Univariate subdivision schemes for noisy data with geometric applications

Nira Dyn · Allison Heard · Kai Hormann · Nir Sharon

Abstract

We introduce and analyse univariate, linear, and stationary subdivision schemes for refining noisy data by fitting local least squares polynomials. This is the first attempt to design subdivision schemes for noisy data. We present primal schemes, with refinement rules based on locally fitting linear polynomials to the data, and study their convergence, smoothness, and basic limit functions. Then, we provide several numerical experiments that demonstrate the limit functions generated by these schemes from initial noisy data. The application of an advanced local linear regression method to the same data shows that the methods are comparable. In addition, several extensions and variants are discussed and their performance is illustrated by examples. We conclude by applying the schemes to noisy geometric data.

Citation Info

Journal
Computer Aided Geometric Design

Volume
37, August 2015

Pages
85–104

1 Introduction

In recent years, subdivision schemes have become an important tool in many applications and research areas, including animation, computer graphics, and computer aided geometric design, just to name a few [1, 20]. A subdivision scheme generates values associated with the vertices of a sequence of nested meshes, with a dense union, by repeated application of a set of local refinement rules. These rules determine the values associated with a refined mesh from the values associated with the coarser mesh. The subdivision scheme is convergent if the generated values converge uniformly to the values of a continuous function, for any set of initial values.

The particular class of interpolatory schemes consists of schemes with refinement rules that keep the values associated with the coarse mesh and only generate new values related to the additional vertices of the refined mesh. An important family of interpolatory schemes is the family of Dubuc–Deslauriers (DD) schemes [5].

Intensive studies have been carried out recently on the generalization of subdivision schemes to more complicated data such as manifold valued data [22, 23], matrices [21], sets [10], curves [17], and nets of functions [4]. In [6] subdivision schemes have been used in a multi-resolution fashion to remove heavy-tail noise. In this paper, we propose a way how to approximate a function from its noisy samples by subdivision schemes.

The linear and symmetric refinement rules of the DD schemes and their dual counterparts [11] are based on *local polynomial interpolation*. These schemes are stationary in the sense that the same rules are applied at all localities in all subdivision steps, and their approximation order is determined by the degree of the local interpolating polynomials.

In this paper we generalize this approach and propose linear and symmetric refinement rules based on *local polynomial approximation*, where the polynomial is determined by a least squares fit to the data. We call these schemes *least squares schemes*. The least squares schemes are designed to fit noisy data. Indeed, our numerical experiments indicate that in some cases these schemes outperform an advanced linear regression method.

A very recent paper [18] computes refined values by local ℓ_1 optimization rather than by local least squares. The lack of explicit expressions for the refined values of the ℓ_1 optimization enables experimental results only, which are compared with the performance of our schemes.

The least squares schemes and their tensor-products can also deal with geometric data, consisting of contaminated samples of curves and of surfaces. The performance of such schemes is demonstrated in the last section on two examples of curves and two examples of surfaces.

The paper is organized as follows. We start by introducing the simplest case of least squares schemes in Section 2. These schemes are based on primal refinement rules and on best fitting linear polynomials to symmetric data points. This is a one parameter family of schemes, with the number of data points as the parameter. We prove convergence and smoothness of these schemes and investigate properties of their basic limit functions. The construction of least squares schemes based on best fitting polynomials of higher degrees and on dual refinement rules is postponed to Section 4. In Section 3 we review a statistical model for fitting noisy data, analyse the suitability of the primal least squares schemes of degree 1 for dealing with this kind of data, and provide several numerical examples. Further numerical examples for primal schemes based on best fitting polynomials of higher degrees are presented in Section 4.4. Section 5 shows the application of the least squares schemes and their tensor-product to geometrical data. Throughout this paper we use several well-known properties of least squares polynomials. A short survey of these properties and a method for the efficient evaluation of our schemes are given in Appendix A.

2 Primal least squares schemes of degree 1

In this paper we consider the univariate setting. We denote by $\boldsymbol{f}^k = (f_i^k)_{i\in\mathbb{Z}}$ the data at refinement level $k \in \mathbb{N}_0$. We assume that the initial data $\boldsymbol{f}^0 = (f_i^0)_{i\in\mathbb{Z}}$ is given at the integers $\mathbb{Z}$ and that f_i^k is associated with the dyadic point $t_i^k = 2^{-k} i$. The main idea of least squares subdivision is to generate the data at level $k+1$ by evaluating a polynomial that locally fits the data at level k in a symmetric neighbourhood.

In particular, we use polynomials that best fit the data in the least squares sense. That is, for given data $y_1,\dots,y_m$ at nodes $x_1,\dots,x_m$, we are interested in the polynomial p_d of degree d that minimizes the sum of squared residuals,

$$\sum_{i=1}^{m} (p_d(x_i) - y_i)^2. \tag{1}$$

For $d < m$ this problem has a unique solution and in Appendix A we provide a summary of the relevant theory, which also includes the case $d \geq m$.

We start by considering the simplest least squares subdivision schemes corresponding to the case $d = 1$, which we denote by S_n for $n \geq 1$. Such a scheme generates the data at level $k+1$ as follows. On the one hand, the value f_{2i}^{k+1}, which replaces f_i^k, is determined by fitting a linear polynomial to the $2n-1$ data values in a symmetric neighbourhood around t_i^k and evaluating it at the associated dyadic point $t_i^k = t_{2i}^{k+1}$. On the other hand, the scheme computes the new value f_{2i+1}^{k+1} between f_i^k and f_{i+1}^k by evaluating at $t_{2i+1}^{k+1} = (t_i^k + t_{i+1}^k)/2$ the linear least squares polynomial with respect to the data at the nearest $2n$ nodes. In this construction the parameter n controls the locality of the scheme and we study its effect in Section 3.

For the case $d = 1$ and equidistant nodes $x_i = a + ih$, let p_1^* be the linear least squares polynomial which minimizes (1). The value of p_1^* at the centre $c = (x_1 + \cdots + x_m)/m$ of the nodes is

$$p_1^*(c) = (y_1 + \cdots + y_m)/m.$$

Thus, the refinement rules of S_n turn out to be

$$f_{2i}^{k+1} = \frac{1}{2n-1} \sum_{j=-n+1}^{n-1} f_{i+j}^k \qquad \text{and} \qquad f_{2i+1}^{k+1} = \frac{1}{2n} \sum_{j=-n+1}^{n} f_{i+j}^k. \tag{2}$$

Consequently, the symbol [8] of the scheme is

$$a_n(z) = \frac{1}{2n} \sum_{j=-n+1}^{n} z^{2j-1} + \frac{1}{2n-1} \sum_{j=-n+1}^{n-1} z^{2j}. \tag{3}$$

It follows from the symmetry of the nodes determining the linear least squares polynomials, that $a_n(z) = a_n(1/z)$, hence the scheme is odd symmetric [12]. As the data at level $k+1$ depends on at most $2n$ values at level k, we conclude that S_n is a primal $2n$-point scheme. The masks of the first three schemes are

$$\begin{aligned}
\boldsymbol{a}_1 &= \qquad [1,2,1] \qquad\quad /2,\\
\boldsymbol{a}_2 &= \quad [3,4,3,4,3,4,3] \quad /12,\\
\boldsymbol{a}_3 &= [5,6,5,6,5,6,5,6,5,6,5]/30.
\end{aligned}$$

n	2	3	4	5	6	7	8	9	10
ρ_n	1.649	1.777	1.816	1.794	1.786	1.776	1.771	1.761	1.753

Table 1: Lower bounds ρ_n on the Hölder regularity of the schemes S_n.

Note that the scheme S_1 is the interpolating 2-point scheme, which generates piecewise linear functions in the limit.

2.1 Convergence and smoothness

Following the usual definition of convergence in [9, Chapter 2], we denote the limit of a convergent subdivision scheme S for initial data $\boldsymbol{f}^0$ by $S^\infty \boldsymbol{f}^0$.

Theorem 1. *The least squares subdivision scheme S_n is convergent for $n \geq 1$.*

Proof. The explicit form of the symbol in (3) implies that $a_n(1) = 2$ and $a_n(-1) = 0$, which are necessary conditions for S_n to be convergent [8, Proposition 2.1]. In addition, since the coefficients of the symbol in (3) are all positive, and there are at least three such coefficients, it follows from [2, Theorem 3.3] that the scheme is convergent. □

Following the analysis in [8], we define

$$q_n(z) = \frac{a_n(z)}{1+z} = \frac{1}{2n(2n-1)}\left(\sum_{j=-n+1}^{n-1} (n-j) z^{2j-1} + \sum_{j=-n+1}^{n-1} (n+j) z^{2j} \right), \tag{4}$$

which is the symbol of the difference scheme associated with S_n. The norm of this scheme,

$$\begin{aligned} \left\| S_{[q_n]} \right\|_\infty &= \max\left\{ \frac{1}{2n(2n-1)} \sum_{j=-n+1}^{n-1} |n-j|, \frac{1}{2n(2n-1)} \sum_{j=-n+1}^{n-1} |n+j| \right\} \\ &= \frac{1}{2n(2n-1)} \sum_{j=1}^{2n-1} j = \frac{1}{2}, \end{aligned}$$

is the least possible, as in the case of the uniform B-spline schemes, indicating "quickest" possible convergence. The structure of q_n further reveals that the limit functions generated by S_n are C^1.

Theorem 2. *The least squares subdivision scheme S_n generates C^1 limit functions for $n \geq 2$.*

Proof. It is known [8, Theorems 3.2 and 3.4] that in order to prove the theorem, it is sufficient to show that the scheme with symbol $2q_n$ is convergent. By (4),

$$2q_n(1) = 2 \qquad \text{and} \qquad 2q_n(-1) = 0,$$

hence $S_{[2q_n]}$ satisfies the necessary conditions for convergence. As in the proof of Theorem 1 we conclude that the scheme $S_{[2q_n]}$, $n \geq 2$ is convergent, and therefore S_n, $n \geq 2$ generates C^1 limit functions. □

The statement in Theorem 2 is confirmed by the numerical results presented in Table 1, which were obtained by using 16 iterations of the algorithm in [13] to compute lower bounds on the Hölder regularity. In addition, it is easy to verify that $q_n'(-1) < 0$ and therefore $(1+z)^2$ is not a factor of $q_n(z)$ or equivalently that $(1+z)^3$ is not a factor of $a_n(z)$. Thus, the scheme S_n does not generate C^2 limits from any initial data [12].

2.2 The basic limit function

Let us denote by $\boldsymbol{\delta}$ the sequence which is zero everywhere except at 0, where it is 1. The *basic limit function* of the convergent subdivision scheme S_n is then defined as

$$\phi_n = S_n^\infty \boldsymbol{\delta}. \tag{5}$$

Some examples of ϕ_n for small values of n are shown in Figure 1.

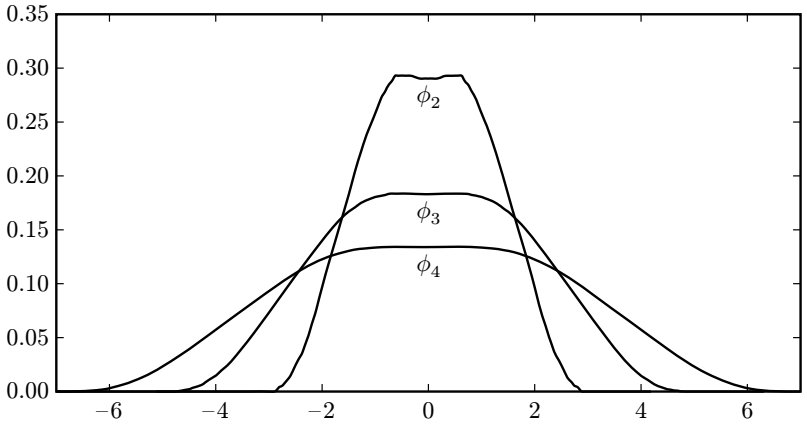

Figure 1: Basic limit functions of the schemes S_2, S_3, and S_4.

Many properties of a linear subdivision scheme can be derived from its basic limit function. In particular, due to linearity, the limit function generated from the initial data $\boldsymbol{f}^0 = (f_i^0)_{i\in\mathbb{Z}}$ by the scheme S_n has the form

$$(S_n^\infty \boldsymbol{f}^0)(x) = \sum_{j\in\mathbb{Z}} f_j^0 \phi_n(x-j). \tag{6}$$

Our first observation is that the support of ϕ_n is $[-2n+1, 2n-1]$, because S_n is a primal $2n$-point scheme [5]. Moreover, ϕ_n is positive inside its support, because the coefficients of the mask $\boldsymbol{a}_n$ are positive in the mask's support, and ϕ_n has the partition of unity property

$$\sum_{j\in\mathbb{Z}} \phi_n(x-j) = 1, \tag{7}$$

due to the reproduction of constant polynomials by S_n.

The simple structure of $\boldsymbol{a}_n$ further allows us to derive several interesting properties regarding the values of the basic limit function ϕ_n at the integers. These values are of importance, because they constitute the filter which operates on the initial data and generates the final values at the integers. Taking into account that ϕ_n is continuous and therefore vanishes at the end points of its support, we conclude from (6) that the limit at the integers $k \in \mathbb{Z}$ is

$$(S_n^\infty \boldsymbol{f}^0)(k) = \sum_{j=-2n+2}^{2n-2} f_{k-j}^0 \phi_n(j). \tag{8}$$

The non-zero values of ϕ_n at the integers constitute an eigenvector $\boldsymbol{v} = \big(\phi_n(-2n+2), \ldots, \phi_n(2n-2)\big)$ corresponding to the eigenvalue 1 of the transposed subdivision matrix [8], which in this case is the $(4n-3)\times(4n-3)$ column stochastic, two-slanted band matrix

$$A_n = \begin{pmatrix}
r & s & 0 & 0 & 0 & & 0 & 0 & 0 \\
r & s & r & s & 0 & \cdots & 0 & 0 & 0 \\
r & s & r & s & r & & 0 & 0 & 0 \\
 & & \vdots & & & \ddots & & \vdots & \\
r & s & r & s & r & & r & s & 0 \\
r & s & r & s & r & \cdots & r & s & r \\
0 & s & r & s & r & & r & s & r \\
 & & \vdots & & & \ddots & & \vdots & \\
0 & 0 & 0 & 0 & 0 & \cdots & 0 & s & r
\end{pmatrix}$$

with entries $r = 1/(2n-1)$ and $s = 1/(2n)$.

The odd symmetry of the mask $\boldsymbol{a}_n$ guarantees that ϕ_n is a symmetric function. Thus, the eigenvector $\boldsymbol{v}$ is also symmetric, as indicated by the structure of A_n. Taking these symmetries into account, we get that the

vector $\tilde{\boldsymbol{v}} = \big(\phi_n(-2n+2),\dots,\phi_n(0)\big)$ is an eigenvector corresponding to the eigenvalue 1 of the $(2n-1)\times(2n-1)$ matrix

$$\tilde{A}_n = \begin{pmatrix}
r & s & 0 & 0 & 0 & & 0 & 0 & 0 & 0 & 0 \\
r & s & r & s & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\
r & s & r & s & r & & 0 & 0 & 0 & 0 & 0 \\
 & & \vdots & & & \ddots & & & \vdots & & \\
r & s & r & s & r & & r & s & 0 & 0 & 0 \\
r & s & r & s & r & & r & s & r & s & 0 \\
r & s & r & s & r & \cdots & r & s & r & 2s & r \\
r & s & r & s & r & & r & 2s & 2r & 2s & r \\
r & s & r & s & r & & 2r & 2s & 2r & 2s & r \\
 & & \vdots & & & \ddots & & & \vdots & & \\
r & s & r & 2s & 2r & & 2r & 2s & 2r & 2s & r \\
r & 2s & 2r & 2s & 2r & \cdots & 2r & 2s & 2r & 2s & r \\
2r & 2s & 2r & 2s & 2r & & 2r & 2s & 2r & 2s & r
\end{pmatrix}.$$

The particular structure of $\tilde{A}_n$ allows us to derive the following observation.

Proposition 3. *The values of ϕ_n at the non-positive integers in its support are strictly increasing,*

$$0 < \phi_n(-2n+2) < \phi_n(-2n+3) < \cdots < \phi_n(-1) < \phi_n(0).$$

Moreover,

$$\phi_n(-n) = \frac{n-1}{2n-1}\phi_n(0).$$

Proof. Note that each row of $\tilde{A}_n$ is equal to the previous row plus at least one positive term. Since $\tilde{\boldsymbol{v}}$ satisfies $\tilde{A}_n\tilde{\boldsymbol{v}} = \tilde{\boldsymbol{v}}$ and its components $\tilde{v}_i = \phi_n(i-2n+1)$, $i = 1,\dots,2n-1$, are positive, the latter must be strictly increasing.

To establish the second statement, consider the $(n-1)$-th and the last row of $\tilde{A}_n$,

$$\tilde{\boldsymbol{\alpha}}_{n-1} = (r,s,r,s,\dots,r,s,0) \qquad \text{and} \qquad \tilde{\boldsymbol{\alpha}}_{2n-1} = (2r,2s,2r,2s,\dots,2r,2s,r),$$

and note that

$$\tilde{\boldsymbol{\alpha}}_{2n-1} = 2\tilde{\boldsymbol{\alpha}}_{n-1} + (0,0,\dots,0,r).$$

Then, since

$$\tilde{v}_{n-1} = \tilde{\boldsymbol{\alpha}}_{n-1}\tilde{\boldsymbol{v}}$$

and

$$\tilde{v}_{2n-1} = \tilde{\boldsymbol{\alpha}}_{2n-1}\tilde{\boldsymbol{v}} = 2\tilde{\boldsymbol{\alpha}}_{n-1}\tilde{\boldsymbol{v}} + r\,\tilde{v}_{2n-1} = 2\,\tilde{v}_{n-1} + r\,\tilde{v}_{2n-1},$$

the second statement follows directly from the definition of $\tilde{\boldsymbol{v}}$, because $r = 1/(2n-1)$. □

By the symmetry of ϕ_n, the statements of Proposition 3 hold analogously for the values of ϕ_n at the non-negative integers. As an immediate consequence we have

$$\phi_n(j) < \frac{1}{2}\phi_n(0), \qquad |j| \geq n, \tag{9}$$

as well as the following bounds on $\phi_n(0)$.

Corollary 4. *The value of $\phi_n(0)$ satisfies*

$$\frac{1}{3n-2} < \phi_n(0) < \frac{1}{n-1}. \tag{10}$$

Proof. The upper bound follows from (7) and Proposition 3, because

$$1 = \sum_{|j|\leq 2n-2} \phi_n(j) > \sum_{|j|\leq n} \phi_n(j) > (2n+1)\phi_n(-n) = \frac{(2n+1)(n-1)}{2n-1}\phi_n(0).$$

Using (9), we further have

$$1 = \sum_{n\le|j|\le 2n-2} \phi_n(j) + \sum_{|j|<n} \phi_n(j) < (2n-2)\frac{1}{2}\phi_n(0) + (2n-1)\phi_n(0),$$

leading to the lower bound. □

Proposition 3 and its consequences clarify the properties of ϕ_n at the integers, which are confirmed by the examples in Figure 1. A further analysis of ϕ_n reveals more details, in particular about the asymptotic behaviour for large n as well as an improvement in the upper bound in (10).

Theorem 5. *The basic limit function ϕ_n and its derivative ϕ_n' converge uniformly to the zero function as n grows. More specifically,*

$$\|\phi_n\|_\infty \sim \frac{1}{n}$$

and

$$\|\phi_n'\|_\infty \sim \frac{1}{n^2}.$$

Proof. We first observe that the masks corresponding to the refinement rules (2) are positive. Thus, for non-negative data such as $\boldsymbol{\delta}$ we have $\|S_n^{k_1}(\boldsymbol{\delta})\|_\infty \le \|S_n^{k_2}(\boldsymbol{\delta})\|_\infty$ for any integers $k_1 > k_2 > 0$. We can therefore bound $\|\phi_n\|_\infty$ from above,

$$\|\phi_n\|_\infty = \|S_n^\infty(\boldsymbol{\delta})\|_\infty \le \|S_n^1(\boldsymbol{\delta})\|_\infty = \frac{1}{2n-1} \sim \frac{1}{n}. \tag{11}$$

A similar behaviour holds for the derivative ϕ_n', which exists since S_n generates C^1 limits by Theorem 2. To see this, first recall the definition of q_n in (4), which implies the relation [8, Section 2.3]

$$\phi_n'(x) = S_{[2q_n]}^\infty(\Delta\boldsymbol{\delta})(x),$$

where Δ is the forward difference operator with $(\Delta\boldsymbol{\delta})_0 = -1$, $(\Delta\boldsymbol{\delta})_{-1} = 1$, and zero otherwise. This implies $\|\phi_n'\|_\infty \le \|S_{[2q_n]}\Delta\boldsymbol{\delta}\|_\infty$. Further note that $S_{[2q_n]}$ has a positive mask $(2\boldsymbol{q}_n)$ with coefficients

$$(2\boldsymbol{q}_n)_{2j-1} = \frac{1}{n(2n-1)}(n-j) \qquad \text{and} \qquad (2\boldsymbol{q}_n)_{2j} = \frac{1}{n(2n-1)}(n+j) \tag{12}$$

for $j = -n+1, \dots, n-1$ and $(2\boldsymbol{q}_n)_{2j-1} = (2\boldsymbol{q}_n)_{2j} = 0$ for $|j| \ge n$.

A direct calculation yields

$$\left|(S_{[2q_n]}\Delta\boldsymbol{\delta})_j\right| = \begin{cases} 1/n(2n-1), & \text{if } -2n-1 < j < 2n-2, \\ 1/n, & \text{if } j = -2n-1 \text{ or } j = 2n-2, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

From (12) and (13) we then conclude that each summand in

$$(S_{[2q_n]}^2\Delta\boldsymbol{\delta})_j = \sum_{i\in\mathbb{Z}} (2\boldsymbol{q}_n)_{j-2i}(S_{[2q_n]}\Delta\boldsymbol{\delta})_i \tag{14}$$

is of order $1/n^3$, except for at most one summand of order $1/n^2$. Since there are at most $2n-1$ non-zero terms in the sum (14), the order of the sum is $1/n^2$. Thus, we have $\|\phi_n'\|_\infty \le \|S_{[2q_n]}^2\Delta\boldsymbol{\delta}\|_\infty \sim 1/n^2$. □

Proposition 3 and Theorem 5 provide a good understanding of the basic limit function ϕ_n, which is supported by our numerical tests.

3 The schemes applied to noisy data

The schemes S_n for $n > 1$ are designed to deal with noisy data, which is confirmed by the following discussions and experiments. We first introduce a statistical model and then compare the performance of our schemes and an advanced local linear regression method.

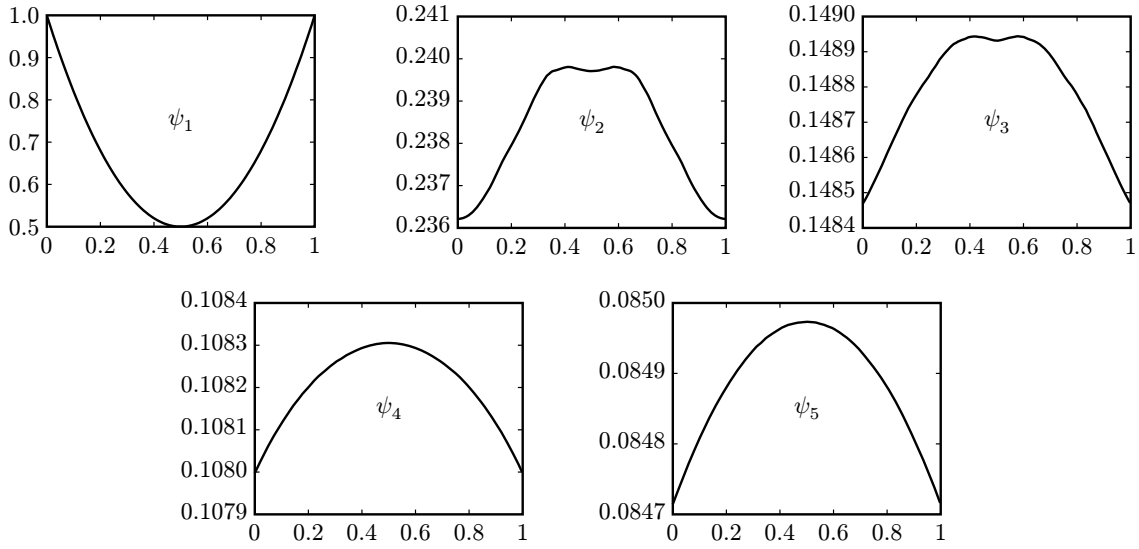

Figure 2: Plots of the functions ψ_n for $n = 1,\dots,5$. Note the different scale in each plot.

3.1 Statistical considerations

Let $f : \mathbb{R} \to \mathbb{R}$ be a continuous scalar function and suppose we are given a discrete set of noisy samples

$$y_i = f(ih) + \varepsilon_i, \quad i \in \mathbb{Z},$$

where $\{\varepsilon_i\}_{i\in\mathbb{Z}}$ are independent random variables, normally distributed with zero mean and variance σ^2. As an estimator $\hat{f}$ of f we use the limit (6) of S_n, that is,

$$\hat{f}(x) = \sum_{j\in\mathbb{Z}} y_j \phi_n(x-j). \tag{15}$$

Note that $\hat{f}(x)$ is a random variable and the estimation quality of $\hat{f}$ is given by the expectation of the squared error.

With E denoting the expectation operator, the "bias-variance decomposition" [16, Chapter 7] of the expected squared error for $x \in \mathbb{R}$ is

$$\mathrm{E}\big[(\hat{f}(x) - f(x))^2\big] = \sigma^2 \sum_{j\in\mathbb{Z}} \phi_n(x-j)^2 + \left(\sum_{j\in\mathbb{Z}} f(jh)\phi_n(x-j) - f(x) \right)^2. \tag{16}$$

The first term in (16) is the product of the variance of the noise σ^2 and the function

$$\psi_n(x) = \sum_{j\in\mathbb{Z}} \phi_n(x-j)^2. \tag{17}$$

The second term is the square of the deterministic approximation error corresponding to data without noise. We first study ψ_n and come back to the second term later.

It follows from (16) that the effect of the noise on the estimator $\hat{f}$ is small if ψ_n is small, which motivates us to further analyse ψ_n and establish upper bounds. First note that by (7) and the positivity of ϕ_n we have

$$\psi_n \le 1, \tag{18}$$

with strict inequality for $n > 1$, namely for non-interpolatory schemes. For the interpolatory scheme S_1, we have $\psi_1(x) = 1$ at $x \in \mathbb{Z}$, which matches the common knowledge that interpolation is not appropriate for noisy data. This behaviour is confirmed by Figure 2, which presents several numerical evaluations of ψ_n and indicates that ψ_n becomes smaller and tends to be the constant zero function as n grows. This is indeed the case, as the following summary of properties of ψ_n shows.

Theorem 6. *The function ψ_n in* (17) *is positive, symmetric, and periodic with period* 1. *Moreover,*

$$\|\psi_n\|_\infty \sim \frac{1}{n}$$

and

$$\|\psi_n'\|_\infty \sim \frac{1}{n^2}.$$

Proof. By definition, ψ_n is positive, periodic, and finite. The symmetry of ϕ_n implies the symmetry of ψ_n. In addition, we have that ψ_n is symmetric about $1/2$ due to the periodicity of ψ_n.

The first asymptotic bound follows from the definition of ψ_n in (17) after noting that only $4n-2$ terms in the sum are non-zero, and that each term is of order $1/n^2$ by Theorem 5. The second asymptotic bound follows by similar arguments using the chain rule, the explicit bound on ϕ_n in (11), and the asymptotic bound on ϕ_n' in Theorem 5,

$$|\psi_n'(x)| \le 2\sum_{j\in\mathbb{Z}} |\phi_n(x-j)\phi_n'(x-j)| \le 2\frac{2}{2n-1}\sum_{j\in\mathbb{Z}} |\phi_n'(x-j)| \sim \frac{1}{n^2}.$$

□

The second term of the expected squared error in (16) is the deterministic error or the approximation error. We use the approximation order as a standard measure for the quality of the approximation [13, Chapter 7]. For the case of schemes based on linear least squares polynomials, the approximation order is h^2, where h is the distance between the sampled points of the initial data. This observation follows from the polynomial reproduction property of our schemes, that is, the reconstruction of any linear polynomial from its samples.

In conclusion, there is a trade-off between the deterministic approximation error and the effect of the noise on the expected squared error. In particular, higher values of n decrease the effect of noise but increase the deterministic error due to averaging of the values $\{f_i\}_{i\in\mathbb{Z}}$ by weights with a large support.

3.2 Numerical examples

We illustrate the performance of some of the schemes by several numerical examples, starting from noisy data. We compare their performance with the algorithm of local linear regression (LLR) for local fitting of noisy data. This local estimator around a given data point x^* is obtained by including kernel weights into the least squares minimization problem in the neighbourhood of x^*,

$$\min_{\alpha,\beta} \sum_{i=0}^{n} \big(y_i - \alpha - \beta(x_i - x^*)\big)^2 \operatorname{Ker}(x_i - x^*).$$

This approach can be generalized to higher degree polynomials as well (see [15, Chapter 4] for more details). Although the concept of LLR is rather simple, it is one of the most important statistical approaches used.

We take the LLR variant which is based on the normal kernel with the kernel parameters chosen dynamically, and we compare it with the limits of several subdivision schemes with different support sizes, for various types of functions and levels of noise. The noise we consider is normally distributed and measured using the signal-to-noise ratio (SNR). The SNR is defined as the ratio between the L_2 norm of the signal (true function and additional noise) and the L_2 norm of the noise. Thus, when this ratio tends to one, the noise becomes as significant as the signal itself. The standard unit is decibel (dB) which is calculated in a logarithmic scale. We examine the range of (roughly) 1–20 dB and consider 1–5 dB a very high level, 5–7 a high level, and 10–12 a low level of noise. Noise levels >12 dB are considered negligible. In each example we plot the relative approximation error of LLR and the subdivision scheme, as a function of the noise levels. This relative error is defined as the ratio between the norm of the approximation error and the norm of the function.

In the first examples we consider the slowly varying function

$$f_1(x) = \sin\frac{x}{10} + \left(\frac{x}{50}\right)^2$$

and examine the three subdivision schemes S_3, S_5, and S_7. Due to the dynamic implementation of LLR, we can use it as a benchmark for all cases. As discussed in Section 3.1, the subdivision scheme S_3 with smaller

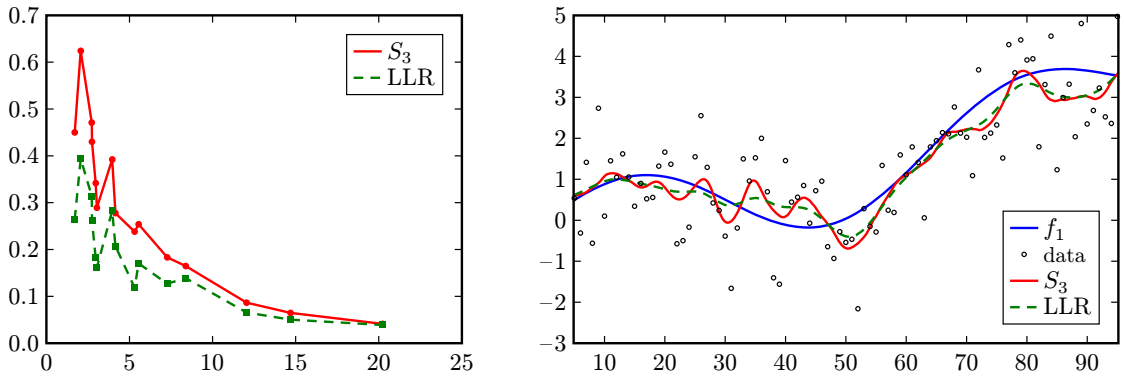

Figure 3: Comparison of S_3 and LLR for f_1. Left: Relative approximation error as a function of the SNR. Right: Reconstruction of f_1 from data with 6.5 dB noise.

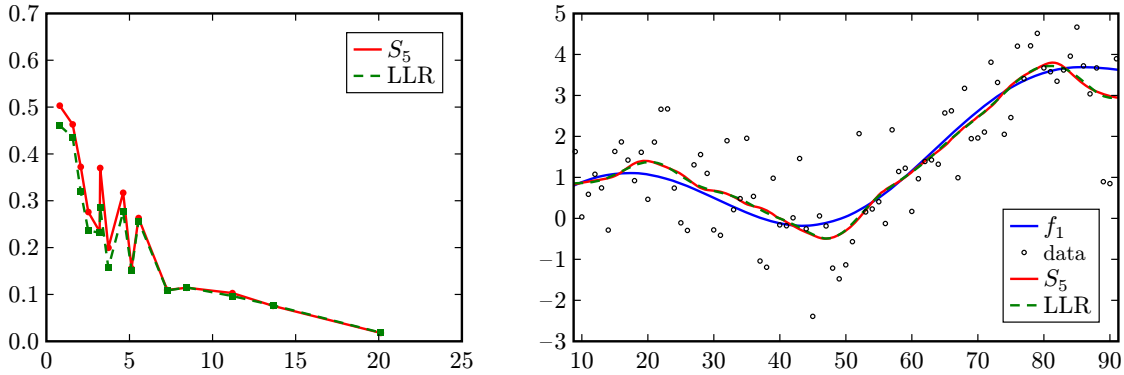

Figure 4: Comparison of S_5 and LLR for f_1. Left: Relative approximation error as a function of the SNR.. Right: Reconstruction of f_1 from data with 7 dB noise.

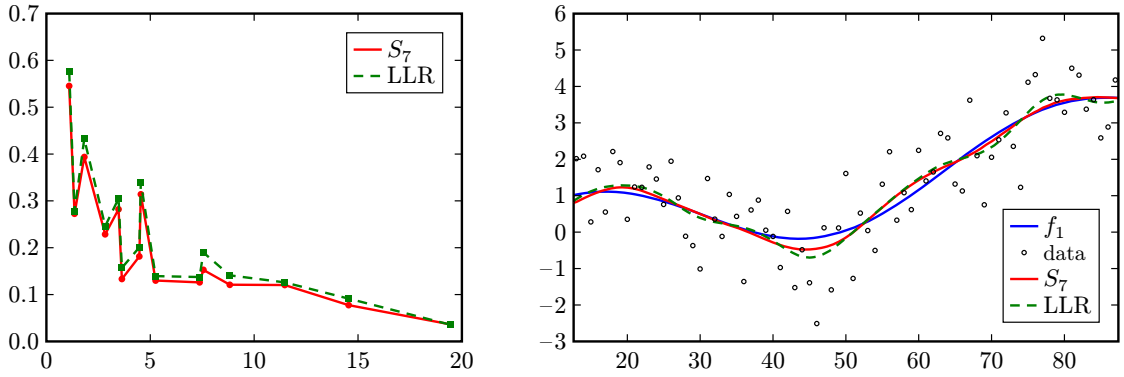

Figure 5: Comparison of S_7 and LLR for f_1. Left: Relative approximation error as a function of the SNR. Right: Reconstruction of f_1 from data with 7.9 dB noise.

support is more sensitive to the variance of the noise than S_5 and S_7. We observe in Figure 3 (left) that for all levels of noise, LLR gives a smaller reconstruction error. The difference in the actual function reconstruction for a specific noise level is illustrated in Figure 3 (right). The same presentation is repeated for S_5 and S_7 in Figures 4 and 5, respectively. For the slowly varying function f_1, the subdivision scheme S_5 behaves almost identically to LLR, while S_7 is even better. These trials match our theory which suggests that as the support gets larger, the corresponding function ψ_n becomes smaller, resulting in a weaker response to noise.

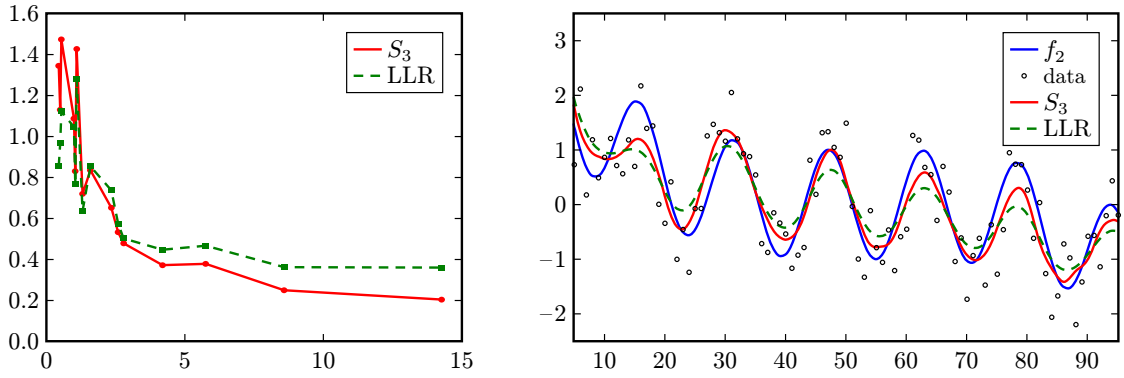

Figure 6: Comparison of S_3 and LLR for f_2. Left: Relative approximation error as a function of the SNR. Right: Reconstruction of f_2 from data with 6.2 dB noise.

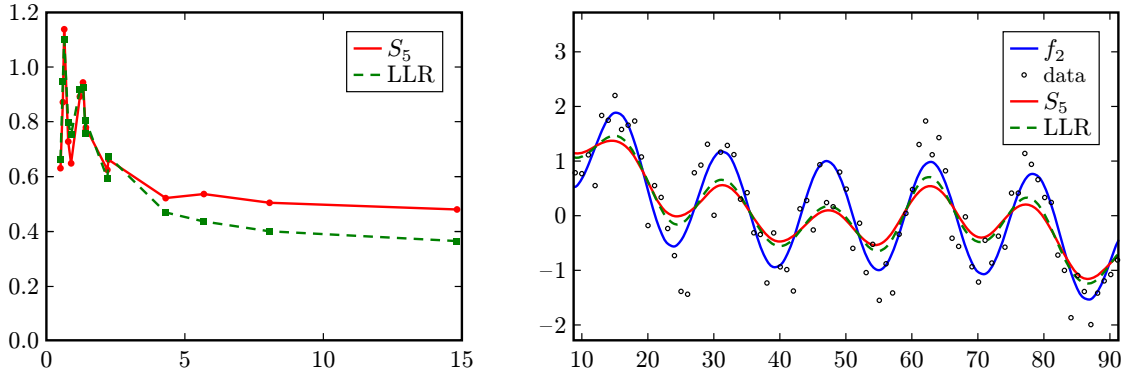

Figure 7: Comparison of S_5 and LLR for f_2. Left: Relative approximation error as a function of the SNR. Right: Reconstruction of f_2 from data with 8 dB noise.

In our second example we sample the oscillatory function

$$f_2(x) = \cos\frac{2x}{5} + \left(\frac{x}{40} - 1\right)^3$$

and compare LLR with S_3 and S_5. The results are presented in Figures 6 and 7 and show that the smaller support of S_3 makes it more suitable for these type of functions (except for extremely high noise), while S_5 provides inferior results for any reasonable level of noise.

To conclude, we observe from the numerical examples that there is a range of parameters for which a subdivision scheme outperforms LLR. Also, the numerical examples support our understanding about the trade-off between the effect of noise and the deterministic approximation error, as discussed in Section 3.1.

4 Extensions and variants

The family of primal least squares schemes of degree 1 can be extended in several ways. We first discuss the extension to dual schemes (Section 4.1), as well as minor variations of both primal and dual schemes (Section 4.2). A further extension relies on fitting least squares polynomials of higher degree (Section 4.3) and we provide a few numerical examples of such schemes (Section 4.4).

4.1 Dual least squares schemes of degree 1

The idea of the schemes S_n in Section 2 is to fit linear least squares polynomials and to evaluate them in a *primal* way, that is, at the points and the midpoints of the current mesh. Another option is to design subdivision schemes based on *dual* evaluation [12]. The dual least squares scheme $\tilde{S}_n$ is obtained by fitting a

n	2	3	4	5	6	7	8	9	10
$\bar{\rho}_n$	2.285	2.647	2.729	2.677	2.664	2.633	2.616	2.594	2.577

Table 2: Lower bounds $\bar{\rho}_n$ on the Hölder regularity of the schemes $\bar{S}_n$.

linear polynomial to the $2n$ data values at the points $t^k_{i-n+1},\ldots,t^k_{i+n}$ at level k and evaluating this polynomial at $1/4$ and $3/4$ between t^k_i and t^k_{i+1} to compute the new data f^{k+1}_{2i} and f^{k+1}_{2i+1}.

The refinement rules of $\bar{S}_n$ are slightly more complicated to derive than those of the primal schemes, but they still have a rather simple closed form,

$$f^{k+1}_{2i}=\frac{1}{2n}\sum_{j=-n+1}^{n}\left(1-\frac{6j-3}{8n^2-2}\right)f^k_{i+j}\qquad\text{and}\qquad f^{k+1}_{2i+1}=\frac{1}{2n}\sum_{j=-n+1}^{n}\left(1+\frac{6j-3}{8n^2-2}\right)f^k_{i+j}. \tag{19}$$

The corresponding symbol is

$$\bar{a}_n(z)=\frac{1}{2n}\sum_{j=-n}^{n-1}\left(1+z+\frac{6j+3}{8n^2-2}(1-z)\right)z^{2j}, \tag{20}$$

and it is easy to verify that $\bar{a}_n(z)z=\bar{a}_n(1/z)$, which confirms that $\bar{S}_n$ is an even symmetric scheme [12]. Overall we conclude that $\bar{S}_n$ is a dual $2n$-point scheme and the support of its basic limit function $\bar{\phi}_n$ is $[-2n,2n-1]$. The masks of the first three schemes are

$$\begin{aligned}\bar{\boldsymbol{a}}_1&=[1,3,3,1]/4,\\ \bar{\boldsymbol{a}}_2&=[7,13,9,11,11,9,13,7]/40,\\ \bar{\boldsymbol{a}}_3&=[55,85,61,79,67,73,73,67,79,61,85,55]/420,\end{aligned}$$

and we recognize $\bar{S}_1$ as Chaikin's corner cutting scheme [3].

The proofs of Theorems 1 and 2 carry over to the dual schemes, and so the limit functions generated by $\bar{S}_n$ are at least C^1 for $n\geq 1$. But unlike the primal schemes, the symbols of the dual schemes are divisible by $(1+z)^3$, and so they may potentially generate C^2 limits. However, there is no simple proof as for C^1 in Theorem 2, because the symbol $4\bar{a}_n(z)/(1+z)^2$ has negative coefficients. Table 2 lists lower bounds on the Hölder regularity of the first few schemes, computed using 16 iterations of the algorithm in [13], demonstrating that the limits of $\bar{S}_n$ are in fact C^2, at least for $2\leq n\leq 10$.

4.2 Variants of linear least squares schemes

In addition to the dual $2n$-point schemes $\bar{S}_n$, it is also possible to define dual $(2n+1)$-point schemes. These schemes fit a linear polynomial to the $2n+1$ data values in a symmetric neighbourhood around f^k_i and evaluate it at $1/4$ the distance to the left (right) neighbour to define the new data f^{k+1}_{2i-1} (f^{k+1}_{2i}). The resulting refinement rules are

$$f^{k+1}_{2i-1}=\frac{1}{2n+1}\sum_{j=-n}^{n}\left(1-\frac{3j}{4n(n+1)}\right)f^k_{i+j}\quad\text{and}\quad f^{k+1}_{2i}=\frac{1}{2n+1}\sum_{j=-n}^{n}\left(1+\frac{3j}{4n(n+1)}\right)f^k_{i+j},$$

and the support of the corresponding basic limit function is $[-2n-1,2n]$. The masks of the first three schemes of this kind are

$$\begin{aligned}n=1:&\quad[5,11,8,8,11,5]/24,\\ n=2:&\quad[6,10,7,9,8,8,9,7,10,6]/40,\\ n=3:&\quad[13,19,14,18,15,17,16,16,17,15,18,14,19,13]/112.\end{aligned}$$

Similarly, we can define primal $(2n+1)$-point schemes as variants of the primal $2n$-point schemes S_n. We simply replace the refinement rule for f^{k+1}_{2i} in (2) by

$$f^{k+1}_{2i}=\frac{1}{2n+1}\sum_{j=-n}^{n}f^k_{i+j}$$

and keep the rule for f_{2i+1}^{k+1}. For these schemes, the support of the basic limit function is $[-2n,2n]$, and the masks of the first three schemes are

$$\begin{aligned} n=1: &\qquad [2,3,2,3,2] &&/6,\\ n=2: &\qquad [4,5,4,5,4,5,4,5,4] &&/20,\\ n=3: &\,[6,7,6,7,6,7,6,7,6,7,6,7,6] &&/42. \end{aligned}$$

Adapting the proofs of Theorems 1 and 2, one can show that both variants generate C^1 limit functions, and our numerical results demonstrate that the dual $(2n+1)$-point schemes are even C^2 for $1 \le n \le 10$.

4.3 Least squares schemes of higher degree

The least squares schemes of degree 1 reproduce linear polynomials by construction, but they do not reproduce polynomials of any higher degree. So, their approximation order is only h^2, unless the data is being pre-processed [12]. We can improve this by using least squares polynomials of higher degrees $d > 1$.

To derive the refinement rules at level k, let $p_{n,i}^d$ be the least squares polynomial of degree d for the $2n-1$ data

$$(t_{i+j}^k, f_{i+j}^k), \qquad j = -n+1, \ldots, n-1$$

in a symmetric neighbourhood of t_{2i}^{k+1}, and let $\tilde{p}_{n,i}^d$ be the polynomial of degree d that fits the $2n$ data

$$(t_{i+j}^k, f_{i+j}^k), \qquad j = -n+1, \ldots, n$$

in a symmetric neighbourhood of t_{2i+1}^{k+1}. The polynomials $p_{n,i}^d$ and $\tilde{p}_{n,i}^d$ are well-defined for $d < 2n-1$ and $d < 2n$, respectively (see Appendix A.1).

The primal $2n$-point least squares scheme of degree d is then characterized by the refinement rules

$$f_{2i}^{k+1} = p_{n,i}^d(t_i^k) \qquad \text{and} \qquad f_{2i+1}^{k+1} = \tilde{p}_{n,i}^d\big((t_i^k + t_{i+1}^k)/2\big), \tag{21}$$

which simplifies to the rules in (2) for $d = 1$. The resulting subdivision scheme S_n^d reproduces polynomials of degree d by construction, and thus has approximation order h^{d+1}. It is well-defined for $d < 2n$, even though for $d = 2n-1$ the rule for f_{2i}^{k+1} is based on an underdetermined problem. In that case we get $f_{2i}^{k+1} = f_i^k$ (see Remark 9 in Appendix A.1), hence S_n^{2n-1} is the interpolating Dubuc–Deslauriers $2n$-point scheme.

As shown in Remark 12 in Appendix A.3, it is sufficient to consider only primal $2n$-point least squares schemes of even degree, because S_n^{2d} and S_n^{2d+1} are identical. This also means that the schemes of degree $2d$ reproduce polynomials of one degree more than expected by construction. This is in accordance to the observation in [12] that the reproduction of odd degree polynomials comes "for free" by the primal symmetry. In particular, this shows that the refinement rule of the interpolating 4-point scheme [7] for f_{2i+1}^{k+1} can be derived not only from fitting a cubic polynomial to the data $f_{i-1}^k, \ldots, f_{i+2}^k$, but also by fitting a quadratic polynomial in the least squares sense to the same data.

We can also generalize the construction in Section 4.1 and define the dual $2n$-point least squares scheme of degree d by the refinement rules

$$f_{2i}^{k+1} = \tilde{p}_{n,i}^d\big((3t_i^k + t_{i+1}^k)/4\big) \qquad \text{and} \qquad f_{2i+1}^{k+1} = \tilde{p}_{n,i}^d\big((t_i^k + 3t_{i+1}^k)/4\big), \tag{22}$$

which simplify to the rules in (19) for $d = 1$. Like S_n^d, the scheme $\bar{S}_n^d$ reproduces polynomials of degree d by construction and its approximation order is h^{d+1}. Moreover, the scheme $\bar{S}_n^{2n-1}$ is the dual $2n$-point scheme [11].

Similar constructions lead to primal and dual $(2n+1)$-point least squares schemes of degree d, but we omit the details as they are straightforward. Apart from the increased approximation order, these schemes also tend to have a higher smoothness. For example, we verified numerically that the schemes $\bar{S}_n^3$ generate C^3 limit functions for $n = 4$ and $n = 5$, but we do not recommend using them, because the rules become more complicated and the benefit of using them for reconstructing functions from noisy data is marginal, as shown in the next section.

d,n	$\|\psi_n^d\|_\infty$	$n\|\psi_n^d\|_\infty$	$\|(\psi_n^d)'\|_\infty$	$n^2\|(\psi_n^d)'\|_\infty$
1,1	1.0000	1.0000	1.9844	1.9844
1,3	0.1489	0.4468	0.0018	0.0163
1,5	0.0849	0.4249	0.0010	0.0256
1,7	0.0592	0.4144	0.0005	0.0245
3,2	1.0000	2.0000	1.0926	4.3704
3,3	0.4156	1.2469	0.0296	0.2661
3,5	0.2254	1.1273	0.0007	0.0195
3,7	0.1565	1.0957	0.0005	0.0280
5,3	1.0000	3.0000	0.9048	8.1432
5,5	0.3793	1.8968	0.0022	0.0561
5,7	0.2574	1.8020	0.0003	0.0148

Table 3: Maxima of ψ_n^d and its derivative for several values of the degree d and the support size n. Note that scaling these maxima with n and n^2, respectively, gives approximately constant values, matching the rates in Conjecture 7. More results are shown in Figure 8.

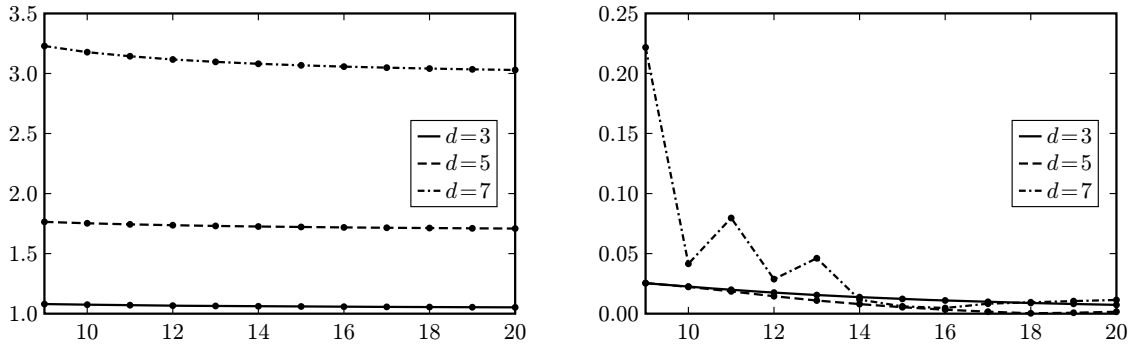

Figure 8: Behaviour of $n\|\psi_n^d\|_\infty$ (left) and $n^2\|(\psi_n^d)'\|_\infty$ (right) as a function of n for three values of the degree d. Both quantities become approximately constant as n increases, as predicted in Conjecture 7.

4.4 Numerical examples for the primal least squares schemes of higher degree

The statistical model presented in Section 3.1 is also valid for schemes based on higher degree least squares polynomials, due to the linearity of the schemes (see also Appendix A.2), but proving asymptotic bounds for ψ_n^d becomes difficult, because the mask of S_n^d is no longer positive and not given explicitly for $d > 1$. However, our numerical tests, which are summarized in Table 3 and Figure 8, indicate that the bounds in Theorem 6 for the special case $d = 1$ also hold for $d > 1$.

Conjecture 7. *Let ψ_n^d be defined as in* (17) *for schemes based on least squares polynomials of degree d. Then,*

$$\|\psi_n^d\|_\infty \sim \frac{1}{n}$$

and

$$\|(\psi_n^d)'\|_\infty \sim \frac{1}{n^2}.$$

The deterministic error in (16) is strongly related to d. This can be seen by the polynomial reproduction property of our schemes, that is, the reconstruction of any polynomial of degree d from its values at the integers by the limit of S_n^d. The latter property implies that the approximation order is at least h^{d+1}. Thus, for larger d the contribution of the deterministic error decreases, while we conjecture that the effect of the noise increases. This relates to the following predicted behaviour of ψ_n^d with respect to d and n and is supported by the results shown in Figure 9.

Conjecture 8. *For any fixed support size n and different degrees d_1 and d_2 with $d_1 > d_2$,*

$$\psi_n^{d_1}(x) \geq \psi_n^{d_2}(x), \qquad x \in [0,1].$$

For any fixed degree d and different support size n_1 and n_2 with $n_1 < n_2$,

$$\psi_{n_1}^{d}(x) \geq \psi_{n_2}^{d}(x), \qquad x \in [0,1].$$

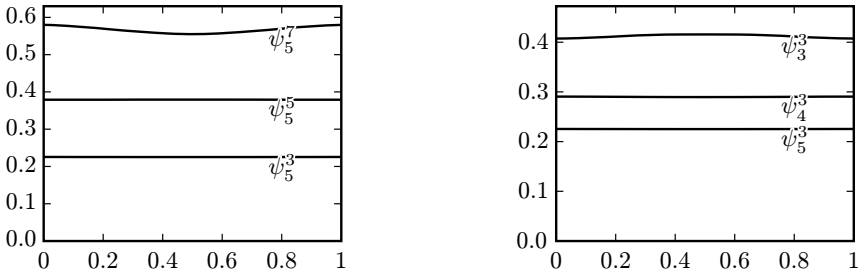

Figure 9: Comparison of ψ_n^d for several values of the degree d and the support size n. Note how ψ_n^d increases pointwise with d for fixed n (left) and decreases as n increases for fixed $d=3$ (right), as predicted by Conjecture 8.

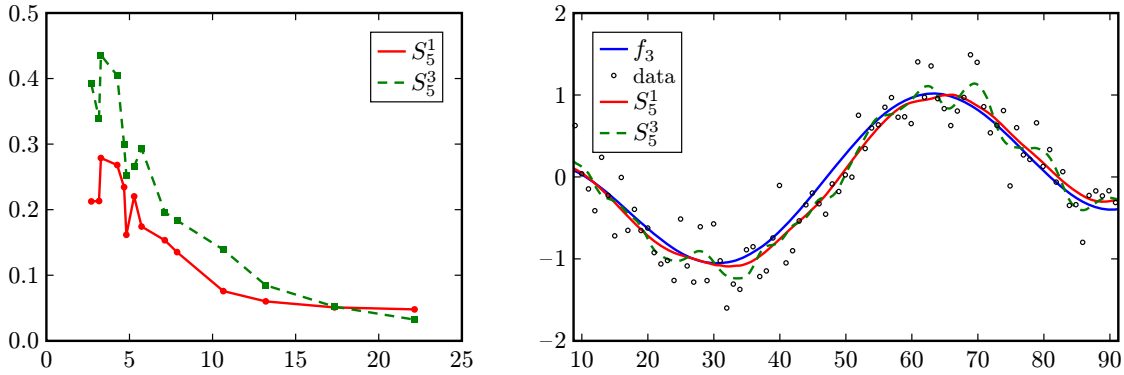

Figure 10: Comparison of S_5^1 and S_5^3 for f_3. Left: Relative approximation error as a function of the SNR. Right: Reconstruction of f_3 from data with 8.2 dB noise.

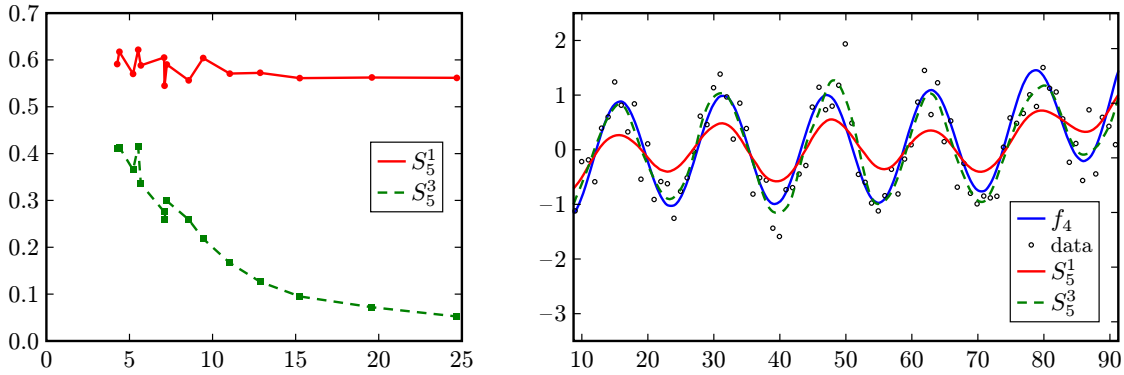

Figure 11: Comparison of S_5^1 and S_5^3 for f_4. Left: Relative approximation error as a function of the SNR. Right: Reconstruction of f_4 from data with 7.2 dB noise.

To further back up this conjecture, let us consider some numerical experiments, similar to those in Section 3.2. We first compare the schemes S_5^1 and S_5^3, applied to noisy data taken from the slowly varying function

$$f_3(x) = \cos\frac{x}{10} - \left(\frac{x}{50} - 1\right)^3, \tag{23}$$

for which the deterministic error is expected to be small. Figure 10 shows that S_5^1, which is based on locally fitting linear polynomials, gives better reconstructions, as long as the noise is significant. However, as the noise decays, the deterministic error becomes more relevant and the scheme S_5^3, which is based on locally fitting cubic polynomials and therefore has approximation order h^4, manages to estimate the function more accurately than S_5^1, whose approximation order is only h^2. This example emphasizes the trade-off between the deterministic approximation error and the effect of noise on the expected squared error, and this effect

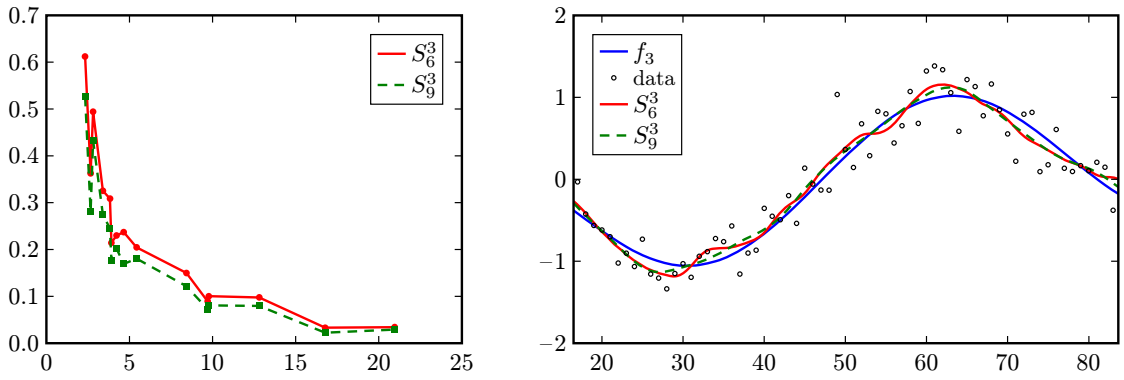

Figure 12: Comparison of S_6^3 and S_9^3 for f_3. Left: Relative approximation error as a function of the SNR. Right: Reconstruction of f_3 from data with 8.4 dB noise.

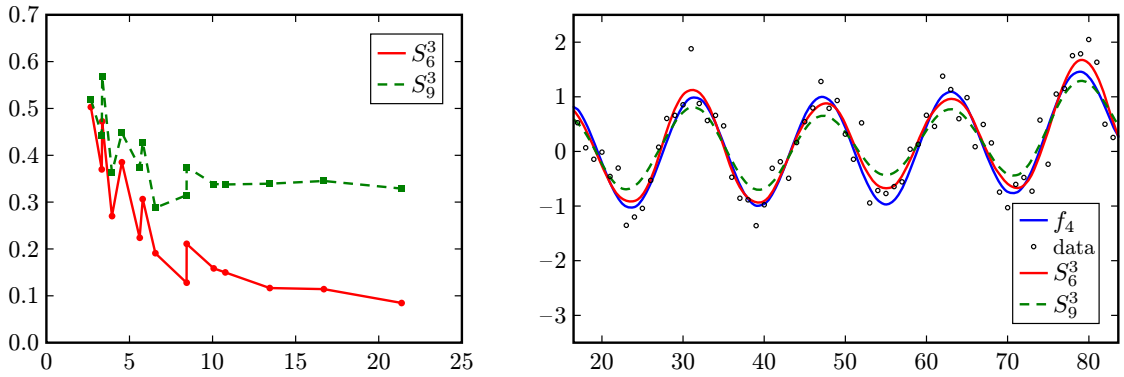

Figure 13: Comparison of S_6^3 and S_9^3 for f_4. Left: Relative approximation error as a function of the SNR. Right: Reconstruction of f_4 from data with 9.5 dB noise.

becomes even clearer if we consider the function

$$f_4(x) = \cos\frac{2x}{5} - \left(\frac{x}{50} - \frac{4}{5}\right)^3. \tag{24}$$

Due to the oscillations of this function, the deterministic error is dominant and the results in Figure 11 confirm that S_5^3 outperforms S_5^1 for all noise levels.

Finally, we repeat the experiments with the test functions f_3 and f_4 for the schemes S_6^3 and S_9^3, which are both based on locally fitting cubic polynomials but have different support sizes. Figure 12 shows that the larger support helps to smooth out noise if the deterministic error is small. But if the deterministic error is more relevant than the noise, than the smaller support leads to smaller reconstruction errors for all noise levels, as illustrated in Figure 13.

5 Application to noisy geometric data

We conclude the paper by presenting applications of our least squares subdivision schemes to noisy samples of curves and of surfaces. We measure the level of the noise by SNR, although this measure in the geometrical setting is less informative than in the functional setting, because the significance of the noise also depends highly on the geometry.

5.1 Examples of curves

The parametrization of a curve enables us to apply our univariate subdivision schemes to each of its components. By doing so, we can construct an approximation to the curve from its noisy samples. We introduce two such examples.

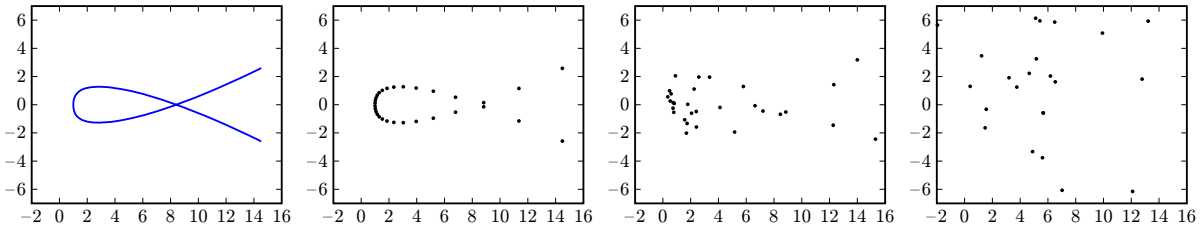

Figure 14: The alpha-like curve (a) from Equation (25) and its sample points (b), contaminated by low level noise of about 17 dB (c) and by high level noise of about 4 dB (d).

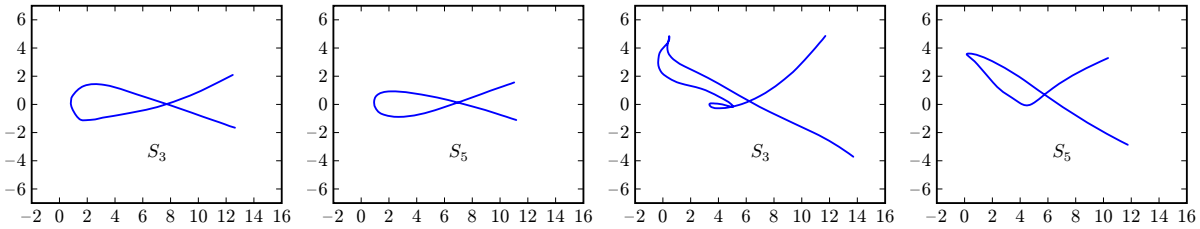

Figure 15: Limits of the two least squares subdivision schemes S_3 and S_5, applied to noisy samples from the alpha-like curve in Figure 14 (a). The two curves on the left (a,b) correspond to the samples with low level noise in Figure 14 (c) and the two curves on the right (c,d) correspond to the samples with high level noise in Figure 14 (d).

The first example consists of an alpha-like curve, given by

$$x(t) = 3t^4 + t^2 + 1, \qquad y(t) = t^5 - 2t, \tag{25}$$

sampled equidistantly over $[-1.4, 1.4]$, that is, with samples taken at $t_i = -7/5 + ih$, where $h = 14/145$ and $i = 0, \ldots, 29$. This curve and its sample points are shown in Figure 14 (a,b). The first set of noisy samples with a relatively low level of noise is shown in Figure 14 (c). We apply S_3 and S_5 (both based on linear fitting, see Section 2) to these samples, giving the limit curves in Figures 15 (a) and 15 (b), respectively. The limits of both schemes retain the general shape of the curve, but a minor artifact appears on the limit curve generated by S_3 since it closely fits the noisy samples. Perturbing the samples with high level noise, as seen in Figure 14 (d), reveals an overfitting by the limit curve of S_3 in Figure 15 (c), while the limit curve generated by S_5 in Figure 15 (d) preserves the topology of the original curve.

In the second example we apply four different least squares schemes to noisy samples of a star-shaped curve, given by

$$x(t) = 4\cos(t) + \cos(4t), \qquad y(t) = 4\sin(t) - \sin(4t), \tag{26}$$

We sample this curve at $t_i = i/(100\pi)$ for $i = 0, \ldots, 49$. This curve and its sample points are shown in Figure 16 (a,b). In this example we compare the performance of four schemes: two schemes based on linear fitting, S_3 and S_5, and two schemes based on cubic fitting, S_4^3 and S_6^3. As in the first example, we start by investigating the case of low level noise, with the samples shown in Figure 16 (c). The limits of all schemes are presented in the upper row of Figure 17. They all have the shape of a star, except for the limit of S_5, which is more similar to a pentagon than to a star. By zooming in, it can be seen that the limit curve generated by S_4^3 suffers from a minor artifact next to its lowest vertex, this being consequence of trying to fit the noisy data. For the set of samples with high level noise in Figure 16 (d), the results confirm our previous observation. Namely, both S_3 and S_6^3 generate reasonable results, while the limit curves generated by S_5 and S_4^3 suffer from geometrical artifacts caused by oversmoothing and overfitting, respectively. These limits are presented in the lower row of Figure 17.

5.2 Examples of surfaces

Equipped with univariate least squares subdivision schemes, we use tensor-product bivariate schemes based on them. These bivariate schemes are applied to noisy samples of surfaces, given at vertices of quadrilateral grids. Two examples are provided to illustrate the application of these bivariate schemes to noisy data.

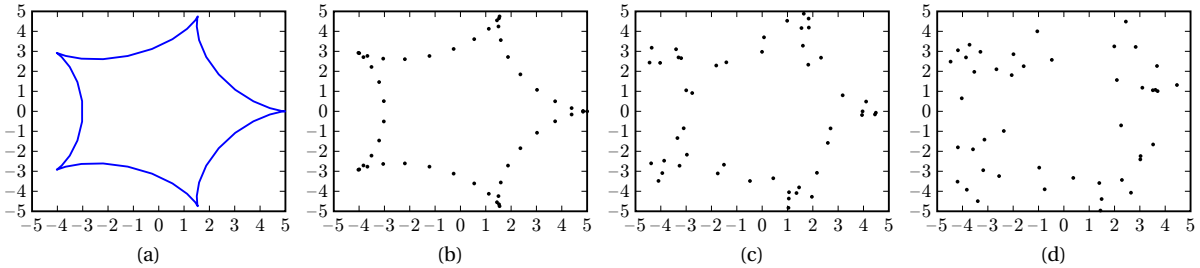

Figure 16: The star-shaped curve (a) from Equation (26) and its sample points (b), contaminated by low level noise of about 19 dB (c) and by high level noise of about 10 dB (d).

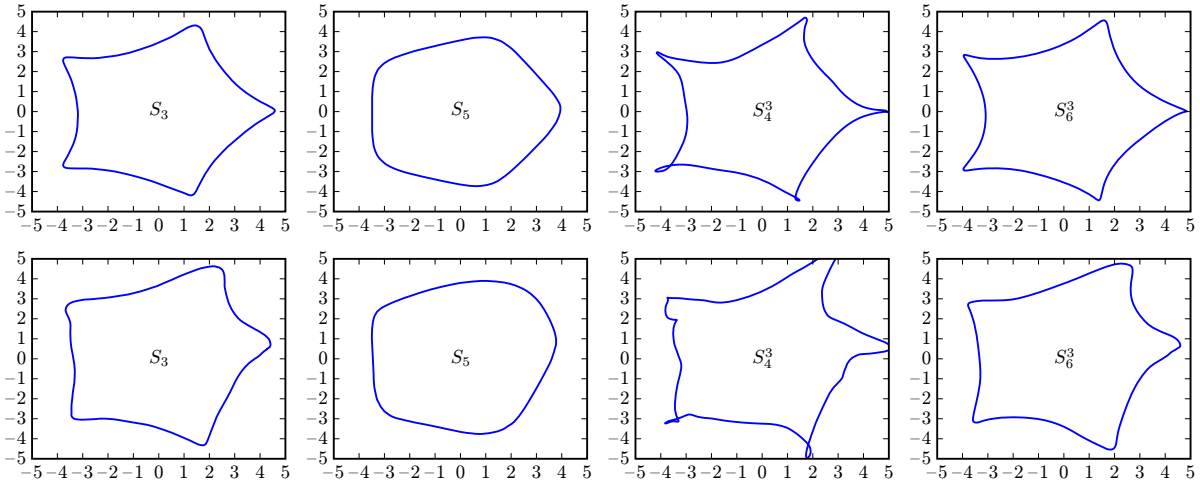

Figure 17: Limits of four least squares subdivision schemes, applied to noisy samples from the star-shaped curve in Figure 16 (a). The curves in the upper row correspond to the samples with low level noise in Figure 16 (c) and the curves in the lower row correspond to the samples with high level noise in Figure 16 (d).

The first surface we examine is a torus surface, given by

$$\begin{aligned} x(u,v) &= \cos(u)(10+5\cos(v)), \\ y(u,v) &= \sin(u)(10+5\cos(v)), \\ z(u,v) &= 5\sin(v), \end{aligned} \tag{27}$$

sampled every 15 degrees, that is, at $u_i = i\pi/12$ and $v_j = j\pi/12$ for $i, j = 0,\ldots,23$. This surface and its sample points are shown in Figure 18 (a,b). We investigate the limits of the bivariate tensor-product schemes $S_3 \otimes S_3$ and $S_5 \otimes S_5$. First, we study the application of these schemes to the samples with low level noise in Figure 18 (c). The limits of both schemes in Figures 19 (a) and 19 (b) are fairly good. For the samples with high level noise in Figure 18 (d), the limit of $S_3 \otimes S_3$ in Figure 19 (c) keeps the general shape but is a poor approximation to the torus, while the limit of $S_5 \otimes S_5$ in Figure 19 (d) provides a better approximation.

The surface of the second example is not a mathematical surface but a scan of a mechanical element, parameterized by a quadrilateral grid, and given in terms of $49 \times 81 = 3969$ vertices. Figures 20 (a) and 20 (b) show the surface and its sample points, respectively. Similarly to the second example in the curve case, we investigate the limit surfaces generated by the four tensor product schemes: $S_3 \otimes S_3$, $S_5 \otimes S_5$, $S_4^3 \otimes S_4^3$, and $S_6^3 \otimes S_6^3$. We compare their limits from a set of samples with a low level of noise and a set of samples with a high level of noise. These sets of samples are given in Figures 20 (c) and 20 (d), respectively. The limit surfaces for the samples with low level noise, shown in the upper row of Figure 21, indicate that $S_3 \otimes S_3$ and $S_6^3 \otimes S_6^3$ outperform the other two schemes. For the samples with high level noise, the performance of $S_5 \otimes S_5$ is superior to that of the other three, as can be seen in the lower row of Figure 21.

Acknowledgements

We wish to thank Prof. Felix Abramovich for his help in formulating the statistical model for the expected squared error.

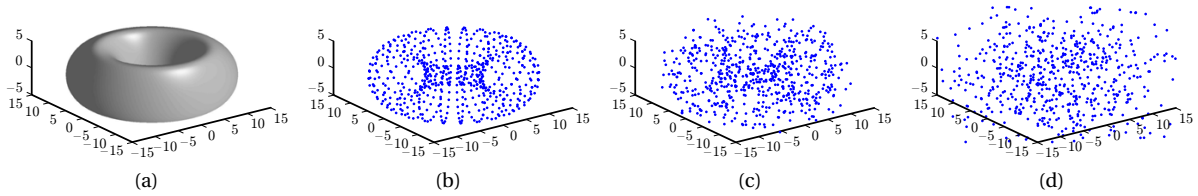

Figure 18: The torus surface (a) from Equation (27) and its sample points (b), contaminated by low level noise of about 15 dB (c) and by high level noise of about 6 dB (d).

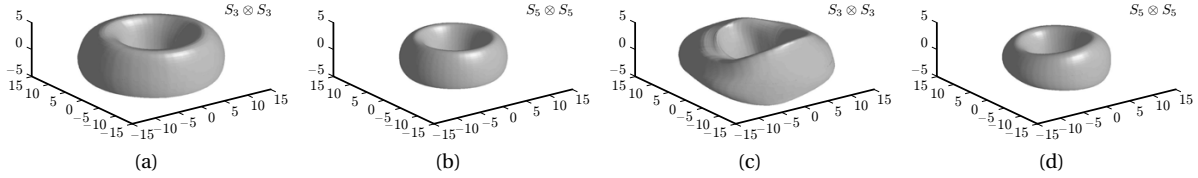

Figure 19: The limits of two bivariate tensor-product schemes, applied to noisy samples from the torus surface in Figure 18 (a). The two surfaces on the left (a,b) are obtained from the samples with low level noise in Figure 18 (c) by $S_3 \otimes S_3$ and $S_5 \otimes S_5$, respectively. The two surfaces on the right (c,d) are obtained from the samples with high level noise in Figure 18 (d) by $S_3 \otimes S_3$ and $S_5 \otimes S_5$, respectively.

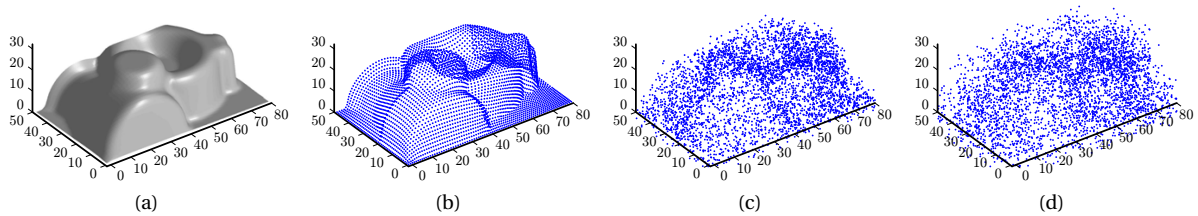

Figure 20: The surface of a mechanical element (a) and its sample points (b), contaminated by low level noise of about 24 dB (c) and by high level noise of about 18 dB (d).

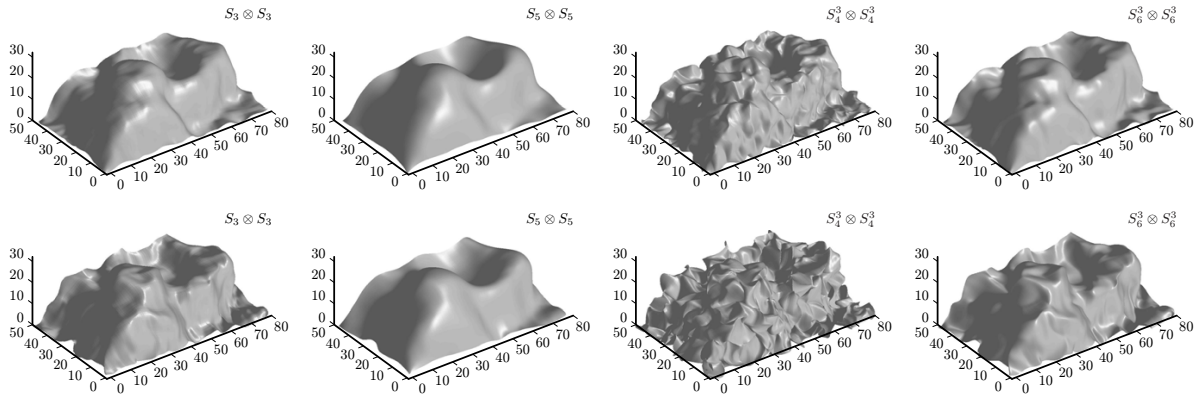

Figure 21: Limits of four bivariate tensor-product schemes, applied to noisy samples from the surface of the mechanical element in Figure 20 (a). The surfaces in the upper row correspond to the samples with low level noise in Figure 20 (c) and the surfaces in the lower row correspond to the samples with high level noise in Figure 20 (d).

References

[1] L.-E. Andersson and N. F. Stewart. *Introduction to the Mathematics of Subdivision Surfaces.* SIAM, Philadephia, PA, 2010.

[2] A. S. Cavaretta, W. Dahmen, and C. A. Micchelli. Stationary subdivision. *Memoirs of the American Mathematical Society*, 93(453):186 pages, Sept. 1991.

[3] G. M. Chaikin. An algorithm for high speed curve generation. *Computer Graphics and Image Processing*, 3(4):346–349, Dec. 1974.

[4] C. Conti and N. Dyn. Analysis of subdivision schemes for nets of functions by proximity and controllability. *Journal of Computational and Applied Mathematics*, 236(4):461–475, Sept. 2011.

[5] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constructive Approximation*, 5(1):49–68, Dec. 1989.

[6] D. L. Donoho and T. P. Y. Yu. Nonlinear pyramid transforms based on median-interpolation. *SIAM Journal on Mathematical Analysis*, 31(5):1030–1061, 2000.

[7] S. Dubuc. Interpolation through an iterative scheme. *Journal of Mathematical Analysis and Applications*, 114(1):185–204, Feb. 1986.

[8] N. Dyn. Subdivision schemes in computer-aided geometric design. In W. Light, editor, *Advances in Numerical Analysis*, volume II, pages 36–104. Oxford University Press, New York, 1992.

[9] N. Dyn. Analysis of convergence and smoothness by the formalism of Laurent polynomials. In A. Iske, E. Quak, and M. S. Floater, editors, *Tutorials on Multiresolution in Geometric Modelling*, Mathematics and Visualization, pages 51–68. Springer, Berlin, Heidelberg, 2002.

[10] N. Dyn and E. Farkhi. Spline subdivision schemes for compact sets. A survey. *Serdica Mathmatical Journal*, 28(4):349–360, 2002.

[11] N. Dyn, M. S. Floater, and K. Hormann. A C^2 four-point subdivision scheme with fourth order accuracy and its extensions. In M. Dæhlen, K. Mørken, and L. L. Schumaker, editors, *Mathematical Methods for Curves and Surfaces: Tromsø 2004*, Modern Methods in Mathematics, pages 145–156. Nashboro Press, Brentwood, 2005.

[12] N. Dyn, K. Hormann, M. A. Sabin, and Z. Shen. Polynomial reproduction by symmetric subdivision schemes. *Journal of Approximation Theory*, 155(1):28–42, Nov. 2008.

[13] N. Dyn and D. Levin. Subdivision schemes in geometric modelling. *Acta Numerica*, 11:73–144, Jan. 2002.

[14] A. Eisinberg and G. Fedele. Discrete orthogonal polynomials on equidistant nodes. *International Mathematical Forum*, 2(21):1007–1020, 2007.

[15] W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and Semiparametric Models*. Springer Series in Statistics. Springer-Verlag, Berlin, Heidelberg, 2004.

[16] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer-Verlag, Berlin, Heidelberg, second edition, 2009.

[17] U. Itai and N. Dyn. Generating surfaces by refinement of curves. *Journal of Mathematical Analysis and Applications*, 388(2):913–928, Apr. 2012.

[18] G. Mustafa, H. Li, J. Zhang, and J. Deng. ℓ_1-Regression based subdivision schemes for noisy data. *Computer-Aided Design*, 58:189–199, Jan. 2015.

[19] R. Penrose. A generalized inverse for matrices. *Proceedings of the American Mathematical Society*, 51(3):406–413, July 1955.

[20] J. Peters and U. Reif. *Subdivision Surfaces*, volume 3 of *Geometry and Computing*. Springer-Verlag, Berlin, Heidelberg, 2008.

[21] N. Sharon and U. Itai. Approximation schemes for functions of positive-definite matrix values. *IMA Journal of Numerical Analysis*, 33(4):1436–1468, 2013.

[22] J. Wallner and N. Dyn. Convergence and C^1 analysis of subdivision schemes on manifolds by proximity. *Computer Aided Geometric Design*, 22(7):593–622, Oct. 2005.

[23] J. Wallner, E. Nava Yazdani, and P. Grohs. Smoothness properties of Lie group subdivision schemes. *Multiscale Modeling & Simulation*, 6(2):493–505, 2007.

A Least squares schemes and orthonormal polynomials

In this appendix we derive several properties of least squares polynomials used throughout this paper. Some of the properties can be considered common knowledge, but we present them here in order to keep the paper as self-contained as possible.

A.1 Least squares polynomials in terms of orthonormal polynomials

Our subdivision schemes are based on least squares polynomial fitting. We denote by Π_d the space of polynomials of degree at most d. Fitting data $y_1,\dots,y_m$ given at the nodes $x_1,\dots,x_m$ by a polynomial $p\in\Pi_d$ with $d<m$ requires finding the polynomial p^* which minimizes the sum of squared errors,

$$\sum_{i=1}^{m}\big(p(x_i)-y_i\big)^2, \tag{A.1}$$

among all $p \in \Pi_d$. The coefficients $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_d)$ of $p^*(x) = \sum_{j=0}^{d} \beta_j x^j$ are typically determined by setting the gradient of the functional in (A.1) to zero, resulting in the normal equations

$$A^T A \boldsymbol{\beta} = A^T \boldsymbol{y},$$

where A is the $m \times (d+1)$ *Vandermonde matrix* with entries $A_{i,j} = (x_i)^j$ and $\boldsymbol{y} = (y_1, \ldots, y_m)$ is the data vector. The matrix $A^T A$ is invertible for any set of distinct nodes $x_1, \ldots, x_m$ and the solution of the normal equations is given by

$$\boldsymbol{\beta} = A^\dagger \boldsymbol{y}, \tag{A.2}$$

where $A^\dagger = (A^T A)^{-1} A^T$ is the *Moore–Penrose pseudoinverse* [19] of A.

Remark 9. If $d = m-1$, then p^* is the unique *interpolating* polynomial to the data. Furthermore, this ansatz can also be used in the case $d \geq m$ to pick among all interpolating polynomials the one with the smallest ℓ_2-norm of the coefficients $\boldsymbol{\beta}$. This can be achieved by using[1] $A^\dagger = A^T (A A^T)^{-1}$ in (A.2), but then the solution depends on the particular basis of Π_d chosen to represent p^*. However, $p^*(x_i) = y_i$, $i = 1, \ldots, m$, independently of that choice.

Let us now express the solution p^* which minimizes (A.1) for $d < m$ in terms of orthonormal polynomials. Recall the notion of orthonormal polynomials with respect to a discrete inner product. Let $\boldsymbol{X} = \{x_1, \ldots, x_m\}$ be a set of distinct nodes and define for any two functions $f, g : \mathbb{R} \to \mathbb{R}$ the discrete inner product

$$\langle f, g \rangle_{\boldsymbol{X}} = \sum_{i=1}^{m} f(x_i) g(x_i). \tag{A.3}$$

A family $\mathcal{L} = \{L^0, \ldots, L^k\}$ of $k+1$ polynomials in Π_d with $k \leq d$, is orthonormal over $\boldsymbol{X}$ if

$$\langle L^i, L^j \rangle_{\boldsymbol{X}} = \delta_{i,j}, \qquad i, j = 0, \ldots, k, \tag{A.4}$$

where $\delta_{i,j}$ is the standard Kronecker delta, that is, $\delta_{i,j} = 1$ for $i = j$ and $\delta_{i,j} = 0$, otherwise. Under the assumption that $L^j \in \Pi_j$, $j = 0, \ldots, k$, there exists a unique family $\mathcal{L}$ satisfying (A.4). The coefficients of the least squares solution p^* with respect to this unique family for $k = d$ are simply

$$\gamma_j = \langle L^j(\boldsymbol{X}), \boldsymbol{y} \rangle = \sum_{i=1}^{m} L^j(x_i) y_i, \qquad j = 0, \ldots, d,$$

that is,

$$p^*(x) = \sum_{j=0}^{d} \gamma_j L^j(x). \tag{A.5}$$

For more details, see [14] and the references therein.

A.2 The masks in terms of orthonormal polynomials

A naive implementation of the refinement rules in (21) and (22) for the least squares schemes of higher degree is computationally expensive, because the solution of each least squares problem is equivalent to the solution of a linear system, and it needs to be solved for every new data value f_i^{k+1}. However, it turns out that the subdivision schemes S_n^d and $\bar{S}_n^d$ for $d > 1$ are stationary, just like the schemes of degree $d = 1$, so that

$$f_i^{k+1} = \sum_j \alpha_{i-2j} f_j^k, \tag{A.6}$$

where the coefficients $\{\alpha_\ell\}_{\ell \in \mathbb{Z}}$ are independent of i and k and only a finite number of these coefficients are non-zero. To see this, we first prove that least squares polynomials are invariant under affine transformations.

Proposition 10. *Let p^* be the least squares polynomial of degree d for the data $\boldsymbol{y} = (y_1, \ldots, y_m)$ given at the nodes $\boldsymbol{X} = \{x_1, \ldots, x_m\}$ and let $\varphi(x) = a\,x + b$ with $a \neq 0$ be an affine transformation. Then $\bar{p}^* = p^* \circ \varphi^{-1}$ is the least squares polynomial of degree d for the same data $\boldsymbol{y}$ given at the transformed nodes $\bar{\boldsymbol{X}} = \varphi(\boldsymbol{X}) = \{\bar{x}_1, \ldots, \bar{x}_m\}$ with $\bar{x}_i = \varphi(x_i)$, $i = 1, \ldots, m$.*

Proof. Let $\mathcal{L} = \{L^0, \ldots, L^d\}$ be the unique family of orthonormal polynomials over $\boldsymbol{X}$. Then the family of polynomials $\bar{\mathcal{L}} = \{\bar{L}^0, \ldots, \bar{L}^d\}$ with $\bar{L}^j = L^j \circ \varphi^{-1}$, $j = 0, \ldots, d$ is orthonormal over $\bar{\boldsymbol{X}}$, because

$$\langle \bar{L}^i, \bar{L}^j \rangle_{\bar{\boldsymbol{X}}} = \sum_{k=1}^{m} \bar{L}^i(\bar{x}_k) \bar{L}^j(\bar{x}_k) = \sum_{k=1}^{m} L^i(x_k) L^j(x_k) = \langle L^i, L^j \rangle_{\boldsymbol{X}} = \delta_{i,j},$$

according to (A.3) and (A.4). The statement then follows using (A.5), since

$$\bar{p}^*(x) = \sum_{j=0}^{d} \sum_{i=1}^{m} \bar{L}^j(\bar{x}_i) y_i \bar{L}^j(x) = \sum_{j=0}^{d} \sum_{i=1}^{m} L^j(x_i) y_i L^j\big(\varphi^{-1}(x)\big) = p^*\big(\varphi^{-1}(x)\big).$$

□

[1] Note that the published version of this article contains a typo, as it incorrectly states to use $A^\dagger = (A A^T)^{-1} A^T$ in this case.

For the derivation of the masks of S_n^d we introduce for $n \geq d \geq 1$ two sets of points

$$\boldsymbol{X}_n = \{-2n+2, -2n+4, \ldots, 2n-2\}, \qquad \hat{\boldsymbol{X}}_n = \{-2n+1, -2n+3, \ldots, 2n-1\},$$

and denote the corresponding families of $d+1$ orthonormal polynomials by

$$\mathcal{L}_n^d = \{L_n^0, \ldots, L_n^d\}, \qquad \hat{\mathcal{L}}_n^d = \{\hat{L}_n^0, \ldots, \hat{L}_n^d\}.$$

Corollary 11. *For any $n \geq 1$ and $d \geq 1$ with $d < 2n$, the subdivision scheme S_n^d is stationary and the coefficients of its mask $[\alpha_{-2n+1}, \ldots, \alpha_{2n-1}]$ are*

$$\alpha_{2i} = \sum_{j=0}^{d} L_n^j(-2i) L_n^j(0), \qquad i = -n+1, \ldots, n-1 \tag{A.7}$$

and

$$\alpha_{2i+1} = \sum_{j=0}^{d} \hat{L}_n^j(-2i-1) \hat{L}_n^j(0), \qquad i = -n, \ldots, n-1.$$

Proof. Let $L^0, \ldots, L^d$ be the orthonormal polynomials over $\{t_{i-n+1}^k, \ldots, t_{i+n-1}^k\}$. Then, by (21) and (A.5),

$$f_{2i}^{k+1} = \sum_{j=0}^{d} \left(\sum_{l=-n+1}^{n-1} L^j(t_{i+l}^k) f_{i+l}^k \right) L^j(t_i^k),$$

Since these nodes relate to the nodes $\boldsymbol{X}_n$ by the affine transformation $\varphi(x) = 2^{-k-1}(x+2i)$, that is, $t_{i+j}^k = \varphi(j)$, $j = -2n+2, -2n+4, \ldots, 2n-2$, we can apply Proposition 10 to get

$$f_{2i}^{k+1} = \sum_{j=0}^{d} \left(\sum_{l=-n+1}^{n-1} L_n^j(2l) f_{i+l}^k \right) L_n^j(0) = \sum_{l=-n+1}^{n-1} \left(\sum_{j=0}^{d} L_n^j(2l) L_n^j(0) \right) f_{i+l}^k.$$

Substituting l by $l-i$ and comparing terms with f_{2i}^{k+1} as given in (A.6) we get the coefficients in (A.7). The coefficients with odd indices can be found similarly, replacing $\boldsymbol{X}_n$ by $\hat{\boldsymbol{X}}_n$. □

Explicit formulas for the mask coefficients of the dual schemes $\bar{S}_n^d$ and the other variants mentioned in Section 4.3 can be derived analogously.

A.3 Computation of the masks

Corollary 11 suggests computing the mask coefficients of S_n^d by evaluating the orthonormal polynomials L_n^i and $\hat{L}_n^i$, which can be derived from the explicit formulae for orthonormal polynomials over equidistant nodes in [14, Proposition 2], using Proposition 10 and suitable affine transformations. For example,

$$L_n^0(x) = \frac{1}{\sqrt{2n-1}}, \qquad L_n^1(x) = \frac{x}{\sqrt{(2n-2)(2n-1)2n/3}}$$

and

$$\hat{L}_n^0(x) = \frac{1}{\sqrt{2n}}, \qquad \hat{L}_n^1(x) = \frac{x}{\sqrt{(2n-1)2n(2n+1)/3}}.$$

Note that L_n^1 and $\hat{L}_n^1$ are odd polynomials. Therefore, $L_n^1(0) = \hat{L}_n^1(0) = 0$, and Corollary 11 confirms that the coefficients of the least squares schemes of degree $d = 1$ are $\alpha_{2i} = 1/(2n-1)$ and $\alpha_{2i-1} = 1/(2n)$, as stated in (2).

Remark 12. More generally, it follows from the formula in [14] that L_n^i and $\hat{L}_n^i$ are odd polynomials for odd i and even polynomials otherwise. Thus, $L_n^{2i+1}(0) = \hat{L}_n^{2i+1}(0) = 0$, and so by Corollary 11 the coefficients of the schemes S_n^{2d} and S_n^{2d+1} are identical .

However, a direct algorithm for computing the mask coefficients of S_n^d, independent of the orthonormal polynomials, is given by the following observation.

Proposition 13. *For any $n \geq 1$ and $d \geq 1$ with $d < 2n$, let A and $\tilde{A}$ be the Vandermonde matrices with $d+1$ columns for the nodes $\boldsymbol{X}_n$ and $\hat{\boldsymbol{X}}_n$, respectively. Further let $A_{1,\bullet}^\dagger$ and $\hat{A}_{1,\bullet}^\dagger$ be the first rows of the pseudo-inverses $A^\dagger$ and $\hat{A}^\dagger$. The mask coefficients of the subdivision scheme S_n^d are then given by*

$$(\alpha_{2n-2}, \alpha_{2n-4}, \ldots, \alpha_{-2n+2}) = A_{1,\bullet}^\dagger \qquad \textit{and} \qquad (\alpha_{2n-1}, \alpha_{2n-3}, \ldots, \alpha_{-2n+1}) = \hat{A}_{1,\bullet}^\dagger.$$

Proof. For any $i \in \{-n+1,\dots,n-1\}$, observe that by (A.5) the least squares polynomial over $\boldsymbol{X}_n$ for the data $\boldsymbol{y}^i = (y^i_{-n+1},\dots,y^i_{n-1})$ with $y^i_j = \delta_{i,j}$ is of the form

$$\ell^i_n(x) = \sum_{j=0}^{d} L^j_n(2i) L^j_n(x) = \sum_{j=0}^{d} \beta^i_j x^j,$$

where according to (A.2) $\boldsymbol{\beta}^i = (\beta^i_0,\dots,\beta^i_d) = A^\dagger \boldsymbol{y}^i$. The statement on the coefficients with even indices then follows from (A.7), because

$$\alpha_{2i} = \ell^{-i}_n(0) = \beta^{-i}_0 = A^\dagger_{1,n-i}.$$

The statement regarding the coefficients with odd indices can be derived analogously. □

Overall, this means that the main cost for computing the mask of S^d_n is the inversion of the two $(d+1)\times(d+1)$ matrices $A^T A$ and $\hat{A}^T \hat{A}$. The masks of the dual schemes $\bar{S}^d_n$ and the other variants mentioned in Section 4.3 can be computed similarly.