Real-Time Computation of Legendre-Sobolev Approximations

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*—The present work is motivated by the problem of mathematical handwriting recognition where symbols are represented as plane curves, $(X(\lambda), Y(\lambda))$ parameterized by arc length $\lambda \in [0, L]$. Earlier work has shown that approximating the coordinate functions as certain truncated orthogonal polynomial series yields fast and effective recognition. It has been previously shown how to compute Legendre series representation in real time, as the curve is being traced out. In this article we show how to compute Legendre-Sobolev series representation in real time. The idea is to numerically integrate the moments of the coordinate functions as the curve is being traced. We show how the Legendre-Sobolev coefficients may be constructed either from the Legendre series coefficients or directly from the moments. Computing via Legendre series coefficients requires two matrix vector products, while the direct method requires only one.**

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

I. Introduction

Among different methods for handwriting recognition, we are mainly interested in online methods in which recognition has to be done in real time. In such methods, recognition can not happen until the process of writing of one stroke is complete. Since we are looking for online recognition methods, it is important to use the power of available processors in our recognition devices, such as pen-based tablets, telephones, whiteboards and PCs, doing calculations as strokes are written rather than waiting until they are complete.

Most of the techniques for handwriting recognition including the ones based on elastic matching [9], [12], [10], need each character to be traced out completely, before any analysis can occur for recognition of the corresponding character. That means one keeps processors idle while characters are being traced out and that causes delay in recognition process.

In [2], the authors propose truncated orthognonal polynomial series representation of parametric plane curves for handwriting recognition. In [3], it is shown how to construct Legendre series coefficients from moments that are computed in real time as the curve is being written. The main point is to avoid having to wait until pen up (when the arc length is known) to compute the functional inner product. The authors of [3] also report on the complexity of their method followed by experimental results that show this representation of coordinate curves is as good as that obtained by earlier stroke-at-once methods. In fact, this method requires constant number of post-pen-up operations to compute the coefficients of coordinate curves in Legendre basis.

Representing parametrized coordinate curves $X(\lambda)$ and $Y(\lambda)$ in orthogonal basis not only helps in establishing online handwriting recognition methods but it helps in analyzing the geometry of each handwritten character, like finding the extremum points of each character. Such analysis becomes important in mathematical handwriting recognition, when the characters appearing in a math formula have varying semantically meaningful baselines [7], [8].

While representing coordinate curves of characters in Legendre basis helps in developing online handwriting recognition methods, authors of [4] have reported experimental results that demonstrate that representing coordinate curves in Legendre-Sobolev basis has higher detection rates compared to when these curves are represented in Legendre basis.

We therefore require an algorithim to calculate in real-time the representation of curves in a Legendre-Sobolev basis. This is the goal of the present paper. We do this by computing matrix M, as in Proposition 2, which leads to a method for computing Legendre-Sobolev coefficients from moment integrals by a matrix multiplication.

Note that for parametrized coordinate curves $X(\lambda)$ and $Y(\lambda)$ in any orthogonal basis, there are three different parametrizations that arise naturally: parametrization by time, by arc length, and by affine arc length. The last two parametrizations depend only on the curve itself, while the first one also depends on the speed with which the handwritten character is being written. That might explain why parametrizations by arc length or affine arc length yield better recognition rates.

In [4], it is shown that Legendre-Sobolev yields similar results for recognition rates with respect to both arc length and affine arc length, while for elastic matching arc length parametrization is better than the rest. In this paper, we only use parametrization by arc length.

This paper is organized as follows. After presenting preliminaries in Section II, we explain how one can compute matrix M, which was mentioned earlier, in Section III. Then Section IV investigates condition number of matrix M. Section V is dedicated to the numerical method we have used in this paper for computing moment integrals. Finally, Section VI applies experimentation around the ideas presented here.

II. Preliminaries

Earlier work [2], [3] has shown that the coordinate curves $X(\lambda)$ and $Y(\lambda)$ for handwritten characters can be modeled by truncated orthogonal series and the series coefficients can be used for classification and therefore recognition of characters. The parameter λ can be chosen to be either time or length of the handwritten curve. In this paper, we assume that sample values of $X(\lambda)$ and $Y(\lambda)$ are received as a real time signal and λ is corresponding to the length of handwritten curves. We use these values to compute "moment integrals" and from moment integrals we approximate the coefficients of curves $X(\lambda)$ and $Y(\lambda)$ in an orthogonal basis. We present several basic notions and ideas that are used throughout this paper.

A. Series of orthogonal functions

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

$$\langle f(\lambda), g(\lambda)\rangle_{\mathrm{L}} := \int_{-1}^{1} f(\lambda)\, g(\lambda) d\lambda. \tag{1}$$

We also consider the following inner product which is given by

$$\langle f(\lambda), g(\lambda)\rangle_{\mathrm{LS}} := \int_{-1}^{1} f(\lambda)\, g(\lambda) d\lambda + \mu \int_{-1}^{1} f'(\lambda) g'(\lambda) d\lambda \tag{2}$$

where $\mu \in \mathbb{R}_{>0}$. The latter is a special case of Legendre-Sobolev inner product. In fact, a Legendre-Sobolev inner product may involve terms corresponding to higher order derivatives, but for the purpose of this paper we restrict ourselves to the first order derivatives. Systems of orthogonal polynomials $\{H_0, H_1, H_2, \ldots\}$ corresponding to either of the above inner product can be computed by Gram-Schmidt orthogonalization to the monomial basis $\{1, \lambda, \lambda^2, \ldots\}$. We denote the Legendre-Sobolev polynomials corresponding to the inner product given by Equation (2) of degree n by $S_n^{\mu}(\lambda)$ or $S_n(\lambda)$, whenever there is no ambiguity. We also denote Legendre polynomials of degree n by $P_n(\lambda)$. These have been studied by Althammer [1] and are thus sometimes given that name.

A function $f : [-1, 1] \to \mathbb{R}$ (under some assumptions) can be represented by an infinite linear combination of the orthogonal polynomials $\{H_0(\lambda), H_1(\lambda), H_2(\lambda), \ldots\}$ as

$$f(\lambda) = \sum_{i=0}^{\infty} \alpha_i H_i(\lambda)$$

where $\{H_0(\lambda), H_1(\lambda), H_2(\lambda), \ldots\}$ is the orthogonal basis corresponding to either Legendre or Legendre-Sobolev inner product. The coefficients of the series in the new basis can be computed by

$$\alpha_i := \frac{\langle f(\lambda), H_i(\lambda)\rangle}{\langle H_i(\lambda), H_i(\lambda)\rangle} \ , \ i = 0, 1, 2, \ldots$$

where $\langle ., . \rangle$ stands for either Legendre or Legender-Sobolev inner product, and it is also assumed that the integrals involved in the inner product are well-defined for f.

Note that the closest polynomial of degree d to the function f with respect to Euclidean norm induced by the given inner product is given by the truncated series

$$f(\lambda) \simeq \sum_{i=0}^{d} \alpha_i H_i(\lambda).$$

Such an approximation allows one to think of functions as points $(\alpha_0, \alpha_1, \ldots, \alpha_d)$ in an $(d+1)$-dimensional vector space. The variational integral of the square distance between two curves is then given by the Euclidean norm in this vector space. We can measure how close two functions are to each other in this vector space as follows:

$$f \simeq \sum_{i=0}^{d} \alpha_i H_i(\lambda) \ , \ g \simeq \sum_{i=0}^{d} \beta_i H_i(\lambda),$$

then one can measure how close f, g are by computing:

$$\| f - g \| \simeq \sqrt{\sum_{i=0}^{d} (\alpha_i - \beta_i)^2}.$$

This method of measuring the distance of two functions is very important in hand-writing recognition method used in [4].

B. Polynomial norms

In this paper, we use three different polynomial norms to compute error rates in approximated polynomials. We use Legendre and Legendre-Sobolev norms which are induced by the inner product given by Equations (1) and (2), respectively. We also use max norm that is given by

$$\|f(\lambda)\|_{max} := \max_{a \le \lambda \le b} (|f(\lambda)|),$$

where $f(\lambda) := \sum_{i=0}^{d} \alpha_i \lambda^i$, and $\lambda \in [a, b]$.

C. Interpolating coordinate curves $X(\lambda)$ and $Y(\lambda)$

The moments of a function f defined on the interval $[a, b]$ are the integrals:

$$\int_a^b \lambda^k\, f(\lambda) d\lambda.$$

A key aspect of the approach used in [2] for the purpose of interpolating the coordinate curves $X(\lambda)$ and $Y(\lambda)$ corresponding to handwritten strokes is to recover these curves from their moments. This is the Hausdorff moment problem [5], [6], known to be ill-conditioned. For the purpose of this paper, the moments of a function f are defined over an unbounded half-line since the curve may be traced over an arbitrary length:

$$m_k(f, \ell) := \int_0^{\ell} \lambda^k\, f(\lambda) d\lambda.$$

In our application, we assume that discrete sample values of f are received as a real-time signal. We use these values to compute approximate values for the moment integrals. After a curve is traced out, we will have computed its moments over some length L, with L known only at the time the pen is lifted. The problem is now to scale L to a standard interval and compute the truncated Legendre series coefficients for the scaled function from the moments of the unscaled function.

Proposition 1 (See Section 6 in [3])**.** *Suppose that $m_k(f(\lambda), L)$ is defined as $\int_0^L f(\lambda)\lambda^k d\lambda$ where L is the length of a given curve and $f(\lambda)$ is either $X(\lambda)$ or $Y(\lambda)$, for $k = 0, \ldots, d$. Let also $\hat{f}(\lambda) = \sum_{k=0}^{d} \alpha_k P_k(\lambda)$ be the corresponding scaled function of $f(\lambda)$ in the interval $[-1, 1]$. Then for $k = 0, \ldots, d$, we can compute α_k as follows:*

$$\alpha_k = (-1)^k \frac{2k+1}{L} \sum_{i=0}^{k} (\frac{-1}{L})^i \binom{k}{i} \binom{k+i}{i} m_i(f, L). \quad (3)$$

Note that the coefficients $(-1)^i \binom{k}{i}\binom{k+i}{i}$ are independent of the problem and may be computed as constants, in advance. Given the first k moments of f and the first $k-1$ powers of L, we may compute α and L^k in a number of arithmetic operations depending only on k (see Section 6 in [3]).

When the last point arrives and pen is lifted up, we apply linear substitution $[0, L] \to [-1, 1]$ to re-scale the moments to the interval $[-1, 1]$ and change the basis from the monomial basis $\{1, \lambda, \lambda^2, \ldots, \lambda^d\}$ to the orthogonal polynomial basis $\{1, P_1(\lambda), P_2(\lambda), \ldots, P_d(\lambda)\}$.

III. Methods

Proposition 1 shows how one can compute the coefficients of coordinate curves $X(\lambda)$ and $Y(\lambda)$ of handwritten characters in the Legendre basis. The goal of this section is to show how to compute such coefficients in a Legendre-Sobolev basis using moment integrals. To this end, we first compute the conversion matrix M_2 where

$$\left[\begin{array}{ccc} P_0(\lambda) & \cdots & P_n(\lambda) \end{array}\right]^{\mathbf{T}} = M_2 \left[\begin{array}{ccc} S_0^\mu(\lambda) & \cdots & S_n^\mu(\lambda) \end{array}\right]^{\mathbf{T}}.$$

Then we compute matrix M from M_2 such that

$$\left[\begin{array}{ccc} \beta_0 & \cdots & \beta_d \end{array}\right] = \left[\begin{array}{ccc} m_0(f, L) & \cdots & m_d(f, L) \end{array}\right] D\, M, \quad (4)$$

where d is the degree, $\beta_0, \beta_1, \ldots, \beta_d$ are the coefficients and f is one of the coordinate curves in Legendre-Sobolev basis and D is a diagonal matrix which will be only known at the time the pen is lifted. Knowing the matrices M and D and also moment integrals, one can compute the coefficients of coordinate curves $X(\lambda)$ and $Y(\lambda)$ in the Legendre-Sobolev basis.

According to Equation (1.7) in [11], for $n \geq 0$ and $\mu \geq 0$, we have

$$S_n^\mu(\lambda) = \sum_{k=0}^{\lfloor \frac{n}{2} \rfloor} a_{n-2k}(\mu) \left(P_{n-2k}(\lambda) - P_{n-2k-2}(\lambda)\right), \quad (5)$$

where $P_{-2}(\lambda) = P_{-1}(\lambda) = 0$, and

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

In the following theorem, we show how one can compute matrix N^μ such that

$$\left[\begin{array}{ccc} S_0^\mu(\lambda) & \cdots & S_n^\mu(\lambda) \end{array}\right]^{\mathbf{T}} = N^\mu \left[\begin{array}{ccc} P_0(\lambda) & \cdots & P_n(\lambda) \end{array}\right]^{\mathbf{T}}.$$

Note that N^μ is a matrix whose entries are polynomials in μ.

Theorem 1. *For $n \geq 1$ and $\mu \geq 0$,*

$$S_n^\mu(\lambda) = a_n(\mu) P_n(\lambda) + \sum_{k=1}^{\lfloor \frac{n-1}{2} \rfloor} (a_n(\mu) - a_{n+2}(\mu)) P_{n-2k}(\lambda). \quad (6)$$

Proof. We consider two different cases; first let n be equal to $2m+1$ for some $m \in \mathbb{Z}_{>0}$. Then,

$$\begin{aligned} S_{2m+1}(\lambda) &= \sum_{k=0}^{m} a_{2m-2k+1}(\mu)(P_{2m-2k+1}(\lambda) - P_{2m-2k-1}(\lambda)) \\ &= a_{2m+1}(\mu)(P_{2m+1}(\lambda) - P_{2m-1}(\lambda)) + \\ &\quad a_{2m-1}(\mu)(P_{2m-1}(\lambda) - P_{2m-3}(\lambda)) + \\ &\quad \vdots \\ &\quad + a_3(\mu)(P_3(\lambda) - P_1(\lambda)) + \\ &\quad a_1(\mu)(P_1(\lambda) - P_{-1}(\lambda)). \end{aligned} \quad (7)$$

Since $P_{-1}(\lambda)$ is defined to be zero, then we can rewrite Equation (7) as follows:

$$\begin{aligned} S_{2m+1}(\lambda) = \; & a_{2m+1}(\mu) P_{2m+1}(\lambda) + \\ & \sum_{k=0}^{m-1} \left(a_{2m-2k-1}(\mu) - a_{2m-2k+1}(\mu)\right) P_{2m-2k-1}(\lambda). \end{aligned} \quad (8)$$

Now let n be of the form $2m$ for $m \in \mathbb{Z}_{>0}$. Then,

$$\begin{aligned} S_{2m}(\lambda) &= S_0 + \textstyle\sum_{k=0}^{m-1} a_{2m-2k}(\mu) \times \\ &\quad (P_{2m-2k}(\lambda) - P_{2m-2k-2}(\lambda)) \\ &= a_{2m}(\mu)(P_{2m}(\lambda) - P_{2m-2}(\lambda)) + \\ &\quad a_{2m-2}(\mu)(P_{2m-2}(\lambda) - P_{2m-4}(\lambda)) + \\ &\quad \vdots \\ &\quad + a_3(\mu)(P_4(\lambda) - P_2(\lambda)) + \\ &\quad a_1(\mu)(P_2(\lambda) - P_0(\lambda)). \end{aligned} \quad (9)$$

Since $S_0(\lambda) = P_0(\lambda)$, we can rewrite Equation (9) as follows:

$$\begin{aligned} S_{2m}(\lambda) = \; & a_{2m}(\mu) P_{2m}(\lambda) + \\ & \sum_{k=0}^{m-2} P_{2m-2k-2}(\lambda) \left(a_{2m-2k-2}(\mu) - a_{2m-2k}(\mu)\right). \end{aligned} \quad (10)$$

Based on Equations (8) and (10), we obtain

$$S_n(\lambda) = a_n(\mu) P_n(\lambda) + \textstyle\sum_{k=1}^{\lfloor \frac{n-1}{2} \rfloor} (a_n(\mu) - a_{n+2}(\mu)) P_{n-2k}(\lambda).$$

□

Based on Equation (6), we have

$$\left[\begin{array}{ccc} S_0^\mu(\lambda) & \cdots & S_n^\mu(\lambda) \end{array}\right]^{\mathbf{T}} = N^\mu \left[\begin{array}{ccc} P_0(\lambda) & \cdots & P_n(\lambda) \end{array}\right]^{\mathbf{T}},$$

where N^μ is a $(n+1) \times (n+1)$ lower triangular matrix whose main diagonal is formed by $a_k(\mu)$ and other entries are formed by either 0 or $a_\ell(\mu) - a_{\ell+2}(\mu)$, for $k = 0, \ldots, n$

and $\ell = 1, \ldots, n-2$. We use N instead of N^{μ} whenever there is no ambiguity. For arbitrary n, matrix N can be formulated as follows, for $i, j = 1, \ldots, n+1$:

$$[N]_{i,j} = \begin{cases} 0 & \text{if } i < j \\ a_{i-1}(\mu) & \text{if } i = j \\ a_{i-2\ell-1}(\mu) - a_{i-2\ell+1}(\mu) = & \text{if } j = i - 2\ell, \text{ for} \\ a_{j-1}(\mu) - a_{j+1}(\mu) & \ell = 1, \ldots, \lfloor \frac{i-1}{2} \rfloor \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

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

$$\begin{bmatrix} P_0(\lambda) & \cdots & P_n(\lambda) \end{bmatrix}^{\boldsymbol{T}} = M_2 \begin{bmatrix} S_0(\lambda) & \cdots & S_n(\lambda) \end{bmatrix}^{\boldsymbol{T}},$$

where for $i, j = 1, \ldots, n+1$,

$$[M_2]_{i,j} = \begin{cases} 0 & \textit{if } i < j \\ \frac{1}{a_{i-1}(\mu)} & \textit{if } i = j \\ \frac{1}{a_{i-2\ell-1}(\mu)} - \frac{1}{a_{i-2\ell+1}(\mu)} = & \textit{if } j = i - 2\ell, \textit{ for} \\ \frac{1}{a_{j-1}(\mu)} - \frac{1}{a_{j+1}(\mu)} & \ell = 1, \ldots, \lfloor \frac{i-1}{2} \rfloor \\ 0 & \textit{otherwise} \end{cases} \tag{12}$$

Proof. First, note that $M_2 = N^{-1}$. The formula for computing the inverse of any invertible lower triangular matrix N is:

$$N^{-1}_{i,j} = \begin{cases} 0 & \text{for } i < j \\ \frac{1}{N_{i,i}} & \text{for } i = j \\ -\frac{1}{N_{i,i}} \sum_{k=j}^{i-1} N_{i,k} N^{-1}_{k,j} & \text{for } i > j \end{cases} \tag{13}$$

Using the formula in Equation (13), we compute the inverse of matrix N given by Equation (11). When $i < j$ or $i = j$ then the results follow immediately based on Equation (13). Suppose that $i > j$. We prove by induction, that the inverse matrix of N can be computed as Equation (12). The induction is done over ℓ, where ℓ corresponds to $(2\,\ell - 1)$th and $(2\,\ell)$th lower diagonals of matrix M_2, for $\ell = 1, \ldots, \lfloor \frac{n}{2} \rfloor$.

Basic step: We first compute the entries of $[M_2]_{i,j}$ that belong to the first lower diagonal of matrix M_2 where $j = i-1$. Based on Formula (13), we have

$$\begin{aligned} [M_2]_{i,i-1} &= \tfrac{-1}{[N]_{i,i}} [N]_{i,i-1} [M_2]_{i-1,i-1} \\ &= -\tfrac{1}{a_{i-1}(\mu)} (0 \times \tfrac{1}{a_{i-2}(\mu)}) = 0. \end{aligned}$$

Now, let $j = i-2$; then again, one can obtain the following results for entries of M_2 on the second lower diagonal:

$$\begin{aligned} [M_2]_{i,i-2} &= \tfrac{-1}{[N]_{i,i}} \left([N]_{i,i-2} [M_2]_{i-2,i-2} + \right. \\ &\quad \left. [N]_{i,i-1} [M_2]_{i-1,i-2} \right) \\ &= -\tfrac{1}{a_{i-1}(\mu)} ((a_{i-3}(\mu) - a_{i-1}(\mu)) \times \tfrac{1}{a_{i-3}(\mu)} + \\ &\quad 0 \times 0) \\ &= \tfrac{1}{a_{i-3}(\mu)} - \tfrac{1}{a_{i-1}(\mu)}. \end{aligned}$$

Induction step: Suppose that for $\ell = 1, \ldots, \ell'$, with $\ell' \in \mathbb{N}$, we have $[M_2]_{i,j} = \frac{1}{a_{j-1}(\mu)} - \frac{1}{a_{j+1}(\mu)}$ when $j = i - 2\,\ell$ and $[M]_{i,j} = 0$ when $j = i - 2\,\ell + 1$. Now let $j = i - 2\,\ell' - 1$, then we have

$$\begin{aligned} [M_2]_{i,i-2\,\ell'-1} &= \tfrac{-1}{[N]_{i,i}} \left(\sum_{\ell=1}^{\ell'} [N]_{i,i-2\,\ell} [M_2]_{i-2\,\ell,i-2\,\ell'-1} + \right. \\ &\quad \left. \sum_{\ell=0}^{\ell'} [N]_{i,i-2\,\ell-1} [M_2]_{i-2\,\ell-1,i-2\,\ell'-1} \right). \end{aligned}$$

Since $[N]_{i,i-2\,\ell-1} = 0$, for $\ell = 0 \ldots, \ell'$, thus

$$[M_2]_{i,i-2\,\ell'-1} = -\frac{1}{[N]_{i,i}} \left(\sum_{\ell=1}^{\ell'} [N]_{i,i-2\,\ell} [M_2]_{i-2\,\ell,i-2\,\ell'-1} \right).$$

Now let $j' := i - 2\,\ell' - 1$ and $i'_\ell = i - 2\,\ell$, for $\ell = 1 \ldots, \ell'$. We want to show that $[M_2]_{i'_\ell,j'} = 0$, for all $\ell = 1 \ldots, \ell'$. In fact, one can show easily that $j' = i'_\ell - 2\,k_\ell + 1$ where $k_\ell := \ell' - \ell + 1$, for $\ell = 1, \ldots, \ell'$. Note that k_ℓ takes its values from the set $\{\ell', \ell' - 1, \ldots, 1\}$, respectively, for $\ell = 1, \ldots, \ell'$. By induction hypothesis, we know that $[M_2]_{i'_\ell,j'} = 0$. Thus

$$[M_2]_{i,i-2\,\ell'-1} = -\frac{1}{[N]_{i,i}} \left(\sum_{\ell=1}^{\ell'} [N]_{i,i'_\ell} [M_2]_{i'_\ell,j'} \right) = 0.$$

To complete the proof, we need to prove that when $j = i - 2\,\ell' - 2$, then $[M_2]_{i,i-2\,\ell'-2} = \frac{1}{a_{i-2\,\ell'-3}(\mu)} - \frac{1}{a_{i-2\,\ell'-1}(\mu)}$.

$$\begin{aligned} [M_2]_{i,i-2\,\ell'-2} &= \tfrac{-1}{[N]_{i,i}} \left(\sum_{\ell=1}^{\ell'+1} [N]_{i,i-2\,\ell} [M_2]_{i-2\,\ell,i-2\,\ell'-2} + \right. \\ &\quad \left. \sum_{\ell=0}^{\ell'} ([N]_{i,i-2\,\ell-1} [M_2]_{i-2\,\ell-1,i-2\,\ell'-2} \right) \\ &= \tfrac{-1}{a_{i-1}(\mu)} \left(\sum_{\ell=1}^{\ell'} [N]_{i,i-2\,\ell} [M_2]_{i-2\,\ell,i-2\,\ell'-2} + \right. \\ &\quad \left. (a_{i-2\,\ell'-3}(\mu) - a_{i-2\,\ell'-1}(\mu)) \tfrac{1}{a_{i-2\,\ell'-3}(\mu)} \right). \end{aligned}$$

Now let $j' := i - 2\,\ell' - 2$ and $i'_\ell = i - 2\,\ell$, for $\ell = 1 \ldots, \ell'$. Again, one can easily show that $j' = i'_\ell - 2\,k_\ell$ where $k_\ell := \ell' - \ell + 1$, for $\ell = 1, \ldots, \ell'$. Note that k_ℓ takes its values from the set $\{\ell', \ell' - 1, \ldots, 1\}$, respectively, for $\ell = 1, \ldots, \ell'$. By induction hypothesis, we know that $[M_2]_{i'_\ell,j'} = \frac{1}{a_{j'-1}(\mu)} - \frac{1}{a_{j'+1}(\mu)} = \frac{1}{a_{i-2\,\ell'-3}(\mu)} - \frac{1}{a_{i-2\,\ell'-1}(\mu)}$. One can obtain

$$\begin{aligned} [M_2]_{i,i-2\,\ell'-2} &= \tfrac{-1}{a_{i-1}(\mu)} \left[(a_{i-2\,\ell'-3}(\mu) - a_{i-2\,\ell'-1}(\mu)) \times \right. \\ &\quad \tfrac{1}{a_{i-2\,\ell'-3}(\mu)} + \left(\tfrac{1}{a_{i-2\,\ell'-3}(\mu)} - \tfrac{1}{a_{i-2\,\ell'-1}(\mu)} \right) \times \\ &\quad [(a_{i-3}(\mu) - a_{i-1}(\mu)) + (a_{i-5}(\mu) - a_{i-3}(\mu)) \\ &\quad \left. + \cdots + (a_{i-2\,\ell'-1}(\mu) - a_{i-2\,\ell'+1}(\mu))] \right] \\ &= \tfrac{1}{a_{i-2\,\ell'-3}(\mu)} - \tfrac{1}{a_{i-2\,\ell'-1}(\mu)}. \end{aligned}$$

□

Now that we have computed M_2, we can compute matrix M which was mentioned at the beginning of this section. Suppose M_1 is the corresponding matrix for computing Legendre coefficients α_k from $m_k(f(\lambda), L)$, for $k = 0, \ldots, d$, see Proposition 1. Thus

$$\begin{bmatrix} \alpha_0 & \cdots & \alpha_d \end{bmatrix} = \begin{bmatrix} m_0(f, L) & \cdots & m_d(f, L) \end{bmatrix} M_1,$$

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

$$[M_1]_{i,j} = \begin{cases} (-1)^{i+j} \frac{(2j-1)}{L^i} \binom{j-1}{i-1} \binom{i+j-2}{j-1} & \text{if } i \le j \\ 0 & \text{if } i > j \end{cases}$$

Note that matrix M_1 can be written as the product of a lower triangular matrix M_1' and a diagonal matrice D as $M_1 = D \times M_1'$, where for $i, j = 1, \dots, n+1$,

$$[M_1']_{i,j} = \begin{cases} (-1)^{i+j} (2j-1) \binom{j-1}{i-1} \binom{i+j-2}{j-1} & \text{if } i \le j \\ 0 & \text{if } i > j \end{cases}$$

$$[D]_{i,j} = \begin{cases} \frac{1}{L^i} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

Proposition 2. *Suppose that $m_k(f(\lambda), L)$ is defined as $\int_0^L f(\lambda)\lambda^k d\lambda$, where L is the length of a given handwritten curve and $f(\lambda)$ is either $X(\lambda)$ or $Y(\lambda)$, the coordinate curves corresponding to the handwritten curve, for $k = 0, \dots, d$. Let also $\hat{f}(\lambda) = \sum_{k=0}^{d} \alpha_k P_k(\lambda) = \sum_{k=0}^{d} \beta_k S_k^{\mu}(\lambda)$ be the corresponding scaled function of $f(\lambda)$ in the interval $[-1, 1]$. Then, for $k = 0, \dots, d$, we can compute β_k as follows:*

$$\left[\, \beta_0 \quad \cdots \quad \beta_d \,\right] = \left[\, m_0(f, L) \quad \cdots \quad m_d(f, L) \,\right] D\, M_1'\, M_2. \tag{14}$$

In the rest of this section, our goal is to compute matrix $M := M_1' M_2$. Entries of M are computed as follows:

$$\begin{aligned} [M]_{i,j} &= \textstyle\sum_{k=1}^{d+1} [M_1']_{i,k} [M_2]_{k,j} = \sum_{k=j}^{d+1} [M_1']_{i,k} [M_2]_{k,j} \\ &= [M_1']_{i,j} [M_2]_{j,j} + \textstyle\sum_{k=j+2}^{d+1} [M_1']_{i,k} [M_2]_{k,j} \\ &= [M_1']_{i,j} [M_2]_{j,j} + \textstyle\sum_{\ell=1}^{\lfloor \frac{d+1-j}{2} \rfloor} [M_1']_{i,j+2\ell} [M_2]_{j+2\ell,j}. \end{aligned} \tag{15}$$

Note that in Equation (15), the last equality in the first line is derived from the fact that M_2 is a lower triangular matrix and the equality in the last line is true because $[M_2]_{j+2\ell-1,j} = 0$, for $\ell = 1, \dots, \lfloor \frac{d+1-j}{2} \rfloor$. For the last equation, we consider two different cases:

- $i \le j$: For this case the elements $[M_1']_{i,j+2\ell}$ are non-zero, for $\ell = 0, \dots, \lfloor \frac{d+1-j}{2} \rfloor$. Thus,

$$\begin{aligned} &[M]_{i,j} = (-1)^{i+j} (2j-1) \binom{j-1}{i-1} \binom{i+j-2}{j-1} \frac{1}{a_{j-1}(\mu)} + \\ &\left(\frac{1}{a_{j-1}(\mu)} - \frac{1}{a_{j+1}(\mu)} \right) \times \\ &\textstyle\sum_{\ell=1}^{\lfloor \frac{d+1-j}{2} \rfloor} (-1)^{i+j} (2j+4\ell-1) \binom{j+2\ell-1}{i-1} \binom{i+j+2\ell-2}{j+2\ell-1}. \end{aligned}$$

- $i > j$: Then the elements $[M_1']_{i,j+2\ell}$ are all zero except for $\ell \ge \frac{i-j}{2}$. Thus we obtain

$$\begin{aligned} [M]_{i,j} &= \textstyle\sum_{\ell=\lceil \frac{i-j}{2} \rceil}^{\lfloor \frac{d+1-j}{2} \rfloor} [M_1']_{i,j+2\ell} [M_2]_{j+2\ell,j} \\ &= (-1)^{i+j} \left(\frac{1}{a_{j-1}(\mu)} - \frac{1}{a_{j+1}(\mu)} \right) \times \\ &\textstyle\sum_{\ell=\lceil \frac{i-j}{2} \rceil}^{\lfloor \frac{d+1-j}{2} \rfloor} (2j+4\ell-1) \binom{j+2\ell-1}{i-1} \binom{i+j+2\ell-2}{j+2\ell-1}. \end{aligned}$$

Note that when $i = d+1$,

$$M_{d+1,j} = \begin{cases} 0 & \text{if d+j is even} \\ \binom{2d}{d} \frac{(2d^2+d-1)(a_{j-1}(\mu) - a_{j+1}(\mu))}{a_{j-1}(\mu) a_{j+1}(\mu)} & \text{otherwise} \end{cases}$$

According to the notations in Proposition 2, for $k = 0, \dots, d$, we have

$$\beta_k = \sum_{j=0}^{d} \frac{m_j(f(\lambda), L) [M]_{j+1,k+1}}{L^{j+1}}. \tag{16}$$

On the one hand, M only depends on d and μ, and thus, this matrix can be precomputed. On the other hand, computation of L^k can be done with one operation, see [3]. Furthermore, each β_k can be computed with $3d+3$ operations, assuming that moment integrals are known. Thus, $\sum_{k=0}^{d} 3d+3 = (d+1)(3d+3)$ operations are required for computation of all β_k, for $k = 0, \dots, d$. Since for the purpose of handwriting recognition, d is fixed and small, thus the quadratic exponent of d is negligible and the coefficients of the coordinate curves of handwritten characters in the Legendre-Sobolev basis can be computed in constant time. In Section V, we explain how moment integrals are computed for the purpose of this paper.

IV. Condition Number

Let $g_d^{\mu}(\mathbf{x})$ be the characteristic polynomial of matrix MM^T in variable $\mathbf{x}$, where $(d+1) \times (d+1)$ is the size and M^T is the transpose matrix of M. Since matrix MM^T is real and symmetric, all the roots of $g_d^{\mu}(\mathbf{x})$ are real numbers. Let $\sigma_1 \le \cdots \le \sigma_{d+1}$ be the roots of $g_d^{\mu}(\mathbf{x})$. Then the norm-2 condition number of M is computed by $\kappa_2(M) = \sqrt{\frac{\sigma_{d+1}}{\sigma_1}}$. For $d = 9$, one can compute $g_9^{\mu}(\mathbf{x})$ which has the following form:

$$\begin{aligned} g_9^{\mu}(\mathbf{x}) = \; & \mathbf{x}^{10} + t_2^9(\mu)\, \mathbf{x}^9 + t_4^8(\mu)\, \mathbf{x}^8 + t_6^7(\mu)\, \mathbf{x}^7 + \\ & t_8^6(\mu)\, \mathbf{x}^6 + t_{10}^5(\mu)\, \mathbf{x}^5 + t_{12}^4(\mu)\, \mathbf{x}^4 + t_{14}^3(\mu)\, \mathbf{x}^3 + \\ & t_{14}^2(\mu)\, \mathbf{x}^2 + t_{14}^1(\mu)\, \mathbf{x}^1 + t_{14}^0(\mu)\, \mathbf{x}^0, \end{aligned} \tag{17}$$

where $t_n^m(\mu)$ is a polynomial of degree n in μ and is the coefficient of the monomial $\mathbf{x}^m$ in $g_9^{\mu}(\mathbf{x})$. Here, we have not shown the polynomials $t_n^m(\mu)$ due to the lack of space. Note that the entries of M are not polynomials in μ. Thus, for simplification of computations, we have computed Taylor expansions of the entries of M around $\mu = 0$, and then attempted to compute the characteristic polynomial $g_9^{\mu}(\mathbf{x})$. Figure 1 demonstrate how the norm-2 condition number of matrix M changes with respect to d, where the size of matrix

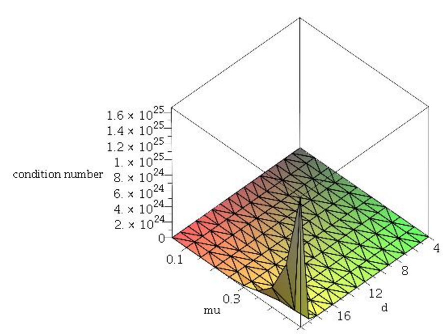

Fig. 1. Norm-2 condition number of matrix M w.r.t d and μ.

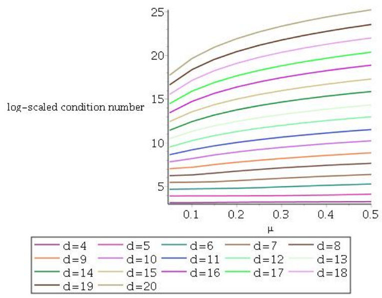

Fig. 2. Based-10-log-scaled norm-2 condition number of M w.r.t d and μ.

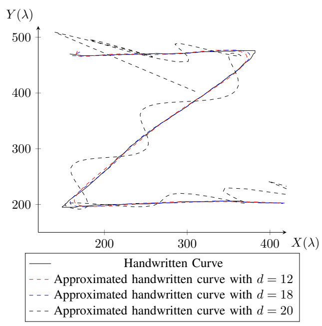

Fig. 3. The black solid curve is a handwritten curve, while the dashed ones are approximations of the handwritten curve in Legendre-Sobolev basis of degrees $12, 18, 20$, and $\mu = 0.25$, showing the effect of condition number.

M is $d+1$, and μ is the parameter appearing in Legendre-Sobolev inner product. Since the changes of the values of the condition number of matrix M are really high, see Figure 1, we have plotted the based-10 log scaled condition number of matrix M, as well, in Figure 2. Note that for creating this plot, condition number of matrix M is computed for $d = 4, \ldots, 20$ and $0 < \mu \leq 0.5$. Based on our experimentation, the value of the condition number increases dramatically when $d > 18$ and $\mu > 0.3$. In Figure 3, the solid curve is a handwritten curve, and the dashed curves are the approximations of the black solid curve using truncated Legendre-Sobolev series of degrees $12, 18, 20$ and $\mu = 0.25$. As one can see, the approximation corresponding to degree 20 becomes very chaotic.

V. Moment Integrals

To compute the integrals $\int_0^L \lambda^k f(\lambda) d\lambda$, where L is the length of the handwritten curve, we use an adapted version of trapezoid method for numerical integration. Suppose that we receive the coordinates $(X(\ell_i), Y(\ell_i))$ of the handwritten curve at time i with corresponding arc lengths $\ell_0 = 0, \ell_1, \ell_2, \ldots, \ell_m = L$ for some $m \in \mathbb{N}$.

Since we receive the coordinates $(X(\ell_i), Y(\ell_i))$ one at a time when tracing out the handwritten curve, we compute moment integrals by computing the following:

$$\sum_{i=0}^{m-1} \int_{\ell_i}^{\ell_{i+1}} \lambda^k f(\lambda) d\lambda.$$

For computing $\int_{\ell_i}^{\ell_{i+1}} \lambda^k f(\lambda) d\lambda$ we use trapezoid method when the number of steps in this integration method is $ns_i \in \mathbb{N}$. Let $\Delta\lambda_i = \frac{\ell_{i+1} - \ell_i}{ns_i}$. Then we approximate the latter integral by

$$\int_{\ell_i}^{\ell_{i+1}} \lambda^k f(\lambda) d\lambda = \sum_{j=0}^{ns_i - 1} \left(\frac{f(\lambda_{i_j})\lambda_{i_j}^k + f(\lambda_{i_{j+1}})\lambda_{i_{j+1}}^k}{2} \right) \Delta\lambda_i$$

where $\lambda_{i_j} := \ell_i + j\,\Delta x$, for $j = 0, \ldots, ns_i - 1$. Note that when $ns_i > 1$, the quantities $f(\lambda_{i_j})$ are not known, for $j = 1, \ldots, ns_i - 1$. Thus we use linear spline interpolation to find such values as following:

$$f(\lambda_{i_j}) := f(\ell_i) + \frac{\lambda_{i_j} - \ell_i}{\ell_{i+1} - \ell_i} \left(f(\ell_{i+1}) - f(\ell_i) \right).$$

In Section VI, we show how the total number of steps used in the numerical integration method for computing moment integrals can affect the forward error in $f(\lambda)$ in Legendre-Sobolev basis.

VI. Experiments

Suppose the coordinate curves of a handwritten letter "S" are given by Equations (18) and (19):

$$\begin{aligned} X(\lambda) = &-\tfrac{1883991}{100}\lambda^3 - \tfrac{138501}{20}\lambda^9 + \tfrac{3041491}{100}\lambda^8 + \\ &\tfrac{2790504}{25}\lambda^6 + \tfrac{263031}{100}\lambda^2 - \tfrac{10513199}{100}\lambda^5 - \\ &\tfrac{3724291}{50}\lambda^7 + \tfrac{16809}{25}\lambda^{10} + \tfrac{604057}{10}\lambda^4 + \\ &\tfrac{1807}{25} - \tfrac{20797}{50}\lambda, \end{aligned} \tag{18}$$

$$\begin{aligned} Y(\lambda) = &\tfrac{855949}{50}\lambda^3 + \tfrac{684931}{100}\lambda^9 - \tfrac{3200151}{100}\lambda^8 - \\ &\tfrac{12334211}{100}\lambda^6 - \tfrac{327317}{100}\lambda^2 + \tfrac{5572319}{50}\lambda^5 + \\ &\tfrac{8169529}{100}\lambda^7 - \tfrac{61831}{100}\lambda^{10} - \tfrac{5848441}{100}\lambda^4 + \\ &\tfrac{6253}{100} + \tfrac{12128}{25}\lambda, \end{aligned} \tag{19}$$

where $0 \leq \lambda \leq 2$. Then one can compute $X(\lambda)$ and $Y(\lambda)$ in Legendre-Sobolev basis, for $\mu = 0.25$. For the given letter "S", one can verify that $L = 690.8961469874615$.

Figure 4 demonstrates how the number of steps in the numerical integration method used for computing moment integrals affects relative error for computing coefficients of $X(\lambda)$ (given by Equation (18)) in Legendre-Sobolev basis w.r.t different polynomial norms. The computations in this figure are done by double precision. Our experiments showed that higher precision results in the same error rates as double precision.

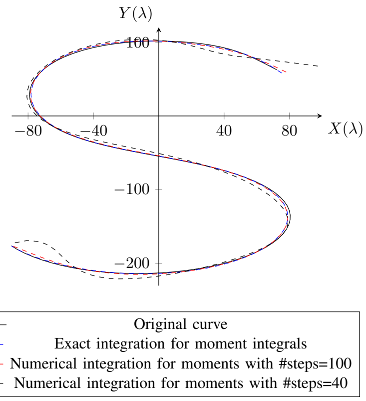

Fig. 5. The solid black curve is the given curve in monomial basis, while the blue curve is the approximated curve in Legendre-Sobolev basis with its moment integrals computed by exact integration. The other two curves are the approximations of the original curve in Legendre-Sobolev basis where the moment integrals are computed by numerical integration w.r.t different number of steps in the integration method.

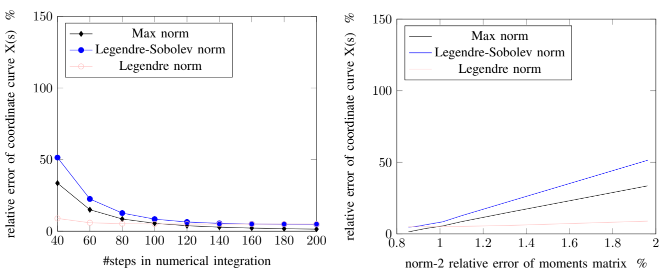

Fig. 4. This figure shows how the number of steps in numerical integration for computing moment integrals affects relative error of both moment integrals w.r.t norm-2 and coordinate curve $X(\lambda)$ w.r.t different norms represented in this paper: (right) relative error of coordinate curve $X(\lambda)$ v.s number of steps in the numerical integration method for computing moment integrals (left) relative error of coordinate curve $X(\lambda)$ v.s norm-2 relative error of moment integrals matrix when these moment integrals are corresponding to the ones computed in the left graph.

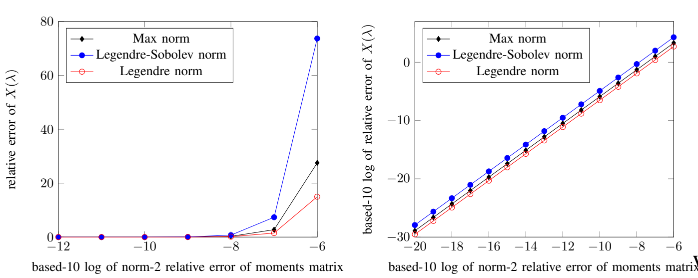

Fig. 6. This figure shows the relation between norm-2 relative error of the matrix of moment integrals and relative error of $X(\lambda)$ in Legendre-Sobolev basis, w.r.t different polynomial norms for letter S.

In Figure 5, the black curve is corresponding to coordinate curves given by Equations (18) and (19). The rest of the curves are corresponding to the same curve, where $X(\lambda)$ and $Y(\lambda)$ are computed in Legendre-Sobolev basis based on computations of moment integrals and multiplying such moments by M; for the blue curve, moment integrals are computed by exact integration and for the other two curves, moment integrals are computed by numerical integration. As one can see for when the number of steps in integration exceeds 100, the difference between the red and blue curve is really small.

Figure 6 demonstrates error rates for $X(\lambda)$, given by Equation (18), arising from our method for computing Legendre-Sobolev coefficients from moment integrals, when there is some errors in such integrals. Our method refers to computing Legendre-Sobolev coefficients of $X(\lambda)$ by multiplying its moments by matrix M. The computations in this graph are done with high precision when the number of digits is 100.

Figure 7 shows the absolute error polynomial $\Delta X(\lambda)$, when $X(\lambda)$ is computed in Legendre-Sobolev basis.

As for letter “S”, we can repeat the same computations for letter “Z” which is given by coordinate curves below:

$$\begin{aligned}X(\lambda) = \;& 8018135\,\lambda^{25} - 184122133\,\lambda^{24} + 1963740491\,\lambda^{23} - \\ & 12887030261\,\lambda^{22} + 58057748146\,\lambda^{21} - \\ & 189363255215\,\lambda^{20} + 457944788302\,\lambda^{19} - \\ & 820603554099\,\lambda^{18} + 1047542179023\,\lambda^{17} - \\ & 805267306124\,\lambda^{16} - 33534962822\,\lambda^{15} + \\ & 1143077678314\,\lambda^{14} - 1924407366996\,\lambda^{13} + \\ & 2003464645667\,\lambda^{12} - 1508577522702\,\lambda^{11} + \\ & 857870845235\,\lambda^{10} - 372653355690\,\lambda^{9} + \\ & 123250439506\,\lambda^{8} - 30628162000\,\lambda^{7} + \\ & 5591644024\,\lambda^{6} - 725374922\,\lambda^{5} + 63700003\,\lambda^{4} - \\ & 3521783\,\lambda^{3} + 109078\,\lambda^{2} - 1091\,\lambda + 161,\end{aligned} \tag{20}$$

$$\begin{aligned}Y(\lambda) = \;& 2067937\,\lambda^{25} - 40256869\,\lambda^{24} + 331918218\,\lambda^{23} - \\ & 1347575656\,\lambda^{22} + 1032521518\,\lambda^{21} + \\ & 19857114176\,\lambda^{20} - 133063329052\,\lambda^{19} + \\ & 494989230846\,\lambda^{18} - 1295311702444\,\lambda^{17} + \\ & 2564667950403\,\lambda^{16} - 3970321001968\,\lambda^{15} + \\ & 4886882153238\,\lambda^{14} - 4821817961103\,\lambda^{13} + \\ & 3823890682331\,\lambda^{12} - 2433321765499\,\lambda^{11} + \\ & 1235747399680\,\lambda^{10} - 496237117534\,\lambda^{9} + \\ & 155475228012\,\lambda^{8} - 37312970133\,\lambda^{7} + \\ & 6691099587\,\lambda^{6} - 866765142\,\lambda^{5} + 77380557\,\lambda^{4} - \\ & 4447024\,\lambda^{3} + 148139\,\lambda^{2} - 2372\,\lambda + 477,\end{aligned} \tag{21}$$

where $0 \leq \lambda \leq 2$.

Figure (8) is Figure (6) counterpart for letter “Z”, and it demonstrates that there is the same pattern for errors in approximating curves corresponding to letters “S” and “Z”, but the error rates for letter “Z” are higher. Furthermore, in Figure (9), the black solid curve is corresponding to Equations (20) and (21), and the dashed curves are the approximations of the black curve using truncated Legendre-Sobolev basis polynomials of degree 10 with $\mu = 0.25$, and different error rates in computation of moment integrals.

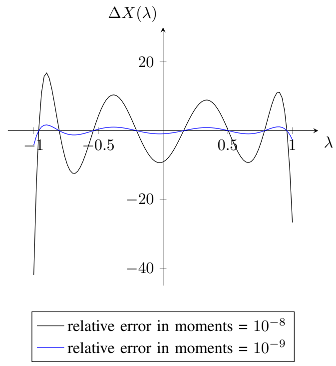

Fig. 7. Absolute error $\Delta X(\lambda)$ in Legendre-Sobolev basis where the norm-2 relative error in the matrix of moment integrals is 10^{-8} and 10^{-9}, and $-1 \leq \lambda \leq 1$.

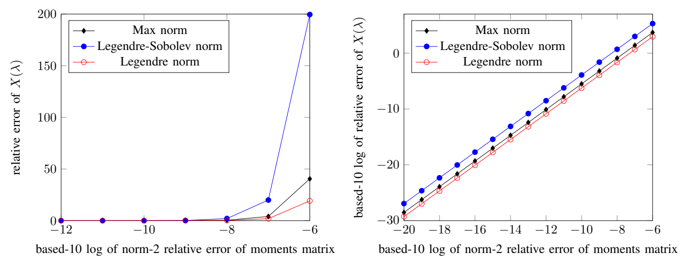

Fig. 8. Relation between norm-2 relative error of the matrix of moment integrals and (left) relative error, (right) based-10 log of relative error of $X(\lambda)$ in Legendre-Sobolev basis, w.r.t different polynomial norms for handwritten letter Z.

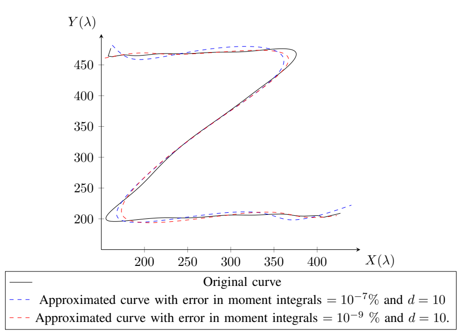

Fig. 9. The solid black curve is the given curve in monomial basis, while the blue and red curves are the approximated curves of black curve of degree 10 in Legendre-Sobolev basis with their moment integrals computed numerically and with different error rates.

VII. Conclusion

We have presented a method for real-time computation of Legendre-Sobolev approximations by means of moment integrals and linear algebra. This method may be used in real-time online handwriting recognition.

One of the main results of this paper is Theorem 2 in which we have computed a matrix M_2 that can be used to compute the coefficients of coordinate curves in a Legendre-Sobolev basis from such coefficients in Legendre basis. Furthermore, computation of another matrix M gives a method for computing Legendre-Sobolev coefficients from moment integrals by one matrix multiplication.

In this paper, we have used linear approximation in computation of moment integrals. One interesting direction would be to investigate how to compute moment integrals efficiently and accuratly with higher order approximations.

References

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

[2] B. W. Char and S. M. Watt. Representing and Characterizing Handwritten Mathematical Symbols through Succinct Functional Approximation. In *9th International Conference on Document Analysis and Recognition (ICDAR 2007)*, pages 1198–1202, 2007.

[3] O. Golubitsky and S. M. Watt. Online Stroke Modeling for Handwriting Recognition. In *Proceedings of the 2008 conference of the Centre for Advanced Studies on Collaborative Research*, pages 6:72–6:80. ACM, 2008.

[4] 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, San Jose, CA, USA, January 20-22, 2009. Proceedings*, page 72470C. SPIE, 2009.

[5] F. Hausdorff. Summationsmethoden und Momentfolgen. I. 9:74–109, 1921.

[6] F. Hausdorff. Summationsmethoden und Momentfolgen. II. 9:280–299, 1921.

[7] R. Hu and S. M. Watt. Determining Points on Handwritten Mathematical Symbols. In *Intelligent Computer Mathematics*, pages 168–183, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[8] R. Hu and S. M. Watt. Identifying Features via Homotopy on Handwritten Mathematical Symbols. In *15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC*, pages 61–67, 2013.

[9] N. Joshi, G. Sita, A. G. Ramakrishnan, and S. Madhvanath. Comparison of Elastic Matching Algorithms for Online Tamil Handwritten Character Recognition. In *Ninth International Workshop on Frontiers in Handwriting Recognition, IWFHR-9*, pages 444–449. IEEE Computer Society, 2004.

[10] L. Prasanth, V. Babu, R. Sharma, G. V. Rao, and Dinesh M. Elastic Matching of Online Handwritten Tamil and Telugu Scripts Using Local Features. In *9th International Conference on Document Analysis and Recognition (ICDAR 2007)*, pages 1028–1032. IEEE Computer Society, 2007.

[11] F. W. Schäfke. Zu den Orthogonalpolynomen von Althammer. 252:195–199, 1972.

[12] S. Uchida and H. Sakoe. A Survey of Elastic Matching Techniques for Handwritten Character recognition. *IEICE Transactions*, 88-D(8):1781–1790, 2005.