CONSTRUCTIVE THEORY OF FUNCTIONS, Sozopol 2016
(K. Ivanov, G. Nikolov and R. Uluchev, Eds.), pp. 29-50
Prof. Marin Drinov Academic Publishing House, Sofia, 2018

Local Splines on Non-uniform Grids Generating Real-time Wavelet Transforms

Amir Averbuch, Pekka Neittaanmäki and
Valery Zheludev

In this paper, local cubic and quadratic quasi-interpolating splines on non-uniform grids are described. The splines are designed by fast computational algorithms that utilize the relation between splines and cubic interpolation polynomials. These splines provide an efficient tool for real-time signal processing. As an input, the splines use either clean or noised arbitrarily-spaced samples. Formulas for the spline's extrapolation beyond the sampling interval are established. Sharp estimations of the approximation errors for cubic splines are presented. The designed splines serve as a source for generating real-time wavelet transforms to apply to signals in scenarios where the signal's samples subsequently arrive one after the other at random times. On arrival of new samples, only a couple of adjacent transform coefficients are updated in a way that no boundary effects arise.

Keywords and Phrases: Local splines, quasi-interpolating splines, real-time wavelet transform, local discrete vanishing moments.

Mathematics Subject Classification 2010: 41A15, 42C40.

1. Introduction

Since their introduction in [14], splines have become one of the most powerful tools in mathematics and computer aided geometric design. In recent decades, splines have served as a source for constructions of wavelets, wavelet packets and wavelet frames. Splines and spline- based wavelets, wavelet packets and frames are extensively used in signal and image processing applications (see, for example, [3, 4]).

Interpolating splines possess exclusive approximation properties. Due to these properties, the splines of order p generated biorthogonal wavelets with

p and, in some cases, $p+1$ vanishing moments ([3, 4]). However, interpolating splines can not be used in real-time processing* due to the fact that for the computation of a spline, a system of equations, involving all the available samples and some boundary conditions, should be solved.

Therefore, the idea to have splines that can be designed and manipulated directly without resorting to systems of equations, while their approximation accuracy is close to what the interpolating splines achieve, is attractive. A method for the design of such splines on the uniform grid was presented in the pioneering spline paper [14]. When the grid is non-uniform, the design and estimation of the approximation properties of such splines is more complicated, especially for higher-order splines (see, for example, [13, 21]). These splines are called local because the computation of a spline's value at a fixed point requires to use only a few adjacent grid samples. Nevertheless, they provide the same approximation order as what interpolating splines provide. That is, there exist local splines of any order p which restore polynomials of degree $p-1$. Such local splines are referred to as quasi-interpolating splines.

In this paper, we describe a procedure to design and analyze local cubic and quadratic quasi-interpolating splines on arbitrary grids. Typically, local splines are designed via their B-spline representation. An alternative approach presented in this paper is based on the relation between quasi-interpolating splines and interpolating polynomials. The advantages of such approach are that, besides of a simple computational algorithm, it provides a smooth extension of a spline to the boundaries of sampling interval without loss of approximation accuracy. A method for accurate extrapolation of the spline beyond the sampling interval is presented. In addition, this approach paved a way to establishing sharp estimations of the approximation errors for cubic splines.

Only six (five) adjacent samples are needed to compute the cubic (quadratic) spline's value at a certain point. Therefore, the design of the spline can be implemented in a real-time mode when samples of a signal arrive dynamically and sequentially at random times. Due to the extension formulas, the design can be carried out without delay up to the latest sample arrival. In addition, the extrapolation algorithm can be applied to a prediction-correction processing of signals evolving in time.

The described splines are used for the construction of wavelet transforms of arbitrarily sampled signals. Most of the existing wavelet transforms operate on uniformly sampled signals. A few works that describe wavelets on non-uniform grids appeared in recent years. Equidistant wavelets, which are utilized for denoising of non-uniformly sampled signals, appear in [6]. Scaling functions on different levels of enclosed irregular grids, which appear in [8], are designed as limit functions of subdivision and wavelets are designed as their linear combinations. The paper [7] describes local decomposition of the spline space constructed on a fine irregular grid onto a coarse-grid spline space and

*Under the real-time processing we mean a situation when signal's samples arrive dynamically and sequentially and the computations are executed with a minimal, if any, delay with respect to the samples arrival.

its orthogonal complement. The bases of these subspaces are formed by quasi-interpolating splines and compactly supported pre-wavelets. A general method for constructing wavelets on irregular lattices in $\mathbb{R}^d$ and a general atomic frame decomposition of the space $L^2(\mathbb{R}^d)$ are described in [2].

In the current paper, the quasi-interpolating splines are utilized as a source of finite impulse response "filters" to generate wavelet transforms of discrete-time arbitrarily sampled signals. When sampling is uniform, these are filters in a proper sense of the word. Neither scaling functions nor continuous wavelets are involved.

A natural way to design and implement wavelet transforms of signals is the Lifting Scheme introduced in [18], which consists of subsequent applications of *prediction* and *updating* operators to signal's samples. The scheme was extended in [19] to non-uniformly sampled signals. The idea which, in the case of uniform sampling, is explored in [4], is to predict odd samples of the signal to be transformed by values of a spline constructed on the even samples of the signal. Then, the detail coefficients are derived by subtraction of the predicted odd samples from the original ones. The next step consists of updating the even samples by values of the spline designed on the detail coefficients. Since quasi-interpolating splines restore cubic polynomials, the detail wavelet transform coefficients are zero if the signal is a sampled cubic polynomial (at least locally). In that sense, we claim that the arising discrete-time wavelet transforms have four discrete vanishing moments. Typically, wavelet transforms of signals (images) defined on limited intervals (areas) require an extension of the object beyond its boundaries ([5]), in order to reduce boundary effects. However, by using the definition of the splines near the boundaries of the sampling interval, the wavelet transforms are implemented without this type of extension. The design of splines make them useful for real-time (with no delay) execution of wavelet transforms in the situation when samples of a signal subsequently arrive at random time moments.

A real-time denoising method, which is based on the lifting wavelet transform of uniformly sampled signals, is presented in [12]. The online wavelet transform is implemented using a moving window (see also [20]), which produces some delay with respect to the sample acquisition and requires an artificial extension of the data beyond the moving window. Our algorithm produces wavelet coefficients with no delay. No extension of the data is needed. Arrival of a new sample leads to the production of a new coefficient and to updating of a couple of already produced coefficients.

The paper is organized as follows. Section 2 introduces the necessary notation and recalls definitions of the divided differences and interpolating polynomials. Section 3 introduces local cubic quasi-interpolating splines, describes algorithms for spline computation on unlimited and finite intervals and an extrapolation method. Approximation properties of the splines are investigated. Section 4 outlines quadratic quasi-interpolating splines, mainly on uniform grids. A super-convergence property of such splines is established, which is valuable for the wavelet transforms. The Lifting Scheme of the wavelet

transforms is presented in Section 5.1. In Sections 5.3 and 5.4, spline-based prediction and updating operators are explicitly described. A scheme for the real-time wavelet transforms computation is outlined in Section 5.5.

In order for the paper to be self-contained, some results about local quasi-interpolating splines, which appeared in [17, 22], are included.

2. Preliminaries

The following notations are used throughout the paper:

The grid on the real axis is denoted by $\mathbf{t} := \{t[k]\}$. The steps of the grid are denoted by $h[k] := t[k+1] - t[k]$.

If $t \in [t[k], t[k+1]]$ then a *local variable* is $\tau := (t - t[k])/h[k] \in [0, 1]$.

The *n-order divided difference* (DD) ([1, 10, 16]) of a sequence $\mathbf{f} = \{f[k]\}$ with respect to the grid $\mathbf{t}$ is

$$f[k, k+1, \ldots, k+n] := \sum_{l=0}^{n} \frac{f[k+l]}{\omega_n'[k]\big(t[k+l]\big)},$$

where

$$\omega_m[k](t) := (t - t[k])(t - t[k+1]) \ldots (t - t[k+m]).$$

If a function $f(t)$ belongs to $C^n[t[k], t[k+n]]$, then for the DD of the sequence $\mathbf{f} := \big\{f[k+\nu] = f(t[k+\nu])\big\}_{\nu=0}^{n}$, there holds

$$f[k, k+1, \ldots, k+n] = \frac{f^{(n)}(\theta)}{n!}, \qquad \theta \in (t[k], t[k+n]). \tag{1}$$

The notation $P(t)[k]$ is used for a cubic polynomial that interpolates a function $f(t)$ at the grid points $\{t[k-1], t[k], t[k+1], t[k+2]\}$. The remainder term of the interpolation is explicitly expressed by

$$R(t) := f(t) - P(t)[k] = f[t, k, k+1, k+2, k+3]\, \omega_3[k](t). \tag{2}$$

3. Local Cubic Splines

The cubic B-spline is

$$b(t)[k] = (t[k+4] - t[k]) \sum_{\nu=0}^{4} \frac{(t - t[k+\nu])_+^3}{w_4'[k](t[k+\nu])}.$$

It is supported on the interval $(t[k], t[k+4])$. Its nodes are located at the grid points $t[k], \ldots, t[k+4]$.

Cubic splines on the grid $\mathbf{t} = \{t[k]\}$ are represented via the B-splines ([9, 15]). For $t \in [t[k], t[k+1]]$, a cubic spline $\mathbf{s}$ is given by

$$s(t) = \sum_{\nu \in \mathbb{Z}} q[\nu+2]\, b(t)[\nu] = \sum_{\nu=k-3}^{k} q[\nu+2]\, b(t)[\nu], \tag{3}$$

where $\mathbf{q} = \{q[\nu]\}$ is a sequence of real numbers, which determines the spline's properties.

For a spline that approximates a function $f(t)$, the coefficients $q[\nu]$ are derived from the samples $\{f[k] := f(t[k])\}$ of $f(t)$ on the grid $\mathbf{t}$. In that case, the spline is denoted by $\mathbf{s}[f]$. For interpolating splines, the coefficients $q[\nu]$ are obtained by solving a tridiagonal system of equations, which involves all the available samples and some boundary conditions. An alternative is provided by the so-called local splines.

Definition 1. If the coefficients $q[\nu]$ in Eq. (3) are finite linear combinations of grid samples $\{f[k]\}$, then the spline $\mathbf{s}[f]$ is referred to as local spline.

Definition 2. If for any polynomial $f(t) = P^n(t)$ of degree n, a spline satisfies $s[f](t) \equiv f(t)$, then it is said that the spline $\mathbf{s}[f]$ restores polynomials of degree n and its approximation order is $n+1$.

It is well-known that cubic interpolating splines restore cubic polynomials, thus having the approximation order 4.

Definition 3. A local cubic spline, whose approximation order is 4, is referred to as the quasi-interpolating local spline.

The quasi-interpolation can be achieved by using six grid samples for computation of the spline's value at a given point.

Set

$$\begin{aligned}
\beta_{-1}[k] &:= \frac{-(h[k])^2}{3\,h[k-1]\,(h[k-1]+h[k])}, \\
\beta_{1}[k] &:= \frac{-(h[k-1])^2}{3\,h[k]\,(h[k-1]+h[k])}, \\
\beta_{0}[k] &:= 1 - \beta_{-1}[k] - \beta_{1}[k],
\end{aligned}$$

where $h[k]$ is the grid step.

Proposition 1 ([21]). *If the coefficients in Eq.* (3) *are chosen as*

$$q_1[\nu] := \beta_{-1}[\nu]\, f[\nu-1] + \beta_0[\nu]\, f[\nu] + \beta_1[\nu]\, f[\nu+1],$$

then the spline

$$s[f](t) = \sum_{\nu\in\mathbb{Z}} q_1[\nu+2]\, b(t)[\nu] = \sum_{\nu=k-4}^{k+1} f[\nu+2]\, L^4(t)[\nu], \tag{4}$$

$$L^4(t)[k] := \sum_{m=-1}^{1} \beta_m[k+2-m]\, b(t)[k-m], \tag{5}$$

restores cubic polynomials. For the computation of the spline's values $s[f](t)$, $t \in [t[k], t[k+1]]$, *six grid samples* $f[k-2], \ldots, f[k+3]$ *are needed.*

3.1. Quasi-interpolating Cubic Splines

The quasi-interpolating cubic splines $\mathbf{s}[f]$ can be represented in a form alternative to Eq. (4), using a relation between the splines $\mathbf{s}[f]$ and cubic interpolation polynomials. Such form facilitates extension of the spline to the boundaries of the intervals and provides sharp estimates of the approximation errors for the splines $\mathbf{s}[f]$. Originally, this representation was introduced in [22].

Recall that $P(t)[k]$ denotes the cubic polynomial which interpolates the function $f(t)$ at the points $\{t[k-1], t[k], t[k+1], t[k+2]\}$, and $f[k] := f(t[k])$.

Theorem 1 ([22]). *For* $t = t[k] + h[k]\,\tau$, $\tau \in [0,1]$, *the cubic spline* $s[f](t)$, *which restores cubic polynomials, is expressed by*

$$s[f](t) = P(t)[k] + F[k-1]\,(1-\tau)^3 + F[k]\,\tau^3, \tag{6}$$

where the coefficients $F[k]$ *are*

$$F[k] := -\,f[k-1, k, k+1, k+2, k+3] \times \frac{(h[k])^2\,(h[k+1])^2\,(t[k+3]-t[k-1])}{3\,(t[k+2]-t[k])}.$$

3.2. Approximation Properties of Quasi-interpolating Cubic Splines

Approximation properties of the spline $\mathbf{s}[f]$ are close to the properties of the cubic spline $\mathbf{s}_i[f](t)$ which interpolates the function $f(t)$ on the grid $\mathbf{t}$ such that $s_i[f](t[k]) = f(t[k]) = f[k]$. Like the interpolating splines, the splines $\mathbf{s}[f]$ restore cubic polynomials. Denote $\bar{h}[k] := \max_{\nu\in\{k-2,\ldots,k+2\}} h[\nu]$ and $I[k] := [t[k-2], t[k+3]]$.

Theorem 2. *For* $t \in [t[k], t[k+1]]$, *the following error estimation holds:*

$$\max_{t\in[t[k],t[k+1]]} |f(t) - s[f](t)| \le \frac{35\bar{h}^4}{48} \max_{t\in I[k]} |f[t, k-1, k, k+1, k+2]|. \tag{7}$$

If the function $f(t)$ belongs to $C^4(I[k])$, then for $t \in [t[k], t[k+1]]$ we have

$$\max_{t\in[t[k],t[k+1]]} |f(t) - s[f](t)| \le \frac{35\,(\bar{h}[k])^4}{1152} \max_{t\in I[k]} \bigl|f^{(4)}(t)\bigr|. \tag{8}$$

If the grid $\boldsymbol{t}$ is uniform, at least locally (that means that $h[\nu] = h$ for all $\nu = k-2, \dots, k+2$), then

$$\max_{t\in[t[k],t[k+1]]} |f(t) - s[f](t)| \le \frac{35\,h^4}{1152} \max_{t\in I[k]} \bigl|f^{(4)}(t)\bigr|. \tag{9}$$

Moreover, $35/1152 \approx 0.0304$ is the least possible constant in inequality (9).

Proof. Set $D(t) := f[t, k-1, k, k+1, k+2]$ and $\tau = (t - t[k])/h[k]$, $\tau \in [0, 1]$. By using Eqs. (6) and (2), we can represent the remainder term of the approximation as follows:

$$\begin{aligned}\rho(t) := f(t) - s[f](t) &= f(t) - P^3(t)[k] + F[k-1]\,(1-\tau)^3 + F[k]\,\tau^3 \\ &= \alpha\, D(t) + \beta\, D(t[k+3]) + \gamma\, D(t[k-2]),\end{aligned}$$

where the coefficients are

$$\begin{aligned}\alpha = \omega_3[k-1](t), \qquad \beta = \chi[k]\,\tau^3, \qquad \gamma = \chi[k-1]\,(1-\tau)^3,\\ \chi[k] = \frac{(h[k])^2\,(h[k+1])^2\,(t[k+3] - t[k-1])}{3\,(t[k+2] - t[k])}.\end{aligned}$$

All the coefficients α, β, γ are non-negative, thus the intermediate value theorem implies $\rho(t) = D(\vartheta)\,H(\tau)$, with $\vartheta \in (t[k-2], t[k+3])$ and $H(\tau) := \alpha+\beta+\gamma$. Then, we have

$$\begin{aligned}H(\tau) = (h[k])^2\Bigl((h[k+1] + h[k](1-\tau))(h[k-1] + h[k]\tau)\,\tau\,(1-\tau)\\ + \frac{(h[k+1])^2\,(t[k+3] - t[k-1])}{3\,(h[k+1] + h[k])}\,\tau^3\\ + \frac{(h[k-1])^2\,(t[k+2] - t[k-2])}{3\,(h[k] + h[k-1])}\,(1-\tau)^3\Bigr)\\ \le \bar{h}^4\Bigl((2-\tau)(1+\tau)\tau(1-\tau) + \frac{2}{3}(\tau^3 + (1-\tau)^3)\Bigr) \le \frac{35}{48}\,\bar{h}^4,\end{aligned}$$

where $\bar{h} := \max_{\nu\in\{k-2,\dots,k+2\}} h[\nu]$. Consequently, for $t \in [t[k], t[k+1]$ we have the estimation

$$|\rho(t)| \le \frac{35\bar{h}^4}{48} \max_{\vartheta\in(t[k-2],t[k+3])} |f[\vartheta, k-1, k, k+1, k+2]|.$$

Equation (8) follows from Eq. (1). If $h[\nu] = h$ for $\nu = k-2, \dots, k+2$, then

$$H(\tau) = h^4\Bigl((2-\tau)(1+\tau)\tau(1-\tau) + \frac{2}{3}(\tau^3 + (1-\tau)^3)\Bigr) \le \frac{35}{48}\,h^4,$$

which implies Eq. (9). In this case we have $H(1/2) = 35/48\,h^4$. Therefore, 35/1152 is the least possible constant in Eq. (9). □

The estimate in Eq. (9) is sharp in the sense that it becomes an identity for the function $f(t) = t^4$ when $t = h(k+1/2)$. Note that the estimates in Eqs. (7)–(9) are local. That means that the estimates on a certain interval depend only on the function's behavior in a close vicinity of this interval. The sharp estimate in Eq. (9) for the uniform grid (non-local) was established in [11] by a technique different from that used in the proof of Theorem 2. The inequality in Eq. (7) makes it possible to establish sharp estimates of the approximation errors for functions from the classes $C^m(I[k])$, $m < 4$. In the case when the available samples $f[k]$ are noisy, the estimation in Eq. (7) enables us to evaluate the noise contribution into the approximation error.

3.3. Quasi-interpolating Cubic Splines at Finite Intervals

Typically, only a finite number of samples $f[\nu] = f(t[\nu])$, $\nu = 0, 1, \ldots, N$, of the function $f(t)$ is available. At the inner interval $[t[2], t[N-2]]$, the representation of the spline $s[f](t)$ given in Eq. (6) is valid. The following Theorem 3 provides a smooth extension of the spline to the boundaries of the sampling interval. As before, $P(t)[k]$ denotes the polynomial which interpolates the function $f(t)$ at the points $\{t[k-1], \ldots, t[k+2]\}$. We denote

$$\begin{aligned} A_0[f] &:= -f[0,1,2,3,4]\,\frac{(h[2])^2\,(t[4]-t[0])}{3\,h[1]\,(t[3]-t[1])},\\ A_N[f] &:= -f[N-4,N-3,N-2,N-1,N]\\ &\quad\times\frac{(h[N-3])^2\,(t[N]-t[N-4])}{3\,h[N-2]\,(t[N-1]-t[N-3])}. \end{aligned}$$

Theorem 3 ([17]). *The function*

$$\bar{s}[f](t) := \begin{cases} P(t)[1] + A_0[f]\,(t-t[1])_+^3, & t \in [t[0], t[2]];\\ s[f](t), & t \in [t[2], t[N-2]];\\ P(t)[N-2] + A_N[f]\,(t[N-1]-t)_+^3, & t \in [t[N-2], t[N]] \end{cases} \tag{10}$$

is a cubic spline that quasi-interpolates the function $f(t)$ on the finite grid $\mathbf{t}_N = \{t[\nu]\}_{\nu=0}^N$.

Remark 1. Five grid samples are needed for the computation of the spline $\bar{s}[f](t)$ at the interval $[t[1], t[2]]$ and only four samples for the interval $[t[0], t[1]]$. A similar situation takes place at the intervals $[t[N-2], t[N-1]]$ and $[t[N-1], t[N]]$.

Remark 2. The spline $\bar{s}[f](t)$ defined by Eq. (10) interpolates the function $f(t)$ at the points $t[0]$, $t[1]$ and $t[N-1]$, $t[N]$. At the intervals $[t[0], t[1]]$ and $[t[N-1], t[N]]$, the spline $\bar{s}[f](t)$ coincides with the interpolating polynomials $P(t)[1]$ and $P(t)[N-2]$, respectively.

Denote $I[0] := [t[0], t[k+4]]$.

Theorem 4 ([17]). *The following error estimate is true for $t \in [t[1], t[2]]$:*

$$\max_{t\in[t[1],t[2]]} |f(t) - \bar{s}[f](t)| \le \frac{\bar{h}^4(16-3\sqrt{2})}{12\sqrt{2}} \max_{t\in I[0]} |f[0,1,2,3,t]|,$$

where $\bar{h} := \max_{\nu\in\{0,1,2,3\}} h[\nu]$. If a function $f(t) \in C^4(I[0])$, then for $t \in [t[1], t[2]]$ we have

$$\max_{t\in[t[1],t[2]]} |f(t) - \bar{s}[f](t)| \le \frac{\bar{h}^4(16-3\sqrt{2})}{288\sqrt{2}} \max_{t\in I[0]} |f^{(4)}(t)|.$$

If the grid $\boldsymbol{t}$ is uniform, at least locally (it means that $h[\nu] = h$ for $\nu = 0, 1, 2, 3$), then

$$\max_{t\in[t[1],t[2]]} |f(t) - \bar{s}[f](t)| \le \frac{h^4(16-3\sqrt{2})}{288\sqrt{2}} \max_{t\in I[0]} |f^{(4)}(t)|. \tag{11}$$

The constant $(16-3\sqrt{2})/(288\sqrt{2}) \approx 0.0289$ is the least possible in inequality (11).

For $t \in [t[0], t[1]]$, the following error estimate is true:

$$\max_{t\in[t[0],t[1]]} |f(t) - \bar{s}[f](t)| \le \frac{\bar{h}^4}{24} \max_{t\in I[0]} |f^{(4)}(t)|.$$

The estimate is sharp even for the uniform grid. It becomes an identity for the function $f(t) = t^4$. Similar estimates hold at the interval $[t[N-2], t[N]]$.

3.4. Extrapolation of Signals Using Cubic Splines

Assume that a continuous function $f(t)$ is supported on $[t[0], t[N+1]]$ but only the samples $f[\nu] = f(t[\nu])$, $\nu = 0, \dots, N$, are available. In order to approximate the function $f(t)$ on the interval $I_r := [t[N], t[N+1]]$ and, in particular, to "predict" the sample $f[N+1]$, we extend the quasi-interpolating spline $\bar{s}[f](t)$ constructed in Theorem 3 to the interval I_r.

An apparent way to extend the spline $\bar{s}[f](t)$, which on $[t[N-1], t[N]]$ coincides with the polynomial $P(t)[N-2]$, is to define $\bar{s}[f](t) = P(t)[N-2]$ as $t \in I_r$. But, in order to achieve better "prediction", we apply the following correction.

Denote

$$C_r[f] := (t[N+1] - t[N-3])\,(t[N+1] - t[N-2])\,(t[N+1] - t[N-1])\,h[N],$$
$$A_r[f] = C_r[f]\,f[N-4, N-3, N-2, N-1, N]/(h[N])^3.$$

Define the extended spline $s_r[f](t)$ by

$$s_r[f](t) := \begin{cases} \bar{s}[f](t), & t \in [t[0], t[N]]; \\ P(t)[N-2](t) + A_r[f]\,(t - t[N])^3, & t \in I_r. \end{cases}$$

The function $s_r[f](t)$ is a cubic spline because it is piece-wise cubic polynomial and continuous together with its first and second derivatives at the point $t[N]$.

Denote $h_r := \max_{\nu\in\{N-3,\dots,N\}} h[\nu]$.

Proposition 2. *If the fifth-order divided difference satisfies*

$$|f[N-4, N-3, N-2, N-1, N, N+1]| \le F,$$

then the extrapolation error at the point $t[N+1]$ is estimated by

$$|f(t[N+1]) - s_r[f](t[N+1])| \le 120F\, h_r^5.$$

If f has a continuous fifth-order derivative on the interval $[t[N-4], t[N+1]]$, then the estimation

$$|f(t[N+1]) - s_r[f](t[N+1])| \le \max_{t\in[t[N-4],t[N+1]]} |f^{(5)}(t)|\, h_r^5$$

is true. In particular, if f coincides with a fourth-degree polynomial on the interval $[t[N-4], t[N+1]]$, then $s_r[f](t[N+1]) = f(t[N+1])$.

A similar design is carried out at the left hand side of the sampling interval.

Figure 1 illustrates the extrapolation. The function $f(t)$ is a fourth-degree polynomial. The spline is constructed using 6 randomly spaced grid points. The first and last samples were taken at $t[0] = 0.49$ and $t[N] = 9$ and the spline was extrapolated to $t[-1] = -3$ and $t[N+1] = 12$.

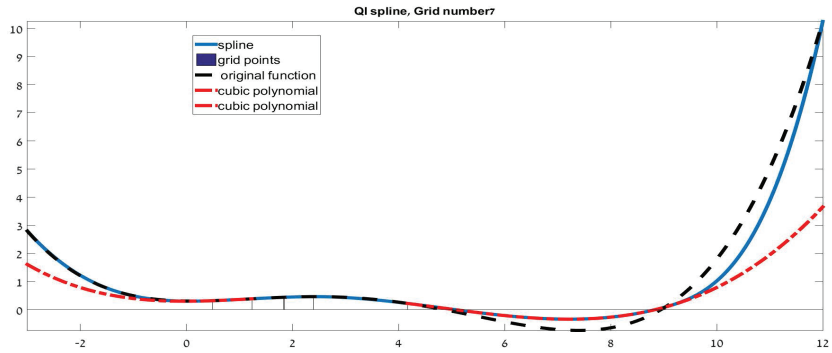

Figure 1. Extrapolation of the spline $\mathbf{s}[f]$ from six grid points denoted by bars. Dashed line denotes the original function (fourth-degree polynomial) and solid line denotes the spline $s[f](t)$.

3.5. Remarks on the Real-time Spline Computation

Due to the fact that no more than 6 adjacent grid samples are needed for the computation of the spline $\mathbf{s}[f]$ at a point t, the spline can be computed in

real time. It means that the spline follows the samples that arrive one after another at random time moments.

Assume that the samples of a function $f(t)$, $f[k]$, $k = 0, \dots, N$, are available, where $f[k] = f(t[k])$. Then, the spline $\mathbf{s}[f]$ can be designed on the interval $[t[0], t[N]]$ in line with the scheme described in Sections 3.2 and 3.3 in the following way:

- At the inner subinterval $[t[2], t[N-2]]$, the spline is constructed by a regular 6-samples algorithm from Section 3.2, while at the intervals $[t[0], t[2]]$ and $[t[N-2], t[N]]$ the 5- and 4-samples extension formulas from Section 3.3 are utilized.
- When the sample $f[N+1] = f(t[N+1])$ arrives, the spline at the interval $[t[N-2], t[N-1]]$ is recomputed by utilizing the 6 samples $f[k]$, $k = N-4, \dots, N+1$. The spline on the interval $[t[N-1], t[N]]$, which was constructed by using the 4 samples $f[k]$, $k = N-3, \dots, N$, is recomputed with the new set of samples $f[k]$, $k = N-3, \dots, N+1$. The spline is extended to the interval $[t[N], t[N+1]]$ by using the samples $f[k]$, $k = N-2, \dots, N+1$. The spline remains unchanged on the interval $[t[0], t[N-2]]$. This fact substantiates our claim that the spline follows the arriving samples.

4. Quadratic Quasi-interpolating Splines

In this section we briefly outline quadratic quasi-interpolating splines keeping in mind their utilization for the wavelet transform design. Recall that nodes of the cubic splines approximating a function f coincided with the sampling points. This is not the case with the quadratic splines. Typically, the sampling points are located at the midpoints between the spline's nodes. It turns out to be an advantage on the uniform grids by providing a "super-convergence" property.

4.1. Arbitrary Grids

In addition to the grid $\mathbf{t} = \{t[k]\}$, we introduce the grid $\mathbf{g} = \{g[k]\}$, where $g[k] = (t[k+1] + t[k])/2$. The function f is sampled on the grid $\mathbf{g}$, that is $f[k] = f(g[k])$. As before, $h[k] = t[k+1] - t[k]$.

Remark 3. Practically, the grid $\mathbf{t}$ is adjusted to the sampling grid $\mathbf{g}$. It is done by an apparent recursion: $t[k+1] = 2g[k] - t[k]$.

Quadratic B-spline is:

$$b^3(t)[k] = (t[k+3] - t[k]) \sum_{\nu=0}^{3} \frac{(t - t[k+\nu])_+^2}{w_3'[k](t[k+\nu])}.$$

It is supported on the interval $(t[k], t[k+3])$. Its nodes are located at the grid points $t[k], \dots, t[k+3]$.

Quadratic splines on the grid $\mathbf{t} = \{t[k]\}$ are represented via the B-splines. For $t \in [t[k], t[k+1]]$, a quadratic spline $s(t)$ is given by

$$s(t) = \sum_{\nu\in\mathbb{Z}} q[\nu+1]\, b^3(t)[\nu] = \sum_{\nu=k-2}^{k} q[\nu+1]\, b^3(t)[\nu]. \tag{12}$$

The simplest spline approximating a function f from its samples on the grid $\mathbf{g}$ is $s_0(t) = \sum_{\nu=k-2}^{k} f[\nu+1]\, b(t)[\nu]$. This spline restores first-degree polynomials. In order to achieve the restoration of quadratic polynomials, which upgrades the spline to become a quasi-interpolating one, the coefficients in Eq. (12) should be chosen as

$$q[\nu] = f[\nu] - \frac{h[\nu]^2}{4} f[\nu-1, \nu, \nu+1].$$

We discuss the quadratic splines on uniform grids in more detail.

4.2. Uniform Grids

Assume the grids $\mathbf{t}$ and $\mathbf{g}$ are uniform. Let the sampling grid points be $g[k] = hk$, $k \in \mathbb{Z}$ and $f[k] = f(hk)$. Then $t[k] = h(k-1/2)$. The centered quadratic B-spline is

$$B_h^3(t) = \frac{1}{2h^2} \sum_{k=0}^{3} (-1)^k \binom{3}{k} \Big(t + h\Big(\frac{3}{2} - k\Big)\Big)_+^2.$$

The quasi-interpolating quadratic spline restoring quadratic polynomials, is

$$S^3[f](t) = \sum_{\nu=k-1}^{k+1} \Big(f[\nu] - \frac{1}{8}\Delta^2 f[\nu-1]\Big) B_h^3(t-h\nu) = \sum_{\nu=k-2}^{k+2} f[\nu]\, L_h^3(t-h\nu),$$

$$L_h^3(t) = \frac{1}{8}\left(10B_h^3(t) - B_h^3(t-h) - B_h^3(t+h)\right) \tag{13}$$

$$= \begin{cases} -\tau^2/16, & \text{if } k=-2; \\ (12\tau^2 - 2\tau - 1)/16 & \text{if } k=-1; \\ (-22\tau^2 + 22\tau + 9)/16 & \text{if } k=0; \\ (12(1-\tau)^2 - 2(1-\tau) - 1)/16, & \text{if } k=1; \\ -(1-\tau)^2/16 & \text{if } k=2, \end{cases} \qquad t = h(k - 1/2 + \tau).$$

Grid samples of the compactly supported spline $L_h^3(t)$ are given in Table 1. As before, $t = h(k-1/2+\tau)$, $\tau \in [0,1)$. Denote by $P[k](t)$ the quadratic polynomial which interpolates the function f at the grid points

$\{h(k-1), h(k), h(k+1)\}$. The remainder term of the approximation is:

$$R(t) = f(t) - P[k](t) = h^3\, f[t, k-1, k, k+1]\, \psi(\tau),$$
$$\psi(\tau) := \Big(\tau + \frac{1}{2}\Big)\Big(\tau - \frac{1}{2}\Big)\Big(\tau - \frac{3}{2}\Big). \tag{14}$$

$-2h$	$-3h/2$	$-h$	$-h/2$	0	$h/2$	h	$3h/2$	$2h$
-1/64	-1/16	1/16	9/16	58/64	9/16	1/16	-1/16	-1/64

Table 1. Grid samples of the spline $L_h^3(t)$.

The remainder term is zero at the grid points $\{h(k-1), h(k), h(k+1)\}$. Therefore, the spline approximating the function $R(t)$ is

$$\begin{aligned} S^3[R](t) &= R[k-2]\, L_h^3\Big(\frac{3h}{2} + h\tau\Big) + R[k+2]\, L_h^3\Big(-\frac{5h}{2} + h\tau\Big) \\ &= -\frac{R[k-2]}{16}\,(1-\tau)^2 - \frac{R[k+2]}{16}\,\tau^2 \\ &= \frac{\Delta^3 f[k-2]}{16}\,(1-\tau)^2 - \frac{\Delta^3 f[k-1]}{16}\,\tau^2. \end{aligned}$$

Since the spline $\mathbf{S}^3[f]$ restores quadratic polynomials, we have

$$S^3[f](t) = P(t) + S^3[R](t) = P(t) + \frac{\Delta^3 f[k-2]}{16}\,(1-\tau)^2 - \frac{\Delta^3 f[k-1]}{16}\,\tau^2, \tag{15}$$

and the remainder term of the spline approximation is:

$$\begin{aligned} \rho(t) &:= f(t) - S^3[f](t) = R(t) - S^3[R](t) \\ &= h^3\, f[t, k-1, k, k+1]\, \psi(\tau) - \frac{\Delta^3 f[k-2]}{16}\,(1-\tau)^2 + \frac{\Delta^3 f[k-1]}{16}\,\tau^2. \end{aligned}$$

If the function f belongs to C^3, then the remainder term is

$$\rho(t) = \frac{h^3}{6}\Big(f^{(3)}(\xi_0)\,\psi(\tau) + \frac{3}{8}\,f^{(3)}(\xi_1)\tau^2 - \frac{3}{8}\,f^{(3)}(\xi_{-1})(1-\tau)^2\Big),$$

where $\xi_i \in [h(k-2), h(k+2)]$, $i = -1, 0, 1$, and the function ψ is defined in Eq. (14).

Super-convergence property. Take $f(t) = t^3$. Then, $f^{(3)}(t) = 6$ and the remainder term is

$$\rho(t) = h^3\Big(\psi(\tau) + \frac{3}{8}\,\tau^2 - \frac{3}{8}\,(1-\tau)^2\Big).$$

If $t = hk \Longleftrightarrow \tau = 1/2$, then $\psi(\tau) = 0$ and, consequently, $\rho(\tau) = 0$. If $t = h(k \mp 1/2) \Longleftrightarrow \tau = 0/1$, then $\psi(\tau) = \pm 3/8$ and, consequently, $\rho(\tau) = 0$. Thus, at the sampling grid points $\{hk\}$ and at the spline's nodes $\{h(k+1/2)\}$, $k \in \mathbb{Z}$, the quadratic spline $\mathbf{S}^3[f]$ restores cubic polynomials, although, globally it restores only quadratic polynomials. We refer to this fact as to the *super-convergence property* of the quadratic quasi-interpolating splines. This property is valuable for the wavelet design.

4.3. Quasi-interpolating Quadratic Splines on Finite Intervals

Assume that only a finite number of samples $f[\nu] = f(h\nu)$, $\nu = 0, 1, \ldots, N$, of the function $f(t)$ are available. At the inner interval $[3h/2, h(N-3/2)]$, the representation of the spline $\mathbf{S}^3[f]$ given in Eq. (15) is valid. In particular, for $t \in [(N-5/2)h, (N-3/2)h]$ the spline is $S[f](t) = P[N-2](t) + S[R_{[2]}](t)$, where $P[N-2](t)$ is the quadratic polynomial interpolating f at the points $(N-3)h, (N-2)h, (N-1)h$ and $R_{[2]}(t) = f(t) - P[N-2](t)$.

Introduce an additional grid point $g[N+1] = h(N+1)$ and define $f[N+1] := P[N-1](h(N+1))$, where $P[N-1](t)$ is the quadratic polynomial interpolating $f(t)$ at $(N-2)h, (N-1)h, Nh$. Denote $R_{[1]}(t) = f(t) - P[N-1](t)$. The function $R_{[1]}(t) = 0$ when $t = Mh$, $M = N-2, \ldots, N+1$. Then, on the interval $[h(N-3/2), h(N-1/2)]$, $t = h(N-3/2+\tau)$, the spline is defined as follows:

$$\begin{aligned} S_N[f](t) &= P[N-1](t) - \frac{R_{[1]}[N-3]}{16}\,(1-\tau)^2 - \frac{R_{[1]}[N+1]}{16}\,\tau^2 \\ &= P[N-1](t) + \frac{\Delta^3 f[N-3]}{16}\,(1-\tau)^2. \end{aligned}$$

At the point $t = h(N-1/2) \Longleftrightarrow \tau = 1$ the spline $S_N[f](t) = P[N-1](t)$ and its derivative $S'_N[f](t) = P'[N-1](t)$. Therefore, it is natural to extend the spline to the interval $[h(N-1/2), hN]$ as $S_N[f](t) = P[N-1](t)$. Thus, on the interval $[h(N-3/2), hN]$ the spline is defined as follows:

$$S_N[f](t) = P[N-1](t) + \frac{\Delta^3 f[N-3]}{16}\,(1-\tau)_+^2, \qquad \tau = t/h - N + 3/2.$$

The reminder term of the spline approximation is

$$\rho(t) = h^3\, f[t, N-2, N-1, N]\psi(\tau) - \frac{\Delta^3 f[k-2]}{16}\,(1-\tau)_+^2.$$

Similarly, on the interval $[0, 3h/2]$ the spline is defined as follows:

$$S_0[f](t) = P[1](t) - \frac{\Delta^3 f[0]}{16}\,\tau_+^2, \qquad \tau = t/h - 1/2.$$

Combining the above equations, we define the spline $\bar{S}^3[f]$ as follows

$$\bar{S}^3[f](t) = \begin{cases} S_N[f](t), & \text{if } \ t \in [h(N-3/2), hN]; \\ S^3[f](t), & \text{if } \ t \in [3h/2, [h(N-3/2)]; \\ S_0[f](t), & \text{if } \ t \in [0, 3h/2]. \end{cases} \tag{16}$$

Remark 4. The spline $\bar{S}^3[f](t)$ defined by Eq. (16) interpolates the function $f(t)$ at boundary points $g[0]$ and $g[N]$. On the intervals $[0, h/2]$ and $[h(N-1/2), hN]$, the spline $\bar{s}[f](t)$ coincides with the quadratic interpolating polynomials $P(t)[1]$ and $P(t)[N-1]$, respectively. Note that the superconvergence property does not persist at the points $h/2$ and $h(N-1/2)$.

4.4. Extrapolation of Quasi-interpolating Quadratic Splines

Assume that a continuous function f is supported on the interval $[0, h(N+1)]$ but only the samples $f[\nu] = f(h\nu)$, $\nu = 0, \dots, N$, are available. We extend the spline $\bar{S}^3[f]$ defined in Eq. (16) onto the interval $I_r := [hN, h(N+1])$ by

$$S_r^3[f](t) := P(t)[N-1](t) + B_r[f]\,(t - hN)^2,$$

where the constant $B_r[f]$ is to be determined in a way that, under some regularity conditions, the difference between the spline and the sample $f[N+1]$ has become minimal. Assume for a moment that $f[N+1]$ is known. The difference is

$$\rho := f[N+1] - S_r^3[f](h(N+1)) = h^3\Big(\Delta^3 f[N-2] - \frac{B_r[f]}{h}\Big).$$

We choose $B_r = h\Delta^3 f[N-3]$. Then, the difference becomes

$$\rho = h^3\big(\Delta^3 f[N-2] - \Delta^3 f[N-3]\big) = -h^4\Delta^4 f[N-3].$$

If $f(t)$ coincides with a cubic polynomial on the interval $[h(N-3), h(N+1)]$, then $S_r^3[f](h(N+1)) = f[N+1]$. Similarly, the extrapolation of the spline to the interval $I_l = [-h, 0]$ is defined by

$$S_l^3[f](t) := P(t)[1](t) - h\,\Delta^3 f[0]t^2.$$

5. Spline-based Wavelet Transform

The *Lifting Scheme* introduced in [18, 19], is a method that constructs bi-orthogonal wavelet transforms and provides their efficient implementation. The main feature of the lifting scheme is that all the constructions are derived directly in the spatial domain and therefore can be custom designed to more general, irregular settings such as non-uniformly spaced data samples and bounded intervals. In addition, the lifting scheme fits well to the real-time execution of the wavelet transforms. In this section, we outline the lifting scheme and describe how to use the local quasi-interpolating splines designed in Sections 3 and 4 for the construction of wavelet transforms of discrete-time signals and in the situation when signals' samples arrive subsequently at random time moments.

5.1. Discrete Lifting Wavelet Transform

Decomposition. The lifting wavelet decomposition of a signal from the space l_1 consists of four steps:

1. **Split**: The signal $\mathbf{f} = \{f[k] = f[t[k]]\}$ is split into the even and odd subarrays such that $\mathbf{f} = \mathbf{e} \cup \mathbf{o}$ where $\mathbf{e} := \{e[k] = f[2k]\}$ and $\mathbf{o} := \{o[k] = f[2k+1]\}$, respectively.

2. **Predict**: The even array $\mathbf{e}$ is used to approximate (predict) the odd array $\mathbf{o}$. Then, the array $\mathbf{o}$ is replaced by the array $\mathbf{d} = \mathbf{o} - \mathcal{P}\,\mathbf{e}$, where $\mathcal{P}$ is a linear *prediction* operator. If the predictor is correctly chosen, then this step decorrelates the signal and reveals its high-frequency components.

3. **Update (Lifting)**: The even array $\mathbf{e}$ is updated by using the new odd array $\mathbf{d}$ to get $\mathbf{a} = \mathbf{e} + \mathcal{U}\,\mathbf{d}$, where $\mathcal{U}$ is a linear *updating* operator, to get $\mathbf{a} = \mathbf{e} + \mathcal{U}\,\mathbf{d}$.

4. **Normalization**: Finally, the *smoothed* $\mathbf{y}^0$ and the *details* $\mathbf{y}^1$ transform coefficient arrays, are obtained by the normalization $\mathbf{y}^0 = \sqrt{2}\,\mathbf{a}$, $\mathbf{y}^1 = \mathbf{d}/\sqrt{2}$.

Reconstruction. Reconstruction of the signal $\mathbf{f}$ from the arrays $\mathbf{y}^0$ and $\mathbf{y}^1$ is implemented in a reverse order:

1. **Undo Normalization**: $\mathbf{a} = \mathbf{y}^0/\sqrt{2}$, $\mathbf{d} = \sqrt{2}\,\mathbf{y}^1$.

2. **Undo Lifting**: The even array is restored by $\mathbf{e} = \mathbf{a} - \mathcal{U}\,\mathbf{d}$.

3. **Undo Predict**: The odd array component is restored by $\mathbf{o} = \mathbf{d} + \mathcal{P}\,\mathbf{e}$.

4. **Undo Split**: Restoration of the signal from its even and odd arrays by $\mathbf{f} = \mathit{Merge}\{\mathbf{e}, \mathbf{o}\}$.

The lifting transform is perfectly invertible with any choice of the operators $\mathcal{P}$ and $\mathcal{U}$. The multilevel wavelet transform is achieved by the iterated application of the lifting operations to the smoothed coefficient arrays $\mathbf{y}^0$, thus producing the coefficient array $\{\mathbf{y}^0_{[m]} \cup \mathbf{y}^1_{[m]} \cup \cdots \cup \mathbf{y}^1\}$. The prediction and updating operators can be different for different decomposition levels.

5.2. Spline-based Prediction and Updating Operators

Wavelet transforms generated by the lifting scheme are determined by the choice of the prediction $\mathcal{P}$ and the updating $\mathcal{U}$ operators. Splines provide flexible tools for the design of such operators. The idea is to construct a spline on the even sub-grid $g[2k]$[†], which (quasi-)interpolates the samples of the signal $\mathbf{e}$. The values of this spline at the odd grid points $g[2k+1]$ are used for the prediction of the odd samples $\mathbf{o}$. The next step is to construct a spline on the odd grid points, which (quasi-)interpolates the samples of the prediction-error signal $\mathbf{d}$. Then, the values of this splines at the even grid points are computed. These values are used for updating the even samples $\mathbf{e}$.

In the case of equally spaced samples, calculations are reduced to low-pass filtering of the corresponding arrays [4]. Application of wavelet transforms to signals sampled on finite grids and, in particular, to images, requires to extend the signals beyond their boundaries, otherwise distortion appears near

[†]For cubic splines the sampling grid $\mathbf{g}$ coincides with the nodes' grid $\mathbf{t}$.

the boundaries. Various extension schemes have been developed to deal with the boundary effects of finite-length signals: zero padding, periodic extension and symmetric extension are basic extension methods ([5]). However, quasi-interpolating splines on finite intervals designed in Sections 3.3 and 4.3, make it possible to implement wavelet transforms of signals sampled on bounded intervals without extending the signals beyond their boundaries.

The prediction and updating operations are reduced to the design of the splines $\mathbf{s}[f]$ on different grids and computation of their values at intermediate points.

5.3. Prediction and Updating Operators on Unlimited Grids

Cubic splines. To highlight the structure of the operators, we use the representation of the quasi-interpolating spline via the splines L^4 as in Eq. (4):

$$s[f](t) = \sum_{\nu=k-4}^{k+1} f[\nu+2]\, L^4(t)[\nu]. \tag{17}$$

As before, $\mathbf{e} = \{e[k] = f[2k]\}$ and $\mathbf{o} = \{o[k] = f[2k+1]\}$. Denote by L^4_e and L^4_o the splines L^4 defined on the even and odd sub-grids, respectively. Application of the prediction operator to the even sub-array $\mathcal{P}\,\mathbf{e}$ consists of computation of values of the spline $\mathbf{s}[e]$ at the odd grid points. Equation (17) implies that these values are

$$s[e](t[2k+1]) = \sum_{\nu=k-4}^{k+1} P_{[k]}[\nu]\, e[\nu], \qquad P_{[k]}[\nu] := L^4_e(t[2k+1])[2\nu]. \tag{18}$$

Application of the updating operator to the $\mathcal{U}\,\mathbf{d}$ consists of computation of values of the spline $s[d]$ at the even grid points:

$$s[d](t[2k]) = \sum_{\nu=k-5}^{k} U_{[k]}[\nu]\, d[\nu], \qquad U_{[k]}[\nu] := L^4_o(t[2k])[2\nu+1]. \tag{19}$$

When processing a single signal and, especially, in the real-time mode, the computations by fast algorithms described in Section 3 are preferable. However, in a case when either multiple signals defined on the same grid or an array of row signals are processed, the necessary values of the splines $L^4(t)$ can be pre-computed. Then, according to Eqs. (18) and (19), the computations of the splines' values $s[e](t[2k+1])$ and $s[d](t[2k])$ can be implemented by the six-tap weighted moving averages of the signal's samples with weights consisting of values of the splines $L^4(t)$ given in Eq. (5). Thus, computations are significantly accelerated.

When the grid is uniform, the prediction and updating operations are reduced to the low-pass filtering of the signals $\mathbf{e}$ and $\mathbf{d}$. We illustrate this situation on the quadratic splines.

Quadratic splines. For simplicity, assume that a signal $\mathbf{f}$ is sampled on the grid $\mathbf{g} = \{k\}$, $k \in \mathbb{Z}$. Thus, $f[k] = f(k)$, $e[k] = f(2k)$ and $o[k] = f(2k+1)$. Then, due to Eq. (13) and Table 1, values of the spline $\mathbf{S}^3[e]$ at odd grid points are

$$\begin{aligned} S^3[e](2(k+1/2)) &= \sum_{\nu=k-2}^{k+2} e[\nu]\, L_2^3(2(k+1/2-\nu)), \\ &= -\frac{e[k-1]}{16} + \frac{9e[k]}{16} + \frac{9e[k+1]}{16} - \frac{e[k+2]}{16}. \end{aligned} \tag{20}$$

It is seen from Eq. (20) that the array $\{S^3[e](2(k+1/2))\}$, $k \in \mathbb{Z}$ arrives as a result of filtering the signal $\mathbf{e}$ by the low-pass filter, whose impulse response is

-1	0	1	2
-1/16	9/16	9/16	-1/16

Similarly, the array $\{S^3[d](2k)\}$, $k \in \mathbb{Z}$ arrives as a result of filtering the signal $\mathbf{d}$ by the low-pass filter, whose impulse response is

-2	-1	0	1
-1/16	9/16	9/16	-1/16

5.4. Prediction and Updating Operations on Limited Grids

Cubic splines. Assume that the function f is sampled on a bounded interval, thus $f[k] = f(t[k])$, $k = 0, \ldots, N$ are available.

Assume that N is an odd integer.

Prediction: The odd samples $f[2k+1]$, $k = 2, \ldots, (N-1)/2-3$, are predicted using the formulas in Eq. (18). The samples $f[1]$ and $f[3]$ as well as $f[N-4]$ and $f[N-2]$ are predicted by the values of the spline $s[e](t)$ at the respective grid points $t[1]$ and $t[3]$, as well as $t[N-4]$ and $t[N-2]$. These values are computed using the formulas in Eq. (10). The remaining sample $f[N]$ is predicted by the extrapolation of the spline $s[e](t)$ to the grid point $t[N]$ using the algorithm described in Section 3.4. Thus, we derive the differences $d[k] = f[2k+1] - s[e](t[2(k+2)+1])$, $k = 0, \ldots, (N-1)/2$.

Update: The even samples $f[2k]$, $k = 3, \ldots, (N-1)/2-2$, are updated by the values of the spline $s[d](t[2k])$ using the formulas in Eq. (19). To update the samples $f[2]$, $f[4]$, $f[N-3]$, $f[N-1]$, the values of $s[d](t)$ are computed using the formulas in Eq. (10). The remaining sample $f[0]$ is predicted by the extrapolation of the spline $s[d](t)$ to the grid point $t[0]$ using the algorithm described in Section 3.4.

A similar design is performed when N is an even integer and for the quadratic splines.

To implement the next step of the wavelet transform, the above operations are applied to the smooth array $\mathbf{y}^0$. As a result, the arrays $\mathbf{y}^0_{[2]}$ and $\mathbf{y}^1_{[2]}$ are produced. This process repeats itself.

Discrete vanishing moments. The spline-based wavelet transform possesses the local "discrete vanishing moments" property, which is formulated next.

Proposition 3. *Assume that $f[\nu]$, $\nu = 2k-4, \ldots, 2k+6$, are samples of the cubic polynomial $f[\nu] = P^3(t[\nu])$. Then, the wavelet transform coefficient $y^1[k] = 0$. If the cubic polynomial $P^3(t[\nu])$ is sampled on a uniform grid $\{hk\}$, then the relation $y^1[k] = 0$ is valid for the wavelet transforms originating from quadratic splines.*

Proof. The first-level transform coefficients are $y^1[k] = (f[2k+1] - s[e](t[2k+1]))/\sqrt{2}$. Theorem 2 implies that this difference is zero. As for the quadratic splines, the relation $y^1[k] = (f[2k+1] - S^3[e]((2k+1)h))/\sqrt{2} = 0$ follows from the super-convergence property of the quadratic quasi-interpolating splines on uniform grids. □

Remark 5. Theorem 4 and Proposition 2 imply that the "discrete vanishing moments" property remains valid near the boundaries of the interval.

We claim that the wavelet transforms originating from cubic splines of arbitrarily sampled signals possess four local discrete vanishing moments (LDVM) in a sense of Proposition 3. The same is true for the transforms originating from quadratic splines in the case when the signals are sampled uniformly.

Remark 6. We emphasize that, although, generally, quadratic-splines-based wavelet transforms have three LDVMs, on the uniform grids they have the same number four LDVMs as the transforms originating from cubic splines.

5.5. Real-time Execution of the Wavelet Transform

The transform scheme described in Section 5.4 makes it possible to execute the wavelet transform of a signal in real time. It means that the samples $f[\nu]$ of a signal arrive sequentially at random time moments $t[\nu]$, $\nu = 0, 1, \ldots$, while the arrival of a new sample requires recomputing of only a few adjacent transform coefficients. We outline the procedure for the cubic-spline transforms. The scheme for the quadratic splines is similar.

- Assume that at the time $t[N]$ the samples $f[\nu]$, $\nu = 0, \ldots, N$, arrived already and N is an odd number. Then, the one-level decomposition

is implemented as was described in Section 5.4. Recall that the initial coefficients $y^0[0]$, $y^0[1]$, $y^0[2]$ and $y^1[0]$, $y^1[1]$ are produced by using the formulas in Eq. (10) and the extrapolation algorithm. The coefficients $y^0[\nu]$, $\nu = 3, \ldots, (N-1)/2-2$, and $y^1[\nu]$, $\nu = 2, \ldots, (N-1)/2-3$, are computed by using the regular formulas in Eq. (18). The coefficients $y^0[(N-1)/2-1]$, $y^0[(N-1)/2]$, and $y^1[(N-1)/2-1]$, $y^1[(N-1)/2-2]$ are produced by using the formulas in Eq. (10). The coefficient $y^1[(N-1)/2]$ is derived by the extrapolation algorithm. The number $(N+1)/2$ of smooth coefficients $y^0[\nu]$ is the same as the number of the detail coefficients $y^1[\nu]$.

- When the sample $f[N+1]$ arrives, the new smooth coefficient $y^0[(N+1)/2]$ is derived by updating the even sample $f[N+1]$ using the extrapolation of the spline $s[d](t)$ to the grid point $t[N+1]$. In addition, the detail coefficient $y^1[(N-1)/2-2]$ is recomputed in a regular way by prediction from 6 even samples $f[2\nu]$, $\nu = (N-1)/2-4, \ldots, (N+1)/2$. The rest of the detail and the smooth coefficients remain unchanged. The number of the smooth coefficients $y^0[\nu]$ is $(N+3)/2$ while the number of the detail coefficients $y^1[\nu]$ is $(N+1)/2$.

- When the sample $f[N+2]$ arrives, the new detail coefficient $y^1[(N+1)/2]$ is derived by predicting the odd sample $f[N+2]$ using the extrapolation of the spline $s[e](t)$ to the grid point $t[N+2]$. In addition, the smooth coefficient $y^0[(N-1)/2-1]$ is recomputed in a regular way by updating from 6 numbers $d[\nu]$, $\nu = (N-1)/2-4, \ldots, (N+1)/2$. The rest of the detail and smooth coefficients remain unchanged. The number $(N+3)/2$ of smooth coefficients $y^0[\nu]$ is the same as the number of the detail coefficients $y^1[\nu]$.

At the same time, the above procedures are applied to the smooth coefficient array $y^0[\nu]$, $\nu = 0, \ldots, (N-1)/2$, to obtain the arrays $\{y^0_{[2]}[\nu]\}$ and $\{y^1_{[2]}[\nu]\}$, and so on.

Summary of the paper's contributions.

- Simple algorithm for the real-time computation of the local cubic and quadratic quasi-interpolating splines including a smooth extension of the splines to the boundaries of the sampling interval.

- Formulas for the extrapolation of the splines beyond the sampling interval, while retaining the approximation accuracy. The formulas can be used for the *prediction-correction* signal processing.

- Sharp estimates of the approximation errors for the local cubic quasi-interpolating splines.

- Design of wavelet transforms of the arbitrarily sampled signals based on the local quasi-interpolating splines. The wavelet transforms have four local "discrete vanishing moments" in the sense of Proposition 3.

- Real-time scheme of the execution of the wavelet transforms of signals whose samples arrive dynamically and sequentially at random time moments.

References

[1] M. Abramowitz and I. A. Stegun, "Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables", Dover, New York, 1972.

[2] A. Aldroubi, C. Cabrellib and U. M. Molterb, Wavelets on irregular grids with arbitrary dilation matrices and frame atoms for $L^2(\mathbb{R}^d)$, *Appl. Comput. Harmon. Anal.* **17** (2004), no. 2, 119–140.

[3] A. Averbuch, P. Neittaanmäki and V. A. Zheludev, "Splines and Spline Wavelet Methods with Application to Signal and Image Processing. Volume I: Periodic Splines", Springer, 2014.

[4] A. Averbuch, P. Neittaanmäki and V. A. Zheludev, "Splines and Spline Wavelet Methods with Application to Signal and Image Processing. Volume II: Non-periodic Splines", Springer, 2015.

[5] C. M. Brislawn, Classification of nonexpansive symmetric extension transforms for multirate filter banks, *Appl. Comput. Harmon. Anal.* **3** (1996), no. 4, 337–357.

[6] T. Cai and L. Brown, Wavelet shrinkage for nonequispaced samples, *Ann. Statist.* **26** (1998), no. 5, 1783–1799.

[7] W. Dahmen, J. M. Carnicer, and J. M. Peña, Local decomposition of refinable spaces and wavelets, *Appl. Comput. Harmon. Anal.* **3** (1996), 127–153.

[8] I. Daubechies, I. Guskov, P. Schroder and W. Sweldens, Wavelets on irregular point sets, *Philos. Trans. R. Soc. Lond. Ser. A* **357** (1999), 2397–2413.

[9] C. de Boor, "A Practical Guide to Splines", Springer, New York, 1978.

[10] C. de Boor, Divided differences, *Surv. Approx. Theory* **1** (2005), 46–69.

[11] N. P. Korneichuk, Approximations by local splines of minimal defect, *Ukrainian Math. J.* **34** (1982), no. 5, 502–505.

[12] Z. Liu, Y. Mi and Y. Mao, An improved real-time denoising method based on lifting wavelet transform, *Measurement Science Review* **14** (2014), no. 3, 152–159.

[13] T. Lyche and L. L. Schumaker, "Local Spline Approximation Methods", MRC Technical Summary Report, Mathematics Research Center, University of Wisconsin, 1974.

[14] I. J. Schoenberg, Contributions to the problem of approximation of equidistant data by analytic functions, *Quart. Appl. Math.* **4** (1946), 45–99, 112–141, Parts A and B.

[15] L. L. Schumaker, "Spline Functions: Basic Theory", John Wiley & Sons, New York, 1981.

[16] J. Stoer and R. Bulirsch, "Introduction to Numerical Analysis", Second edition, Springer-Verlag, New York, 1993.

[17] M. G. Suturin and V. Zheludev, On the approximation on finite intervals and local spline extrapolation, *Russian J. Numer. Anal. Math. Modelling* **9** (1994), no. 1, 75–89.

[18] W. Sweldens, The lifting scheme: A custom-design construction of biorthogonal wavelets, *Appl. Comput. Harmon. Anal.* **3** (1996), no. 2, 186–200.

[19] W. Sweldens, The lifting scheme: A construction of second generation wavelets, *SIAM J. Math. Anal.* **29** (1997), no. 2, 511–546.

[20] R. Xia, K. Meng, F. Qian and Z. L. Wang, Online wavelet denoising via a moving window, *Acta Automat. Sinica* **33** (2007), no. 9, 897–901.

[21] Yu. S. Zav'yalov, B. I. Kvasov and V. L. Miroshnichenko, "Methods of Spline-functions", Nauka, Moscow, 1980 [in Russian].

[22] V. Zheludev, Local spline approximation on arbitrary meshes, *Sov. Math.* **8** (1987), 16–21.

Amir Averbuch

School of Computer Science
Tel Aviv University
Tel Aviv 69978
ISRAEL
E-mail: amir1@post.tau.ac.il

Pekka Neittaanmäki

Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, P.O. Box 35 (Agora)
FINLAND
E-mail: pn@jyu.fi

Valery Zheludev

School of Computer Science
Tel Aviv University
Tel Aviv 69978
ISRAEL
E-mail: zhel@post.tau.ac.il

and

Department of Mathematical Information Technology
University of Jyväskylä
Jyväskylä, P.O. Box 35 (Agora)
FINLAND