A Family of 4-point Dyadic High Resolution Subdivision Schemes

Daniel Lemire

Research Officer

National Research Council of Canada (NRC)

email: lemire@ondelette.com

Subdivision: why care?

- ✔ multiscale approach

Subdivision: why care?

- ✔ multiscale approach
- ✔ local interpolation

Subdivision: why care?

- ✔ multiscale approach
- ✔ local interpolation
- ✔ lead to compactly supported wavelets

How Dubuc did it!

✔ 4 points $\longrightarrow$ cubic polynomial

How Dubuc did it!

✔ 4 points $\longrightarrow$ cubic polynomial
$\longrightarrow$ midpoint value *(dyadic interpolation)*

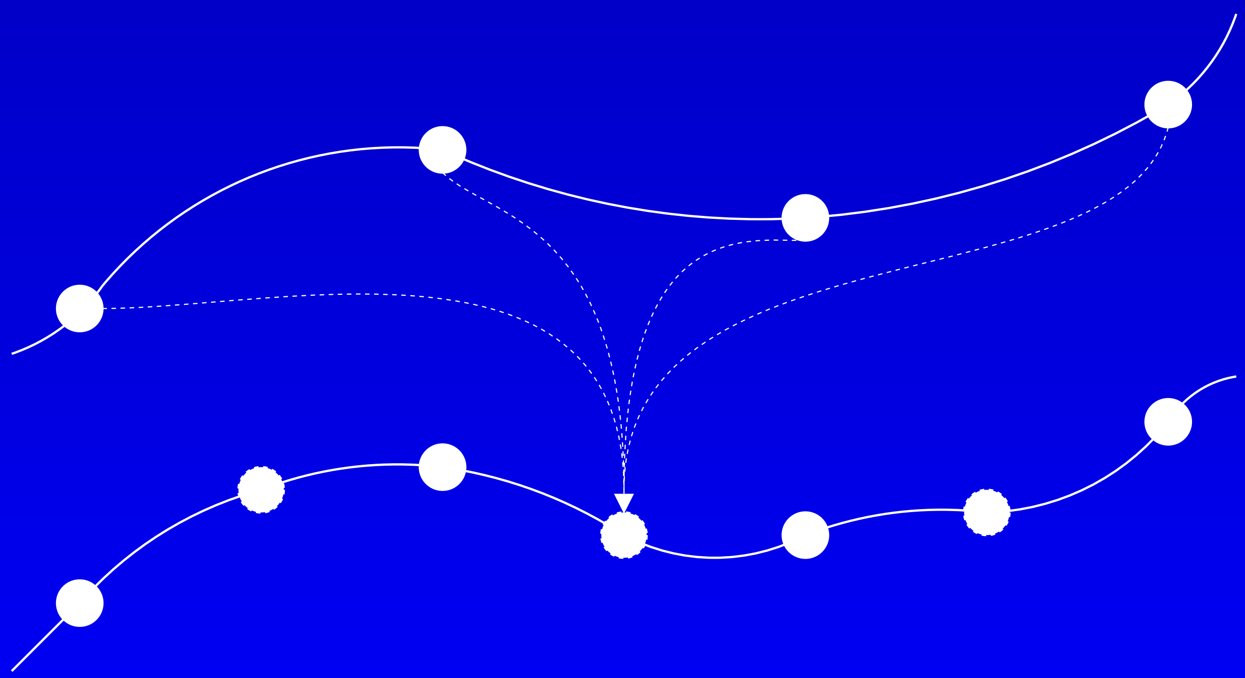
How Dubuc did it!

- ✔ 4 points $\longrightarrow$ cubic polynomial
  $\longrightarrow$ midpoint value *(dyadic interpolation)*
- ✔ repeat at finer scales

How Dubuc did it!

- ✔ 4 points $\longrightarrow$ cubic polynomial
  $\longrightarrow$ midpoint value *(dyadic interpolation)*

- ✔ repeat at finer scales



One follow-up...

Same story but...

One follow-up...

Same story but...

✔ 4 points $\longrightarrow$ cubic polynomial

One follow-up...

Same story but...

✔ 4 points $\longrightarrow$ cubic polynomial

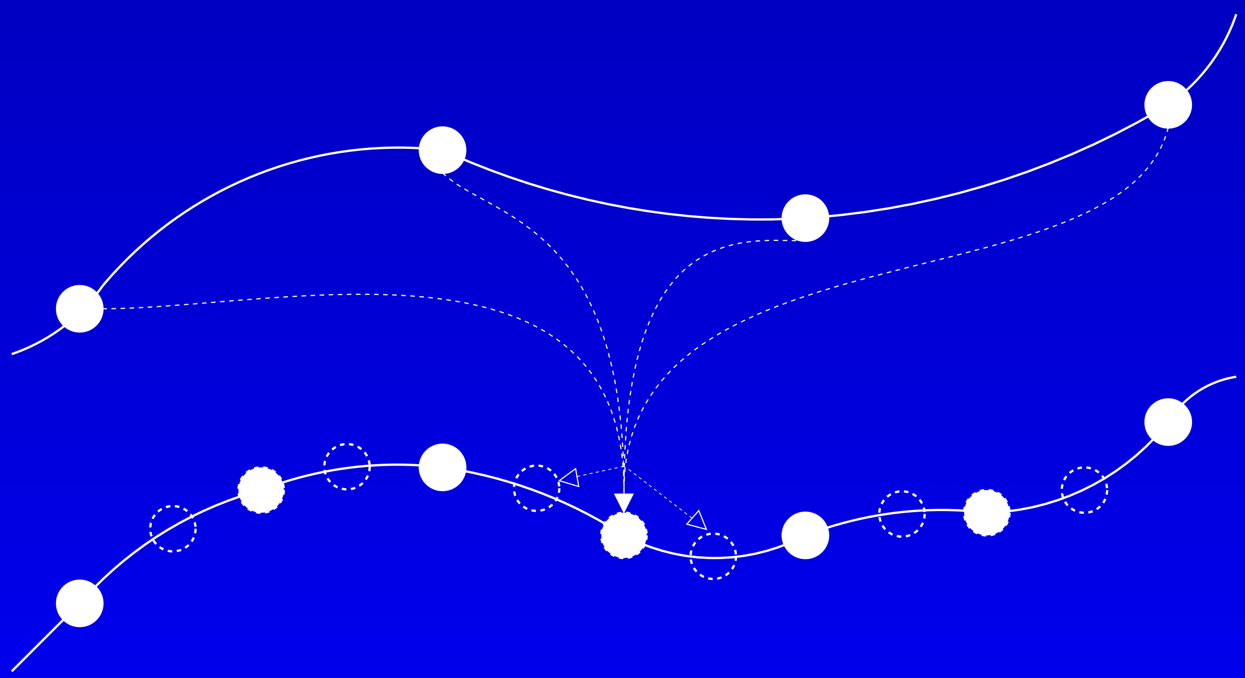
$\longrightarrow$midpoint **and quartile** value *(tetradic interpolation)*

One follow-up...

Same story but...

✔ 4 points $\longrightarrow$ cubic polynomial

$\longrightarrow$midpoint **and quartile** value *(tetradic interpolation)*



One more simple trick?

Recall Richardson's extrapolation:

One more simple trick?

Recall Richardson's extrapolation:

1. *an error or order p*

One more simple trick?

Recall Richardson's extrapolation:

1. *an error or order p*

$$a_{\Delta x} = a_{true} +$$

One more simple trick?

Recall Richardson's extrapolation:

1. *an error or order p*

$$a_{\Delta x} = a_{true} + \underbrace{k\left(\Delta x\right)^p + O\left(\Delta x^{p+1}\right)}_{\text{some error}}$$

2. $\Delta x \rightarrow \Delta x/2$

One more simple trick?

Recall Richardson's extrapolation:

1. *an error or order p*

$$a_{\Delta x} = a_{true} + \underbrace{k(\Delta x)^p + O\left(\Delta x^{p+1}\right)}_{\text{some error}}$$

2. $\Delta x \to \Delta x/2$

$$a_{\frac{\Delta x}{2}} = a_{true} +$$

One more simple trick?

Recall Richardson's extrapolation:

1. *an error or order p*

$$a_{\Delta x} = a_{true} + \underbrace{k\left(\Delta x\right)^p + O\left(\Delta x^{p+1}\right)}_{\text{some error}}$$

2. $\Delta x \rightarrow \Delta x/2$

$$a_{\frac{\Delta x}{2}} = a_{true} + \underbrace{k\left(\frac{\Delta x}{2}\right)^p + O\left(\Delta x^{p+1}\right)}_{\text{somewhat smaller error}}$$

3. Combine $a_{\Delta x}$ and $a_{\frac{\Delta x}{2}}$,

3. Combine $a_{\Delta x}$ and $a_{\frac{\Delta x}{2}}$,

$$\frac{2^p \times a_{\frac{\Delta x}{2}} - a_{\Delta x}}{2^p - 1} = a_{true} +$$

3. Combine $a_{\Delta x}$ and $a_{\frac{\Delta x}{2}}$,

$$\frac{2^p \times a_{\frac{\Delta x}{2}} - a_{\Delta x}}{2^p - 1} = a_{true} + \underbrace{k\left(\frac{\Delta x}{2}\right)^p + O\left(\Delta x^{p+1}\right)}_{\text{much smaller error}}$$

A recipe

Mix cubic schemes

A recipe

Mix cubic schemes

✔ coarse scale =

A recipe

Mix cubic schemes

✔ coarse scale = tetradic scheme

A recipe

Mix cubic schemes

- ✔ <u>coarse scale</u> = tetradic scheme
- ✔ <u>fine scale</u> =

A recipe

Mix cubic schemes

- ✔ coarse scale = tetradic scheme
- ✔ fine scale = dyadic scheme

A recipe

Mix cubic schemes

✔ coarse scale = tetradic scheme

✔ fine scale = dyadic scheme

cubic (coarse) + cubic (fine) = cubic + 1 = quartic

Different kind of multiscale

Subdivision schemes are multiscale?

Different kind of multiscale

Subdivision schemes are multiscale? Yes! One scale at a time!

Different kind of multiscale

Subdivision schemes are multiscale? Yes! One scale at a time!

Now take two scales at once!

✔ ”multistep” subdivision scheme or

Different kind of multiscale

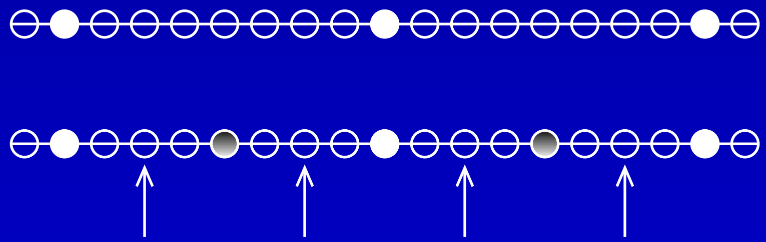
Subdivision schemes are multiscale? Yes! One scale at a time!

Now take two scales at once!

✔ ”multistep” subdivision scheme or

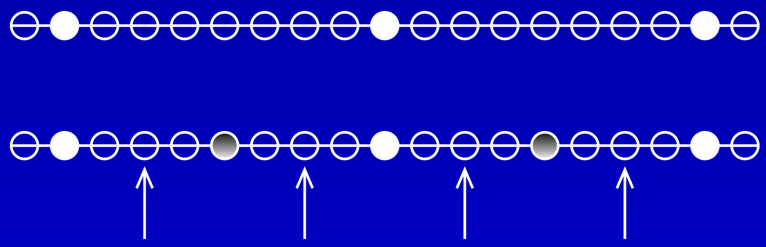
✔ “High Resolution Subdivision” (HRS) Scheme

Guessing early or coarsing it up



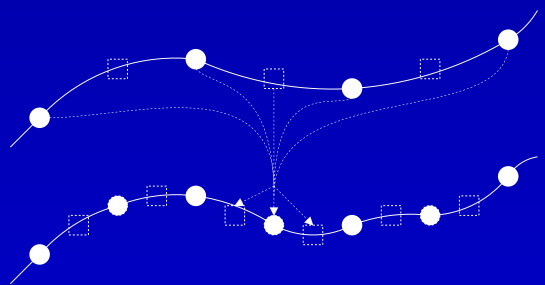
✔ Ample storage: why not use it early?

Guessing early or coarsing it up



- ✔ Ample storage: why not use it early?
- ✔ Good guesses for the quartiles

Crunch the tetradic tree into a dyadic one



How does it compare to Vector Subdivision Schemes?

- ✔ Vector $\Rightarrow$ several fixed value per node

How does it compare to Vector Subdivision Schemes?

- ✔ Vector $\Rightarrow$several fixed value per node
- ✔ HRS$\Rightarrow$one value per node but **allowed to changed over time**

HRS Algorithm

1. recopy stable data: $y_{j+1,4k} = y_{j,2k}$;

HRS Algorithm

1. recopy stable data: $y_{j+1,4k} = y_{j,2k}$;
2. Tetradic scheme on even (stable) nodes:

HRS Algorithm

1. recopy stable data: $y_{j+1,4k} = y_{j,2k}$;

2. Tetradic scheme on even (stable) nodes:

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

HRS Algorithm

1. recopy stable data: $y_{j+1,4k} = y_{j,2k}$;

2. Tetradic scheme on even (stable) nodes:

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

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

HRS Algorithm

1. recopy stable data: $y_{j+1,4k} = y_{j,2k}$;

2. Tetradic scheme on even (stable) nodes:

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

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

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

3. Update midpoint :

3. Update midpoint :

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

3. Update midpoint :

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

Both $y^{temporary}_{j+1,4k+2}$ and $y_{j,2k+1}$ = good guesses (cubic)

3. Update midpoint :

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

Both $y^{temporary}_{j+1,4k+2}$ and $y_{j,2k+1}$ = good guesses (cubic)

✔ $\alpha = 0 \longrightarrow y^{temporary}_{j+1,4k+2}$

3. Update midpoint :

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

Both $y^{temporary}_{j+1,4k+2}$ and $y_{j,2k+1}$ = good guesses (cubic)

✔ $\alpha = 0 \longrightarrow y^{temporary}_{j+1,4k+2}$

✔ $\alpha = 1 \longrightarrow$ only $y_{j,2k+1}$

Key Mathematical Argument

We replace

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

with

Key Mathematical Argument

We replace

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

with

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

Smoothness

- $\alpha = 0 \Rightarrow$ Dubuc

Smoothness

- $\alpha = 0 \Rightarrow$Dubuc
- Dubuc$\Rightarrow C^1$.
- For $-25/56 < \alpha < 15/32$, the HRS schemes are C^1.

Reproduced polynomials

▷ HRS always reproduce cubic polynomials.

Reproduced polynomials

- HRS always reproduce cubic polynomials.
- Reproduce quartic polynomials when $\alpha = -3/32$.

Reproduced polynomials

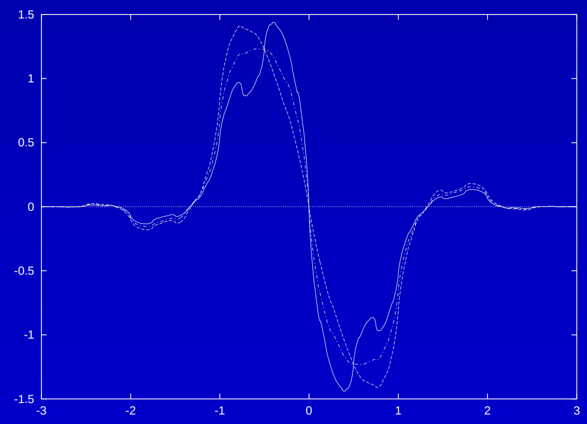
- $\triangleright$ HRS always reproduce cubic polynomials.
- $\triangleright$ Reproduce quartic polynomials when $\alpha = -3/32$.

Gotcha! Need proper initialization!

Conclusion (a comparative table)

scheme	regularity	reproduced polynomials
Dubuc[5]	C^1	cubic
Deslauriers-Dubuc[2]	C^1	cubic
Dyn-Gregory-Levin[8]	up to C^1	up to cubic
Hassan et al.[10]	C^2	quadratic
presented HRS	up to C^1	cubic to quartic

Bonus material 1 - nice pictures



Derivatives of the fundamental functions for $\alpha = -0.2$ (continuous line), $\alpha = 0$ (dash-dot line), and $\alpha = 0.15$ (dashed line).

Bonus material 2 - intermediate result

- (Dyn) Given trigonometric polynomials $\Gamma_1(z)$ and $\Gamma_2(z)$, the HRS scheme defined by

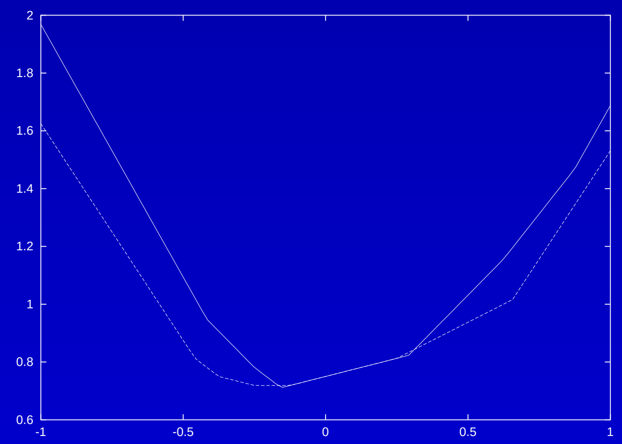
$$P^{j+1}(z) = \Gamma_1(z)P^j\left(z^2\right) + \Gamma_2(z)P^j\left(-z^2\right)$$

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.

Bonus material 3 - more on proof



For a given α, an HRS scheme is differentiable if

$$\lambda_{HR}(\alpha) = \max\{\lambda_1(\alpha), \lambda_2(\alpha)\} < 1.$$

Bonus material 4 - proper initialization

1. recopy data at $x_{j+1,2k} = x_{j,k}$: $y_{j+1,2k} = y_{j,k}$;

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

$$\gamma_{j,k} = 5y_{j,k-2} - 24y_{j,k-1} + 45y_{j,k} - 40y_{j,k+1} + 15y_{j,k+1}; \qquad (1)$$

3. interpolate midpoint :

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

References

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

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

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