SPLINE APPROXIMATION VIA THE CONTROL POLYGON

by

Håkon Mørk

THESIS
for the degree of
MASTER'S DEGREE IN APPLIED MATHEMATICS AND MECHANICS

(Master i Anvendt matematikk og mekanikk)

Faculty of Mathematics and Natural Sciences
University of Oslo

December 2012

Det matematisk-naturvitenskapelige fakultet
Universitetet i Oslo

Contents

1 Introduction **2**

2 Spline functions **3**
2.1 The basis functions, spline formulation 3
2.2 Algorithm for calculating splines 4
2.3 The control polygon 5
2.4 Knot insertion 6

3 Existing approximation methods **7**
3.1 Variation-diminishing spline approximation 7
3.2 Spline interpolation 9
3.2.1 Error bounds 10

4 Control polygon approximation **11**
4.1 Formulation of the interpolation problem 11
4.2 The linear equation system 12

5 Numerical analysis of specific approximation problems **13**
5.1 Norms and error measures 13
5.2 Quadratic splines 14
5.2.1 The simplest method 14
5.2.2 Changing interpolation points 18
5.2.3 Spline interpolation 19
5.2.4 Intermediate methods 21
5.3 Cubic splines 23
5.3.1 Variation-diminishing approximation 23
5.3.2 Spline interpolation 25
5.3.3 Intermediate methods 25
5.4 Shape-preserving properties 26

6 Conclusion **28**

Chapter 1

Introduction

Spline functions are piecewise polynomials, a popular tool for representation of more general functions in many applications. The spline function is uniquely determined by a set of knots, between which the function is polynomial, the degree of the polynomial, and a set of coefficients determining its values.

The name spline was originally applied to an instrument which helped draughtsmen and designers to draw smooth interpolating curves by hand. Splines can be made to fit a wide variety of properties.

Splines may be created to approximate of functions and data sets, or as part of design of a model of some physical or engineering phenomenon.

The approximating properties of splines are already well-studied in the literature, with methods including, but not limited to: direct interpolation, i.e. forcing the spline to agree at certain interpolation points, variation-diminishing spline approximation, which uses the values of the function to be approximated as the coefficients of the new spline, or least squares approximation, which attempts to minimise the distance between the spline and the function at some given data points.

All these approximation methods attempt to impose conditions on the spline function itself. However, there exists a piecewise linear function related to the spline, known as the control polygon, which converges to the spline as the knots are refined. Thus, we can attempt to enforce the interpolation conditions on this control polygon, giving rise to a whole new family of approximation problems.

It is of interest to measure some properties of these approximation problems, for example the speed of convergence to the function, and how these compare with the more well-studied spline approximation methods. This will be the main aim of the following thesis.

Chapter 2

Spline functions

In the introduction, we introduced a number of terms relating to spline functions, which we will now define more precisely. The spline functions are defined by the degree of the piecewise polynomial pieces, p, as well as the intersections of the polynomials, known as **knots**, and usually defined as a vector $\boldsymbol{t}$. These all specify a space of functions called a **spline space**, with **order** p and an associated **knot vector** $\boldsymbol{t}$.

The spline space is spanned by a vector basis of functions. One such basis is known as the **B-spline** basis. This basis spans the space of polynomials of degree p, and it is relatively easy to construct algorithms evaluating the splines and to construct spline operators which either interpolate or approximate other non-polynomial functions. Also, because these operators are relatively simple, it is possible to prove certain properties of stability and accuracy.

2.1 The basis functions, spline formulation

Definition 1. Representation of a spline. *A spline g is represented as a linear combination of B-splines $B_{j,p}(x)$ such that $g(x) = \sum_{j=1}^{n} c_j B_{j,p}(x)$.*

The B-splines $B_{j,p}(x)$ can be calculated and defined recursively, as shown by de Boor in [3] and others. He described these as normalized B-splines, using the notation $N_{i,k}(t)$ for what we will annotate $B_{j,p}(x)$.

Given a knot vector $\boldsymbol{\tau} = [\tau_1,\ \tau_2,\ \ldots,\ \tau_{n+p+1}]$ which is nondecreasing, then define

$$B_{j,0}(x) = \begin{cases} 1, & \tau_j \leq x < \tau_{j+1} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

Now for any $p \in \mathbb{N}$, $B_{j,p}(x)$ is given by the recurrence relation

$$B_{j,p}(x) = \frac{x - \tau_j}{\tau_{j+p} - \tau_j} B_{j,p-1}(x) + \frac{\tau_{j+1+p} - x}{\tau_{j+1+p} - \tau_{j+1}} B_{j+1,p-1}(x). \tag{2.2}$$

The knot vector of the spline thus defines the boundaries of the subintervals of the piecewise polynomials. Each B-spline $B_{j,p}$ is only non-zero on the interval $[\tau_j, \tau_{j+p+1}]$, and on any interval $[\tau_\mu, \tau_{\mu+1}]$ there are only $p+1$ non-zero B-splines, $\{B_{j,p}\}_{j=\mu-p}^{\mu}$ [3].

The c_js in the definition of a spline function are called the function's coefficients. As we now have a basis for the spline space, by varying c_j in $\mathbb{R}$, we can make the following definition:

Definition 2. The spline space. *The spline space $\mathbb{S}_{p,\tau}$ is defined to contain functions defined as follows:*

$$\mathbb{S}_{p,\boldsymbol{\tau}} := \left\{ \sum_{j=1}^{n} c_j B_{j,p} | c_j \in \mathbb{R} \, for \, 1 \leq j \leq n \right\}. \tag{2.3}$$

From [6] we cite some definitions about knot vectors, which will be important when describing knot vectors for our interpolation problems later.

Definition 3. Properties of the knot vector. *A knot vector is said to be $p+1$-extended if*

1. $n \geq p+1$.
2. $t_{p+1} < t_{p+2}$ *and* $t_n < t_{n+1}$.
3. $t_j < t_{j+p+1}$ *for* $j = 1, 2, \ldots, n$.

A $p+1$-extended knot vector for which $t_1 = t_{p+1}$ and $t_{n+1} = t_{n+p+1}$ is called $p+1$-regular.

Finally, from [5] we note a couple of important facts; this basis spans the space of polynomials of degree p and the basis functions $B_{j,p}$ are linearly independent on $[\tau_{p+1}, \tau_{n+1})$ for a $p+1$-extended knot vector. Therefore, that if we are to solve an interpolation problem with a unique function from a spline space on $[a, b]$, then we need to choose a knot vector such that $[a, b] \subset [\tau_{p+1}, \tau_{n+1}]$.

2.2 Algorithm for calculating splines

If we wish to calculate the value of a spline function at $x \in [a, b]$, we must first determine its location in the knot vector - thus we must find an integer μ such that $\tau_\mu \leq x < \tau_{\mu+1}$. Then, as stated in section 2.1, only the $p+1$ B-splines $\{B_{j,p}\}_{j=\mu-p}^{\mu}$ are non-zero, so the sum in definition 1 simplifies to

$$\sum_{j=\mu-p}^{\mu} c_j B_{j,p}(x).$$

The recursive formula defining the B-splines suggests a recursive algorithm for calculating exactly these $p+1$ B-splines. This recursion can be represented as a succession of matrix multiplications, which is very convenient for computer calculations. We now describe the algorithm following the arguments in [6].

Assume you have a spline space $\mathbb{S}_{p,\boldsymbol{\tau}}$ and let μ be the integer such that $\tau_\mu \leq x < \tau_{\mu+1}$ and $p+1 \leq \mu \leq n$. For each positive integer $k \leq p$, define the matrix $\boldsymbol{R}_k(x)$ by

$$\boldsymbol{R}_k(x) = \begin{pmatrix} \frac{\tau_{\mu+1}-x}{\tau_{\mu+1}-\tau_{\mu+1-k}} & \frac{x-\tau_{\mu+1-k}}{\tau_{\mu+1}-\tau_{\mu+1-k}} & 0 & \cdots & 0 \\ 0 & \frac{\tau_{\mu+2}-x}{\tau_{\mu+2}-\tau_{\mu+2-k}} & \frac{x-\tau_{\mu+2-k}}{\tau_{\mu+2}-\tau_{\mu+2-k}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{\tau_{\mu+k}-x}{\tau_{\mu+k}-\tau_\mu} & \frac{x-\tau_\mu}{\tau_{\mu+k}-\tau_\mu} \end{pmatrix}.$$

Then the $p+1$ non-zero B-splines at x, $\boldsymbol{B}_{p,\mu}(x) := [B_{p,\mu-p}, B_{p,\mu-p+1}, \dots, B_{p,\mu}]$ are given by

$$\boldsymbol{B}_{p,\mu}(x) = \boldsymbol{R}_1(x)\boldsymbol{R}_2(x)\cdots\boldsymbol{R}_p(x). \tag{2.4}$$

The algorithm to calculate a vector of length $p+1$ containing these B-splines now follows almost directly from (2.4); starting with the $\boldsymbol{R}_1$-matrix, we multiply from the right with $\boldsymbol{R}_2(x)$ to produce $\boldsymbol{B}_{2,\mu}(x)$, then $\boldsymbol{R}_3(x)$ to produce $\boldsymbol{B}_{3,\mu}(x)$ etc. up to $\boldsymbol{B}_{p,\mu}(x)$. Multiplying this by the coefficient vector $[c_{\mu-p}, c_\mu]$ then given the spline value $f(x)$.

2.3 The control polygon

The main attempt to approximate a function by imposing conditions on the function known as the control polygon of the spline. We must therefore define exactly how to calculate this function.

The control polygon is a piecewise function, linear between the **knot averages** of the spline's knot vector.

Definition 4. The knot averages. *Given a spline space $\mathbb{S}_{p,\boldsymbol{t}}$ with degree p and knot vector $\boldsymbol{t} = [t_1, t_2, \dots, t_{n+p+1}$ such that $t_i \leq t_{i+1}]$, the vector of knot averages $\boldsymbol{t}^\star$ are defined as*

$$t_i^\star := \frac{t_{i+1} + \cdots + t_{i+p}}{p}$$

Note that for a $p+1$-regular knot vector, $t_1^\star = t_1$ and $t_n^\star = t_{n+1}$. Let us now define how to calculate the control polygon $\Gamma_{p,\boldsymbol{t}}$.

Definition 5. The control polygon. *Assume we have a spline f with coefficients $\boldsymbol{c}$, degree p and knot vector $\boldsymbol{t}$. Let $x \in [\tau_1^\star, \tau_n^\star]$, and let i be the integer in $[1, n]$ such that $x \in [\tau_i^\star, \tau_{i+1}^\star]$. Then*

$$\Gamma_{p,\boldsymbol{t}}(f)(x) := \left(\frac{t_{i+1}^\star - x}{t_{i+1}^\star - t_i^\star}\right) c_i + \left(\frac{x - t_i^\star}{t_{i+1}^\star - t_i^\star}\right) c_{i+1}. \tag{2.5}$$

This function is piecewise linear, and is linear on the intervals $[\tau_i^\star, \tau_{i+1}^\star]$. From [6] we have the following theorem about the control polygon.

Theorem 1. *Let f be a spline and $\Gamma_{p,\boldsymbol{t}}$ its control polygon as defined in definition 5. Also, assume that $\boldsymbol{t}$ is a $p+1$-regular knot vector, and define $h := \max_i t_{i+1} - t_i$ as the maximum distance between two knots in the knot vector. Then there exists some constant $K \in \mathbb{R}$ such that*

$$|\Gamma_{p,\boldsymbol{t}}(f)(x) - f(x)| \leq Kh^2 \max_{y \in [\tau_1, \tau_{n+p+1}]} |D^2 f(y)| \tag{2.6}$$

for all $x \in [\tau_1^\star, \tau_n^\star]$.

As the spline f is piecewise continuous, we know that $|D^2 f(y)| \leq C$, and the control polygon converges to the spline as h goes to 0. This suggests that interpolating via the control polygon may give an accurate interpolation method.

2.4 Knot insertion

We saw in the previous section that by inserting knots into the knot vector $\boldsymbol{\tau}$ and producing a knot vector with smaller h, the control polygon will converge towards the spline function. We will now outline the calculations required to convert one set of coefficients $\boldsymbol{c}$ valid on a knot vector $\boldsymbol{\tau}$ to a new set of coefficients $\boldsymbol{d}$ associated with a new knot vector $\boldsymbol{t} \supset \boldsymbol{\tau}$. This new knot vector is associated with an m-dimensional spline space $\mathbb{S}_{p,\boldsymbol{t}}$.

This can be done by exploiting the triangular formulations of the B-splines. In [6] the authors show how to produce a knot insertion matrix $A_{\boldsymbol{t}}^{\boldsymbol{\tau}}$ of dimension $m \times n$ and with entries $\alpha_{i,j}$ such that $d_i = \sum_{j=1}^{n} \alpha_{i,j} c_j$, which converts the coefficients c to the higher-dimension coefficients d.

This matrix is built up by using the $\boldsymbol{R}_k$-matrices of section 2.2 on the knot vector $\boldsymbol{\tau}$, evaluated in the knot vector values $t_{i+1}, \ldots, t_{i+p}$. If $\alpha_{i,j}$ is the ith row and jth column of $\boldsymbol{A}_{\boldsymbol{t}}^{\boldsymbol{\tau}}$, and μ is the integer such that $\tau_\mu \leq t_i < \tau_{\mu+1}$, then the ith row of $\boldsymbol{A}_{\boldsymbol{t}}^{\boldsymbol{\tau}}$ is the transpose of the following vector:

$$(\alpha_{i,\mu-p}, \cdots, \alpha_{i,\mu}) = (\boldsymbol{R}_1(t_{i+1}) \cdots \boldsymbol{R}_p(t_{i+p})). \tag{2.7}$$

The algorithm for calculating knot insertion thus involves going systematically through the $\boldsymbol{t}$ vector to calculate the m rows. There exist simpler algorithms for the insertion of few knots, however.

Chapter 3

Existing approximation methods

There already exists literature on a number of spline approximation methods, attempting to form hypotheses and prove certain properties of the methods. Error measures and shape-preserving properties have been particular fields of interest. In order to compare the method of approximation by the control polygon to the already existing methods, we shall now outline some well-known methods described in [6] and some of the knowledge that has been obtained about these methods.

3.1 Variation-diminishing spline approximation

The variation-diminishing spline approximation was introduced by Schoenberg in [1] and also studied in [2, 4] among others.

Definition 6. *The variation-diminishing spline approximation. Assume we have a spline space* $\mathbb{S}_{p,\boldsymbol{\tau}}$ *with associated B-spline basis* $B_{j,p}$. *Let* $\boldsymbol{\tau}^\star$ *be the knot averages as defined in section 2.3. Let* $x \in [\tau_{p+1}, \tau_{n+1}]$. *Then the variation-diminishing spline approximation* $f(x)$ *of a function* g *is given such that*

$$f(x) := \sum_{j=1}^{n} g(\tau_j^\star) B_{j,p}(x).$$

Thus, the coefficients c_j *are set equal to* $g(\tau_j^\star)$.

The term **variation-diminishing** has a specific meaning in the literature, relating to the shape of the approximation. Following [7], we define the variation-diminishing property by first introducing a notation for the number of sign changes, letting $S^-(f, X)$ be defined as the number of sign changes as f traverses a subset X of the real line. Zeros of $f(x)$ are not counted as changes in sign.

We define a **variation-diminishing** transformation T on a family of functions $\mathbb{F}$ defined on X into a family of functions $\mathbb{G}$ defined on X_1 to be such that

$$S^-(Tf, X_1) \leq S^-(f, X)$$

It can be proved that the method defined in 6 is variation-diminishing on an interval X. From [6] we know that

$$S^-(f, X) \leq S^-(\boldsymbol{c}),$$

where $\boldsymbol{c}$ is the coefficient vector of the spline. Since $c_j = g(\tau_j^\star)$, if c_{j-1} and c_j have different signs, then there is at least one sign change in g between $\tau_{j-1}^\star$ and $\tau_j^\star$ by the intermediate value theorem. Therefore

$$S^-(\boldsymbol{c}) \leq S^-(g, X),$$

and hence

$$S^-(f, X) \leq S^-(g, X).$$

In [6], the authors describe a number of other well-known properties of this approximation, which we will quickly outline.

Proposition 1. Preservation of bounds. *Let g be a function from $[a, b] \in \mathbb{R}$ to $\mathbb{R}$ such that*

$$g_{\min} \leq g(x) \leq g_{\max} \text{ for all } x \in [a, b]$$

Let Vg be the variation-diminishing spline approximation of f in the spline space $\mathbb{S}_{p,\boldsymbol{t}}$. Then Vg obeys

$$g_{\min} \leq Vg(x) \leq g_{\max} \text{ for all } x \in [a, b]$$

The value of the spline is bounded by its coefficients, and as the coefficients are direct evaluations of the function at the knot averages, this means the value of the spline is bounded by the maximum value of the function.

Proposition 2. Preservation of monotonicity. *Let g be a function from $[a, b] \in \mathbb{R}$ to $\mathbb{R}$ such that if $x_0 < x_1$ then $g(x_0) \leq g(x_1)$. The function g is then called an **increasing** function. Let Vg be the variation-diminishing spline approximation of f in the spline space $\mathbb{S}_{p,\boldsymbol{t}}$ with $\boldsymbol{t}$ a $p+1$-extended knot vector. Then Vg is also an increasing function, and*

$$Vg(x_0) \leq Vg(x_1)$$

*Also, if g is a function such that f $x_0 < x_1$ then $g(x_0) \geq g(x_1)$ for all $x_0, x_1 \in [a, b]$. g is then called a **decreasing** function. If Vg is defined as above, then Vg is a decreasing function.*

Again, the direct control on the coefficients is the main property required to prove this, as a spline is increasing if the coefficients are increasing.

Proposition 3. Preservation of convexity. *Let g be a function from $[a,b] \in \mathbb{R}$ to $\mathbb{R}$ such that*

$$f\left((1-\lambda)\,x_0 + \lambda x_1\right) \leq (1-\lambda) f(x_0) + \lambda f(x_1)$$

*for all $x_0, x_1 \in [a,b]$ and for all $\lambda \in [0,1]$. The function g is then called a **convex** function on $[a,b]$ Now, let Vg be the variation-diminishing spline approximation of f in the spline space $\mathbb{S}_{p,\boldsymbol{t}}$ with $\boldsymbol{t}$ a $p+1$-extended knot vector. Then Vg is also a convex function on $[a,b]$.*

We see that the variation-diminishing spline approximation has some very nice shape-preserving properties, and that these can be proved directly because the coefficients are determined by direct evaluation. This may be useful to remember as we now attempt to describe interpolation by imposing conditions on the control polygon; we now no longer determine the coefficients directly, but by linear interpolations between data points.

3.2 Spline interpolation

Spline interpolation approximation methods are a class of methods where the data set $\{x_i, g(x_i)\}$ with $i \in [1,n] \cap \mathbb{N}$ is directly interpolated. The general problem formulation is, given a degree p and a set of knots $\boldsymbol{\tau}$, find an $f \in \mathbb{S}_{p,\boldsymbol{\tau}}$, such that is, $f(x_i) = g(x_i)$ for all i in the data set.

Note that this description is invariant of degree and knot vector. The classic implementation of spline interpolation, with origins in mechanical models, requires cubic polynomial pieces and interpolation at the knots themselves. As the spline is only defined for $[\tau_{p+1}, \tau_{p+n}]$, this gives only n conditions on the spline, while $n+p-1$ conditions are required to determine the spline uniquely. Thus, constraints on the derivative at the end points are usually added.

In [6], two such methods are introduced and described: the cubic Hermite spline interpolation method, with Hermite boundary conditions

$$f'(x_1) = g'(x_1) \quad f'(x_n) = g'(x_n)$$

and a method with the natural boundary conditions

$$f''(x_1) = 0 \quad f''(x_n) = 0.$$

These have some remarkable properties, as they are functions which minimize, in some sense, the integral of the second derivative. The Hermite interpolation method is a minimizer among all functions in $C^2[a,b]$ which interpolate the original function g at the points $x_i$and obey the Hermite boundary conditions at x_1 and x_n, while the natural boundary condition minimizes among all C^2 functions which interpolates g at the data points.

3.2.1 Error bounds

For any approximation method S, the **error** $|Sg(x)-g(x)|$ is a natural quantity to measure. If we now let g be a continuous function from $[a,b]$ into $\mathbb{R}$ with n continuous derivatives - that is, g is a member of the continuity class $C^n([a,b])$ - we can cite from the existing literature some theorems about whether

$$\max_{g\in\mathbb{C}^n([a,b])}\max_{x\in[a,b]}|Sg(x)-g(x)| \tag{3.1}$$

is bounded by some expression.(3.1) will be referred to in the following as $||Sg-g||_\infty$, the supremum norm of $Sg-g$. Hall & Meyer showed in [8] that for $g\in C^4([a,b])$, cubic Hermite spline interpolation obeys the following inequality

$$||Sg-g||_\infty \leq C||g^{(4)}||_\infty h^4,$$

and more generally that for $r=0,1,2,3$

$$||(Sg-g)^{(r)}||_\infty \leq C||g^{(4)}||_\infty h^{4-r}$$

where the C is not dependent on f or h. Hall & Meyer also found exact numerical values for these Cs, and showed that these also held if the boundary conditions were taken to be $f''(a)=g''(a)$ and $f''(b)=g''(b)$. The bound on $||Sg-g||_\infty$ depends on h^4, and one may therefore expect that when one halves the maximum mesh size, one should expect an approximation error $\frac{1}{16}$ the size of the error in the old approximation.

This also suggests a way to estimate the error bound of other approximation methods by numerical analysis. If we are approximating a known function $g(x)$, we can produce approximations with varying h, and then measure the error $|f(x)-g(x)|$ and find whether it is proportional to h^q where q is some exponent to be determined in the numerical experiments.

However, despite the improvement in error bounds, the shape properties of the variation-diminishing spline approximation rarely hold when a function is approximated by spline interpolation. There is an example in [6] of a Hermite spline interpolation which is negative for some values in $[a,b]$, even though f is nonnegative in the entire interval, so the preservation of bounds does not hold.

Chapter 4

Control polygon approximation

In this chapter, we will describe and specify the equations involved in the problem of creating a control polygon which interpolates a given data set $\{x_i, y_i\}$ of size n, in order to produce a spline approximation. As this determines an n-dimensional spline space, the knot vector $\boldsymbol{\tau}$ will be of size $n + p + 1$.

Also, we will look at what equations arise when inserting more knots in the knot vector; as described in section 2.3 this will mean the control polygon will be closer to the spline. Even when the knot vector is expanded for the control polygon calculation, we will have to restrict ourselves to the subspace of splines valid on the original knot vector $\boldsymbol{\tau}$ in order to find a unique solution to the interpolation problem.

4.1 Formulation of the interpolation problem

Assume we have an interpolation problem with n known data points $g(x_i) = y_i$. We wish to interpolate this data set with a spline f in a spline space $\mathbb{S}_{p,\boldsymbol{\tau}}$ of dimension n, degree p, and knot vector $\boldsymbol{\tau}$. We have a number of interpolation choices, but the following discussion will concern a method of interpolation by the control polygon $\Gamma_{p,\boldsymbol{t}}(f)$, where $\boldsymbol{t}$ is a knot vector that includes, but is not limited to, the knots in $\boldsymbol{\tau}$, and $\Gamma_{p,\boldsymbol{t}}(f)$ is as defined in section 2.3.

If we insert more knots and produce a new knot vector $\boldsymbol{t} \supseteq \boldsymbol{\tau}$, we find that f is also a member of the higher-dimensional space $\mathbb{S}_{p,\mathbf{t}}$ with the control polygon $\Gamma_{p,\boldsymbol{t}}(f)$. Now let $\boldsymbol{t}^j$ be a sequence of knot vectors such that $\boldsymbol{t}^{j+1} \supset \boldsymbol{t}^j$ and $\boldsymbol{t}^1 \supset \boldsymbol{\tau}$, and let f_j be a sequence of functions in $\mathbb{S}_{p,\boldsymbol{\tau}}$ such that $\Gamma_{p,\boldsymbol{t}^j}(f_j)(x_i) := y_i$. In theorem 1, we cited that as the distance between knots of a spline is reduced, the control polygon will converge to the spline. As $j \to \infty$, the distance $h_{\max} := \max_i t^j_{i+1} - t^j_i$ will converge to 0, and f_j will therefore converge towards a spline function $\hat{f}$ such that $\hat{f}(x_i) := y_i$.

4.2 The linear equation system

Let us now see how we can find an f with conditions on the control polygon in $\mathbb{S}_{p,\boldsymbol{t}}$ as discussed in the previous section, that is

$$\Gamma_{p,\boldsymbol{t}}(f)(x_i) := y_i. \tag{4.1}$$

We must first express the value of the control polygon at any $x \in [t_1^\star, t_m^\star]$ in terms of the coefficients of f in $\mathbb{S}_{p,\boldsymbol{t}}$. Let these coefficients be known as $\boldsymbol{d} = \{d_i\}$, with $i = 1, \ldots, m$. Let $\nu \in [1, m]$ such that $x \in [t_\nu^\star, t_{\nu+1}^\star]$, then there exists a $\lambda \in [0, 1]$ such that $x = t_\nu^\star + \lambda\left(t_{\nu+1}^\star - t^\star\right)$. Then, as the control polygon is defined to be linear between $(t_\nu^\star, d_\nu)$ and $(t_{\nu+1}^\star, d_{\nu+1})$, we get the following expression for the value of the control polygon at x:

$$\Gamma_{p,\boldsymbol{t}}(f)(x) = (1 - \lambda)d_\nu + \lambda d_{\nu+1}.$$

Now, let ν_i be such that $x_i \in [t_{\nu_i}^\star, t_{\nu_{i+1}}^\star]$. There then exists an associated $\lambda_i := \frac{x_i - t_{\nu_i}^\star}{t_{\nu_i+1}^\star - t_{\nu_i}^\star}$, and we can rewrite the interpolation problem (4.1) as

$$(1 - \lambda_i)d_{\nu_i} + \lambda_i d_{\nu_{i+1}} = y_i.$$

This can be written in matrix form as

$$\boldsymbol{Ld} = \boldsymbol{y} \tag{4.2}$$

with the entries of the matrix $\boldsymbol{L}$ set to be $L_{i,\nu_i} = 1 - \lambda_i$ and $L_{i,\nu_i+1} = \lambda_i$. The $\boldsymbol{L}$ matrix is an $n \times m$-matrix, and thus this is an overdetermined system if we were to solve it on the entire spline space spanned by $\boldsymbol{t}$.

However, we also wish to restrict ourselves to the spline space $\mathbb{S}_{p,\boldsymbol{\tau}}$, where the spline f is expressed as follows:

$$f = \boldsymbol{B}_{\boldsymbol{\tau}}^T \boldsymbol{c}$$

where $\boldsymbol{B}_{\boldsymbol{\tau}}^T$ is the B-spline matrix of multiplied $\boldsymbol{R}$-matrices and $\boldsymbol{c}$ are the coefficients. We must now convert this expression for f into the higher-dimensional spline space with knot vector $\boldsymbol{t}$.

We refer back to 2.4 and the knot insertion matrix $\boldsymbol{A}_{\boldsymbol{t}}^{\boldsymbol{\tau}}$ from $\boldsymbol{\tau}$ to $\boldsymbol{t}$, which has the property that $A_{\boldsymbol{t}}^{\boldsymbol{\tau}}\boldsymbol{c} = \boldsymbol{d}$.

We thus have in the higher spline space that

$$f = B_{\boldsymbol{t}}^T \boldsymbol{A}_{\boldsymbol{t}}^{\boldsymbol{\tau}} \boldsymbol{c}$$

with $A_{\boldsymbol{t}}^{\boldsymbol{\tau}}\boldsymbol{c} = \boldsymbol{d}$.

As we found in (4.2), the interpolation problem can be expressed as $\boldsymbol{Ld} = \boldsymbol{y}$, and we can now bring these problems together, forming the following linear problem for the coefficients $\boldsymbol{c}$ in the n-dimensional spline space $\mathbb{S}_{p,\boldsymbol{\tau}}$:

$$\boldsymbol{L}\boldsymbol{A}_{\boldsymbol{t}}^{\boldsymbol{\tau}}\boldsymbol{c} = \boldsymbol{y}. \tag{4.3}$$

As $\boldsymbol{L}\boldsymbol{A}_{\boldsymbol{t}}^{\boldsymbol{\tau}}$ is an $n \times n$-matrix, we are left with a linear problem which has exactly one solution if $\boldsymbol{L}\boldsymbol{A}_{\boldsymbol{t}}^{\boldsymbol{\tau}}$ is nonsingular, and otherwise no solutions (apart from the trivial $\boldsymbol{y} = 0$).

Chapter 5

Numerical analysis of specific approximation problems

In this chapter, we will describe more specific implementations of the interpolation problems described in chapter 4, and describe some properties of these implementions, including accuracy, convergence speed, and shape preservation. We will describe both quadratic and cubic spline approximations.

5.1 Norms and error measures

First, we need to define what we mean by the error in an approximation method. The simplest measure is to look at $|f(x) - g(x)|$ at points $x \in [\tau_{p+1}, \tau_{n+1}]$, which is the interval on which the spline f is defined. We thus get a maximum metric

$$d(f, g) = \sup\{|f(x) - g(x)| : x \in [\tau_{p+1}, \tau_{n+1}]\}.$$

A good estimate of this metric, assuming the function g is sufficiently continuous, can be found numerically by sampling f and g at a number of points x_j in the interval $[\tau_{p+1}, \tau_{n+1}]$. Let $e := \max x_{j+1} - x_j$, then $|d(f, g) - (\max_j f(x_j) - g(x_j))| \leq \max_{x \in [\tau_{p+1}, \tau_{n+1}]} |f'(x)|e$. Now, if f is a C^2-continuous function, then there exists some $C \in \mathbb{R}$ such that $C > |f'(x)|$ for all x in $[\tau_{p+1}, \tau_{n+1}]$ and hence $|f'(x)|e$ converges to 0 as e converges to 0. Thus, $|d(f, g) - (\max_j f(x_j) - g(x_j))|$ also converges to 0, and $\max_j |f(x_j) - g(x_j)|$ should be a sufficiently good estimate of $d(f, g)$.

In the remaining sections of this chapter, the measure of the quantity

$$\max_j |f(x_j) - g(x_j)|$$

at $1,000$ different points x_j will be taken to be our estimate of the error of the approximation method to a specific function g.

5.2 Quadratic splines

First, we shall examine the case of $p = 2$ and approximate some common polynomial and trigonometric functions with a number of conceivable methods.

5.2.1 The simplest method

The naive interpolation method would be to simply let $\boldsymbol{\tau} = \boldsymbol{t}$ and force the interpolation conditions on the control polygon $\Gamma_{p,\boldsymbol{\tau}}(f)$. This means that the knot insertion matrix $\boldsymbol{A}$ is simply the identity matrix. We also need to choose our interpolation points x_i ; given that the control polygon is piecewise linear between the knot averages, it seems natural to interpolate at these knot averages. Thus, any linear polynomial will be interpolated exactly, up to the machine's numerical precision.

We recall that in the variation-diminishing spline approximation, which we described in section 3.1, the coefficients c_j are set equal to $g(\tau_j^\star)$. The definition of the control polygon now gives that $\Gamma_{p,\boldsymbol{\tau}}(f)(\tau_j^\star) = g(\tau_j^\star)$, which is exactly the interpolation problem defined in section 4.1 with $\boldsymbol{t} = \boldsymbol{\tau}$ and interpolation points $x_i = \tau_i^\star$. This means that this simple method will be identical to the variation-diminishing spline approximation method.

The linear polynomials are interpolated exactly, as the control polygon of a linear spline is exactly the linear spline. However, there are small errors in the interpolation of quadratic polynomials. An approximation of $g(x) = x^2 + 2x + 1$ on a 3-regular knot vector with 11 internal knots uniformly spaced in the interval $[0, 1]$, gives an error which is plotted in figure 5.1.

5.2.1.1 Approximation of polynomials

We will try to systematically analyse the error $||f - g||$ in this approximation on a number of polynomial functions. The numerical estimate $\max_j |f(x_j) - g(x_j)|$ as outlined in section 5.1 with the x_js uniformly distributed $[\tau_{p+1}, \tau_{n+1}]$ will be the measure of the error; we also define $j_{\max}$ to be the integer value such that $|f(x_{j_{\max}}) - g(x_{j_{\max}})| = \max_j |f(x_j) - g(x_j)|$, and $x_{\max} := x_{j_{\max}}$.

To estimate what kind of error bound we will be expecting to find, we will cite a theorem from [6]. Let $g \in C^{p+1}[a, b]$ and assume we have a spline space $\mathbb{S}_{p,\boldsymbol{\tau}}$. There is a there exists an $f \in \mathbb{S}_{p,\boldsymbol{\tau}}$ such that

$$d(f, g) \leq D_p h^{p+1} ||D^{p+1} g||_{\infty,[a,b]} \tag{5.1}$$

where D_p is a constant depending only on p.

As we are only looking at a subspace of functions in $\mathbb{S}_{p,\boldsymbol{\tau}}$, and that we found in figure 5.1 that only linear polynomials are reproduced faithfully by this method, it seems that the error is not bounded by $||D^3 g||_\infty$. We thus first attempt to determine whether there is a linear relationship between $||g''||$ and $||f - g||$ in this error bound.

The results of these experiments and other information about the polynomials are shown in table 5.1.

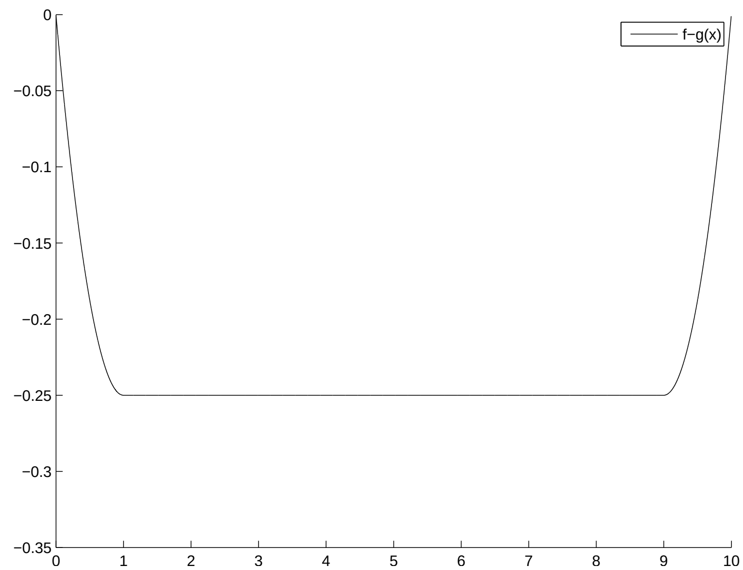

Figure 5.1: Measuring the error $f - g(x)$ when approximating $g(x) = x^2 + 2x + 1$ with the simplest approximation method, with $x \in [0, 1]$.

g	$g^{''}$	$\|\|f-g\|\|$	$x_{\max}$
x^2+2x+1	2	2.5×10^{-3}	0.88
$2x^2+2x+1$	4	5.0×10^{-3}	0.84
$3x^2+x+1$	6	7.5×10^{-3}	0.88
$3x^2+6x+2$	6	7.5×10^{-3}	0.68
$2x^3+7x^2-4x-2$	$12x+14$	3.1×10^{-2}	0.90
$3x^3+7x^2-4x-2$	$18x+14$	3.8×10^{-2}	0.90
$-8x^3+4x^2+4x+1$	$-48x+8$	4.4×10^{-2}	0.90
$2x^4+6x^3-3x^2+4x+1$	$-24x^2+36x-6$	9.3×10^{-3}	0.75
$x^4+2x^3-6x^2+10x-4$	$12x^2+12x-12$	1.3×10^{-2}	0.10
$3x^4-4x^3+7x^2-3x-1$	$36x^2-24x+14$	2.7×10^{-2}	0.90

Table 5.1: Error measures for certain polynomial functions with their derivatives.

$\|\|g''\|\|$	$\|g''(x_{\max})\|$	$\|\|f-g\|\|$
2	2	2.5×10^{-3}
4	4	5.0×10^{-3}
6	6	7.5×10^{-3}
6	6	7.5×10^{-3}
26	25	3.1×10^{-2}
32	30	3.8×10^{-2}
40	35	4.4×10^{-2}
7.5	7.5	9.3×10^{-3}
12	11	1.3×10^{-2}
26	22	2.7×10^{-2}

Table 5.2: Measuring the numerical values of the errors and certain second derivatives.

The hypothesis of a linear relationship between $||g''||$ and $|f-g|$ is tested in table 5.2 and the error is plotted against $||g''||$ in figure 5.2. We see that the relationship is not exactly linear; for example the errors obtained for $2x^3+7x^2-4x-2$ and $3x^4-4x^3+7x^2-3x-1$ are different, though both have the same norm for the 2nd derivative. However, if we plot the error against $|g''(x_{\max})|$ as shown in figure 5.3, the relationship seems to be linear (ignoring rounding errors arising from measurements taken to 2 significant figures).

Given that the location of the maximum error can not be known with certainty before doing the calculation, it is natural to use $||g''||$ as a proxy; we know that $|g''(x_{\max})| < ||g''||$ by definition. However, when analysing the numerical data, it seems useful to be aware of the difference.

5.2.1.2 Estimating the effect of mesh size

Having first examined the effects of the $||D^q g||_{\infty,[a,b]}$ condition on the error bound, it seems natural to investigate the effect of changing $h = \max_i \tau_{i+1} - \tau_i$. We do this by defining a sequence $h_1, \cdots, h_6$ where $h_1 = \frac{1}{10}$ and $h_i = \frac{h_{i-1}}{2}$. Then, if $||f-g|| \leq C|h|^q$, where C is dependent on other factors kept constant in the experiment, we should see that

$$\frac{||f_i - g_i||}{||f_{i-1} - g_{i-1}||} = \left(\frac{|h_i|}{|h_{i-1}|}\right)^q. \tag{5.2}$$

As $\left(\frac{|h_i|}{|h_{i-1}|}\right) = \frac{1}{2}$ for all elements in the sequence, we can solve for q if this brings a similar result for all elements of the sequence. Using the polynomial $2x^4+6x^3-3x^2+4x+1$, and otherwise keeping the experiment as in section 5.2.1.1 we obtain the data shown in table 5.3. The observed values of $\frac{||f_i-g_i||}{||f_{i-1}-g_{i-1}||}$ are very close to $\frac{1}{4}$, which implies that $q=2$ in this case. We can thus hypothesise that the error bounds for these methods will be on the form

$$||f-g|| \leq Dh^q ||D^q f||_{\infty,[a,b]}. \tag{5.3}$$

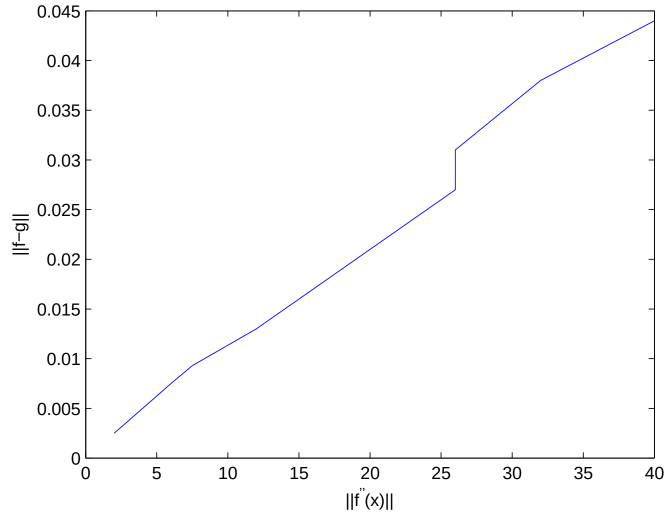

Figure 5.2: Plotting the error $||f - g||$ against $||g^{''}||$

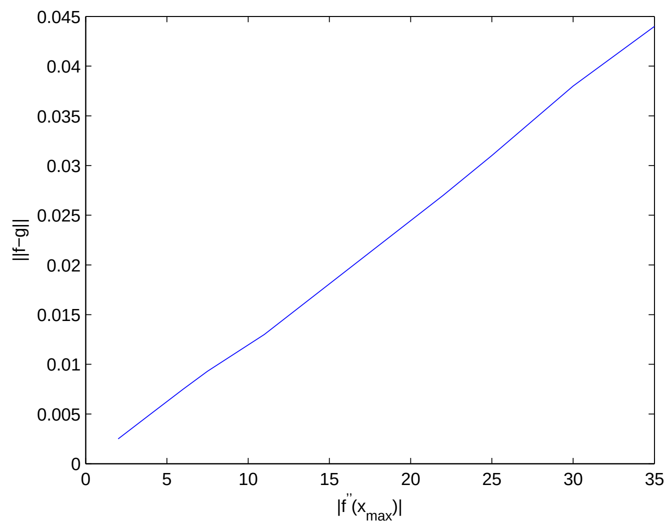

Figure 5.3: Plotting the error $||f - g||$ against $|g^{''}(x_{\max})|$

i	$\frac{\|\|f_i - g_i\|\|}{\|\|f_{i-1} - g_{i-1}\|\|}$
2	2.51×10^{-1}
3	2.50×10^{-1}
4	2.50×10^{-1}
5	2.50×10^{-1}
6	2.50×10^{-1}

Table 5.3: Observations of $\frac{||f_i - g_i||}{||f_{i-1} - g_{i-1}||}$ for varying items of the sequence $h_i = \frac{h_{i-1}}{2}$. All simulations were performed with uniform knot vectors.

5.2.2 Changing interpolation points

Instead of interpolation at the knot averages, which for a uniform knot vector are not uniformly spaced in most cases, we can attempt to change the x_i in $[t_1^\star, t_m^\star]$ to obtain better interpolations. The method described in 5.2.1 was accurate at the edges of the interval, but the error in the main part of the interpolation interval changed, see figure 5.1.

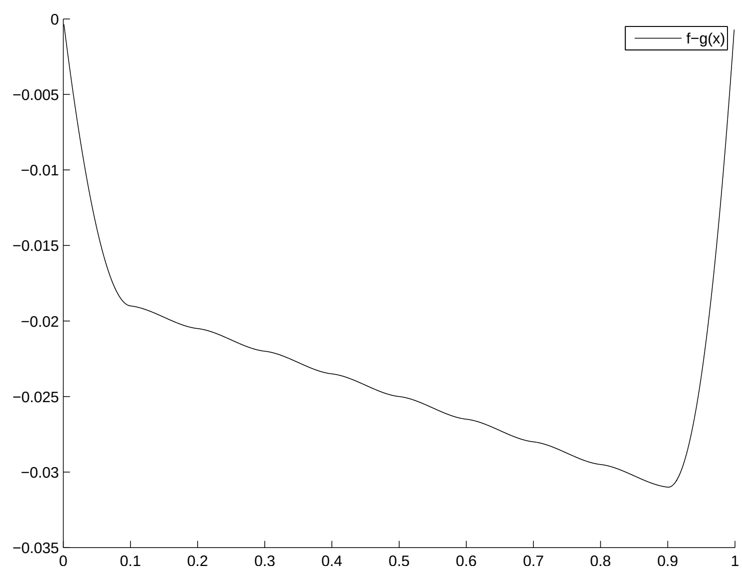

Figure 5.4: Measuring the error $f - g(x)$ when approximating $g(x) = 2x^3 + 7x^2 - 4x - 2$ with the simplest approximation method, with $x \in [0, 1]$.

Local characteristics of the function g may change this; consider figure 5.4 where the error in the approximation of $g(x) = 2x^3 + 7x^2 - 4x - 2$ is plotted. $g''(x)$ has a local maximum at 1, and $g'''(x)$ is always positive, and hence the error grows towards 1 - though it shrinks to 0 near the end points.

The set of knot averages of the regular knot vector with 11 internal knots in

$[0, 1]$ is

$$\boldsymbol{x} = \{0, \frac{1}{20}, \frac{3}{20}, \frac{5}{20}, \frac{7}{20}, \frac{9}{20}, \frac{11}{20}, \frac{13}{20}, \frac{15}{20}, \frac{17}{20}, \frac{19}{20}, 1\}$$

while the uniform interpolation points will be

$$\boldsymbol{x} = \{0, \frac{1}{11}, \frac{2}{11}, \frac{3}{11}, \frac{4}{11}, \frac{5}{11}, \frac{6}{11}, \frac{7}{11}, \frac{8}{11}, \frac{9}{11}, \frac{10}{11}, 1\}$$

Changing the interpolation point vector reduces the maximum distance between any $x \in [0, 1]$ and the closest interpolation point x_i from $\frac{1}{20}$ to $\frac{1}{22}$. As illustrated in figure 5.5 this reduced the maximum error from 3.10×10^{-2} to 2.04×10^{-2}, and in more than half of the interval the error is smaller than 10^{-2}.

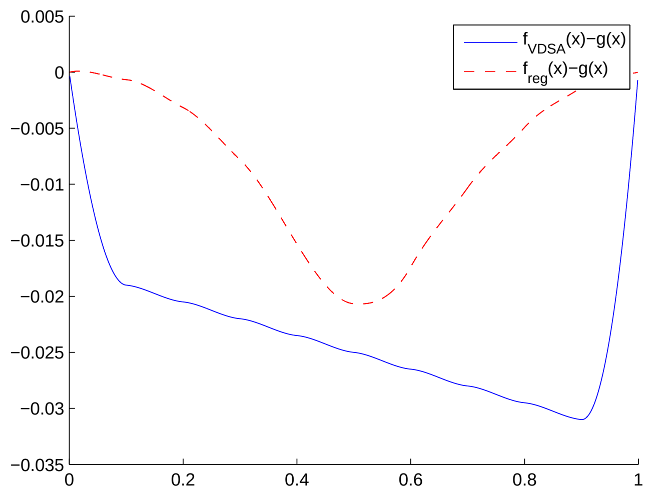

Figure 5.5: Comparing $f - g(x)$ when approximating $g(x) = 2x^3 + 7x^2 - 4x - 2$ with the variation-diminishing approximation method (solid line) and the method utilising uniform interpolation points (dashed line) for $x \in [0, 1]$.

5.2.3 Spline interpolation

Now, let us consider an expanded knot vector $\boldsymbol{t} \neq \boldsymbol{\tau}$, that is, augmenting $\boldsymbol{\tau}$ with extra knots to approach f closer. It is possible to produce another well-studied algorithm by this method, namely direct spline interpolation. We shall describe this method for general p, and then implement it for $p = 2$.

Consider a knot with multiplicity at least p, that is, $t_i = t_{i+1} = \cdots = t_{i+p-1}$. Then

$$B_{i-1}(t_i) = 1 \tag{5.4}$$

and $B_j(t_i) = 0$ for all $j \neq i-1$. Also, $t^\star_{i-1} = t_i$. Also, if $\Gamma_{p,\boldsymbol{\tau}}(f)(t_i) = g(t_i)$, then

$$g(t_i) = c_{i-1}. \tag{5.5}$$

Thus from (5.4) and (5.5), combined with the definition of splines in definition 1 we have that

$$f(t_i) = c_{i-1} = \Gamma_{p,\tau}(f)(t_i). \tag{5.6}$$

Thus, the spline and the control polygon agree at knots with multiplicity p.

If we have a knot vector $\boldsymbol{\tau}$ and wish to interpolate at the knot averages $\boldsymbol{\tau}^\star$, we must make sure that the knot averages repeat at least p times in the new knot vector $\boldsymbol{t}$. Assume our knot vector $\boldsymbol{\tau}$ is uniform, such that $\tau_{i+1} = \tau_i + h$ for all internal knots, then

$$\tau^\star_i = \frac{\tau_i + \tau_i + h + \tau_i + 2h + \ldots + \tau_i + ph}{p}$$
$$\tau^\star_i = \frac{p(\tau_i) + (\sum_1^p i)h}{p}$$
$$\tau^\star_i = \frac{p(\tau_i) + (2p + \frac{(p)(p-1)}{2})h}{p},$$

which leads to the following expression for the knot averages in a uniform knot vector:

$$\tau^\star_i = \tau_i + (2 + \frac{(p-1)}{2}h). \tag{5.7}$$

In the quadratic case, then, $\tau^\star_i = \tau_i + (2 + \frac{(1)}{2}h)$, and so these knot averages do not intersect - except for possibly at the end points, but as the knot vector is 3-regular, we already know the spline will interpolate the function at the end points.

Let us now attempt to approximate the cubic polynomial $g(x) = 2x^3 + 7x^2 - 4x - 2$ with this spline interpolation method. By decreasing h as in section 5.2.1.2, we obtain table 5.4. In the table, we have divided the errors by the errors in the previous element in the sequence, as done in section 5.2.1.2, we see that the observed values of $\frac{||f_i - g_i||}{||f_{i-1} - g_{i-1}||}$ are very close to $\frac{1}{8}$, and so the error bound converges as h^3, one order better than the VDSA algorithm in section 5.2.1.2. This cubic relationship of the error against h is shown in the plot in figure 5.6.

This characteristic seems to be specific for the spline interpolation method. We can test this by inserting random knots in $\boldsymbol{t}$. We start with a uniform knot vector $\boldsymbol{\tau}$ with $h = 10^{-1}$ and interpolate in the knot averages. If we now generate a random set of new knots using MATLAB's random number generator - call them $\hat{\boldsymbol{t}}$ - we obtain a new knot vector $\boldsymbol{t} = [\boldsymbol{\tau}, \hat{\boldsymbol{t}}]$.

To measure the effect of halving h, we can keep our old random knots, but add new in the middle of each knot interval - such that $t^i_{2j} = \frac{t^{i-1}_j + t^{i-1}_{j+1}}{2}$. Now we are unlikely to have repeating knots, and only the control polygon will interpolate, not the spline.

h	$\|\|f_i - g_i\|\|$	$\|\| \frac{\|\|f_i - g_i\|\|}{\|\|f_{i-1} - g_{i-1}\|\|}$
$1e-1$	9.62×10^{-5}	
$5e-2$	1.20×10^{-5}	0.125
$2.5e-2$	1.50×10^{-6}	0.125
$1.25e-2$	1.88×10^{-7}	0.125
$6.25e-3$	2.35×10^{-8}	0.125
$3.125e-3$	2.94×10^{-9}	0.125

Table 5.4: Observations of $||f - g||$ for varying h using spline interpolation. All simulations were performed with uniform knot vectors.

n	$\|\|f_i - g_i\|\|$	$\|\| \frac{\|\|f_i - g_i\|\|}{\|\|f_{i-1} - g_{i-1}\|\|}$
0	2.12×10^{-3}	
11	6.25×10^{-4}	3.0×10^{-1}
22	1.93×10^{-4}	3.1×10^{-1}
44	4.30×10^{-5}	2.2×10^{-1}
88	1.21×10^{-5}	2.8×10^{-1}
176	3.44×10^{-6}	2.8×10^{-1}

Table 5.5: Observations of $||f - g||$ for inserting n random knots into the new knot vector.

The results are shown in table 5.5. We do not get as clean a rate of convergence as in the spline interpolation case - probably due to the random spacing of the knots - but the error bound for this method seems to be of order h^2, as in the variation diminishing method.

5.2.4 Intermediate methods

A method that now suggests itself is to augment $\boldsymbol{\tau}$ with exactly 1 knot at each knot average point $\tau_i^\star$, which should then be an intermediate method between the variation-diminishing spline approximation algorithm and the spline interpolation algorithm. We still let $p = 2$, which means the knot averages for a

h	$\|\|f_i - g_i\|\|$	$\|\| \frac{\|\|f_i - g_i\|\|}{\|\|f_{i-1} - g_{i-1}\|\|}$
$1e-2$	9.62×10^{-5}	
$5e-3$	1.20×10^{-5}	0.125
$2.5e-3$	1.50×10^{-6}	0.125
$1.25e-3$	1.88×10^{-7}	0.125
$6.25e-4$	2.35×10^{-8}	0.125
$3.125e-4$	2.94×10^{-9}	0.125

Table 5.6: Observations of $||f - g||$ for varying h using the intermediate method. All simulations were performed with uniform knot vectors.

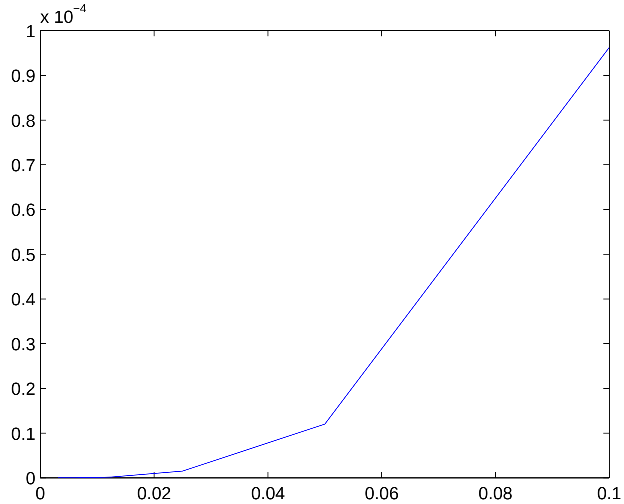

Figure 5.6: Measuring $|f_i - g_i(x)|$ when approximating $g(x) = 2x^3 + 7x^2 - 4x - 2$ with a spline approximation method for different values $h \in [1 \times 10^{-5}, 1 \times 10^{-1}]$.

uniform knot vector $\boldsymbol{\tau}$, as shown in the discussion in 5.2.3, are exactly $\tau_{i+2} + \frac{1}{2}h$; this means the augmented knot vector $\boldsymbol{t}$ will be uniform with spacing $\frac{h}{2}$.

Performing the same experiment and approximating $g(x) = 2x^3 + 7x^2 - 4x - 2$ with uniform, 3-regular knots with varying spacing h_i such that $h_i = \frac{h_{i-1}}{2}$, we find the error bounds shown in table 5.6.

Interestingly, however, with uniform knots, this intermediate method also ends up with reproducing almost exactly the same spline coefficients $\boldsymbol{c}$; the difference between the coefficients of this spline interpolation method and this method is of size around 10^{-13}, and the two methods have almost exactly the same error measures, as can be seen by comparing tables 5.4 and 5.6. It seems as though obtaining a spline which interpolates at the knot averages $\boldsymbol{\tau}^\star$, we only need to include 1 repetition of the knot averages, and not 2 as theoretically suggested at the start of section 3.2.

5.3 Cubic splines

Many of the experiments in section 5.2 have a natural analogue in the spline space $\mathbb{S}_{3,t}$ of cubic splines. We will now show the results of error measurements with uniform knot vectors, but with cubic splines.

5.3.1 Variation-diminishing approximation

We start by attempting to measure the errors of the variation-diminishing method, where the value of the knot averages $\boldsymbol{\tau}^\star$ are slightly different - they are now the averages of $\tau_i, \ldots, \tau_{i+3}$ rather than $\tau_i, \ldots, \tau_{i+2}$. We again use a $p+1$-regular and uniform knot vector, with spacing h between the internal knots, so this time the end points a and b are repeated $p+1 = 4$ times at each point, and we vary h from $\frac{1}{10}$ to $\frac{1}{320}$ by halving h.

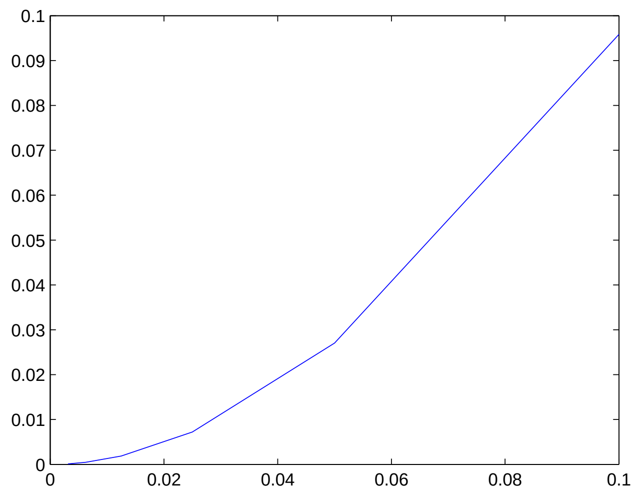

Figure 5.7: Measuring $|f_i - g_i(x)|$ when approximating $g(x) = 4x^4 + 2x^3 + 7x^2 + 4x - 2$ with the variation -diminishing spline approximation method for different values $h \in [1 \times 10^{-4}, 1 \times 10^{-1}]$.

Despite increasing the degree of the spline, there is no corresponding increase in the degree of $|h|$ in the error bound of this method. The results of an experiment of interpolating the quartic polynomial $g(x) = 4x^4 + 2x^3 + 7x^2 + 4x - 2$ are shown in figure 5.7. We see that if h is halved, the error is roughly $\frac{1}{4}$ of the previous level, and we can hypothesise that the error bound is of the form

$$||f - g|| \leq Dh^2 ||D^2 f||_{\infty,[a,b]} \tag{5.8}$$

which is the same as that we observed with quadratic splines.

h_i	$\|\|f_i - g_i\|\|$	$\frac{h_i}{h_{i-1}}$
$1e-1$	9.58×10^{-2}	
$5e-2$	2.71×10^{-2}	0.28
$2.5e-2$	7.21×10^{-3}	0.27
$1.25e-2$	1.86×10^{-3}	0.26
$6.25e-3$	4.73×10^{-4}	0.25
$3.125e-3$	1.19×10^{-4}	0.25

Table 5.7: Observations of $||f-g||$ for varying h using variation-diminishing approximation for $g(x) = 4x^4 + 2x^3 + 7x^2 + 4x - 2$. All simulations were performed with uniform knot vectors.

h_i	$\|\|f_i - g_i\|\|$	$\frac{h_i}{h_{i-1}}$
$1e-1$	3.00×10^{-5}	
$5e-2$	1.88×10^{-6}	0.0625
$2.5e-2$	1.17×10^{-7}	0.0625
$1.25e-2$	7.33×10^{-9}	0.0625
$6.25e-3$	4.58×10^{-10}	0.0625
$3.125e-3$	2.86×10^{-11}	0.0625

Table 5.8: Observations of $||f-g||$ for varying h using spline approximation for $g(x) = 4x^4 + 2x^3 + 7x^2 + 4x - 2$. All simulations were performed with uniform knot vectors.

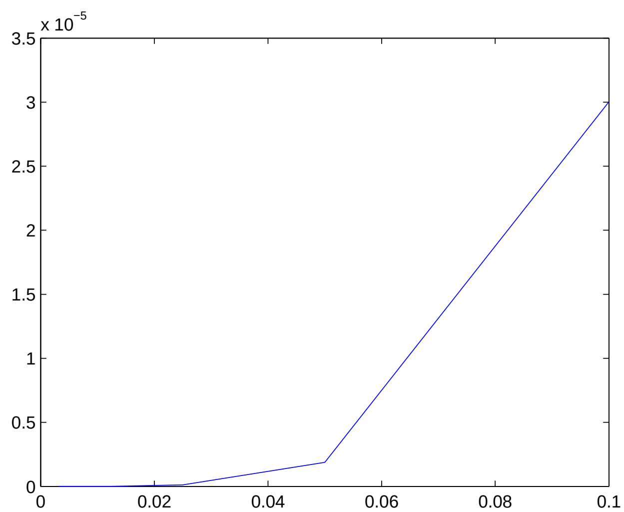

Figure 5.8: Measuring $|f_i - g_i(x)|$ when approximating $g(x) = 4x^4 + 2x^3 + 7x^2 + 4x - 2$ with the spline interpolation method for different values $h \in [1 \times 10^{-4}, 1 \times 10^{-1}]$.

h_i	$\|\|f_i - g_i\|\|$	$\|\|f_i - g_i\|\|$
$1e-1$	9.58×10^{-2}	8.29×10^{-2}
$5e-2$	2.71×10^{-2}	2.52×10^{-2}
$2.5e-2$	7.21×10^{-3}	6.96×10^{-3}
$1.25e-2$	1.86×10^{-3}	1.83×10^{-3}
$6.25e-3$	4.73×10^{-4}	4.69×10^{-4}
$3.125e-3$	1.19×10^{-4}	1.19×10^{-4}

Table 5.9: Observations of $||f-g||$ for varying h using variation-diminishing approximation (column 2) and an intermediate method with two extra knots (column 3) for $g(x) = 4x^4 + 2x^3 + 7x^2 + 4x - 2$. All simulations were performed with uniform knot vectors.

5.3.2 Spline interpolation

Now, let us look at spline interpolation in the knot averages. We have from (5.7) in section 5.2.3 that $\tau_i^\star = \tau_i + (2 + \frac{(p-1)}{2}h)$ for the internal knots of a uniform knot vector. As $p = 3$ is odd, this works out to be $\tau_{i+2+\frac{(p-1)}{2}}$.

Therefore, we now only need to insert $p-1$ of the knot averages, as they already occur once in the knot vector - except near the end points, where the assumption in section 5.2.3 that $\tau_{i+1} = \tau_i + h$ is not correct. So to obtain a knot vector with 3 repetitions of the knot averages, our new augmented knot vector becomes $\boldsymbol{t} = [\boldsymbol{\tau}, \boldsymbol{\tau}, \boldsymbol{\tau}, \tau_2^\star, \tau_{n-1}^\star]$.

We again use the quartic polynomial $g(x) = 4x^4+2x^3+7x^2+4x-2$. Plotting the error against the mesh size h, as shown in figure 5.8, we see that this is not a quadratic relationship; the data of errors are shown in table 5.8. Repeated halving of h now gives a reduction of $\frac{1}{16}$ in the error, and this method thus seems to have an error bound of the form

$$||f-g|| \leq Dh^4 ||D^4 f||_{\infty,[a,b]} \tag{5.9}$$

and that spline interpolation methods in general have an error bound on the form

$$||f-g|| \leq Dh^{p+1} ||D^{p+1} f||_{\infty,[a,b]}. \tag{5.10}$$

5.3.3 Intermediate methods

As in the case of quadratic splines, intermediate methods with 1 repetition (adding the two missing knot averages near the end points) or 2 repetitions of the knot averages can also be determined. As we have now created methods with error bounds less than h^4 and h^2 respectively, it may be possible through such a method to create an error bound dependent on h^3.

However, as in the quadratic case, the method which repeats the knot averages exactly 2 - or $p-1$ - times reproduced the spline interpolation exactly, while inserting the two knots $\tau_2^\star$ and $\tau_{n-1}^\star$ merely improved the error bound for large h. For small h such as $h = 3.125e-3$, there is very little difference, see table 5.9.

$t_i^\star$	Coefficients of the variation-diminishing method	Coefficients of the intermediate method	Coefficients of the spline interpolation method
0	0	0	0
$\frac{1}{30}$	0.03333	0.03333	0.03333
$\frac{1}{10}$	0.09983	0.09994	0.09999
$\frac{2}{10}$	0.19867	0.19867	0.19900
$\frac{3}{10}$	0.29552	0.29552	0.29601
$\frac{4}{10}$	0.38942	0.38942	0.39007
$\frac{5}{10}$	0.47943	0.47943	0.48023
$\frac{6}{10}$	0.56464	0.56464	0.56558
$\frac{7}{10}$	0.64422	0.64422	0.64529
$\frac{8}{10}$	0.71736	0.71736	0.71855
$\frac{9}{10}$	0.78333	0.78405	0.78463
$\frac{29}{30}$	0.82300	0.82333	0.82346
1	0.84147	0.84147	0.84147

Table 5.10: Observations of c_i and the knot averages for various approximations of $g(x) = \sin(x)$.

5.4 Shape-preserving properties

Let us also look briefly at the values of the spline coefficients, this time using the non-polynomial function $\sin(x)$. We saw in chapter 3 that having control of the coefficients led to some shape-preserving properties. Therefore, their exact value may be of interest when attempting to draw conclusions about the approximation method. This also allows us to look at how well the approximation method performs when applied to non-polynomial functions.

Let us, as in most of the previous examples, take a 4-regular knot vector $\boldsymbol{\tau}$ with internal knots uniformly spaced between 0 and 1. To produce an approximation with observable errors, we will let $h = 0.1$. We will compare the three methods described in the previous sections of this chapter, the variation-diminishing method with $\boldsymbol{\tau} = \boldsymbol{t}$, the spline interpolation method with $\boldsymbol{t} = [\boldsymbol{\tau}, \boldsymbol{\tau}, \boldsymbol{\tau}, \tau_2^\star, \tau_{n-1}^\star]$, and the intermediate method of inserting $\tau_2^\star$ and $\tau_{n-1}^\star$ to repeat all knot averages exactly once in the control polygon.

The coefficients found in this experiment are shown in table 5.10 and the difference between the methods are shown graphically in figure 5.9. We see that the method of inserting knots at $\tau_2^\star$ and $\tau_{n-1}^\star$ only improves the accuracy around that point; unlike spline interpolation which improves the accuracy for the entire interval - likely a function of the spline interpolation being a method where knots are inserted into $\boldsymbol{t}$ throughout the interval.

We can also conclude from table 5.10, however, that the spline coefficients c_2 and c_{n-1} of the intermediate method are larger than those of the variation-diminishing method. This indicates that the shape-preserving properties of chapter 3 do not hold for this intermediate method.

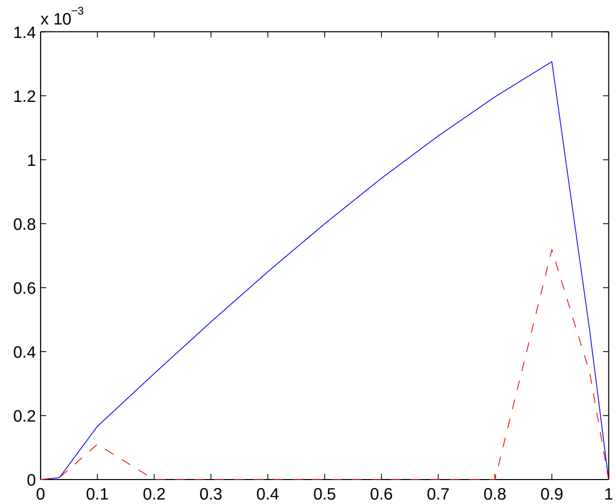

Figure 5.9: The difference between the coefficients of two spline approximations of $\sin(x)$, plotted against the knot averages $t_i^\star$. Here c_i are the coefficients of the variation diminishing method. the solid line plots $|\hat{c}_i - c_i|$ for $\hat{\boldsymbol{c}}$ being the coefficients of the spline interpolation method, and the difference between the variation diminishing method and the spline interpolation method, and the dashed line plots $|\tilde{c}_i - c_i|$ for $\tilde{\boldsymbol{c}}$ being the coefficients of the intermediate method with interpolation at each knot average.

Chapter 6

Conclusion

Unfortunately, the numerical experiments in chapter 5 suggest that the new methods do not have significant advantages over the well-known existing methods. The method intended to be an intermediate method between the variation-diminishing method and spline approximation had the same relatively poor convergence rate of the variation-diminishing method, both in the quadratic and cubic case, but lost the shape-preserving properties of this method.

From the experiments in section 5.4 it seems likely that this method, even for higher degrees, will have incremental benefits where the knots are inserted, but will do little to improve the exponent q in the error bound. However, in the cubic case, it was not possible to create a method where the augmented knot vector $\boldsymbol{t}$ included at least 1 repetition of the knot averages that were not in the original knot vector $\boldsymbol{\tau}$, without producing a spline which interpolated the function g exactly. This may be possible in a higher degree setting, which may also allow for a method which has a convergence exponent between $q = 2$ of the variation-diminishing method and $q = p + 1$ of the spline interpolation method.

In the quadratic case, we also looked at the choice of interpolation points in the control polygon and that it improved the accuracy of the method; however, the exponent q in the error bound remained the same, so it was merely an incremental improvement, and may be specific to the polynomial functions we examined. Also, it may not always be possible to pick and choose interpolation points in order to minimise the error, and so this may not be a very practical exercise.

Further research could be done to determine the number of calculation operations required to find the control polygon implementation of spline interpolation, compared to solving over the entire spline space. Another research point into these methods could be how to choose knots $\boldsymbol{\tau}$ and $\boldsymbol{t}$, with knot insertion matrix $\boldsymbol{A}$ and interpolation condition matrix $\boldsymbol{L}$ as described in section 4.2 such that the linear system $\boldsymbol{LAc} = \boldsymbol{y}$ has a solution.

Bibliography

[1] I. J. Schoenberg, *On spline functions*. In: O. Shisha (ed.), *Inequalities*, proceedings, Vol. 1, Academic Press, 1967, pp. 255-291.

[2] M. J. Marsden, *An identity for spline functions with applications to variation-diminishing spline approximation*. In: *Journal of Approximation Theory*, Vol. 3, No. 1, Elsevier, 1970, pp. 7-49.

[3] C. de Boor, *On Calculating with B-Splines*. In: *Journal of Approximation Theory*, Vol. 6, No. 1, Elsevier, 1972, pp. 50-62.

[4] M. J. Marsden, *A Voronovskaya theorem for variation-diminishing spline approximation*. In: *Canadian Journal of Mathematics*, Vol. 38, No. 5, Canadian Mathematical Society, 1986, pp. 1081-1093.

[5] L. L. Schumaker, *Spline Functions - Basic Theory*. 1981, John Wiley & Sons Inc. (USA), pp. 123-24.

[6] T. Lyche, K. Mørken, *Spline Methods - Draft*. 2011, University of Oslo, lecture notes in spline methods for the 2011 spring semester. URL:
`http://www.uio.no/studier/emner/matnat/ifi`
`/INF-MAT5340/v11/undervisningsmateriale/book.pdf`
Retrieved 6 December 2012.

[7] S. Karlin, J. M. Karon, *A Variation-Diminishing Generalized Spline Approximation Method*. In: *Journal of Approximation Theory*, Vol. 1, No. 3, Elsevier, 1968, pp. 255-268.

[8] C. A. Hall, W. W. Meyer, *Optimal error bounds for cubic spline interpolation*. In: *Journal of Approximation Theory*, Vol. 16, No. 2, Elsevier, 1976, pp. 105-122.