A FAMILY OF 4-POINT DYADIC HIGH RESOLUTION SUBDIVISION SCHEMES

DANIEL LEMIRE

ABSTRACT. We present a new family of multistep iterative interpolation schemes generalizing subdivision schemes so that quartic polynomials can be reproduced with a $4-$point approach. Interpolation requires two steps: a coarse scale interpolation followed by a fine scale interpolation. The interpolants are C^1, have good local properties and no additional memory requirement.

1. 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 [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 [13, 14, 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) is not differentiable and only reproduces linear polynomials. There has also been considerable interest in vector subdivision schemes and variants of Merrien's Hermite subdivision schemes [9, 15, 11] all characterized by the idea that each node should store a vector instead of a single value.

Adding extra nodes to improve an interpolation scheme is not a new approach and has been used to make spline interpolation local [4]. However, doubling the number of nodes doubles the memory requirements. On the other hand, dyadic subdivision scheme double their memory usage at each step. Hence, we can choose to use one step earlier the upcoming extra storage space without any cost. 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). Because the placeholders are used as predictors, it is reasonable to expect that the new schemes will be as local as usual subdivision schemes. 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". Whereas subdivision schemes immediately fix the interpolated value, we choose to only record the interpolated values as "reasonable guesses" and allow to scheme to correct the guess later based on the results at finer scales. Whereas a vector subdivision scheme would store two or more values per node, we store only one value, but we allow it change over time.

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 at least cubic polynomials. It is also shown that whereas $4-$point subdivision scheme can reproduce at most cubic polynomials, $4-$point high resolution subdivision (HRS) schemes can reproduce quartic polynomials (see Table 1). While there exists $5-$point quartic subdivision schemes, none of them is as local as the presented HRS.

The paper is organized as follows. We begin by a brief review of subdivision schemes and give explicit algorithms for both the dyadic and tetradic $4-$point Deslauriers-Dubuc schemes. Combining these subdivision schemes, we present a family of HRS schemes and show that this new family can reproduce cubic and even quartic polynomials. We conclude by proving that some of these schemes are smooth (C^1).

Key words and phrases. Subdivision Schemes, Fourier transform, CAGD, Wavelets, Lagrange Interpolation, Richardson Extrapolation.

scheme	regularity	number of samples at step j	support of fund. function	reproduced polynomials
Dubuc[5]	C^1	$2^j N$	$[-3,3]$	cubic
Dyn-Gregory-Levin[8]	up to C^1	$2^j N$	$[-3,3]$	up to cubic
Hassan et al.[10]	C^2	$3^j N$	$[-\frac{5}{2},\frac{5}{2}]$	quadratic
presented HRS	C^1	$2^{j+1}N$	$[-3,3]$ or $[-3,4]$	cubic or quartic

TABLE 1. Comparaison between some 4−point iterative interpolation schemes. The quartric HRS scheme is slightly less local because it requires initialization by a one-step 5−point scheme.

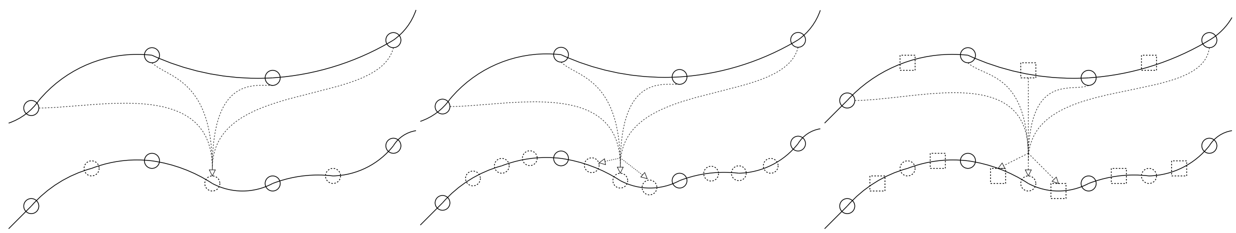

FIGURE 1.1. Diagrams of 4−point subdivision schemes in the dyadic (left) and tetradic (center) cases, and HRS schemes (right). Arrows symbolize the interpolation process. The circles are data samples and, in the HRS diagram, the squares represent placeholders recording "guesses".

2. SUBDIVISION SCHEMES

Let $b>1$ be an integer, 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 a smooth function f such that $f(x_{J,k})=y_{J,k}\forall k\in\mathbb{Z}$. Starting with $(y_{J,k})$ and using the formula

$$y_{j+1,l}=\sum_{k\in\mathbb{Z}}\gamma_{bk-l}y_{j,k} \tag{2.1}$$

for some 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 $f(x_{J,k})=y_{J,k}$ 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 (does not depend on j). Because γ does not depend explicitely on l but rather on $bk-l$ the scheme is *translation invariant* or *homogeneous*. A subdivision scheme is $2N-$point if $\gamma_l=0$ for $|l|\geq Nb$. The fundamental function of an interpolatory $2N-$point $b-$adic scheme has initial data $y_{0,0}=1$ and $y_{0,k}=0$ for all $k\neq 0$; it has a compact support of $[-(Nb-1)/(b-1),(Nb-1)/(b-1)]$ (or $[1-2N,2N-1]$ when $b=2$).

For $N=1,2,3,\ldots$ 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 with 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 with 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 they both converge to differentiable (C^1) interpolation functions. Because we later borrow from these two subdivision schemes, we give explicit algorithms for both schemes.

Algorithm 2.1. *(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 2.2. *(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.*

3. High resolution subdivision schemes

3.1. **Definitions.** In this paper, we want to extend subdivision schemes by hybrid schemes: mixing tetradic and dyadic subdivision schemes for example. Given some data $\{y_{j-1,k}\}_{k \in \mathbb{Z}}$ on the dyadic $x_{j-1,k}$ grid, we must apply a dyadic subdivision scheme as an initialisation step: since HRS schemes can be seen as multistep subdivision schemes, it is not surprising that they require an initialisation step. For 4−point HRS schemes, a sensible choice for the initialization step is the 4−point Deslauriers-Dubuc dyadic scheme (see lemma 3.3) which copies the data at even nodes ($y_{j,2k} = y_{j-1,k}$) and insert new values ("guesses") at odd nodes ($y_{j,2k+1}$ for $k \in \mathbb{Z}$). A value $y_{j,k}$ is "stable" if $y_{j,k} = y_{j+1,2k}$ and other nodes are said to be temporary or are referred to as "placeholders". A HRS on a dyadic grid is interpolatory if all $y_{j,2k}$ values on even nodes ($x_{j,2k}$) are stable so that $y_{j,2k} = y_{j+1,4k} \forall k \in \mathbb{Z}$. Assuming we used an interpolatory subdivision scheme as an initialization step, the following HRS algorithm is interpolatory.

Algorithm 3.1. *(4−point Dyadic HRS Schemes) The following iteration steps depend on* $\alpha \in \mathbb{R}$*, 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 are 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 a more general definition: stationary subdivision schemes (equation 2.1) can be generalized by the linear equation

$$y_{j+1,l} = \sum_{m=1}^{M} \sum_{k \in \mathbb{Z}} \gamma^{(m)}_{Mbk+m-1-l} y_{j,Mk+m-1} \tag{3.1}$$

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 HRS scheme. We say it is interpolatory if it satisfies $y_{j+1,Mbk} = y_{j,Mk}$ and it is $2N-$point if $\gamma_l^{(m)} = 0$ for $|l| \geq MNb$ and $m = 1,\ldots,M$. For $b = M = 2$ the general equation 3.1 becomes

$$y_{j+1,l} = \sum_{k\in\mathbb{Z}} \gamma^{(1)}_{4k-l} y_{j,2k} + \gamma^{(2)}_{4k+1-l} y_{j,2k+1}. \tag{3.2}$$

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

The interpolatory algorithm 3.1 amounts to choosing $\gamma^{(1)}$ and $\gamma^{(2)}$ to be:

$$\gamma^{(1)}_{2k} = \gamma^{DD4}_{2k} + \alpha\left(\delta_{k,0} - \gamma^{DD2}_{k}\right), \quad \gamma^{(1)}_{2k+1} = \gamma^{DD4}_{2k+1}\, \forall k \in \mathbb{Z} \tag{3.3}$$

$$\gamma^{(2)}_0 = \alpha, \qquad \gamma^{(2)}_k = 0\, otherwise \tag{3.4}$$

for some parameter $\alpha \in \mathbb{R}$. Indeed, since $\gamma^{DD4}_{2k} = \gamma^{DD2}_{k}\, \forall k \in \mathbb{Z}$, we can rewrite equation 3.2 for even and odd terms. Firstly, setting $l = 2s$ (l even), we have

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

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

$$y_{j+1,4r} = y_{j,2r} \tag{3.5}$$

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

$$y_{j+1,4r+2} = \alpha y_{j,2r+1} + (1-\alpha) \sum_{k\in\mathbb{Z}} \gamma^{DD2}_{2k-2r-1} y_{j,2k}. \tag{3.6}$$

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

$$\begin{aligned} y_{j+1,2s+1} &= \sum_{k\in\mathbb{Z}} \gamma^{(1)}_{4k-2s-1} y_{j,2k} + \gamma^{(2)}_{4k-2s-1} y_{j,2k+1} \\ &= \sum_{k\in\mathbb{Z}} \gamma^{DD4}_{4k-2s-1} y_{j,2k}. \end{aligned} \tag{3.7}$$

Equations 3.5, 3.6, and 3.7 can be used to describe the presented HRS schemes: while equation 3.5 is the interpolatory condition, equation 3.7 fills the placeholders with tetradic (coarse scale) interpolated values whereas equation 3.6 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 3.2 becomes

$$y_{j+1,l} = \sum_{k\in\mathbb{Z}} \gamma^{DD4}_{4k-l} y_{j,2k}. \tag{3.8}$$

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 3.8 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

$$\widetilde{y}_{j+1,2l} = \sum_{k\in\mathbb{Z}} \gamma^{DD2}_{2k-l} \widetilde{y}_{j,2k} \tag{3.9}$$

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

Proposition 3.2. *For $\alpha = 0$, the HRS scheme given by algorithm 3.1 is equivalent to the $4-$point dyadic Deslauriers-Dubuc subdivision scheme.*

Thus we can say that the presented family of HRS schemes generalizes the dyadic Deslauriers-Dubuc scheme [5, 2].

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

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

and thus equation 3.6 becomes $y_{j+1,4r+2} = p_3\left(x_{j,2r+1}\right)$ for any $\alpha \in \mathbb{R}$. Similarly, equation 3.7 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 HRS schemes defined by equation 3.2 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

$$y_{j+1,l} = \sum_{k\in\mathbb{Z}} \gamma^{DD2}_{2k-l} y_{j,2k} \tag{3.10}$$

followed by algorithm 3.1 with $j+1$. By induction on j using algorithm 3.1, we get the following lemma. Note that while this algorithm is as local as the corresponding Deslauriers-Dubuc subdivision scheme in the sense that the fundamental function has support $[-3,3]$.

Lemma 3.3. *HRS schemes given by algorithm 3.1 reproduce cubic polynomials and are interpolatory when using a one step interpolatory 4-point dyadic Deslauriers-Dubuc interpolation (equation 3.10) as initialization.*

We can also get a stronger result by choosing a specific α. We can write any quartic polynomial p_4 as $p_4(x) = a_4x^4 + p_3(x)$ where p_3 is some cubic polynomial. Because of the Generalized Rolle's theorem, given any 4 points ξ_1,ξ_2,ξ_3, and ξ_4, the corresponding cubic polynomial $p_{Lagrange3}$ approximates p_4 with error

$$\begin{aligned} p_4(x) - p_{Lagrange3}(x) &= \frac{p_4'(\xi)\left(x-\xi_1\right)\left(x-\xi_2\right)\left(x-\xi_3\right)\left(x-\xi_4\right)}{4!} \\ &= a_4\left(x-\xi_1\right)\left(x-\xi_2\right)\left(x-\xi_3\right)\left(x-\xi_4\right) \end{aligned}$$

for some ξ. In other words, the error depends only on a_4 and the geometry of the sample points ξ_i with respect to x. This makes the task of cancelling out the errors given more two different approximating cubic polynomials convenient as we shall see.

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 3.7 into 3.6 to get

$$\begin{aligned} y_{j+1,4r+2} &= \alpha y_{j,2r+1} + (1-\alpha)\sum_{k\in\mathbb{Z}} \gamma^{DD2}_{2k-2r-1} y_{j,2k} \\ &= \alpha \sum_{k\in\mathbb{Z}} \gamma^{DD4}_{4k-2r-1} y_{j-1,2k} + (1-\alpha)\sum_{k\in\mathbb{Z}} \gamma^{DD2}_{2k-2r-1} y_{j,2k}. \end{aligned} \tag{3.11}$$

We want to show that $y_{j+1,4r+2} = p_4\left(x_{j,2r+1}\right)$ for some $\alpha \in \mathbb{R}$ 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 3.11 explicitely:

$$\sum_{k\in\mathbb{Z}} \gamma^{DD4}_{4k-2r-1} y_{j-1,2k} = p_3\left(x_{j,2r+1}\right) + a_4 \sum_{k\in\mathbb{Z}} \gamma^{DD4}_{4k-2r-1} \times \left(x_{j-1,2k} = x_{j,4k}\right)^4 \tag{3.12}$$

$$= p_4\left(x_{j,2r+1}\right) - \frac{105a_4}{2^{4j}} \tag{3.13}$$

and

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

Incidently, these two results can be checked using the Generalized Rolle's theorem. Hence, setting $\alpha=-3/32$ in equation 3.11, 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 HRS scheme with $\alpha=-3/32$ reproduces quartic polynomials.

For pratical applications, we wish to initialize HRS 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 $y_{j,2k+1}=p_4\left(x_{j,2k+1}\right)-\frac{105a_4}{16\times2^{4j}}$ 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 3.4. *($5-$point "Initialization" 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*

$$\gamma_{j,k}=5y_{j,k-2}-24y_{j,k-1}+45y_{j,k}-40y_{j,k+1}+15y_{j,k+1}\,\forall k\in\mathbb{Z}; \tag{3.14}$$

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

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

To see that algorithm 3.4 properly initializes 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 $y_{J+1,2k+1}=p_4\left(x_{J+1,2k+1}\right)-\frac{105a_4}{16\times2^{4(J+1)}}$. 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 3.14 can be derived by finding the quartic polynomial $p_{J,k}$ satisfying $p_{J,k}\left(x_{j,l}\right)=y_{J,l}$ for $l=k-2,...,k+2$ and setting $\gamma_{J,k}=p_{J,k}\left(x_{J,k+4}\right)$. Hence, by formula 3.13, we have the following lemma.

Lemma 3.5. *Algorithm 3.4 describes a $5-$point dyadic subdivision scheme such that with $y_{j-1,k}=p_4\left(x_{j-1,k}\right)$ where $p_4(x)=a_4x^4+a_3x^3+a_2x^2+a_1x+a_0$ is a quartic polynomial, $y_{j,2k+1}=p_4\left(x_{j,2k+1}\right)-\frac{105a_4}{16\times2^{4j}}$ and $y_{j,2k}=p_4\left(x_{j,2k}\right)$ for all $k\in\mathbb{Z}$.*

Then , because we have a proper initialization scheme, we can reproduce quartic polynomials as the next proposition states.

Proposition 3.6. *For any given j and $y_{j-1,k}=p_4\left(x_{j-1,k}\right)$ where p_4 is a quartic polynomial, algorithm 3.4 followed by algorithm 3.1 with $\alpha=-3/32$ for the following steps guarantees 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, the $4-$point HRS algorithm reproduces quartic polynomials when $\alpha=-3/32$.*

This result is significant because it is not possible for $4-$point subdivision schemes to reproduce quartic polynomials. 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 quartic polynomials as shown in the next proposition.

Proposition 3.7. *A $4-$point dyadic subdivision scheme cannot reproduce quartic polynomials.*

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 interpolate $y_{1,3}=0\neq P_4\left(\frac{3}{2}\right)=\frac{9}{16}$. □

By the proof of proposition 3.7, we see that only subdivision schemes using 5 points can interpolate quartic polynomials. Starting with $y_{0,k}=\delta_{k,0}\,\forall k\in\mathbb{Z}$, in the best possible case, a $5-$point subdivision scheme gives 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 HRS scheme described by proposition 3.6 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 $[-3,4]$of size 7 taking into account the $5-$point initialization scheme. Therefore, we have a new quartic interpolation scheme more local than $5-$point quartic dyadic subdivision schemes. Such good local properties are possible because in the presented HRS scheme, the placeholders are only used to predict upcoming interpolations except maybe during the initialization.

3.3. **Sufficient conditions for regularity.** Given that for $\alpha = 0$, the presented HRS scheme is equivalent to the Deslauriers-Dubuc subdivision which is C^1, it is reasonable to expect HRS schemes to be C^1 for some range of α values. Moreover, motivated by proposition 3.6, we need to show that this range of values include $\alpha = -3/32$. At this point, it is convenient to rewrite formula 3.2 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 3.10), 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 $P^{j+1}(z) = P_4(z) P^j(z^2)$. We can rewrite the general equation for HRS 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 HRS sheme is (see equation 3.2 and algorithm 3.1)

$$\begin{aligned}
P^{j+1}(z) &= \left\{P_4(z) - \alpha P_2\left(z^2\right) + \alpha\right\}\left(\frac{P^j\left(z^2\right) + P^j\left(-z^2\right)}{2}\right) + \alpha\left(\frac{P^j\left(z^2\right) - P^j\left(-z^2\right)}{2}\right) \\
&= \left\{\frac{P_4(z) - \alpha P_2\left(z^2\right)}{2} + \alpha\right\} P^j\left(z^2\right) + \frac{P_4(z) - \alpha P_2\left(z^2\right)}{2} P^j\left(-z^2\right) \\
&= \Gamma_1(z) P^j\left(z^2\right) + \Gamma_2(z) P^j\left(-z^2\right). \qquad (3.15)
\end{aligned}$$

Because $\gamma_{2k}^{DD4} = \gamma_k^{DD2}\, \forall k \in \mathbb{Z}$, 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 3.2 becomes

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

Averaging $P^{j+1}(z)$ and $P^{j+1}(-z)$

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

we rediscover that when $\alpha = 0$ the HRS scheme becomes the dyadic Deslauriers-Dubuc scheme (proposition 3.2).

Following Dyn [7], we want 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

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

Note that $D_{j,k} = D^1_{j,k}$. Let H_1^j be the symbol for $dy_{j,k}/dx_j$, then

$$\begin{aligned} H_1^j(z) &= \sum_{k\in\mathbb{Z}} 2^j \left(y_{j,k+1} - y_{j,k}\right) z^k \\ &= \sum_{k\in\mathbb{Z}} 2^j y_{j,k} z^{k-1} - \sum_{k\in\mathbb{Z}} 2^j y_{j,k} z^k \\ &= 2^j(1/z-1)P^j(z) = 2^j(1-z)P^j(z)/z, \end{aligned}$$

and thus $P^j\left(z^2\right) = z^2 2^j H_1^j\left(z^2\right)/(1-z^2)$, $P^j\left(-z^2\right) = -z^2 2^j H_1^j\left(-z^2\right)/(1+z^2)$, and $P^{j+1}(z) = z2^{j+1}H_1^{j+1}(z)/(1-z)$. 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 3.15) gives

$$H_1^{j+1}(z) = \frac{2z(1-z)}{(1-z^2)}\Gamma_1(z)H_1^j\left(z^2\right) - \frac{2z(1-z)}{(1+z^2)}\Gamma_2(z)H_1^j\left(-z^2\right). \tag{3.16}$$

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 they can be computed by (see derivation of equation 3.16 above)

$$H_n^{j+1}(z) = \left(\frac{2z}{1+z}\right)^n \Gamma_1(z)H_n^j\left(z^2\right) + \left(\frac{-2z(1-z)}{1+z^2}\right)^n \Gamma_2(z)H_n^j\left(-z^2\right). \tag{3.17}$$

H_n is said to be the symbol of a HRS 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

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

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

Lemma 3.8. *For HRS schemes given by algorithm 3.1, the finite differences of order* $1,2,3,$ *and* 4*, that is* $d^{(n)}y_{j,k}$ *with* $n = 1,2,3,$ *and* 4*, are given by a corresponding HRS scheme.*

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

$$dH_{n-1}^j(z) = \frac{(1-z)}{z}H_{n-1}^j(z) = \frac{2^{j(n-1)}(1-z)^n}{z^n}P^j(z). \tag{3.18}$$

Replacing H_{n-1} by dH_{n-1} in equation 3.17, we find

$$dH_{n-1}^{j+1}(z) = \frac{1}{2}\left\{\left(\frac{2z}{1+z}\right)^n \Gamma_1(z)dH_{n-1}^j\left(z^2\right) + \left(\frac{-2z(1-z)}{1+z^2}\right)^n \Gamma_2(z)dH_{n-1}^j\left(-z^2\right)\right\}. \tag{3.19}$$

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

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

Theorem 3.9. *(Dyn) Given trigonometric polynomials* $\Gamma_1(z)$ *and* $\Gamma_2(z)$*, the HRS scheme defined by*

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

is C^n *if the symbol corresponding to finite differences of order* $n+1$

$$dH_n^j(z) = \frac{2^{jn}(1-z)^{n+1}}{z^{n+1}}P^j(z)$$

is the symbol of a HRS scheme converging uniformly to zero for all bounded initial data.

Proof. See the proof of theorem 3.4 [7] as it applies toHRS schemes. The key point being that for an iterative interpolation scheme to be C^n, it is sufficient for the finite differences $(d^{n+1}y_{j,k}/\left(dx_j\right)^n)$ to converge uniformly to zero. □

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 $M_{j+1}\le\max_{l=0,1}\left\{\sum_{k\in\mathbb{Z}}|\gamma_{2k-l}|\right\}M_j$ 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 HRS 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 $M_{j+1}\le\lambda M_j$. Given a symbol $Q(z)=\sum_k q_kz^k$, define $\|Q(z)\|_{sup}=\sup_k\{|q_k|\}$ and $\|Q(z)\|_{\Sigma}=\max\{\sum_k|q_k|\}$. With $P^{j+1}(z)=\Phi_1(z)P^j\left(z^2\right)+\Phi_2(z)P^j\left(-z^2\right)$, λ_{HR} is given by

$$\lambda_{HR} = \max\{\lambda_1,\lambda_2\} \tag{3.20}$$

$$= \max\left\{\left\|\frac{\Phi_1(z)+\Phi_1(-z)+\Phi_2(z)-\Phi_2(-z)}{2}\right\|_{\Sigma},\left\|\frac{\Phi_1(z)-\Phi_1(-z)+\Phi_2(z)+\Phi_2(-z)}{2}\right\|_{\Sigma}\right\} \tag{3.21}$$

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

Lemma 3.10. *A HRS 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 3.21.*

We are now ready to prove the following theorem.

Theorem 3.11. *For $-25/56<\alpha<15/32$, the HRS schemes given by equation 3.15 (algorithm 3.1) are C^1.*

Proof. The symbol of the HRS scheme $dD_{j,k}=dD^1_{j,k}$, dH_1 is given by (see equation 3.19)

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

By theorem 3.9, it is enough to show that $dD_{j,k}$ converges uniformly to zero for all bounded initial data. However, using lemma 3.10, we know that it is sufficient to prove that $\lambda_{HR}<1$ with $\Phi_1(z)=2z^2\Gamma_1(z)/\left(1+z\right)^2$ and $\Phi_2(z)=2z^2(1-z)^2\Gamma_2(z)/\left(1+z^2\right)^2$. We have

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

Hence, for $-25/56<\alpha<15/32$, we have $\lambda_1<1$, whereas for $-7/12<\alpha<5/8$, $\lambda_2<1$. Therefore, 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. 3.2). ☐

Theorem 3.11 is illustrated by Fig. 3.1 where the derivatives 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 and is local, we can easily prove that it must be "stable". In general terms, an algorithm R is said to be stable if for any data z, $|R(z+\delta z)-R(z)|\le K|\delta z|$ [12].

Corollary 3.12. *For $-25/56<\alpha<15/32$, the HRS schemes (algorithm 3.1) 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 3.11, 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 $\left|z_{j,k}-\widetilde{z}_{j,k}\right|<\delta\,\forall k\in\mathbb{Z}$, 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 $\|f\|_{L^\infty}\le 6M\delta$. It means that the values of the stable nodes are bounded by $-6M\delta$ and $6M\delta$. The placeholders must also be bounded by $6M\delta\sum_{k\in\mathbb{Z}}\left|\gamma^{DD4}_{4k-1}\right|$(see equation 3.7). ☐

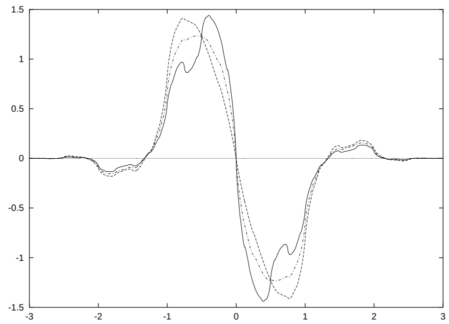

FIGURE 3.1. Derivatives of the fundamental functions for $\alpha = -0.2$ (continuous line), $\alpha = 0$ (dash-dot line), and $\alpha = 0.15$ (dashed line). The fundamental functions are defined as the interpolation of $y_{0,k} = \delta_{k,0}$ by the HRS scheme initialized with the $4-$point Deslauriers-Dubuc dyadic scheme. Derivatives were estimated using first-order forward finite differences after 8 iterations of the HRS (discarding the placeholders at the last iteration). The $\alpha = 0$ case is in fact the derivative of the Deslauriers-Dubuc fundamental function.

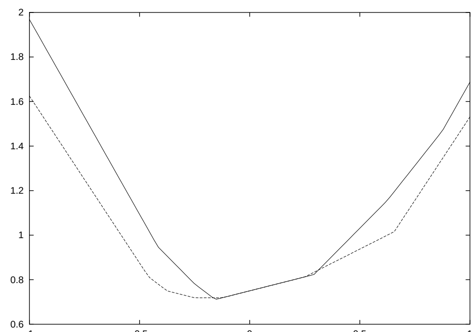

FIGURE 3.2. $\lambda_1(\alpha)$ (continuous line) and $\lambda_2(\alpha)$ (dashed line) as in the proof of theorem 3.11). For a given α, an HRS scheme is differentiable if $\lambda_{HR}(\alpha) = \max\{\lambda_1(\alpha), \lambda_2(\alpha)\} < 1$.

REFERENCES

[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] B. Han, Approximation Properties and Construction of Hermite Interpolants and Biorthogonal Multiwavelets, Journal of Approximation Theory, 110 no.1, pp. 18-53, 2001.

[10] M.F. Hassan, I.P. Ivrissimitzis, N.A. Dodgson, and M.A. Sabin, An Interpolating $4-$Point C^2 Ternary Stationary Subdivision Scheme, submitted for publication to CAGD (December 18, 2001).

[11] C. Heil and D. Colella, Matrix refinement equations and subdivision schemes, J. Fourier Anal. Appl., 2, pp. 363-377, 1996.

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

[13] J.-L. Merrien, A family of Hermite interpolants by bissection algorithms. Numer. Algorithms 2, pp. 187-200, 1992.
[14] J.-L. Merrien, Interpolants d'Hermite C^2 obtenus par subdivision. M2An Math. Model. Numer. Anal. 33, pp. 55-65, 1999.
[15] G. Plonka, Approximation order provided by refinable function vectors, Constr. Approx. 13, pp. 221-244, 1997.