eig(A)
```

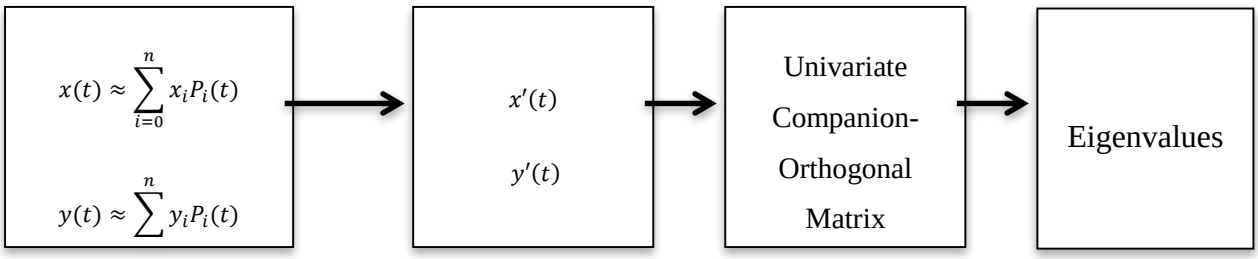


Figure 6.1: Flowchart of our approach for computing critical points of parametric curves approximated by orthogonal polynomials. First, we represent handwritten characters as parametric curves approximated by orthogonal truncated series. Then, we compute the first derivative of the orthogonal series. After that, we find the roots of orthogonal polynomials by computing the eigenvalues of the orthogonal companion matrix.

Algorithm for Computing Critical Points of Parametric Curves Expressed in Orthogonal Bases:

1. Compute the coefficents a_i of the first derivative of the expanded orthogonal series.
2. Form a companion matrix for the orthogonal polynomial with the a_n insterted as the coefficents of the derivative orthogonal series.
3. Compute the eignvalues of the orthogonal companion matrix.

6.3 Numerical Examples

In this section we present some examples to illustrate our technique for finding critical points of orthogonal series. We use different approaches: The first approach is to convert the orthogonal series to the power basis, and then compute the derivative and the roots of the derivative in the power basis. The second approach is to compute the derivative and the roots of derivative in the orthogonal basis.

Example 6.1 Consider the Legendre series $f(x) = 7P_0(x) + 23P_1(x) + 55P_2(x) + 12P_4(x) + 405P_7(x) + 598P_9(x) + 444P_{13}(x) + 663P_{17}(x)$.

1- First approach (convert the Legendre series to the monomial series, and then compute critical points in monomial basis):

$$f(x) = -16 + \frac{135141493}{32768}x + \frac{75}{2}x^2 - \frac{675098565}{4096}x^3 + \frac{105}{2}x^4 + \frac{16770760083}{8192}x^5 - \frac{49743153603}{4096}x^7 + \frac{651677744575}{16384}x^9 - \frac{309748225059}{4096}x^{11} + \frac{679524501075}{8192}x^{13} - \frac{199258149285}{4096}x^{15} + \frac{386795230965}{32768}x^{17}$$

$$\frac{df(x)}{dx} = \frac{135141493}{32768} + 75x - \frac{2025295695}{4096}x^2 + 210x^3 + \frac{83853800415}{8192}x^4 - \frac{348202075221}{4096}x^6 + \frac{5865099701175}{16384}x^8 - \frac{3407230475649}{4096}x^{10} + \frac{8833818513975}{8192}x^{12} - \frac{2988872239275}{4096}x^{14} + \frac{6575518926405}{32768}x^{16}$$

2- Second approach (computing critical points in the Legendre basis):

$f(x) = 7P_0(x) + 23P_1(x) + 55P_2(x) + 12P_4(x) + 405P_7(x) + 598P_9(x) + 444P_{13}(x) + 663P_{17}(x)$.

$$\frac{df(x)}{dx} = 21879P_{16}(x) + 19227P_{14}(x) + 27675P_{12}(x) + 23247P_{10}(x) + 28985P_8(x) + 27430P_6(x) + 18990P_4(x) + 84P_3(x) + 10550P_2(x) + 201P_1(x) + 2133P_0(x).$$

Compute eigenvalues of Legendre-companion matrix for Legendre series. Table 6.4 shows roots of the polynomial in Legendre basis and monomial basis.

Table 6.4 Critical Points of the Legendre Truncated Series of Degree 17

Critical points in Legendre basis	Critical points in monomial basis
-0.9561 + 0.0000i	0.9578
-0.8556 + 0.0000i	0.9102
-0.8534 + 0.0000i	0.8513
0.9556 + 0.0000i	0.7123
0.8681 + 0.0000i	0.5586
0.8400 + 0.0000i	0.4925
-0.6267 + 0.0000i	−0.9584
-0.4800 + 0.0000i	−0.9078
-0.2965 + 0.2957i	−0.8543
-0.2965 - 0.2957i	−0.7093
0.6308 + 0.0000i	−0.5644
0.4759 + 0.0000i	−0.4877
0.2967 + 0.2953i	0.3056
0.2967 - 0.2953i	−0.3067
0.0005 + 0.0906i	0.1021
0.0005 - 0.0906i	−0.1018

Example 6.2 Consider the Chebyshev series $f(x) = 7T_0(x) + 23T_1(x) + 55T_2(x) +$

$12T_4(x) + 405T_7(x) + 598T_9(x) + 444T_{13}(x) + 663T_{17}(x)$

1- First approach (convert the Chebyshev series to monomial series):

$$f(x) = 43450368x^{17} - 184664064x^{15} + 324980736x^{13} - 305989632x^{11} + 166236928x^9 - 52360128x^7 + 9080016x^5 + 96x^4 - 751704x^3 + 14x^2 + 19613x - 36.$$

2- Second approach (computing roots in the Chebyshev basis):

Compute eigenvalues of Chebyshev-companion matrix for Chebyshev series. Table 6.5 shows roots of the polynomial in Chebyshev basis and monomial basis.

Table 6.5 Roots of Chebyshev Truncated Series of Degree 17

Roots in Chebyshev basis	Roots in monomial basis
0.9908 + 0.0000i	-0.9916 + 0.0000i
0.9195 + 0.0120i	-0.9199 + 0.0182i
0.9195 - 0.0120i	-0.9199 - 0.0182i
0.8117 + 0.0000i	-0.8101 + 0.0000i
0.6298 + 0.0000i	-0.6260 + 0.0000i
0.5343 + 0.0000i	-0.5436 + 0.0000i
0.4383 + 0.0000i	-0.4307 + 0.0000i
0.2105 + 0.0000i	0.9908 + 0.0000i
0.0018 + 0.0000i	0.9195 + 0.0120i
-0.2145 + 0.0000i	0.9195 - 0.0120i
-0.9916 + 0.0000i	0.8117 + 0.0000i
-0.9199 + 0.0182i	0.6298 + 0.0000i
-0.9199 - 0.0182i	0.5343 + 0.0000i
-0.8101 + 0.0000i	0.4383 + 0.0000i
-0.4307 + 0.0000i	-0.2145 + 0.0000i
-0.6260 + 0.0000i	0.2105 + 0.0000i
-0.5436 + 0.0000i	0.0018 + 0.0000i

Chapter 7

Conclusion and Future work

7.1 Summary

The original motivation for this thesis was to find the critical points of parametric curve representations in handwriting recognition. Handwritten symbols can be represented as parametric curves approximated in orthogonal bases such as Chebyshev, Legendre, or Legendre-Sobolev bases. It is easy in this representation to find all the critical points such as self-intersection points, number of local maxima, minima, loops, cusps, and so on. These critical points are used to determine features for recognition.

These points can be computed by finding the roots for $x'(t) = 0$ and $y'(t) = 0$. This is done by univariate polynomial root finding. Roots of univariate-approximated polynomials can be found by computing the eigenvalues of the companion resultant matrix. Another approach to finding critical points is to convert parametric representations to implicit representations of the form $f(x, y) = 0,$ where $f(x, y)$ is a bivariate polynomial. Singular points of parametric curves can be computed based on resultants. First, we convert the parametric curves to implicit curves of the form $f(x, y) = 0$. This operation can be done as $Res(X - x(t), Y - y(t))$. Then, the critical points are computed from the implicit representation by solving the roots of the system of polynomial equations $f = f_x = f_y = 0$. The common zeros of bivariate polynomial equations are computed based on the resultant. The method described in this thesis is based on the hidden variable resultant method for polynomials. The hidden variable resultant method is based on selecting one variable, say y, and rewriting bivariate

polynomials f_x and f_y in x with coefficients in that are polynomials in y. The critical points of truncated orthogonal series are computed by first computing the first derivative of the orthogonal series, and then computing the roots of the derivative series. One way to find the derivative and the roots of the derivative of a truncated orthogonal series $P(x)$ is to express $P(x)$ as a sum of monomials, and then to calculate the derivative and the roots as the eigenvalues of the standard companion resultant matrix. On the other hand, the conversion between an orthogonal basis polynomial and its power basis form is ill conditioned (change the representation of a polynomial from orthogonal basis to power basis can amplify numerical errors. Error bounds can grow exponentially with the degree of the polynomial, and the relative errors can be infinitely larger in one basis than in another. Our contribution in this thesis is to determine whether the mathematical tools (roots, derivatives, resultants) exist to perform all the necessary operations in the orthogonal polynomial basis without converting to monomial bases. We introduce our approaches for computing critical points based on resultants in an orthogonal basis, rather than perform ill-conditioned conversions. First, we develop an algorithm to compute the coefficients of the derivative of orthogonal series in orthogonal bases, and then we compute the roots of the derivative based on resultants matrices built in orthogonal bases such as the Sylvester matrix and the companion matrix. In this work, we compute the Sylvester resultant matrix for the Legendre basis.

7.2 Future Work

The treatment of handwritten symbols as parametric curves approximated by polynomials in non-monomial bases points to several directions for further investigation. Transformation from orthogonal bases to the power basis is ill conditioned. We need to avoid such conversion, and compute all polynomial manipulations such as resultants, derivatives, QR decompositions, SVDs, and GCDs in the right basis. These computations are to be done

without conversion into the standard power basis. Moreover, as long as we have simple multiplication formulas for orthogonal polynomial bases, we can compute all resultants matrices in orthogonal bases based on multiplication formulas. We wish to simplify the multiplication formula for Legendre polynomials in order to have a simple algorithm to compute resultant matrices in the Legendre basis.

Bibliography

[1] S.M. Watt, "Polynomial approximation in handwriting recognition" In: Proceedings of the 2011 International Workshop on *Symbolic-Numeric Computation*. SNC '11, ACM (2011) 3–7.

[2] R. Hu and S. M. Watt, " Determining Points on Handwritten Mathematical symbols" pp. 168-183, *Proc. 2013 Conferences on Intelligent Computer Mathematics*, (CICM 2013), July 8-12 2013, Bath, UK, Springer Verlag LNAI 7961.

[3] S. M. Watt, " On the mathematics of Mathematical Handwriting Recognition", *Proc. 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, September 23-26 2010, Timişoara, Romania.

[4] B.W. Char and S. M. Watt, "Representing and Characterizing Handwritten Mathematical Symbols Through Succinct Functional Approximation", pp. 1198–1202, *Proc. International Conf. on Document Analysis and Recognition,* (ICDAR 2007), IEEE Press.

[5] O. Golubitsky and S. M. Watt, "Online Recognition of Multi-Stroke Symbols with Orthogonal Series", pp. 1265-1269, *Proc. 10th International Conf. on Document Analysis and Recognition*, (ICDAR 2009), IEEE Computer Society.

[6] O. Golubitsky and S. M. Watt, "Online Stroke Modeling for Handwriting Recognition", pp. 72-80, Proc. *18th International Conf. on Computer Science and Software Engineering*, (CASCON 2008), IBM Canada.

[7] V. Mazalov and S. M. Watt, " Digital Ink Compression via Function Approximation", pp. 688-694, Proc. *12th International Conference on Frontiers in Handwriting Recognition*, (ICFHR 2010), November 16-18 2010, Kolkata, India, IEEE Computer Society.

[8] O. Golubitsky and S. M. Watt, "Distance-Based Classification of Handwritten Symbols", *International Journal of Document Analysis and Recognition,* 13 (2) pp. 133-146, June 2010, Springer.

[9] B. Keshari and S. M. Watt, "Online Mathematical Symbol Recognition using SVMs with Features from Functional Approximation", *Electronic Proc. Mathematical User-Interfaces Workshop 2008*, (MathUI 2008), http://www.activemath.org/workshops/MathUI/08/proceedings.

[10] O. Golubitsky and S. M. Watt, "Online Computation of Similarity between Handwritten Characters", pp. C1-C10, *Proc. Document Recognition and Retrieval XVI, (DRR XVI 2009), SPIE and IS&T.*

[11] R. Hu, "Representation, Recognition and Collaboration with Digital Ink" University of Western Ontario – Electronic Thesis and Dissertation Repository. Paper 1771, 2013.

[12] C. T. C Wall , *Singular points of plane curves*, London Mathematical Society student texts, 2004.

[13] V. C Mavron and T. N Phillips, *Elements of Mathematics for Economics and Finance*, Chapter 7: "Maxima and Minima" , 2007.

[14] A. Rivera-Figueroa and J. C. Ponce-Campuzano, “Derivative, maxima and minima in a graphical context”, Mathematics Education Department, Centre for Research and Advanced Studies, Av. Instituto Polite´cnico Nacional 2508, Col. San Pedro Zacatenco, Mexico City, 07360 Mexico, (Received 24 September 2011).

[15] G. E. Farin, J. Hoschek, and M. Kim, *Handbook of Computer Aided Geometric Design*, Elsevier, 2002.

[16] R. J. Walker, *Algebraic Curves* , Springer-Verlag, New York, Heidelberg, Berlin, 1978.

[17] C. Li and R. Wang, “Curvatures at the singular points of algebraic curves and surfaces”, School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China, May 2014.

[18] K. Thapa “ Data compression and critical points detection using normalized symmetric scattered matrix ”, Department of Surveying and MappingFenis State University, Big Rapids, Michigan 49307

[19] F. Chen, W. Wang, and Y. Liu, “Computing singular points of plane rational curves”, Journal of Symbolic Computation 43 (2008) 92–117, Department of Mathematics, Received 11 August 2006; accepted 3 October 2007, Available online 24 October 2007.

[20] R. J. Walker, “Algebraic Curves” , Princeton University Press, 1950.

[21] N. Yang, “Structured matrix methods for computations on Bernstein basis polynomials”, Sheffield University –PhD thesis, England, January 2013.

[22] T. W. Sederberg, D. C. Anderson, and R. N. Goldman, “Implicit representation of parametric curves and surfaces” School of Mechanical Engineering, Purdue University, Computer Vision, Graphics and Image Processing, 28:72–84, 1984.

[23] D. Manocha and J. Demmel, “Algorithms for intersecting parametric and algebraic curves I: simple intersections”, ACM Trans. Graphics, pp. 73–100, (1994).

[24] G. Jónsson and S. Vavasis, “Accurate solution of polynomial equations using Macaulay resultant matrices”, Math. Comp., 74, pp. 221–262, (2005).

[25] S. S. Abhyankar and C. L. Bajaj, "Computations with Algebraic Curves", Computer Science Technical Reports, Paper 688, (1988), http://docs.lib.purdue.edu/cstech/688.

[26] Ioannis Z. Emiris, Victor Y. Pan, Elias Tsigaridas. Algebraic Algorithms. Teofilo Gonzalez. Computing Handbook Set - Computer Science, I, CRC Press, <hal 00776270>, 15 Jan 2013.

[27] Y. Nakatsukasa, V. Noferini, and A. Townsend, “Computing the common zeros of two bivariate functions via Bézout resultants”, Published in *Journal Numerische Mathematik*, pringer-Verlag New York, Inc. Secaucus, NJ, USA, 01 Jan 2015-.

[28] B. L. van der Wareden , *Modern Algebra*, V.1, Ungar, New York, 1950.

[29] J. Von Zur Gathen and J. Gerhard, *Modern computer algebra* , Cambridge University Press, 1999.

[30] K. O. Geddes, S. R. Czapor, and G. Labahn, *Algorithms for Computer Algebra*, Boston, Kluwer Academic, 1992.

[31] D. P. Jacobs, M. O. Reyes, and V. Trevisan, " The Resultant of Chebyshev Polynomials", *Canadian mathematical society*, pp.288-296, 2011.

[32] R. Barrio and J. Pena, "Basis conversions among univariate polynomial representations" . C. R. Math. Acad. Sci. Paris, 339(4):293{298, 2004.

[33] Y.-M. Li and X.-Y. Zhang, "Basis conversion among Bezier, Tchebyshev and Legendre. Comput. Aided Geom. Design, 15(6):637{642, 1998.

[34] R. M . Corless and N. Fillion. *A Graduate Introduction to Numerical Methods*: *From the Viewpoint of Backward Error Analysis.* New York, NY : Springer New York : Imprint: Springer, 2013.

[35] A. Bostan, B. Salvy, and E. Schost, "Fast conversion algorithms for orthogonal polynomials", Linear Algebra and its Applications 432(1): 249-258, 2010.

[36] A. Bostan, B. Salvy, and E. Schost, "Power series composition and change of basis", Proceedings ISSAC'08, pp. 269-276, ACM Press, 2008.

[37] J. Gerhard, " Modular algorithms for polynomial basis conversion and greatest factorial factorization", pp. 125-141, 2000.

[38] O. Ruatta, " Algorithms for polynomials in Chebyshev basis", XLIM-DMI UMR 6172 Universite de Limoges, CNRS, 2011.

[39] Z. Yang, and B. Cui, " On the Bezoutian Matrix for Chebyshev Polynomials", Applied Mathematics and computation, Linear Algebra, Elsevier, 2012.

[40] P. Du, H. Jiang, and L. Cheng, " Accurate, evaluation of polynomials in Legendre Basis", Journal of Applied Mathematics, 23 July 2014.

[41] H. Cheng and G. Labahn, " On computing polynomial GCDs in alternate bases", Dept. of Mathematics and Computer Science, University of Lethbridge, Lethbridge, Canada and Symbolic Computation Group, School of Computer Science, University of Waterloo, Waterloo, Canada, 2008.

[42] R. A. Horn and C. R. Johnson, Matrix Analysis, Press, University of Cambridge, 1985.

[43] B. Beckermann and G. Labahn, " Fraction –free computation of matrix rational interpolants and matrix GCDs", SIAM J. Matrix Anal. Appl., 2000.

[44] W. Gautschi, *Orthogonal polynomials: applications and computation*, Cambridge University Press, 1996, Ada Numerica (1996), pp. 45-119.

[45] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, National Bureau of Standard Applied Mathematics Series 55, 1964.

[46] A. D. Poularikas, *Handbook of Formulas and Tables for Signal Processing*, Chapter 21,

"Legendre Polynomials", CRC Press 1998.

[47] T.S. Chihara, *An introduction to orthogonal polynomials*, Gordon and Breach Science Publishers, New York, 1978. Mathematics and its Applications, Vol.13.

[48] T. Sakkalis and R. Farouki, " Singular points of algebraic curves", *Symbolic Computation 405-421*,Waston Research Center, 1988.J. S

[49] M. M Flood, " The Resultant matrix of two polynomials", *the American Mathematical society*, 10/1937, volume 43, issued 5, 10.

[50] A. Townsend, "Computing with functions in two dimensions", PhD thesis, University of Oxford, 2014.

[51] A. Wallack, I. Z. Emiris, D. Manocha, " MARS: A Maple/Matlab/ C Resultant –based Solver".

[52] O. J. D. Barrowclough, " Approximate methods for change of representation and their applications in CAGD", 2012.

[53] J. P. Boyd, "Roots, Extrema & Contours for Bivariate Polynomials: Chebyshevizing Algebraic Geometry", University of Michigan, September 2012.

[54] A. J. Sommese and C. W. Wampler, "The Numerical Solution of Systems of Polynomials Arising in Engineering and Science", World Scientific, Singapore, 2005.

[55] J.C. Adams," On the expression for the product of any two Legendre's coefficients by means of a series of Legendre coefficients" , Proc. Roy. Soc. London, 27, 63-71, 1878.

[56] P. Boito, *Structured Matrix Based Methods for Approximate Polynomial GCD*, volume 15, pp21-43, 2011.

[57] B. Bastl and F. Jezek, *Comparison of implicitization methods, Journal for Geometry and Graphics*, Volume 9 (2005), No. 1, 11-29.

[58] J. P. Boyd, " Computing zeros on a real interval through Chebyshev expansion and polynomial rootfnding", SIAM J. Numer. Anal, volume 40, pp. 1666 -1682, 2002.

[59] D. S. Bernstein, *Matrix mathematics: theory, fact, and formulas*, Princeton, N.J. : Princeton University Press, 2nd ed., c2009.

[60] S. P. Diaz and J. R. Sendra, " *A univariate resultant-based implicitization algorithm for surfaces", Journal of symbolic computation*, Elsevier Ltd, 43, 2008, 118-139.

[61] R.M. Corless, M.W. Giesbrecht, I. Kotsireas and S.M. Watt, Numerical Implicitization of Parametric Hypersurfaces with Linear Algebra, pp. 174-183, *Proc. Artificial Intelligence with Symbolic Computation, (AISC 2000),* July 17-19 2000, Madrid, Spain, Lecture Notes in Artificial Intelligence, No. 1930, Springer Verlag.

[62] J. P. Boyd, *Chebyshev and Fourier spectral methods*, 2nd edn, 2001, Dover, Mineola, New York

[63] D. Day and L. Romero, “Roots of polynomials expressed in terms of orthogonal polynomials”, 2005, SIAM J Num Anal 43:1969–1987.

[64] J. P. Boyd, *Solving Transcendental EquationsolThe Chebyshev Polynomial Proxy and Other Numerical Rootfinders, Perturbation Series, and Oracles*, SIAM, Oct 23, 2014 - Mathematics 462 pages.

[65] T. N. Phillips, “On the Legendre Coefficients of a General-Order Derivative of- an Infinitely Differentiable Function”, *IMA Journal of Numerical Analysis* (1988) 8, 455-459, [Received 2 December 1987 and in revised form 16 March 1988.

[66] A. Karageorghis, “A note on the Chebyshev coefficients of the general order derivative of an infinitely differentiable function”. J. Comp. Appl. Math. 21, 129-132., 1988.

[67] J. P. Boyd, “Computing the zeros, maxima and inflection points of Chebyshev, Legendre and Fourier series: solving transcendental equations by spectral interpolation and polynomial rootfinding” , Received: 3 January 2006 , Accepted: 26 July 2006 ,Published online: *18 November 2006, Springer Science and Business Media B.V. 2006.*

Curriculum Vitae

Name:	Aoesha Alsobhe
Date of Birth:	01/16/1989
Post-Secondary: Education and Degrees:	Taibah University (TU) Faculty of Computer Sciences and Engineering Madinah, Saudi Arabia 2006 – 2011 B.Sc (Major Computer Sciences) York University (YUELI program) Toronto, Canada 2012- 2013 CultureWorks ESL London, Canada 2013
Honours and Awards:	Excellent and Second Degree Honour in Computer Science Department Taibah University, Madinah 2011 King Abdullah Program for Scholarship 2012