Handwriting Feature Extraction via Legendre-Sobolev Matrix Representation

Parisa Alvandi
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
palvandi@uwaterloo.ca

Stephen M. Watt
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada
smwatt@uwaterloo.ca

***Abstract*—Despite the advancement of handwriting recognition, the mathematical expression-level recognition rates are still well below the threshold that is acceptable by a mathematics oriented system. Two-dimensionality nature of math formulas and the large set of math characters with different variations in style and size form the main challenges that the mathematical handwriting recognition problem faces. To address these difficulties, the way handwritten data is represented and the methods to compute certain features from the chosen representation are the two critical questions to answer.**

To this aim, we treat handwritten characters as approximated parametrized coordinate curves in Legendre-Sobolev bases. This representation empowers us to study the geometrical features of handwritten characters as a whole. These geometrical features are equivalent to baselines, bounding boxes, loops, and cusps appearing in handwritten characters. In this paper, we propose methods for computing the derivative, roots, and gcd of polynomials in Legendre-Sobolev bases to find such features without needing to convert the approximations to the monomial basis.

***Keywords*-mathematical handwriting recognition, numerical approximation, algebraic curves, orthogonal polynomial series**

I. Introduction

Recognition of mathematical handwritten expressions on pen-based devices has a history since 1969 (see [5]) and plays an important role in the development of software in areas such as physics [22], [23], geometric theorem proving [21], and algebraic Intelligent Tutoring Systems (ITS) [3].

In 2012, the authors of [3] estimated that 91-97% recognition accuracy is required for acceptance in a mathematics oriented ITS system. However, based on results from the fourth international Competition on Recognition of Online Handwritten Mathematical Expressions (CROHME) [25], expression-level recognition rates are still well below this threshold. In fact, there are several reasons that turn mathematical handwriting recognition into a more challenging problem compared to natural language recognition. In math formulas, appearance of subscripts and superscripts is common, thus, one needs to deal with a 2-dimensional handwriting recognition problem. Furthermore, the set of math characters have more varieties than any natural language. Math characters also appear in different sizes and places which in turn can change the meaning of a math expression. Since there is not any mathematical dictionary to consult about what an expression means, thus it is necessary to establish smart ideas to increase mathematical handwriting recognition rates.

One approach, for increasing mathematical handwriting recognition rates, is via locating some of the significant features of handwritten characters by identifying some special points. To refer to these points of any kind, we use the term “determining points”. In 2010, Infante Velázquez [30] developed an annotation tool to record determining points manually for handwritten characters represented in InkML [31]. However, their method is subject to device resolution and variations in style. Similar problems exist in [8]. In addition, Zanibbiet al. [33] proposed a technique to automatically improve the legibility of handwriting; once a formula is written, the individual hand-drawn symbols can be translated and scaled to closely approximate their relative positions and sizes in a corresponding typeset version. This technique detects baseline locations by comparing symbols bounding boxes, which leads to trouble with vertical placement and scale. For example, this method fails to distinguish between “$x2$” and “x^2”. In 2012, Hu and Watt [17] presented an algorithm to find turning points that determine the shape of characters, but that approach lacked the ability to capture the geometric meaning of each determining point and therefore does not provide sufficient information such as the location of baseline. Harouniet al. [15] later proposed a method to find determining points in handwritten Arabic characters. The method splits the input character into several pieces, and then calculates the extremum points of each piece and records them as determining points. However, this method is not optimal as it may generate undesired points that lack meaning. More precisely, the extremum points of the individual strokes of a handwritten character might not be the extremum points of that character. In [29], the authors introduced a pen-based user interface for simplifying the task of handwriting of mathematical expressions. Visible bounding boxes around sub-expressions are automatically generated as the system detects relevant spatial relationships between symbols including superscripts, subscripts, and fractions.

Among different methods, we are interested in online handwriting recognition. In [32], the authors use segment-and-decode classifiers for online handwriting recognition. The work [28] focuses on solving online handwriting recognition by making use of Hidden Markov Models (HMMs) [16] or hybrid approaches combining HMMs and Feed-forward Neural Networks [4]. The first HMM-free models were based on Time Delay Neural Networks (TDNNs) [10], [20], [27], and more recent work focuses on Recurrent Neural Network (RNN) variants such as Long-Short-Term-Memory networks (LSTMs) [11], [14]. In [6], the authors describe an online handwriting system that uses a deep neural network architecture. The system combines methods from sequence recognition with an input encoding using Bézier curves.

In online recognition methods, how to encode handwritten data is a crucial question to researchers. What a computer sees as a handwritten character is a collection of points (x, y, t), with position (x, y) and timestamp t. One common character recognition approach is to resample the data and then match the output against N models by sequence alignment. Another approach is to identify some of the features like the number of loops, cusps, and so on, and use these features in a classifier. Then recognition is done by ranking choices by consulting a dictionary. But the difficulty with these methods is that there are so many similar math characters and that makes the comparison against a symbol model slow. An approach to overcome these obstacles is to treat traces as curves.

In [7], the authors propose truncated Chebyshev polynomial series representation of parametric plane curves for representing handwritten characters. The authors have shown that degree 10 approximations for handwritten characters yield high recognition rates. In [12], the authors propose an online handwriting recognition method based on [7] for representing coordinate curves of each character in the Legendre basis. Authors of [13] have reported experimental results that demonstrate that representing coordinate curves in the Legendre-Sobolev basis has higher detection rates compared to when these curves are represented in the Legendre basis. In [2], we have proposed an online method for computing Legendre-Sobolev representation of handwritten characters.

Representing parametrized coordinate curves in orthogonal basis helps in analyzing the geometry of each character. This analysis is crucial in mathematical handwriting recognition when the characters in math formulas have varying semantically meaningful baselines (see [18], [19]).

In [18], Hu and Watt proposed an algorithm to compute determining points by relying on computation of the local minimum and maximum of the parametrized Legendre-Sobolev approximations of handwritten characters. The authors suggest computing the points on the curve corresponding to values of parameter s such that $Y'(s) = 0$ with $Y(s)$ being the y-coordinate curve. Computation of such points is done by relying on the Newton method. This implies that conversion from a Legendre-Sobolev basis to the monomial basis is required which is known to be ill-conditioned. Thus, in this paper, we are interested in computing geometric operations such as derivative, root finding, and gcd in Legendre-Sobolev bases without transforming into the monomial basis. Our methods rely on linear algebra arithmetic operations such as matrix multiplications and solving Diophantine equations.

This paper is organized as follows. After presenting preliminaries in Section II, we explain how to compute derivative, roots, and gcd of polynomials in a Legendre-Sobolev basis in Section III. Then Section IV investigates backward error of computing the roots in Legendre-Sobolev basis and sensitivity analysis of computing the critical points for handwritten characters. Section V concludes the article by discussing the possible approaches to use the methods presented in this paper for building new handwriting recognition models.

II. Preliminaries

Digital ink is generated by sampling points from handwritten curves and is essentially a series of points (x, y). We use these points to compute "moment integrals" and from them we approximate the coefficients of the coordinate curves $X(\lambda)$ and $Y(\lambda)$ on an orthogonal basis, where λ can be either time or length of handwritten curves, see [2]. The work [7], [12], [2] have shown that the coordinate curves $X(\lambda)$ and $Y(\lambda)$ for handwritten characters can be modeled by truncated Chebyshev, Legendre, and Legendre-Sobolev series, respectively. The coefficients of such series can be used for classification and therefore recognition of each character.

Suppose that an inner product between two functions $f, g : [-1, 1] \to \mathbb{R}$ is defined by

$$\langle f(\lambda), g(\lambda)\rangle = \int_{-1}^{1} f(\lambda)\, g(\lambda) d\lambda + \mu \int_{-1}^{1} f'(\lambda) g'(\lambda) d\lambda, \quad (1)$$

where $\mu \in \mathbb{R}_{\geq 0}$. Equation (1) is a special case of Legendre-Sobolev inner product, where we have restricted ourselves to the first order derivatives. We denote the Legendre-Sobolev polynomials corresponding to the inner product given by Equation (1) (which are also called Althammer polynomials, see [1]) of degree n by $S_n^{\mu}(\lambda)$. When μ is zero in Equation (1), then we denote the corresponding orthogonal polynomial $S_n^0(\lambda)$ of degree n by $P_n(\lambda)$, as well. In fact, the polynomial $P_n(\lambda)$ is the Legendre polynomial of degree n.

In this paper, we represent handwritten characters as truncated parametrized coordinate curves of degree n as

$$f(\lambda) \approx \sum_{i=0}^{n} \alpha_i S_i^{\mu}(\lambda),$$

where λ is arc length and $f(\lambda)$ is either of coordinate curves $X(\lambda)$ or $Y(\lambda)$. We compute the coefficients α_i by a matrix multiplication from moment integrals, where moments are computed numerically (see [2]). When $f(\lambda)$ is known in any basis, the coefficients α_i can also be computed by the equation

$$\alpha_i = \frac{\langle f(\lambda), S_i^{\mu}(\lambda)\rangle}{\langle S_i^{\mu}(\lambda), S_i^{\mu}(\lambda)\rangle}\ , \ i = 0, \dots, n,$$

where $\langle ., . \rangle$ is the inner product given by Equation (1).

III. Methods

The goal of this section is to provide methods for computing derivative, roots and gcd of the polynomials in a given Legendre-Sobolev basis without needing to change the basis. These results are mainly based on the following theorem.

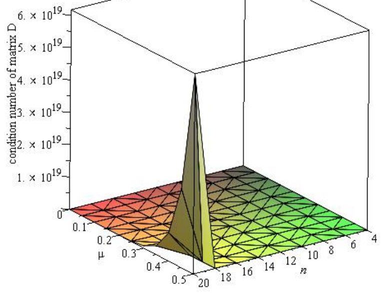


Fig. 1. Norm-2 condition number of matrix D w.r.t its size n and μ (3d plot).

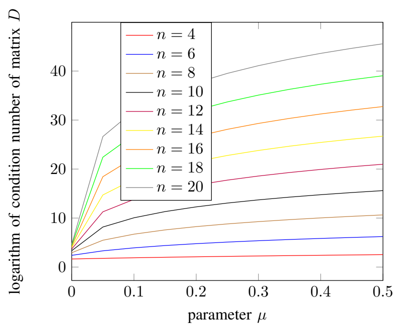


Fig. 2. Norm-2 condition number of matrix D w.r.t its size n and μ (2d plot), where the condition number axis is based-10-log-scaled.

Theorem 1 ([2]). *For $n \geq 1$ and $\mu \geq 0$, we have*

$$S^{\mu}_{i-1}(\lambda) = \sum_{j=1}^{i} N_{ij} P_{j-1}(\lambda),$$

where for $i, j = 1, \ldots, n+1$, matrix N can be formulated as

$$N_{ij} = \begin{cases} a_{j-1}(\mu) & \text{for } i = j \\ c_{j-1}(\mu) & \text{for } j = i - 2\ell, \ell = 1, \ldots, \lfloor \frac{i-1}{2} \rfloor \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

where

$a_0(\mu) = 1, a_v(\mu) = \sum_{k=0}^{\lfloor \frac{v-1}{2} \rfloor} \left(\frac{\mu}{4}\right)^k \frac{(v+2k-1)!}{(2k)!(v-2k-1)!}$, *for $v \geq 1$,*

$c_j(\mu) = a_j(\mu) - a_{j+2}(\mu)$, *for $j \geq 0$.*

Moreover,

$$N^{-1}_{ij} = \begin{cases} \dfrac{1}{a_{j-1}(\mu)} & \text{for } i = j \\ b_{j-1}(\mu) & \text{for } j = i - 2\ell, \ell = 1, \ldots, \lfloor \frac{i-1}{2} \rfloor \\ 0 & \text{otherwise,} \end{cases}$$

where $b_j(\mu) = \frac{1}{a_j(\mu)} - \frac{1}{a_{j+2}(\mu)}$, for $j \geq 0$.

A. Derivative of Legendre-Sobolev polynomials

The main result of this section is Theorem 2 which gives a formula for computing the derivative of a Legendre-Sobolev polynomial by a matrix multiplication. To prove this theorem, we have first computed the derivative of the Legendre-Sobolev polynomials in the Legendre basis. As a direct consequence of Theorem 1, we have the following corollary.

Corollary 1. *For $n \geq 1$, and $\mu \geq 0$, we have*

$$P_n(\lambda) = \frac{S^{\mu}_n(\lambda)}{a_n(\mu)} + \sum_{\ell=1}^{\lfloor \frac{n}{2} \rfloor} b_{n-2\ell}(\mu) S^{\mu}_{n-2\ell}(\lambda). \quad (3)$$

The following lemma represents the derivative of a Legendre-Sobolev polynomial of degree n as a linear combination of the Legendre polynomials.

Lemma 1 ([26]). *For $n \geq 1$ and $\mu \geq 0$, we have*

$$\frac{dS^{\mu}_n(\lambda)}{d\lambda} = \sum_{\ell=0}^{\lfloor \frac{n-1}{2} \rfloor} (2n - 4\ell - 1)\, a_{n-2\ell}(\mu) P_{n-2\ell-1}(\lambda). \quad (4)$$

The result below gives a method for computing the derivative of Legendre-Sobolev polynomials in the Legendre basis.

Proposition 1. *Let $f(\lambda) = \sum_{j=0}^{n} \alpha_j S^{\mu}_j(\lambda)$. Then the derivative $f'(\lambda) = \sum_{j=0}^{n-1} \beta_j P_j(\lambda)$ may be computed as $\beta_j = \sum_{i=0}^{n} H_{ij}\alpha_i$, where for $i, j = 1, \ldots, n$, we have*

$$H_{ij} = \begin{cases} (2\,j - 1) a_j(\mu) & \text{for } j = i - 2\,\ell, \ell = 0, \ldots, \lfloor \frac{i-1}{2} \rfloor \\ 0 & \text{otherwise.} \end{cases}$$

Proof. The result follows by letting $n = i$ and $j = i - 2\,\ell$ in Lemma 1. □

Finally, we claim Theorem 2 which gives a relation between Legendre-Sobolev polynomials and their derivatives.

Theorem 2. *Let $f(\lambda) = \sum_{j=0}^{n} \alpha_j S^{\mu}_j(\lambda)$. Then the derivative $f'(\lambda) = \sum_{j=0}^{n-1} \beta_j S^{\mu}_j(\lambda)$ may be computed as $\beta_j = \sum_{i=1}^{n} D_{i+1\ j}\, \alpha_i$, where for $i, j = 1, \ldots, n$, we have*

$$D_{ij} = \begin{cases} \frac{(2j-1)a_j(\mu)}{a_{j-1}(\mu)} + b_j(\mu) \sum_{k=0}^{\ell-1} (2i - 4k - 1) a_{i-2k}(\mu) \\ \qquad \text{for } j = i - 2\ell, \quad \ell = 0, \ldots, \lfloor \frac{i-1}{2} \rfloor \\ 0 \qquad \text{otherwise.} \end{cases}$$

Proof. To prove, first, note that $D = H\,N^{-1}$. Since both matrices, H and N^{-1} are lower triangular, thus for $i > j$, $D_{ij} = 0$. As we know $D_{ij} = \sum_{v=1}^{n} H_{iv} N^{-1}_{vj}$, and because of the shape of the matrices H and N^{-1}, we have $D_{ij} = \sum_{v=j}^{i} H_{iv} N^{-1}_{vj}$. We can rewrite the latter equation as follows:

$$D_{ij} = H_{ij} N^{-1}_{jj} + \sum_{v=j-1}^{i} H_{iv} N^{-1}_{vj}.$$

By applying the change of coordinate $v = i - 2\,k$, we obtain

$$D_{ij} = H_{ij} N^{-1}_{jj} + \sum_{k=0}^{\lfloor \frac{i-j}{2} \rfloor - 1} H_{i\ i-2\,k} N^{-1}_{i-2\,k\ \ j}.$$

If $i-j$ is odd, then all entries $N^{-1}_{i-2\,k\ j}$ are zero, for $k = 0,\dots,\lfloor\frac{i-j}{2}\rfloor - 1$. Suppose that $i-j$ is even, thus

$$\begin{aligned} D_{ij} &= H_{ij}N^{-1}_{jj} + \textstyle\sum_{k=0}^{\lfloor\frac{i-j}{2}\rfloor-1} H_{i\ i-2\,k}N^{-1}_{i-2\,k\ j} \\ &= \tfrac{(2\,j-1)a_j(\mu)}{a_{j-1}(\mu)} \\ &\quad + b_{j-1}(\mu)\textstyle\sum_{k=0}^{\ell-1}(2\,i-4\,k-1)\,a_{i-2k}(\mu), \end{aligned}$$

where $\lfloor\frac{i-j}{2}\rfloor = \ell$, and this completes the proof. □

Figures 1 and 2 demonstrate the relation between the condition number of matrix D, n and parameter μ, where n is the size of matrix D and μ is the parameter appearing in the Legendre-Sobolev inner product given by Equation (1).

B. Roots of Legendre-Sobolev polynomials

In this section, we propose a method to compute roots of polynomials in Legendre-Sobolev bases by computing the eigenvalues of some matrices. For such matrices the name of "comrade" matrix is suggested, and we use the same term here. The results of this section are the generalization of [9].

To compute the comrade matrix whose characteristic polynomial is equal to a given polynomial in Legendre-Sobolev bases (up to a constant), we first compute the recurrence relation between Legendre-Sobolev polynomials. To be more precise, we need to compute coefficients $h_{i,n-1}$, for a given $n\in\mathbb{N}$, and $i=0,\dots,n$, such that

$$\lambda S^{\mu}_{n-1}(\lambda) = \sum_{i=0}^{n} h_{i,n-1}S^{\mu}_{i}(\lambda). \tag{5}$$

In fact, the coefficients h_{ij}, for $i,j=0,\dots,n$, form the entries of the comrade matrix for a Legendre-Sobolev polynomial of degree n. To do so, we first compute $\lambda P_n(\lambda)$ in a Legendre-Sobolev basis.

Proposition 2. *For $n\geq 1$, and $\mu\geq 0$, we have*

$$\begin{aligned} \lambda P_n(\lambda) &= \tfrac{n+1}{2n+1}\tfrac{S^{\mu}_{n+1}(\lambda)}{a_{n+1}(\mu)} + (\tfrac{1}{a_{n-1}(\mu)} - \tfrac{n+1}{(2n+1)a_{n+1}(\mu)})S^{\mu}_{n-1}(\lambda) \\ &\quad + \textstyle\sum_{k=2}^{\lfloor\frac{n+1}{2}\rfloor} b_{n-2k+1}(\mu)S^{\mu}_{n-2k+1}(\lambda). \end{aligned} \tag{6}$$

Proof. Equation (7) is known for the Legendre polynomials:

$$\lambda P_n(\lambda) = \frac{n+1}{2n+1}P_{n+1}(\lambda) + \frac{n}{2n+1}P_{n-1}(\lambda). \tag{7}$$

By using Equations (3) and (7), we obtain:

$$\begin{aligned} \lambda P_n(\lambda) &= \tfrac{n+1}{2n+1}P_{n+1}(\lambda) + \tfrac{n}{2n+1}P_{n-1}(\lambda) \\ &= \tfrac{n+1}{2n+1}\left[\tfrac{S^{\mu}_{n+1}(\lambda)}{a_{n+1}(\mu)}\right] + \tfrac{n}{2n+1}\left[\tfrac{S^{\mu}_{n-1}(\lambda)}{a_{n-1}(\mu)}\right] \\ &\quad + \tfrac{n+1}{2n+1}\textstyle\sum_{k=1}^{\lfloor\frac{n+1}{2}\rfloor} b_{n-2k+1}(\mu)S^{\mu}_{n-2k+1}(\lambda) \\ &\quad + \tfrac{n}{2n+1}\textstyle\sum_{k=1}^{\lfloor\frac{n+1}{2}\rfloor-1} b_{n-2k+1}(\mu)S^{\mu}_{n-2k-1}(\lambda) \\ &= \tfrac{n+1}{2n+1}\tfrac{S^{\mu}_{n+1}(\lambda)}{a_{n+1}(\mu)} + (\tfrac{1}{a_{n-1}} - \tfrac{n+1}{(2n+1)a_{n+1}(\mu)})S^{\mu}_{n-1}(\lambda) \\ &\quad + \textstyle\sum_{k=2}^{\lfloor\frac{n+1}{2}\rfloor} b_{n-2k+1}(\mu)S^{\mu}_{n-2k+1}(\lambda). \end{aligned}$$

This proves the correctness of Equation (6). □

Thus, the conversion matrix E from Legendre-Sobolev polynomials to the Legendre polynomials multiplied by λ, is formulated in the following corollary.

Corollary 2. *For $n\geq 1$ and $\mu\geq 0$, we have*

$$\lambda P_{i-1}(\lambda) = \sum_{j=1}^{i+1} E_{ij}\,S^{\mu}_{j-1}(\lambda),$$

where for $i=1\dots,n$ and $j=1,\dots,n+1$,

$$E_{ij} = \begin{cases} \dfrac{j-1}{(2j-3)a_{j-1}(\mu)} & \textit{for } i+1=j \\ \dfrac{1}{a_{j-1}(\mu)} - \dfrac{j+1}{(2j+1)a_{j+1}(\mu)} & \textit{for } i-1=j \\ b_{j-1}(\mu) & \textit{for } i-2\ell+1=j, \\ & \quad \ell=2,\dots,\lfloor\frac{i}{2}\rfloor \\ 0 & \textit{otherwise.} \end{cases}$$

The following theorem formulates the conversion matrix for representing $\lambda\,S^{\mu}_{n}(\lambda)$ back in a Legendre-Sobolev basis.

Proposition 3. *For $n\geq 1$ and $\mu\geq 0$, we have*

$$\lambda\,S^{\mu}_{i-1}(\lambda) = \sum_{j=1}^{i+1} A_{ij}\,S^{\mu}_{j-1}(\lambda),$$

where for $i=1,\dots,n$ and $j=1,\dots,n+1$, we have

$$A_{ij} = \begin{cases} \frac{(j-1)\,a_{j-2}(\mu)}{(2j-3)\,a_{j-1}(\mu)} & \textit{for } j=i+1 \\ \frac{(j-2)\,a_j(\mu)+(j-1)\,a_{j-2}(\mu)}{(2\,j-3)\,a_{j-1}(\mu)} - \frac{(j+1)\,a_j(\mu)}{(2\,j+1)\,a_{j+1}(\mu)} & \\ & \textit{for } j=i-1 \\ \frac{(j-2)\,a_j(\mu)+(j-1)\,a_{j-2}(\mu)}{(2\,j-3)\,a_{j-1}(\mu)} - \frac{(j+1)\,a_j(\mu)+j\,a_{j+2}(\mu)}{(2\,j+1)\,a_{j+1}(\mu)} & \\ & \textit{for } j=i-2\ell+1, \ell=2,\dots,e-1 \\ \frac{a_j}{a_{j-1}} - \frac{(j+1)\,a_j(\mu)+j\,a_{j+2}(\mu)}{(2\,j+1)\,a_{j+1}(\mu)} & \textit{for } j=i-2\,e+1 \\ 0 & \textit{otherwise} \end{cases}$$

and $e=\lfloor\frac{i}{2}\rfloor$.

Proof. First, note that $A_{ij} = \sum_{k=1}^{n} N_{ik}E_{kj}$.To prove this theorem, we consider different cases:

- if $i>j+1$, then $A_{ij} = \sum_{k=i+1}^{n} N_{ik}E_{kj} = 0$. The first equation is valid because $N_{ik}=0$, for $k>i$ and the second equation is valid since $E_{kj}=0$, for $j>k+1$ and $i<j$.
- If $i<j+1$ and $j=i-2\,\ell$, for $\ell=1,\dots,\lfloor\frac{i}{2}\rfloor$, then

$$A_{i\ i-2\,\ell} = \sum_{\ell'=0}^{\lfloor\frac{i}{2}\rfloor} N_{i\ i-2\,\ell'}E_{i-2\,\ell'\ i-2\,\ell} = 0.$$

In fact, the first equality is true since $N_{ik}=0$ when $k \neq i-2\,\ell'$, for $\ell'=0,\ldots,\lfloor\frac{i}{2}\rfloor$. The second equality is true because $E_{i-2\,\ell'\ \, i-2\,\ell}=0$, for all ℓ'.

- When $j=i+1$, the result easily follows.
- If $j=i-1$, then we have

$$\begin{aligned} A_{ij} &= a_{i-1}(\mu)\left(\tfrac{1}{a_{j-1}(\mu)}-\tfrac{j+1}{(2j+1)a_{j+1}(\mu)}\right) \\ &\quad+\left(a_{i-3}(\mu)-a_{i-1}(\mu)\right)\tfrac{j-1}{(2j-3)a_{j-1}(\mu)} \\ &= \tfrac{(j-2)\,a_j(\mu)}{(2\,j-3)\,a_{j-1}(\mu)}-\tfrac{(j+1)\,a_j(\mu)}{(2\,j+1)\,a_{j+1}(\mu)} \\ &\quad+\tfrac{(j-1)\,a_{j-2}(\mu)}{(2\,j-3)\,a_{j-1}(\mu)}. \end{aligned}$$

- When $j=i-2\ell+1$, for $\ell=2,\ldots,\lfloor\frac{i}{2}\rfloor-1$, then:

$$\begin{aligned} A_{ij} &= b_{j-1}(\mu)\left(a_{i-1}(\mu)+\textstyle\sum_{k=1}^{\ell-2}c_{i-2k-1}(\mu)\right) \\ &\quad+c_{i-2\,\ell+1}(\mu)\left(\tfrac{1}{a_{j-1}(\mu)}-\tfrac{j+1}{(2j+1)\,a_{j+1}(\mu)}\right) \\ &\quad+c_{i-2\ell-1}(\mu)\left(\tfrac{j-1}{(2j-3)\,a_{j-1}(\mu)}\right) \\ &= \tfrac{(j-2)\,a_j(\mu)}{(2\,j-3)\,a_{j-1}(\mu)}-\tfrac{(j+1)\,a_j(\mu)}{(2\,j+1)\,a_{j+1}(\mu)} \\ &\quad+\tfrac{(j-1)\,a_{j-2}(\mu)}{(2\,j-3)\,a_{j-1}(\mu)}-\tfrac{j\,a_{j+2}(\mu)}{(2\,j+1)\,a_{j+1}(\mu)}. \end{aligned}$$

- When $j=i-2\ell+1$, for $\ell=\lfloor\frac{i}{2}\rfloor$, then:

$$\begin{aligned} A_{ij} &= c_{i-2\,\ell+1}(\mu)\left(\tfrac{1}{a_{j-1}(\mu)}-\tfrac{j+1}{(2j+1)a_{j+1}(\mu)}\right) \\ &\quad+b_{j-1}(\mu)\left(a_{i-1}(\mu)+\textstyle\sum_{k=1}^{\ell-2}c_{i-2k-1}(\mu)\right) \\ &= \tfrac{a_j}{a_{j-1}}-\tfrac{(j+1)\,a_j}{(2\,j+1)\,a_{j+1}}-\tfrac{j\,a_{j+2}}{(2\,j+1)\,a_{j+1}}. \end{aligned}$$

This completes the proof. □

Theorem 3. *Let C be defined such that for $i,j=1,\ldots,n$, $C_{ij}=A_{ij}$. Then the roots of Legendre-Sobolev polynomial $S_n^\mu(\lambda)$ coincide with the eigenvalues of matrix C.*

Proof. The proof is a corollary of Theorem 4. □

In fact, matrix C is the comrade matrix corresponding to polynomial $S_n^\mu(\lambda)$. Note that matrix C is lower Hessenberg, that is, $C_{ij}=0$ for $i>j+1>0$. Moreover, we have

$$\begin{aligned} \lambda\left[\ S_0^\mu(\lambda),\ldots,S_{n-1}^\mu(\lambda)\ \right]^T &= C\left[\ S_0^\mu(\lambda),\ldots,S_{n-1}^\mu(\lambda)\ \right]^T \\ &\quad+A_{n\ n+1}\,S_n^\mu(\lambda)\,\mathbf{e}_n, \end{aligned}$$

where $\mathbf{e}_n^T=\left[\ 0\ ,\ \ldots\ ,0\ ,\ 1\ \right]$ is a column vector of dimension n. Let ξ be a root of $S_n^\mu(\lambda)$, then we have

$$\xi\left[\ S_0^\mu(\xi),\ \ldots,\ S_{n-1}^\mu(\xi)\ \right]^T=C\left[\ S_0^\mu(\xi),\ \ldots,\ S_{n-1}^\mu(\xi)\ \right]^T.$$

The former equation implies that $\left[\ S_0^\mu(\xi),\ \ldots,\ S_{n-1}^\mu(\xi)\ \right]^T$ is the right eigenvector of ξ.

Example 1. *Consider polynomial $S_3^{0.2}(\lambda)=-3\,\lambda+4\,\lambda^3$. Matrix C for arbitrary parameter μ has the following form:*

$$C=\begin{bmatrix} 0 & \frac{1}{a_1(\mu)} & 0 \\ \frac{a_1(\mu)}{a_0(\mu)}-\frac{2\,a_1(\mu)}{3\,a_2(\mu)} & 0 & \frac{2\,a_1(\mu)}{3\,a_2(\mu)} \\ 0 & \frac{a_2(\mu)}{a_1(\mu)}-\frac{3\,a_2(\mu)}{5\,a_3(\mu)} & 0 \end{bmatrix}.$$

By letting $\mu=0.2$ in matrix C, we obtain

$$C=\begin{bmatrix} 0 & 1 & 0 \\ 0.3333333333 & 0 & 0.6666666667 \\ 0 & 0.625 & 0 \end{bmatrix}.$$

One can verify that the eigenvalues of matrix C form the set

$$\left\{5.03\times10^{-17},\ 0.866025403777222,\ -0.866025403777222\right\}.$$

In fact, the set $\left\{0,\pm\frac{\sqrt{(15+45\,\mu)(1+5\,\mu)}}{5(1+3\,\mu)}\right\}$ forms the roots of $S_3^\mu(\lambda)$, for any $\mu\geq0$.

Theorem 4. *Let $f(\lambda)=\sum_{i=0}^n\alpha_iS_i^\mu(\lambda)$ and*

$$B=C-\frac{na_{n-1}(\mu)}{(2n-1)a_n(\mu)}\frac{\mathbf{e}_n}{\alpha_n}\mathbf{c}^T,$$

where $\mathbf{e}_n^T=\left[\ 0\ ,\ \ldots\ ,0\ ,\ 1\ \right]$ is a column vector of dimension n and $\mathbf{c}^T=\left[\ \alpha_0,\ \ldots,\ \alpha_{n-1}\ \right]$. Then the eigenvalues of matrix B form the roots of $f(\lambda)$.

Proof. The steps of this proof follow exactly the ones in Theorem 2.3 in [9]. Note that with notations of [9], the quantity $\frac{h_{n,n-1}}{\gamma_n}$ in Theorem 2.3 is equal to $\frac{na_{n-1}(\mu)}{(2n-1)a_n(\mu)\alpha_n}$ when the basis is Legendre-Sobolev. □

C. Gcd of two polynomials in Legendre-Sobolev basis

Theorem 5 gives a method for computation of the monic gcd of two polynomials in Legendre-Sobolev basis. The computation of the coefficients of the gcd in this theorem is done by solving a set of Diophantine equations given by Equation (8).

Theorem 5. *Let $f(\lambda)=\sum_{i=0}^n\alpha_iS_i^\mu(\lambda)$ and $h(\lambda)=\sum_{i=0}^m\beta_iS_i^\mu(\lambda)$, where $n>m$, and matrix B is the matrix defined in Theorem 4 for polynomial $f(\lambda)$. Let also $g(\lambda)$ be the monic gcd of polynomials $f(\lambda)$ and $h(\lambda)$ and $g(\lambda)=\lambda^k+\sum_{i=0}^{k-1}\hat{\zeta}_i\lambda^i=\sum_{i=0}^k\zeta_iS_i^\mu(\lambda)$. Define $h(B)=\beta_0I_n+\beta_1S_1^\mu(B)+\cdots+\beta_mS_m^\mu(B)$, then $k=n-\mathrm{rank}(h(B))$. Furthermore, let c_i be the ith column of matrix $h(B)$, for $i=1,\ldots,n$. Then the columns $c_{k+1},\ldots,c_n$ are linearly independent and if the numbers x_{ij} are defined by*

$$c_i=x_{i,k+1}c_{k+1}+\sum_{j=k+2}^n x_{ij}c_j,\quad for\ i=1,\ldots,k,\quad x_{ij}\in\mathbb{R}, \tag{8}$$

then $\zeta_{i-1}=\zeta_kx_{i,k+1}$, for $i=1,\ldots,k$, where $\zeta_k=\prod_{\ell=1}^kB_{\ell\ \ell+1}$.

Proof. The proof follows the one of Theorem 1 in [24]. One key aspect here is that $B=S\,B'\,S^{-1}$, where B' is the companion matrix corresponding to $f(\lambda)$ in the monomial basis and S is the matrix whose rows are the coefficients of $S_0^\mu(\lambda), \ldots, S_{n-1}^\mu(\lambda)$ in the monomial basis. □

Example 2. *Suppose that*

$$\begin{aligned} f(\lambda) &= (\lambda-1)^2(\lambda+2)^3 = \tfrac{82}{15}S_0^\mu(\lambda) - \tfrac{73}{28}S_1^\mu(\lambda) \\ &\quad -\tfrac{388}{105}S_2^\mu(\lambda) + \tfrac{1529}{2556}S_3^\mu(\lambda) + \tfrac{8}{35}S_4^\mu(\lambda) + \tfrac{40}{4473}S_5^\mu(\lambda) \end{aligned}$$

and

$$\begin{aligned} h(\lambda) &= (\lambda-1)^3(\lambda+2) = -\tfrac{14}{5}S_0^\mu(\lambda) + \tfrac{17}{4}S_1^\mu(\lambda) \\ &\quad -\tfrac{44}{35}S_2^\mu(\lambda) - \tfrac{1}{4}S_3^\mu(\lambda) + \tfrac{2}{35}S_4^\mu(\lambda), \end{aligned}$$

where $\mu = \frac{1}{5}$. To compute the monic gcd of two polynomials $f(\lambda)$ and $h(\lambda)$ in the corresponding Legendre-Sobolev basis, one can obtain the following matrices:

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 & 0 \\ 0 & \frac{5}{8} & 0 & \frac{3}{8} & 0 \\ -\frac{1}{5} & 0 & \frac{34}{35} & 0 & \frac{8}{35} \\ -\frac{287}{3} & \frac{181}{4} & \frac{194}{3} & -\frac{37}{4} & -4 \end{bmatrix},$$

$$h(B) = \begin{bmatrix} -\frac{14}{5} & \frac{17}{4} & -\frac{44}{35} & -\frac{1}{4} & \frac{2}{35} \\ -4 & -1 & \frac{44}{7} & -1 & -\frac{2}{7} \\ \frac{211}{5} & -\frac{173}{8} & -\frac{1034}{35} & \frac{61}{8} & \frac{47}{35} \\ -\frac{1796}{5} & 227 & \frac{1012}{5} & -61 & -\frac{46}{5} \\ \frac{20266}{5} & -\frac{10975}{4} & -\frac{72644}{35} & \frac{2687}{4} & \frac{3302}{35} \end{bmatrix}.$$

The rank of matrix $h(B)$ is three. Thus $\gcd(f,h)$ has degree two. By solving the Diophantine equations in Equation (8), one can compute $\zeta_0 = 2$, $\zeta_1 = -\frac{9}{4}$, $\zeta_2 = 0$, and $\zeta_3 = \frac{1}{4}$. Thus

$$\gcd(f,h) = 2\,S_0^\mu(\lambda) - \frac{9}{4}\,S_1^\mu(\lambda) + \frac{1}{4}\,S_3^\mu(\lambda) = (\lambda+2)(\lambda-1)^2.$$

IV. Feature extraction for handwritten Symbols

As we explained in Section I, one of the main concerns in the problem of mathematical handwriting recognition is the two-dimensionality nature of this problem. Having handwritten characters represented as parametrized curves in an orthogonal basis, one can obtain interesting geometrical insights about each handwritten character, and therefore, determine features that make in-lined expressions distinct from subscripts and superscripts. One can name the baseline, loops, and cusps of handwritten characters as some of the examples of such interesting geometrical features.

Mathematically, computation of the baseline on a handwritten character is equivalent to finding the points on the approximated parametrized curve of a handwriten character, where the slope of the tangent line is zero. Furthermore, the points on the approximated parametrized curve, corresponding to the ones with their slope having infinity tangent line, give bounding lines for the width of a handwritten character. Thus, one can use the points on the approximated curve with $X'(\lambda) = 0$ or $Y'(\lambda) = 0$ to find the baseline and bounding box of a handwritten character. Additionally, the number of such points and the order of them, based on their arc length, gives useful information about each handwritten character. Another interesting feature that can be captured is self-intersection points or more precisely, loops. Self-intersection points are in fact singular points corresponding to handwritten curves. Thus one might try to compute such points by finding values of λ for which $\gcd(X'(\lambda), Y'(\lambda)) = 0$. The problem with this method is that a parametrization of a curve might not capture all of the singular points. An alternative solution to compute self-intersection points is to solve the system $\{X(\lambda_1) = X(\lambda_2), Y(\lambda_1) = Y(\lambda_2)\}$, for distinct values of λ_1 and λ_2. The question here then is how to find self-intersection points in a Legendre-Sobolev basis efficiently and accurately.

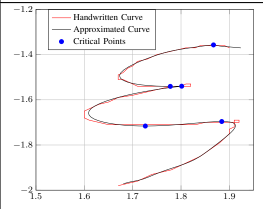
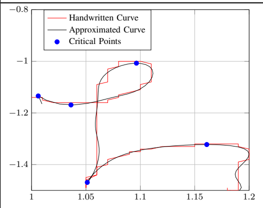
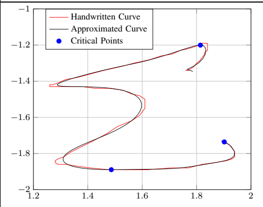
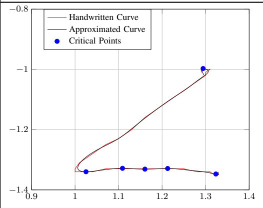
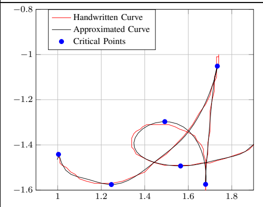
We have computed some of the features that were explained earlier for a handwritten character "m". Figure 3 shows the handwritten character "m" with its approximation by degree 18 parametrized coordinate curves $X(\lambda)$ and $Y(\lambda)$ in the Legendre-Sobolev basis with $\mu = \frac{1}{8}$. This approximation is computed by multiplying a (precomputed) matrix and the vector of moment integrals [2]. We have marked up some red and green points on the approximated curve which are corresponding to the points with $Y'(\lambda) = 0$ and $X'(\lambda) = 0$, respectively. To compute these points, we have first used Theorem 2 to compute the derivatives and then Theorem 4 to compute the roots of the derivatives of $X(\lambda)$ and $Y(\lambda)$. These calculations are done in the Legendre-Sobolev basis without converting to the monomial basis. These points demonstrate that letter "m" has two singular points (cusps) and five local maximum or minimum.

Figure 4 shows the backward error analysis for computing the roots of $Y'(\lambda) = 0$ via computing the eigenvalues of matrix B, where $Y(\lambda)$ is the approximated coordinate curve of the handwritten character "m" shown in Figure 3. The approximation $Y(\lambda)$ is computed in Legendre-Sobolev bases and w.r.t different values of μ. This figure shows that one might double the precision that was used for calculation w.r.t $\mu = \frac{1}{8}$ to obtain the same accuracy with $\mu = \frac{1}{2}$ for computing the eigenvalues of matrix B. Furthermore, Figure 5 demonstrates the effect of having different precisions for calculation of critical points on the average residual at potential real roots of $Y'(\lambda) = 0$, where $Y(\lambda)$ is the y-coordinate curve corresponding to the handwritten character "m" given in Figure 3. Note that $Y(\lambda)$ is approximated in Legendre-Sobolev basis and w.r.t different values of μ such that $0 \le \mu \le 1$.

Table I shows Legendre-Sobolev approximations of different handwritten characters and their corresponding critical points. The calculation of the approximations and critical points is done with double and quadruple precision, respectively. Furthermore, m is the number of real values of ξ for which $Y'(\xi) = 0$.

Since matrix B is lower Hessenberg, the eigenvalues of this matrix are computed by QR algorithm which is backward stable for such matrices (see [9], for a more detailed discussion).

TABLE I
LEGENDRE-SOBOLEV APPROXIMATIONS OF HANDWRITTEN CURVES WITH $d = 18$ AND $\mu = \frac{1}{8}$, AND THEIR CRITICAL POINTS

Handwritten Curve	$\|\Delta Y(\lambda)\|_2$ Digits = 17	$\frac{\sum_{i=1}^{m} \|Y'(\xi_i)\|}{m}$ Digits = 34
	0.0005	2.359×10^{-18}
	0.0008	3.893×10^{-19}
	0.0002	2.344×10^{-18}
	0.0006	1.55×10^{-19}
	0.0008	4.471×10^{-19}

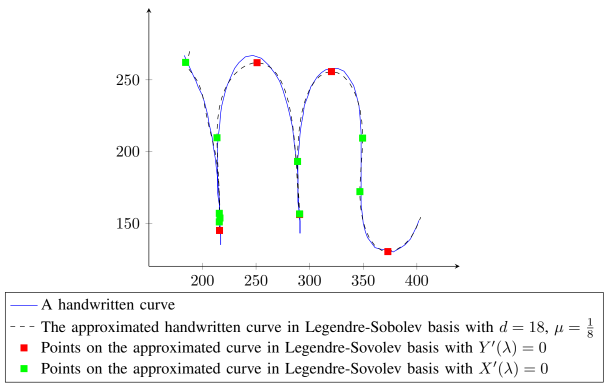


Fig. 3. Approximated handwritten curve in Legendre-Sobolev basis with points whose tangent lines have slopes equal to either zero or infinity. The approximation and calculation of such points is done with double precision.

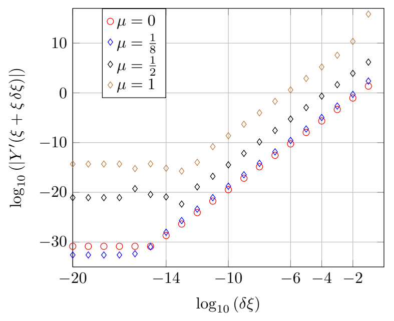


Fig. 4. The value of $Y'(\xi + \xi\delta\xi)$ is shown on a based 10 logarithmic scale, where ξ is one of the exact roots of $Y'(\lambda) = 0$, $\delta\xi$ is the relative error corresponding to ξ, and $Y(\lambda)$ is the Legendre-Sobolev approximation of degree 18 of the letter "m" in Figure 3 and w.r.t different values of μ. Here, all the calculations are done in quadruple precision.

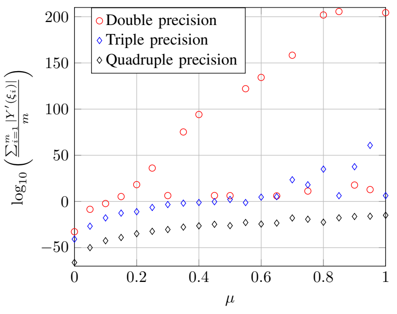


Fig. 5. The value of $\frac{\sum_{i=1}^{m} |Y'(\xi_i)|}{m}$ is shown on a based 10 logarithmic scale w.r.t different values of parameter μ (on the x-axis), where m is the number of total real roots of $Y'(\lambda) = 0$, ξ_i is one of the real roots of $Y'(\lambda) = 0$, and $Y(\lambda)$ is the Legendre-Sobolev approximation of degree 18 w.r.t parameter μ of letter "m" in Figure 3. Here, the calculations are done in double, triple, and quadruple precision.

V. Conclusion and Future Work

In this paper, we presented methods for computing the derivatives, roots, and gcd of polynomials in Legendre-Sobolev bases. The main goal is to provide sufficient tools to extract certain features from Legendre-Sobolev approximations without converting the approximations to the monomial basis.

While these features can be used to distinguish the relative positioning of different characters, it would also be interesting to see how these features work with coefficients of Legendre-Sobolev approximations to have higher recognition rates. Especially, with the growth of the power of neural networks, that would be interesting to see how different variations of neural networks work with the problem of mathematical handwriting recognition, where handwritten characters are represented as coefficients of parametrized curves in an orthogonal basis.

References

[1] P. Althammer. Eine Erweiterung des Orthogonalitätsbegriffes bei Polynomen und deren Anwendung auf die beste approximation. *J. Reine Ang. Math.*, 211:192–204, 1962.

[2] P. Alvandi and S. M. Watt. Real-Time Computation of Legendre-Sobolev Approximations. In *SYNASC*, pages 67–74, 2018.

[3] L. Anthony, J. Yang, and K. R. Koedinger. A paradigm for handwriting-based intelligent tutors. *International Journal of Human-Computer Studies*, 70(11):866 – 887, 2012.

[4] Y. Bengio, Y. LeCun, C. Nohl, and C. Burges. LeRec: A NN/HMM Hybrid for On-Line Handwriting Recognition. *Neural Computation*, 7(6):1289–1303, Nov 1995.

[5] F. W. Blackwell and R. H. Anderson. An On-line Symbolic Mathematics System Using Hand-printed Two-dimensional Notation. In *ACM*, pages 551–557, 1969.

[6] V. Carbune, P. Gonnet, T. Deselaers, H. A. Rowley, A. N. Daryin, M. Calvo, L. Wang, D. Keysers, S. Feuz, and P. Gervais. Fast Multi-language LSTM-based Online Handwriting Recognition. *ArXiv e-prints*, 2019.

[7] B. W. Char and S. M. Watt. Representing and Characterizing Handwritten Mathematical Symbols through Succinct Functional Approximation. In *ICDAR*, pages 1198–1202, 2007.

[8] S. D. Connell and A. K. Jain. Template-based online character recognition. *Pattern Recognition*, 34(1):1–14, 2001.

[9] D. M. Day and L. A. Romero. Roots of Polynomials Expressed in Terms of Orthogonal Polynomials. *SIAM J. Numerical Analysis*, 43(5):1969–1987, 2005.

[10] M. Franzini, K. Lee, and A. Waibel. Connectionist Viterbi training: a new hybrid method for continuous speech recognition. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 425–428, 1990.

[11] V. Frinken and S. Uchida. Deep BLSTM neural networks for unconstrained continuous handwritten text recognition. In *ICDAR*, pages 911–915, 2015.

[12] O. Golubitsky and S. M. Watt. Online stroke modeling for handwriting recognition. In *CASCON*, pages 72–80, 2008.

[13] O. Golubitsky and S. M. Watt. Online computation of similarity between handwritten characters. In *Document Recognition and Retrieval XVI, part of the IS&T-SPIE Electronic Imaging Symposium*, pages C1–C10, 2009.

[14] A. Graves, S. Fernández, M. Liwicki, H. Bunke, and J. Schmidhuber. Unconstrained Online Handwriting Recognition with Recurrent Neural Networks. In *NIPS*, pages 577–584, 2007.

[15] M. Harouni, D. Mohamad, and A. Rasouli. Deductive method for recognition of on-line handwritten Persian/Arabic characters. In *ICCAE*, volume 5, pages 791–795, 2010.

[16] J. Hu, M. K. Brown, and W. Turin. HMM based online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, 1996.

[17] R. Hu and S. M. Watt. Optimization of Point Selection on Digital Ink Curves. In *ICFHR*, pages 527–532, 2012.

[18] R. Hu and S. M. Watt. Determining Points on Handwritten Mathematical Symbols. *ArXiv e-prints*, 2013.

[19] R. Hu and S. M. Watt. Identifying Features via Homotopy on Handwritten Mathematical Symbols. In *SYNASC*, pages 61–67, 2013.

[20] S. Jäger, S. Manke, J. Reichert, and A. H. Waibel. Online handwriting recognition: the NPen++ recognizer. *International Journal on Document Analysis and Recognition*, 3:169–180, 2001.

[21] Y. Jiang, F. Tian, H. Wang, X. Zhang, X. Wang, and G. Dai. Intelligent Understanding of Handwritten Geometry Theorem Proving. In *IUI*, pages 119–128, New York, NY, USA, 2010. ACM.

[22] J. J. LaViola, Jr. and R. C. Zeleznik. MathPad2: A System for the Creation and Exploration of Mathematical Sketches. *ACM Trans. Graph.*, 23(3):432–440, 2004.

[23] W. Lee, R. de Silva, E. Jeffrey Peterson, R. C. Calfee, and T. F. Stahovich. Newton's Pen - A Pen-based Tutoring System for Statics. In *SBM*, 2007.

[24] John Maroulas and Stephen Barnett. Greatest common divisor of generalized polynomials and polynomial matrices. *Linear Algebra and its Applications*, 22:195 – 210, 1978.

[25] H. Mouchre, C. Viard-Gaudin, R. Zanibbi, and U. Garain. ICFHR 2014 Competition on Recognition of On-Line Handwritten Mathematical Expressions (CROHME 2014). In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 791–796, Sep. 2014.

[26] A. B. Owen. On L^2 norms of derivatives of orthogonal polynomials with respect to Sobolev inner products. Technical report, 2017.

[27] J. A. Pittman. Handwriting Recognition: Tablet PC Text Input. *Computer*, 40(9):49–54, Sep. 2007.

[28] R. Plamondon and S. N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, Jan 2000.

[29] E. M. Taranta, II, A. N. Vargas, S. P. Compton, and J. J. Laviola, Jr. A Dynamic Pen-Based Interface for Writing and Editing Complex Mathematical Expressions With Math Boxes. *ACM Trans. Interact. Intell. Syst.*, 6(2):13:1–13:25, July 2016.

[30] I. Velázquez. Metrics and neatening of handwritten characters. Master's thesis, University of Western Ontario, Canada, 2010.

[31] S. M. Watt and T. Underhill. Ink markup language (InkML). W3C recommendation, W3C, September 2011. http://www.w3.org/TR/2011/REC-InkML-20110920/.

[32] L. S. Yaeger, B. J. Webb, and R. F. Lyon. Combining Neural Networks and Context-Driven Search for On-Line, Printed Handwriting Recognition in the Newton. In *Neural Networks: Tricks of the Trade*, 1996.

[33] R. Zanibbi, K. L. Novins, J. Arvo, and K. Zanibbi. Aiding Manipulation of Handwritten Mathematical Expressions through Style-Preserving Morphs. In *Graphics Interface*, 2001.