Local Interpolation by High Resolution Subdivision Schemes

Daniel Lemire

Acadia University

E-mail address: lemire@ondelette.com

URL: http://www.ondelette.com/acadia/

Subdivision schemes are used to interpolate data samples locally. By using temporary placeholders on a dense grid, we improve one of the best known subdivision scheme (Deslauriers-Dubuc). Interpolated values require 2 steps to stabilize as they are first interpolated on a coarse scale through a tetradic filter and then on a finer scale using a dyadic filter. The interpolants are C^1 and can be made to reproduce polynomials of degree 4 unlike regular subdivision schemes. These generalized interpolatory subdivision schemes have minimal support and no additional memory requirement.

Subdivision Schemes, Fourier transform, CAGD, Wavelets, Lagrange Interpolation

Introduction

Interpolatory subdivision schemes interpolate a discrete set of data points in a local manner, that is, the value of the interpolation function at a given point depends on a small number of nearby data points. The classical dyadic algorithm introduced by Deslauriers and Dubuc [**5, 2**] finds the midpoint values by fitting a Lagrange polynomial through the $2N$ closest data points. By repeating this algorithm again and again, each time doubling the number of data points or nodes by midpoint interpolation, we eventually have a dense set of data points and can determine uniquely a smooth interpolation function. Because interpolatory subdivision schemes relate data points from one scale to another scale, it is not surprising that they are a key ingredient in the construction of compactly supported wavelets [**1, 3**].

More recently, Merrien [**10, 11, 6**] introduced Hermite subdivision schemes. Since Merrien subdivision schemes use Hermite nodes, they have twice the approximation order and better regularity for a given support. For example, $2-$point Hermite schemes are differentiable and can reproduce quadratic or cubic polynomials whereas the corresponding $2-$point Deslauriers-Dubuc scheme (the linear spline) isn't differentiable and only reproduces linear polynomials. Also, local quadratic spline Hermite interpolation can be made possible by adding intermediate nodes[**4**].

Therefore, we propose to add intermediate nodes to Deslauriers-Dubuc schemes and subdivision schemes in general to improve approximation and local properties. Doubling the number of nodes is costly: it doubles the memory requirements. However, since a dyadic subdivision scheme doubles its memory usage at each step, we can choose to use right away this upcoming extra storage space without any cost. In effect, we can simply make use of the memory that will be allocated later in any case. Therefore, we can freely increase the number of nodes in intermediate steps. These new intermediate nodes or placeholders can then be used to record a coarse scale guess (using a tetradic filter) which we can later combine with a finer scale interpolation (using a dyadic filter). These schemes are said to be “high resolution” because we no longer consider only the next finer scale, but actually the next two finer scales; alternatively, we could describe these algorithms as “two-step subdivision schemes”. The main result of this paper is that by summing up the tetradic (coarse) interpolation recorded in placeholders and dyadic (fine) interpolations, we get a range of smooth (C^1) high resolutions schemes reproducing cubic polynomials. It is also shown that whereas $4-$point subdivision scheme can reproduce at most cubic polynomials, $4-$point high resolution subdivision schemes can reproduce polynomials of degree 4 and we show that the high resolution approach leads to many of the same properties (smoothness, interpolation) but with better local properties.

Subdivision schemes

Let $b>1$ be an integer then given two integers k,j, the number $x_{j,k}=k/b^{j}$ is said to be $b-$adic (of depth j). For a fixed j, the $b-$adic numbers form a regularly spaced set of nodes. For a fixed J, given some data $\left\{y_{J,k}\right\}_{k\in\mathbb{Z}}$, we want to build a smooth function f such that $f\left(x_{J,k}\right)=y_{J,k}\,\forall k\in\mathbb{Z}$. Starting with this initial data $(y_{J,k})$ and using the linear formula

$$y_{j+1,l}=\sum_{k\in\mathbb{Z}}\gamma_{bk-l}y_{j,k} \tag{3.0.1}$$

for some constant array γ, we get values $y_{j,k}$ for any $j>J$ and since $b-$adic numbers form a dense set in $\mathbb{R}$, there is at most one continuous function such that $f\left(x_{j,k}\right)=y_{j,k}$ for all $k\in\mathbb{Z}, j>J$.

A subdivision scheme is interpolatory and satisfies $f\left(x_{J,k}\right)=y_{J,k}$ if $\gamma_{bk}=0\,\forall k\in\mathbb{Z}$ except for $\gamma_0=1$. We say that a subdivision scheme is stationary if the array γ is constant (doesn't depend on j). Because γ doesn't depend explicitely on l the scheme is translation invariant or homogeneous. A subdivision scheme is said to be $2N-$point if $\gamma_l=0$ for $|l|\geq Nb$. The interpolation function f computed from an interpolatory $2N-$point $b-$adic scheme with initial data $y_{0,0}=1$ and $y_{0,k}=0$ for all $k\neq 0$ is said to be the fundamental function and has a compact support of $[-(Nb-1)/(b-1),(Nb-1)/(b-1)]$ or $[1-2N,2N-1]$ when $b=2$.

For $N=1,2,3,...$ there are corresponding interpolatory $2N-$point interpolatory Deslauriers-Dubuc subdivision schemes built from the midpoint evaluation of Lagrange polynomial of degree $2N-1$. For $b=2$ (dyadic case), the $4-$point Deslauriers-Dubuc scheme can be defined from the array γ^{DD2} given by $\gamma_0^{DD2}=1,\gamma_1^{DD2}=\gamma_{-1}^{DD2}=-9/16,\gamma_3^{DD2}=\gamma_{-3}^{DD2}=-1/16$ with $\gamma_k^{DD2}=0$ otherwise; for $b=4$ (tetradic case), the scheme is defined from the array γ^{DD4} given by $\gamma_{2k}^{DD4}=\gamma_k^{DD2}\,\forall k\in\mathbb{Z},\ \gamma_{-1}^{DD4}=\gamma_1^{DD4}=105/128,\gamma_{-3}^{DD4}=\gamma_{-3}^{DD4}=35/128,\gamma_{-5}^{DD4}=\gamma_5^{DD4}=-7/128,\gamma_{-7}^{DD4}=\gamma_7^{DD4}=-5/128,$ with $\gamma_k^{DD2}=0$ otherwise.

Because $4-$point Deslauriers-Dubuc schemes are derived from cubic Lagrange polynomials, they reproduce cubic polynomials, that is, if the initial data $y_{j,k}$ satisfies $y_{j,k} = p\left(x_{j,k}\right)\ \forall k \in \mathbb{Z}$ for some cubic polynomial p then the interpolation function f is this same cubic polynomial $f = p$. The two cases presented above (γ^{DD2} and γ^{DD4}) reproduce cubic polynomials and it can also be shown that they both give differentiable (C^1) interpolation functions. Because we later borrow from these two subdivision schemes, we give explicit algorithms for both schemes.

ALGORITHM 3.0.1. *($4-$point Deslauriers-Dubuc Dyadic Scheme) For a given integer j, begin with some initial $y-$values $y_{j,k}\, k \in \mathbb{Z}$ over dyadic numbers $x_{j,k} = k/2^j$,*

1. *recopy data at $x_{j+1,2k} = x_{j,k}$: $y_{j+1,2k} = y_{j,k}\, \forall k \in \mathbb{Z}$;*
2. *interpolate midpoint value by the corresponding cubic Lagrange polynomial:*

$$y_{j+1,2k+1} = \frac{-y_{j,k-1} + 9y_{j,k} + 9y_{j,k+1} - y_{j,k+2}}{128}\, \forall k \in \mathbb{Z};$$

3. *Repeat with $j \to j+1$ and using y_{j+1} as initial data.*

ALGORITHM 3.0.2. *($4-$point Deslauriers-Dubuc Tetradic Subdivision Scheme) For a given integer j, begin with some initial $y-$values $y_{j,k}\, k \in \mathbb{Z}$ over $4-$adic numbers $x_{j,k} = k/4^j$*

1. *recopy data at $x_{j+1,4k} = x_{j,k}$: $y_{j+1,4k} = y_{j,k}\, \forall k \in \mathbb{Z}$;*
2. *interpolate quartertile point values by the corresponding cubic Lagrange polynomial:*

$$y_{j+1,4k+1} = \frac{-7y_{j,k-1} + 105y_{j,k} + 35y_{j,k+1} - 5y_{j,k+2}}{128};$$

$$y_{j+1,4k+2} = \frac{-y_{j,k-1} + 9y_{j,k} + 9y_{j,k+1} - y_{j,k+2}}{128};$$

$$y_{j+1,4k+3} = \frac{-5y_{j,k-1} + 35y_{j,k} + 105y_{j,k+1} - 7y_{j,k+2}}{128}\, \forall k \in \mathbb{Z};$$

3. *Repeat with $j \to j+1$ and using y_{j+1} as initial data.*

High resolution subdivision schemes

3.1. Definitions

In this paper, we want to show how the subdivision scheme framework can be extended by hybrid schemes: mixing tetradic and dyadic subdivision schemes for example. Given some data $\{y_{j-1,k}\}_{k\in\mathbb{Z}}$ to interpolate on the dyadic $x_{j-1,k}$ grid, we first apply a dyadic subdivision scheme as an initialisation step: since high resolution subdivision schemes can be seen as multistep subdivision schemes, it is not surprising that they require an initialisation step. For $4-$point high resolution subdivision scheme, a sensible choice for the initialization step is the $4-$point Deslauriers-Dubuc dyadic scheme (see lemma 3.3.1) which will copy the data at even nodes ($y_{j,2k} = y_{j-1,k}$) and insert new values at odd nodes ($y_{j,2k+1}$ for $k \in \mathbb{Z}$). A value $y_{j,k}$ is said to be "stable" if $y_{j,k} = y_{j+1,2k}$, other nodes are said to be temporary or are referred to as "placeholders". A high resolution scheme on a dyadic grid is said to be interpolatory if all $y_{j,2k}$ values on even nodes ($x_{j,2k}$) are stable. Assuming we used an interpolatory subdivision scheme as an initialization step, the following algorithm is interpolatory.

ALGORITHM 3.1.1. *($4-$point Dyadic High Resolution Subdivision Scheme) The following iteration steps depend on α, a constant parameter. For a given integer j, begin with some initial $y-$values $y_{j,k}\, k \in \mathbb{Z}$ over dyadic numbers $x_{j,k} = k/2^j$ where only the even nodes will be interpolated ($y_{j,2k}$),*

1. *recopy stable data: $y_{j+1,4k} = y_{j,2k}\, \forall k \in \mathbb{Z}$;*
2. *Apply the $4-$point Deslauriers-Dubuc tetradic scheme on even (stable) nodes:*

$$y_{j+1,4k+1} = \frac{-7y_{j,2k-2} + 105y_{j,2k} + 35y_{j,2k+2} - 5y_{j,2k+4}}{128};$$

$$y^{temporary}_{j+1,4k+2} = \frac{-y_{j,2k-2} + 9y_{j,2k} + 9y_{j,2k+2} - y_{j,2k+4}}{128};$$

$$y_{j+1,4k+3} = \frac{-5y_{j,2k-2} + 35y_{j,2k} + 105y_{j,2k+2} - 7y_{j,2k+4}}{128} \; \forall k \in \mathbb{Z};$$

3. *Update midpoint (which then becomes stable):*

$$y_{j+1,4k+2} = (1-\alpha) y^{temporary}_{j+1,4k+2} + \alpha y_{j,2k+1};$$

4. *Repeat with $j \to j+1$ and using y_{j+1} as initial data.*

This new algorithm is not a subdivision scheme in general and thus we need to propose a more general definition: stationary subdivision schemes (equation 3.0.1) can be generalized by the linear equation

$$y_{j+1,l} = \sum_{m=1}^{M} \sum_{k \in \mathbb{Z}} \gamma^{(m)}_{Mbk+m-1-l} y_{j,Mk+m-1} \tag{3.1.1}$$

where $\gamma^{(1)}, \ldots, \gamma^{(M)}$ are constant arrays (independent from j). It can be said to be $b-$adic because the number of nodes is increasing by a factor of b with each iteration but because we have $M > 1$ arrays γ, the scheme is said to be a high resolution subdivision scheme. We say it is interpolatory if it satisfies $y_{j+1,Mbk} = y_{j+1,Mk}$ and it is $2N-$point if $\gamma^{(m)}_l = 0$ for $|l| \geq MNb$ and $m = 1, \ldots, M$. For $b = M = 2$ the general equation 3.1.1 becomes

$$y_{j+1,l} = \sum_{k \in \mathbb{Z}} \gamma^{(1)}_{4k-l} y_{j,2k} + \gamma^{(2)}_{4k+1-l} y_{j,2k+1}. \tag{3.1.2}$$

It is interpolatory if $y_{j+1,4k} = y_{j+1,2k}$ and $2N-$point if $\gamma^{(m)}_l = 0$ for $|l| \geq 4N$ and $m = 1, 2$. It should be noted that for an algorithm based on high resolution subdivision scheme to be interpolatory, the initialization step must be interpolatory ($y_{j,bk} = y_{j-1,k}$).

The interpolatory algorithm 3.1.1 amounts to choosing $\gamma^{(1)}$ and $\gamma^{(2)}$ to be:

$$\gamma^{(1)}_{2k} = \gamma^{DD4}_{2k} + \alpha\left(\delta_{k,0} - \gamma^{DD2}_{k}\right), \quad \gamma^{(1)}_{2k+1} = \gamma^{DD4}_{2k+1} \; \forall k \in \mathbb{Z} \tag{3.1.3}$$

$$\gamma^{(2)}_{-1} = \alpha, \qquad \gamma^{(2)}_{k} = 0 \, otherwise \tag{3.1.4}$$

for some parameter $\alpha \in \mathbb{R}$. Indeed, since $\gamma_{2k}^{DD4} = \gamma_k^{DD2}\, \forall k \in \mathbb{Z}$, we can rewrite equation 3.1.2 for even and odd terms. Firstly, setting $l = 2s$ (l even), we have

$$\begin{aligned} y_{j+1,2s} &= \textstyle\sum_{k\in\mathbb{Z}} \quad \gamma^{(1)}_{4k-2s} y_{j,2k} + \gamma^{(2)}_{4k+1-2s} y_{j,2k+1} \\ &= \textstyle\sum_{k\in\mathbb{Z}} \quad \left(\gamma^{DD4}_{4k-2s} - \alpha\gamma^{DD2}_{2k-s} + \alpha\delta_{4k,2s}\right) y_{j,2k} + \alpha\delta_{4k+2,s} y_{j,2k+1} \\ &= \textstyle\sum_{k\in\mathbb{Z}} \quad \left((1-\alpha)\gamma^{DD2}_{2k-s} + \alpha\delta_{2k,s}\right) y_{j,2k} + \delta_{2k+1,s}\alpha y_{j,2k+1} \end{aligned}$$

so that when s is even ($l = 2s = 4r$), we have the interpolatory condition

$$y_{j+1,4r} = y_{j,2r} \tag{3.1.5}$$

otherwise, when s is odd $(l = 2s = 4r+2)$

$$y_{j+1,4r+2} = \alpha y_{j,2r+1} + (1-\alpha) \sum_{k\in\mathbb{Z}} \gamma^{DD2}_{2k-2r-1} y_{j,2k}. \tag{3.1.6}$$

Secondly, if l is odd ($l = 2s+1$), we have

$$\begin{aligned} y_{j+1,2s+1} &= \sum_{k\in\mathbb{Z}} \gamma^{(1)}_{4k-2s-1} y_{j,2k} + \gamma^{(2)}_{4k-2s-1} y_{j,2k+1} \\ &= \sum_{k\in\mathbb{Z}} \gamma^{DD4}_{4k-2s-1} y_{j,2k}. \end{aligned} \tag{3.1.7}$$

Equations 3.1.5, 3.1.6, and 3.1.7 can be used to describe the chosen high resolution schemes: while equation 3.1.5 is the interpolatory condition, equation 3.1.7 fills the placeholders with tetradic (coarse scale) interpolated values whereas equation 3.1.6 combines the value stored in the placeholder with the newly available interpolated value (fine scale) given by the summation term which we recognize from the dyadic Deslauriers-Dubuc interpolation.

In the simplest case, $\alpha = 0$, equation 3.1.2 becomes

$$y_{j+1,l} = \sum_{k\in\mathbb{Z}} \gamma^{DD4}_{4k-l} y_{j,2k}. \tag{3.1.8}$$

Because $\alpha = 0 \Rightarrow \gamma^{(2)} = 0$, we see that the placeholders (odd nodes) are effectively ignored. Indeed, we observe that this last equation discards odd nodes at each step: $y_{j+1,l}$ depends only on even nodes ($y_{j,2k}$) and not at all on the odd nodes ($y_{j,2k+1}$). Hence, we can replace equation 3.1.8 by

$$y_{j+1,2l} = \sum_{k\in\mathbb{Z}} \gamma^{DD4}_{4k-2l} y_{j,2k}$$

but because $\gamma^{DD4}_{2k} = \gamma^{DD2}_{k}$, this last equation becomes $y_{j+1,2l} = \gamma^{DD2}_{2k-l} y_{j,2k}$ and if we define $\widetilde{y}_{j,k} = y_{j,2k}$ then

$$\widetilde{y}_{j+1,2l} = \sum_{k\in\mathbb{Z}} \gamma^{DD2}_{2k-l} \widetilde{y}_{j,2k} \tag{3.1.9}$$

which we recognize as the cubic Deslauriers-Dubuc scheme and we have proved the next proposition.

PROPOSITION 3.1.2. *For $\alpha = 0$, the high resolution scheme given by algorithm 3.1.1 (or equations 3.1.2, 3.1.3, and 3.1.4) is equivalent to the $4-$point dyadic Deslauriers-Dubuc subdivision scheme.*

3.2. Local properties

$2N-$point high resolution subdivision schemes are "as local as" the corresponding $2N-$point subdivision scheme: it is not surprising since interpolated values depend only on the closest $2N$ nodes and we use a dyadic tree. As an example, consider any $2N-$point dyadic high resolution scheme. Since the algorithm is linear, stationary and homogeneous, we can choose $y_{0,0} = 1$ and $y_{0,k} = 0$ for $k \neq 0$ to characterize locality. As an initialization step, apply any interpolatory $2N-$point dyadic subdivision scheme: the farthest non-zero interpolated values will be $y_{1,\pm(2N-1)}$ at $\pm\xi_0$ where $\xi_0 = \frac{2N-1}{2}$. We then apply the $4-$point high resolution subdivisions scheme itself treating the odd nodes $y_{1,-(2N-1)},\ldots,y_{1,2N-1}$ as placeholders, the farthest non-zero interpolated values will be at $\pm\xi_1$ where $\xi_1 = \xi_0 + \frac{2N-1}{2^2}$: in this respect, a high resolution subdivision scheme behaves just like subdivision schemes. Of course, these farthest values are not stable yet since they are located at odd nodes, but they will become stable on the next interation. By induction, the farthest non-zero interpolated values after n interations is at $\xi_n = \sum_{k=1}^{n} \frac{2N-1}{2^k}$ which converges to $2N-1$ as $n \to \infty$ and hence, the support of the interpolated function has to be in $[1-2N, 2N-1]$. A similar argument could be made when $b \neq 2$ (non-dyadic case).

3.3. Reproduced polynomials

Assume that for some j, $y_{j,k}=p_3\left(x_{j,k}\right)\ \forall k\in\mathbb{Z}$ for some cubic polynomial p_3. Because $4-$point Deslauriers-Dubuc schemes reproduce cubic polynomials, we have

$$\sum_{k\in\mathbb{Z}}\gamma_{2k-2r-1}^{DD2}y_{j,2k}=y_{j,2r+1}=p_3\left(x_{j,2r+1}\right)$$

and thus equation 3.1.6 becomes $y_{j+1,4r+2}=p_3\left(x_{j,2r+1}\right)$ for any α. Similarly, equation 3.1.7 implies $y_{j+1,2s+1}=p_3\left(x_{j+1,2s+1}\right)$. We can conclude that $y_{j+1,k}=p_3\left(x_{j+1,k}\right)\ \forall k\in\mathbb{Z}$ if $y_{j,k}=p_3\left(x_{j,k}\right)\ \forall k\in\mathbb{Z}$ and thus high resolution schemes defined by equation 3.1.2 reproduce cubic polynomials. As we have seen, for practical implementations of a high subdivision scheme, it is necessary to first apply a one-step subdivision scheme. This can be solved by a (one-step) dyadic Deslauriers-Dubuc interpolation. Let $\left\{y_{j,k}\right\}_k$ be some initial data. As a first step, we apply equation

$$y_{j+1,l}=\sum_{k\in\mathbb{Z}}\gamma_{2k-l}^{DD2}y_{j,2k} \tag{3.3.1}$$

followed by algorithm 3.1.1 with $j+1$. By induction on j using algorithm 3.1.1, we get the following lemma.

LEMMA 3.3.1. *High resolution schemes given by algorithm 3.1.1 (or equations 3.1.2, 3.1.3, and 3.1.4) using a one step interpolatory 4-point dyadic Deslauriers-Dubuc interpolation (equation 3.3.1) as an initialization step reproduce cubic polynomials and are interpolatory.*

We can also get a stronger result by choosing a specific α. We can write any polynomial of degree 4, $p_4$as $p_4(x)=a_4x^4+p_3(x)$ where p_3 is some cubic polynomial. Suppose that for some j, $y_{j,2k}=p_4\left(x_{j,2k}\right)$ and $y_{j-1,k}=p_4\left(x_{j-1,k}\right)\ \forall k\in\mathbb{Z}$. We can write $y_{j+1,4r+2}$ for any $r\in\mathbb{Z}$ in terms of this initial data (y_j and y_{j-1}) by substituting equation 3.1.7 into 3.1.6 to get

$$\begin{aligned} y_{j+1,4r+2} &= \alpha y_{j,2r+1}+(1-\alpha)\sum_{k\in\mathbb{Z}}\gamma_{2k-2r-1}^{DD2}y_{j,2k} \\ &= \alpha\sum_{k\in\mathbb{Z}}\gamma_{4k-2r-1}^{DD4}y_{j-1,2k}+(1-\alpha)\sum_{k\in\mathbb{Z}}\gamma_{2k-2r-1}^{DD2}y_{j,2k}. \end{aligned} \tag{3.3.2}$$

We want to show that $y_{j+1,4r+2} = p_4\left(x_{j,2r+1}\right)$ for some α and so we substitute $y_{j,2k} = p_4\left(x_{j,2k}\right)$ and $y_{j-1,k} = p_4\left(x_{j-1,k}\right)$ into the two sums of this last equation. Because of the identities

$$\begin{aligned}
\frac{-9}{2^{4j}} &= \frac{-\left(x_{j,2r-2}\right)^4 + 9\left(x_{j,2r}\right)^4 + 9\left(x_{j,2r+2}\right)^4 - \left(x_{j,2r+4}\right)^4}{128} - \left(x_{j,2r+1}\right)^4 \\
\frac{-105}{2^{4j}} &= \frac{-7\left(x_{j,2r-4}\right)^4 + 105\left(x_{j,2r}\right)^4 + 35\left(x_{j,2r+4}\right)^4 - 5\left(x_{j,2r+8}\right)^4}{128} - \left(x_{j,2r+1}\right)^4 \\
&= \frac{-5\left(x_{j,2r-4}\right)^4 + 35\left(x_{j,2r}\right)^4 + 105\left(x_{j,2r+4}\right)^4 - 7\left(x_{j,2r+8}\right)^4}{128} - \left(x_{j,2r+1}\right)^4,
\end{aligned}$$

we can compute both sums in equation 3.3.2 explicitely:

$$\sum_{k\in\mathbb{Z}} \gamma^{DD4}_{4k-2r-1} y_{j-1,2k} = p_3\left(x_{j,2r+1}\right) + a_4 \sum_{k\in\mathbb{Z}} \gamma^{DD4}_{4k-2r-1} \times \left(x_{j-1,2k} = x_{j,4k}\right)^4 \tag{3.3.3}$$

$$= p_4\left(x_{j,2r+1}\right) - \frac{105 a_4}{2^{4j}} \tag{3.3.4}$$

and

$$\begin{aligned}
\sum_{k\in\mathbb{Z}} \gamma^{DD2}_{2k-2r-1} y_{j,2k} &= p_3\left(x_{j,2r+1}\right) + a_4 \sum_{k\in\mathbb{Z}} \gamma^{DD2}_{2k-2r-1} \times \left(x_{j,2k}\right)^4 \\
&= p_4\left(x_{j,2r+1}\right) - \frac{9 a_4}{2^{4j}}
\end{aligned}$$

Hence, setting $\alpha = -3/32$ in equation 3.3.2, we get

$$y_{j+1,4r+2} = p_4\left(x_{j,2r+1}\right) - \frac{105\alpha + 9(1-\alpha)}{2^{4j}} a_4 = p_4\left(x_{j+1,4r+2}\right)$$

since for $\alpha = -3/32$, $105\alpha + 9(1-\alpha) = 0$. Therefore, the scheme reproduces polynomials of degree 4 when $\alpha = -3/32$.

This last result assumes that we initialize the data so that $y_{j,2k+1} = p_4\left(x_{j,2k+1}\right) - \frac{105 a_4}{16\times 2^{4j}}$ and $y_{j,2k} = p_4\left(x_{j,2k}\right)$ for all $k \in \mathbb{Z}$. Unfortunately, there is no interpolatory 4−point dyadic subdivision scheme able to provide this initialization but we can get this result by using the initial data $y_{j-1,k} = p_4\left(x_{j,k}\right)$ and applying first the high resolution scheme (equation 3.1.2) with $\alpha = 1$ as an initialization step, since equations 3.1.5 and 3.1.6 will guarantee $y_{j,2k} = p_4\left(x_{j,k}\right)$ whereas equation 3.1.7 will initialize the placeholders properly. The case $\alpha = 1$ essentially relies only on the tetradic interpolation and discard the finer scale guesses (dyadic). It is not interpolatory however since we only have $y_{j,4k} = y_{j-1,2k}$ and not the stronger condition $y_{j,2k} = y_{j-1,k}$. For pratical applications, we may wish to initialize high resolution subdivision schemes with an interpolatory subdivision scheme so that the whole process remains interpolatory. While there are no $4-$point subdivision scheme capable of interpolating $y_{j-1,k} = p_4\left(x_{j,k}\right)$ into $y_{j,2k+1} = p_4\left(x_{j,2k+1}\right) - \frac{105a_4}{16\times 2^{4j}}$ and $y_{j,2k} = p_4\left(x_{j,2k}\right)$ for all $k \in \mathbb{Z}$, there exist $5-$point subdivision schemes such as the subdivision scheme described by the next algorithm.

ALGORITHM 3.3.2. *($5-$point Dyadic Subdivision Scheme) For a given integer j, begin with some initial $y-$values $y_{j,k}\, k \in \mathbb{Z}$ over dyadic numbers $x_{j,k} = k/2^j$,*

1. *recopy data at $x_{j+1,2k} = x_{j,k}$: $y_{j+1,2k} = y_{j,k}\, \forall k \in \mathbb{Z}$;*
2. *extrapolate $y_{j,k+4}$ using $y_{j,k-2}, y_{j,k-1}, y_{j,k}, y_{j,k+1}, y_{j,k+2}$ by the formula*

$$\gamma_{j,k} = 5y_{j,k-2} - 24y_{j,k-1} + 45y_{j,k} - 40y_{j,k+1} + 15y_{j,k+1}\, \forall k \in \mathbb{Z}; \tag{3.3.5}$$

3. *interpolate midpoint value using the tetradic Deslauriers-Dubuc formula:*

$$\frac{-7y_{j,k-2} + 105y_{j,k} + 35y_{j,k+2} - 5\gamma_{j,k}}{128}\, \forall k \in \mathbb{Z}.$$

To see that algorithm 3.3.2 properly initialize the placeholders, observe that if we assume that $y_{J,k} = p_4\left(x_{J,k}\right)$, then we only need to check that $y_{J+1,2k+1} = p_4\left(x_{J+1,2k+1}\right) - \frac{105a_4}{16\times 2^{4(J+1)}}$. However, if $y_{J,k} = p_4\left(x_{J,k}\right)$ is satisfied, then $\gamma_{J,k} = p_4\left(x_{J,k+4}\right)$ since formula 3.3.5 can be derived by finding the polynomial of degree 4 $p_{J,k}$ satisfying $p_{J,k}\left(x_{j,l}\right) = y_{J,l}$ for $l = k-2, \ldots, k+2$ and setting $\gamma_{J,k} = p_{J,k}\left(x_{J,k+4}\right)$. Hence, by formula 3.3.4, we have the following lemma.

LEMMA 3.3.3. *Algorithm 3.3.2 describes a $5-$point dyadic subdivisions such that when applied on $y_{j-1,k} = p_4\left(x_{j-1,k}\right)$ where p_4 is a polynomial of degree 4 gives $y_{j,2k+1} = p_4\left(x_{j,2k+1}\right) - \frac{105a_4}{16\times 2^{4j}}$ and $y_{j,2k} = p_4\left(x_{j,2k}\right)$ for all $k \in \mathbb{Z}$.*

Because we have a proper initialization scheme, we can now efficiently reproduce polynomials of degree 4.

PROPOSITION 3.3.4. *For any given j, if $y_{j-1,k} = p_4\left(x_{j-1,k}\right)$ where p_4 is a polynomial of degree 4 then applying algorithm 3.3.2 following algorithm 3.1.1 with $\alpha = -3/32$ for the following steps will guarantee that $y_{j',2k} = p_4\left(x_{j',2k}\right)$ for $\forall k \in \mathbb{Z}$ and all $j' \geq j-1$. In other words, this $4-$point high resolution subdivision algorithm reproduces polynomials of degree 4.*

This result is significant because it is not possible for $4-$point subdivision schemes to reproduce polynomials of degree 4. Even if we include non-interpolatory subdivision schemes, for a given $k \in \mathbb{Z}$, $y_{j+1,2k+1}$ cannot be computed solely from the neighbouring values $y_{j,k-1}$, $y_{j,k}$,$y_{j,k+1}$, and $y_{j,k+1}$ while reproducing polynomials of degree 4.

PROPOSITION 3.3.5. *A $4-$point dyadic subdivision scheme cannot reproduce polynomials of degree 4.*

PROOF. Let $P_4(x) = x(x-1)(x-2)(x-3)$ and consider $y_{0,k} = P_4(k)$. All $4-$point subdivision schemes will interpolate $y_{1,3} = 0 \neq P_4\left(\frac{3}{2}\right) = \frac{9}{16}$. □

By the proof of proposition 3.3.5, we see that only subdivision schemes using 5 points can interpolate polynomials of degree 4. Starting with $y_{0,k} = \delta_{k,0} \forall k \in \mathbb{Z}$, it can be seen that in the best possible case, a $5-$point subdivision scheme will give an interpolation function having a support of size 8. For example, consider schemes of the form $y_{j+1,l} = \sum_{k=-2}^{2} \tau_{2k-l} y_{j,k}$ with $\tau_{-5} = \frac{3}{128}$,$\tau_{-3} = \frac{-5}{32}$,$\tau_{-1} = \frac{45}{64}$,$\tau_1 = \frac{15}{32}$,$\tau_3 = \frac{-5}{128}$ and $\tau_0 = 1$, $\tau_k = 0$ otherwise which has support $[-3,5]$. On the other hand, applying the high resolution scheme described by proposition 3.3.4 with the same initial data ($y_{0,k} = \delta_{k,0} \forall k \in \mathbb{Z}$) leads to an interpolation function having a compact support of size 7 taking into account the $5-$point initialization scheme. Therefore, we have a new interpolation scheme with many of the same properties but with better local properties.

3.4. Sufficient conditions for regularity

To study the regularity of high resolution schemes, it is convenient to rewrite formula 3.1.2 in terms of (trigonometric) polynomials. Given some data $y_{j,k}$, define $P^j(z) = \sum_{k\in\mathbb{Z}} y_{j,k} z^k$. If $P_2(z) = \sum_{k\in\mathbb{Z}} \gamma_k^{DD2} z^k$, then the equation of the $4-$point dyadic Deslauriers-Dubuc scheme (equation 3.3.1), can be rewritten $P^{j+1}(z) = P_2(z) P^j(z^2)$. Similarly, if $P_4(z) = \sum_{k\in\mathbb{Z}} \gamma_k^{DD4} z^k$, then the tetradic subdivision scheme is given by $P^{j+1}(z) = P_4(z)P^j(z^2)$. It can be shown that we can rewrite the general equation for high resolution subdivision schemes as

$$P^{j+1}(z) = \sum_{i=1}^{M} \Phi_i(z) P^j\left(e^{2\pi i/b} z^b\right).$$

where Γ_i must be Laurent polynomials and similarly for dyadic schemes ($b = 2$),

$$P^{j+1}(z) = \Phi_1(z) P^j\left(z^2\right) + \Phi_2(z) P^j\left(-z^2\right).$$

The equation of symbols for the 4−point cubic high resolution sheme is (see equation 3.1.2 and algorithm 3.1.1)

$$\begin{aligned} P^{j+1}(z) &= \left\{P_4(z) - \alpha P_2\left(z^2\right) + \alpha\right\}\left(\frac{P^j\left(z^2\right) + P^j\left(-z^2\right)}{2}\right) + \alpha\left(\frac{P^j\left(z^2\right) - P^j\left(-z^2\right)}{2}\right) \\ &= \left\{\frac{P_4(z) - \alpha P_2\left(z^2\right)}{2} + \alpha\right\} P^j\left(z^2\right) + \frac{P_4(z) - \alpha P_2\left(z^2\right)}{2} P^j\left(-z^2\right) \\ &= \Gamma_1(z) P^j\left(z^2\right) + \Gamma_2(z) P^j\left(-z^2\right). \end{aligned} \tag{3.4.1}$$

Because $\gamma_{2k}^{DD4} = \gamma_k^{DD2}\, \forall k \in \mathbb{Z}$, we observe that P_2 is not needed and everything can be written in terms of P_4, indeed,

$$P_4(z) - \alpha P_2\left(z^2\right) = \frac{P_4(z) - P_4(-z)}{2} + (1-\alpha)\frac{P_4(z) + P_4(-z)}{2}$$

and thus, the symbols $\Gamma_1$and Γ_2 can be written

$$\begin{aligned} \Gamma_1(z) &= \Gamma_2(z) + \alpha \\ \Gamma_2(z) &= \frac{P_4(z) - P_4(-z)}{4} + (1-\alpha)\frac{P_4(z) + P_4(-z)}{4}. \end{aligned}$$

When $\alpha = 0$ (Deslauriers-Dubuc case), $\Gamma_1(z) = \Gamma_2(z) = \frac{P_4(z)}{2}$ and equation 3.1.2 becomes

$$P^{j+1}(z) = P_4(z)\left(\frac{P^j\left(z^2\right) + P^j\left(-z^2\right)}{2}\right)$$

and it can be shown to be equivalent to the Deslauriers-Dubuc dyadic scheme by using the last equation for averaging $P^{j+1}(z)$ and $P^{j+1}(-z)$,

$$\begin{aligned}\frac{P^{j+1}(z)+P^{j+1}(-z)}{2} &= \left(\frac{P_4(z)-P_4(-z)}{2}\right)\left(\frac{P^j\left(z^2\right)+P^j\left(-z^2\right)}{2}\right)\\ &= P_2\left(z^2\right)\left(\frac{P^j\left(z^2\right)+P^j\left(-z^2\right)}{2}\right).\end{aligned}$$

which can be used to prove that when $\alpha = 0$ the high resolution subdivision scheme becomes the dyadic Deslauriers-Dubuc scheme (proposition 3.1.2).

In order to study the regularity and stability of the chosen high resolution schemes, we need to find corresponding schemes for the (forward) finite differences. Let $dx_j = 1/2^j$ and write

$$D_{j,k} = \frac{dy_{j,k}}{dx_j} = 2^j\left(y_{j,k+1} - y_{j,k}\right),$$

and define higher order finite differences recursively

$$D^n_{j,k} = d^{(n)}y_{j,k}/\left(dx_j\right)^n = d\left(d^{(n-1)}y_{j,k}\right)/\left(dx_j\right)^n = 2^{jn}\times d^{(n)}y_{j,k}.$$

Note that $D_{j,k} = D^1_{j,k}$. Let H^j_1 be the symbol for $dy_{j,k}/dx_j$, then

$$\begin{aligned}H^j_1(z) &= \sum_{k\in\mathbb{Z}} 2^j\left(y_{j,k+1}-y_{j,k}\right)z^k\\ &= \sum_{k\in\mathbb{Z}} 2^j y_{j,k}z^{k-1} - \sum_{k\in\mathbb{Z}} 2^j y_{j,k}z^k\\ &= 2^j(1/z-1)P^j(z) = 2^j(1-z)P^j(z)/z,\end{aligned}$$

and thus $P^j\left(z^2\right) = z^2 2^j H^j_1\left(z^2\right)/(1-z^2)$, $P^j\left(-z^2\right) = -z^2 2^j H^j_1\left(-z^2\right)/(1+z^2)$, and $P^{j+1}(z) = z2^{j+1}H^{j+1}_1(z)/(1-z)$. Substituting these three equations into $P^{j+1}(z) = \Gamma_1(z)P^j\left(z^2\right)+\Gamma_2(z)P^j\left(-z^2\right)$ (equation 3.4.1) gives

$$H^{j+1}_1(z) = \frac{2z(1-z)}{(1-z^2)}\Gamma_1(z)H^j_1\left(z^2\right) - \frac{2z(1-z)}{(1+z^2)}\Gamma_2(z)H^j_1\left(-z^2\right). \tag{3.4.2}$$

Similarly, the higher order finite differences are given by

$$H_n^j(z) = \frac{2(1-z)}{z} H_{n-1}^j(z) = \left(\frac{2(1-z)}{z}\right)^n P^j(z)$$

where $H_0(z) = P(z)$ and it can be seen that they can be computed by (see derivation of equation 3.4.2 above)

$$H_n^{j+1}(z) = \left(\frac{2z}{1+z}\right)^n \Gamma_1(z) H_n^j\left(z^2\right) + \left(\frac{-2z(1-z)}{1+z^2}\right)^n \Gamma_2(z) H_n^j\left(-z^2\right). \tag{3.4.3}$$

H_n is said to be the symbol of a high resolution subdivision scheme if $\Gamma_1(z)/(1+z)$ and $\Gamma_2(z)/(1+z^2)$ are Laurent polynomials. $\Gamma_1(z)/(1+z)^n$ and $\Gamma_2(z)/(1+z^2)^n$ are Laurent polynomial for n $= 1,2,3,4$ because

$$P_4(z) = \frac{-(1+z)^4\left(1+z^2\right)^4\left(5z^2-12z+5\right)}{128z^7}.$$

Therefore, H_n is the symbol of a high resolution subdivision scheme if $n = 1,2,3,4$.

LEMMA 3.4.1. *For high resolution subdivision schemes given by algorithm 3.1.1 (or equations 3.1.2, 3.1.3, and 3.1.4), the finite differences $d^{(n)}y_{j,k}$ can be computed by a corresponding high resolution subdivision scheme for $n = 1,2,3,4$.*

We can define dH_n^j as the symbol of

$$dD_{j,k}^{n-1} = d\left(\frac{d^{(n-1)}y_{j,k}}{\left(dx_j\right)^{n-1}}\right) = \frac{d^n y_{j,k}}{\left(dx_j\right)^{n-1}} = \frac{D_{j,k}^n}{2^j}$$

or $dH_n^j(z) = H_{n+1}^j(z)/2^j$ and thus

$$dH_{n-1}^j(z) = \frac{(1-z)}{z} H_{n-1}^j(z) = \frac{2^{j(n-1)}(1-z)^n}{z^n} P^j(z). \tag{3.4.4}$$

Replacing H_{n-1} by dH_{n-1} in equation 3.4.3, we find

$$dH_{n-1}^{j+1}(z) = \frac{1}{2}\left\{\left(\frac{2z}{1+z}\right)^n \Gamma_1(z) dH_{n-1}^j\left(z^2\right) + \left(\frac{-2z(1-z)}{1+z^2}\right)^n \Gamma_2(z) dH_{n-1}^j\left(-z^2\right)\right\}. \tag{3.4.5}$$

And because $dH_n^j(z) = H_{n+1}^j(z)/2^j$, dH_{n-1} is the symbol of a high resolution subdivision scheme for $n = 1,2,3,4$.

Using results from Dyn [**7**], we have the following theorem.

THEOREM 3.4.2. *(Dyn) If dH_n as in equations3.4.4 and 3.4.5 is the symbol of a high resolution subdivision scheme converging uniformly to zero for all bounded initial data, then the corresponding scheme P as in equation 3.4.1 is C^n, that is, all interpolation functions f are C^n*

PROOF. See the proof of theorem 3.4 [**7**] as it applies to high resolution subdivision schemes. □

In general, given $y_{j+1,l} = \sum_{k\in\mathbb{Z}} \gamma_{2k-l} y_{j,k}$, a sufficient condition for $y_{j,k} \to 0$ uniformly as $j \to \infty$ is that $\lambda = \max_{l=0,1} \left\{\sum_{k\in\mathbb{Z}} |\gamma_{2k-l}|\right\} < 1$, indeed, if $M_j = \sup\left\{\left|y_{j,k}\right| : k \in \mathbb{Z}\right\}$ then $M_{j+1} \leq \max_{l=0,1} \left\{\sum_{k\in\mathbb{Z}} |\gamma_{2k-l}|\right\} M_j$ because $y_{j+1,2l} = \sum_{k\in\mathbb{Z}} \gamma_{2k-2l} y_{j,k}$ and $y_{j+1,2l+1} = \sum_{k\in\mathbb{Z}} \gamma_{2k-2l-1} y_{j,k}$. For a high resolution subdivision scheme given by $y_{j+1,l} = \sum_{k\in\mathbb{Z}} \gamma^{(1)}_{4k-l} y_{j,2k} + \gamma^{(2)}_{4k+1-l} y_{j,2k+1}$, we proceed in the same manner. Firstly $y_{j+1,2l} = \sum_{k\in\mathbb{Z}} \gamma^{(1)}_{4k-2l} y_{j,2k} + \gamma^{(2)}_{4k+1-2l} y_{j,2k+1}$ and secondly $y_{j+1,2l+1} = \sum_{k\in\mathbb{Z}} \gamma^{(1)}_{4k-2l-1} y_{j,2k} + \gamma^{(2)}_{4k-2l} y_{j,2k+1}$. Thus if $\lambda_{HR} = \max_{l=0,1} \left\{\sum_{k\in\mathbb{Z}} \left|\gamma^{(1)}_{2k-l}\right| + \left|\gamma^{(2)}_{2k+1-l}\right|\right\}$ then $M_{j+1} \leq \lambda M_j$ Given a symbol $Q(z) = \sum_k q_k z^k$, define $\|Q(z)\|_{sup} = \sup_k \{|q_k|\}$ and $\|Q(z)\|_{\Sigma} = \max\{\sum_k |q_k|\}$. For high resolution subdivision schemes, starting with $P^{j+1}(z) = \Phi_1(z) P^j\left(z^2\right) + \Phi_2(z) P^j\left(-z^2\right)$,we see that λ_{HR} is given by

$$\lambda_{HR} = \max\{\lambda_1, \lambda_2\} \tag{3.4.6}$$

$$= \max\left\{\left\|\frac{\Phi_1(z)+\Phi_1(-z)+\Phi_2(z)-\Phi_2(-z)}{2}\right\|_{\Sigma}, \left\|\frac{\Phi_1(z)-\Phi_1(-z)+\Phi_2(z)+\Phi_2(-z)}{2}\right\|_{\Sigma}\right\} \tag{3.4.7}$$

and $\left\|P^{j+1}(z)\right\|_{sup} \leq \lambda_{HR} \left\|P^j(z)\right\|_{sup}$.

LEMMA 3.4.3. *A high resolution subdivision scheme given by the symbol equation $P^{j+1}(z) = \Phi_1(z) P^j\left(z^2\right) + \Phi_2(z) P^j\left(-z^2\right)$ converges uniformly to zero for all bounded initial values if $\lambda_{HR} < 1$ where λ_{HR} is as in equation 3.4.7.*

We are now ready to prove the next theorem.

THEOREM 3.4.4. *For $-25/56 < \alpha < 15/32$, the high resolution subdivision scheme given by equation3.4.1 are C^1.*

PROOF. The symbol of the high resolution subdivision scheme $dD_{j,k} = dD^1_{j,k}$, dH_1 is given by (see equation 3.4.5)

$$dH_1^{j+1}(z) = 2\left(\frac{z}{1+z}\right)^2 \Gamma_1(z) dH_1^j\left(z^2\right) + 2\left(\frac{-z(1-z)}{1+z^2}\right)^2 \Gamma_2(z) dH_1^j\left(-z^2\right)$$

By theorem 3.4.2, it is enough to show that $dD_{j,k}$ converges uniformly to zero for all bounded initial data. However, using lemma 3.4.3, we know that it is sufficient to prove that $\lambda_{HR} < 1$ with $\Phi_1(z) = 2z^2\Gamma_1(z)/(1+z)^2$ and $\Phi_2(z) = 2z^2(1-z)^2\Gamma_2(z)/\left(1+z^2\right)^2$. We get

$$\begin{aligned} \lambda_1 &= \frac{5+2|4\alpha+1|+2|7-8\alpha|+2|5+12\alpha|+|32\alpha+5|+|5-8\alpha|+|24\alpha-7|}{64} \\ \lambda_2 &= \frac{5+2|4\alpha+1|+2|3+8\alpha|+2|1-4\alpha|+|21-32\alpha|+|1+8\alpha|+|24\alpha+11|}{64}. \end{aligned}$$

For $-25/56 < \alpha < 15/32$, we have $\lambda_1 < 1$, whereas for $-7/12 < \alpha < 5/8$, $\lambda_2 < 1$ Hence, we have that $\lambda_{HR} = \max\{\lambda_1, \lambda_2\} < 1$ for $-25/56 < \alpha < 15/32$ or $-\sim 0.45 < \alpha <\sim 0.47$ (see Fig. 3.4.2). □

Theorem 3.4.4 is illustrated by Fig. 3.4.1 where the derivative of three interpolants are given for $\alpha = -0.2, 0, 0.15$. These three examples show that there are many interpolatory 4−point subdivision schemes having the same properties as the corresponding Deslauriers-Dubuc scheme (linearity, stationarity, and homogeneity) which reproduce cubic polynomials and are differentiable.

Given that the algorithm converges to continuous functions, we can prove that it must be "stable". In general terms, an algorithm R is said to be stable if for any data z, $|R(z+\delta z) - R(z)| \leq K|\delta z|$ [**9**].

COROLLARY 3.4.5. *For $-25/56 < \alpha < 15/32$, high resolution subdivision schemes are stable, that is, given $\left|z_{j,k} - \widetilde{z}_{j,k}\right| < \delta \forall k \in \mathbb{Z}$ then $\left|z_{j+n,k} - \widetilde{z}_{j+n,k}\right| < K\delta \forall k \in \mathbb{Z}$ for all integers $n > 0$ and a constant K independent of δ.*

PROOF. Assume we use any 4−point subdivision scheme as an initialisation step on the initial data on $z_{j,k}, \widetilde{z}_{j,k}$. For $-25/56 < \alpha < 15/32$, by theorem 3.4.4, given the initial data $y_{j,k} = \delta_{k,0} \forall k \in \mathbb{Z}$, we get a continuous ($C^1$) interpolation function $F(x)$. Let $M = \|F\|_{L^\infty}$, assume

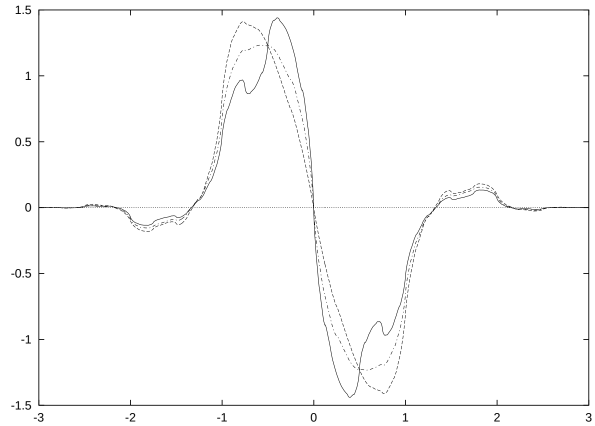

FIGURE 3.4.1. Derivatives of the fundamental functions for $\alpha = -0.2$ (continuous line), $\alpha = 0$ (dash-dot line), and $\alpha = 0.15$ (dashed line). The fundamental functions are defined as the interpolation of $y_{0,k} = \delta_{k,0}$ by the high resolution subdivision scheme initialized with the 4−point Deslauriers-Dubuc dyadic scheme. Derivatives were estimated using first-order forward finite differences after 8 iterations of the high resolution scheme (discarding the placeholders at the last iteration). The $\alpha = 0$ case is in fact the derivative of the Deslauriers-Dubuc fundamental function.

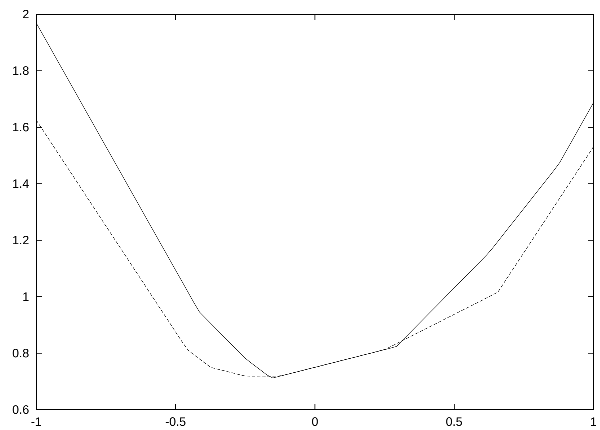

FIGURE 3.4.2. $\lambda_1(\alpha)$ (continuous line) and $\lambda_2(\alpha)$ (dashed line) as in the proof of theorem 3.4.4). The high resolution scheme is differentiable if $\lambda_{HR} = \max\{\lambda_1, \lambda_2\} < 1$.

$\left|z_{j,k} - \widetilde{z}_{j,k}\right| < \delta \forall k \in \mathbb{Z}$, by linearity, the interpolation function of $z_{j,k} - \widetilde{z}_{j,k}$ is given by $f(x) = \sum_{k=-\infty}^{\infty}\left(z_{j,k} - \widetilde{z}_{j,k}\right) F_j\left(x - x_{j,k}\right)$ but since F has compact support $[x_{j,-3}, x_{j,3}]$ then $\|f\|_{L^\infty} \leq 6M\delta$. It means that the values of the stable nodes are bounded by $-6M\delta$ and $6M\delta$. The placeholders must also be bounded by $6M\delta \sum_{k\in\mathbb{Z}}\left|\gamma_{4k-1}^{DD4}\right|$(see equation 3.1.7). □

The author would like to thank S. Dubuc for his help in preparing the manuscript.

Bibliography

[1] I. Daubechies, Orthonormal bases of compactly supported wavelets, Comm. Pure & Appl. Math. 41, pp. 909–996, 1988.

[2] G. Deslauriers and S. Dubuc, Symmetric iterative interpolation processes, Constr. Approx., 5, pp. 49-68, 1989.

[3] G. Deslauriers, S. Dubuc, and D. Lemire, Une famille d'ondelettes biorthogonales sur l'intervalle obtenue par un schéma d'interpolation itérative, Ann. Sci. Math. Québec 23 no. 1, pp. 37-48, 37-48, 1999.

[4] F. Dubeau and R. Gervais, Procédures locale et non locale d'interpolation à l'aide de fonctions splines quadratiques, Ann. Sci. Math. Québec 23 no. 1, pp. 49-61, 1999.

[5] S. Dubuc, Interpolation through an iterative scheme, J. Math. Anal. Appl., 114, pp. 185-204,1986.

[6] S. Dubuc, D. Lemire, J.-L. Merrien, Fourier analysis of 2-point Hermite interpolatory subdivision schemes, J. of Fourier Anal. Appl., 7 no. 5, 2001.

[7] N. Dyn, Subdivision schemes in computer-aided geometric design, Advances in numerical analysis (W. Light, ed.), vol. 2, Clarendon Press, pp. 36-104, 1992.

[8] N. Dyn, J.A. Gregory, and D. Levin, A 4-point interpolatory subdivision scheme for curve design. Comput. Aided Geom. Design, 4, pp. 257-268, 1987.

[9] F. Kuijt and R. van Damme, Stability of subdivision schemes, Memorandum no. 1469, Faculty of Applied Mathematics, University of Twente, the Netherlands.

[10] J.-L. Merrien, A family of Hermite interpolants by bissection algorithms. Numer. Algorithms 2, pp. 187-200, 1992.

[11] J.-L. Merrien, Interpolants d'Hermite C^2 obtenus par subdivision. M2An Math. Model. Numer. Anal. 33, pp. 55-65, 1999.