Multi-scale mathematical techniques for signal processing

Summary

Sergio López Ureña
Supervisor: Rosa Donat

VNIVERSITAT DE VALÈNCIA

PhD in Mathematics
July 2019, Burjassot (Valencia)

Acknowledgments

I would not like to miss this opportunity to thank those people who have made my PhD studies an incredible experience, in all its aspects, and that I will fondly remember my whole life.

First of all, I want to express my gratitude to Rosa Donat. During these years she has taught me that to develop new mathematical knowledge is as important as knowing how to transmit the results to other people. I admire her efforts to find the best way to express the ideas we want to convey.

To Professors Mª Celia García and José Ramón Torres, from the Department of Analytical Chemistry, for the interesting research they proposed and that we have enjoyed side by side. With you I have learned that each of us comes from a school, with our own methodologies and knowledge, and that in a team with willingness they all add up.

To my PhD student colleagues, almost all doctors already, for the day-to-day, for constantly reminding me that the doctorate is much more than writing a thesis.

To Professors Costanza Conti (U. Firenze) and Lucia Romani (U. Milano-Bicocca), of whom I admire their collaborative and persevering relationship, for maintaining a constant routine of working with me for three months, which led to a very exciting research.

To Professor Tomas Sauer and my colleagues in the FORWISS department, for welcoming me to the University of Passau, treating me as one of the team, and sharing with me the German culture. I am fascinated by your work environment and the relationship you have with other colleagues.

To all the workers of the department, of the faculty and of the University of Valencia, for building a pleasant and motivating work environment.

To the Ministry of Science, Innovation and Universities, for their financial support with the FPU14/02216 grant.

To my family and my friends, for their support and trust, for helping me see my situation with perspective and learn to appreciate it. Especially to my parents and my brother, for always listening and understanding me.

And of course, to Lydia, thank you so much for daily joining me on this adventure. Thanks to you I finish the doctorate with the feeling that this has only just begun.

Index

1 Introduction: Signal processing

We understand by *signal* a function that contains information about the behavior (or attributes) of a system or phenomenon [77]. Voice recordings, electrocardiograms, photographs, chromatograms, weather maps, etc. are examples of signals.

The signals usually contain large amounts of data from which the information, relevant to a particular application, may be difficult to extract. An input signal is *processed* to obtain an output signal in order to extract information or modify certain characteristics. For example, to a music recording, the Fourier transform [66] can be applied to convert the original time-intensity signal into another frequency-intensity. An electrocardiogram can present interferences that may be removed [73], perhaps coming from nearby medical instruments.

The processing is usually faster and simpler in a *sparse* representation of the signal, where a few coefficients reveal the most characteristic or representative information. These representations can be constructed by decomposing the signals using elementary wave shapes belonging to a given family.

The Fourier series are a classic example of representation of functions, which associated family is $\{\exp(\imath nt)\}_{n\in\mathbb{N}}$. If a function is $\mathcal{C}^\alpha$, that is, it is $\alpha \in \mathbb{N}$ times differentiable and continuous, then the sequence of its Fourier coefficients, $\{c_n\}_{n\geq 0}$, decreases with speed $O(n^{-\alpha})$. Therefore, for smooth functions the coefficients decay quickly and, if those coefficients with a value below a certain threshold are ignored, a *sparse* representation is obtained, based on a few terms of the Fourier series. However, for functions with discontinuities, the coefficients barely decrease at the rate $O(n^{-1})$.

Signal processing techniques based on Fourier decomposition have become basic tools in a wide variety of applications in many fields of science.

Despite the ability of Fourier analysis to represent smooth functions, it is a global decomposition. An isolated singularity dominates the behavior of all the coefficients of a discontinuous function. The approximations of functions based on truncating the Fourier series present the so-called *Gibbs Phenomenon*, which consists in the appearance of oscillations around the discontinuity that do not disappear, no matters how many terms are added to the truncated series. The Gibbs phenomenon makes the Fourier transform no longer a useful tool in many contexts. In particular, in image processing this phenomenon is visualized as an artifact around the contours of objects (which can be interpreted as discontinuities of the signal) [59].

In natural languages, a richer family of words (dictionary) helps to build shorter and more precise sentences. Similarly, suitable families are needed to construct sparse representations of complex signals. A "good" representation can improve pattern recognition, data compression or noise reduction. The discovery of orthogonal families of time-frequency local functions [38, 67, 71], such as the orthogonal

wavelet bases, opened the door to other types of transformations capable to obtain space-frequency local representations. Mallat used the orthonormal wavelet bases as a mathematical tool to describe the increase of information between different 'resolution levels' in a multi-scale decomposition of an image. This type of decomposition was obtained from *filter schemes*.

In a typical two-band filter scheme, the input signal is convolved separately with two different filters, a low-pass one and a high-pass one. The low-pass filter discards the highly oscillating part of the data, allowing only the low frequency part to "pass". While the high-pass filter extracts the high frequency part of the signal. Once obtained the two sequences resulting from the convolutions, both are *downsampled* to retain the even (or odd) elements and discard the rest. The process of obtaining the two new signals through the convolution and downsampling operators is known as *analysis* or *encoding*. From both contributions the original signal can be reconstructed, or at least an approximation to it, being necessary to apply an *upsampling* operator and two new filters, suitable and according to the downsampling and to the initial filters. These operations constitute the process of *synthesis* or *decoding*.

From 1986, Meyer and Mallat developed the bases of the *Multiresolution Analysis*, and the associated multiscale transformations. A multi-scale representation of a (discrete) signal consists of a low-resolution approximation of the original signal and a sequence of "details", which are the difference of information between consecutive resolution levels. From this, the relationship between these mathematical tools and filter schemes, which are widely used in Electronic Engineering for signal processing [38].

In the papers that make up this doctoral thesis, it was used the multiresolution framework designed by A. Harten at the end of the 1980s. In the following section, we describe in some detail this framework, which in a certain sense can consider a generalization of the multiresolution analyzes based on wavelet theory.

2 Harten's Multiresolution Framework

The development of wavelet theory [68, 67, 70, 71] can be considered as the starting point for local decompositions by scales, which have undoubtedly had a great impact in various fields of science.

The construction of *wavelet basis* relies on functions defined from the shifts and dilations of a single 'mother' function. Initially, the design and analysis used techniques of Harmonic Analysis in an intensive way, which made the extension to delimited domains and general geometries difficult.

In [55, 56], A. Harten develops a general *MultiResolution Framework* for data representation (HMRF), which is based on the Approximation Theory, allowing a better adaptation to all types of geometries.

The HMRF [10, 11, 12, 13, 56] is based on two *operators*, *decimation* and *prediction*, which link discrete data associated with two consecutive resolution levels. From an algebraic point of view, decimation and prediction can be considered simply as operators that connect linear vector spaces of countable dimension, V^k, that represent in some way the different levels of resolution of the (discrete) data that are intended to be analyzed (the resolution will be increase with k), that is,

$$D_{k+1}^k : V^{k+1} \longrightarrow V^k, \quad P_k^{k+1} : V^k \longrightarrow V^{k+1}.$$

While the decimation D_{k+1}^k is assumed to be linear, there are no *a priori* restrictions on the HMRF for the prediction P_k^{k+1} to be. The only restriction between both operators in this environment is the *consistency*, this is

$$D_{k+1}^k P_k^{k+1} = I_{V^k}, \tag{1}$$

where I_{V^k} is the identity operator in V^k.

If at the finest level we have the data $v^L \in V^L$, the decimation can restrict the data to coarser spaces, defining recursively

$$v^k := D_{k+1}^k v^{k+1}, \qquad k = 0, \ldots, L-1.$$

Prediction performs the opposite process, that is, generates new data in a finer space. The error associated with the use of a prediction operator is measured by the expression

$$e^k := v^{k+1} - P_k^{k+1} D_{k+1}^k v^{k+1} = (I_{V_{k+1}} - P_k^{k+1} D_{k+1}^k) v^{k+1}. \tag{2}$$

Note that v^{k+1} can be calculated from v^k and e^k:

$$v^{k+1} = e^k + P_k^{k+1} v^k,$$

being $e^k \neq 0$, in general. From the compatibility condition (1) follows that

$$D_{k+1}^k e^k = (D_{k+1}^k - D_{k+1}^k P_k^{k+1} D_{k+1}^k) v^{k+1} = (D_{k+1}^k - D_{k+1}^k) v^{k+1} = 0,$$

that is, e^k belongs to the kernel[1] of D_{k+1}^k, i.e. $e^k \in \ker(D_{k+1}^k)$.

Let d^k be the set of coefficients that express e^k with respect to some basis of $\ker(D_{k+1}^k)$, which contains the non-redundant information of e^k . Denoting $e^k = E_k d^k$, It can be written

$$v^{k+1} = E_k d^k + P_k^{k+1} v^k.$$

Therefore, we have a bijection $v^{k+1} \leftrightarrow (v^k, d^k)$ which, if applied repeatedly, allows obtaining a *multi-scale decomposition* of v^L:

$$v^L \leftrightarrow (v^{L-1}, d^{L-1}) \leftrightarrow (v^{L-2}, d^{L-2}, d^{L-1}) \leftrightarrow \cdots \leftrightarrow (v^0, d^0, d^1, \ldots, d^{L-1}). \quad (3)$$

Therefore, formally the multiresolution representations proposed by Harten have the same structure as the standard wavelet transforms, so that the basic coding and decoding steps incorporated in (3) can be reinterpreted in terms of Electronic Engineering, as the steps of analysis and synthesis of a subband filtering scheme with exact reconstruction. The operator D_{k+1}^k would play the role of a low pass filter while from the operator $I_{V^{k+1}} - P_k^{k+1} D_{k+1}^k$ a high pass filter would be obtained.

Harten's point of view when introducing the HMRF is that the way in which the data have been generated determines its *nature* and must provide an adequate configuration to perform a multi-scale analysis of them. In practice, his proposal was based on the construction of D_{k+1}^k and P_k^{k+1} through two operators that link the discrete data with the functions from which they come: the *discretization* and the *reconstruction*. The discretization operator $\mathcal{D}_k$ is a linear operator that extracts discrete information from the functions of a certain space $\mathcal{F}$, $\mathcal{D}_k : \mathcal{F} \longrightarrow V_k$, at a resolution level specified by a Ξ^k mesh. The reconstruction operator $\mathcal{R}_k : V_k \longrightarrow \mathcal{F}$ generates an approximation to a given function $f \in \mathcal{F}$ from discrete values $\mathcal{D}_k f$. Between these operators the condition of *consistency*, or *compatibility*, must be met,

$$\mathcal{D}_k \mathcal{R}_k = I_{V^k}. \quad (4)$$

Given a sequence of discretization and reconstruction operators with the above characteristics, it is possible to define the decimation and prediction operators as follows:

$$D_{k+1}^k := \mathcal{D}_k \mathcal{R}_{k+1}, \qquad P_k^{k+1} := \mathcal{D}_{k+1} \mathcal{R}_k. \quad (5)$$

[1] The *kernel* of a linear application $A : V \longrightarrow W$ is the set of vectors in V whose image is 0,

$$\ker(A) := \{v \in V \ : \ Av = 0\}.$$

There seems to be an explicit dependency of D_{k+1}^k on $\mathcal{R}_k$, but it is easy to verify that the decimation is completely independent of reconstruction when the sequence of discretizations is *nested*, that is, if

$$\forall f \in \mathcal{F} \ : \ \mathcal{D}_{k+1} f = 0 \quad \Longrightarrow \quad \mathcal{D}_k f = 0.$$

In this case, the operator D_{k+1}^k is characterized by the following property

$$D_{k+1}^k(\mathcal{D}_{k+1} f) = \mathcal{D}_k f \qquad \forall f \in \mathcal{F}. \tag{6}$$

The discretization process chosen for each application is related to the nature of the data. In many applications, the data is associated with an underlying mesh, which can be considered as the finest level within a hierarchy of nested meshes. In the numerical solution of some ordinary and partial differential equations, for example, the discrete solution represents an approximation to the *point values* of the exact solution in a mesh. In other applications, such as the processing of medical images, the scanner has a fixed resolution and the information at coarser resolution levels must be obtained by *gathering* the data to simulate the effects of the scanner (and the image) at a coarser resolution. In this case, the data is *naturally* associated with the *cells* that define the underlying mesh.

Once the configuration is specified, the choice of an appropriate reconstruction operator provides the key step for the configuration of a multiresolution scheme. The reconstruction process is at the heart of a "a la Harten" multiresolution scheme [10, 11].

An advantage of the HMRF compared to other multi-scale frameworks is that the reconstruction can be nonlinear, giving rise to nonlinear P_k^{k+1} prediction operators or to P_k^{k+1} operators adapted to specific geometries, which translates in *synthesis* (or decoding) operators with these characteristics. The possibility of using nonlinear reconstruction operators, capable of obtaining precise representations of discontinuous functions, has been used successfully in applications involving data with strong gradients, as in the case of images [3, 5, 6, 7, 8, 29].

In the context of the papers collected in this summary, it is important to note that prediction operators can be understood as *recursive subdivision schemes*, which is a technique widely used in data refinement and in Computer Aided Design (CAD), to which Section 3 will be dedicated.

In particular, we have applied the HMRF in areas as diverse as Analytical Chemistry [C4, C5] (Section 5), the improvement of foil section designs in a context of multi-parametric optimization [C3] (Section 4.1) or the estimation of statistical parameters (*Uncertainty Quantification*, Section 3.3) [C2]. In these cases, we have mainly used the *point value* framework, which we introduce in Section 2.1. For a more complete revision of the HMRF, we recommend, for instance, [11, 56].

2.1 The point value framework

Reconstructing a function, from a discrete set of data representative of the function, is a classic problem in Approximation Theory that depends on the interpretation that is assigned to the discrete data. Probably the simplest case is the interpolation of *point values*, in which an attempt is made to reconstruct an approximation to an unknown function from a table of values of it. The reconstruction operators of the *interpolatory* or *point value* framework of the HMRF are based on the use of interpolation techniques. Next, this multiresolution framework will be described in the univariate case, which is the one used in the papers that constitute this doctoral thesis.

We assume that at the resolution level k the data is naturally associated with the mesh $\Xi^k = (\xi_i^k)_{i\in\mathbb{Z}}$, where

$$\xi_i^k = \xi_{2i}^{k+1}, \qquad \forall i \in \mathbb{Z}. \tag{7}$$

If the associated discretization operators are defined as

$$\mathcal{D}_k f := (f(\xi_i^k))_{i\in\mathbb{Z}},$$

a nested sequence is obtained, since $\Xi^k \subset \Xi^{k+1}$. From this sequence of discretization operators, from (6) it is easy to see that the decimation operator is

$$v^k := D_{k+1}^k v^{k+1} := (v_{2i}^{k+1})_{i\in\mathbb{Z}}.$$

That is, the decimation matches the *downsampling* operator of Filter Theory.

The prediction can be related to the reconstruction, as indicated in (5):

$$\begin{aligned}(P_k^{k+1}v^k)_{2i} &= (\mathcal{D}_{k+1}\mathcal{R}_k v^k)_{2i} = (\mathcal{R}_k v^k)(\xi_{2i}^{k+1}) = (\mathcal{R}_k v^k)(\xi_i^k),\\ (P_k^{k+1}v^k)_{2i+1} &= (\mathcal{D}_{k+1}\mathcal{R}_k v^k)_{2i+1} = (\mathcal{R}_k v^k)(\xi_{2i+1}^{k+1}).\end{aligned}$$

The consistency condition (1), $D_{k+1}^k P_k^{k+1} = I_{V^k}$, in this framework translates to

$$(P_k^{k+1}v^k)_{2i} = (D_{k+1}^k P_k^{k+1} v^k)_i = v_i^k, \qquad \forall i \in \mathbb{Z}, \tag{8}$$

that is, the prediction must keep all values of v^k in the even positions. So

$$f(\xi_i^k) = v_i^k = (\mathcal{R}_k v^k)(\xi_i^k),$$

that is, $\mathcal{R}_k$ interpolates the data v^k in the mesh Ξ^k. If the interpolation technique used to define the reconstruction operator is denoted by $\mathcal{I}(\xi, v^k)$, it can written

$$(\mathcal{R}_k v^k)(\xi) = \mathcal{I}(\xi, v^k).$$

On the one hand, the associated prediction error (2), e^k, is zero in the even positions as a consequence of (8),

$$e^k_{2i} = v^{k+1}_{2i} - (P^{k+1}_k D^k_{k+1} v^{k+1})_{2i} = v^k_i - (P^{k+1}_k v^k)_{2i} = 0, \qquad \forall i \in \mathbb{Z}.$$

On the other hand,

$$e^k_{2i+1} := v^{k+1}_{2i+1} - (P^{k+1}_k v^k)_{2i+1} = f(\xi^{k+1}_{2i+1}) - \mathcal{I}(\xi^{k+1}_{2i+1}, v^k), \qquad \forall i \in \mathbb{Z},$$

that is, in the odd positions the error is precisely the interpolation error of the interpolation technique.

These identities suggest defining the coefficients d^k (the non-redundant information of e^k) as the set of interpolation errors produced in the odd positions of the mesh, $d^k_i := e^k_{2i+1}$.

The bijection $v^{k+1} \leftrightarrow (v^k, d^k)$, which allows to transfer the information between resolution levels, takes the form

$$\begin{aligned} &v^{k+1}_{2i} = v^k_i,\ v^{k+1}_{2i+1} = d^k_i + \mathcal{I}(\xi^{k+1}_{2i+1}, v^k), \qquad \forall i \in \mathbb{Z}, \\ &\qquad\qquad \updownarrow \\ &v^k_i = v^{k+1}_{2i},\ d^k_i = v^{k+1}_{2i+1} - \mathcal{I}(\xi^{k+1}_{2i+1}, v^k), \qquad \forall i \in \mathbb{Z}. \end{aligned}$$

A classic example of reconstruction operator in this framework is the piecewise polynomial interpolation that follows: For each interval of the mesh, $[\xi^k_i, \xi^k_{i+1}]$, the $2q-1$ degree polynomial $Q_{i,k}$ such that $Q_{i,k}(\xi^k_{i+j}) = v^k_{i+j}$, $j = -q+1, \ldots, q$ is taken, and the reconstruction is defined as

$$(\mathcal{R}_k v^k)(\xi) := Q_{i,k}(\xi), \qquad \forall \xi \in [\xi^k_i, \xi^k_{i+1}]. \tag{9}$$

If $Q_{i,k}$ is expressed using the Lagrange basis,

$$Q_{i,k}(\xi) = \sum_{j=-q+1}^{q} v^k_{i+j} L_j(2^k \xi - i), \quad L_j(\xi) := \prod_{\substack{l=-q+1 \\ l \neq j}}^{q} \frac{\xi - l}{j - l},$$

the prediction operator that is obtained from this interpolation technique can be expressed as

$$(P^{k+1}_k v^k)_{2i+1} = Q_{i,k}(\xi^{k+1}_{2i+1}) = \sum_{j=-q+1}^{q} v^k_{i+j} a^q_j. \tag{10}$$

being $a^q_j := L_j\left(\frac{1}{2}\right)$, $j = -q+1, \ldots, q$, which depends on j and q but is independent of i and k. Its linear dependence on the data is clearly visible.

Classical examples of prediction operators are those based on polynomial interpolations of first and third degree ($q = 1, 2$):

$$(P_k^{k+1} v^k)_{2i+1} = \frac{1}{2} v_i^k + \frac{1}{2} v_{k+1}^k, \qquad (q = 1) \tag{11}$$

$$(P_k^{k+1} v^k)_{2i+1} = -\frac{1}{16} v_{i-1}^k + \frac{9}{16} v_i^k + \frac{9}{16} v_{i+1}^k - \frac{1}{16} v_{i+2}^k. \qquad (q = 2) \tag{12}$$

In some applications it is necessary to reconstruct functions from discrete data that contain sudden variations of magnitude, which may be associated with discontinuities of the underlying function. If a linear interpolation is used in these cases, undesirable oscillations could appear around the discontinuity (similar to the Gibbs phenomenon). There are nonlinear reconstruction operators specially designed to interpolate the data without producing oscillations, such as the reconstructions ENO [57], WENO [65], PCHIP [9] and PPH [4, 5].

3 Subdivision schemes

Subdivision schemes are a technique for recursive data refinement. The recursive subdivision stands out for its inherent simplicity, which has promoted its use as a reconstruction and approximation tool, particularly in the efficient generation of curves and surfaces in Computer Aided Design (CAD) [40].

A subdivision scheme [22, 43] is an iterative process that, from an initial data set f^0, calculates a sequence of data sets $(f^k)_{k\geq 0}$, associated with increasingly high levels of refinement. Each new data set, f^{k+1}, is defined from the previous one, f^k, by means of a finite set of 'simple' operations, which makes these processes very efficient tools in various applications.

The 'canonical' example will be used to introduce the most relevant concepts in this theory: the polygonal scheme[2]. In the *univariate* case, where the data are bi-infinite sequences $f^k = (f_i^k)_{i=-\infty}^{+\infty}$, the scheme consists of two subdivision *rules* that distinguish between odd and even positions:

$$f_{2i}^{k+1} := f_i^k, \quad f_{2i+1}^{k+1} := \frac{1}{2} f_i^k + \frac{1}{2} f_{i+1}^k, \qquad \forall i \in \mathbb{Z}. \tag{13}$$

The sequence of data f^k can be associated with the mesh $2^{-k}\mathbb{Z}$, which makes possible to understand the data as points $(i2^{-k}, f_i^k)$ in $\mathbb{R}^2$ and give a geometric interpretation of (13): In the $k+1$ iteration the points of the k iteration are preserved,

$$((2i)2^{-(k+1)}, f_{2i}^{k+1}) = (i2^{-k}, f_i^k),$$

and the average of each pair of consecutive points is added,

$$((2i+1)2^{-(k+1)}, f_{2i+1}^{k+1}) = \frac{1}{2}(i2^{-k}, f_i^k) + \frac{1}{2}((i+1)2^{-k} f_{i+1}^k).$$

In general, a subdivision scheme defines each new generated data by a set of simple operations that involve a finite amount of data from the previous iteration. This property, known as *locality*, implies that possible perturbations in the data are propagated in a controlled manner throughout the iterations [43]. That is, if an initial data is modified, the points affected by such variation are in a bounded region of the mesh, in this example, the open interval $(i-1, i+1)$ [43].

Note that all the points generated at the $k+1$ level belong to the polygonal of vertices $(i2^{-k}, f_i^k)_{i\in\mathbb{Z}}$. From this we deduce that the points of any iteration are on the initial polygonal, with vertices $(i, f_i^0)_{i\in\mathbb{Z}}$, and that the subdivision scheme (asymptotically) generates all the values of this polygonal function in the dyadic points, which is a dense set in the set of real numbers. In these cases, it is said

[2] A polygonal is a function piecewisely defined by first degree polynomials. The point that connects two straight segments is known as vertex.

that the subdivision scheme *converges* to a function F, known as *limit function*, which depends on the initial data.

The subdivision scheme (13) is an example of a *univariate* scheme, that is, the data on which it acts are sequences, $f^k = (f_i^k)_{i\in\mathbb{Z}}$. If the scheme converges, from an initial sequence $f^0 = (f_i^0)_{i\in\mathbb{Z}}$ a limit function of one variable is obtained (e.g. curves). In practice, it is only necessary to execute a finite amount of refinements (iterations) to 'generate' the function, for example to plot it in a specific application [43].

Multi-variate subdivision schemes manipulate data which are structured with multi-dimensional meshes, $f^k = (f_\alpha^k)_{\alpha\in\mathbb{Z}^s}$, $s > 1$, and can converge to functions of several variables. In this case, we can talk about convergence to surfaces and even to differentiable manifolds, as long as the initial data set and subdivision rules are adequate. This adds diversity to the type of situations in which the recursive subdivision is applied [5, 16, 58, 81].

In addition to convergence, another fundamental property of subdivision schemes is *stability*, which determines the magnitude of the changes in the limit function derived from perturbations in the initial data.

For schemes such as (13), where f_i^{k+1} depends *linearly* on the data in f^k, the convergence study is carried out in a systematic way by the corresponding theory [22, 43], which is well established and consolidated. In this case, stability is a consequence of convergence. When the subdivision rules are *nonlinear* [2, 5, 15, 29], completely different techniques are required and the underlying theory is much more recent [28, 36, 37, 42, 45, 51, 54, 61, 79]. The convergence and stability of these recursive processes are essential for their applications and, therefore, their study has been and continues to be an active research topic.

Note that the convergence, the locality and the stability of subdivision schemes has a positive impact on the generation and manipulation of geometric objects. In terms of the CAD, given a control polygon f^0, a subdivision scheme defines an associated curve. If a point in the control polygon is modified, the curve only changes in a well-delimited region, which surrounds the modified point. These properties allow a graphic designer to retouch and shape their models locally, without altering other parts that may already be to his/her liking, which is attractive in object modeling. In animation cinema, the recursive subdivision was first used in the Pixar short "Geri's Game" [40], in the late 1990s.

When the initial data comes from a smooth function, a fairly common requirement is that the recursive subdivision generates an *accurate enough* approximation of the original function. The accuracy, or the approximation capability, of the data generated by a subdivision scheme is an important factor to consider in many applications.

Another requirement that is useful in many applications is *reproduction*, that

is, the exact reconstruction of a family of functions. For example, the subdivision scheme (13) is able to *reproduce* polygonal functions.

The reproduction of polynomials [25, 33, 43] and exponential polynomials [26, 32, 34, 44, 78] is interesting from a theoretical point of view, since it is related to other properties of subdivision schemes (as the accuracy and the convergence), but also in practice, since it allows to draw accurately relevant curves in geometry, such as conic sections.

Linear subdivision schemes capable of reproducing exponential polynomials necessarily have subdivision rules that change throughout the iterations, and are therefore called *non-stationary*. In [C6] it is shown that stationary *nonlinear* schemes can also reproduce exponential polynomials and that they have some advantages over non-stationary linear schemes.

In other situations it may be important to establish certain restrictions on f^k and the limit function. For example, if the data represents a physical quantity that must have a positive real value, the new data generated by subdivision must also be positive. Similarly, the preservation of *monotonicity* or *convexity* may be required. The maintenance of some of these properties can be understood as particular cases of *shape preservation* [61], which has motivated the development of nonlinear schemes specifically designed to maintain one (or more) of these properties. Some examples that can be found in the literature are the *essentially non-oscillatory* schemes [28] (obtained from certain nonlinear prediction operators in the HMRF [57, 65]) or schemes that preserve the monotonicity [15, 62] or the convexity [5] in the data. The research carried out during this thesis project has resulted in two new subdivision schemes in this line [C1, C6].

Subdivision schemes can be considered the core of my activity during the doctoral studies. On the one hand, subdivision operators used as prediction operators within the HMRF have been applied in various contexts: Estimation of statistical parameters (Uncertainty Quantification) [C2], optimization of plane sections in the design of sailing yachts [C3] and analysis tools improvement in Analytical Chemistry [C4, C5]. On the other hand, we have theoretically developed and studied new nonlinear schemes with properties oriented to specific applications [C1, C6].

Since the articles that make up this doctoral thesis focus on the study and use of *univariate, uniform* and *binary* subdivision schemes, in Section 3.1 the main properties of the recursive subdivision will be defined in this context.

3.1 The univariate case

Definition 1. An *univariate subdivision scheme* is a sequence of operators $\{S^k\}_{k\geq 0}$, $S^k : \ell_\infty(\mathbb{Z}) \longrightarrow \ell_\infty(\mathbb{Z})$, which allows to recursively define a sequence of bounded sequences $(f^k)_{k\geq 0} \subset \ell_\infty(\mathbb{Z})$ from an initial sequence of bounded data $f^0 = (f_i^0)_{i\in\mathbb{Z}} \in$

$\ell_\infty(\mathbb{Z})$, as follows:

$$f^{k+1} := S^k f^k, \qquad k \geq 0.$$

The polygonal scheme (13), defined in Section 3, is an example of a *uniform*, *binary* and *univariate* scheme. Subdivision operators of this class are defined from two subdivision rules Ψ_0^k and Ψ_1^k that distinguish between odd and even data,

$$f_{2i}^{k+1} = \Psi_0^k(f_{i-q}^k, f_{i-q+1}^k, \ldots, f_{i+q}^k), \qquad f_{2i+1}^{k+1} = \Psi_1^k(f_{i-q}^k, f_{i-q+1}^k, \ldots, f_{i+q}^k),$$

for certain $q > 0$. If $\{\Psi_0^k, \Psi_1^k\}_{k\geq 0}$ are linear functions, then the scheme is *linear*. If the rules $\{\Psi_0^k\}_{k\geq 0}$ are such that $f_{2i}^{k+1} = f_i^k$, then it is *interpolatory*. If the subdivision rules $\{\Psi_0^k, \Psi_1^k\}_{k\geq 0}$ are the same throughout the iterations, i.e. do not depend on k, the subdivision scheme is *stationary*. In this case it will be denoted:

$$\Psi_0 := \Psi_0^k, \quad \Psi_1 := \Psi_1^k, \quad S := S^k.$$

The subdivision scheme (13) is stationary, linear and interpolatory.

The subdivision schemes used in practical applications must be *convergent*, a concept that is precisely defined below.

Definition 2. A subdivision scheme is *convergent* if

$$\forall f^0 \in \ell_\infty(\mathbb{Z}) \quad \exists S^\infty f^0 \in \mathcal{C}(\mathbb{R}) \; : \quad \lim_{k\to\infty} \sup_{i\in\mathbb{Z}} |f_i^k - (S^\infty f^0)(i2^{-k})| = 0.$$

The operator that sends each initial data f^0 to its corresponding limit function is denoted by $S^\infty : \ell_\infty(\mathbb{Z}) \longrightarrow \mathcal{C}(\mathbb{R})$.

It can be shown [43] that this definition of convergence is equivalent to the fact that the polygonal functions $\mathbb{P}^k$ such that $\mathbb{P}^k(i2^{-k}) = f_i^k$ form a Cauchy sequence.

Another important property is *stability*, which is formally defined as follows.

Definition 3. A convergent subdivision scheme is *stable* if the operator $S^\infty : \ell_\infty(\mathbb{Z}) \longrightarrow \mathcal{C}(\mathbb{R})$ is Lipschitz continuous:

$$\exists L > 0 \; : \; \|S^\infty f^0 - S^\infty g^0\|_\infty \leq L\|f^0 - g^0\|_\infty, \qquad \forall f^0, g^0 \in \ell_\infty(\mathbb{Z}).$$

For linear schemes, it is easy to see that stability is a direct consequence of the convergence of the scheme. However, the situation is very different in the nonlinear case.

In general, the subdivision schemes theory tries to infer properties of the limit functions $S^\infty f^0$ from the definition of the subdivision rules. In this way, it can be studied basic properties such as convergence or stability, and also others that may be suitable in various applications, such as regularity, approximation capability, exact reproduction, shape preservation, etc. All this from the expression of subdivision rules Ψ_j^k.

In some applications, such as in Computer Aided Design, it may be interesting that the curves generated by subdivision schemes have some regularity.

Definition 4. A convergent subdivision scheme is $\mathcal{C}^\alpha$ if[3]

$$S^\infty f^0 \in \mathcal{C}^\alpha, \qquad \forall f^0 \in \ell_\infty(\mathbb{Z}).$$

It is also often important to know the *approximation capability* of a subdivision scheme, in the sense of the following definition.

Definition 5. A convergent scheme has *approximation order* r if for any sufficiently smooth function F,

$$\|F(h\,\bullet) - S^\infty f^0\|_\infty \leq Ch^r, \qquad f^0 = F|_{h\mathbb{Z}}, \quad \forall 0 < h < h_0.$$

That is, the approximation order of a subdivision scheme measures how the error is reduced when trying to approximate F by applying the subdivision scheme on $(F(ih))_{i\in\mathbb{Z}}$, being the mesh spacing h small enough.

In addition, in some applications it is desired to exactly reconstruct certain functions $F \in \mathcal{F}$, and not approximately. The reconstruction of circumferences, ellipses, hyperbolas, etc. may be of practical interest and this can be done efficiently if subdivision schemes that *reproduce* the class of functions that define the previous curves are used [34].

Definition 6. A convergent subdivision scheme *reproduces* a family of functions $\mathcal{F}$, if for any function $F \in \mathcal{F}$ the scheme converges to F from the initial data $f^0 = F|_\mathbb{Z}$:

$$S^\infty F|_\mathbb{Z} = F, \qquad \forall F \in \mathcal{F}.$$

Deslauriers-Dubuc schemes [41] are a classic example of interpolatory linear schemes that reproduce polynomials of arbitrarily high (but fixed) degree. They can be constructed from the piecewise polynomial interpolation described in Section 2.1. As can be seen in (10), they are stationary schemes, because the coefficients a_j^q of the linear combination that define their rules are independent of k.

From a theoretical point of view, the reproduction of polynomials and exponential polynomials is interesting because it is related to the approximation capability and the smoothness of the scheme [31, 35, 43, 61]. In addition, a linear scheme that reproduces exponential polynomials is necessarily *non-stationary* [26, 32, 34, 44], and its subdivision rules Ψ_0^k, Ψ_1^k depend on certain parameters involved in the expression of the exponential polynomial space it reproduces. However, in [C6] a *nonlinear* subdivision scheme that reproduces trigonometric functions (which are a particular case of exponential polynomials) is obtained, whose rules are stationary and do not depend on the mentioned parameters.

[3] $\mathcal{C}^\alpha$ is defined as the set of functions α-times differentiable and continuous.

It is well known that linear and non-stationary subdivision schemes can achieve this goal [34, 44]. But its application requires the practical determination of the parameters, that define the level-dependent rules, by pre-processing the available data [17, 44].

Since different conic sections require different refinement rules to guarantee exact reproduction, it is not possible to reproduce with the same linear scheme a form composed, piecewisely, by several trigonometric functions. In [C6] it is shown that the exact reproduction of different conical shapes can be achieved using the same nonlinear scheme, without any prior data processing.

For applications where, due to the nature of the problem, the data is positive, or monotone, or convex, the subdivision scheme must preserve the type (or the *shape*) of the data.

Definition 7. A subdivision scheme (strictly) *preserves* the *positivity* of the data, or equivalently, is (strictly) positive, if for any $f \in \ell_\infty(\mathbb{Z})$,

$$f_i(>) \geq 0 \ \forall i \in \mathbb{Z} \quad \rightarrow \quad Sf_i > 0 \quad \forall i \in \mathbb{Z}.$$

A scheme (strictly) *preserves* the *monotonicity*, or it is monotone, if

$$\nabla f_i(>) \geq 0 \ \forall i \in \mathbb{Z} \quad \rightarrow \quad \nabla S f_i > 0 \ \forall i \in \mathbb{Z},$$

where $\nabla : \ell_\infty(\mathbb{Z}) \longrightarrow \ell_\infty(\mathbb{Z})$ is the finite differences operator, $\nabla f_i := f_{i+1} - f_i$.

A scheme (strictly) *preserves* the *convexity*, or it is convex, if

$$\nabla^2 f_i(>) \geq 0 \ \forall i \in \mathbb{Z} \quad \rightarrow \quad \nabla^2 S f_i > 0 \ \forall i \in \mathbb{Z}.$$

High accuracy interpolatory subdivision schemes that preserve the shape of the data are very useful in certain applications [5], which has motivated various authors to design nonlinear schemes with these properties [15, 62]. We have addressed this topic during the thesis project, having defined two new subdivision schemes [C1, C6].

The convergence of a linear subdivision scheme is a sufficient condition for stability, but it is not in the nonlinear case. It is more complex to prove that a nonlinear scheme is convergent and stable. Therefore, theoretical results are available [2, 5, 15, 37, 39, C1, 54] that ensure these properties if certain requirements are met. To introduce these results, which we have used in [C1, C6], it is necessary to define the concept of *difference scheme*. It is worth mentioning that nonlinear schemes are usually stationary, so results are limited to this case.

Definition 8. A subdivision scheme S has *difference scheme* of order n if there exists a scheme $S^{[n]}$ such that

$$\nabla^n S = S^{[n]} \nabla^n.$$

It should be noted that the existence of the difference scheme $S^{[n]}$ is not guaranteed, except for the linear case, where any convergent scheme has a difference scheme of order $n = 1$.

The difference scheme allows to analyze the behavior of the finite differences of the data f^k from $S^{[n]}$, by means of the expression

$$\nabla^n f^k = (S^{[n]})^k \nabla^n f^0.$$

This property is key in the convergence analysis, both in the linear and nonlinear cases.

In [2, 5, 15, 37, 39, C1, 54, C6], the authors construct and analyze various nonlinear subdivision schemes that can be described as a nonlinear perturbation of a linear convergent scheme T:

$$Sf = Tf + \mathcal{F}(\nabla^n f), \qquad \forall f \in \ell_\infty(\mathbb{Z}), \tag{14}$$

where $\mathcal{F} : \ell_\infty(\mathbb{Z}) \to \ell_\infty(\mathbb{Z})$ is a (possibly nonlinear) operator. If $T^{[n]}$ exists, which is easy to check [43], then a scheme of the form (14) has a difference scheme of order n. Specifically this is

$$S^{[n]} f = T^{[n]} f + \nabla^n \mathcal{F}(f), \qquad \forall f \in \ell_\infty(\mathbb{Z}). \tag{15}$$

Specific results of [2, 15] have been used to analyze their convergence and stability.

3.2 [Adv. Comput. Math., 2017] A family of non-oscillatory 6-point interpolatory subdivision schemes

A linear interpolatory scheme, with an approximation order $r > 2$, is sure to produce oscillations and lose all accuracy when the data present sudden variations. An example of this is shown in Figure 1. Using the 6-point Deslauriers-Dubuc (DD) scheme, $S_{3,3}$, which is linear and has order 6, a limit function that oscillates around the jump has been obtained. This means that around the discontinuity the scheme loses its capability to reconstruct the function.

Various piecewise polynomial interpolation techniques have been considered in the literature to construct interpolatory subdivision schemes that avoid undesirable oscillations. Examples of such schemes are the ENO-WENO [28, 57, 65], the PPH [4, 5], the Power$_p$ [2, 14, 36] and those shape preserving schemes described in [61]. The latter owe their non-oscillatory nature to the judicious use of certain nonlinear averages.

In this work, a new family of nonlinear subdivision schemes, the $S\mathrm{WH}_{p,q}$, are proposed and analyzed, which can be considered non-oscillatory versions of the $S_{3,3}$ scheme, like the Power$_p$ schemes are considered nonlinear and non-oscillatory

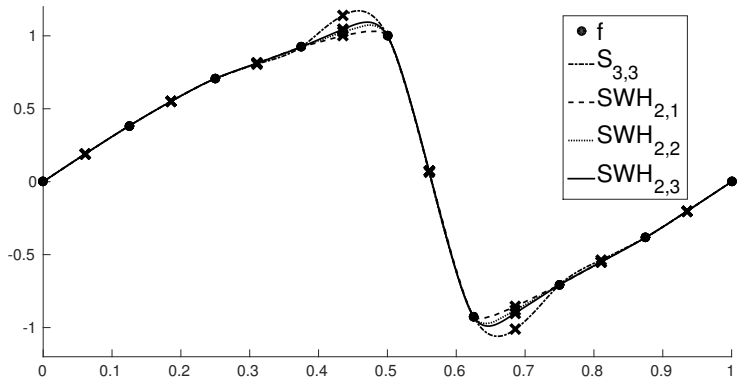


Fig. 1: From the initial data (•), limit functions are generated by various subdivision schemes. The crosses (×) are the data generated after an iteration.

versions of the 4-point interpolatory DD scheme. In fact, its design is related to that of Power$_p$ schemes.

It is shown that the new schemes exactly reproduce polynomials of third degree and that the distance in infinite norm to the 6-point DD scheme is small in smooth regions, as shown in Figure 1.

In addition, it is proved that the first and second difference scheme are well defined for each family member, which allows to give a simple proof of the uniform convergence of these schemes and also to study their stability as in [15, 54].

However, the theoretical study of stability based on the results of [54] is inconclusive in the case study. Therefore, a series of numerical experiments are carried out, that seem to indicate that only a few members of the new family of schemes are stable.

On the other hand, the exhaustive numerical tests reveal that, for smooth data, the order of approximation and the regularity of the limit function can be similar to those of the 6-point DD scheme and higher than those obtained with the Power$_p$ schemes.

3.3 [Applied Mathematics and Nonlinear Sciences, 2017] High-accuracy approximation of piecewise smooth functions using the truncation and encode approach

Numerous physical and industrial processes can be simulated by a partial differential equation (PDE). For example, in the design of appendages for sailboats, the flow of water around the profile must be simulated to calculate the drag coefficient. For this, the Navier-Stockes equations are used, whose numerical resolution

is complex and the calculation time is triggered by increasing accuracy.

It is possible that the simulation depends on multiple physical parameters whose value is variable, for example the speed of the sailboat and the inclination of the bow, and therefore should be treated as random variables. So, the drag coefficient is not unique: it depends on the value of each parameter. In practice, a mesh can be established and the associated PDE can be resolved for each pair of values of the speed-inclination mesh. As one might imagine, the computational cost is exorbitant if the mesh is very fine, and some strategy must be considered.

In [1, 49], a method called *Truncate and Encode* (TE) was defined, which takes advantage of Harten's Multi-Resolution Framework to adaptively approximate the solution of a PDE and estimate certain statistics parameters in the context of Uncertainty Quantification. Roughly speaking, at each resolution level it is decided whether to solve the PDE or interpolate the solution with the existing data, thus reducing the calculation time. The decision is based on the accuracy that the interpolation had at the previous level, and it is convenient to choose a high order approximation and preferably non-oscillatory interpolation technique. In fact, interpolation is equivalent here to a subdivision operator, so the PCHIP [15] and $S\mathrm{WH}_{p,q}$ [C1] schemes can be applied and are highly recommended.

In this paper, we analyze the TE algorithm applied to the approximation of functions and, in particular, its performance for piecewise smooth functions. Some numerical experiments are carried out, comparing the performance of the algorithm when different linear and nonlinear interpolation techniques are used. Some recommendations that are useful for achieving high algorithm performance are provided. The results indicate that in order to increase the TE performance it is convenient to use subdivision schemes with high approximation order.

4 Multi-scale strategies in large-scale optimization

Optimization [74] is an important tool in decision making and in the analysis of physical systems. In an optimization process, an *objective function* must be first identified, which measures the performance of the system being studied, for example time, energy, economic benefits, or any quantity or combination of them that can be represented with a single number. This function depends on certain system features, called *variables* or *parameters*.

The purpose of the process is to find the values of the parameters that optimize the objective function. Often the parameters are restricted, or limited, in some way. For example, the quantities that represent the mass of objects cannot be negative.

The process of identifying the objectives, variables and constraints of a given problem is known as *modeling*. Building an appropriate model is the first step, often the most important, in the optimization process. Once the model is obtained, the solution is found through the application of an optimization algorithm, usually with the assistance of a computer.

In mathematical terms, an optimization problem consists in minimizing (or maximizing) a *objective function* F within a space of possible *feasible* solutions, say X. That is, find $u_{\min} \in X$ such that $F(u_{\min}) \leq F(u)\ \forall u \in X$.

The objective functions must be defined in a finite dimensional space, i.e. $F : X \subset \mathbb{R}^N \longrightarrow \mathbb{R}$, to be able to address the optimization problem computationally using some algorithm, called *optimizer*. When the number of variables N is large, it is said to be a *large-scale optimization*. This type of problem often appears from the discretization of an infinite dimensional problem, for example in the context of optimal design, optimal control, estimation of parameters in systems governed by PDEs [19, 64, 80] and image processing [24, 23, 76, 75, 82].

There is no universal optimizer, rather a whole collection of them, each of them tailored for some kind of problem. The responsibility for choosing the algorithm appropriately for a specific application lies with the user. This decision is important, because it will determine if the problem is resolved quickly or slowly and, certainly, if the solution will be found.

In large-scale optimization, it can often be applied optimizers that involve a prohibitive computational effort due to the large number of variables involved.

The success of the multigrid [20, 21, 52, 53, 69] methods, as an efficient solver of discretized elliptical PDEs, boosted the development of multi-level iterative methods in optimization [18, 27, 30, 47, 48, 50, 60, 72] since the mid-80s.

The idea that these multi-level methods share is the application of a particular optimizer to solve reduced auxiliary problems of smaller dimensions, derived from the discretization of the infinite dimensional problem with less accuracy, and therefore are faster to solve (in terms of calculation).

Although multi-level methods share a common structure, an effort has been made to develop separately the multi-level versions of the most common optimizers [30, 47, 48, 50]. In this doctoral thesis a multi-level structure based on the HMRF is proposed, that allows any optimizer to be implemented arbitrarily. In other words, this method allows the optimizer to be treated as a 'black box', allowing the user to use the more suitable optimizer among those available.

My work in this field arises from an internship in IS&3D ENG[4], during my master studies. The proposed problem was to improve the performance of plane sections that are used in the design of appendages of racing sailboats. In particular, we wanted to reduce the drag[5] of rudders, keels and bulbs in water. The challenge was formulated as an optimization problem in which the theoretical drag was wanted to be reduced taking into account certain physical and design constraints.

The objective function included a CFD simulation performed by an external black box routine (`xfoil`[6]). IS&3D ENG proposed to use the optimizers available in Matlab.

During the optimization process, the section had to be modified by smooth perturbations and without oscillations, introducing firstly global variations to the section so that, as it improved, focusing in the more local details. This idea fitted in the multi-scale structure of the HMRF. The synthesis of this approach led us to define a new optimization strategy.

Part of the work done during the collaboration with IS&3D ENG has given rise to various appearances in scientific meetings (see Curriculum Vitae) and to the publication included in the compilation of articles [C3, Section 4.1].

Through the collaboration with the research team FUSCHROM[7] of Liquid Chromatography[8], this optimization strategy was successfully applied in a completely different context [C4, Section 5.1]. Here, the objective was to maximize the separation between substances that have been injected into a separation device. Again, the objective function was quite complex, since it contains a chemical simulation, and Matlab's routine `patternsearch` was used as the optimizer.

[4] www.is3de.com

[5] The resistance to movement of an object through a *flow*, such as air or water.

[6] XFOIL is an interactive program for the design and analysis of subsonic isolated aerodynamic profiles.

[7] https://sites.google.com/site/fuschrom/

[8] A technique that allows to separate, identify and quantify each substance present in a mixture.

4.1 [Progress in Industrial Mathematics at ECMI 2016] A novel multi-scale strategy for multi-parametric optimization

The movement of a sailboat is a consequence of the balance between the aerodynamic forces, induced by the wind on the sails, and the hydrodynamic forces, resulting from the contact of the water with the submerged parts of the ship, which are the hull and the appendages. Each appendix fulfills a function. For example, the rudder marks the direction of movement, the keel avoids lateral displacements and the bulb influences the righting moment[9], preventing the boat from heeling.

The modeling of these appendices is made from their cross section, a closed plane curve called *profile*, like the one shown in Figure 2. The objective addressed in this work [C3] is to modify (optimize) a given profile to reduce the drag in the water, subject to restrictions of various kinds. There are structural ones, such as that the profile must have a certain length, and there are physical ones, such as that the lifting coefficient must be between two admissible values. The restrictions will depend on the purpose of the appendix (rudder, keel ...). Seen in another way, the restrictions imposed on the optimization make any initial profile the desired type of appendix.

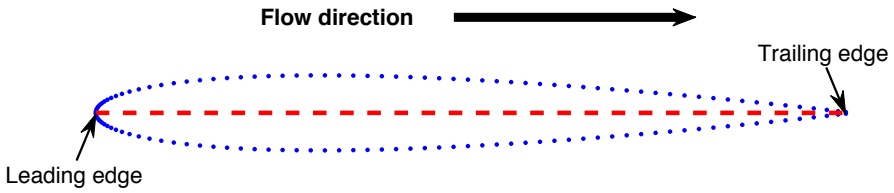


Fig. 2: An example of profile: The NACA0010 described by $N = 129$ points.

The proposed strategy provides a sequence of sub-optimal solutions, one at each level of resolution, so that in the last step the complete optimization problem is solved, but with an initial guess much closer to the desired solution than the initially provided one (which is often chosen arbitrarily, but can also be provided by the user), making feasible the calculation effort required by the chosen optimizer.

This technique exhaustively applies a subdivision scheme, which must be chosen taking into account the nature of the manipulated data. Since a profile is wanted to be *smoothly* modify, but avoiding producing oscillations, it is proposed to use the B-Splines subdivision scheme of order 5 [43].

The paper analyzes the behavior of the algorithm by applying it to an academic problem, obtaining a drastic reduction in computational cost compared to the direct application (without multi-scale strategy) of the chosen optimizer.

[9] The righting moment measures the ability of a boat to stay in an upright position.

An optimization is proposed for the design of an appendix, where only the multi-scale strategy was able to provide satisfactory results.

5 Applications in liquid chromatography

Liquid chromatography is a technique used in Analytical Chemistry to separate, identify and quantify each of the solutes present in a mixture.

By inserting the mixture together with a solvent along a tube, called *column*, the different solutes of the mixture flow (*precipitate*) at different speeds when they interact with the porous medium inside the column.

If the experiment is configured correctly, each solute leaves (*elutes*) the end of the column separately. A sensor records at each instant of time the amount of eluting mixture, and the graph obtained with this time/amount ratio[10] is called *chromatogram*. The different solutes of the mixture appear as 'peaks' in the chromatogram, as illustrated in Figures 3 and 4.

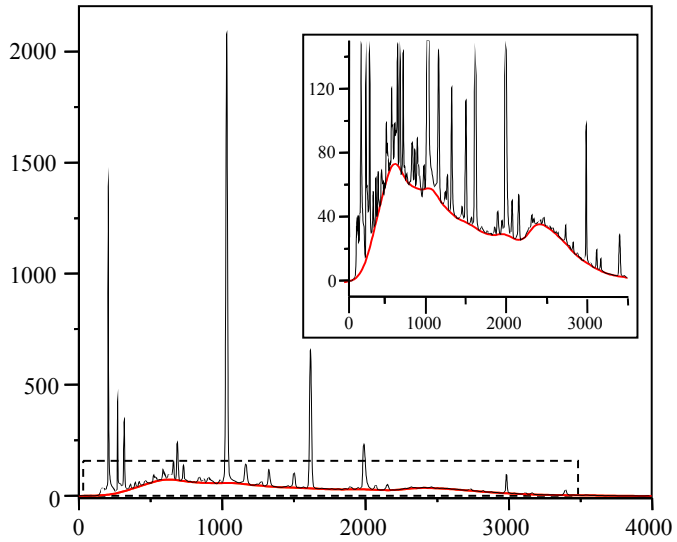


Fig. 3: Example of baseline detection. The zoom of the area marked with a dashed rectangle is shown in the upper right box.

Some problems that can be found to correctly quantify the amount of each solute in the mixture, and which we have addressed during the PhD through mathematical techniques, are: the presence of a baseline, which is derived from the use of the solvent; the presence of *noise*, which comes from both environmental and specific factors of the mixture; and the overlapping of some peaks with others. To avoid this last problem, and thus obtain well *resolved* peaks, it is necessary to

[10] The measure unit is provided by the device that measures how much light has not been absorbed by the mixture when leaving the column. If, for example, the light receiver is electronic, the unit of measurement would be millivolts.

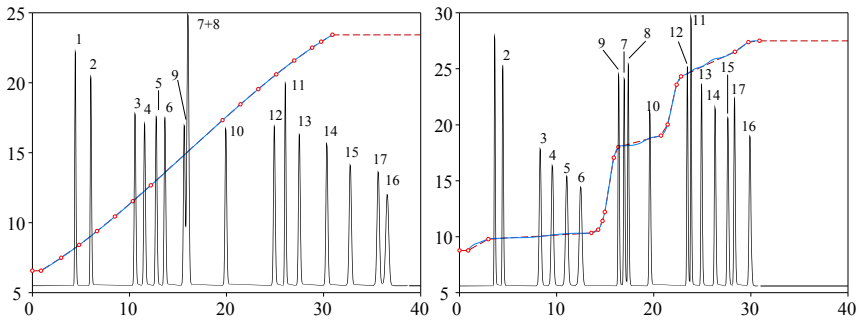


Fig. 4: Increased resolution of the peaks, associated with 17 essential amino acids, and reduced elution time. On the left, the initial gradient program. On the right, the optimized one. On the horizontal axis the elution time (in minutes) is shown and, on the vertical, the concentration of the solvent injected into the column.

preset the concentration of solvent to be injected into the column in each instant of time.

The collaboration with Analytical Chemistry projects began during my undergraduate studies. I collaborated with the CLECEM[11] research team, in the bachelor subject 'internship in company' under the supervision of Guillermo Ramis Ramos. As a result of this collaboration, we published an article [C8] (previous to my doctoral studies) on the analysis and classification of chromatograms.

A collaboration began with the FUSCHROM[12] research group, also of Analytical Chemistry, at the beginning of my PhD. They were interested in the postprocessing of chromatographic signals. In particular, certain *baselines* were wanted to be eliminated, which are usually present in the data, by some mathematical algorithm implemented computationally.

Through the contacts of my thesis supervisor, Rosa Donat, we met an algorithm, BEADS, which was providing excellent results. It is based on the optimization of an objective function, carefully designed, using the *majorization-minimization* approach [46, 63]. It should be said that BEADS is also prepared to *denoise*.

Applying BEADS to different chromatograms, some limitations and difficulties associated with its use appeared. A procedure was proposed to correctly and easily apply this algorithm in [C7] (not included in the compilation of articles). An example of baseline detection is shown in Figure 3 using this procedure. Once detected, it is only necessary to subtract it.

[11] https://www.uv.es/clecem/
[12] https://sites.google.com/site/fuschrom/

Afterwards, a new research was raised: find a new way to design, efficiently, *gradient programs*. Mathematically speaking, it consists in finding a function that maximizes the *resolution* of the peaks.

The procedure used so far was to consider an arbitrary polygonal function with a fixed number of vertices. By using some optimizer, e.g. a genetic algorithm, the optimal position of the vertices is determined.

That strategy requires a lot of calculation time and a very limited number of nodes. As it is a large-scale problem, the use of the multi-scale optimization strategy of Section 4 was proposed, which resulted from the collaboration with the company IS&3D ENG. Very satisfactory results were obtained, as set out in the following article [C4, Section 5.1].

5.1 [J. Chromatogr. A, 2018] Gradient design for liquid chromatography using multi-scale optimization

The design of *gradient programs*, where the concentration of solvent, that must be introduced in the column in each instant of time, is specified to the chromatographic machine, is essential to obtain well resolved peaks, without overlaps, and thus be able to correctly measure the amount of each solute that forms the mixture. The objective is to find the gradient program function that maximizes the resolution while verifying a series of conditions, necessary for its correct implementation in the laboratory.

The multi-scale optimization [C3] has been successfully applied to this problem, achieving not only a high resolution, but also the reduction of elution time, which translates into less work hours for the instruments and for the staff of the laboratory. An example of optimization is shown in Figure 4.

5.2 [J. Chromatogr. A, 2019] Enhancement in the computation of gradient retention times in liquid chromatography using root-finding methods

In this work [C5], a new method is proposed for the simulation of the peaks positions depending on the gradient program. These types of simulations are necessary to carry out studies such as the previous one [C4, Section 5.1].

The t value that solves the integral equation

$$f(t) = \int_0^{g(t)} h(\tau) d\tau,$$

for certain functions f, g, h defined from the conditions of the experiment, represents the position on the abscissa axis in which a certain peak of the chromatogram

appears, or in chemical terms, the time it takes for a solute to exit the chromatographic column.

The approach used so far was to discretize the integral as follows,

$$\int_0^{g(n\delta)} h(t)dt \approx \delta \sum_{i=0}^{n-1} h(g(i\delta)), \qquad n \in \mathbb{N}$$

typically being $\delta = 10^{-3}$, and to find the value of n such that the previous sum was as close as possible to $f(n\delta)$. Then, it is deduced that $t \approx n\delta$.

First, the approximation to the integral that was being used was very poor, since it was based on approximating h through step functions, when in addition, in some cases, h had primitive. As an improvement, it is proposed to use the primitive to greatly reduce the computational cost and, if there is no primitive, approximate h by some polynomial and use the polynomial primitive.

Second, the way in which the n value was found was very rudimentary. Only its value was increased until a certain stop condition was verified. Having an (approximate) primitive of h, say H, the numerical resolution of the integral equation can be understood as finding the zero of the function

$$F(t) = f(t) - H(g(t)) + H(0).$$

My proposal was to apply a root-finding method, derived from Newton's method and the bisection method, which combined the convergence speed with the convergence guarantees of both algorithms.

This new approach allows not only to compute the retention time much faster, but also to increase the accuracy, which was now less than 10^{-3}. In this work, a theoretical analysis is also done to ensure that the retention time approximations are calculated with an error below a chosen threshold.

6 Conclusions, work in progress and future perspectives

In this PhD thesis, different subdivision schemes have been proposed, studied and analyzed, paying special attention to the nonlinear case.

Nonlinear subdivision schemes can avoid some of the limitations that linear schemes present in certain applications. In this PhD thesis, a nonlinear non-oscillatory interpolatory scheme with high order of approximation and reproducing polynomials of up to third degree has been obtained [C1].

In addition, several applications have been considered in which subdivision schemes played a relevant role through the Harten's Multi-Resolution Framework.

We have researched the use of nonlinear prediction (subdivision) operators in Uncertainty Quantification, implemented in the *Truncate and Encode* strategy [1, C2].

We have proposed a new optimization strategy based on the Harten's Multi-Resolution framework, which has been applied in the design of plane sections of certain appendages of racing sailboats to reduce the drag in the water, in order to improve their efficiency [C3].

This optimization strategy has also been used in problems related to the signal processing in Liquid Chromatography. We have proposed a method for the design of *gradient programs* [C4]. As a consequence of this work, the importance of efficiently simulating the *elution time* was emphasized. In [C5] we have proposed a new method that drastically reduces the time needed to design a gradient program, while increasing the accuracy of the results.

The publications [C1, C2, C3, C4, C5] actually represent the consolidated part of my research work. In addition to these publications, the following three articles (submitted for publication) must also mentioned.

1- Nonlinear stationary subdivision schemes that reproduce trigonometric functions. *R. Donat and S. López-Ureña*

As stated in Section 3.1, the work carried out in this doctoral thesis has allowed us to design a new family of nonlinear interpolatory subdivision schemes with the capability to reproduce trigonometric functions and second degree polynomials. Obviously, this property is interesting for the CAD, because the schemes can reproduce shapes piecewisely defined by conic sections (circumferences, hyperbolas, ellipses and parabolas). The paper has been submitted for publication, and after receiving the comments of the reviewers and making the necessary modifications, we are waiting for a definitive answer for publication. It is currently available in arXiv [C6].

2- A Multiresolution approach to solve large-scale optimization problems. *R. Donat and S. López-Ureña*

This article formalizes the multi-scale optimization strategy outlined in [C3, Section 4], and compares it to other multi-level optimization methods. Through various numerical experiments, both one-dimensional and two-dimensional, its performance is studied and the impact of the prediction operator, chosen to define the multiresolution framework, is analyzed. It is concluded that it is convenient to use, as prediction operators, subdivision schemes with high order of approximation. The work has been submitted for publication.

3- Multi-scale optimisation vs. genetic algorithms in the separation of diuretics by reversed-phase liquid chromatography . *T. Álvarez-Segura, S. López-Ureña, J.R. Torres-Lapasió and M.C. García-Alvarez-Coque*

In [C4, Section 5.1] it is clear that the previous optimization strategy, based on the HMRF, can be applied in the design of *gradient programs*. This work compares its performance with another method, based on genetic algorithms, which is known to provide good results in this problem. It is concluded that the multi-objective approach of genetic algorithms is very convenient, since it gives the user some freedom to decide which gradient program is more suitable. Consequently, it might be very convenient to use genetic algorithms as an optimizer within the multi-scale strategy. This is a possibility considered in [C3, Section 4], and this issue is reserved for the future. We have modified the article according to the indications of the reviewers and we are waiting for a final decision on its publication.

On the other hand, the results of the research lines, which were opened during my stays in Italy and Germany, are still being drafted.

The reproduction of *exponential polynomials*, which generalize trigonometric functions, in a multi-variate context was studied in a stay with Professor Tomas Sauer (U. Passau). In addition, motivated by the nonlinear, highly accurate and non-oscillatory scheme [C1], in this stay a subdivision scheme of the same type was also designed, but in an *tri-variate* environment for the refinement of voxel tomographic data.

During the stay with Professors Costanza Conti (U. Firenze) and Lucia Romani (U. Milano-Bicocca), a question that naturally arose was whether the ideas involved in [C6] can be extended to a multi-variate context. We have defined a bivariate scheme that reproduces trigonometric surfaces, and therefore can be used to draw spheres, ellipsoids, hyperboloids and paraboloids, or any composition by parts of them thanks to the locality of subdivision schemes.

The collaboration initiated with professors C. Conti and L. Romani can continue in several ways. On the one hand, the underlying ideas of the subdivision scheme [C6] can be generalized for the reproduction of exponential polynomials, and this would bring benefits in certain applications. On the other hand, a limitation that reproducing subdivision schemes present (in general) is the need for

data to come from a known underlying mesh. We think that some ideas of [C6] can be used to define reproducing schemes that do not require prior knowledge of the mesh.

The work done during this PhD thesis reinforces the relevance of mathematics in other fields, scientific or not. The collaboration with the Analytical Chemistry research team (FUSCHROM) has been very beneficial to both parties, and we have several research proposals for the future. It should be highlighted that the team has recently acquired a new chromatographic machine that generates bi-dimensional data, which can be understood as images. FUSCHROM is very interested in developing new methods and algorithms for the analysis and processing of this type of signals, which will allow to extract more information from the laboratory samples.

Herramientas matemáticas multi-escala para el procesamiento de señales

Resumen

Sergio López Ureña
Directora: Rosa Donat

VNIVERSITAT DE VALÈNCIA

Doctorado en Matemáticas
Julio de 2019, Burjassot (Valencia)

Agradecimientos

No quisiera dejar pasar esta oportunidad para agradecer a aquellas personas que han hecho de mis estudios de doctorado una experiencia increíble, en todos sus aspectos, y que recordaré con cariño toda la vida.

En primer lugar, quiero expresar mi gratitud a Rosa Donat. Durante estos años me ha enseñado que tan importante es desarrollar nuevo conocimiento matemático como saber transmitir los resultados a otras personas. Admiro su empeño por buscar la mejor manera de expresar las ideas que queremos transmitir.

A los profesores Mª Celia García y José Ramón Torres, del departamento de Química Analítica, por las investigaciones tan interesantes que proponen y que hemos disfrutado codo con codo. Con vosotros he aprendido que cada uno venimos de una escuela, con nuestras propias metodologías y conocimientos, y que en un equipo con voluntad todas ellas suman.

A mis compañeros doctorandos, casi todos doctores ya, por el día a día, por recordarme constantemente que el doctorado es mucho más que escribir una tesis.

A las profesoras Costanza Conti (U. Firenze) y Lucia Romani (U. Milano-Bicocca), de las que admiro su relación colaborativa y perseverante, por mantener durante tres meses una rutina constante de trabajo conmigo, y que dieron lugar a una investigación que me apasionó.

Al profesor Tomas Sauer y a mis compañeros del departamento de FORWISS, por acogerme en la Universidad de Passau, tratándome como uno más del equipo, y compartir conmigo la cultura alemana. Me fascina vuestro ambiente de trabajo y la relación que mantenéis con los demás compañeros.

A todos los trabajadores del departamento, de la facultad y de la Universidad de Valencia, por crear un ambiente de trabajo agradable y motivador.

Al Ministerio de Ciencia, Innovación y Universidades, por su soporte económico con la ayuda FPU14/02216.

A mi familia y a mis amigos, por su apoyo y su confianza, por ayudarme a ver mi situación con perspectiva y a aprender a apreciarla. En especial a mis padres y mi hermano, por escucharme y entenderme siempre.

Y por su puesto, a Lydia, mil gracias por acompañarme a diario en esta aventura. Gracias a ti termino el doctorado con la sensación de que esto no ha hecho más que empezar.

Índice

1. Introducción: El procesamiento de señales

Se entiende por *señal* una función que contiene información sobre el comportamiento (o los atributos) de un sistema o fenómeno [77]. Grabaciones de voz, electrocardiogramas, fotografías, cromatogramas, mapas meteorológicos, etc. son ejemplos de señales.

Las señales suelen contener grandes cantidades de datos delos que la información relevante para una aplicación concreta puede ser difícil de extraer. Mediante el *procesamiento* se obtiene una señal de salida, a partir de una señal de entrada, con el fin de extraer información o de modificar determinadas características. Por ejemplo, a la grabación de una pieza musical se le puede aplicar la transformada de Fourier [66] para convertir la señal original tiempo-intensidad en otra frecuencia-intensidad. A un electrocardiograma, se le pueden borrar las interferencias que presente [73], procedentes quizás de los instrumentos médicos cercanos.

El procesamiento suele ser más rápido y simple en una representación *dispersa* de la señal, donde unos pocos coeficientes revelan la información más característica o representativa. Dichas representaciones pueden construirse descomponiendo las señales utilizando formas de onda elementales elegidas en una familia determinada.

Las series de Fourier son un ejemplo clásico de representación de funciones, donde la familia asociada es $\{\exp(\imath nt)\}_{n\in\mathbb{N}}$. Si una función es $\mathcal{C}^\alpha$, es decir, es $\alpha \in \mathbb{N}$ veces diferenciable y continua, entonces la sucesión de sus coeficientes de Fourier, $\{c_n\}_{n\geq 0}$, decrece con velocidad $O(n^{-\alpha})$. Por lo tanto, para funciones suaves los coeficientes de la serie decaen rápidamente y, si se ignoran aquellos coeficientes con un valor inferior a un cierto umbral, se consigue una representación *dispersa*, basada en unos pocos términos de la serie de Fourier. Sin embargo, para funciones con discontinuidades, los coeficientes apenas menguan a ritmo $O(n^{-1})$.

Las técnicas de procesado de señales basadas en la descomposición de Fourier se han convertido en herramientas básicas en una gran variedad de aplicaciones en muchos campos de la ciencia.

A pesar de la capacidad del análisis de Fourier para representar funciones suaves, se trata de una descomposición global. Una singularidad aislada domina el comportamiento de todos los coeficientes en la descomposición de una función discontinua. Las aproximaciones de funciones basadas en el truncamiento de la serie de Fourier presentan el llamado *Fenómeno de Gibbs*, que consiste en la aparición de oscilaciones alrededor de la discontinuidad, y que no desaparecen por muchos términos que se añadan a la serie truncada. El fenómeno de Gibbs hace que la transformada de Fourier deje de ser una herramienta útil en muchos contextos. En particular, en el tratamiento de imágenes se manifiesta como un artefacto visualmente identificable alrededor de los contornos de los objetos (que se pueden interpretar como discontinuidades en la señal)[59].

En lenguajes naturales, una familia de palabras (diccionario) más rica ayuda a

construir oraciones más cortas y más precisas. De manera similar, son necesarias familias adecuadas para construir representaciones dispersas de señales complejas. Una "buena" representación puede mejorar el reconocimiento de patrones, la compresión de datos o la reducción de ruido. El descubrimiento de familias ortogonales de funciones locales en tiempo-frecuencia [38, 67, 71], entre ellas las bases de wavelets ortogonales, abrió las puerta a otro tipo de transformaciones capaces de obtener representaciones 'locales' tanto en espacio como en frecuencia. Mallat utilizó las bases ortonormales de wavelets como herramienta matemática para describir el 'incremento de información' entre diferentes 'niveles de resolución' en una descomposición multi-escala de una imagen. Este tipo de descomposiciones se obtenían a partir de *esquemas de filtrado*.

En un esquema típico de filtrado a dos bandas, la señal de entrada es convolucionada separadamente con dos filtros diferentes, uno de paso bajo y otro de paso alto. El filtro paso bajo desecha la parte altamente oscilatoria de los datos, dejando 'pasar' únicamente la parte de baja frecuencia. Mientras que un filtro paso alto extrae la parte de alta frecuencia de la señal. Una vez obtenidas las dos sucesiones resultantes de las convoluciones, ambas son *sub-muestreadas* (*downsampled*) para retener los elementos pares (o impares) y desechar el resto. Al proceso de obtener las dos nuevas señales a través de los operadores de convolución y sub-muestro se le conoce como *análisis* o *codificación*. A partir de ambas contribuciones puede reconstruirse la señal original, o al menos una aproximación a ella, siendo necesario aplicar un *sobre-muestreo* (*upsampling*) y dos nuevos filtros, adecuados y acordes al sub-muestreo y a los filtros iniciales. Estas operaciones constituyen el proceso de *síntesis* o *decodificación*.

A partir de 1986, Meyer y Mallat desarrollaron las bases de los *Análisis de Multi-resolución*, y las transformaciones multi-escala asociadas. Una representación multi-escala de una señal (discreta) se compone de una aproximación a baja resolución de la señal original más una sucesión de 'detalles', que son la diferencia de información entre niveles de resolución consecutivos, de lo que rápidamente se estableció la relación entre estas herramientas matemáticas y los esquemas de filtrado largamente utilizados en Ingenieria Electrónica para el procesamiento de señales [38].

En los artículos que componen esta tesis doctoral se utiliza el entorno de multi-resolución diseñado por A. Harten a finales de los años 80. En la siguiente sección describimos con cierto detalle este entorno, que en cierto sentido se puede considerar una generalización de los análisis de multi-resolución basados en la teoría de wavelets.

2. El entorno de multi-resolución de Harten

El desarrollo de la teoría de wavelets [68, 67, 70, 71] se puede considerar como el punto de partida de las descomposiciones locales por escalas, que han tenido sin duda un gran impacto en diversos campos de la ciencia.

La construcción de las *bases de wavelets* se apoyan en funciones que resultan del desplazamiento y la dilatación de una única función 'madre'. Inicialmente, el diseño y análisis utilizaba de manera intensiva técnicas de Análisis Armónico, que hacían difícil la extensión a dominios delimitados y geometrías generales.

En [55, 56], A. Harten desarrolla un entorno general de multi-resolución para la representación de datos (HMRF, del inglés *Harten's Multi-Resolution Framework*), que se apoya en la Teoría de la Aproximación, permitiendo una mejor adaptación a todo tipo de geometrías.

El HMRF [10, 11, 12, 13, 56] se sustenta en dos *operadores*, la *decimación* y la *predicción*, que relacionan datos discretos asociados a dos niveles de resolución consecutivos. Desde un punto de vista algebraico, la decimación y la predicción se pueden considerar simplemente como operadores que conectan espacios vectoriales lineales de dimensión numerable, V^k, que representan de alguna manera los diferentes niveles de resolución de los datos (discretos) que se pretenden analizar (la resolución se incrementa con k), es decir,

$$D_{k+1}^k : V^{k+1} \longrightarrow V^k, \quad P_k^{k+1} : V^k \longrightarrow V^{k+1}.$$

Mientras que la decimación D_{k+1}^k se asume lineal, no hay restricciones *a priori* en el HMRF para que la predicción P_k^{k+1} lo sea. La única restricción entre ambos operadores en este entorno es la *consistencia*, esto es

$$D_{k+1}^k P_k^{k+1} = I_{V^k}, \tag{1}$$

donde I_{V^k} es el operador identidad en V^k.

Si en el nivel más fino se tienen los datos $v^L \in V^L$, mediante la decimación pueden restringirse los datos a espacios más toscos (gruesos), definiendo recursivamente

$$v^k := D_{k+1}^k v^{k+1}, \qquad k = 0, \ldots, L-1.$$

La predicción realiza el proceso opuesto, es decir, genera nuevos datos en un espacio más amplio (fino). El error asociado al uso de un operador de predicción se mide mediante la expresión

$$e^k := v^{k+1} - P_k^{k+1} D_{k+1}^k v^{k+1} = (I_{V_{k+1}} - P_k^{k+1} D_{k+1}^k) v^{k+1}. \tag{2}$$

Nótese que v^{k+1} puede calcularse a partir de v^k y e^k:

$$v^{k+1} = e^k + P_k^{k+1} v^k,$$

siendo $e^k \neq 0$, en general. De la condición de compatibilidad (1), se deduce que

$$D_{k+1}^k e^k = (D_{k+1}^k - D_{k+1}^k P_k^{k+1} D_{k+1}^k) v^{k+1} = (D_{k+1}^k - D_{k+1}^k) v^{k+1} = 0,$$

es decir, e^k pertenece al núcleo[1] de D_{k+1}^k, i.e. $e^k \in \ker(D_{k+1}^k)$.

Sea d^k el conjunto de coeficientes que expresan e^k respecto de alguna base de $\ker(D_{k+1}^k)$, que contiene la información no-redundante de e^k. Denotando $e^k = E_k d^k$, se puede escribir

$$v^{k+1} = E_k d^k + P_k^{k+1} v^k.$$

Por lo tanto, se tiene una biyección $v^{k+1} \leftrightarrow (v^k, d^k)$ que, si se aplica reiteradamente, permite obtener una *descomposición multi-escala* de v^L:

$$v^L \leftrightarrow (v^{L-1}, d^{L-1}) \leftrightarrow (v^{L-2}, d^{L-2}, d^{L-1}) \leftrightarrow \cdots \leftrightarrow (v^0, d^0, d^1, \ldots, d^{L-1}). \quad (3)$$

Por tanto, formalmente las representaciones de multi-resolución propuestas por Harten tienen la misma estructura que las transformadas de wavelet estándar, de manera que los pasos básicos de codificación y decodificación incorporados en (3) se pueden re-interpretar en términos de Ingeniería Electrónica, como los pasos de análisis y síntesis de un esquema de filtrado por bandas con reconstrucción exacta. El operador D_{k+1}^k desempeñaría el papel de un filtro de paso bajo y del operador $I_{V^{k+1}} - P_k^{k+1} D_{k+1}^k$, se obtendría un filtro de paso de alto.

El punto de vista de Harten al introducir el HMRF es que la forma en que se han generado los datos determina su *naturaleza* y debe proporcionar una configuración adecuada para efectuar un análisis multi-escala de los mismos. En la práctica, su propuesta se basaba en la construcción de D_{k+1}^k y P_k^{k+1} a través de dos operadores que relacionan los datos discretos con las funciones de las que provienen: la *discretización* y la *reconstrucción*. El operador de discretización $\mathcal{D}_k$ es un operador lineal que extrae información discreta de las funciones de un cierto espacio $\mathcal{F}$, $\mathcal{D}_k : \mathcal{F} \longrightarrow V_k$, en un nivel de resolución especificado por una malla Ξ^k. El operador de reconstrucción $\mathcal{R}_k : V_k \longrightarrow \mathcal{F}$ genera una aproximación a una función dada $f \in \mathcal{F}$ a partir de los valores discretos $\mathcal{D}_k f$. Entre estos operadores debe cumplirse la condición de *consistencia*, o *compatibilidad*,

$$\mathcal{D}_k \mathcal{R}_k = I_{V^k}. \quad (4)$$

Dada una sucesión de operadores de discretización y reconstrucción con las características anteriores, es posible definir los operadores de decimación y predicción de la siguiente manera:

$$D_{k+1}^k := \mathcal{D}_k \mathcal{R}_{k+1}, \qquad P_k^{k+1} := \mathcal{D}_{k+1} \mathcal{R}_k. \quad (5)$$

[1] El *núcleo*, o *kernel*, de una aplicación lineal $A : V \longrightarrow W$ son el conjunto de vectores en V cuya imagen es 0,

$$\ker(A) := \{v \in V \ : \ Av = 0\}.$$

Parece haber una dependencia explícita de D_{k+1}^k a $\mathcal{R}_k$, pero es fácil comprobar que la decimación es completamente independiente a la reconstrucción cuando la sucesión de discretizaciones es *anidada*, es decir, si

$$\forall f \in \mathcal{F} \ : \ \mathcal{D}_{k+1} f = 0 \quad \Longrightarrow \quad \mathcal{D}_k f = 0.$$

En este caso, el operador D_{k+1}^k está caracterizado por la siguiente propiedad

$$D_{k+1}^k(\mathcal{D}_{k+1} f) = \mathcal{D}_k f \qquad \forall f \in \mathcal{F}. \tag{6}$$

El proceso de discretización elegido para cada aplicación está relacionado con la naturaleza de los datos. En muchas aplicaciones, los datos se asocian a una malla subyacente, que puede considerarse como el nivel más fino dentro de una jerarquía de mallas anidadas. En la solución numérica de algunas ecuaciones diferenciales ordinarias y parciales, por ejemplo, la solución discreta representa una aproximación a los *valores puntuales* de la solución exacta en una malla. En otras aplicaciones, como el tratamiento de imágenes médicas, el escáner tiene una resolución fija y la información en niveles de resolución más bajos debe obtenerse *agrupando* los datos para simular los efectos del escáner (y la imagen) en una menor resolución. En este caso, los datos están *naturalmente* asociados a las *celdas* que definen la malla subyacente.

Una vez especificada la configuración, la elección de un operador de reconstrucción apropiado proporciona el paso clave para la configuración de un esquema de multi-resolución. El proceso de reconstrucción se encuentra en el corazón mismo de un esquema de multi-resolución construido "a la Harten" [10, 11].

Una ventaja del HMRF en comparación con otros entornos multi-escala es que la reconstrucción puede ser no-lineal, dando lugar a operadores de predicción P_k^{k+1} no-lineales o adaptados a geometrías concretas, lo que se traduce en operadores de *síntesis* (o decodificación) con estas características. La posibilidad de utilizar operadores de reconstrucción no-lineales, capaces de obtener representaciones precisas de funciones discontínuas, ha sido utilizada con éxito en aplicaciones en las que intervienen datos con fuertes gradientes, como es el caso de las imágenes [3, 5, 6, 7, 8, 29].

En el contexto de los artículos recogidos en esta memoria, es importante remarcar que los operadores de predicción pueden ser entendidos como *esquemas de subdivisión recursiva*, que es una técnica ampliamente utilizada en el refinamiento de datos y en el diseño asistido por ordenador (CAD, del inglés *Computer Aided Design*), a la que se dedicará la Sección 3.

En particular, hemos aplicado el HMRF en ámbitos tan diversos como Química Analítica [C4, C5] (Sección 5), la mejora de diseños de secciones planas en un contexto de optimización multi-paramétrica [C3] (Sección 4.1) o la estimación

de parámetros estadísticos (*Uncertainty Quantification*, Sección 3.3) [C2]. En estos casos, hemos utilizado fundamentalmente el entorno de *valores puntuales*, que introducimos en la Sección 2.1. Para una revisión más completa del HMRF, recomendamos, por ejemplo, [11, 56].

2.1. El entorno de valores puntuales

Reconstruir una función, a partir de un conjunto discreto de datos representativo de la función, es un problema clásico en Teoría de Aproximación que depende de la interpretación que se asigne a los datos discretos. Probablemente el caso más simple sea el de la interpolación de *valores puntuales*, en el que se busca reconstruir una aproximación a una función desconocida a partir de una tabla de valores de la misma. Los operadores de reconstrucción del entorno *interpolador* o de *valores puntuales* del HMRF están basados en el uso de técnicas de interpolación. A continuación se describirá este entorno de multi-resolución en el caso univariante, que es el utilizado en los artículos que constituyen esta tesis doctoral.

Suponemos que en el nivel de resolución k los datos están naturalmente asociados a la malla $\Xi^k = (\xi_i^k)_{i\in\mathbb{Z}}$, donde

$$\xi_i^k = \xi_{2i}^{k+1}, \qquad \forall i \in \mathbb{Z}. \tag{7}$$

Si los operadores de discretización asociados se definen como

$$\mathcal{D}_k f := (f(\xi_i^k))_{i\in\mathbb{Z}},$$

se obtiene una sucesión anidada, ya que $\Xi^k \subset \Xi^{k+1}$. De esta sucesión de operadores de discretización, de (6) es fácil ver que se obtiene el operador decimación

$$v^k := D_{k+1}^k v^{k+1} := (v_{2i}^{k+1})_{i\in\mathbb{Z}}.$$

Es decir, la decimación coincide con el operador *downsampling* de teoría de filtros.

La predicción se puede relacionar con la reconstrucción, como se indica en (5):

$$\begin{aligned}(P_k^{k+1}v^k)_{2i} &= (\mathcal{D}_{k+1}\mathcal{R}_k v^k)_{2i} = (\mathcal{R}_k v^k)(\xi_{2i}^{k+1}) = (\mathcal{R}_k v^k)(\xi_i^k),\\ (P_k^{k+1}v^k)_{2i+1} &= (\mathcal{D}_{k+1}\mathcal{R}_k v^k)_{2i+1} = (\mathcal{R}_k v^k)(\xi_{2i+1}^{k+1}).\end{aligned}$$

La condición de consistencia (1), $D_{k+1}^k P_k^{k+1} = I_{V^k}$, en este entorno se traduce en

$$(P_k^{k+1}v^k)_{2i} = (D_{k+1}^k P_k^{k+1} v^k)_i = v_i^k, \qquad \forall i \in \mathbb{Z}, \tag{8}$$

es decir, la predicción debe conservar todos los valores de v^k en las posiciones pares. Por tanto

$$f(\xi_i^k) = v_i^k = (\mathcal{R}_k v^k)(\xi_i^k),$$

es decir, $\mathcal{R}_k$ interpola los datos v^k en la malla Ξ^k. Si se denota por $\mathcal{I}(\xi, v^k)$ a la técnica de interpolación utilizada para definir el operador reconstrucción, se puede escribir

$$(\mathcal{R}_k v^k)(\xi) = \mathcal{I}(\xi, v^k).$$

Por un lado, nótese que el error de predicción asociado (2), e^k, es cero en las posiciones pares como consecuencia de (8),

$$e^k_{2i} = v^{k+1}_{2i} - (P^{k+1}_k D^k_{k+1} v^{k+1})_{2i} = v^k_i - (P^{k+1}_k v^k)_{2i} = 0, \qquad \forall i \in \mathbb{Z}.$$

Por otro lado,

$$e^k_{2i+1} := v^{k+1}_{2i+1} - (P^{k+1}_k v^k)_{2i+1} = f(\xi^{k+1}_{2i+1}) - \mathcal{I}(\xi^{k+1}_{2i+1}, v^k), \qquad \forall i \in \mathbb{Z},$$

es decir, en las posiciones impares el error es precisamente el error de interpolación de la técnica de interpolación.

Estas identidades sugieren definir los coeficientes d^k (la información no-redundante de e^k) como el conjunto de errores de interpolación producidos en las posiciones impares de la malla, $d^k_i := e^k_{2i+1}$.

La biyección $v^{k+1} \leftrightarrow (v^k, d^k)$, que permite transvasar la información entre niveles resolución, toma la forma

$$\begin{aligned} v^{k+1}_{2i} = v^k_i,\ v^{k+1}_{2i+1} = d^k_i + \mathcal{I}(\xi^{k+1}_{2i+1}, v^k), \qquad \forall i \in \mathbb{Z}, \\ \updownarrow \\ v^k_i = v^{k+1}_{2i},\ d^k_i = v^{k+1}_{2i+1} - \mathcal{I}(\xi^{k+1}_{2i+1}, v^k), \qquad \forall i \in \mathbb{Z}. \end{aligned}$$

Un ejemplo clásico de operador reconstrucción en este entorno es la interpolación polinómica a trozos que sigue: Para cada intervalo de la malla, $[\xi^k_i, \xi^k_{i+1}]$, se toma el polinomio $Q_{i,k}$ de grado $2q-1$ tal que $Q_{i,k}(\xi^k_{i+j}) = v^k_{i+j}$, $j = -q+1, \ldots, q$, y se define

$$(\mathcal{R}_k v^k)(\xi) := Q_{i,k}(\xi), \qquad \forall \xi \in [\xi^k_i, \xi^k_{i+1}]. \tag{9}$$

Si se expresa $Q_{i,k}$ mediante la base de Lagrange,

$$Q_{i,k}(\xi) = \sum_{j=-q+1}^{q} v^k_{i+j} L_j(2^k \xi - i), \quad L_j(\xi) := \prod_{\substack{l=-q+1 \\ l \neq j}}^{q} \frac{\xi - l}{j - l},$$

el operador de predicción que se obtiene a partir de esta técnica de interpolación puede expresarse como

$$(P^{k+1}_k v^k)_{2i+1} = Q_{i,k}(\xi^{k+1}_{2i+1}) = \sum_{j=-q+1}^{q} v^k_{i+j} a^q_j. \tag{10}$$

siendo $a_j^q := L_j\left(\frac{1}{2}\right)$, $j = -q+1, \ldots, q$, que depende de j y q pero es independiente de i y k. Se aprecia claramente su dependencia lineal respecto de los datos.

Ejemplos clásicos de operadores de predicción son los basados en las interpolaciones polinómicas de grado uno y grado tres ($q = 1, 2$):

$$(P_k^{k+1} v^k)_{2i+1} = \frac{1}{2} v_i^k + \frac{1}{2} v_{k+1}^k, \qquad (q = 1) \qquad (11)$$

$$(P_k^{k+1} v^k)_{2i+1} = -\frac{1}{16} v_{i-1}^k + \frac{9}{16} v_i^k + \frac{9}{16} v_{i+1}^k - \frac{1}{16} v_{i+2}^k. \qquad (q = 2) \qquad (12)$$

En el contexto de la subdivisión recursiva, estos operadores de predicción polinómicos se conocen como los esquemas de Deslauriers-Dubuc [41], sobre los que se volverá a hablar en la Sección 3.1.

En algunas aplicaciones se necesita reconstruir funciones a partir de datos discretos que contienen variaciones rápidas de magnitud, las cuales pueden estar asociadas a discontinuidades de la función subyacente. De utilizarse una interpolación lineal en estos casos, podrían aparecer oscilaciones no deseadas alrededor de la discontinuidad (similar al fenómeno de Gibbs). Existen operadores de reconstrucción no-lineales especialmente diseñados para interpolar los datos sin producir oscilaciones, como por ejemplo las reconstrucciones ENO [57], WENO [65], PCHIP [9] y PPH [4, 5].

3. Esquemas de subdivisión

Los esquemas de subdivisión son una técnica para el refinamiento recursivo de datos. La subdivisión recursiva destaca por su simplicidad inherente, que ha promovido su uso como herramienta de reconstrucción y aproximación, en particular en la generación eficiente de curvas y superficies en el diseño asistido por ordenador (CAD) [40].

Un esquema de subdivisión [22, 43] es un proceso iterativo que, a partir de un conjunto de datos inicial f^0, calcula una sucesión de conjuntos de datos $(f^k)_{k\geq 0}$, asociados a niveles de refinamiento cada vez mas elevados. Cada nuevo conjunto de datos, f^{k+1}, es definido a partir del anterior, f^k, mediante un conjunto finito de operaciones 'sencillas', que es lo que convierte estos procesos en herramientas muy eficientes en diversas aplicaciones.

Se utilizará el ejemplo 'canónico' para introducir los conceptos más relevantes en esta teoría: el esquema de las poligonales[2]. En el caso *univariante*, donde los datos son sucesiones bi-infinitas $f^k = (f_i^k)_{i=-\infty}^{+\infty}$, el esquema está formado por dos *reglas* de subdivisión que distinguen entre posiciones pares e impares:

$$f_{2i}^{k+1} := f_i^k, \quad f_{2i+1}^{k+1} := \frac{1}{2} f_i^k + \frac{1}{2} f_{i+1}^k, \qquad \forall i \in \mathbb{Z}. \tag{13}$$

La sucesión de datos f^k puede asociarse a la malla $2^{-k}\mathbb{Z}$, lo cual posibilita entender los datos como puntos $(i2^{-k}, f_i^k)$ en $\mathbb{R}^2$ y dar una interpretación geométrica de (13): En la iteración $k+1$ se conservan los puntos de la iteración k,

$$((2i)2^{-(k+1)}, f_{2i}^{k+1}) = (i2^{-k}, f_i^k),$$

y se añade la media de cada par de puntos consecutivos,

$$((2i+1)2^{-(k+1)}, f_{2i+1}^{k+1}) = \frac{1}{2}(i2^{-k}, f_i^k) + \frac{1}{2}((i+1)2^{-k} f_{i+1}^k).$$

En general, un esquema de subdivisión define cada nuevo dato generado mediante un conjunto de operaciones sencillas que involucran una cantidad finita de datos de la iteración anterior. Esta propiedad, conocida como *localidad*, implica que posibles perturbaciones en los datos se propaguen controladamente a lo largo de las iteraciones [43]. Es decir, si se modifica un dato inicial, los puntos afectados por tal variación están en una región acotada de la malla, en este ejemplo, el intervalo abierto $(i-1, i+1)$ [43].

Nótese que todos los puntos generados en el nivel $k+1$ pertenecen a la poligonal de vértices $(i2^{-k}, f_i^k)_{i\in\mathbb{Z}}$. De esto deducimos que los puntos de cualquier

[2] Una poligonal es una función definida a trozos por polinomios de primer grado. El punto que conecta dos segmentos rectos es conocido como vértice.

iteración están sobre la poligonal inicial, con vértices $(i, f_i^0)_{i\in\mathbb{Z}}$ y que el esquema de subdivisión genera (asintóticamente) todos los valores de esta función poligonal en los puntos diádicos, que es un conjunto denso en los reales. En estos casos, se dice que el esquema de subdivisión *converge* a una función F, conocida como *función límite*, que depende de los datos iniciales.

El esquema de subdivisión (13) es un ejemplo de esquema *univariante*, es decir, que los datos sobre los que actúa son sucesiones, $f^k = (f_i^k)_{i\in\mathbb{Z}}$. Si el esquema converge, a partir de una sucesión inicial $f^0 = (f_i^0)_{i\in\mathbb{Z}}$ se obtiene una función límite de una variable (v.g. curvas). En la práctica, solo es necesario ejecutar una cantidad finita de refinamientos (iteraciones) para 'generar' la función, por ejemplo para su visualización en una aplicación concreta [43].

Los esquemas de subdivisión *multi-variantes* manipulan datos estructurados mediante mallas multi-dimensionales, $f^k = (f_\alpha^k)_{\alpha\in\mathbb{Z}^s}$, $s > 1$, y pueden converger a funciones de varias variables. En este caso, se puede hablar de convergencia a superficies e incluso variedades diferenciables, siempre y cuando el conjunto de datos inicial y las reglas de subdivisión sean adecuadas. Esto añade diversidad al tipo de situaciones en las que se aplica la subdivisión recursiva [5, 16, 58, 81].

Además de la convergencia, otra propiedad fundamental de los esquemas de subdivisión es la *estabilidad*, que determina la magnitud de las modificaciones en la función límite derivadas de perturbaciones en los datos iniciales.

Para esquemas como (13), donde f_i^{k+1} depende *linealmente* de los datos en f^k, el estudio de la convergencia se realiza de una manera sistemática mediante la teoría correspondiente [22, 43], que está bien establecida y consolidada. En este caso, la estabilidad es una consecuencia de la convergencia. Cuando las reglas de subdivisión son *no-lineales* [2, 5, 15, 29], se requieren técnicas completamente diferentes y la teoría subyacente es mucho más reciente [28, 36, 37, 42, 45, 51, 54, 61, 79]. La convergencia y la estabilidad de estos procesos recursivos son esenciales para sus aplicaciones y, por lo tanto, su estudio ha sido y sigue siendo un tema de investigación activo.

Obsérvese que la convergencia, localidad y estabilidad de los esquemas de subdivisión tiene un impacto positivo en la generación y manipulación de objectos geométricos. En términos del CAD, dado un polígono de control f^0, un esquema de subdivisión define una curva asociada. Si se modifica un punto del polígono de control, la curva solo cambia en una región bien delimitada, que envuelve el punto modificado. Estas propiedades permiten que un diseñador gráfico pueda retocar y perfilar sus modelos de manera local, sin alterar otras partes que quizás ya estén de su agrado, lo cual resulta atractivo en el modelaje de objetos. En el cine de animación, la subdivisión recursiva se utilizó por primera vez en el corto de Pixar "Geri's Game" [40], a finales de los noventa.

Cuando los datos iniciales provienen de una función suave, un requerimiento

bastante habitual es que la subdivisión recursiva genere una aproximación *suficientemente precisa* de la función original. La capacidad de aproximación, o de precisión, de los datos generados por un esquema de subdivisión es un factor importante a tener en cuenta en muchas aplicaciones.

Otro requerimiento que resulta útil en muchas aplicaciones es la *reproducción*, es decir, la reconstrucción exacta de una familia de funciones. Por ejemplo, el esquema de subdivisión (13) es capaz de *reproducir* funciones poligonales.

La reproducción de polinomios [25, 33, 43] y de polinomios exponenciales [26, 32, 34, 44, 78] es interesante desde un punto de vista teórico, pues está relacionada con otras propiedades del esquema de subdivisión (como la aproximación y la convergencia), pero también desde la vertiente práctica, ya que permite dibujar con exactitud curvas relevantes en geometría, como las secciones cónicas.

Los esquemas de subdivisión lineales capaces de reproducir polinomios exponenciales tienen, necesariamente, reglas de subdivisión que varían a lo largo de las iteraciones, y por ello se denominan *no-estacionarios*. En [C6] se demuestra que los esquemas estacionarios *no-lineales* también pueden reproducir polinomios exponenciales y que presentan algunas ventajas respecto a los esquemas lineales no-estacionarios.

En otras situaciones puede ser importante establecer ciertas restricciones sobre f^k y sobre la función límite. Por ejemplo, si los datos representan una cantidad física que debe tener un valor real positivo, es necesario que los nuevos datos generados por subdivisión sean también positivos. De manera análoga, puede requerirse la preservación de *monotonía* o la *convexidad*. El mantenimiento de alguna de estas propiedades puede entenderse como casos particulares de *preservación de la forma* [61], lo que ha motivado el desarrollo de esquemas *no-lineales* específicamente diseñados para mantener una (o más) de estas propiedades. Algunos ejemplos que se pueden encontrar en la literatura son los esquemas *esencialmente no-oscilatorios* [28] (obtenidos a partir de ciertos operadores de predicción no-lineales en el HMRF [57, 65]) o esquemas que preservan la monotonía [15, 62] o la convexidad [5] en los datos. La investigación realizada durante este proyecto de tesis ha dado lugar a dos nuevos esquemas de subdivisión en esta línea [C1, C6].

Los esquemas de subdivisión pueden considerarse el núcleo de mi actividad durante los estudios de doctorado. Por una parte, los operadores de subdivisión utilizados como operadores de predicción dentro del HMRF se han aplicado en diversos contextos: Estimación de parámetros estadisticos (Uncertainty Quantification) [C2], optimización de secciones planas en el diseño de veleros de competición [C3] y mejora de herramientas de análisis en Química Analítica [C4, C5]. Por otro lado, hemos desarrollado y estudiado teóricamente nuevos esquemas no-lineales con propiedades orientadas a aplicaciones específicas [C1, C6].

Dado que los artículos que conforman esta tesis doctoral se centran en el estudio

y el uso de esquemas de subdivisión *univariantes, uniformes* y *binarios*, en la Sección 3.1 se definirán las propiedades principales de la subdivisión recursiva en este contexto.

3.1. El caso univariante

Definición 1. Un *esquema de subdivisión univariante* es una sucesión de operadores $\{S^k\}_{k\geq 0}$, $S^k : \ell_\infty(\mathbb{Z}) \longrightarrow \ell_\infty(\mathbb{Z})$, que permite definir recursivamente una sucesión de sucesiones acotadas $(f^k)_{k\geq 0} \subset \ell_\infty(\mathbb{Z})$ a partir de una sucesión inicial acotada de datos $f^0 = (f_i^0)_{i\in\mathbb{Z}} \in \ell_\infty(\mathbb{Z})$, de la siguiente manera:

$$f^{k+1} := S^k f^k, \qquad k \geq 0.$$

El esquema de las poligonales (13), definido en la Sección 3, es un ejemplo de esquema *uniforme*, *binario* y *univariante*. Los operadores de subdivisión de esta clase se definen a partir de dos reglas de subdivisión Ψ_0^k y Ψ_1^k que distinguen entre datos pares e impares,

$$f_{2i}^{k+1} = \Psi_0^k(f_{i-q}^k, f_{i-q+1}^k, \ldots, f_{i+q}^k), \qquad f_{2i+1}^{k+1} = \Psi_1^k(f_{i-q}^k, f_{i-q+1}^k, \ldots, f_{i+q}^k),$$

para cierto $q > 0$. Si $\{\Psi_0^k, \Psi_1^k\}_{k\geq 0}$ son funciones lineales, entonces el esquema es *lineal*. Si las reglas $\{\Psi_0^k\}_{k\geq 0}$ son tales que $f_{2i}^{k+1} = f_i^k$, entonces es *interpolador*. Si las reglas de subdivisión $\{\Psi_0^k, \Psi_1^k\}_{k\geq 0}$ son las mismas a lo largo de las iteraciones, i.e. no dependen de k, el esquema de subdivisión es *estacionario*. En este caso se denotará:

$$\Psi_0 := \Psi_0^k, \quad \Psi_1 := \Psi_1^k, \quad S := S^k.$$

El esquema de subdivisión (13) es estacionario, lineal e interpolador.

Los esquemas de subdivisión utilizados en aplicaciones prácticas han de ser *convergentes*, un concepto que se define a continuación de manera precisa.

Definición 2. Un esquema de subdivisión es *convergente* si

$$\forall f^0 \in \ell_\infty(\mathbb{Z}) \quad \exists S^\infty f^0 \in \mathcal{C}(\mathbb{R}) : \quad \lim_{k\to\infty} \sup_{i\in\mathbb{Z}} |f_i^k - (S^\infty f^0)(i2^{-k})| = 0.$$

Se denota por $S^\infty : \ell_\infty(\mathbb{Z}) \longrightarrow \mathcal{C}(\mathbb{R})$ al operador que envía cada dato inicial f^0 a su correspondiente función límite.

Se puede demostrar [43] que esta definición de convergencia es equivalente a que las funciones poligonales $\mathbb{P}^k$ tales que $\mathbb{P}^k(i2^{-k}) = f_i^k$ formen una sucesión de Cauchy.

Otra propiedad igualmente importante es la *estabilidad*, que se define formalmente como sigue.

Definición 3. Un esquema de subdivisión convergente es *estable* si el operador $S^\infty : \ell_\infty(\mathbb{Z}) \longrightarrow \mathcal{C}(\mathbb{R})$ es Lipschitz continuo:

$$\exists L > 0 \ : \ \|S^\infty f^0 - S^\infty g^0\|_\infty \leq L\|f^0 - g^0\|_\infty, \qquad \forall f^0, g^0 \in \ell_\infty(\mathbb{Z}).$$

En esquemas lineales, es fácil ver que la estabilidad es una consecuencia directa de la convergencia del esquema. Sin embargo, la situación es muy diferente en el caso no-lineal.

En general, la teoría de esquemas de subdivisión trata de inferir propiedades de las funciones límites $S^\infty f^0$ a partir de la definición de las reglas de subdivisión. De esta manera se pueden estudiar propiedades básicas como la convergencia o la estabilidad del esquema, y también otras que pueden ser convenientes en diversas aplicaciones, como la regularidad, la capacidad de aproximación, la reproducción exacta, la preservación de la forma, etc. Todo ello a partir de la expresión de las reglas de subdivisión Ψ_j^k.

En algunas aplicaciones, como en el diseño asistido por ordenador, puede interesar que las curvas que se generen a partir de esquemas de subdivisión tengan cierta regularidad.

Definición 4. Un esquema de subdivisión convergente es $\mathcal{C}^\alpha$ si[3]

$$S^\infty f^0 \in \mathcal{C}^\alpha, \qquad \forall f^0 \in \ell_\infty(\mathbb{Z}).$$

También suele ser importante conocer la *capacidad de aproximación* de un esquema de subdivisión, en el sentido de la siguiente definición.

Definición 5. Un esquema convergente tiene *orden de aproximación* r si para cualquier función suficientemente suave F,

$$\|F(h\,\bullet) - S^\infty f^0\|_\infty \leq Ch^r, \qquad f^0 = F|_{h\mathbb{Z}}, \quad \forall 0 < h < h_0.$$

Es decir, el orden de aproximación de un esquema de subdivisión, mide como se reduce el error al intentar aproximar F aplicando el esquema de subdivisión sobre $(F(ih))_{i\in\mathbb{Z}}$, siendo el espaciado de la malla h suficientemente pequeño.

Además, en algunas aplicaciones se desea reconstruir ciertas funciones $F \in \mathcal{F}$ de manera exacta, y no aproximada. Puede ser de interés práctico la reconstrucción de circunferencias, elipses, hipérbolas, etc. y esto puede hacerse eficientemente si se emplean esquemas de subdivisión que *reproduzcan* la clase de funciones que definen las curvas anteriores [34].

Definición 6. Un esquema de subdivisión convergente *reproduce* una familia de funciones $\mathcal{F}$, si para cualquier función $F \in \mathcal{F}$ el esquema converge a F a partir de los datos iniciales $f^0 = F|_\mathbb{Z}$:

$$S^\infty F|_\mathbb{Z} = F, \qquad \forall F \in \mathcal{F}.$$

[3] Se define $\mathcal{C}^\alpha$ como el conjunto de funciones α veces diferenciables y continuas.

Los esquemas de Deslauriers-Dubuc [41] son un ejemplo clásico de esquemas lineales interpoladores que reproducen polinomios de grado arbitrariamente alto (pero fijo). Se pueden construir a partir de la interpolación polinómica a trozos descrita en la Sección 2.1. Como puede observarse en (10), son esquemas estacionarios, porque los coeficientes a_j^q de la combinación lineal que definen sus reglas son independientes de k.

Desde un punto de vista teórico, la reproducción de polinomios y de polinomios exponenciales es interesante porque está relacionada con la capacidad de aproximación y la suavidad del esquema [31, 35, 43, 61]. Además, un esquema lineal que reproduzca polinomios exponenciales es necesariamente *no-estacionario* [26, 32, 34, 44], y sus reglas de subdivisión Ψ_0^k, Ψ_1^k dependen de ciertos parámetros implicados en la expresión del espacio de polinomios exponenciales que reproduce. No obstante, en [C6] se obtiene un esquema de subdivisión *no-lineal* que reproduce funciones trigonométricas (que son un caso particular de polinomios exponenciales) cuyas reglas son estacionarias y no dependen de los parámetros mencionados.

Es bien sabido que los esquemas de subdivisión lineales y no-estacionarios pueden lograr este objetivo [34, 44]. Pero su aplicación requiere la determinación práctica de los parámetros, que definen las reglas dependientes del nivel, mediante el procesamiento previo de los datos disponibles [17, 44].

Dado que diferentes secciones cónicas requieren diferentes reglas de refinamiento para garantizar la reproducción exacta, no es posible reproducir una forma compuesta, por partes, por varias funciones trigonométricas con el mismo esquema lineal. En [C6] se muestra que la reproducción exacta de diferentes formas cónicas se puede lograr utilizando el mismo esquema no-lineal, sin ningún procesamiento previo de los datos.

Para aplicaciones donde, por exigencias de la naturaleza del problema, los datos son positivos, o monótonos, o convexos, el esquema de subdivisión debe conservar el tipo (o la *forma*) de los datos.

Definición 7. Un esquema de subdivisión *preserva* (estrictamente) la *positividad* de los datos, o equivalentemente, es (estrictamente) positivo, si para cualquier $f \in \ell_\infty(\mathbb{Z})$,

$$f_i(>) \geq 0 \ \forall i \in \mathbb{Z} \quad \rightarrow \quad Sf_i > 0 \quad \forall i \in \mathbb{Z}.$$

Un esquema *preserva* (estr.) la *monotonía*, o es monótono, si

$$\nabla f_i(>) \geq 0 \ \forall i \in \mathbb{Z} \quad \rightarrow \quad \nabla Sf_i > 0 \ \forall i \in \mathbb{Z},$$

donde $\nabla : \ell_\infty(\mathbb{Z}) \longrightarrow \ell_\infty(\mathbb{Z})$ es el operador en diferencias finitas, $\nabla f_i := f_{i+1} - f_i$.

Un esquema *preserva* (estr.) la *convexidad*, o es convexo, si

$$\nabla^2 f_i(>) \geq 0 \ \forall i \in \mathbb{Z} \quad \rightarrow \quad \nabla^2 Sf_i > 0 \ \forall i \in \mathbb{Z}.$$

Los esquemas de subdivisión interpoladores, de alta precisión y que preservan de la forma de los datos resultan de gran utilidad en ciertas aplicaciones [5], lo cual ha motivado a diversos autores a diseñar esquemas no-lineales con estas propiedades [15, 62]. Hemos abordado este tema durante el proyecto de tesis, habiendo definido dos nuevos esquemas de subdivisión [C1, C6].

La convergencia de un esquema de subdivisión lineal es una condición suficiente para la estabilidad, pero no lo es en el caso no-lineal. Es más complejo demostrar que un esquema no-lineal es convergente y estable. Por ello, se dispone de resultados teóricos [2, 5, 15, 37, 39, C1, 54] que aseguran estas propiedades si se cumplen ciertos requisitos. Para introducir estos resultados, que hemos empleado en [C1, C6], es necesario definir el concepto de *esquema en diferencias*. Cabe mencionar que los esquemas no-lineales suelen ser estacionarios, por lo que los resultados están limitados a este caso.

Definición 8. Un esquema de subdivisión S tiene *esquema en diferencias* de orden n si existe un esquema $S^{[n]}$ tal que

$$\nabla^n S = S^{[n]} \nabla^n.$$

Cabe destacar que la existencia del esquema en diferencias $S^{[n]}$ no está garantizada, exceptuando el caso lineal, donde cualquier esquema convergente tiene esquema en diferencias de orden $n = 1$.

El esquema en diferencias permite analizar el comportamiento de las diferencias finitas de los datos f^k a partir de $S^{[n]}$, mediante la expresión

$$\nabla^n f^k = (S^{[n]})^k \nabla^n f^0.$$

Esta propiedad es clave en el análisis de la convergencia, tanto en el caso lineal como no-lineal.

En [2, 5, 15, 37, 39, C1, 54, C6], los autores construyen y analizan diversos esquemas de subdivisión no-lineales que pueden describirse como una perturbación no-lineal de un esquema convergente lineal T:

$$Sf = Tf + \mathcal{F}(\nabla^n f), \qquad \forall f \in \ell_\infty(\mathbb{Z}), \tag{14}$$

donde $\mathcal{F} : \ell_\infty(\mathbb{Z}) \to \ell_\infty(\mathbb{Z})$ es un operador (posiblemente no-lineal). Si existe $T^{[n]}$, lo cual es fácil de comprobar [43], entonces un esquema de la forma (14) tiene esquema en diferencias de orden n. En concreto este es

$$S^{[n]} f = T^{[n]} f + \nabla^n \mathcal{F}(f), \qquad \forall f \in \ell_\infty(\mathbb{Z}). \tag{15}$$

Para analizar su convergencia y estabilidad se han empleado resultados específicos de [2, 15].

3.2. [Adv. Comput. Math., 2017] A family of non-oscillatory 6-point interpolatory subdivision schemes

Un esquema lineal interpolador, con un orden de aproximación $r > 2$, es seguro que producirá oscilaciones y perderá toda su precisión cuando los datos presenten variaciones súbitas. Se muestra un ejemplo de esto en la Figura 1. Utilizando el esquema de 6 puntos de Deslauriers-Dubuc (DD), $S_{3,3}$, que es lineal y tiene orden 6, se ha obtenido una función límite que oscila alrededor del salto. Esto quiere decir que en los alrededores de la discontinuidad el esquema pierde su capacidad para reconstruir la función.

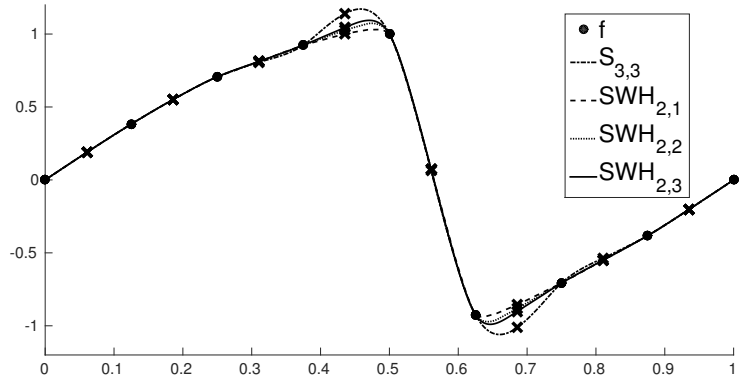


Fig. 1: A partir de los datos iniciales (•) se generan funciones límite mediante diversos esquemas de subdivisión. Las equis (×) son los datos generados después de una iteración.

Diversas técnicas de interpolación polinómica por segmentos se han considerado en la literatura para construir esquemas de subdivisión interpoladores que evitan oscilaciones no deseadas. Ejemplos de tales esquemas son los esquemas ENO-WENO [28, 57, 65], el esquema PPH [4, 5], los esquemas Power$_p$ [2, 14, 36] y los esquemas de conservación de la forma descritos en [61]. Estos últimos deben su carácter no-oscilatorio al juicioso uso de ciertos promedios no-lineales.

En este trabajo, se propone y analiza una nueva familia de esquemas de subdivisión no-lineales, los $S\mathrm{WH}_{p,q}$, que pueden considerarse versiones no-oscilatorias del esquema $S_{3,3}$, al igual que los esquemas Power$_p$ se consideran versiones no-lineares y no-oscilatorias del esquema interpolador DD de 4 puntos. De hecho, su diseño está relacionado con el de los esquemas Power$_p$.

Se demuestra que los nuevos esquemas reproducen exactamente polinomios de grado tres y que la distancia en norma infinito al esquema DD de 6 puntos es pequeña en regiones suaves, como se aprecia en la Figura 1.

Además, se prueba que el primer y el segundo esquema en diferencias están bien definidos para cada miembro de la familia, lo que permite dar una prueba simple de la convergencia uniforme de estos esquemas y también estudiar su estabilidad como en [15, 54].

Sin embargo, el estudio teórico de la estabilidad basado en los resultados de [54] no es concluyente en el caso de estudio, por lo cual, se realizan una serie de experimentos numéricos que parecen indicar que solo unos pocos miembros de la nueva familia de esquemas son estables.

Por otro lado, las exhaustivas pruebas numéricas revelan que, para datos suaves, el orden de aproximación y la regularidad de la función límite pueden ser similares a los del esquema de DD de 6 puntos y superiores a los obtenidos con los esquemas de Power$_p$.

3.3. [Applied Mathematics and Nonlinear Sciences, 2017] High-accuracy approximation of piecewise smooth functions using the truncation and encode approach

Numerosos procesos físicos e industriales pueden simularse mediante una ecuación en derivadas parciales (EDP). Por ejemplo, en el diseño de apéndices para veleros, debe simularse el flujo del agua alrededor del perfil para calcular el coeficiente de arrastre. Para ello se usan las ecuaciones de Navier-Stockes, cuya resolución numérica es compleja y el tiempo de cálculo se dispara al aumentar la precisión.

Es posible que la simulación dependa de múltiples parámetros físicos cuyo valor es variable, por ejemplo la velocidad del velero y la inclinación de la proa, y por tanto deben tratarse como variables aleatorias. Entonces, el coeficiente de arrastre no es único, sino que depende del valor de cada parámetro. En la práctica, se puede establecer un mallado y resolver la EDP asociada a cada par de valores de la malla velocidad-inclinación. Como puede imaginarse, el coste computational es desorbitado si la malla es muy fina, y debe plantearse alguna estrategia.

En [1, 49] se definió un método llamado *Truncate and Encode* (TE, truncar y codificar), que aprovecha el entorno de multi-resolución de Harten para aproximar adaptativamente la solución de una EDP y estimar ciertos parámetros estadísticos en el contexto de Uncertainty Quantification. A groso modo, en cada nivel de resolución se decide si resolver la EDP o interpolar la solución con los datos existentes, reduciendo así el tiempo de cálculo. La decisión se basa en la precisión que tuvo la interpolación en el nivel anterior, y conviene escoger una técnica de interpolación de alto orden de aproximación y preferiblemente no-oscilatoria. De hecho, la interpolación es equivalente a un operador de subdivisión, por lo que los esquemas PCHIP [15] y $S\text{WH}_{p,q}$ [C1] pueden aplicarse y son muy recomendables.

En este artículo, se analiza el algoritmo TE aplicado a la aproximación de funciones y, en particular, su rendimiento para funciones suaves por partes. Se llevan a cabo algunos experimentos numéricos, comparando el rendimiento del algoritmo cuando se usan diferentes técnicas de interpolación lineal y no-lineal y se proporcionan algunas recomendaciones que nos parecen útiles para lograr un alto rendimiento del algoritmo. Los resultados indican que para incrementar el rendimiento de TE es conveniente utilizar esquemas de subdivisión de alto orden de aproximación.

4. Estrategias multi-escala en optimización a gran escala

La optimización [74] es una herramienta importante en la toma de decisiones y en el análisis de sistemas físicos. En un proceso de optimización se debe, en primera instancia, identificar una *función objetivo*, que mida el rendimiento del sistema que se esté estudiando, por ejemplo tiempo, energía, beneficios económicos, o cualquier cantidad o combinación de ellas que pueda representarse con un único número. Esta función depende de ciertas características del sistema, llamadas *variables* o *incógnitas*.

La finalidad del proceso es encontrar los valores de las variables que optimicen la función objetivo. A menudo las variables están restringidas, o limitadas, de alguna manera. Por ejemplo, las cantidades que representen la masa de objetos no pueden ser negativas.

Al proceso de identificar los objetivos, las variables y las restricciones de un problema dado se le conoce como *modelaje*. La construcción de un modelo adecuado es el primer paso, a menudo el más importante, en el proceso de optimización. Una vez obtenido el modelo, la solución se encuentra mediante la aplicación de un algoritmo de optimización, habitualmente con la asistencia de un ordenador.

En términos matemáticos, un problema de optimización consiste en minimizar (o maximizar) una *función objetivo* F dentro de un espacio de posibles soluciones *factibles*, digamos X. Esto es, hallar $u_{\text{mín}} \in X$ tal que $F(u_{\text{mín}}) \leq F(u)\ \forall u \in X$.

Las funciones objetivo deben estar definidas en un espacio de dimensión finita, i.e. $F : X \subset \mathbb{R}^N \longrightarrow \mathbb{R}$, para poder abordar el problema de optimización computacionalmente mediante algún algoritmo, llamado *optimizador*. Cuando la cantidad de variables N es grande, se habla de *optimización a gran escala*. Este tipo de problemas aparece a menudo a partir de la discretización de un problema de dimensión infinita, por ejemplo en el contexto del diseño óptimo, en el control óptimo, en la estimación de parámetros en sistemas gobernados por EDPs [19, 64, 80] y en el procesamiento de imágenes [24, 23, 76, 75, 82].

No existe un optimizador universal, más bien toda una colección de ellos, cada uno de los cuales hecho a medida para algún tipo de problema. La responsabilidad de elegir el algoritmo apropiadamente para una aplicación concreta recae sobre el usuario. Esta decisión es importante, pues determinará si el problema se resuelve rápida o lentamente y, ciertamente, si la solución será hallada.

En optimización a gran escala a menudo se pueden aplicar optimizadores que conllevan un esfuerzo computacional prohibitivo debido a la gran cantidad de variables involucradas.

El éxito de los métodos multigrid [20, 21, 52, 53, 69], como resolvedor eficiente de EDPs elípticas discretizadas, impulsó el desarrollo de métodos iterativos multinivel en optimización [18, 27, 30, 47, 48, 50, 60, 72] desde mediados de los años 80.

La idea que comparten estos métodos multi-nivel es la aplicación de un optimizador particular para resolver problemas auxiliares reducidos de menor dimensión, derivados de la discretización del problema infinito-dimensional con menor exactitud, y que por lo tanto son más rápidos de resolver (en términos de cálculo).

Aunque los métodos multi-nivel comparten una estructura común, se ha realizado el esfuerzo de desarrollar por separado las versiones multi-nivel de los optimizadores más comunes [30, 47, 48, 50]. En esta tesis doctoral se propone una estructura multi-nivel basada en el HMRF que permite implementar cualquier optimizador de manera arbitraria. En otras palabras, este método permite tratar al optimizador como a una 'caja negra', permitiendo al usuario utilizar el optimizador que más le convenga entre aquellos de los que disponga.

Mi trabajo en este campo surge de unas prácticas realizadas en IS&3D ENG[4], durante mis estudios de máster. El problema propuesto fue el de mejorar el rendimiento de secciones planas que se usan en el diseño de apéndices de veleros de competición. En particular, se quería reducir el arrastre[5] de timones, quillas y bulbos dentro del agua. El reto se planteó como un problema de optimización en el que se quería reducir el arrastre teórico teniendo en cuenta ciertas restricciones físicas y de diseño.

La función objetivo incluía una simulación CFD realizada por una rutina externa del tipo caja negra (`xfoil`[6]). IS&3D ENG propuso utilizar los optimizadores integrados en Matlab.

Durante el proceso de mejora debía modificarse la sección mediante perturbaciones suaves y sin oscilaciones, introduciendo en primer lugar variaciones globales a la sección para, conforme fuera mejorando, incidir en los detalles más locales. Esta idea encajaba con la estructura multi-escala del HMRF. La síntesis de este planteamiento nos condujo a definir una nueva estrategia de optimización.

Parte del trabajo realizado durante la colaboración con IS&3D ENG ha dado lugar a diversas participaciones en reuniones científicas (ver el Curriculum Vitae) y a la publicación incluida en esta memoria [C3, Sección 4.1].

A través de la colaboración con el equipo de investigación FUSCHROM[7] de Cromatografía Líquida[8], se aplicó exitósamente esta estrategia de optimización en un contexto completamente diferente [C4, Sección 5.1]. Aquí, el objetivo era maximizar la separación entre sustancias que han sido inyectadas en un dispositivo de separación. Nuevamente, la función objetivo era bastante compleja, ya que contiene

[4] www.is3de.com

[5] La resistencia al movimiento de un objeto a través de un *flujo*, como el aire o el agua.

[6] XFOIL es un programa interactivo para el diseño y análisis de perfiles aerodinámicos aislados subsónicos.

[7] https://sites.google.com/site/fuschrom/

[8] Una técnica que permite separar, identificar y cuantificar cada sustancia presente en una mezcla.

una simulación química, y se empleó como optimizador la rutina `patternsearch` de Matlab.

4.1. [Progress in Industrial Mathematics at ECMI 2016] A novel multi-scale strategy for multi-parametric optimization

El movimiento de un velero es consecuencia del equilibrio existente entre las fuerzas aerodinámicas, inducidas por el viento sobre las velas, y las fuerzas hidrodinámicas, resultantes del contacto del agua con las partes sumergidas del barco, que son el casco y los apéndices. Cada apéndice cumple una función. Por ejemplo, el timón marca la dirección del movimiento, la quilla evita desplazamientos laterales y el bulbo influye en el momento adrizante[9], evitando que el barco escore[10].

El modelado de estos apéndices se realiza a partir de su sección transversal, una curva plana cerrada llamada *perfil*, como el que se muestra en la Figura 2. El objetivo que se aborda en este trabajo [C3] es modificar (optimizar) un perfil dado para reducir el arrastre con el agua, sujeto a restricciones de diversa índole. Las hay estructurales, como por ejemplo que el perfil debe tener una longitud determinada, y las hay físicas, como que el coeficiente de sustentación debe estar comprendido entre dos valores admisibles. Las restricciones dependerán de la finalidad del apéndice (timón, quilla...). Visto de otro modo, las restricciones impuestas en la optimización convierten un perfil inicial cualquiera en el tipo de apéndice deseado.

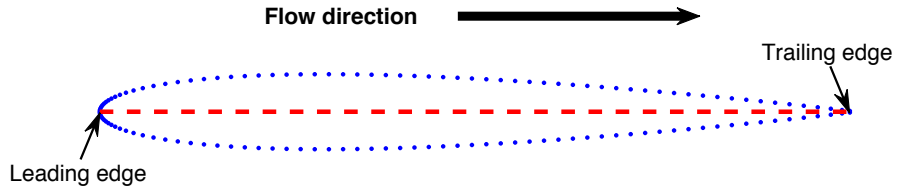


Fig. 2: Un ejemplo de perfil: el NACA0010 descrito por $N = 129$ puntos.

La estrategia que se plantea proporciona una sucesión de soluciones sub-óptimas, una a cada nivel de resolución, de modo que en el último paso se resuelve el problema de optimización completo (a gran escala), pero con una estimación inicial mucho más cercana a la solución deseada que la proporcionada inicialmente (que a menudo se elige arbitrariamente, pero también puede ser facilitada por el usuario), haciendo que el esfuerzo de cálculo requerido por el optimizador elegido sea factible.

[9] Mide la capacidad de una embarcación para mantenerse en posición vertical.
[10] Escorar: Inclinarse un barco sobre uno de sus costados.

Esta técnica aplica exhaustivamente un esquema de subdivisión, que debe ser escogido teniendo en cuenta la naturaleza de los datos manipulados. Puesto que se quiere modificar *suavemente* un perfil, pero evitando producir oscilaciones, se propone utilizar el esquema de subdivisión de B-Splines de orden 5 [43].

En el artículo se analiza el comportamiento del algoritmo aplicándolo a un problema académico, obteniendo una drástica reducción del coste computacional en comparación con la aplicación directa (sin estrategia multi-escala) del optimizador elegido.

Se plantea una optimización para el diseño de un apéndice, donde solo la estrategia multi-escala fue capaz de proporcionar resultados satisfactorios.

5. Aplicaciones en cromatografía líquida

La cromatografía líquida es una técnica utilizada en Química Analítica para separar, identificar y cuantificar cada uno de los solutos presentes en una mezcla.

Al insertar la mezcla junto con un disolvente a lo largo de un tubo, llamado *columna*, los distintos solutos de la mezcla fluyen (*precipitan*) a diferentes velocidades cuando interaccionan con el medio poroso del interior de la columna.

Si el experimento se configura correctamente, cada soluto sale (*eluye*) por el final de la columna de manera separada. Un sensor registra la cantidad de mezcla que eluye en cada instante de tiempo, y la gráfica que se obtiene con esta relación tiempo-cantidad[11] eluida se denomina *cromatograma*. Los diferentes solutos de la mezcla aparecen como 'picos' en el cromatograma, como se ilustra en las Figuras 3 y 4.

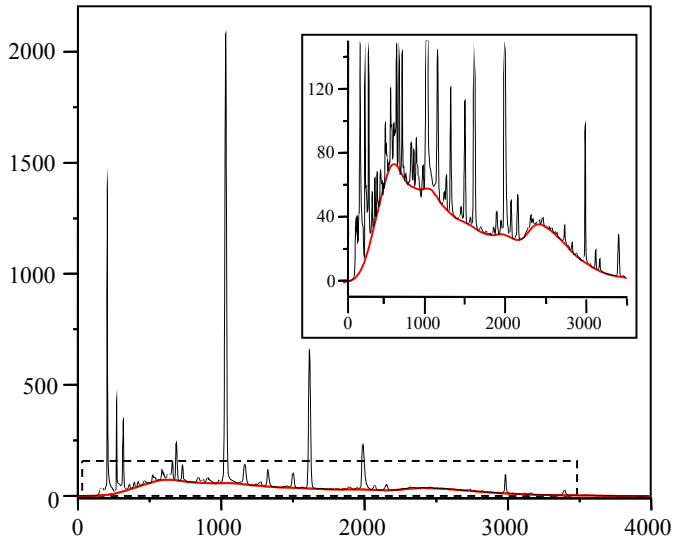


Fig. 3: Ejemplo de detección de línea base. En el recuadro superior derecho se muestra la ampliación de la zona marcada con un rectángulo discontinuo.

Algunos problemas que se pueden encontrar para cuantificar correctamente la cantidad de cada soluto en la mezcla, y que hemos abordado durante el doctorado mediante técnicas matemáticas, son: la presencia de una línea base, que se deriva del uso del disolvente; la presencia de *ruido*, que proviene de factores tanto ambientales como propios de la química de la mezcla; y el solapamiento de unos picos con otros. Para evitar este último problema, y así obtener picos bien *resueltos*, se

[11] La unidad de medida es aquella proporcionada por el dispositivo que mide cuanta luz no ha sido absorbida por la mezcla al salir de la columna. Si, por ejemplo, el receptor de luz es electrónico, la unidad de medida sería milivoltios.

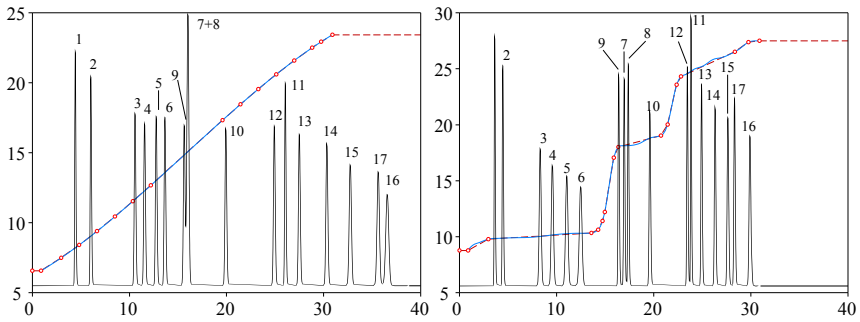


Fig. 4: Aumento de la resolución de los picos, asociados a 17 aminoácidos esenciales, y reducción del tiempo de elución. A la izquierda, el programa de gradiente inicial. A la derecha, el optimizado. En el eje horizontal se muestra el tiempo de elución (en minutos), en el vertical la concentración del disolvente inyectado en la columna.

necesita preestablecer la concentración de disolvente a inyectar en la columna en cada lapso de tiempo.

La colaboración con proyectos de Química Analítica se inició durante mis estudios de grado. Colaboré con el equipo de investigación CLECEM[12], en la asignatura optativa 'prácticas en empresa' bajo la tutela de Guillermo Ramis Ramos. Fruto de esta colaboración, publicamos un artículo [C8] (anterior a mis estudios de doctorado) sobre el análisis y la clasificación de cromatogramas.

Al comienzo de mi doctorado, se inició una colaboración con el grupo de investigación FUSCHROM[13], también de Química Analítica. Estaban interesados en el posprocesado de señales cromatográficas. En particular, se quería eliminar ciertas *líneas base* presentes habitualmente en los datos mediante algún algoritmo matemático implementado computacionalmente.

A través de los contactos de mi directora de tesis, Rosa Donat, conocimos un algoritmo, BEADS, que estaba proporcionando excelentes resultados. Está basado en la optimización de una función objetivo, concienzudamente diseñada, mediante la técnica *mayorización-minimización* [46, 63]. Cabe decir que BEADS también está preparado para eliminar *ruido*.

Aplicando BEADS a diferentes cromatogramas aparecieron algunas limitaciones y dificultades asociadas a su uso. Se planteó un seguido de procedimientos para aplicar correcta y fácilmente este algoritmo en [C7] (no incluido en el compendio de artículos). En la Figura 3 se muestra un ejemplo de detección de línea base mediante el procedimiento que se plantea. Una vez detectada, tan solo es necesario

[12] https://www.uv.es/clecem/
[13] https://sites.google.com/site/fuschrom/

sustraerla.

Posteriormente, se planteó una nueva investigación: encontrar una nueva manera de diseñar, eficientemente, *programas de gradiente*. Matemáticamente hablando, consiste en encontrar una función que maximice la *resolución* de los picos.

El procedimiento que hasta el momento se empleaba consistía en considerar una función poligonal arbitraria con un número de vértices fijo. Mediante el uso de algún optimizador, v.g. un algoritmo genético, se determinaba la posición óptima de los vértices.

Esa estrategia requería mucho tiempo de cálculo y una cantidad de nodos muy limitada. Al tratarse de un problema a gran escala, se planteó el uso de la estrategia de optimización multi-nivel de la Sección 4 fruto de la colaboración con la empresa IS&3D ENG. Se obtuvieron resultados muy satisfactorios, como se recoge en el siguiente artículo [C4, Sección 5.1].

5.1. [J. Chromatogr. A, 2018] Gradient design for liquid chromatography using multi-scale optimization

El diseño de *programas de gradiente*, donde se especifica a la máquina cromatográfica la concentración de disolvente que debe introducirse en la columna en cada lapso de tiempo, es esencial para obtener picos bien resueltos, sin solapamientos, y así poder medir correctamente la cantidad de cada soluto que forma la mezcla. El objetivo que se plantea es encontrar la función programa de gradiente tal que maximiza la resolución a la vez que se verifican una serie de condiciones, necesarias para su correcta implementación en el laboratorio.

Se ha aplicado exitósamente la optimización multi-escala en este problema, consiguiendo no solo una resolución alta, sino también la reducción del tiempo de elución, que se traduce en menos horas de trabajo para el personal y el instrumental de laboratorio. Se muestra un ejemplo de optimización en la Figura 4.

5.2. [J. Chromatogr. A, 2019] Enhancement in the computation of gradient retention times in liquid chromatography using root-finding methods

En este trabajo [C5] se propone un nuevo método para la simulación de la posición de los picos en función del programa de gradiente. Este tipo de simulaciones son necesarias para llevar a cabo estudios como el anterior [C4, Sección 5.1].

El valor t que resuelve la ecuación integral

$$f(t) = \int_0^{g(t)} h(\tau) d\tau,$$

para ciertas funciones f, g, h definidas a partir de las condiciones del experimento, representa la posición en el eje de las abscisas en el que aparece un determinado pico del cromatograma, o en términos químicos, el tiempo que tarda un soluto en salir de la columna cromatográfica.

El enfoque que se empleaba hasta el momento consistía en discretizar la integral de la siguiente manera,

$$\int_0^{g(n\delta)} h(t)dt \approx \delta \sum_{i=0}^{n-1} h(g(i\delta)), \qquad n \in \mathbb{N}$$

siendo habitualmente $\delta = 10^{-3}$, y encontrar el valor de n de manera que la suma anterior estaba lo más cerca posible de $f(n\delta)$. Entonces, se deducía que $t \approx n\delta$.

En primer lugar, la aproximación a la integral que se estaba utilizando era muy pobre, pues se basaba en aproximar h mediante funciones escalonadas, cuando además, en algunos casos, h tenía primitiva. Como mejora, se propone utilizar la primitiva para reducir enormemente el coste computacional y, en caso de no haber primitiva, aproximar h mediante algún polinomio y emplear la primitiva del polinomio.

En segundo lugar, el modo en que se hallaba el valor n era muy rudimentario. Tan solo se incrementaba su valor hasta que se verificase cierta condición de parada. Teniendo una primitiva (aproximada) de h, digamos H, la resolución numérica de la ecuación integral puede entenderse como hallar el cero de la función

$$F(t) = f(t) - H(g(t)) + H(0).$$

Mi propuesta fue aplicar un algoritmo de búsqueda de ceros, derivado del método de Newton y del método de la bisección, que combinaba la velocidad de convergencia con las garantías de convergencia de ambos algoritmos.

Este nuevo enfoque permite, no solo calcular el tiempo de retención mucho más rápido, sino también incrementar la precisión, que ahora era inferior a 10^{-3}. En este trabajo también se hace un análisis teórico para garantizar que las aproximaciones al tiempo de retención se calculen con un error inferior a un umbral escogido.

6. Conclusiones, trabajo en progreso y perspectivas de futuro

En esta tesis doctoral se han propuesto, estudiado y analizado distintos esquemas de subdivisión, prestando especial atención al caso no-lineal.

Los esquemas de subdivisión no-lineales pueden esquivar algunas de las limitaciones que presentan los esquemas lineales en ciertas aplicaciones. En esta memoria, se ha obtenido un esquema no-lineal interpolador, con alta capacidad de aproximación, no-oscilatorio y que reproduce polinomios de hasta tercer grado [C1].

Además, se han considerado diversas aplicaciones en las que los esquemas de subdivisión juegan un papel relevante a través del entorno de multi-resolución de Harten.

Hemos investigado el uso de operadores de predicción (subdivisión) no-lineales en Uncertainty Quantification, implementados en la estrategia *Truncate and Encode* [1, C2].

Hemos propuesto una nueva estrategia de optimización basada en el entorno de multi-resolución de Harten, la cual ha sido aplicada en el diseño de secciones planas de ciertos apéndices de veleros de competición para reducir el arrastre con el agua con el objetivo de mejorar su eficiencia [C3].

Esta estrategia de optimización se ha utilizado también en problemas relacionados con el tratamiento de señales en Cromatografía Líquida. Hemos propuesto un método para el diseño de *programas de gradiente* [C4]. Como consecuencia de este trabajo, se puso de manifiesto la importancia de simular el *tiempo de elución* eficientemente. En [C5] hemos planteado una nueva manera de hacerlo que reduce drásticamente el tiempo necesario para diseñar un programa de gradiente, a la par que incrementa la precisión de los resultados.

Las publicaciones [C1, C2, C3, C4, C5] representan en realidad la parte consolidada de mi trabajo de investigación. Además de estas publicaciones, he de mencionar también los siguientes tres artículos (sometidos a publicación).

1- Nonlinear stationary subdivision schemes that reproduce trigonometric functions. *R. Donat and S. López-Ureña*

Como se ha expuesto en la Sección 3.1, el trabajo realizado en esta tesis doctoral ha permitido diseñar una nueva familia de esquemas de subdivisión interpoladora no-lineales con la capacidad de reproducir funciones trigonométricas y polinomios de segundo grado. Evidentemente, esta propiedad es interesante para el CAD, pues los esquemas pueden reproducir formas definidas a trozos mediante secciones cónicas (circunferencias, hipérbolas, elipses y parábolas). El artículo ha sido sometido a publicación, y después de recibir los comentarios de los revisores y realizar las modificaciones oportunas, estamos esperando una respuesta definitiva para su publicación. Se encuentra actualmente disponible en arXiv [C6].

2- A Multiresolution approach to solve large-scale optimization problems. *R. Donat and S. López-Ureña*

Este artículo formaliza la estrategia de optimización multi-escala que se plantea en [C3, Sección 4], y se compara con otros métodos de optimización multi-nivel. A través de diversos experimentos numéricos, tanto unidimensionales como bidimensionales, se estudia su rendimiento y se analiza el impacto del operador de predicción escogido para definir el entorno de multiresolución. Se llega a la conclusión de que es conveniente emplear, como operadores de predicción, esquemas de subdivisión de alto orden de aproximación. El trabajo ha sido sometido a publicación.

3- Multi-scale optimisation vs. genetic algorithms in the separation of diuretics by reversed-phase liquid chromatography . *T. Álvarez-Segura, S. López-Ureña, J.R. Torres-Lapasió and M.C. García-Alvarez-Coque*

En [C4, Sección 5.1] se pone de manifiesto que la estrategia de optimización anterior, basada en el HMRF, puede aplicarse en el diseño de *programas de gradiente*. En este trabajo se compara su rendimiento con otro método, basado en algoritmos genéticos, que es conocido por proporcionar buenos resultados en este problema. Se concluye que el enfoque multi-objetivo de los algoritmos genéticos es muy conveniente, pues da cierta libertad al usuario para que decida que programa de gradiente es más adecuado. En consecuencia, podría resultar muy conveniente utilizar los algoritmos genéticos como optimizador dentro de la estrategia multi-escala. Esta es una posibilidad que se contempla en [C3, Sección 4], y se reserva esta cuestión para el futuro. Hemos modificado el artículo de acuerdo con las indicaciones de los revisores y estamos esperando una decisión definitiva sobre su publicación.

Por otro lado, los resultados de las líneas de investigación, que se abrieron durante mis estancias en Italia y Alemania, siguen en proceso de redacción.

La reproducción de *polinomios exponenciales*, que generalizan las funciones trigonométricas, en un contexto multi-variante fue estudiada en una estancia con el profesor Tomas Sauer (U. Passau). Además, motivado por el esquema no-lineal, de alta precisión y no-oscilatorio [C1], en esta estancia también se diseñó un esquema de subdivisión del mismo tipo, pero en un entorno *tri-variante* para el refinamiento de datos tomográficos vóxel.

En la estancia con las profesoras Costanza Conti (U. Firenze) y Lucia Romani (U. Milano-Bicocca), surgió de manera natural la pregunta de si las ideas involucradas en [C6] pueden extenderse a un contexto multi-variante. Hemos definido un esquema bivariante que reproduce superficies trigonométricas, y que por tanto puede usarse para dibujar esferas, elipsoides, hiperboloides y paraboloides, o cual-

quier composición por partes de todas ellas gracias a la localidad de los esquemas de subdivisión.

La colaboración iniciada con las profesores C. Conti y L. Romani puede continuar de varias maneras. Por un lado, las ideas subyacentes del esquema de subdivisión [C6] pueden generalizarse para la reproducción de polinomios exponenciales, y esto reportaría beneficios en ciertas aplicaciones. Por otro lado, una limitación que presentan en general los esquemas reproductores es la necesidad de que los datos provengan de una malla subyacente conocida. Pensamos que algunas ideas de [C6] pueden emplearse para definir esquemas reproductores que no precisan del conocimiento previo de la malla.

El trabajo realizado durante esta tesis doctoral refuerza la relevancia de las matemáticas en otros ámbitos, científicos o no. La colaboración con el equipo de investigación FUSCHROM, de Química Analítica, ha resultado muy beneficiosa para ambas partes, y tenemos diversas propuestas de investigación para el futuro. Cabría destacar que el equipo ha adquirido recientemente una nueva máquina cromatográfica que genera datos bi-dimensionales, que pueden entenderse como imágenes. FUSCHROM está muy interesado en desarrollar nuevos métodos y algoritmos para el análisis y procesado de este tipo de señales, que permitirán extraer más información de las muestras de laboratorio.

We summarize here the collection of works that has been carried out along the PhD thesis, which were cited in this summary. The set of papers which conform the compilation of articles of the PhD thesis are attach at the end of the book.

Ferramentes matemàtiques multi-escala per al processament de senyals

Resum

Sergio López Ureña
Directora: Rosa Donat

VNIVERSITAT DE VALÈNCIA

Doctorat en Matemàtiques
Juliol de 2019, Burjassot (València)

Agraïments

No voldria deixar passar aquesta oportunitat per agrair a aquelles persones que han fet els meus estudis de doctorat una experiència increïble, en tots els seus aspectes, i que recordaré amb afecte tota la vida.

En primer lloc, vull expressar la meva gratitud a Rosa Donat. Durant aquests anys m'ha ensenyat que tan important és desenvolupar nou coneixement matemàtic com saber transmetre els resultats a altres persones. Admire el seu esforç i dedicació per buscar la millor manera d'expressar les idees que volem transmetre.

Als professors Mª Celia García i José Ramón Torres, del departament de Química Analítica, per les investigacions tan interessants que proposen i que hem gaudit colze a colze. Amb vosaltres he après que cadascú venim d'una escola, amb les nostres pròpies metodologies i coneixements, i que en un equip amb voluntat totes elles sumen.

Als meus companys doctorands, gairebé tots doctors ja, pel dia a dia, per recordar-me constantment que el doctorat és molt més que escriure una tesi.

A les professores Costanza Conti (U. Firenze) i Lucia Romani (U. Milano-Bicocca), de les què admire la seua relació col·laborativa i perseverant, per mantenir durant tres mesos una rutina constant de treball amb mi, i que van donar lloc a una investigació que em va apassionar.

Al professor Tomas Sauer i a tots els meus companys del departament de FORWISS, per acollir-me a la Universitat de Passau, tractant-me com un més de l'equip, i compartir amb mi la cultura alemanya. Em fascina el vostre ambient de treball i la relació que manteniu amb els altres companys.

A tots els treballadors del departament, de la facultat i de la Universitat de València, per crear un ambient de treball agradable i motivador.

Al Ministeri de Ciència, Innovació i Universitats, pel suport econòmic amb l'ajuda FPU14/02216.

A la meua família i als meus amics, pel seu suport i la seua confiança, per ajudar-me a veure la meua situació amb perspectiva i a aprendre a apreciar-la. Especialment als meus pares i el meu germà, per escoltar-me i entendre'm sempre.

I per descomptat, a Lydia, mil gràcies per acompanyar-me cada dia en aquesta aventura. Gràcies a tu acabe el doctorat amb la sensació que açò no ha fet més que començar.

Índex

1 Introducció: El processament de senyals

S'entén per *senyal* una funció que conté informació sobre el comportament (o els atributs) d'un sistema o fenomen [77]. Enregistraments de veu, electrocardiogrames, fotografies, cromatogrames, mapes meteorològics, etc. són exemples de senyals.

Els senyals solen contenir grans quantitats de dades, de les quals pot ser difícil d'extreure la informació rellevant per a una aplicació concreta. Mitjançant el *processament* s'obté un senyal d'eixida, a partir d'un senyal d'entrada, per tal d'extreure informació o de modificar determinades característiques. Per exemple, a la gravació d'una peça musical se li pot aplicar la transformada de Fourier [66] per convertir el senyal original temps-intensitat en un altre freqüència-intensitat. A un electrocardiograma, se li poden esborrar les interferències que presente [73], procedents potser dels instruments mèdics propers.

El processament sol ser més ràpid i simple en una representació *dispersa* del senyal, on uns pocs coeficients revelen la informació més característica o representativa. Aquestes representacions poden construir-se descomponent els senyals utilitzant formes d'ona elementals triades en una família determinada.

Les sèries de Fourier són un exemple clàssic de representació de funcions, on la família associada és $\{\exp(\imath nt)\}_{n\in\mathbb{N}}$. Si una funció és $\mathcal{C}^\alpha$, és a dir, és $\alpha \in \mathbb{N}$ vegades diferenciable i contínua, aleshores la successió dels seus coeficients de Fourier, $\{c_n\}_{n\geq 0}$, decreix amb velocitat $O(n^{-\alpha})$. Per tant, per a funcions suaus els coeficients de la sèrie decauen ràpidament i, si s'ignoren aquells coeficients amb un valor inferior a un cert llindar, s'aconsegueix una representació *dispersa*, basada en uns pocs termes de la sèrie de Fourier. No obstant això, per a funcions amb discontinuïtats, els coeficients tot just minven a ritme $O(n^{-1})$.

Les tècniques de processament de senyals basades en la descomposició de Fourier s'han convertit en eines bàsiques en una gran varietat d'aplicacions en molts camps de la ciència.

Malgrat la capacitat de l'anàlisi de Fourier per representar funcions suaus, es tracta d'una descomposició global. Una singularitat aïllada domina el comportament de tots els coeficients en la descomposició d'una funció discontínua. Les aproximacions de funcions basades en el truncament de la sèrie de Fourier presenten l'anomenat *Fenomen de Gibbs*, que consisteix en l'aparició d'oscil·lacions al voltant de la discontinuïtat, i que no desapareixen per molts termes que s'afegeixen a la sèrie truncada. El fenomen de Gibbs fa que la transformada de Fourier deixe de ser una eina útil en molts contextos. En particular, en el tractament d'imatges es manifesta com un artefacte visualment identificable al voltant dels contorns dels objectes (que es poden interpretar com discontinuïtats en el senyal) [59].

En llenguatges naturals, una família de paraules (diccionari) més rica ajuda a construir oracions més curtes i més precises. De manera similar, són necessàries

famílies adequades per construir representacions disperses de senyals complexes. Una "bona" representació pot millorar el reconeixement de patrons, la compressió de dades o la reducció de soroll. El descobriment de famílies ortogonals de funcions locals en temps-freqüència [38, 67, 71], entre elles les bases de wavelets ortogonals, va obrir les portes a un altre tipus de transformacions capaces d'obtenir representacions 'locals' tant en espai com en freqüència. Mallat va utilitzar les bases ortonormals de wavelets com a eina matemàtica per descriure l''increment d'informació' entre diferents 'nivells de resolució' en una descomposició multi-escala d'una imatge. Aquest tipus de descomposicions s'obtenien a partir de *esquemes de filtrat*.

En un esquema típic de filtrat a dues bandes, el senyal d'entrada és convolucionat separadament amb dos filtres diferents, un de pas baix i un altre de pas alt. El filtre pas baix rebutja la part altament oscil·latòria de les dades, deixant 'passar' únicament la part de baixa freqüència. Mentre que un filtre pas alt extrau la part d'alta freqüència del senyal. Una vegada obtingudes les dues successions resultants de les convolucions, ambdues són *sub-mostrejades* (*downsampled*) per retenir els elements parells (o senars) i rebutjar la resta. Al procés d'obtenir les dues noves senyals a través dels operadors de convolució i sub-mostreig se li coneix com *anàlisi* o *codificació*. A partir de les dues contribucions es pot reconstruir el senyal original, o almenys una aproximació a ella, sent necessari aplicar un *sobre-mostreig* (*up-sampling*) i dos nous filtres, adequats i acords al sub-mostreig i als filtres inicials. Aquestes operacions constitueixen el procés de *síntesi* o *descodificació*.

A partir de 1986, Meyer i Mallat van desenvolupar les bases dels *Anàlisi de Multi-resolució*, i les transformacions multi-escala associades. Una representació multi-escala d'un senyal (discret) es compon d'una aproximació a baixa resolució del senyal original més una successió de 'detalls', que són la diferència d'informació entre nivells de resolució consecutius, del que ràpidament es va establir la relació entre aquestes eines matemàtiques i els esquemes de filtrat llargament utilitzats en Enginyeria Electrònica per al processament de senyals [38].

En els articles que componen aquesta tesi doctoral s'utilitza l'entorn de multi-resolució dissenyat per A. Harten a finals dels anys 80. En la següent secció descrivim amb cert detall aquest entorn, que en cert sentit es pot considerar una generalització de les anàlisis de multi-resolució basats en la teoria de wavelets.

2 L'entorn de multi-resolució de Harten

El desenvolupament de la teoria de wavelets [68, 67, 70, 71] es pot considerar com el punt de partida de les descomposicions locals per escales, que han tingut sens dubte un gran impacte en diversos camps de la ciència.

La construcció de les *bases de wavelets* es recolzen en funcions que resulten del desplaçament i la dilatació d'una única funció 'mare'. Inicialment, el disseny i anàlisi utilitzava de manera intensiva tècniques de Anàlisi Harmònica, que feien difícil l'extensió a dominis delimitats i geometries generals.

En [55, 56], A. Harten desenvolupa un entorn general de multi-resolució per a la representació de dades (HMRF, de l'anglès *Harten 's MultiResolution Framework*), que es recolza en la Teoria de l'Aproximació, permetent una millor adaptació a tot tipus de geometries.

L'HMRF [10, 11, 12, 13, 56] se sustenta en dos *operadors*, la *decimació* i la *predicció*, que relacionen dades discretes associades a dos nivells de resolució consecutius. Des d'un punt de vista algebraic, la decimació i la predicció es poden considerar simplement com operadors que connecten espais vectorials lineals de dimensió numerable, V^k, que representen d'alguna manera els diferents nivells de resolució de les dades (discrets) que es pretenen analitzar (la resolució s'incrementa amb k), és a dir,

$$D_{k+1}^k : V^{k+1} \longrightarrow V^k, \quad P_k^{k+1} : V^k \longrightarrow V^{k+1}.$$

Mentre que la decimació D_{k+1}^k s'assumeix lineal, no hi ha restriccions *a priori* al HMRF perquè la predicció P_k^{k+1} ho siga. L'única restricció entre els dos operadors en aquest entorn és la *consistència*, açò és

$$D_{k+1}^k P_k^{k+1} = I_{V^k}, \tag{1}$$

on I_{V^k} és l'operador identitat en V^k.

Si en el nivell més fi es tenen les dades $v^L \in V^L$, mitjançant la decimació es poden restringir les dades a espais més grossers, definint recursivament

$$v^k := D_{k+1}^k v^{k+1}, \qquad k = 0, \ldots, L-1.$$

La predicció realitza el procés oposat, és a dir, genera noves dades en un espai més fi. L'error associat a l'ús d'un operador de predicció es mesura mitjançant l'expressió

$$e^k := v^{k+1} - P_k^{k+1} D_{k+1}^k v^{k+1} = (I_{V_{k+1}} - P_k^{k+1} D_{k+1}^k) v^{k+1}. \tag{2}$$

Cal notar que v^{k+1} es pot calcular a partir de v^k i e^k:

$$v^{k+1} = e^k + P_k^{k+1} v^k,$$

sent $e^k \neq 0$, en general. De la condició de compatibilitat (1), es dedueix que

$$D_{k+1}^k e^k = (D_{k+1}^k - D_{k+1}^k P_k^{k+1} D_{k+1}^k) v^{k+1} = (D_{k+1}^k - D_{k+1}^k) v^{k+1} = 0,$$

és a dir, e^k pertany al nucli[1] de D_{k+1}^k, i.e. $e^k \in \ker(D_{k+1}^k)$.

Siga d^k el conjunt de coeficients que expressen e^k respecte d'alguna base de $\ker(D_{k+1}^k)$, que conté la informació no-redundant de e^k . Denotant $e^k = E_k d^k$, es pot escriure

$$v^{k+1} = E_k d^k + P_k^{k+1} v^k.$$

Per tant, es té una bijecció $v^{k+1} \leftrightarrow (v^k, d^k)$ que, si s'aplica reiteradament, permet obtenir una *descomposició multi-escala* de v^L:

$$v^L \leftrightarrow (v^{L-1}, d^{L-1}) \leftrightarrow (v^{L-2}, d^{L-2}, d^{L-1}) \leftrightarrow \cdots \leftrightarrow (v^0, d^0, d^1, \ldots, d^{L-1}). \quad (3)$$

Per tant, formalment les representacions de multi-resolució propostes per Harten tenen la mateixa estructura que les transformades de wavelet estàndard, de manera que els passos bàsics de codificació i descodificació incorporats a (3) es poden re-interpretar en termes d'Enginyeria Electrònica, com els passos d'anàlisi i síntesi d'un esquema de filtrat per bandes amb reconstrucció exacta. L'operador D_{k+1}^k exerciria el paper d'un filtre de pas baix i de l'operador $I_{V^{k+1}} - P_k^{k+1} D_{k+1}^k$, s'obtindria un filtre de pas d'alt.

El punt de vista de Harten en introduir el HMRF és que la forma en què s'han generat les dades determina la seua *naturalesa* i ha de proporcionar una configuració adequada per efectuar una anàlisi multi-escala dels mateixes. A la pràctica, la seua proposta es basava en la construcció de D_{k+1}^k i P_k^{k+1} a través de dos operadors que relacionen les dades discretes amb les funcions de les que provenen: la *discretització* i la *reconstrucció*. L'operador de discretització $\mathcal{D}_k$ és un operador lineal que extrau informació discreta de les funcions d'un cert espai $\mathcal{F}$, $\mathcal{D}_k : \mathcal{F} \longrightarrow V_k$, en un nivell de resolució especificat per una malla Ξ^k. L'operador de reconstrucció $\mathcal{R}_k : V_k \longrightarrow \mathcal{F}$ genera una aproximació a una funció donada $f \in \mathcal{F}$ a partir dels valors discrets $\mathcal{D}_k f$. Entre aquests operadors s'ha de complir la condició de *consistència*, o *compatibilitat*,

$$\mathcal{D}_k \mathcal{R}_k = I_{V^k}. \quad (4)$$

Donada una successió d'operadors de discretització i reconstrucció amb les característiques anteriors, és possible definir els operadors de decimació i predicció de la següent manera:

$$D_{k+1}^k := \mathcal{D}_k \mathcal{R}_{k+1}, \qquad P_k^{k+1} := \mathcal{D}_{k+1} \mathcal{R}_k. \quad (5)$$

[1] El *nucli*, o *kernel*, d'una aplicació lineal $A : V \longrightarrow W$ són el conjunt de vectors en V, la imatge dels quals és 0,

$$\ker(A) := \{v \in V \ : \ Av = 0\}.$$

Sembla haver-hi una dependència explícita de D_{k+1}^k a $\mathcal{R}_k$, però és fàcil comprovar que la decimació és completament independent a la reconstrucció quan la successió de discretitzacions és *imbricada*, és a dir, si

$$\forall f \in \mathcal{F} \ : \ \mathcal{D}_{k+1} f = 0 \quad \Longrightarrow \quad \mathcal{D}_k f = 0.$$

En aquest cas, l'operador D_{k+1}^k està caracteritzat per la següent propietat

$$D_{k+1}^k(\mathcal{D}_{k+1} f) = \mathcal{D}_k f \qquad \forall f \in \mathcal{F}. \tag{6}$$

El procés de discretització escollit per a cada aplicació està relacionat amb la naturalesa de les dades. En moltes aplicacions, les dades s'associen a una malla subjacent, que pot considerar-se com el nivell més fi dins d'una jerarquia de malles imbricades. En la solució numèrica d'algunes equacions diferencials ordinàries i parcials, per exemple, la solució discreta representa una aproximació als *valors puntuals* de la solució exacta en una malla. En altres aplicacions, com el tractament d'imatges mèdiques, l'escàner té una resolució fixa i la informació en nivells de resolució més baixos s'ha d'obtenir *agrupant* les dades per simular els efectes de l'escàner (i la imatge) en una menor resolució. En aquest cas, les dades estan *naturalment* associats a les *cel·les* que defineixen la malla subjacent.

Una vegada especificada la configuració, l'elecció d'un operador de reconstrucció apropiat proporciona el pas clau per a la configuració d'un esquema de multiresolució. El procés de reconstrucció es troba al cor mateix d'un esquema de multi-resolució construït "a la Harten" [10, 11].

Un avantatge de l'HMRF en comparació amb altres entorns multi-escala és que la reconstrucció pot ser no-lineal, donant lloc a operadors de predicció P_k^{k+1} no-lineals o adaptats a geometries concretes, el que es tradueix en operadors de *síntesi* (o descodificació) amb aquestes característiques. La possibilitat d'utilitzar operadors de reconstrucció no-lineals, capaços d'obtenir representacions precises de funcions discontínues, ha estat utilitzada amb èxit en aplicacions en què intervenen dades amb forts gradients, com és el cas de les imatges [3, 5, 6, 7, 8, 29].

En el context dels articles recollits en aquesta memòria, és important remarcar que els operadors de predicció poden ser entesos com *esquemes de subdivisió recursiva*, que és una tècnica àmpliament utilitzada en el refinament de dades i en el disseny assistit per ordinador (CAD, de l'anglès *Computer Aided Design*), a la qual es dedicarà la Secció 3.

En particular, hem aplicat l'HMRF en àmbits tan diversos com Química Analítica [C4, C5] (Secció 5), la millora de dissenys de seccions planes en un context d'optimització multi-paramètrica [C3] (Secció 4.1) o l'estimació de paràmetres estadístics (*Uncertainty Quantification*, Secció 3.3) [C2]. En aquests casos, hem utilitzat fonamentalment l'entorn de *valors puntuals*, que introduïm en la secció 2.1. Per a una revisió més completa de l'HMRF, recomanem, per exemple, [11, 56].

2.1 L'entorn de valors puntuals

Reconstruir una funció, a partir d'un conjunt discret de dades representatiu de la funció, és un problema clàssic en Teoria d'Aproximació que depèn de la interpretació que s'assigne a les dades discretes. Probablement el cas més simple siga la interpolació de *valors puntuals*, en què es busca reconstruir una aproximació a una funció desconeguda a partir d'una taula de valors de la mateixa. Els operadors de reconstrucció de l'entorn *interpolador* o de *valors puntuals* de l'HMRF estan basats en l'ús de tècniques d'interpolació. A continuació es descriurà aquest entorn de multi-resolució en el cas univariant, que és l'utilitzat en els articles que constitueixen aquesta tesi doctoral.

Suposem que en el nivell de resolució k les dades estan naturalment associades a la malla $\Xi^k = (\xi_i^k)_{i\in\mathbb{Z}}$, on

$$\xi_i^k = \xi_{2i}^{k+1}, \qquad \forall i \in \mathbb{Z}. \tag{7}$$

Si els operadors de discretització associats es defineixen com

$$\mathcal{D}_k f := (f(\xi_i^k))_{i\in\mathbb{Z}},$$

s'obté una successió imbricada, ja que $\Xi^k \subset \Xi^{k+1}$. D'aquesta successió d'operadors de discretització, de (6) és fàcil veure que s'obté l'operador decimació

$$v^k := D_{k+1}^k v^{k+1} := (v_{2i}^{k+1})_{i\in\mathbb{Z}}.$$

És a dir, la decimació coincideix amb l'operador *downsampling* de teoria de filtres.

La predicció es pot relacionar amb la reconstrucció, com s'indica en (5):

$$\begin{aligned}(P_k^{k+1}v^k)_{2i} &= (\mathcal{D}_{k+1}\mathcal{R}_k v^k)_{2i} = (\mathcal{R}_k v^k)(\xi_{2i}^{k+1}) = (\mathcal{R}_k v^k)(\xi_i^k),\\ (P_k^{k+1}v^k)_{2i+1} &= (\mathcal{D}_{k+1}\mathcal{R}_k v^k)_{2i+1} = (\mathcal{R}_k v^k)(\xi_{2i+1}^{k+1}).\end{aligned}$$

La condició de consistència (1), $D_{k+1}^k P_k^{k+1} = I_{V^k}$, en aquest entorn es tradueix en

$$(P_k^{k+1}v^k)_{2i} = (D_{k+1}^k P_k^{k+1} v^k)_i = v_i^k, \qquad \forall i \in \mathbb{Z}, \tag{8}$$

és a dir, la predicció ha de conservar tots els valors de v^k en les posicions parelles. Per tant

$$f(\xi_i^k) = v_i^k = (\mathcal{R}_k v^k)(\xi_i^k),$$

és a dir, $\mathcal{R}_k$ interpola les dades v^k en la malla Ξ^k. Si es denota per $\mathcal{I}(\xi, v^k)$ a la tècnica d'interpolació utilitzada per definir l'operador reconstrucció, es pot escriure

$$(\mathcal{R}_k v^k)(\xi) = \mathcal{I}(\xi, v^k).$$

D'una banda, cal notar que l'error de predicció associat (2), e^k, és zero en les posicions parells com a conseqüència de (8),

$$e^k_{2i} = v^{k+1}_{2i} - (P^{k+1}_k D^k_{k+1} v^{k+1})_{2i} = v^k_i - (P^{k+1}_k v^k)_{2i} = 0, \qquad \forall i \in \mathbb{Z}.$$

D'altra banda,

$$e^k_{2i+1} := v^{k+1}_{2i+1} - (P^{k+1}_k v^k)_{2i+1} = f(\xi^{k+1}_{2i+1}) - \mathcal{I}(\xi^{k+1}_{2i+1}, v^k), \qquad \forall i \in \mathbb{Z},$$

és a dir, en les posicions senars l'error és precisament l'error d'interpolació de la tècnica d'interpolació.

Aquestes identitats suggereixen definir els coeficients d^k (la informació no-redundant de e^k) com el conjunt d'errors d'interpolació produïts en les posicions senars de la malla, $d^k_i := e^k_{2i+1}$.

La bijecció $v^{k+1} \leftrightarrow (v^k, d^k)$, que permet transvasar la informació entre nivells resolució, pren la forma

$$\begin{aligned} &v^{k+1}_{2i} = v^k_i,\ v^{k+1}_{2i+1} = d^k_i + \mathcal{I}(\xi^{k+1}_{2i+1}, v^k), \qquad \forall i \in \mathbb{Z}, \\ &\qquad\qquad \updownarrow \\ &v^k_i = v^{k+1}_{2i},\ d^k_i = v^{k+1}_{2i+1} - \mathcal{I}(\xi^{k+1}_{2i+1}, v^k), \qquad \forall i \in \mathbb{Z}. \end{aligned}$$

Un exemple clàssic d'operador reconstrucció en aquest entorn és la interpolació polinòmica a trossos que segueix: Per a cada interval de la malla, $[\xi^k_i, \xi^k_{i+1}]$, es pren el polinomi $Q_{i,k}$ de grau $2q-1$ tal que $Q_{i,k}(\xi^k_{i+j}) = v^k_{i+j}$, $j = -q+1, \ldots, q$, i es defineix

$$(\mathcal{R}_k v^k)(\xi) := Q_{i,k}(\xi), \qquad \forall \xi \in [\xi^k_i, \xi^k_{i+1}]. \tag{9}$$

Si s'expressa $Q_{i,k}$ mitjançant la base de Lagrange,

$$Q_{i,k}(\xi) = \sum_{j=-q+1}^{q} v^k_{i+j} L_j(2^k \xi - i), \quad L_j(\xi) := \prod_{\substack{l=-q+1 \\ l \neq j}}^{q} \frac{\xi - l}{j - l},$$

l'operador de predicció que s'obté a partir d'aquesta tècnica d'interpolació pot expressar-se com

$$(P^{k+1}_k v^k)_{2i+1} = Q_{i,k}(\xi^{k+1}_{2i+1}) = \sum_{j=-q+1}^{q} v^k_{i+j} a^q_j. \tag{10}$$

sent $a^q_j := L_j\left(\frac{1}{2}\right)$, $j = -q+1, \ldots, q$, que depèn de j i q però és independent de i i k. S'aprecia clarament la seua dependència lineal respecte de les dades.

Exemples clàssics d'operadors de predicció són els basats en les interpolacions polinòmiques de grau u i grau tres ($q = 1, 2$):

$$(P_k^{k+1} v^k)_{2i+1} = \frac{1}{2} v_i^k + \frac{1}{2} v_{k+1}^k, \qquad (q = 1) \qquad (11)$$

$$(P_k^{k+1} v^k)_{2i+1} = -\frac{1}{16} v_{i-1}^k + \frac{9}{16} v_i^k + \frac{9}{16} v_{i+1}^k - \frac{1}{16} v_{i+2}^k. \qquad (q = 2) \qquad (12)$$

En el context de la subdivisió recursiva, aquests operadors de predicció polinòmics es coneixen com els esquemes de Deslauriers-Dubuc [41], sobre els quals es tornarà a parlar en la Secció 3.1.

En algunes aplicacions es necessita reconstruir funcions a partir de dades discretes que contenen variacions ràpides de magnitud, les quals poden estar associades a discontinuïtats de la funció subjacent. Si s'utilitza una interpolació lineal en aquests casos, podrien aparèixer oscil·lacions no desitjades al voltant de la discontinuïtat (similar al fenomen de Gibbs). Existeixen operadors de reconstrucció no-lineals especialment dissenyats per a interpolar les dades sense produir oscil·lacions, com ara les reconstruccions ENO [57], WENO [65], PCHIP [9] y PPH [4, 5].

3 Esquemes de subdivisió

Els esquemes de subdivisió són una tècnica per al refinament recursiu de dades. La subdivisió recursiva destaca per la seva simplicitat inherent, que ha promogut el seu ús com a eina de reconstrucció i aproximació, en particular en la generació eficient de corbes i superfícies en el disseny assistit per ordinador (CAD) [40].

Un esquema de subdivisió [22, 43] és un procés iteratiu que, a partir d'un conjunt de dades inicial f^0, calcula una successió de conjunts de dades $(f^k)_{k\geq 0}$, associats a nivells de refinament cada vegada mes elevats. Cada nou conjunt de dades, f^{k+1}, és definit a partir de l'anterior, f^k, mitjançant un conjunt finit d'operacions 'senzilles', que és el que converteix aquests processos en eines molt eficients en diverses aplicacions.

S'utilitzarà l'exemple 'canònic' per introduir els conceptes més rellevants en aquesta teoria: l'esquema de les poligonals[2]. En el cas *univariant*, on les dades són successions bi-infinites $f^k = (f_i^k)_{i=-\infty}^{+\infty}$, l'esquema està format per dos *regles* de subdivisió que distingeixen entre posicions parells i imparells:

$$f_{2i}^{k+1} := f_i^k, \quad f_{2i+1}^{k+1} := \frac{1}{2} f_i^k + \frac{1}{2} f_{i+1}^k, \qquad \forall i \in \mathbb{Z}. \tag{13}$$

La successió de dades f^k pot associar-se a la malla $2^{-k}\mathbb{Z}$, la qual cosa fa possible entendre les dades com a punts $(i2^{-k}, f_i^k)$ a $\mathbb{R}^2$ i donar una interpretació geomètrica de (13): A la iteració $k+1$ es conserven els punts de la iteració k,

$$((2i)2^{-(k+1)}, f_{2i}^{k+1}) = (i2^{-k}, f_i^k),$$

i s'afegeix la mitjana de cada parell de punts consecutius,

$$((2i+1)2^{-(k+1)}, f_{2i+1}^{k+1}) = \frac{1}{2}(i2^{-k}, f_i^k) + \frac{1}{2}((i+1)2^{-k} f_{i+1}^k).$$

En general, un esquema de subdivisió defineix cada nova dada generada mitjançant un conjunt d'operacions senzilles que involucren una quantitat finita de dades de la iteració anterior. Aquesta propietat, coneguda com *localitat*, implica que possibles pertorbacions en les dades es propaguen controladament al llarg de les iteracions [43]. És a dir, si es modifica una dada inicial, els punts afectats per tal variació estan en una regió acotada de la malla, en aquest exemple, l'interval obert $(i-1, i+1)$ [43].

Cal notar que tots els punts generats en el nivell $k+1$ pertanyen a la poligonal de vèrtexs $(i2^{-k}, f_i^k)_{i\in\mathbb{Z}}$. D'això deduïm que els punts de qualsevol iteració estan sobre la poligonal inicial, amb vèrtexs $(i, f_i^0)_{i\in\mathbb{Z}}$ i que l'esquema de subdivisió

[2] Una poligonal és una funció definida a trossos per polinomis de primer grau. El punt que connecta dos segments rectes és conegut com a vèrtex.

genera (asimptòticament) tots els valors d'aquesta funció poligonal en els punts diàdics, que és un conjunt dens en els reals. En aquests casos, es diu que l'esquema de subdivisió *convergeix* a una funció F, coneguda com *funció límit*, que depèn de les dades inicials.

L'esquema de subdivisió (13) és un exemple d'esquema *univariant*, és a dir, que les dades sobre els quals actua són successions, $f^k = (f_i^k)_{i\in\mathbb{Z}}$. Si l'esquema convergeix, a partir d'una successió inicial $f^0 = (f_i^0)_{i\in\mathbb{Z}}$ s'obté una funció límit d'una variable (v.g. corbes). A la pràctica, només cal executar una quantitat finita de refinaments (iteracions) per 'generar' la funció, per exemple per a la seva visualització en una aplicació concreta [43].

Els esquemes de subdivisió *multi-variants* manipulen dades estructurades mitjançant malles multi-dimensionals, $f^k = (f_\alpha^k)_{\alpha\in\mathbb{Z}^s}$, $s > 1$, i poden convergir a funcions de diverses variables. En aquest cas, es pot parlar de convergència a superfícies i fins i tot varietats diferenciables, sempre que el conjunt de dades inicial i les regles de subdivisió siguen adequades. Això afegeix diversitat al tipus de situacions en què s'aplica la subdivisió recursiva [5, 16, 58, 81].

A més de la convergència, una altra propietat fonamental dels esquemes de subdivisió és la *estabilitat*, que determina la magnitud de les modificacions en la funció límit derivades de pertorbacions en les dades inicials.

Per esquemes com (13), on f_i^{k+1} depèn *linealment* de les dades en f^k, l'estudi de la convergència es realitza d'una manera sistemàtica mitjançant la teoria corresponent [22, 43], que està ben establerta i consolidada. En aquest cas, l'estabilitat és una conseqüència de la convergència. Quan les regles de subdivisió són *no-lineals* [2, 5, 15, 29], es requereixen tècniques completament diferents i la teoria subjacent és molt més recent [28, 36, 37, 42, 45, 51, 54, 61, 79]. La convergència i l'estabilitat d'aquests processos recursius són essencials per a les seves aplicacions i, per tant, el seu estudi ha estat i segueix sent un tema d'investigació actiu.

Cal observar que la convergència, localitat i estabilitat dels esquemes de subdivisió té un impacte positiu en la generació i manipulació d'objectes geomètrics. En termes del CAD, donat un polígon de control f^0, un esquema de subdivisió defineix una corba associada. Si es modifica un punt del polígon de control, la corba només canvia en una regió ben delimitada, que envolta el punt modificat. Aquestes propietats permeten que un dissenyador gràfic puga retocar i perfilar els seus models de manera local, sense alterar altres parts que potser ja estiguen del seu gust, la qual cosa resulta atractiva en el modelatge d'objectes. En el cinema d'animació, la subdivisió recursiva es va utilitzar per primera vegada en el curt de Pixar " Geri 's Game" [40], a finals dels noranta.

Quan les dades inicials provenen d'una funció suau, un requeriment bastant habitual és que la subdivisió recursiva genere una aproximació *prou precisa* de la funció original. La capacitat d'aproximació, o de precisió, de les dades generades

per un esquema de subdivisió és un factor important a tenir en compte en moltes aplicacions.

Un altre requeriment que resulta útil en moltes aplicacions és la *reproducció*, és a dir, la reconstrucció exacta d'una família de funcions. Per exemple, l'esquema de subdivisió (13) és capaç de *reproduir* funcions poligonals.

La reproducció de polinomis [25, 33, 43] i de polinomis exponencials [26, 32, 34, 44, 78] és interessant des d'un punt de vista teòric, ja que està relacionada amb altres propietats de l'esquema de subdivisió (com l'aproximació i la convergència), però també des de la vessant pràctica, ja que permet dibuixar amb exactitud corbes rellevants en geometria, com les seccions còniques.

Els esquemes de subdivisió lineals capaços de reproduir polinomis exponencials tenen, necessàriament, regles de subdivisió que varien al llarg de les iteracions, i per això es denominen *no-estacionaris*. En [C6] es demostra que els esquemes estacionaris *no-lineals* també poden reproduir polinomis exponencials i que presenten alguns avantatges respecte als esquemes lineals no-estacionaris.

En altres situacions pot ser important establir certes restriccions sobre f^k i sobre la funció límit. Per exemple, si les dades representen una quantitat física que ha de tenir un valor real positiu, cal que les noves dades generades per subdivisió siguen també positives. De manera anàloga, pot requerir-se la preservació de *monotonia* o la *convexitat*. El manteniment d'alguna d'aquestes propietats es pot entendre com a casos particulars de *preservació de la forma* [61], el que ha motivat el desenvolupament d'esquemes *no-lineals* específicament dissenyats per mantenir una (o més) d'aquestes propietats. Alguns exemples que es poden trobar a la literatura són els esquemes *essencialment no-oscil·latoris* [28] (obtinguts a partir de certs operadors de predicció no-lineals en el HMRF [57, 65]) o esquemes que preserven la monotonia [15, 62] o la convexitat [5] en les dades. La investigació realitzada durant aquest projecte de tesi ha donat lloc a dos nous esquemes de subdivisió en aquesta línia [C1, C6].

Els esquemes de subdivisió poden considerar-se el nucli de la meva activitat durant els estudis de doctorat. D'una banda, els operadors de subdivisió utilitzats com a operadors de predicció dins el HMRF s'han aplicat en diversos contextos: Estimació de paràmetres estadístics (Uncertainty Quantification) [C2], optimització de seccions planes en el disseny de velers de competició [C3] i millora d'eines d'anàlisi en Química Analítica [C4, C5]. D'altra banda, hem desenvolupat i estudiat teòricament nous esquemes no-lineals amb propietats orientades a aplicacions específiques [C1, C6].

Atès que els articles que conformen aquesta tesi doctoral se centren en l'estudi i l'ús d'esquemes de subdivisió *univariants, uniformes* i *binaris*, a la Secció 3.1 es definiran les propietats principals de la subdivisió recursiva en aquest context.

3.1 El cas univariant

Definició 1. Un *esquema de subdivisió univariant* és una successió d'operadors $\{S^k\}_{k\geq 0}$, $S^k : \ell_\infty(\mathbb{Z}) \longrightarrow \ell_\infty(\mathbb{Z})$, que permet definir recursivament una successió de successions fitades $(f^k)_{k\geq 0} \subset \ell_\infty(\mathbb{Z})$ a partir d'una successió inicial fitada de dades $f^0 = (f_i^0)_{i\in\mathbb{Z}} \in \ell_\infty(\mathbb{Z})$, de la següent manera:

$$f^{k+1} := S^k f^k, \qquad k \geq 0.$$

L'esquema de les poligonals (13), definit en la secció 3, és un exemple d'esquema *uniforme*, *binari* i *univariant*. Els operadors de subdivisió d'aquesta classe es defineixen a partir de dues regles de subdivisió Ψ_0^k y Ψ_1^k que distingeixen entre dades parells i senars,

$$f_{2i}^{k+1} = \Psi_0^k(f_{i-q}^k, f_{i-q+1}^k, \ldots, f_{i+q}^k), \qquad f_{2i+1}^{k+1} = \Psi_1^k(f_{i-q}^k, f_{i-q+1}^k, \ldots, f_{i+q}^k),$$

per a cert $q > 0$. Si $\{\Psi_0^k, \Psi_1^k\}_{k\geq 0}$ són funcions lineals, aleshores l'esquema és *lineal*. Si les regles $\{\Psi_0^k\}_{k\geq 0}$ són tals que $f_{2i}^{k+1} = f_i^k$, aleshores és *interpolador*. Si les regles de subdivisió $\{\Psi_0^k, \Psi_1^k\}_{k\geq 0}$ són les mateixes al llarg de les iteracions, i.e. no depenen de k, l'esquema de subdivisió és *estacionari*. En aquest cas es denotarà:

$$\Psi_0 := \Psi_0^k, \quad \Psi_1 := \Psi_1^k, \quad S := S^k.$$

L'esquema de subdivisió (13) és estacionari, lineal i interpolador.

Els esquemes de subdivisió utilitzats en aplicacions pràctiques han de ser *convergents*, un concepte que es defineix a continuació de manera precisa.

Definició 2. Un esquema de subdivisió és *convergent* si

$$\forall f^0 \in \ell_\infty(\mathbb{Z}) \quad \exists S^\infty f^0 \in \mathcal{C}(\mathbb{R}) \; : \quad \lim_{k\to\infty} \sup_{i\in\mathbb{Z}} |f_i^k - (S^\infty f^0)(i2^{-k})| = 0.$$

Es denota per $S^\infty : \ell_\infty(\mathbb{Z}) \longrightarrow \mathcal{C}(\mathbb{R})$ a l'operador que envia cada dada inicial f^0 a la seua corresponent funció límit.

Es pot demostrar [43] que aquesta definició de convergència és equivalent al fet que les funcions poligonals $\mathbb{P}^k$ tals que $\mathbb{P}^k(i2^{-k}) = f_i^k$ formen una successió de Cauchy.

Una altra propietat igualment important és la *estabilitat*, que es defineix formalment com segueix.

Definició 3. Un esquema de subdivisió convergent és *estable* si l'operador $S^\infty : \ell_\infty(\mathbb{Z}) \longrightarrow \mathcal{C}(\mathbb{R})$ és Lipschitz continu:

$$\exists L > 0 \; : \; \|S^\infty f^0 - S^\infty g^0\|_\infty \leq L\|f^0 - g^0\|_\infty, \qquad \forall f^0, g^0 \in \ell_\infty(\mathbb{Z}).$$

En esquemes lineals, és fàcil veure que l'estabilitat és una conseqüència directa de la convergència de l'esquema. No obstant això, la situació és molt diferent en el cas no-lineal.

En general, la teoria d'esquemes de subdivisió tracta d'inferir propietats de les funcions límits $S^\infty f^0$ a partir de la definició de les regles de subdivisió. D'aquesta manera es poden estudiar propietats bàsiques com la convergència o l'estabilitat de l'esquema, i també d'altres que poden ser convenients en diverses aplicacions, com la regularitat, la capacitat d'aproximació, la reproducció exacta, la preservació de la forma, etc. Tot això a partir de l'expressió de les regles de subdivisió Ψ_j^k.

En algunes aplicacions, com en el disseny assistit per ordinador, pot interessar que les corbes que es generen a partir d'esquemes de subdivisió tinguen certa regularitat.

Definició 4. Un esquema de subdivisió convergent és $\mathcal{C}^\alpha$ si [3]

$$S^\infty f^0 \in \mathcal{C}^\alpha, \qquad \forall f^0 \in \ell_\infty(\mathbb{Z}).$$

També sol ser important conèixer la *capacitat d'aproximació* d'un esquema de subdivisió, en el sentit de la següent definició.

Definició 5. Un esquema convergent té *ordre d'aproximació* r si per a qualsevol funció prou suau F,

$$\|F(h\,\bullet) - S^\infty f^0\|_\infty \leq Ch^r, \qquad f^0 = F|_{h\mathbb{Z}}, \quad \forall 0 < h < h_0.$$

És a dir, l'ordre d'aproximació d'un esquema de subdivisió, mesura com es redueix l'error en intentar aproximar F aplicant l'esquema de subdivisió sobre $(F(ih))_{i\in\mathbb{Z}}$, sent l'espaiat de la malla h prou petit.

A més, en algunes aplicacions es desitja reconstruir certes funcions $F \in \mathcal{F}$ de manera exacta, i no aproximada. Pot ser d'interès pràctic la reconstrucció de circumferències, el·lipses, hipèrboles, etc. i això pot fer-se eficientment si s'empren esquemes de subdivisió que *reproduïsquen* la classe de funcions que defineixen les corbes anteriors [34].

Definició 6. Un esquema de subdivisió convergent *reprodueix* una família de funcions $\mathcal{F}$, si per a qualsevol funció $F \in \mathcal{F}$ l'esquema convergeix a F a partir de les dades inicials $f^0 = F|_{\mathbb{Z}}$:

$$S^\infty F|_{\mathbb{Z}} = F, \qquad \forall F \in \mathcal{F}.$$

Els esquemes de Deslauriers-Dubuc [41] són un exemple clàssic d'esquemes lineals interpoladors que reprodueixen polinomis de grau arbitràriament alt (però fix).

[3] Es defineix $\mathcal{C}^\alpha$ com el conjunt de funcions α vegades diferenciables i contínues.

Es poden construir a partir de la interpolació polinòmica a trossos descrita a la Secció 2.1. Com pot observar-se en (10), són esquemes estacionaris, perquè els coeficients a_j^q de la combinació lineal que defineixen les seues regles són independents de k.

Des d'un punt de vista teòric, la reproducció de polinomis i de polinomis exponencials és interessant perquè està relacionada amb la capacitat d'aproximació i la suavitat de l'esquema [31, 35, 43, 61]. A més, un esquema lineal que reproduïsca polinomis exponencials és necessàriament *no-estacionari* [26, 32, 34, 44], i les seues regles de subdivisió Ψ_0^k, Ψ_1^k depenen de certs paràmetres implicats en l'expressió de l'espai de polinomis exponencials que reprodueix. No obstant això, en [C6] s'obté un esquema de subdivisió *no-lineal* que reprodueix funcions trigonomètriques (que són un cas particular de polinomis exponencials) les regles del qual són estacionàries i no depenen dels paràmetres esmentats.

És ben sabut que els esquemes de subdivisió lineals i no-estacionaris poden assolir aquest objectiu [34, 44]. Però la seua aplicació requereix la determinació pràctica dels paràmetres, que defineixen les regles dependents del nivell, mitjançant el processament previ de les dades disponibles [17, 44].

Atès que diferents seccions còniques requereixen diferents regles de refinament per garantir la reproducció exacta, no és possible reproduir una forma composta, per parts, per diverses funcions trigonomètriques amb el mateix esquema lineal. En [C6] es mostra que la reproducció exacta de diferents formes còniques es pot aconseguir utilitzant el mateix esquema no-lineal, sense cap processament previ de les dades.

Per a aplicacions on, per exigències de la naturalesa del problema, les dades són positives, o monòtones, o convexes, l'esquema de subdivisió ha de conservar el tipus (o la *forma*) de les dades.

Definició 7. Un esquema de subdivisió *preserva* (estrictament) la *positivitat* de les dades, o equivalentment, és (estrictament) positiu, si per a qualsevol $f \in \ell_\infty(\mathbb{Z})$,

$$f_i(>) \geq 0 \ \forall i \in \mathbb{Z} \quad \rightarrow \quad Sf_i > 0 \quad \forall i \in \mathbb{Z}.$$

Un esquema *preserva* (estr.) la *monotonia*, o és monòton, si

$$\nabla f_i(>) \geq 0 \ \forall i \in \mathbb{Z} \quad \rightarrow \quad \nabla S f_i > 0 \ \forall i \in \mathbb{Z},$$

on $\nabla : \ell_\infty(\mathbb{Z}) \longrightarrow \ell_\infty(\mathbb{Z})$ és l'operador en diferències finites, $\nabla f_i := f_{i+1} - f_i$.

Un esquema *preserva* (estr.) la *convexitat*, o és convex, si

$$\nabla^2 f_i(>) \geq 0 \ \forall i \in \mathbb{Z} \quad \rightarrow \quad \nabla^2 S f_i > 0 \ \forall i \in \mathbb{Z}.$$

Els esquemes de subdivisió interpoladors, d'alta precisió i que preserven de la forma de les dades resulten de gran utilitat en certes aplicacions [5], la qual cosa ha

motivat a diversos autors a dissenyar esquemes no-lineals amb aquestes propietats [15, 62]. Hem abordat aquest tema durant el projecte de tesi, havent definit dos nous esquemes de subdivisió [C1, C6].

La convergència d'un esquema de subdivisió lineal és una condició suficient per a l'estabilitat, però no ho és en el cas no-lineal. És més complex demostrar que un esquema no-lineal és convergent i estable. Per això, es disposa de resultats teòrics [2, 5, 15, 37, 39, C1, 54] que asseguren aquestes propietats si es compleixen certs requisits. Per introduir aquests resultats, que hem emprat en [C1, C6], cal definir el concepte de *esquema en diferències*. Cal esmentar que els esquemes no-lineals solen ser estacionaris, de manera que els resultats estan limitats a aquest cas.

Definició 8. Un esquema de subdivisió S té *esquema en diferències* d'ordre n si existeix un esquema $S^{[n]}$ tal que

$$\nabla^n S = S^{[n]} \nabla^n.$$

Cal destacar que l'existència de l'esquema en diferències $S^{[n]}$ no està garantida, exceptuant el cas lineal, on qualsevol esquema convergent té esquema en diferències d'ordre $n = 1$.

L'esquema en diferències permet analitzar el comportament de les diferències finites de les dades f^k a partir de $S^{[n]}$, mitjançant l'expressió

$$\nabla^n f^k = (S^{[n]})^k \nabla^n f^0.$$

Aquesta propietat és clau en l'anàlisi de la convergència, tant en el cas lineal com no-lineal.

En [2, 5, 15, 37, 39, C1, 54, C6], els autors construeixen i analitzen diversos esquemes de subdivisió no-lineals que poden descriure com una pertorbació no-lineal d'un esquema convergent lineal T:

$$Sf = Tf + \mathcal{F}(\nabla^n f), \qquad \forall f \in \ell_\infty(\mathbb{Z}), \tag{14}$$

on $\mathcal{F} : \ell_\infty(\mathbb{Z}) \to \ell_\infty(\mathbb{Z})$ és un operador (possiblement no-lineal). Si hi ha $T^{[n]}$, la qual cosa és fàcil de comprovar [43], llavors un esquema de la forma (14) té esquema en diferències d'ordre n. En concret aquest és

$$S^{[n]} f = T^{[n]} f + \nabla^n \mathcal{F}(f), \qquad \forall f \in \ell_\infty(\mathbb{Z}). \tag{15}$$

Per analitzar la seva convergència i estabilitat s'han emprat resultats específics de [2, 15].

3.2 [Adv. Comput. Math., 2017] A family of non-oscillatory 6-point interpolatory subdivision schemes

Un esquema lineal interpolador, amb un ordre d'aproximació $r > 2$, és segur que produirà oscil·lacions i perdrà tota la seva precisió quan les dades presenten variacions súbites. Es mostra un exemple d'això a la Figura 1. Utilitzant l'esquema de 6 punts de Deslauriers-Dubuc (DD), $S_{3,3}$, que és lineal i té ordre 6, s'ha obtingut una funció límit que oscil·la al voltant del salt. Això vol dir que als voltants de la discontinuïtat l'esquema perd la seua capacitat per reconstruir la funció.

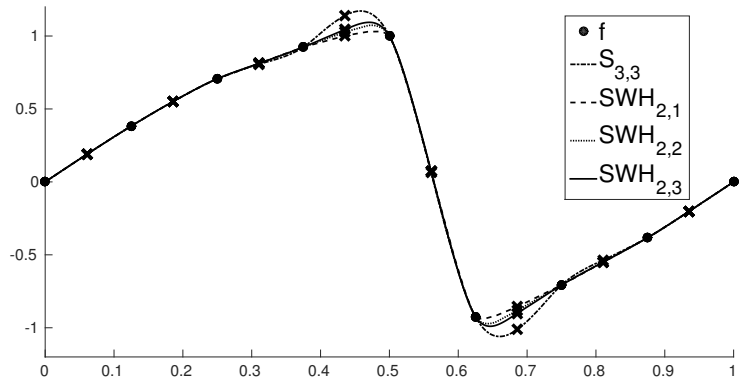


Fig. 1: A partir de les dades inicials (•) es generen funcions límit mitjançant diversos esquemes de subdivisió. Les ics (×) són les dades generades després d'una iteració.

Diverses tècniques d'interpolació polinòmica per segments s'han considerat en la literatura per construir esquemes de subdivisió interpoladors que eviten oscil·lacions no desitjades. Exemples de tals esquemes són els esquemes ENO-WENO [28, 57, 65], l'esquema PPH [4, 5], els esquemes Power$_p$ [2, 14, 36] i els esquemes de conservació de la forma descrits en [61]. Aquests últims deuen el seu caràcter no-oscil·latori a l'ús de certes mitjanes no-lineals escollides amb bon criteri.

En aquest treball, es proposa i analitza una nova família d'esquemes de subdivisió no-lineals, els $S\text{WH}_{p,q}$, que poden considerar versions no-oscil·latòries de l'esquema $S_{3,3}$, igual que els esquemes Power$_p$ es consideren versions no-lineals i no-oscil·latòries de l'esquema interpolador DD de 4 punts. De fet, el seu disseny està relacionat amb el dels esquemes Power$_p$.

Es demostra que els nous esquemes reprodueixen exactament polinomis de grau 3 i que la distància en norma infinit l'esquema DD de 6 punts és petita en regions suaus, com s'aprecia a la Figura 1.

A més, es prova que el primer i el segon esquema en diferències estan ben

definits per a cada membre de la família, el que permet donar una prova simple de la convergència uniforme d'aquests esquemes i també estudiar la seva estabilitat com en [15, 54].

No obstant això, l'estudi teòric de l'estabilitat basat en els resultats de [54] no és concloent en el cas d'estudi, per la qual cosa, es realitzen una sèrie d'experiments numèrics que semblen indicar que només uns pocs membres de la nova família d'esquemes són estables.

D'altra banda, les exhaustives proves numèriques revelen que, per a dades suaus, l'ordre d'aproximació i la regularitat de la funció límit poden ser similars als de l'esquema de DD de 6 punts i superiors als obtinguts amb els esquemes de Power_p.

3.3 [Applied Mathematics and Nonlinear Sciences, 2017] High-accuracy approximation of piecewise smooth functions using the truncation and encode approach

Nombrosos processos físics i industrials poden simular mitjançant una equació en derivades parcials (EDP). Per exemple, en el disseny d'apèndixs per a velers, s'ha de simular el flux de l'aigua al voltant del perfil per calcular el coeficient d'arrossegament. Amb aquesta finalitat es fan servir les equacions de Navier-Stockes, la resolució numèrica de les quals és complexa i el temps de càlcul es dispara en augmentar la precisió.

És possible que la simulació depenga de múltiples paràmetres físics el valor dels quals és variable, per exemple la velocitat del veler i la inclinació de la proa, i per tant s'han de tractar com a variables aleatòries. Llavors, el coeficient d'arrossegament no és únic, sinó que depèn del valor de cada paràmetre. A la pràctica, es pot establir un mallat i resoldre l'EDP associada a cada parell de valors de la malla velocitat-inclinació. Com es pot imaginar, el cost computacional és desorbitat si la malla és molt fina, i s'ha de plantejar alguna estratègia.

En [1, 49] es va definir un mètode anomenat *Truncat and Encode* (TE, truncar i codificar), que aprofita l'entorn de multi-resolució de Harten per aproximar adaptativament la solució d'una EDP i estimar certs paràmetres estadístics en el context de Uncertainty Quantification. A grans trets, en cada nivell de resolució es decideix si resoldre l'EDP o interpolar la solució amb les dades existents, reduint així el temps de càlcul. La decisió es basa en la precisió que va tenir la interpolació en el nivell anterior, i convé escollir una tècnica d'interpolació d'alt ordre d'aproximació i preferiblement no-oscil·latòria. De fet, la interpolació és equivalent a un operador de subdivisió, de manera que els esquemes PCHIP [15] i $S\text{WH}_{p,q}$ [C1] poden aplicar-se i són molt recomanables.

En aquest article, s'analitza l'algoritme ET aplicat a l'aproximació de funcions

i, en particular, el seu rendiment per a funcions suaus per parts. Es duen a terme alguns experiments numèrics, comparant el rendiment de l'algoritme quan es fan servir diferents tècniques d'interpolació lineal i no-lineal i es proporcionen algunes recomanacions que ens semblen útils per aconseguir un alt rendiment de l'algoritme. Els resultats indiquen que per incrementar el rendiment de TE és convenient utilitzar esquemes de subdivisió d'alt ordre d'aproximació.

4 Estratègies multi-escala en optimització a gran escala

L'optimització [74] és una eina important en la presa de decisions i en l'anàlisi de sistemes físics. En un procés d'optimització es deu, en primera instància, identificar una *funció objectiu*, que mesure el rendiment del sistema que s'estiga estudiant, per exemple temps, energia, beneficis econòmics, o qualsevol quantitat o combinació d'elles que puga representar-se amb un únic número. Aquesta funció depèn de certes característiques del sistema, anomenades *variables* o *incògnites*.

La finalitat del procés és trobar els valors de les variables que optimitzen la funció objectiu. Sovint les variables estan restringides, o limitades, d'alguna manera. Per exemple, les quantitats que representin la massa d'objectes no poden ser negatives.

Al procés d'identificar els objectius, les variables i les restriccions d'un problema donat se li coneix com *modelatge*. La construcció d'un model adequat és el primer pas, sovint el més important, en el procés d'optimització. Un cop obtingut el model, la solució es troba mitjançant l'aplicació d'un algoritme d'optimització, habitualment amb l'assistència d'un ordinador.

En termes matemàtics, un problema d'optimització consisteix a minimitzar (o maximitzar) una *funció objectiu* F dins d'un espai de possibles solucions *factibles*, diguem X. És a dir, trobar $u_{\min} \in X$ tal que $F(u_{\min}) \leq F(u)\ \forall u \in X$.

Les funcions objectiu han d'estar definides en un espai de dimensió finita, i.e. $F : X \subset \mathbb{R}^N \longrightarrow \mathbb{R}$, per poder abordar el problema d'optimització computacionalment mitjançant algun algoritme, anomenat *optimitzador*. Quan la quantitat de variables N és gran, es parla de *optimització a gran escala*. Aquest tipus de problemes apareixen sovint a partir de la discretització d'un problema de dimensió infinita, per exemple en el context del disseny òptim, en el control òptim, en l'estimació de paràmetres en sistemes governats per EDP [19, 64, 80] i en el processament d'imatges [24, 23, 76, 75, 82].

No hi ha un optimitzador universal, més aviat tota una col·lecció d'ells, cada un dels quals fet a mida per a algun tipus de problema. La responsabilitat d'escollir l'algoritme apropiadament per a una aplicació concreta recau sobre l'usuari. Aquesta decisió és important, ja que determinarà si el problema es resol ràpidament o lentament i, certament, si la solució serà trobada.

En optimització a gran escala sovint es poden aplicar optimitzadors que comporten un esforç computacional prohibitiu causa de la gran quantitat de variables involucrades.

L'èxit dels mètodes multigrid [20, 21, 52, 53, 69], com a resoledor eficient de EDPs el·líptiques discretitzades, va impulsar el desenvolupament de mètodes iteratius multi-nivell en optimització [18, 27, 30, 47, 48, 50, 60, 72] des de mitjans dels anys 80.

La idea que comparteixen aquests mètodes multi-nivell és l'aplicació d'un op-

timitzador particular per resoldre problemes auxiliars reduïts de menor dimensió, derivats de la discretització del problema infinit-dimensional amb menor exactitud, i que per tant són més ràpids de resoldre (en termes de càlcul).

Tot i que els mètodes multi-nivell comparteixen una estructura comuna, s'ha realitzat l'esforç de desenvolupar per separat les versions multi-nivell dels optimitzadors més comuns [30, 47, 48, 50]. En aquesta tesi doctoral es proposa una estructura multi-nivell basada en el HMRF que permet implementar qualsevol optimitzador de manera arbitrària. En altres paraules, aquest mètode permet tractar el optimitzador com a una 'caixa negra', permetent a l'usuari utilitzar l'optimitzador que més li convinga entre aquells dels quals dispose.

El meu treball en aquest camp sorgeix d'unes pràctiques realitzades en IS&3D ENG[4], durant els meus estudis de màster. El problema proposat va ser el de millorar el rendiment de seccions planes que es fan servir en el disseny d'apèndixs de velers de competició. En particular, es volia reduir l'arrossegament[5] de timons, quilles i bulbs dins l'aigua. El repte es va plantejar com un problema d'optimització en el qual es volia reduir l'arrossegament teòric tenint en compte certes restriccions físiques i de disseny.

La funció objectiu incloïa una simulació CFD realitzada per una rutina externa del tipus caixa negra (`xfoil` [6]). IS&3D ENG va proposar utilitzar els optimitzadors integrats en Matlab.

Durant el procés de millora s'havia de modificar la secció mitjançant pertorbacions suaus i sense oscil·lacions, introduint en primer lloc variacions globals a la secció per, a mesura que fora millorant, incidir en els detalls més locals. Aquesta idea encaixava amb l'estructura multi-escala de l'HMRF. La síntesi d'aquest plantejament ens va conduir a definir una nova estratègia d'optimització.

Part del treball realitzat durant la col·laboració amb IS&3D ENG ha donat lloc a diverses participacions en reunions científiques (veure el currículum vitae) i a la publicació inclosa en aquesta memòria [C3, Secció 4.1].

A través de la col·laboració amb l'equip d'investigació FUSCHROM[7] de Cromatografia Líquida[8], es va aplicar exitósament aquesta estratègia d'optimització en un context completament diferent [C4, Secció 5.1]. En aquest cas, l'objectiu era maximitzar la separació entre substàncies que han estat injectades en un dispositiu de separació. Novament, la funció objectiu era força complexa, ja que conté una simulació química, i es va emprar com optimitzador la rutina `patternsearch` de

[4] www.is3de.com

[5] La resistència al moviment d'un objecte a través d'un *flux*, com l'aire o l'aigua.

[6] XFOIL és un programa interactiu per al disseny i anàlisi de perfils aerodinàmics aïllats subsònics.

[7] https://sites.google.com/site/fuschrom/

[8] Una tècnica que permet separar, identificar i quantificar cada substància present en una barreja .

Matlab.

4.1 [Progress in Industrial Mathematics at ECMI 2016] A novel multi-scale strategy for multi-parametric optimization

El moviment d'un veler és conseqüència de l'equilibri existent entre les forces aerodinàmiques, induïdes pel vent sobre les veles, i les forces hidrodinàmiques, resultants del contacte de l'aigua amb les parts submergides del vaixell, que són el casc i els apèndixs. Cada apèndix compleix una funció. Per exemple, el timó marca la direcció del moviment, la quilla evita desplaçaments laterals i el bulb influeix en el moment dreçador[9], Evitant que el vaixell escore[10].

El modelatge d'aquests apèndixs es realitza a partir de la seva secció transversal, una corba plana tancada anomenada *perfil*, com el que es mostra a la Figura 2. L'objectiu que s'aborda en aquest treball [C3] és modificar (optimitzar) un perfil donat per reduir l'arrossegament amb l'aigua, subjecte a restriccions de diversa índole. N'hi ha estructurals, com per exemple que el perfil ha de tenir una longitud determinada, i n'hi ha físiques, com que el coeficient de sustentació ha d'estar comprès entre dos valors admissibles. Les restriccions dependran de la finalitat de l'apèndix (timó, quilla ...). Vist d'una altra manera, les restriccions imposades en l'optimització converteixen un perfil inicial qualsevol en el tipus d'apèndix desitjat.

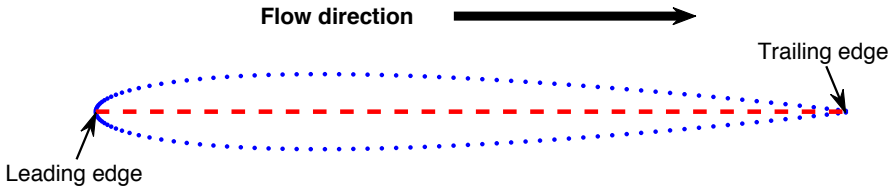


Fig. 2: Un exemple de perfil: el NACA0010 descrit per $N = 129$ punts.

L'estratègia que es planteja proporciona una successió de solucions sub-òptimes, una a cada nivell de resolució, de manera que en l'últim pas es resol el problema d'optimització complet (a gran escala), però amb una estimació inicial molt més propera a la solució desitjada que la proporcionada inicialment (que sovint es tria arbitràriament, però també pot ser facilitada per l'usuari), fent que l'esforç de càlcul requerit per l'optimitzador triat siga factible.

Aquesta tècnica aplica exhaustivament un esquema de subdivisió, que ha de ser escollit tenint en compte la naturalesa de les dades manipulades. Ja que es

[9] Mesura la capacitat d'una embarcació per mantenir-se en posició vertical.
[10] Escorar: Inclinar un vaixell sobre un dels seus costats.

vol modificar *suaument* un perfil, però evitant produir oscil·lacions, es proposa utilitzar l'esquema de subdivisió de B-Splines d'ordre 5 [43].

En l'article s'analitza el comportament de l'algoritme aplicant-lo a un problema acadèmic, obtenint una dràstica reducció del cost computacional en comparació amb l'aplicació directa (sense estratègia multi-escala) de l'optimitzador triat.

Es planteja una optimització per al disseny d'un apèndix, on només l'estratègia multi-escala va ser capaç de proporcionar resultats satisfactoris.

5 Aplicacions en cromatografia líquida

La cromatografia líquida és una tècnica utilitzada en Química Analítica per separar, identificar i quantificar cadascun dels soluts presents en una barreja.

En inserir la barreja juntament amb un dissolvent al llarg d'un tub, anomenat *columna*, els diferents soluts de la barreja flueixen (*precipiten*) a diferents velocitats quan interaccionen amb el medi porós de l'interior de la columna .

Si l'experiment es configura correctament, cada solut surt pel final de la columna de manera separada. Un sensor registra la quantitat de barreja que surt en cada instant de temps, i la gràfica que s'obté amb aquesta relació temps-quantitat[11] s'anomena *cromatograma*. Els diferents soluts de la barreja apareixen com 'pics' al cromatograma, com s'il·lustra en les Figures 3 i 4.

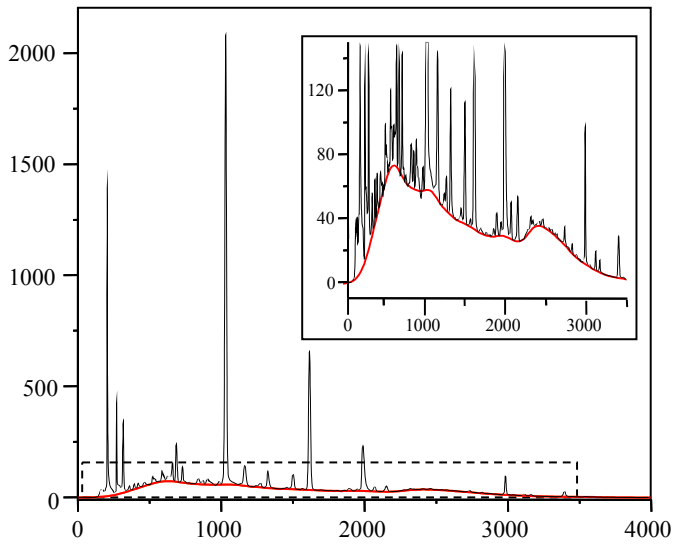


Fig. 3: Exemple de detecció de línia base. En el requadre superior dret es mostra l'ampliació de la zona marcada amb un rectangle discontinu.

Alguns problemes que es poden trobar per quantificar correctament la quantitat de cada solut en la barreja, i que hem abordat durant el doctorat mitjançant tècniques matemàtiques, són: la presència d'una línia base, que es deriva de l'ús del dissolvent; la presència de *soroll*, que prové de factors tant ambientals com a propis de la química de la barreja; i el solapament d'uns pics amb altres. Per evitar aquest últim problema, i així obtenir pics ben *resolts*, es necessita preestablir la concentració de dissolvent a injectar a la columna en cada instant de temps.

[11] La unitat de mesura és aquella proporcionada pel dispositiu que mesura quanta llum no ha estat absorbida per la barreja en sortir de la columna. Si, per exemple, el receptor de llum és electrònic, la unitat de mesura seria mil·livolts.

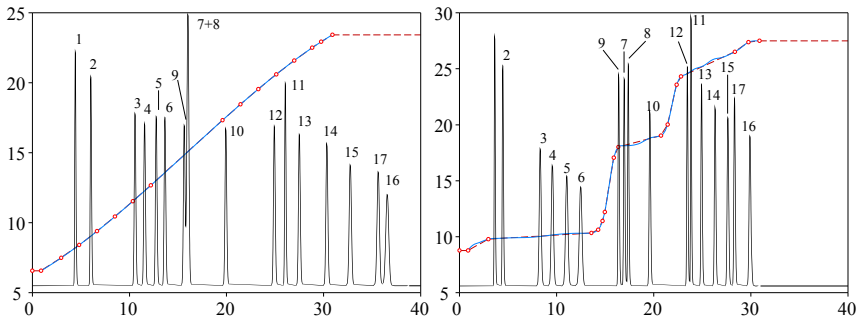


Fig. 4: Augment de la resolució dels pics, associats a 17 aminoàcids essencials, i reducció del temps de sortida. A l'esquerra, el programa de gradient inicial. A la dreta, el optimitzat. A l'eix horitzontal es mostra el temps de sortida (en minuts), en el vertical la concentració del dissolvent injectat a la columna.

La col·laboració amb projectes de Química Analítica es va iniciar durant els meus estudis de grau. Vaig col·laborar amb l'equip de recerca CLECEM[12], a l'assignatura optativa 'pràctiques en empresa' sota la tutela de Guillem Ramis Ramos. Fruit d'aquesta col·laboració, vam publicar un article [C8](anterior als meus estudis de doctorat) sobre l'anàlisi i la classificació de cromatogrames.

Al començament del meu doctorat, es va iniciar una col·laboració amb el grup de recerca FUSCHROM[13], també de Química Analítica. Estaven interessats en el posprocessat de senyals cromatogràfiques. En particular, es volia eliminar certes *línies base* presents habitualment en les dades mitjançant algun algoritme matemàtic implementat computacionalment.

A través dels contactes de la meva directora de tesi, Rosa Donat, vam conèixer un algoritme, BEADS, que estava proporcionant excel·lents resultats. Està basat en l'optimització d'una funció objectiu, dissenyada a consciència, mitjançant la tècnica *majorització-minimització* [46, 63]. Cal dir que BEADS també està preparat per eliminar *soroll*.

Aplicant BEADS a diferents cromatogrames van aparèixer algunes limitacions i dificultats associades al seu ús. Es va plantejar un seguit de procediments per aplicar correctament i fàcilment aquest algoritme en [C7] (no inclòs en el compendi d'articles). A la Figura 3 es mostra un exemple de detecció de línia base mitjançant el procediment que es planteja. Un cop detectada, tan sols cal sostreure-la.

Posteriorment, es va plantejar una nova investigació: trobar una nova manera de dissenyar, eficientment, *programes de gradient*. Matemàticament parlant,

[12] https://www.uv.es/clecem/
[13] https://sites.google.com/site/fuschrom/

consisteix a trobar una funció que maximitze la *resolució* dels pics.

El procediment que fins al moment s'emprava consistia a considerar una funció poligonal arbitrària amb un nombre de vèrtexs fix. Mitjançant l'ús d'algun optimitzador, v.g. un algoritme genètic, es determinava la posició òptima dels vèrtexs.

Aquesta estratègia requeria molt de temps de càlcul i una quantitat de nodes molt limitada. En tractar-se d'un problema a gran escala, es va plantejar l'ús de l'estratègia d'optimització multi-nivell de la Secció 4 fruit de la col·laboració amb l'empresa IS&3D ENG. Es van obtenir resultats molt satisfactoris, com es recull en el següent article [C4, Secció 5.1].

5.1 [J. Chromatogr. A, 2018] Gradient design for liquid chromatography using multi-scale optimization

El disseny de *programes de gradient*, on s'especifica a la màquina cromatogràfica la concentració de dissolvent que s'ha d'introduir a la columna en cada instant de temps, és essencial per obtenir pics ben resolts, sense solapaments, i així poder mesurar correctament la quantitat de cada solut que forma la mescla. L'objectiu que es planteja és trobar la funció programa de gradient tal que maximitza la resolució alhora que es verifiquen una sèrie de condicions, necessàries per a la correcta implementació al laboratori.

S'ha aplicat exitósament l'optimització multi-escala a aquest problema, aconseguint no només una resolució alta, sinó també la reducció del temps de sortida, que es tradueix en menys hores de treball per al personal i l'instrumental de laboratori. Es mostra un exemple d'optimització a la Figura 4.

5.2 [J. Chromatogr. A, 2019] Enhancement in the computation of gradient retention times in liquid chromatography using root-finding methods

En aquest treball [C5] es proposa un nou mètode per a la simulació de la posició dels pics en funció del programa de gradient. Aquest tipus de simulacions són necessàries per dur a terme estudis com l'anterior [C4, Secció 5.1].

El valor t que resol l'equació integral

$$f(t) = \int_0^{g(t)} h(\tau) d\tau,$$

per a certes funcions f, g, h definides a partir de les condicions de l'experiment, representa la posició en l'eix de les abscisses en el qual apareix un determinat pic

del cromatograma, o en termes químics, el temps que tarda un solut en sortir de la columna cromatogràfica.

L'enfocament que s'emprava fins al moment consistia en discretitzar la integral de la següent manera,

$$\int_0^{g(n\delta)} h(t)dt \approx \delta \sum_{i=0}^{n-1} h(g(i\delta)), \qquad n \in \mathbb{N}$$

sent habitualment $\delta = 10^{-3}$, i trobar el valor de n de manera que la suma anterior estava el més a prop possible de $f(n\delta)$. Aleshores, es deduïa que $t \approx n\delta$.

En primer lloc, l'aproximació a la integral que s'estava utilitzant era molt pobre, ja que es basava en aproximar h mitjançant funcions esglaonades, quan a més, en alguns casos, h tenia primitiva. Com a millora, es proposa utilitzar la primitiva per reduir enormement el cost computacional i, en cas de no haver primitiva, aproximar h mitjançant algun polinomi i emprar la primitiva del polinomi.

En segon lloc, la manera en què es trobava el valor n era molt rudimentària. Tan sols s'incrementava el seu valor fins que es verifiqués certa condició de parada. Tenint una primitiva (aproximada) de h, diguem H, la resolució numèrica de l'equació integral es pot entendre com trobar el zero de la funció

$$F(t) = f(t) - H(g(t)) + H(0).$$

La meva proposta va ser aplicar un algoritme de recerca de zeros, derivat del mètode de Newton i del mètode de la bisecció, que combinava la velocitat de convergència amb les garanties de convergència de tots dos algoritmes.

Aquest nou enfocament permet, no només calcular el temps de retenció molt més ràpid, sinó també incrementar la precisió, que ara era inferior a 10^{-3}. En aquest treball també es fa una anàlisi teòrica per garantir que les aproximacions al temps de retenció es calculen amb un error inferior a un llindar escollit.

6 Conclusions, treball en progrés y perspectives de futur

En aquesta tesi doctoral s'han proposat, estudiat i analitzat diferents esquemes de subdivisió, prestant especial atenció al cas no-lineal.

Els esquemes de subdivisió no-lineals poden esquivar algunes de les limitacions que presenten els esquemes lineals en certes aplicacions. En aquesta memòria, s'ha obtingut un esquema no-lineal interpolador, amb alta capacitat d'aproximació, no-oscil·latori i que reprodueix polinomis de fins tercer grau [C1].

A més, s'han considerat diverses aplicacions en les que els esquemes de subdivisió juguen un paper rellevant a través de l'entorn de multi-resolució de Harten.

Hem investigat l'ús d'operadors de predicció (subdivisió) no-lineals en Uncertainty Quantification, implementats en l'estratègia *Truncate and Encode* [1, C2].

Hem proposat una nova estratègia d'optimització basada en l'entorn de multi-resolució de Harten, la qual ha estat aplicada en el disseny de seccions planes de certs apèndixs de velers de competició per reduir l'arrossegament amb l'aigua amb l'objectiu de millorar la seva eficiència [C3].

Aquesta estratègia d'optimització s'ha utilitzat també en problemes relacionats amb el tractament de senyals en Cromatografia Líquida. Hem proposat un mètode per al disseny d'*programes de gradient* [C4]. Com a conseqüència d'aquest treball, es va posar de manifest la importància de simular el *temps de sortida* eficientment. En [C5] hem plantejat una nova manera de fer-ho que redueix dràsticament el temps necessari per dissenyar un programa de gradient, al mateix temps que incrementa la precisió dels resultats.

Les publicacions [C1, C2, C3, C4, C5] representen en realitat la part consolidada del meu treball de recerca. A més d'aquestes publicacions, he d'esmentar també els següents tres articles (sotmesos a publicació).

1- Nonlinear stationary subdivision schemes that reproduce trigonometric functions. *R. Donat and S. López-Ureña*

Com s'ha exposat en la secció 3.1, el treball realitzat en aquesta tesi doctoral ha permès dissenyar una nova família d'esquemes de subdivisió interpoladora no-lineals amb la capacitat de reproduir funcions trigonomètriques i polinomis de segon grau. Evidentment, aquesta propietat és interessant per al CAD, ja que els esquemes poden reproduir formes definides a trossos mitjançant seccions còniques (circumferències, hipèrboles, el·lipses i paràboles). L'article ha estat sotmès a publicació, i després de rebre els comentaris dels revisors i realitzar les modificacions oportunes, estem esperant una resposta definitiva per a la seva publicació. Es troba actualment disponible a arXiv [C6].

2- A Multiresolution approach to solve large-scale optimization problems. *R. Donat and S. López-Ureña*

Aquest article formalitza l'estratègia d'optimització multi-escala que es planteja en [C3, Secció 4], i es compara amb altres mètodes d'optimització multinivell. A través de diversos experiments numèrics, tant uni-dimensionals com bi-dimensionals, s'estudia el seu rendiment i s'analitza l'impacte de l'operador de predicció escollit per definir l'entorn de multiresolució. S'arriba a la conclusió que és convenient emprar, com a operadors de predicció, esquemes de subdivisió d'alt ordre d'aproximació. El treball ha estat sotmès a publicació.

3- Multi-scale optimisation vs. genetic algorithms in the separation of diuretics by reversed-phase liquid chromatography. *T. Álvarez-Segura, S. López-Ureña, J.R. Torres-Lapasió and M.C. García-Alvarez-Coque*

En [C4, Secció 5.1] es posa de manifest que l'estratègia d'optimització anterior, basada en el HMRF, pot aplicar-se en el disseny de *programes de gradient*. En aquest treball es compara el seu rendiment amb un altre mètode, basat en algoritmes genètics, que és conegut per proporcionar bons resultats en aquest problema.

Es conclou que l'enfocament multi-objectiu dels algoritmes genètics és molt convenient, ja que dóna certa llibertat a l'usuari perquè decidisca que programa de gradient és més adequat. En conseqüència, podria resultar molt convenient utilitzar els algoritmes genètics com optimitzador dins de l'estratègia multi-escala. Aquesta és una possibilitat que es contempla en [C3, Secció 4], i es reserva aquesta qüestió per al futur.

Hem modificat l'article d'acord amb les indicacions dels revisors i estem esperant una decisió definitiva sobre la seva publicació.

D'altra banda, els resultats de les línies de recerca, que es van obrir durant les meues estades a Itàlia i Alemanya, segueixen en procés de redacció.

La reproducció de *polinomis exponencials*, que generalitzen les funcions trigonomètriques, en un context multi-variant va ser estudiada en una estada amb el professor Tomas Sauer (U. Passau). A més, motivat per l'esquema no-lineal, d'alta precisió i no-oscil·latori [C1], en aquesta estada també es va dissenyar un esquema de subdivisió del mateix tipus, però en un entorn *tri-variant* per al refinament de dades tomogràfics vòxel.

En l'estada amb les professores Costanza Conti (U. Firenze) i Lucia Romani (U. Milano-Bicocca), va sorgir de manera natural la pregunta de si les idees involucrades en [C6] es poden estendre a un context multi-variant.

Hem definit un esquema bivariant que reprodueix superfícies trigonomètriques, i que per tant pot usar-se per dibuixar esferes, el·lipsoides, hiperboloides i paraboloides, o qualsevol composició per parts de totes elles gràcies a la localitat dels esquemes de subdivisió.

La col·laboració iniciada amb les professors C. Conti i L. Romani pot continuar de diverses maneres.

D'una banda, les idees subjacents de l'esquema de subdivisió [C6] es poden generalitzar per a la reproducció de polinomis exponencials, i això reportaria beneficis en certes aplicacions. D'altra banda, una limitació que presenten en general els esquemes reproductors és la necessitat que les dades provinguen d'una malla subjacent coneguda. Pensem que algunes idees de [C6] es poden emprar per definir esquemes reproductors que no necessiten el coneixement previ de la malla.

El treball realitzat durant aquesta tesi doctoral reforça la rellevància de les matemàtiques en altres àmbits, científics o no. La col·laboració amb l'equip d'investigació FUSCHROM, de Química Analítica, ha resultat molt beneficiosa per a ambdues parts, i tenim diverses propostes d'investigació per al futur. Es podria destacar que l'equip ha adquirit recentment una nova màquina cromatogràfica que genera dades bi-dimensionals, que es poden entendre com imatges. FUSCHROM està molt interessat en desenvolupar nous mètodes i algoritmes per a l'anàlisi i processat d'aquest tipus de senyals, que permetran extreure més informació de les mostres de laboratori.

Contributions/Contribuciones/Contribucions

[C1] R. Donat, S. López-Ureña, and M. Santágueda. A family of non-oscillatory 6-point interpolatory subdivision schemes. *Adv. Comput. Math.*, 2017.

[C2] S. López-Ureña and R. Donat. High-accuracy approximation of piecewise smooth functions using the truncation and encode approach. *Applied Mathematics and Nonlinear Sciences*, 2(2):367–384, 2017.

[C3] R. Donat, S. López-Ureña, and M. Menec. A novel multi-scale strategy for multi-parametric optimization. In *European Consortium for Mathematics in Industry*, pages 593–600. Springer, 2016.

[C4] S. López-Ureña, J. Torres-Lapasió, R. Donat, and M. García-Alvarez-Coque. Gradient design for liquid chromatography using multi-scale optimization. *Journal of Chromatography A*, 1534:32–42, 2018.

[C5] S. López-Ureña, J. Torres-Lapasió, and M. García-Alvarez-Coque. Enhancement in the computation of gradient retention times in liquid chromatography using root-finding methods. *Journal of Chromatography A*, 1600:137–147, 2019.

[C6] R. Donat and S. López-Ureña. Nonlinear stationary subdivision schemes that reproduce trigonometric functions. 2018.

[C7] J. Navarro-Huerta, J. Torres-Lapasió, S. López-Ureña, and M. García-Alvarez-Coque. Assisted baseline subtraction in complex chromatograms using the beads algorithm. *Journal of Chromatography A*, 1507:1–10, 2017.

[C8] S. López-Ureña, M. Beneito-Cambra, R. M. Donat-Beneito, and G. Ramis-Ramos. Overlapped moving windows followed by principal component analysis to extract information from chromatograms and application to classification analysis. *Analytical Methods*, 7(7):3080–3088, 2015.

References/Referencias/Referències

[1] R. Abgrall, P. Congedo, and G. Geraci. A one-time truncate and encode multiresolution stochastic framework. *Journal of Computational Physics*, 257, Part A:19 – 56, 2014.

[2] S. Amat, K. Dadourian, and J. Liandrat. Analysis of a class of nonlinear subdivision schemes and associated multiresolution transforms. *Advances in Computational Mathematics*, 34(3):253–277, 2011.

[3] S. Amat, F. Aràndiga, A. Cohen, and R. Donat. Tensor product multiresolution analysis with error control for compact image representation. *Signal Processing*, 82(4):587–608, 2002.

[4] S. Amat, S. Busquier, and V. Candela. A polynomial approach to the piecewise hyperbolic method. *International Journal of Computational Fluid Dynamics*, 17(3):205–217, 2003.

[5] S. Amat, R. Donat, J. Liandrat, and J. C. Trillo. Analysis of a new nonlinear subdivision scheme. applications in image processing. *Foundations of Computational Mathematics*, 6(2):193–225, 2006.

[6] S. Amat, R. Donat, J. Liandrat, and J. C. Trillo. A fully adaptive multiresolution scheme for image processing. *Mathematical and computer modelling*, 46(1-2):2–11, 2007.

[7] F. Aràndiga, A. M. Belda, and P. Mulet. Point-value weno multiresolution applications to stable image compression. *Journal of Scientific Computing*, 43(2):158–182, 2010.

[8] F. Arandiga, G. Chiavassa, and R. Donat. Harten framework for multiresolution with applications: From conservation laws to image compression. *Boletín SEMA*, (31), 2009.

[9] F. Aràndiga. On the order of nonuniform monotone cubic hermite interpolants. *SIAM Journal on Numerical Analysis*, 51(5):2613–2633, 2013.

[10] F. Arandiga and R. Donat. Building adaptive multiresolution schemes within harten's framework. *Curve and Surface Fitting*, pages 19–26, 1999.

[11] F. Aràndiga and R. Donat. Nonlinear multiscale decompositions: The approach of a. harten. *Numerical Algorithms*, 23(2-3):175–216, 2000.

[12] F. Aràndiga, R. Donat, and A. Harten. Multiresolution based on weighted averages of the hat function i: Linear reconstruction techniques. *SIAM Journal on Numerical Analysis*, 36(1):160–203, 1998.

[13] F. Aràndiga, R. Donat, and A. Harten. Multiresolution based on weighted averages of the hat function ii: Nonlinear reconstruction techniques. *SIAM Journal on Scientific Computing*, 20(3):1053–1093, 1998.

[14] F. Aràndiga, R. Donat, and M. Santágueda. Weighted-power p nonlinear subdivision schemes. In *International Conference on Curves and Surfaces*, pages 109–129. Springer, 2010.

[15] F. Aràndiga, R. Donat, and M. Santágueda. The PCHIP subdivision scheme. *Applied Mathematics and Computation*, 272, Part 1:28 – 40, 2016. Subdivision, Geometric and Algebraic Methods, Isogeometric Analysis and Refinability.

[16] C. Bajaj, S. Schaefer, J. Warren, and G. Xu. A subdivision scheme for hexahedral meshes. *The visual computer*, 18(5):343–356, 2002.

[17] C. Beccari, G. Casciola, and L. Romani. A non-stationary uniform tension controlled interpolating 4-point scheme reproducing conics. *Computer Aided Geometric Design*, 24(1):1–9, 2007.

[18] S. Benson, L. McInnes, J. Moré, and J. Sarich. Scalable algorithms in optimization: Computational experiments. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 4450, 2004.

[19] A. Borzì and K. Kunisch. A globalization strategy for the multigrid solution of elliptic optimal control problems. *Optimisation Methods and Software*, 21(3):445–459, 2006.

[20] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation*, 31(138):333–390, 1977.

[21] W. Briggs. *A Multigrid Tutorial*. 1987.

[22] A. S. Cavaretta, C. A. Micchelli, and W. Dahmen. *Stationary Subdivision*. American Mathematical Society, Boston, MA, USA, 1991.

[23] R. H. Chan and K. Chen. A multilevel algorithm for simultaneously denoising and deblurring images. *SIAM Journal on Scientific Computing*, 32(2):1043–1063, 2010.

[24] T. F. Chan and K. Chen. An optimization-based multilevel algorithm for total variation image denoising. *Multiscale Modeling & Simulation*, 5(2):615–645, 2006.

[25] M. Charina and C. Conti. Polynomial reproduction of multivariate scalar subdivision schemes. *Journal of Computational and Applied Mathematics*, 240:51–61, 2013.

[26] M. Charina, C. Conti, and L. Romani. Reproduction of exponential polynomials by multivariate non-stationary subdivision schemes with a general dilation matrix. *Numerische Mathematik*, 127(2):223–254, 2014.

[27] C. Chen, Z. Wen, and Y.-x. Yuan. A general two-level subspace method for nonlinear optimization. *Journal of Computational Mathematics*, 36(6), 2018.

[28] A. Cohen, N. Dyn, and B. Matei. Quasilinear subdivision schemes with applications to eno interpolation. *Applied and Computational Harmonic Analysis*, 15(2):89–116, 2003.

[29] A. Cohen and B. Matei. Nonlinear subdivision schemes: applications to image processing. In *Tutorials on Multiresolution in Geometric Modelling*, pages 93–97. Springer, 2002.

[30] B. Colson, M. Porcelli, and P. L. Toint. Aircraft fuselage sizing with multilevel optimization. 2013.

[31] C. Conti, M. Cotronei, and L. Romani. Beyond b-splines: exponential pseudo-splines and subdivision schemes reproducing exponential polynomials. *Dolomites Research Notes on Approximation*, 10(Special_Issue), 2017.

[32] C. Conti, M. Cotronei, and T. Sauer. Factorization of hermite subdivision operators preserving exponentials and polynomials. *Advances in Computational Mathematics*, 42(5):1055–1079, 2016.

[33] C. Conti and K. Hormann. Polynomial reproduction for univariate subdivision schemes of any arity. *Journal of Approximation Theory*, 163(4):413–437, 2011.

[34] C. Conti and L. Romani. Algebraic conditions on non-stationary subdivision symbols for exponential polynomial reproduction. *J. Comput. Appl. Math.*, 236(4):543–556, September 2011.

[35] C. Conti, L. Romani, and J. Yoon. Approximation order and approximate sum rules in subdivision. *Journal of Approximation Theory*, 207:380–401, 2016.

[36] K. Dadourian. *Schémas de subdivision, analyses multirésolutions non-linéaires. Applications.* Theses, Université de Provence - Aix-Marseille I, October 2008.

[37] K. Dadourian and J. Liandrat. Analysis of some bivariate non-linear interpolatory subdivision schemes. *Numerical Algorithms*, 48(1):261–278, 2008.

[38] I. Daubechies. *Ten lectures on wavelets*, volume 61. Siam, 1992.

[39] I. Daubechies, O. Runborg, and W. Sweldens. Normal multiresolution approximation of curves. *Constructive Approximation*, 20(3):399–463, 2004.

[40] T. DeRose, M. Kass, and T. Truong. Subdivision surfaces in character animation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 85–94. ACM, 1998.

[41] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. In *Constructive approximation*, pages 49–68. Springer, 1989.

[42] T. Duchamp, G. Xie, and T. Yu. A necessary and sufficient proximity condition for smoothness equivalence of nonlinear subdivision schemes. *Foundations of Computational Mathematics*, 16(5):1069–1114, 2016.

[43] N. Dyn. Subdivision schemes in cagd. In *Advances in Numerical Analysis*, pages 36–104. Univ. Press, 1992.

[44] N. Dyn, D. Levin, and A. Luzzatto. Exponentials reproducing subdivision schemes. *Foundations of Computational Mathematics*, 3(2):187–206, 2003.

[45] N. Dyn and P. Oswald. Univariate subdivision and multi-scale transforms: The nonlinear case. In *Multiscale, Nonlinear and Adaptive Approximation*, pages 203–247. Springer, 2009.

[46] M. A. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak. Majorization–minimization algorithms for wavelet-based image restoration. *IEEE Transactions on Image processing*, 16(12):2980–2991, 2007.

[47] E. Frandi and A. Papini. Coordinate search algorithms in multilevel optimization. *Optimization Methods and Software*, 29(5):1020–1041, 2014.

[48] E. Frandi and A. Papini. Improving direct search algorithms by multilevel optimization techniques. *Optimization Methods and Software*, 30(5):1077–1094, 2015.

[49] G. Geraci, P. M. Congedo, R. Abgrall, and G. Iaccarino. A novel weakly-intrusive non-linear multiresolution framework for uncertainty quantification in hyperbolic partial differential equations. *Journal of Scientific Computing*, 66(1):358–405, 2016.

[50] S. Gratton, A. Sartenaer, and P. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization*, 19(1):414–444, 2008.

[51] P. Grohs. A general proximity analysis of nonlinear subdivision schemes. *SIAM Journal on Mathematical Analysis*, 42(2):729–750, 2010.

[52] W. Hackbusch. Convergence of multigrid iterations applied to difference equations. *Mathematics of Computation*, 34(150):425–440, 1980.

[53] W. Hackbusch. *Multi-grid methods and applications*, volume 4. 1985.

[54] S. Harizanov and P. Oswald. Stability of nonlinear subdivision and multiscale transforms. *Constructive Approximation*, 31(3):359–393, 2010.

[55] A. Harten. Discrete multi-resolution analysis and generalized wavelets. *Applied numerical mathematics*, 12(1-3):153–192, 1993.

[56] A. Harten. Multiresolution representation of data: A general framework. *SIAM Journal on Numerical Analysis*, 33(3):1205–1256, 1996.

[57] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. In *Upwind and high-resolution schemes*, pages 218–290. Springer, 1987.

[58] M. Hofer, H. Pottmann, and B. Ravani. Subdivision algorithms for motion design based on homologous points. In *Advances in Robot Kinematics*, pages 235–244. Springer, 2002.

[59] R. Hovden, Y. Jiang, H. L. Xin, and L. F. Kourkoutis. Periodic artifact reduction in fourier transforms of full field atomic resolution images. *Microscopy and Microanalysis*, 21(2):436–441, 2015.

[60] M. Kočvara and S. Mohammed. A first-order multigrid method for bound-constrained convex optimization. *Optimization Methods and Software*, 31(3):622–644, 2016.

[61] F. Kuijt. *Convexity preserving interpolation - stationary nonlinear subdivision and splines*. PhD thesis, Enschede, October 1998.

[62] F. Kuijt and R. van Damme. Monotonicity preserving interpolatory subdivision schemes. *Journal of Computational and Applied Mathematics*, 101(1):203 – 229, 1999.

[63] K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of computational and graphical statistics*, 9(1):1–20, 2000.

[64] R. M. Lewis and S. G. Nash. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM Journal on Scientific Computing*, 26(6):1811–1837, 2005.

[65] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of computational physics*, 115(1):200–212, 1994.

[66] N. C. Maddage, K. Wan, C. Xu, and Y. Wang. Singing voice detection using twice-iterated composite fourier transform. In *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, volume 2, pages 1347–1350. IEEE, 2004.

[67] S. Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[68] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of $l^2(\mathbb{R})$. *Transactions of the American mathematical society*, 315(1):69–87, 1989.

[69] S. F. McCormick. *Multigrid methods*. 1987.

[70] Y. Meyer. Ondelettes et operateurs, i, ii, iii. *Hermann, Paris*, 1991, 1990.

[71] Y. Meyer. *Wavelets and operators*, volume 1. Cambridge university press, 1995.

[72] S. G. Nash. A multigrid approach to discretized optimization problems. *Optimization Methods and Software*, 14(1-2):99–116, 2000.

[73] X. Ning and I. W. Selesnick. Ecg enhancement and qrs detection based on sparse derivatives. *Biomedical Signal Processing and Control*, 8(6):713 – 723, 2013.

[74] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[75] S. Oh, C. A. Bouman, and K. J. Webb. Multigrid tomographic inversion with variable resolution data and image spaces. *IEEE Transactions on Image Processing*, 15(9):2805–2819, 2006.

[76] S. Oh, A. B. Milstein, C. A. Bouman, and K. J. Webb. A general framework for nonlinear multigrid inversion. *IEEE Transactions on image Processing*, 14(1):125–140, 2005.

[77] R. Priemer. *Introductory signal processing*, volume 6. World Scientific Publishing Company, 1990.

[78] T. Sauer. Kernels of discrete convolutions and application to stationary subdivision operators. *Acta Applicandae Mathematicae*, 145(1):115–131, 2016.

[79] S. Schaefer, E. Vouga, and R. Goldman. Nonlinear subdivision through nonlinear averaging. *Comput. Aided Geom. Des.*, 25(3):162–180, March 2008.

[80] M. Vallejos and A. Borzì. Multigrid optimization methods for linear and bilinear elliptic optimal control problems. *Computing*, 82(1):31–52, 2008.

[81] J. Wallner and N. Dyn. Convergence and c1 analysis of subdivision schemes on manifolds by proximity. *Computer Aided Geometric Design*, 22(7):593–622, 2005.

[82] J. C. Ye, C. A. Bouman, K. J. Webb, and R. P. Millane. Nonlinear multigrid algorithms for bayesian optical diffusion tomography. *IEEE Transactions on Image Processing*, 10(6):909–922, 2001.

Published articles/Artículos publicados/Articles publicats

Adv Comput Math (2017) 43:849–883
DOI 10.1007/s10444-016-9509-5

A family of non-oscillatory 6-point interpolatory subdivision schemes

Rosa Donat[1] · Sergio López-Ureña[1] · Maria Santágueda[2]

Received: 17 May 2016 / Accepted: 18 December 2016/
Published online: 14 February 2017

Abstract In this paper we propose and analyze a new family of nonlinear subdivision schemes which can be considered non-oscillatory versions of the 6-point Deslauries-Dubuc (DD) interpolatory scheme, just as the Power$_p$ schemes are considered nonlinear non-oscillatory versions of the 4-point DD interpolatory scheme. Their design principle may be related to that of the Power$_p$ schemes and it is based on a weighted analog of the Power$_p$ mean. We prove that the new schemes reproduce exactly polynomials of degree three and stay 'close' to the 6-point DD scheme in smooth regions. In addition, we prove that the first and second difference schemes are well defined for each member of the family, which allows us to give a simple proof of the uniform convergence of these schemes and also to study their stability as in [19, 22]. However our theoretical study of stability is not conclusive and we perform a series of numerical experiments that seem to point out that only a few members of the new family of schemes are stable. On the other hand, extensive numerical testing reveals that, for smooth data, the approximation order and the regularity of the limit function may be similar to that of the 6-point DD scheme and larger than what is obtained with the Power$_p$ schemes.

Communicated by: Yuesheng Xu

✉ Rosa Donat
donat@uv.es

Sergio López-Ureña
sergio.lopez-urena@uv.es

Maria Santágueda
santague@edu.uji.es

[1] Departament de Matemàtiques, Universitat de València, Doctor Moliner Street 50, 46100 Burjassot, Valencia, Spain

[2] Departament d'Educació, Universitat Jaume I de Castelló de la Plana, Castelló de la Plana, Spain

Keywords Nonlinear subdivision schemes · Convergence · Stability · Approximation order · Non-oscillatory

1 Introduction

Subdivision schemes are recursive processes used for the fast generation of curves and surfaces in computer-aided geometric design, as well as an essential ingredient in many multiscale algorithms used in data compression. In some applications, the given data need to be retained at each step of the refinement process, which requires the use of interpolatory subdivision schemes. The so-called Deslauries-Dubuc (DD henceforth) subdivision schemes [6] are a well known family of interpolatory subdivision schemes which can be interpreted as a recursive application of a piecewise polynomial interpolatory tool [10, 11]. A general setting by which a piecewise polynomial interpolation technique can be used to provide the set of local rules that defines a subdivision scheme has been described in [10]: Assuming that $\chi^l \subset \chi^{l+1}$ are two nested grids on $\mathbb{R}^m$, f^l is a set of known data associated to the grid χ^l and $\mathcal{I}[x, \cdot]$ is a piecewise polynomial reconstruction technique, new data associated to the grid χ^{l+1} can be generated as follows

$$f_i^{l+1} = \mathcal{I}[x_i^{l+1}, f^l], \qquad \text{for } x_i^{l+1} \in \chi^{l+1}. \tag{1}$$

This process allows to define a recursive subdivision scheme where sequences of values on denser and denser meshes are obtained according to the set of local rules derived from $\mathcal{I}[x, \cdot]$. Clearly (1) leads to an interpolatory subdivision scheme if $\mathcal{I}$ is an interpolatory reconstruction, i.e. $\mathcal{I}[x_i^l, f^l] = f_i^l$, $\forall x_i^l \in \chi^l$. For $m = 1$ and a binary refinement strategy, i.e. $x_{2i}^{l+1} = x_i^l$ and $\chi^{l+1} \backslash \chi^l \equiv \{x_{2j+1}^{l+1}\}_{j\in\mathbb{Z}}$, we have

$$\begin{aligned} f_{2i}^{l+1} &= \mathcal{I}[x_{2i}^{l+1}, f^l] = \mathcal{I}[x_i^l, f^l] = f_i^l \\ f_{2i+1}^{l+1} &= \mathcal{I}[x_{2i+1}^{l+1}, f^l] \end{aligned} \tag{2}$$

so that values on a given mesh are 'copied' at the same location on higher resolution levels, while the interpolatory technique $\mathcal{I}[\cdot, \cdot]$ specifies the local rules used for the generation of new data values.

It is well known (see e.g. [9, 10]) that the DD subdivision schemes can be written in the form (2) with $\mathcal{I}(\cdot, \cdot)$ a Lagrange interpolatory reconstruction that considers an interpolatory *stencil* centered around the evaluation point. In general, the use of piecewise polynomial Lagrange interpolation based on a stencil that uses l points to the left and r points to the right of the evaluation point leads to a *linear* (i.e. data-independent) subdivision scheme that can be written as

$$(S_{l,r} f)_{2i} = f_i, \quad (S_{l,r} f)_{2i+1} = \psi_{l,r}(f_{i-l}, \ldots, f_{i+r-1}) = \sum_{j=-l}^{r-1} a_j^{l,r} f_{i+j}, \quad f \in l_\infty(\mathbb{Z}) \tag{3}$$

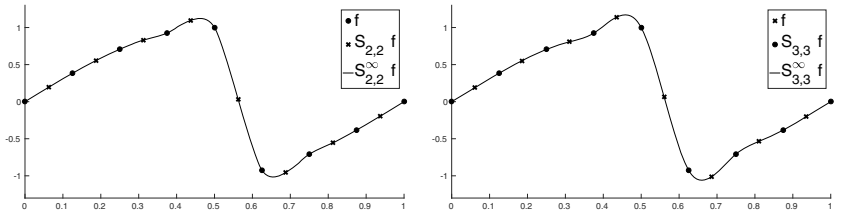


Fig. 1 Oscillatory behavior of DD schemes. For each scheme S, $S^\infty f$ is the corresponding limit function

with mask coefficients $a_j^{l,r}$ that can be computed from the interpolatory rule $\mathcal{I}[\cdot,\cdot]$ (see [9, 10]). It is well known that these schemes lead to a Gibbs-like oscillatory behavior when applied to discrete data with large gradients (see Fig. 1) and several nonlinear piecewise polynomial interpolatory techniques have been considered in the literature, within this same framework, in an attempt to construct interpolatory subdivision schemes that avoid undesired oscillations. Examples of such schemes are the ENO-WENO subdivision schemes [10, 11, 14, 16] or the PPH scheme [2, 4].

The oscillatory behavior displayed in Fig. 1 is typical of data-independent subdivision schemes based on Lagrange interpolation, which does not preserve the shape properties of data with large gradients when the degree of the polynomial pieces is larger than 1. ENO-WENO subdivision schemes manage to avoid the Gibbs-like oscillatory behavior by selecting an appropriate stencil for the interpolatory reconstruction [10, 11, 14, 16]. Other nonlinear interpolatory subdivision schemes, like the Power$_p$ schemes [17] or the shape-preserving schemes in [8], owe their non-oscillatory character to the judicious use of certain nonlinear averages.

The aim of this paper is to design, and analyze, non-oscillatory 6-point schemes that can be considered nonlinear analogs of the 6-point DD linear scheme, $S_{3,3}$, in the same sense as the Power$_p$ schemes are considered nonlinear, non-oscillatory versions of $S_{2,2}$, the 4 point DD subdivision scheme. For this, we shall use a weighted nonlinear average defined in [23] which generalizes the Power$_p$ mean defined in [15] and used in the design of the Power$_p$ schemes [17]. The new schemes proposed in this paper can be written in the following general form

$$(S_{\mathcal{N}} f)_n = (S_{\mathcal{L}} f)_n + \mathcal{F}(\delta f)_n, \quad \forall n \in \mathbb{Z}, \qquad \forall f \in l^\infty(\mathbb{Z}), \tag{4}$$

where $\mathcal{F} : l^\infty(\mathbb{Z}) \to l^\infty(\mathbb{Z})$ is a nonlinear operator, $\delta : l^\infty(\mathbb{Z}) \to l^\infty(\mathbb{Z})$ is linear and continuous and $S_{\mathcal{L}}$ is a linear and convergent subdivision scheme. The Power$_p$ schemes and other related subdivision schemes considered in [1, 3, 22] can also be written in the form (4), which allowed the authors to study their convergence and stability by using the following results [1].

Theorem 1 *Let $S_{\mathcal{N}}$ be a nonlinear subdivision scheme which can be written in the form* (4).

The scheme $S_{\mathcal{N}}$ is uniformly convergent provided that $\mathcal{F}$ and δ satisfy the following conditions:

C1. $\exists M > 0:$ $\quad ||\mathcal{F}(f)||_\infty \leq M||f||_\infty \quad \forall f \in l^\infty(\mathbb{Z})$

C2. $\exists L > 0,\ 0 < T < 1:$ $||\delta S^L_{\mathcal{N}}(f)||_\infty \leq T||\delta f||_\infty \quad \forall f \in l^\infty(\mathbb{Z})$

The scheme $S_{\mathcal{N}}$ is (Lipschitz) stable, i.e. $\exists C > 0$ such that

$$\|S^j_{\mathcal{N}} f - S^j_{\mathcal{N}} g\|_\infty \leq C\|f - g\|_\infty \qquad \forall f, g \in l^\infty(\mathbb{Z}), \quad \forall j \geq 0, \tag{5}$$

provided that $\mathcal{F}$ and δ satisfy the following conditions:

S1. $\exists M > 0:$ $\quad ||\mathcal{F}(f) - \mathcal{F}(g)||_\infty \leq M||f - g||_\infty, \qquad \forall f, g \in l^\infty(\mathbb{Z})$

S2. $\exists L > 0,\ 0 < T < 1:$ $||\delta(S^L_{\mathcal{N}}(f) - S^L_{\mathcal{N}}(g))||_\infty \leq T||\delta(f - g)||_\infty,\ \forall f, g \in l^\infty(\mathbb{Z})$

Remark 2 If a scheme of the form (4) is convergent, the smoothness of the limit functions $S^\infty_{\mathcal{N}} f$, $f \in l_\infty(\mathbb{Z})$, may be related to the smoothness of $S^\infty_{\mathcal{L}} f$. In particular it can be shown (see [12, 17]) that if $S_{\mathcal{L}}$ is C^{r-} convergent[1] then $S_{\mathcal{N}}$ is at least C^{s-} convergent with $s = \min\{-\frac{\log_2(T)}{L}, r\}$.

The new schemes proposed in this paper share other features with the Power$_p$ schemes. For both families of schemes the linear operator in Eq. 4 can be considered as $\delta = \nabla^2$, where

$$(\nabla f)_n = f_{n+1} - f_n, \qquad (\nabla^{m+1} f)_n = (\nabla^m f)_{n+1} - (\nabla^m f)_n, \quad m \geq 1, \quad n \in \mathbb{Z},$$

and the subdivision schemes are defined by piecewise smooth functions that are globally Lipschitz.

The paper is organized as follows: in Section 2 we provide an explanation of the non-oscillatory character of the Power$_p$ schemes which can be used as a design tool to obtain new families of non-oscillatory 6-point interpolatory subdivision schemes. These shall require a nonlinear analog of the Power$_p$ mean, the Weighted Power$_p$, proposed in [23]. The new families of 6-point schemes are defined and analyzed in Section 3. In this section we examine the polynomial reproduction properties and the existence of *difference schemes*, as well as the convergence and approximation properties of the new families of schemes.

Section 4 is devoted to the issue of the stability of the new schemes. In Section 4.1 we study the Weighted Power$_p$ mean, and its *Generalized Gradient*, an essential ingredient in the application of the theory developed in [19] for the study of the stability of a nonlinear scheme. The limitations of this theory in the study of the stability of the proposed schemes are analyzed in Section 4.2. In Section 5 we study the stability issue from a computational point of view, and also present several numerical examples that illustrate our theoretical results. We close in Section 6 with some conclusions.

[1] For $0 < r \leq 1$, C^{r-} is the space of bounded continuous functions satisfying $|F(x) - F(y)| \leq C|x - y|^{r_1}$, $\forall r_1 < r$, $\forall x, y \in \mathbb{R}$, $|x - y| < 1$, with $C > 0$ independent of x, y. For $r > 1$, $r = p + \beta$, $p \in \mathbb{N}$, $0 < \beta \leq 1$, it is required that $F^{(p)} \in C^{\beta-}$.

In addition, the relation between the Generalized Gradients of the piecewise smooth functions that define a nonlinear scheme, and the contractivity of such scheme is carefully explained in an Appendix to this paper.

2 Nonlinear averages and Non-oscillatory schemes

The Power$_p$ interpolatory subdivision schemes [17, 19] are binary interpolatory subdivision schemes for which the generation of new data values (at odd points) is given by the following rule

$$(S_{\mathrm{H}_p} f)_{2n+1} = \psi_{\mathrm{H}_p}(f_{n-1}, f_n, f_{n+1}, f_{n+2}) = \frac{1}{2}(f_n + f_{n+1}) - \frac{1}{8}\mathrm{H}_p(\nabla^2 f_{n-1}, \nabla^2 f_n) \tag{6}$$

where

$$\mathrm{H}_p(x, y) = \frac{\mathrm{sgn}(x) + \mathrm{sgn}(y)}{2} \frac{|x + y|}{2} \left(1 - \left|\frac{x - y}{x + y}\right|^p\right) \tag{7}$$

is the so-called Power$_p$ mean [15], a nonlinear function that satisfies (see [15, 17] for details)

$$\text{(a) } \mathrm{H}_p(x, x) = x, \quad \text{(b) } \min\{|x|, |y|\} \le |\mathrm{H}_p(x, y)| \le \min\{\max(|x|, |y|), p\min(|x|, |y|)\}. \tag{8}$$

It is straightforward to see that $\psi_{2,2}$ in Eq. 3 can be written as

$$\psi_{2,2}(f_{n-1}, f_n, f_{n+1}, f_{n+2}) = \frac{1}{2}(f_n + f_{n+1}) - \frac{1}{8}\mathrm{ave}_{\frac{1}{2},\frac{1}{2}}(\nabla^2 f_{n-1}, \nabla^2 f_n), \mathrm{ave}_{\frac{1}{2},\frac{1}{2}}(x, y) = \frac{1}{2}x + \frac{1}{2}y. \tag{9}$$

The obvious similarity between Eq. 9 and Eq. 6 makes the Power$_p$ schemes *nonlinear versions* of the 4-point DD scheme. In addition, if $f = (f_i)_{i\in\mathbb{Z}}$, $f_i = F(x_i)$ with $F(x)$ a smooth convex function, and $\mathcal{X} = \{x_i\}$ an h-uniform grid, it can be proven that

$$||S_{2,2} f - S_{\mathrm{H}_p} f||_\infty = \mathcal{O}(h^{p+2}). \tag{10}$$

This property is obtained from the following relation (see e.g. [17]), which holds for $x \cdot y > 0$, $p \ge 1$

$$|\mathrm{ave}_{\frac{1}{2},\frac{1}{2}}(x, y) - \mathrm{H}_p(x, y)| = \frac{1}{2}\frac{|x - y|^p}{|x + y|^{p-1}}. \tag{11}$$

On the other hand, as shown in Fig. 2, the behavior of S_{H_p} when refining discrete data with large gradients is quite different from that of $S_{2,2}$. In what follows, we give an explanation of the lack of oscillations observed in Fig. 2 which may be used to design 6-point nonlinear analogs of $S_{3,3}$, the 6 point DD scheme. The starting point

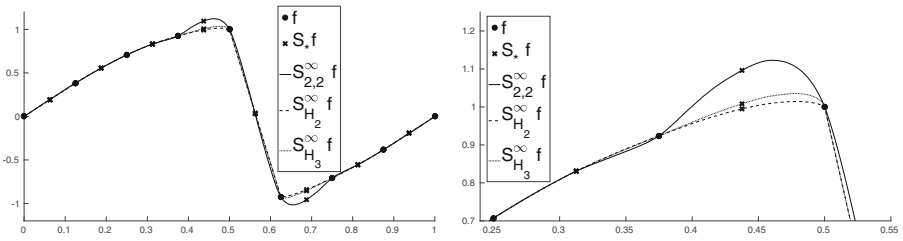


Fig. 2 Non-oscillatory behavior of Power$_p$ schemes, S_{H_p}, for $p = 2, 3$, compared to the 4-point DD scheme, $S_{2,2}$

in our construction is the following relation,

$$S_{l,r} = \frac{r - 1/2}{l + r - 1} S_{l,r-1} + \frac{l - 1/2}{l + r - 1} S_{l-1,r}, \qquad l, r \geq 1 \tag{12}$$

which follows from Neville's Algorithm for Lagrange interpolation (see e.g. [13]). Moreover, it is not difficult to see that we can write, for $l + r > 1$

$$S_{l,r} = S_{1,1} + \mathcal{L}_{l,r} \circ \nabla^2, \tag{13}$$

where $\mathcal{L}_{l,r}$ is a linear operator such that $(\mathcal{L}_{l,r} f)_{2n} = 0$, due to the interpolatory property. For the two schemes below

$$(S_{2,1} f)_{2n+1} = (S_{1,1} f)_{2n+1} - \frac{1}{8} \nabla^2 f_{n-1}, \quad (S_{1,2} f)_{2n+1} = (S_{1,1} f)_{2n+1} - \frac{1}{8} \nabla^2 f_n, \tag{14}$$

hence

$$(\mathcal{L}_{2,1} f)_{2n+1} = -\frac{1}{8} f_{n-1} \qquad (\mathcal{L}_{1,2} f)_{2n+1} = -\frac{1}{8} f_n.$$

From Eq. 12, for $l = 2, r = 2$, we get

$$S_{2,2} = \frac{1}{2} S_{2,1} + \frac{1}{2} S_{1,2} = S_{1,1} + \mathrm{ave}_{\frac{1}{2},\frac{1}{2}}(\mathcal{L}_{1,2} \circ \nabla^2, \mathcal{L}_{2,1} \circ \nabla^2) \tag{15}$$

while for S_{H_p} in Eq. 6 we can write

$$S_{\mathrm{H}_p} = S_{1,1} + \mathrm{H}_p(\mathcal{L}_{1,2} \circ \nabla^2, \mathcal{L}_{2,1} \circ \nabla^2). \tag{16}$$

Taking into account (15) and Eq. 16, the behavior of the $S_{2,2}$ and S_{H_p} schemes may be explained in terms of the interpolatory stencils associated to the schemes $S_{2,1}$ and $S_{1,2}$, shown in Fig. 4.

It is a well known fact that any Lagrange-type interpolatory technique suffers an $\mathcal{O}(1)$ accuracy loss as soon as the interpolatory stencil *crosses* a discontinuity. The data in Figs. 1-2 correspond to $f_i = F(x_i), i \in \mathbb{Z}$, $F(x)$ smooth except for an isolated discontinuity $\theta \in (x_m, x_{m+1})$. For these data

$$\nabla^2 f_j = \mathcal{O}(h^2), \quad j \neq m - 1, m, \qquad \nabla^2 f_{m-1} = \mathcal{O}(1) = \nabla^2 f_m.$$

Since

$$\mathrm{ave}_{\frac{1}{2},\frac{1}{2}}(\mathcal{O}(h^r), \mathcal{O}(h^s)) = \mathcal{O}(h^{\min(r,s)}), \qquad r > 0, s > 0, \quad 0 < h < 1, \tag{17}$$

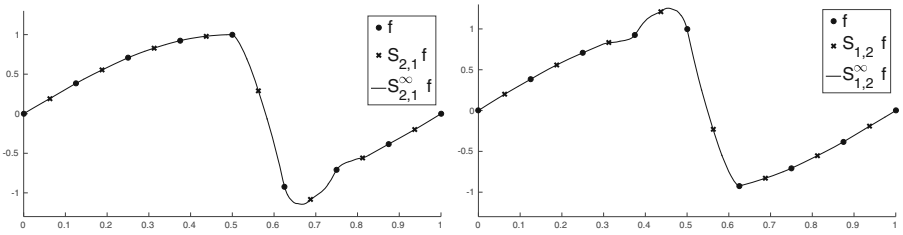


Fig. 3 Gibbs-like oscillatory behavior of $S_{1,2}$, $S_{2,1}$ schemes

we get (see Fig. 4)

$$(S_{2,2}f)_{2j+1} = (S_{1,1}f)_{2j+1} + \mathcal{O}(h^2) \qquad j \notin \{m-1, m, m+1\}, \tag{18}$$

$$(S_{2,2}f)_{2j+1} = (S_{1,1}f)_{2j+1} + \mathcal{O}(1) \qquad j \in \{m-1, m, m+1\}. \tag{19}$$

The values $(S_{2,2}f)_{2j+1}$ are displayed as $\times$ in Figs. 1-2, and the $\mathcal{O}(1)$ perturbations in Eq. 19 are clearly visible in Fig. 2 at the intervals adjacent to the one containing the discontinuity. After repeated application of the subdivision scheme, they cause the oscillations observed in the limit function, $S^{\infty}_{2,2}f$.

On the other hand, since $\min\{|x|, |y|\} \leq |H_p(x, y)| \leq p \min\{|x|, |y|\}$ for $xy > 0$, we may write

$$(S_{\mathrm{H}_p}f)_{2j+1} \approx (S_{1,1}f)_{2j+1} + \begin{cases} (\mathcal{L}_{2,1} \circ \nabla^2 f)_{2n+1} & j = m-1 \\ (\mathcal{L}_{1,2} \circ \nabla^2 f)_{2n+1} & j = m+1 \end{cases} = \begin{cases} (S_{2,1}f)_{2j+1} & j = m-1 \\ (S_{1,2}f)_{2j+1} & j = m+1. \end{cases}$$

Thus, the behavior of the S_{H_p} schemes at the intervals neighboring the singularity is closer to the behavior of the $S_{l,r}$ scheme, $(l, r) = \{(1, 2), (2, 1)\}$, which uses a stencil that does not cross the singularity, see Figs. 2, 3 and 4, which is, ultimately, the reason for the lack of oscillatory behavior.

We would like to proceed in a similar manner in order to limit the influence of schemes with singularity-crossing stencils for the 6-point DD linear scheme. For $S_{3,3}$ we may write

$$S_{3,3} = \frac{1}{2}S_{2,3} + \frac{1}{2}S_{3,2} = \frac{1}{2}(\frac{3}{8}S_{3,1} + \frac{5}{8}S_{2,2}) + \frac{1}{2}(\frac{3}{8}S_{1,3} + \frac{5}{8}S_{2,2}). \tag{20}$$

The stencils associated to the schemes $S_{2,3}$ and $S_{3,2}$ would not allow to avoid oscillations at all intervals neighboring a singularity in the data. On the other hand,

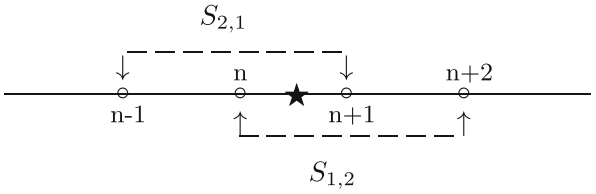


Fig. 4 ★ evaluation point, ∘ points in $\mathcal{S}_{2,2}$-stencil. Discontinuous lines: stencils for $\mathcal{S}_{1,2}$, $\mathcal{S}_{2,1}$

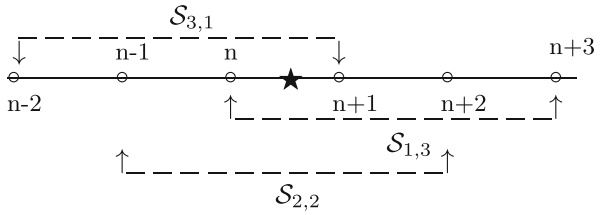


Fig. 5 ★: evaluation point, ∘ points in $\mathcal{S}_{3,3}$-stencil. Discontinuous lines: stencils for $\mathcal{S}_{2,2}$, $\mathcal{S}_{1,3}$ y $\mathcal{S}_{3,1}$

given the distribution of the stencils for $S_{3,1}$, $S_{1,3}$, $S_{2,2}$ shown in Fig. 5, we conclude that it might be possible to design non-oscillatory versions of the $S_{3,3}$ scheme by considering nonlinear analogs of the linear averages involved in Eq. 20 that allow us to remain *closer* to the $S_{l,r}$ scheme whose interpolatory stencil does not cross the singularity. For the general weighted average expression

$$\text{ave}_{a,b}(x, y) := ax + by, \qquad 0 \le a, b \le 1, \quad a + b = 1, \tag{21}$$

we may consider the *Weighted Power$_p$ mean* proposed in [23].

Definition 3 Weighted-Power$_p$ mean [23]. Let be $a > 0, b > 0$ satisfying $a+b = 1$, and $p \ge 1$. Then, $\forall x, y \in \mathbb{R}$.

$$\mathrm{W}_{p,a,b}(x, y) := sgn(x, y)\,|ax + by|\left(1 - \frac{|x - y|^p}{(M + \frac{m}{\alpha})(M + \alpha m)^{p-1}}\right), \tag{22}$$

where $M = \max\{|x|, |y|\}$, $m = \min\{|x|, |y|\}$, $\alpha = \max\{a, b\}/\min\{a, b\}$, $sgn(x, y) = \frac{1}{2}(\text{sign}(x) + \text{sign}(y))$.

It is proven in [23] that $\mathrm{W}_{p,a,b}(x, y)$ generalizes the H_p mean. Indeed, it can be easily checked that

$$\mathrm{W}_{p,a,b}(x, x) = x, \quad \mathrm{W}_{p,a,b}(x, y) = \mathrm{W}_{p,b,a}(y, x), \quad \mathrm{W}_{p,\frac{1}{2},\frac{1}{2}}(x, y) = \mathrm{H}_p(x, y). \tag{23}$$

We recall next some of the properties of $\mathrm{W}_{p,a,b}(x, y)$ in Eq. 22. The reader is referred to [23] for the proofs.

Proposition 4 *The function $W_{p,a,b}(x, y)$ in* (22) *satisfies the following properties.*

$$\text{(a) } |W_{p,a,b}(x, y)| \le |ax + by| \quad \text{(b) } \frac{1}{\alpha}\min\{|x|, |y|\} \le |W_{p,a,b}(x, y)| \le p\alpha \min\{|x|, |y|\}. \tag{24}$$

3 6-point Nonlinear, Non-oscillatory schemes

Taking into account Eqs. 20 and 13, we can write

$$S_{3,3} = S_{1,1} + \text{ave}_{\frac{3}{8},\frac{5}{8}}(\text{ave}_{\frac{1}{2},\frac{1}{2}}(\mathcal{L}_{1,3}\circ\nabla^2, \mathcal{L}_{3,1}\circ\nabla^2), \mathcal{L}_{2,2}\circ\nabla^2), \tag{25}$$

$$S_{3,3} = S_{1,1} + \text{ave}_{\frac{1}{2},\frac{1}{2}}(\text{ave}_{\frac{3}{8},\frac{5}{8}}(\mathcal{L}_{1,3}\circ\nabla^2, \mathcal{L}_{2,2}\circ\nabla^2), \text{ave}_{\frac{3}{8},\frac{5}{8}}(\mathcal{L}_{3,1}\circ\nabla^2, \mathcal{L}_{2,2}\circ\nabla^2). \tag{26}$$

where it can easily be shown that

$$(\mathcal{L}_{3,1}f)_{2n+1} = -\frac{1}{16}(-f_{n-2} + 3f_{n-1}), \quad (\mathcal{L}_{1,3}f)_{2n+1} = -\frac{1}{16}(3f_n - f_{n+1}),$$
$$(\mathcal{L}_{2,2}f)_{2n+1} = -\frac{1}{16}(f_{n-1} + f_n).$$

We may obtain two families of nonlinear 6-point schemes simply by replacing each linear average by the appropriate nonlinear mean (recall that $\text{W}_{p,\frac{1}{2},\frac{1}{2}} = \text{H}_p$).

$$\text{SWH}_{p,q} = S_{1,1} + \text{W}_{p,\frac{3}{8},\frac{5}{8}}(\text{H}_q(\mathcal{L}_{1,3}\circ\nabla^2, \mathcal{L}_{3,1}\circ\nabla^2), \mathcal{L}_{2,2}\circ\nabla^2), \tag{27}$$

$$\text{SHW}_{q,p} = S_{1,1} + \text{H}_q(\text{W}_{p,\frac{3}{8},\frac{5}{8}}(\mathcal{L}_{1,3}\circ\nabla^2, \mathcal{L}_{2,2}\circ\nabla^2), \text{W}_{p,\frac{3}{8},\frac{5}{8}}(\mathcal{L}_{3,1}\circ\nabla^2, \mathcal{L}_{2,2}\circ\nabla^2). \tag{28}$$

Because of Eq. 24-(b), these schemes remain closer to the subdivision scheme with the least oscillatory behavior, hence they are expected to display a non-oscillatory behavior similar to that of the Power$_p$ schemes, see Fig. 6.

In addition, since these subdivision schemes can be written as a nonlinear perturbation of the monotone $S_{1,1}$ linear scheme, many of their properties can be analyzed with the same tools used in [19] for the Power$_p$ schemes. We examine next the polynomial reproduction properties of these families of nonlinear subdivision schemes and the existence of difference schemes.

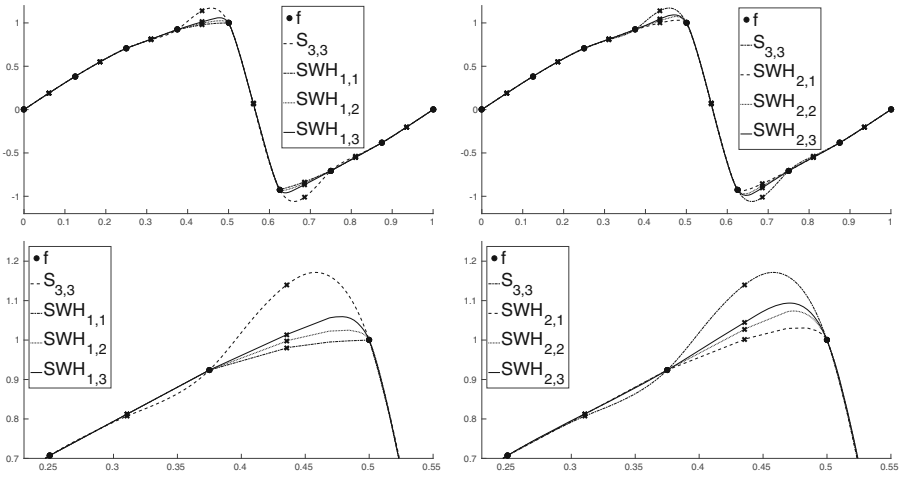


Fig. 6 Non-oscillatory behavior of SWH$_{p,q}$. For each scheme S: *Crosses* denote Sf. Lines denote $S^\infty f$

3.1 Polynomial reproduction properties and difference schemes

Throughout this section, we denote by Π_k the set of polynomials of degree $\leq k$, and by $\mathbf{1}$ the constant sequence given by $\mathbf{1}_i = 1,\ i \in \mathbb{Z}$.

Proposition 5 *The schemes $SWH_{p,q}$, $SHW_{q,p}$ reproduce exactly Π_3.*

Proof Since $S_{3,1}$, $S_{2,2}$ and $S_{1,3}$ reproduce Π_3 exactly, we have that for $P \in \Pi_3$ and $f = P\big|_{\mathbb{Z}}$

$$(S_{3,1}f)_{2n+1} = (S_{2,2}f)_{2n+1} = (S_{1,3}f)_{2n+1} = P(n+1/2), \quad \forall n \in \mathbb{Z}$$

and, from Eq. 13, $(\mathcal{L}_{3,1}(\nabla^2 f))_{2n+1} = (\mathcal{L}_{2,2}(\nabla^2 f))_{2n+1} = (\mathcal{L}_{1,3}(\nabla^2 f))_{2n+1}$. Since $\mathrm{W}_{p,a,b}(x,x) = x$, $\forall p, q \geq 1$

$$\begin{aligned}(\mathrm{SWH}_{p,q}f)_{2n+1} &= (\mathrm{SHW}_{q,p}f)_{2n+1} = (S_{1,1}f)_{2n+1} - (\mathcal{L}_{2,2}(\nabla^2 f))_{2n+1} \\ &= (S_{2,2}f)_{2n+1} = P(n+1/2).\end{aligned}$$ □

In the linear case, the relation between exact polynomial reproduction and the existence of the associated *difference* schemes $S^{[l]}$ with $S^{[0]} = S$, and $\nabla^l S = S^{[l]}\nabla^l$ is well known [20]. As observed in [19], offset invariance [18] is the right concept to characterize the existence of difference schemes in the nonlinear case.

Definition 6 [19] A binary subdivision operator S is offset invariant (OSI) for Π_k if for each $f \in l_\infty(\mathbb{Z})$ and any polynomial $P(x) \in \Pi_m$, $m \leq k$ there exists a polynomial, Q, of degree $< m$ such that

$$S(f + P\big|_{\mathbb{Z}}) = Sf + (P+Q)\big|_{2^{-1}\mathbb{Z}}.$$

Schemes of the form (4) with $\delta = \nabla^k$ are offset invariant for Π_{k-1}. To check this, let $P(x) \in \Pi_{k-1}$. Since $\nabla^k(P\big|_{\mathbb{Z}}) = 0$, we have $\forall f \in l_\infty(\mathbb{Z})$

$$\begin{aligned}S_{\mathcal{N}}(f + P\big|_{\mathbb{Z}}) &= S_{\mathcal{L}}(f + P\big|_{\mathbb{Z}}) + \mathcal{F}(\nabla^k f) = S_{\mathcal{L}}(f) + P\big|_{2^{-1}\mathbb{Z}} + \mathcal{F}(\nabla^k f) \\ &= S_{\mathcal{N}}(f) + P\big|_{2^{-1}\mathbb{Z}}.\end{aligned}$$

It is proven in [19] that offset invariance for Π_k guarantees the existence of the *difference* schemes $S^{[l]}$ for $l \leq k+1$. Thus the new families of nonlinear subdivision schemes are offset invariant for Π_1, which guarantees the existence of the first and second difference schemes. For the families in Eqs. 27-28, these schemes can be easily computed by elementary means. Introducing the restriction operator ($m < n$)

$$\chi_{m,n} : l_\infty(\mathbb{Z}) \to \mathbb{R}^{n-m+1}, \qquad \chi_{m,n}(f) = (f_m, f_{m+1}, \ldots, f_n),$$

and the functions $L_{l,r} : \mathbb{R}^4 \to \mathbb{R}$

$$L_{3,1}(x) = -x_1 + 3x_2, \quad L_{2,2}(x) = x_2 + x_3, \quad L_{1,3}(x) = 3x_3 - x_4, \quad x = (x_1, x_2, x_3, x_4) \in \mathbb{R}^4,$$

so that

$$(\mathcal{L}_{l,r}f)_{2n+1} = -\frac{1}{16} L_{l,r} \circ \chi_{n-2,n+1} f, \qquad (l,r) = (1,3), (3,1), (2,2),$$

we can write $\forall f \in l_\infty(\mathbb{Z})$,

$$\begin{cases} (\mathrm{SWH}_{p,q} f)_{2n} &= f_n \\ (\mathrm{SWH}_{p,q} f)_{2n+1} &= (S_{1,1} f)_{2n+1} + G_{p,q} \circ \chi_{n-2,n+1} \nabla^2 f \end{cases} \tag{29}$$

with

$$G_{p,q}(x) = -\tfrac{1}{16} \mathrm{W}_{p,\frac{3}{8},\frac{5}{8}} \left(\mathrm{H}_q \left(L_{3,1}(x), L_{1,3}(x) \right), L_{2,2}(x) \right). \tag{30}$$

From Eq. 29, it can be easily deduced that

$$\begin{cases} (\mathrm{SWH}^{[2]}_{p,q} w)_{2n} &= 2 G_{p,q} \circ \chi_{n-2,n+1} w, \\ (\mathrm{SWH}^{[2]}_{p,q} w)_{2n+1} &= \dfrac{w_n}{2} + G_{p,q} \circ \chi_{n-2,n+1} w + G_{p,q} \circ \chi_{n-1,n+2} w. \end{cases} \tag{31}$$

We obtain a similar result for $\mathrm{SHW}_{q,p}$, substituting $G_{p,q}$ by $R_{q,p}$

$$R_{q,p}(x) = -\tfrac{1}{16} \mathrm{H}_q \left(\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}} \left(L_{3,1}(x), L_{2,2}(x) \right), \mathrm{W}_{p,\frac{3}{8},\frac{5}{8}} \left(L_{1,3}(x), L_{2,2}(x) \right) \right). \tag{32}$$

We remark that the new schemes reproduce exactly Π_3, however they are only offset invariant for Π_1 and, hence, difference schemes $S^{[k]}$ do not exist for $k > 2$.

3.2 Convergence

As observed in [19, 22], the existence of difference schemes may be very helpful in proving the convergence of a nonlinear scheme. Since $\delta = \nabla^2$,

$$\delta S^L = \nabla^2 S S^{L-1} = S^{[2]} \nabla^2 S^{L-1} = (S^{[2]})^L \nabla^2, \tag{33}$$

so that **C2** in Theorem 1 is equivalent to the following condition

$$\exists L > 0,\ 0 < T < 1 : \qquad ||(S^{[2]})^L(f)||_\infty \le T ||f||_\infty \qquad \forall f \in l^\infty(\mathbb{Z}). \tag{34}$$

Theorem 7 *The schemes $SWH_{p,q}$ and $SHW_{q,p}$ are uniformly convergent, for all $p, q \ge 1$.*

Proof We shall check the conditions in Theorem 1. To check **C1**, we need to find a uniform bound for the non-linear functions $G_{p,q}$ and $R_{q,p}$ in Eqs. 30, Eq. 32. Using Eq. 24-(a) we get that $\forall x \in \mathbb{R}^4$

$$|G_{p,q}(x)| \le \frac{1}{16} |\frac{3}{8} \mathrm{H}_q(L_{3,1}(x), L_{1,3}(x)) + \frac{5}{8} L_{2,2}(x)| \le \frac{11}{64} \|x\|_\infty,$$

$$|R_{q,p}(x)| \le \frac{1}{16} |\frac{1}{2} \left(\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}} \left(L_{3,1}(x), L_{2,2}(x) \right) + \frac{1}{2} \mathrm{W}_{p,\frac{3}{8},\frac{5}{8}} \left(L_{1,3}(x), L_{2,2}(x) \right) \right)| \le \frac{11}{64} \|x\|_\infty.$$

To check **C2** we consider its equivalent formulation (34). For even indexes,

$$|(\mathrm{SWH}^{[2]}_{p,q} f)_{2n}| \le \tfrac{11}{32} ||f||_\infty, \qquad |(\mathrm{SHW}^{[2]}_{q,p} f)_{2n}| \le \tfrac{11}{32} ||f||_\infty,$$

using the previously computed bounds. For odd components we get

$$\max\{|(\mathrm{SWH}^{[2]}_{p,q} f)_{2n+1}|, (\mathrm{SHW}^{[2]}_{q,p} f)_{2n+1}|\} \leq \frac{1}{2}||f||_\infty + \frac{11}{64}\Big(||f||_\infty + ||f||_\infty\Big) \leq \frac{27}{32}||f||_\infty.$$

Hence,

$$\max\{\|\mathrm{SWH}^{[2]}_{p,q}\|_\infty, \|\mathrm{SHW}^{[2]}_{q,p}\|_\infty\} \leq \frac{27}{32} < 1. \tag{35}$$

□

Remark We notice that the bound in Eq. 35 implies that for $S = \mathrm{SWH}_{p,q}, \mathrm{SHW}_{q,p}$ and $f \in l_\infty(\mathbb{Z})$ the limit function $S^\infty f$ is at least $C^{\beta-}$ with $\beta = \min\{-\log_2\left(\frac{27}{32}\right), 1\} \simeq 0.2540$. This result appears to be suboptimal, since all numerical evidence suggests that $S^\infty f \in C^{1-}$.

3.3 Order of accuracy

The order of approximation of a subdivision scheme measures the approximation properties of the recursive process when applied to discrete data coming from smooth functions.

Definition 8 A convergent subdivision scheme S has approximation order r if

$$||S^\infty f^0 - F||_\infty \leq Dh^r \tag{36}$$

when $f^0_i = F(ih), i \in \mathbb{Z}$ for any $F(x)$ sufficiently smooth.

For a given subdivision scheme, the *order of approximation after one iteration* is usually much easier to obtain.

Definition 9 Let S be a subdivision scheme that satisfies

$$\max_i |f^1_{2i+1} - F(ih + h/2)| \leq Ch^r, \quad C < \infty \tag{37}$$

with $f^1 = Sf^0$ and $f^0_i = F(ih), i \in \mathbb{Z}$ for any $F(x)$ sufficiently smooth. Then r is called the *order of approximation after one iteration* of S.

Obviously, the order of approximation after one iteration of interpolatory subdivision schemes based on Lagrange interpolation is at least as high as that of the interpolatory reconstruction used in its design. We notice that Eq. 10 implies that the order of approximation after one iteration of the Power_p schemes is at least 4, when refining smooth convex functions and $p \geq 2$, since

$$||S_{\mathrm{H}_p} f - F|_{2^{-1}h\mathbb{Z}}||_\infty \leq ||S_{\mathrm{H}_p} f - S_{2,2} f||_\infty + ||S_{2,2} f - F|_{2^{-1}h\mathbb{Z}}||_\infty = O(h^4). \tag{38}$$

For the schemes defined in this paper, we can also measure how close the new schemes are to the 6-point DD scheme for smooth convex functions.

Remark 10 We know that if $f_i = F(ih)$ and $F(x)$ is smooth and convex, $(\nabla^2 f)_i$ do not change sign. We can show, by straightforward Taylor expansions, that

$$(\mathcal{L}_{l,r}\nabla^2 f)_i = 2F''(x_i)h^2 + F'''(x_i)h^3 + O(h^4) \quad (l,r) \in \{(1,3),(3,1),(2,2)\} \tag{39}$$

hence we also expect that, for smooth convex functions and h small enough, $(\mathcal{L}_{l,r}\nabla^2 f)_i$ will not change sign either.

Proposition 11 *Let $F : \mathbb{R} \to \mathbb{R}$ a smooth function, and $f = \{F(ih)\}_{i\in\mathbb{Z}}$. If $(\mathcal{L}_{l,r}\nabla^2 f)_n$ have the same sign $\forall n \in \mathbb{Z}$, $(l,r) \in \{(1,3),(3,1),(2,2)\}$, and $|F''(x)| > \rho > 0$, $x \in \mathbb{R}$, then*

$$||S_{3,3}f - SWH_{p,q}f||_\infty = \mathcal{O}(h^r) = ||S_{3,3}f - SHW_{q,p}f||_\infty, \qquad r = \min\{2p+2, 3q+2\}.$$

Proof Notice that

$$(S_{3,3}f)_{2n+1} - (\mathrm{SWH}_{p,q}f)_{2n+1} = -\tfrac{1}{16}\left(\mathrm{ave}_{\frac{3}{8},\frac{5}{8}}\left(\mathrm{ave}_{\frac{1}{2},\frac{1}{2}}(x,z),y\right) - \mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}\left(\mathrm{H}_q(x,z),y\right)\right)$$

with

$$x := 3\nabla^2 f_{n-1} - \nabla^2 f_{n-2}, \quad y := \nabla^2 f_{n-1} + \nabla^2 f_n, \quad z := 3\nabla^2 f_n - \nabla^2 f_{n+1}.$$

Since F is a smooth function, the Taylor expansions in Eq. 39 show that x, y, z are $O(h^2)$ and non-zero, provided that h is sufficiently small. Since (24)-(b) ensures that $\mathrm{W}_{p,a,b}(O(h^r), O(h^s)) = O(h^{\min\{r,s\}})$, we have that $\mathrm{H}_q(x,z) = \mathcal{O}(h^2)$, $\mathrm{W}_{p,a,b}(x,y) = \mathcal{O}(h^2) = \mathrm{W}_{p,a,b}(y,z)$. We write

$$\mathrm{ave}_{\frac{3}{8},\frac{5}{8}}\left(\mathrm{ave}_{\frac{1}{2},\frac{1}{2}}(x,z),y\right) - \mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}\left(\mathrm{H}_q(x,z),y\right) = Z_1(x,y,z) + Z_2(x,y,z)$$

with

$$Z_1(x,y,z) := \mathrm{ave}_{\frac{3}{8},\frac{5}{8}}\left(\mathrm{ave}_{\frac{1}{2},\frac{1}{2}}(x,z),y\right) - \mathrm{ave}_{\frac{3}{8},\frac{5}{8}}\left(\mathrm{H}_q(x,z),y\right) = \tfrac{3}{8}(\mathrm{ave}_{\frac{1}{2},\frac{1}{2}}(x,z) - \mathrm{H}_q(x,z))$$
$$Z_2(x,y,z) := \mathrm{ave}_{\frac{3}{8},\frac{5}{8}}\left(\mathrm{H}_q(x,z),y\right) - \mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}\left(\mathrm{H}_q(x,z),y\right).$$

Notice that (using Taylor expansions, when necessary)

$$x - z = 3\nabla^2 f_{n-1} - \nabla^2 f_{n-2} - 3\nabla^2 f_n + \nabla^2 f_{n+1} = -\nabla^4 f_{n-2} + \nabla^4 f_{n-1} = \nabla^5 f_{n-2} = O(h^5)$$
$$x + z = 3\nabla^2 f_{n-1} - \nabla^2 f_{n-2} + 3\nabla^2 f_n - \nabla^2 f_{n+1} = O(h^2).$$

Hence, assuming without loss of generality that $x, z \geq 0$, Eq. 7 leads to

$$Z_1(x,y,z) = \frac{3}{16}\frac{|x-z|^q}{|x+z|^{q-1}} = \mathcal{O}(h^{3q+2}).$$

For $Z_2(x,y,z)$, denoting $s = H_p(x,z)$ and using Eq. 22, we get that $(s, y > 0)$,

$$Z_2(x,y,z) = \mathrm{ave}_{\frac{3}{8},\frac{5}{8}}(s,y) - \mathrm{W}_{p,a,b}(s,y) = \left(\frac{3}{8}s + \frac{5}{8}y\right)\frac{|s-y|^p}{(M+\frac{3}{5}m)(M+\frac{5}{3}m)^{p-1}} \tag{40}$$

with $M = \max\{s, y\}$, $m = \min\{s, y\}$. Notice that $s = O(h^2)$, hence

$$\text{ave}_{\frac{3}{8},\frac{5}{8}}(s, y) = O(h^2), \quad M = \min\{s, y\} = O(h^2), \quad m = \max\{s, y\} = O(h^2).$$

Moreover,

$$s - y = \text{H}_q(x, z) - y = \tfrac{x-2y+z}{2} - \tfrac{1}{2}\tfrac{|x-z|^q}{(x+z)^{q-1}} = \mathcal{O}(h^4) + \mathcal{O}(h^{3q+2}),$$

because

$$x - 2y + z = -\nabla^2 f_{n-2} + \nabla^2 f_{n-1} + \nabla^2 f_n - \nabla^2 f_{n+1} = \nabla^3 f_{n-2} - \nabla^3 f_n = \nabla^4 f_{n-2} = O(h^4).$$

Since $4 < 3q + 2$ for $q \geq 1$, we may conclude that

$$Z_2(x, y, z) = O(h^2)O(h^{4p-2p}) = O(h^{2p+2}),$$

from which we deduce the desired result for the schemes $\text{SWH}_{p,q}$.

For the $\text{SHW}_{q,p}$ family, we proceed in a similar way. Assume $x, y, z > 0$ and write

$$(S_{3,3} f)_{2n+1} - (\text{SHW}_{q,p} f)_{2n+1} = -\tfrac{1}{16}(Y_1(x, y, z) + Y_2(x, y, z))$$

with

$$\begin{aligned} Y_1(x, y, z) &= \text{ave}_{\frac{1}{2},\frac{1}{2}}\left(\text{ave}_{\frac{3}{8},\frac{5}{8}}(x, y), \text{ave}_{\frac{3}{8},\frac{5}{8}}(z, y)\right) - \text{ave}_{\frac{1}{2},\frac{1}{2}}\left(\text{W}_{p,\frac{3}{8},\frac{5}{8}}(x, y), \text{W}_{p,\frac{3}{8},\frac{5}{8}}(z, y)\right), \\ &= \frac{1}{2}(\text{ave}_{\frac{3}{8},\frac{5}{8}}(x, y) - \text{W}_{p,\frac{3}{8},\frac{5}{8}}(x, y)) + \frac{1}{2}(\text{ave}_{\frac{3}{8},\frac{5}{8}}(z, y) - \text{W}_{p,\frac{3}{8},\frac{5}{8}}(z, y))) \\ Y_2(x, y, z) &= \text{ave}_{\frac{1}{2},\frac{1}{2}}\left(\text{W}_{p,\frac{3}{8},\frac{5}{8}}(x, y), \text{W}_{p,\frac{3}{8},\frac{5}{8}}(z, y)\right) - \text{H}_q\left(\text{W}_{p,\frac{3}{8},\frac{5}{8}}(x, y), \text{W}_{p,\frac{3}{8},\frac{5}{8}}(z, y)\right)). \end{aligned}$$

As before, it is easy to deduce that

$$\begin{aligned} x - y &= -\nabla^4 f_{n-2} = O(h^4), \quad x + y = O(h^2), \\ z - y &= -\nabla^4 f_{n-1} = O(h^4), \quad z + y = O(h^2). \end{aligned}$$

Hence, for $x, y, z \geq 0$, using Eq. 22 and proceeding as in Eq. 40 we get,

$$\text{ave}_{\frac{3}{8},\frac{5}{8}}(x, y) - \text{W}_{p,\frac{3}{8},\frac{5}{8}}(x, y) = O(h^{2p+2}) = \text{ave}_{\frac{3}{8},\frac{5}{8}}(z, y) - \text{W}_{p,\frac{3}{8},\frac{5}{8}}(z, y).$$

Thus $Y_1(x, y, z) = O(h^{2p+2})$ (notice that the two terms of the $\frac{1}{2}, \frac{1}{2}$ average in Y_1 have the same sign).

For $Y_2(x, y, z)$, we use that $\text{W}_{p,a,b} : \mathbb{R}^2 \to \mathbb{R}$ is a Lipschitz function (see next section), hence

$$|\text{W}_{p,\frac{3}{8},\frac{5}{8}}(x, y) - \text{W}_{p,\frac{3}{8},\frac{5}{8}}(z, y)| \leq L|x - z| = O(h^5) \tag{41}$$

being L the Lipschitz constant of $\text{W}_{p,\frac{3}{8},\frac{5}{8}}$. Then, defining $u := \text{W}_{p,\frac{3}{8},\frac{5}{8}}(x, y)$, $v := \text{W}_{p,\frac{3}{8},\frac{5}{8}}(z, y)$ and noticing $u, v > 0$ if $x, y, z > 0$ and $u = O(h^2) = v$, we have, using Eqs. 7 and 41,

$$Y_2(x, y, z) = \text{ave}_{\frac{1}{2},\frac{1}{2}}(u, v) - \text{H}_q(u, v) = \frac{1}{2}\frac{|u - v|^q}{|u + v|^{q-1}} = \frac{O(h^{5q})}{O(h^{2q-2})} = O(h^{3q+2}),$$

from which we deduce the desired result also for the schemes $\text{SHW}_{q,p}$. □

Corollary 12 *Under the same conditions as in the previous proposition*

$$||SWH_{p,q}\, f - F|_{2^{-1}h\mathbb{Z}}||_\infty = O(h^r) = ||SHW_{q,p}\, f - F|_{2^{-1}h\mathbb{Z}}||_\infty, \qquad r = \min\{2p+2, 3q+2, 6\}.$$

As shown in [5], the order of approximation of a *stable* subdivision scheme can be deduced from its order of approximation after one iteration. The following result from [5] holds for linear as well as nonlinear subdivision schemes and it serves also as a motivation to study the stability of the nonlinear schemes under consideration.

Theorem 13 *Let S be a convergent subdivision scheme whose approximation order after one iteration is $r \geq 1$. Then if S is stable, it has approximation order r.*

In [19] it is shown that the Power$_p$ schemes are stable subdivision schemes for $p < 3$ and unstable for $p \geq 4$, hence (38) and Theorem 13 ensure that the order of approximation of the Power$_p$ schemes is 4, when refining smooth convex functions, for $p < 3$.

In the following section we examine the question of stability for the families of schemes (27) and Eq. 28, in order to check whether or not similar conclusions can be extracted for the new families of schemes presented in this paper.

4 Stability of the 6-point nonlinear schemes

Theorem 1 establishes that stability follows from two facts: Lipschitz-continuity of the operator $\mathcal{F}$ and contractivity of $\delta S^L_{\mathcal{N}}$, for some $L > 0$. When $\delta = \nabla^2$ and $S_{\mathcal{N}}$ is offset invariant for Π_1 (the case of the new families of 6-point schemes), condition **S2** can be equivalently expressed as follows

$$\exists\, L > 0, 0 < \mu < 1 : \ ||(S^{[2]}_{\mathcal{N}})^L f - (S^{[2]}_{\mathcal{N}})^L g||_\infty = \mu ||f - g||_\infty, \ \forall f, g \in l_\infty(\mathbb{Z}). \tag{42}$$

We shall see next that the second difference schemes in Eq. 31 are defined by nonlinear functions that admit uniformly bounded *Generalized Gradients*. In [22], this fact was used to show the stability of a nonlinear, monotonicity preserving, scheme by expressing the contractivity condition (42) in terms of the *Generalized Jacobian* of the scheme and using Corollary 24 in the Appendix (or see [19]). However, we shall see that this technique does not seem to be as useful for the schemes considered in this paper.

The first step is to show that the Weighted-Power$_p$ mean (22) belongs to a special class of continuous, piecewise smooth functions: the class of $C^1_{pw}(\mathbb{R}^2)$ functions. Functions in this class are continuous, piecewise smooth and have uniformly bounded directional derivatives except (maybe) at $0 \in \mathbb{R}^m$ and *across* certain hyperplanes separating regions of C^1 smoothness. Directional derivatives *along* the separating hyperplanes do, also, exist. For this class of functions it is possible to define a *Generalized Gradient* using only the gradients on smooth regions. As the classical gradient

for smooth functions, the linear map associated to any Generalized Gradient recovers all directional derivatives that *'make sense'*, and satisfies a chain rule property for the composition with Lipschitz curves. We refer the reader to [19] or the Appendix to this paper for the definition, and the main properties, of this class of functions.

4.1 The Weighted-Harmonic mean: Generalized Gradients

Property (24)-(a) implies that $\mathrm{W}_{p,a,b}(x, y)$ is a continuous function in $\mathbb{R}^2$. It is obviously differentiable in the interior of the sectors in $\mathbb{R}^2$ separated by the three hyperplanes $\mathcal{H}_1 = \{x = 0\}$, $\mathcal{H}_2 = \{y = 0\}$, $\mathcal{H}_3 = \{x = y\}$. As observed in [19] (see also the Appendix) a *Generalized Gradient* for $\mathrm{W}_{p,a,b}(x, y)$ can be defined using only the gradients in smoothness regions (see Eq. A.3), provided that certain compatibility conditions are satisfied over the separating hyperplanes. Since $\mathrm{W}_{p,a,b}(x, y) = -\mathrm{W}_{p,a,b}(-x, -y)$, it is enough to consider the half plane $y \geq 0$. Then, the compatibility conditions (A.2) amount to showing that

$$\lim_{0\neq x>y\to 0} \nabla \mathrm{W}_{p,a,b}(x, y)\cdot(1, 0) = 0, \qquad \lim_{0\neq y>x\to 0} \nabla \mathrm{W}_{p,a,b}(x, y)\cdot(0, 1) = 0, \tag{43}$$

$$\lim_{y>x\neq 0,(x,y)\to(d,d)} \nabla \mathrm{W}_{p,a,b}(x, y)\cdot(1, 1) = (\mathrm{W}_{p,a,b}(x, x))' = 1, \quad \forall d > 0, \tag{44}$$

$$\lim_{0\neq y<x,(x,y)\to(d,d)} \nabla \mathrm{W}_{p,a,b}(x, y)\cdot(1, 1) = (\mathrm{W}_{p,a,b}(x, x))' = 1, \quad \forall d > 0. \tag{45}$$

To check these conditions, we first notice that $\mathrm{W}_{p,b,a}(y, x) = \mathrm{W}_{p,a,b}(x, y)$, hence it is enough to consider the gradients of the 1-homogeneous[2] function (see Fig. 7)

$$\phi_{p,a,b}(x, y) = (ax + by)\left(1 - \frac{(y - x)^p}{(y + \frac{1}{\alpha}x)(y + \alpha x)^{p-1}}\right), \qquad \alpha = \max(a, b)/\min(a, b).$$

A straightforward computation leads to

$$\begin{aligned}
\nabla\phi_{p,a,b}(x, y) &= (y, -x)\rho_{p,a,b}(x, y) + (a, b)\sigma_{p,\alpha}(x, y),\\
\rho_{p,a,b}(x, y) &= \frac{\alpha(\alpha + 1)(y - x)^{p-1}(ax + by)(x(\alpha + p - 1) + y(\alpha(p - 1) + 1))}{(x + \alpha y)^2(\alpha x + y)^p},\\
\sigma_{p,\alpha}(x, y) &= 1 - \frac{\alpha(y - x)^p}{(x + \alpha y)(\alpha x + y)^{p-1}}.
\end{aligned}$$

Since $\nabla\phi_{a,b}$ is 0-homogeneous, for $y > 0$, $\nabla\phi_{a,b}(x, y) = \nabla\phi_{a,b}(x/y, y/y) = \nabla\phi_{a,b}(t, 1)$, $t = x/y$, hence

$$\nabla\phi_{a,b}(t, 1) = \mu_{p,a,b}(t)(1, -t) + \eta_{p,\alpha}(t)(a, b). \tag{46}$$

with

[2] A function F is n-homogeneous if $F(\lambda x) = \lambda^n F(x)$, being x a vector and λ a scalar.

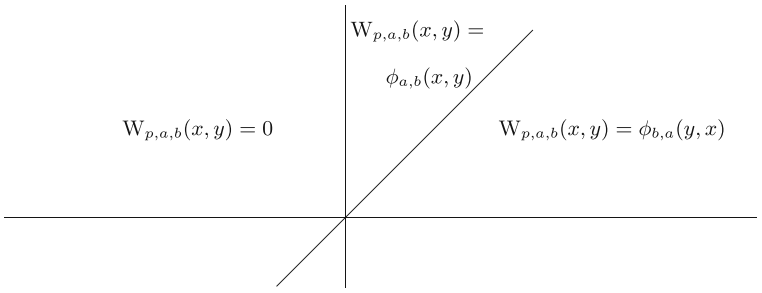


Fig. 7 $\mathrm{W}_{p,a,b}(x, y)$ in its smoothness sectors

$$\rho_{p,a,b}(x, y) = \mu_{p,a,b}(x/y), \quad \mu_{p,a,b}(t) := \frac{c_1(1-t)^{p-1}(at+b)(c_2t+c_3)}{(t+\alpha)^2(\alpha t+1)^p} \begin{cases} c_1 = \alpha(\alpha+1) \\ c_2 = \alpha+p-1 \\ c_3 = \alpha(p-1)+1 \end{cases},$$

$$\sigma_{p,\alpha}(x, y) = \eta_{p,\alpha}(x/y), \quad \eta_{p,\alpha}(t) := 1 - \frac{\alpha(1-t)^p}{(t+\alpha)(\alpha t+1)^{p-1}}.$$

Clearly, the functions $\mu_{p,a,b}(t)$, $t\mu_{p,a,b}(t)$, are continuous in $[0, 1]$. Moreover,

$$\lim_{t\to 0} t\mu_{p,a,b}(t) = 0, \quad \lim_{t\to 0} \mu_{p,a,b}(t) = \frac{c_1 b c_3}{\alpha^2}, \qquad \lim_{t\to 0} \eta_{p,\alpha}(t) = 0,$$

$$\lim_{t\to 1} \mu_{p,a,b}(t) = \lim_{t\to 1} t\mu_{p,a,b}(t) = \begin{cases} c_1(c_2+c_3)/\alpha^2 & p = 1 \\ 0 & p > 1 \end{cases}, \qquad \lim_{t\to 1} \eta_{p,\alpha}(t) = 1.$$

Taking into account Fig. 7, we observe that the compatibility conditions (43), Eq. 44, Eq. 45 follow from the fact that $\forall p \geq 1, \forall a, b \geq 0, a + b = 1$,

$$\lim_{0\neq y>x\to 0} \nabla\phi_{p,a,b}(x, y) \cdot (0, 1) = \lim_{t\to 0} -t\mu_{p,a,b}(t) + b\eta_{p,\alpha}(t) = 0,$$
$$\lim_{y>x, x, y\to d\neq 0} \nabla\phi_{p,a,b}(x, y) \cdot (1, 1) = \lim_{t\to 1} (1-t)\mu_{p,a,b}(t) + (a+b)\eta_{p,\alpha}(t) = 1.$$

In addition, $\nabla\phi_{a,b}(x, y)$ is uniformly bounded in $\{(x, y) : y > x > 0\}$ because $\nabla\phi_{a,b}(t, 1)$ is bounded for $0 \leq t \leq 1$. Hence, $\nabla \mathrm{W}_{p,a,b}(x, y)$ is also uniformly bounded for (x, y) in any region of smoothness and, thus, $\mathrm{W}_{p,a,b}(x, y)$ in Eq. 22 belongs to the space $C^1_{pw}(\mathbb{R}^2)$. From Corollary 19, $\mathrm{W}_{p,a,b}(x, y)$ is a Lipschitz function. However, uniform bounds for $||D\mathrm{W}_{p,a,b}(x, y)||$ depend on the parameters p, a, b in a more involved way than for the Power_p averages. We illustrate the required computations by examining the cases $p = 1, 2$.

Proposition 14 *$W_{1,a,b}(x, y)$ admits a generalized gradient, which satisfies $\forall(x, y) \in \mathbb{R}^2$*

$$||DW_{1,a,b}(x, y)||_1 \leq 1 + 2(c - d), \qquad c = \max\{a, b\}, d = \min\{c, d\} \qquad (47)$$

Proof For $p = 1$,

$$\mu_{1,a,b}(t) = \frac{\alpha(\alpha+1)(at+b)}{(t+\alpha)^2}, \qquad \eta_{1,\alpha}(t) = 1 - \frac{\alpha(1-t)}{t+\alpha} = (1+\alpha)\frac{t}{t+\alpha}.$$

Since $\alpha = c/d$, and $a + b = c + d = 1$, Eq. 46 becomes

$$\nabla\phi_{1,a,b}(t,1) = c\frac{at+b}{(dt+c)^2}(1,-t) + \frac{t}{dt+c}(a,b).$$

If $d = a \leq b = c$, we get

$$\nabla\phi_{1,a,b}(t,1) = \frac{c}{(dt+c)}(1,-t) + \frac{t}{dt+c}(d,c) = (1,0) \quad \Rightarrow \quad ||\nabla\phi_{1,a,b}(t,1)||_1 = 1.$$

If $d = b \leq a = c$, after some algebraic manipulations (notice that $c^2 - d^2 = c - d$) we get

$$\nabla\phi_{1,a,b}(x,y) = (1,0) + \tfrac{c-d}{(dt+c)^2}(dt^2 + 2ct - c, -t^2) \tag{48}$$

hence $||\nabla\phi_{1,a,b}(t,1)||_1 = 1 + \frac{c-d}{(dt+c)^2}(dt^2 + 2ct - c + 1)$. This is an increasing function in $[0, 1]$, hence

$$1 \leq ||\nabla\phi_{1,a,b}(0,1)||_1 \leq ||\nabla\phi_{1,a,b}(t,1)||_1 \leq ||\nabla\phi_{1,a,b}(1,1)||_1 = 1 + 2(c-d)$$

which proves the result. □

Proposition 15 *$W_{2,a,b}$ admits a generalized gradient that satisfies $\forall (x, y) \in \mathbb{R}^2$*

$$0 \leq D_1 W_{2,a,b}(x,y) \leq \frac{1}{a}, \qquad 0 \leq D_2 W_{2,a,b}(x,y) \leq \frac{1}{b}, \qquad ||DW_{2,a,b}(x,y)||_1 \leq \frac{1}{d}. \tag{49}$$

Proof We proceed as in Proposition 14. After straightforward manipulations we get

$$\nabla\phi_{2,a,b}(t,1) = \frac{1}{(bt+a)^2}(a, bt^2).$$

Notice that $D_1\phi_{2,a,b}(t,1) = a/(bt+a)^2$ is a decreasing function in $[0, 1]$ while $D_2\phi_{2,a,b}(t) = bt^2/(bt+a)^2$ is increasing. Moreover, both components are positive, hence $\|\nabla\phi_{2,a,b}(t)\|_1 = D_1\phi_{2,a,b}(t) + D_2\phi_{2,a,b}(t) = (a+bt^2)(bt+a)^2$, which is a decreasing function. Hence we readily conclude

$$a \leq D_1\phi_{2,a,b}(t,1) \leq \frac{1}{a}, \qquad 0 \leq D_2\phi_{2,a,b}(t,1) \leq b, \qquad 1 \leq ||\nabla\phi_{2,a,b}(t,1)||_1 \leq \frac{1}{a},$$

and we deduce (49) from the relations above and Fig. 7. □

Remark 16 We may also obtain specific bounds for the components of $D\mathrm{W}_{1,a,b}(x,y)$, although the bounds are not as simple as for $p = 2$. It is easy to

see that the function $1 + \frac{c-d}{(dt+c)^2}(dt^2 + 2ct - c)$ is increasing and $\frac{c-d}{(dt+c)^2}(-t^2)$ is decreasing in $[0, 1]$. Therefore $\forall t \in (0, 1)$

$$D_1\phi_{1,a,b}(0,1) \le D_1\phi_{1,a,b}(t,1) \le D_1\phi_{1,a,b}(1,1), \qquad D_2\phi_{1,a,b}(1,1) \le D_2\phi_{1,a,b}(t,1) \le D_2\phi_{1,a,b}(0,1)$$
$$\frac{d}{c} \le D_1\phi_{1,a,b}(x,y) \le 1 + (c-d), \qquad -(c-d) \le D_2\phi_{1,a,b}(x,y) \le 0, \qquad \text{if } a > b. \tag{50}$$

From these bounds and Fig. 7, we may deduce general, uniform, bounds for the components of $D\mathrm{W}_{1,a,b}(x, y)$.

4.2 The Generalized Jacobian of the second difference schemes

Let us consider the family of schemes $\mathrm{SWH}_{p,q}$ in Eq. 31. As specified in the Appendix, in order to define a Generalized Jacobian of $\mathrm{SWH}^{[2]}_{p,q}$ we need to justify the existence of uniformly bounded Generalized Gradients for the function $G_{p,q}$ in Eq. 30, which satisfy the chain rule (A.10).

Proposition 17 *Let $\gamma : [a, b] \to \mathbb{R}^m$ be a Lipschitz curve. Then $\tilde{\gamma} = G_{p,q} \circ \gamma : [a, b] \to \mathbb{R}$ is also Lipschitz and*

$$\tilde{\gamma}'(t) = DG_{p,q}(\gamma(t))\gamma'(t) \qquad \text{a.e. in } [a, b] \tag{51}$$

where, $\forall x \in \mathbb{R}^4$,

$$DG_{p,q}(x) := -\frac{1}{16} DW_{p,\frac{3}{8},\frac{5}{8}}(\xi) \begin{pmatrix} DH_q(\rho) & 0 \\ (0,0) & 1 \end{pmatrix} \begin{pmatrix} -1 & 3 & 0 & 0 \\ 0 & 0 & 3 & -1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \tag{52}$$

with $\rho = (L_{3,1}(x), L_{1,3}(x))$, $\quad \xi = (H(L_{3,1}(x), L_{1,3}(x)), L_{2,2}(x))$.

Proof Notice that $G_{p,q} = -\frac{1}{16}\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}} \circ \psi \circ M$ where $\psi : \mathbb{R}^3 \to \mathbb{R}^2$ and $M : \mathbb{R}^4 \to \mathbb{R}^3$ are as follows

$$\psi(x) = (\mathrm{H}_q(x_1, x_2), x_3), \qquad M = \begin{pmatrix} -1 & 3 & 0 & 0 \\ 0 & 0 & 3 & -1 \\ 0 & 1 & 1 & 0 \end{pmatrix}.$$

Thus,

$$\tilde{\gamma} = G_{p,q} \circ \gamma = \mathrm{W}_{p,\frac{3}{8},\frac{5}{8}} \circ \gamma_2, \qquad \gamma_2 := \psi \circ \gamma_1, \qquad \gamma_1 := M \circ \gamma$$

Since $\gamma : [a, b] \to \mathbb{R}^4$ is Lipschitz, we have that a.e. in (a, b)

1. $\gamma_1 : [a, b] \to \mathbb{R}^3$ is Lipschitz and $\gamma_1'(t) = M\gamma'(t)$,
2. $\gamma_2 : [a, b] \to \mathbb{R}^2$ is Lipschitz and $\gamma_2'(t) = D\psi(\gamma_1(t))\gamma_1'(t)$, with (see Theorem 22 in the Appendix)

$$D\psi(x) = \begin{pmatrix} D\mathrm{H}_q(x_1, x_2) & 0 \\ (0,0) & 1 \end{pmatrix}, \qquad \forall x = (x_1, x_2, x_3) \in \mathbb{R}^3$$

3. $\tilde{\gamma} = \mathrm{W}_{p,\frac{3}{8},\frac{5}{8}} \circ \gamma_2$ is Lipschitz (see Theorem 20 in the Appendix) $\tilde{\gamma}'(t) = D\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\gamma_2(t))\gamma_2'(t)$.

Collecting all of the above we have

$$\tilde{\gamma}'(t) = D\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\psi \circ M \circ \gamma(t))D\psi(M \circ \gamma(t))M\gamma'(t), \quad a.e.(0,1)$$

so that Eq. 52 provides an adequate definition of a Generalized Jacobian of $G_{p,q}$. □

Uniform bounds for the *Generalized Jacobian* $DG_{p,q}$ defined in Eq. 52 can be readily computed. Since

$$DG_{p,q}(x) = -\tfrac{1}{16} D\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\xi) \left(\begin{array}{c|c|c} -D_1\mathrm{H}_q(\rho) & 3D\mathrm{H}_q(\rho) & -D_2\mathrm{H}_q(\rho) \\ \hline 0 & (1,1) & 0 \end{array}\right),$$

$\xi = (H(L_{3,1}(x), L_{1,3}(x)), L_{2,2}(x))$, $\rho = (L_{3,1}(x), L_{1,3}(x))$, we can write

$$16D_1G_{p,q}(x) = D_1\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\xi)D_1\mathrm{H}_q(\rho), \qquad 16D_4G_{p,q}(x) = D_1\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\xi)D_2\mathrm{H}_q(\rho),$$

$$16(D_2G_{p,q}(x), D_3G_{p,q}(x)) = -3D_1\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\xi)D\mathrm{H}_q(\rho) - D_2\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\xi)(1,1).$$

Taking into account that (see [19])

$$0 \le D_1\mathrm{H}_q(x) \le q, \quad 0 \le D_2\mathrm{H}_q(x) \le q, \quad ||D\mathrm{H}_q(x)||_1 \le q, \quad \forall x \in \mathbb{R}^2,$$

we have

$$||DG_{p,q}(x)||_1 \le \frac{1}{16}\left(4|D_1\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\xi)| \cdot ||D\mathrm{H}_q(\rho)||_1 + 2|D_2\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\xi)|\right) \le \frac{q}{4}|D_1\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\xi)| + \frac{1}{8}|D_2\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\xi)|.$$

As observed in the Appendix, the bi-infinite matrix with the following non-zero entries $\forall n \in \mathbb{Z}$

$$\begin{aligned} (D\mathrm{SWH}^{[2]}_{p,q}(w))_{[2n,n-2:n+1]} &= 2DG_{p,q}(\chi_{n-2,n+1}w), \\ (D\mathrm{SWH}^{[2]}_{p,q}(w))_{[2n+1,n-2:n+2]} &= \tfrac{1}{2}(0,0,1,0,0) - (DG_{p,q}(\chi_{n-2,n+1}w) + DG_{p,q}(\chi_{n-1,n+2}w)) \end{aligned} \tag{53}$$

defines a Generalized Jacobian of the second difference scheme. Then,

$$\|D(\mathrm{SWH}^{[2]}_{p,q})_{[2n,:]}(w)\|_1 \le \frac{q}{2}|D_1\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\xi)| + \frac{1}{4}|D_2\mathrm{W}_{p,\frac{3}{8},\frac{5}{8}}(\xi)|, \tag{54}$$

$$\|D(\mathrm{SWH}^{[2]}_{p,q})_{[2n+1,:]}(w)\|_1 \le \frac{1}{2} + \|D(\mathrm{SWH}^{[2]}_{p,q})_{[2n,:]}(w)\|_1. \tag{55}$$

The strategy advocated in [22] for establishing stability relies on the relations (A.19), Eq. A.20 and Corollary 24, in the Appendix. The desired contractivity property (42) follows from the ability to find a uniform bound for products of Generalized Jacobians of the second difference scheme.

Taking into account the bounds obtained previously for the generalized gradient of the function $\mathrm{W}_{p,a,b}(x, y)$ for $p = 1, 2$ (these bounds are optimal for $p = 2$) and the

known (optimal) bounds for $\mathrm{H}_q(x, y)$, we display in Table 1 the values of the bound in Eq. 54 for $1 \leq p, q \leq 2$.

Since (S represents any scheme in the family)

$$||DS^{[2]}||_\infty = \max_n\{||(DS^{[2]})_{[2n,:]}||_1, ||(DS^{[2]})_{[2n+1,:]}||_1\},$$

from Eq. 54, Eq. 55 and the results in Table 1, we cannot ensure $||DS^{[2]}||_\infty < 1$ for any member of our family of schemes, or, equivalently, we cannot ensure the contractivity of the second difference scheme.

As observed in [22], the technique described in the Appendix will be successful when the 1-norm of some row of the Generalized Jacobian is strictly uniformly bounded by 1. In this case, and by carefully considering the form of the matrix products, it may be possible to arrive at products of Generalized Jacobians whose norms are strictly bounded by 1. Taking into account the bounds in Table 1, it seems that this strategy might only be feasible for $p = q = 1$, because such case is the only one where Eq. 54 is strictly bounded by 1. Since the task of obtaining theoretically bounds for products of Generalized Jacobian is very involved, we examine the issue numerically in the following section.

5 Numerical experiments

In this section we carry out a series of numerical experiments that illustrate the theoretical developments of the previous sections. We consider first the issue of the stability of the new schemes, from a numerical perspective. In addition, we also consider the smoothness of limit functions, as well as the approximation order of the non-oscillatory 6-point schemes, comparing the numerical results with those obtained for the Power_p schemes.

5.1 Stability

To examine the question of stability for each chosen subdivision scheme, S, we compute the quantities $C_S^j(h)$ for each $j \geq 1$, $0 < h < 1$,

$$C_S^j(h) \approx \sup_{f^0} \sup_{\|\theta\|_\infty = 1} \frac{1}{h} \|S^j(f^0 + h\theta) - S^j(f^0)\|, \qquad h > 0. \tag{56}$$

For the computation we consider a sufficiently large set of sequences $f^0 = \{f_i^0\}$ and perturbation sequences $\theta = \{\theta_i\}$, with components randomly chosen from the set $\{-1, 0, 1\}$ (hence $\|\theta\|_\infty = 1$). We notice that if S is Lipschitz stable, $C_S^j(h) \leq C$,

Table 1 Bounds of $||D(\mathrm{SWH}^{[2]}_{p,q})_{[2n,:]}||_1$

$p\backslash q$	1	2
1	13/16	21/16
2	52/30	92/30

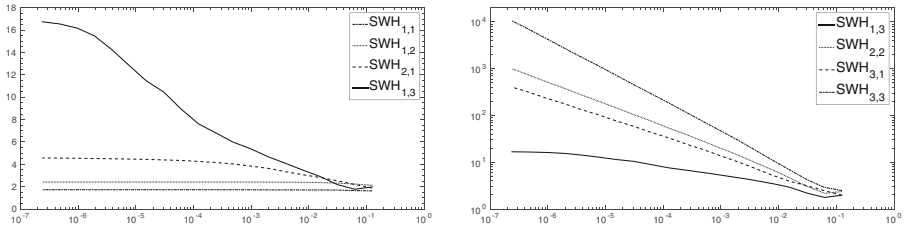


Fig. 8 Plot of $C_S^L(h)$, with respect to h for $L = 10$, $S = \mathrm{SWH}_{p,q}$ for several values of p, q

$\forall j \geq 1, \forall h > 0$, so that any deviation with respect to this behavior may be considered a sign of the instability of the scheme.

Figure 8 displays $C_S^{10}(h)$ as a function of h, for $S = \mathrm{SWH}_{p,q}$ and various values of p, q. The plots in Fig. 8 seem to indicate that the scheme $\mathrm{SWH}_{p,q}$ is not stable for $p + q > 3$.

As observed in Theorem 1, Lipschitz stability follows from the contractivity of an appropriate power of the second difference scheme, which is a condition that can also be examined numerically. For $h = 10^{-7}$, the smallest value of h considered in Fig. 8, we compute

$$T_S^j(h) \approx \sup_{f^0} \sup_{\|\theta\|_\infty = 1} \frac{1}{h} \|(S^{[2]})^j(f^0 + h\theta) - (S^{[2]})^j(f^0)\|, \qquad h > 0, \tag{57}$$

in order to check if the hypothesis **S2**, in its equivalent formulation (42), is fulfilled.

In Fig. 9 we display the values of $T_S^j(h)$ for $1 \leq j \leq 6$. We clearly notice that $\exists L \geq 1$ such that $T_S^L(h) < 1$ for $S = \mathrm{SWH}_{p,q}$, $(p, q) = (1, 1), (1, 2)$, a behavior that would be obtained if these schemes were stable. On the other hand, $T_S^j(h)$ appears to grow with j for $p + q \geq 4$ indicating that condition (42) is not fulfilled.

We also observe that the T_S^j does not grow for $S = \mathrm{SWH}_{2,1}$, but it does not appear to become smaller than one, an indication that condition (42) is only a sufficient condition for stability.

Table 2 summarizes the observations that might be deduced from our testing process. The same experiments were performed for the $\mathrm{SHW}_{q,p}$ family, with similar results.

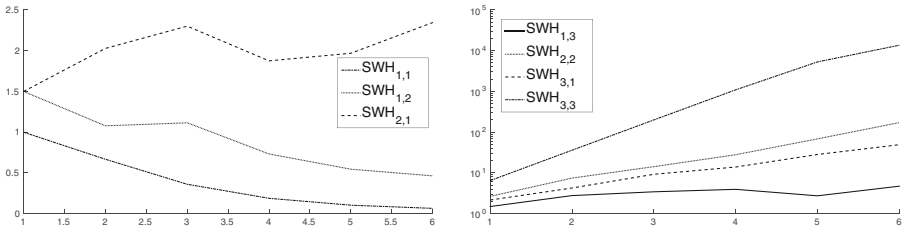


Fig. 9 Plot of $(T_S^{[2]}(h))^j$ with respect to j for $S = \mathrm{SWH}_{p,q}$, $h = 10^{-7}$ and several values of p, q

Table 2 Stability perspective on $\mathrm{SWH}_{p,q}$ and $\mathrm{SHW}_{q,p}$. ✓ suspected stability. × unstability

$q\backslash p$	1	2	3
1	✓	✓	×
2	✓	×	×
3	×	×	×

5.2 Smoothness of the limit function

In Section 3.2 we have shown that the schemes proposed in this paper are convergent, i.e. $S^\infty f^0$ is a continuous function $\forall f^0 \in l_\infty(\mathbb{Z})$ for $S = \mathrm{SWH}_{p,q}, \mathrm{SHW}_{q,p}$. The regularity of the limit function obtained from Theorem 1 (see Remark 10) seems to be much smaller than what is observed in practice. In this section we perform a numerical study of the regularity of the proposed schemes based on the following observations (see [5]): Let us assume that $f(x) = S^\infty f^0 \in C^{r-}$, S an interpolatory subdivision scheme, and $r = l + \beta$, $l \in \mathbb{N}$, $0 \leq \beta < 1$. Then,

$$f^{(l)}(x_{i+1}^k) - f^{(l)}(x_i^k) \approx C(x_{i+1}^k - x_i^k)^\beta = Ch_k^\beta.$$

Since $f_i^k = f(x_i^k)$, and $f^{(l)}(x_i^k) \approx \nabla^l f_i^k/(h_k)^l = 2^{lk}\nabla^l f_i^k/h_0$, we have

$$f^{(l)}(x_{i+1}^k) - f^{(l)}(x_i^k) \approx 2^{lk}(\nabla^l f_{i+1}^k - \nabla^l f_i^k)/h_0 = 2^{lk}\nabla^{l+1} f_i^k/h_0.$$

Hence, we expect that

$$\frac{2^{lk}\nabla^{l+1} f_i^k}{2^{l(k+1)}\nabla^{l+1} f_i^{k+1}} \approx \frac{h_k^\beta}{h_{k+1}^\beta} = 2^\beta \quad \leftrightarrow \quad \frac{\nabla^{l+1} f_i^k}{\nabla^{l+1} f_i^{k+1}} \approx 2^{l+\beta} \leftrightarrow r = l+\beta \approx \log_2\left(\frac{\|\nabla^{l+1} f^k\|_\infty}{\|\nabla^{l+1} f^{k+1}\|_\infty}\right).$$

Therefore, in order to estimate the numerically regularity of a subdivision scheme, S, in a given region, $[a, b]$, we compute (for several values of l)

$$\mathcal{R}_S^l([a,b]) = \log_2\left(\frac{\varrho_6^l}{\varrho_7^l}\right), \quad \varrho_k^l := \sup\{|(\nabla^{l+1} f^k)_i| : x_i^k \in [a,b]\}. \tag{58}$$

For our numerical testing process we consider $f_i^0 = F(x_i^0)$, $F(x) = e^{-2x^2}$, $(x_i^0)_i$ an h_0-uniform grid. In Table 3 we display the results corresponding to an initial mesh with $N = 18$ points in $[-6, 6]$, so that $x = 0$ does not belong to the initial mesh, and in Table 4 we display the results obtained when a uniform mesh with $N = 21$ points in $[-6, 6]$, so that $x = 0$ belongs to the initial grid, is used to compute f^0 (see also Fig. 10).

The tables show that, for these examples, the global regularity of the limit function is at least $r = 1$. Moreover, when $x = 0$ (the abscissa of the maximum of $F(x)$) is included in the initial grid, the global regularity of the limit function obtained with the Power_p schemes is smaller than that obtained with the new schemes when $\max\{p, q\} > 1$. According to Table 3, in this case the limit function seems to be globally C^1 for $\max\{p, q\} > 1$. In addition, we also observe in both tables that for $(p, q) = (2, 2)$ we get the same smoothness as for the $S_{3,3}$ scheme, around $x = 0$, much higher than that of the Power_2 scheme.

Table 3 Regularity of $S^\infty f^0$: $\mathcal{R}^l_S([a,b])$ in Eq. 58 for $S = S_{3,3}$, S_{H_2} and $\mathrm{SWH}_{p,q}$, for $3 \le p+q \le 4$

l	$S_{3,3}$		$\mathcal{R}^l_{\mathrm{SWH}_{1,2}}$		$\mathcal{R}^l_{\mathrm{SWH}_{2,1}}$		$\mathcal{R}^l_{\mathrm{SWH}_{2,2}}$		$\mathcal{R}^l_{S_{H_2}}$	
0	0.95	1.00	1.00	1.00	0.96	1.00	0.95	1.00	0.94	1.00
1	1.99	1.99	1.00	1.00	1.75	1.50	1.99	1.48	1.90	1.08
2	2.81	2.84	1.00	1.00	1.64	1.01	2.84	1.00	2.06	1.08
3	2.82	2.83	1.00	1.00	1.64	1.00	2.91	1.00	2.04	1.07
4	2.83	2.83	1.00	1.00	1.64	1.00	2.85	1.00	1.78	1.07

Left columns: $[a,b] = [-0.1, 0.1]$. Right columns: $[a,b] = [-3, 3]$. Initial data and limit functions displayed in Fig. 10, left column

5.3 Approximation order

The order of approximation of a subdivision scheme measures its ability to reconstruct smooth functions from relatively coarse samples. Given $f^0 = \{F(nh)\}_{n\in\mathbb{Z}}$ where $F(x)$ is a smooth function, we study the difference between the limit function $f^\infty(x) = S^\infty f^0(x)$ and $F(x)$ in a given region by considering $S^\infty f^0 \approx S^L f^0$ (with $L = 7$ in all test cases) and measuring

$$E_{S,[a,b]}(h) := \max\{|(S^L f^0)_n - F(n2^{-L}h)|,\ n2^{-L}h \in [a,b]\} \approx ||S^\infty f^0 - F||_{L^\infty([a,b])}. \quad (59)$$

In Tables 5-8 we display $E_{S,[a,b]}(h)$ for different values of h, different regions $[a,b]$ and different functions $F(x)$. The tables also show the numerical order of accuracy, r_n, obtained by a least squares fit of the data $(\log_2(h_l), \log_2(E_S(h_l))$ for $h_l = h_0/2^l$, and a given value of h_0.

The ultimate purpose of the numerical testing process is twofold. On one hand the Tables show that, by choosing p, q appropriately, it is possible to obtain the same order of approximation (as well as errors of a similar magnitude) as that of the $S_{3,3}$ scheme. In addition, the Tables also show that, for each scheme in the family, the order of approximation of the limit function is the same as the theoretical order of approximation after one iteration, r_t in the Tables. For the new schemes proposed in the paper, r_t is obtained in Corollary 12 for convex regions, but it can also be

Table 4 Same as Table 3. Initial data and limit functions displayed in Fig. 10, right column

l	$\mathcal{R}^l_{S_{3,3}}$		$\mathcal{R}^l_{\mathrm{SWH}_{1,2}}$		$\mathcal{R}^l_{\mathrm{SWH}_{2,1}}$		$\mathcal{R}^l_{\mathrm{SWH}_{2,2}}$		$\mathcal{R}^l_{S_{H_2}}$	
0	0.91	1.00	1.00	1.00	1.00	1.00	0.95	1.00	1.00	1.00
1	1.99	1.99	1.69	1.69	1.44	1.44	1.93	1.93	1.00	1.00
2	2.82	2.82	1.63	1.63	1.48	1.48	2.47	1.34	1.00	1.00
3	2.83	2.83	1.63	1.63	1.48	1.48	2.58	1.27	1.00	1.00
4	2.83	2.83	1.38	1.38	1.47	1.47	2.64	1.30	1.00	1.00

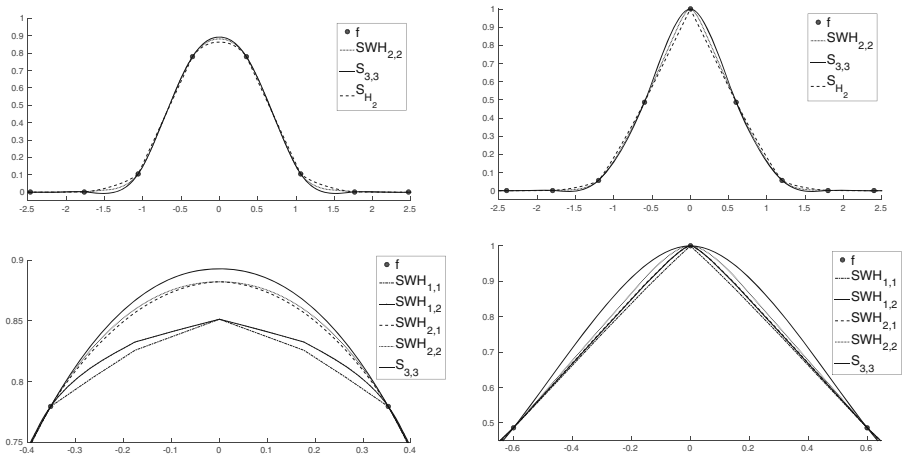


Fig. 10 Limit function from coarse Gaussian data. Initial data marked with •

obtained by Taylor expansions in regions where $\mathcal{L}_{l,r} \circ \nabla^2$ does not change sign. For the purpose of illustration, we include the result of the algebraic computation (done with Mathematica) for $p = 2, q = 2$,

$$(\mathrm{SWH}_{p,q} f)_{2j+1} = F((2n+1)h/2) + \frac{h^6}{2048}\Big(\frac{15F^{IV}((2j+1)h/2)^2}{F''((2j+1)h/2)} + 10F^{VI}((2j+1)h/2)\Big), \tag{60}$$

which holds for smooth functions, provided that $(\mathcal{L}_{l,r} \circ \nabla^2 f)_{2j+1}$ have the same sign. We notice further that if F is a smooth function and $f_j = F(x_j)$, straightforward Taylor expansions lead to

$$\begin{aligned}
(\mathcal{L}_{2,2}\nabla^2 f)_{2n+2m+1} &= 2F''(x_n)h^2 + (1+2m)F'''(x_n)h^3 + (\tfrac{2}{3} + m + m^2)F^{IV}(x_n)h^4 + O(h^5),\\
(\mathcal{L}_{3,1}\nabla^2 f)_{2n+2m+1} &= 2F''(x_n)h^2 + (1+2m)F'''(x_n)h^3 + (-\tfrac{1}{3} + m + m^2)F^{IV}(x_n)h^4 + O(h^5),\\
(\mathcal{L}_{1,3}\nabla^2 f)_{2n+2m+1} &= 2F''(x_n)h^2 + (1+2m)F'''(x_n)h^3 + (-\tfrac{1}{3} + m + m^2)F^{IV}(x_n)h^4 + O(h^5).
\end{aligned}$$

Table 5 Approximation order for Gaussian data: $E_{S,[a,b]}(h_l)$, $[a,b] = [-0.4, 0.4]$, $h_l = 2^{-l}h_0$, and several schemes

l	$S_{3,3}$	S_{H_2}	S_{H_3}	$\mathrm{SWH}_{1,1}$	$\mathrm{SWH}_{2,1}$	$\mathrm{SWH}_{2,2}$	$\mathrm{SWH}_{3,1}$	$\mathrm{SWH}_{3,2}$
0	4.3e-6	2.4e-4	1.1e-4	1.7e-4	1.7e-5	6.3e-6	1.6e-5	3.5e-6
1	7.2e-8	1.8e-5	7.0e-6	1.1e-5	5.4e-7	1.0e-7	5.3e-7	5.7e-8
2	1.1e-9	1.2e-6	4.4e-7	7.5e-7	1.7e-8	1.7e-9	1.6e-8	9.0e-10
3	1.8e-11	8.0e-8	2.7e-8	4.7e-8	5.3e-10	2.7e-11	5.3e-10	1.4e-11
r_n	5.96	3.85	3.98	3.95	4.99	5.94	4.98	5.97
r_t	6	4	4	4	5	6	5	6

Hence, at smooth convex regions (at least for h small enough) $\mathcal{L}_{l,r} \circ \nabla^2$ does not change sign. Moreover, if $x_n = nh$ is an inflection point and $F'''(x_n) \neq 0$, the formulas above show that, for h small enough and $(l, r) = (1, 3), (3, 1), (2, 2)$, $(\mathcal{L}_{l,r} \nabla^2 f)_{2n+m}$ do not change sign for $m \geq 0$ or $m < 0$, and the calculation of r_t by Taylor expansions is feasible. This is the case of the two functions considered in the numerical tests.

The testing process below is carried out for the family of schemes $\mathrm{SWH}_{p,q}$. We have also performed the same study for the $\mathrm{SHW}_{q,p}$ family. As expected, the resulting tables are similar and the conclusions are also the same, hence we omit them. We include the errors corresponding to the $S_{3,3}$ and S_{H_p} schemes for the sake of comparison. An $*$ in the Table 6 means that it is not possible to find r_t by Taylor expansions.

5.3.1 Gaussian data

We consider $F(x) = e^{-2x^2}$, $h_0 = 0.1$. Table 5 displays the results for the region $[a, b] = [-0.4, 0.4]$, where F is convex and $|F''(x)| \geq |F''(0.4)| \approx 1.04$. We observe that the estimated order of approximation of $\mathrm{SWH}_{p,q}$ coincides with r_t, the order of approximation after one iteration in the convex region.

In Table 6 we display the corresponding results for $[a, b] = [-1, -0.3]$, which contains the inflection point $x = 0.5$. The Table indicates that the computed r_n coincides with r_t for all the nonlinear 6-point schemes. As observed above, r_t can still be computed by Taylor expansions for the 6-points nonlinear schemes. It should be noticed that for $F(x) = e^{-2x^2}$

$$\frac{F^{IV}(x)^2}{F''(x)} = 64 e^{-2x^2} \frac{\left(16x^4 - 24x^2 + 3\right)^2}{4x^2 - 1}$$

which has a vertical asymptote in $x = \pm 0.5$ and hence it is unbounded around the inflection points. Since $F''((2n+1)h/2) = F'''(nh)h/2 + O(h^2)$, Eq. 60 leads to

$$(\mathrm{SWH}_{p,q} ddf)_{2n+1} = F((2n+1)h/2) + O(h^5),$$

around the inflection point, i.e., the order of approximation after one iteration is 5 in the non-convex region. Performing the analogous computation for $(p, q) = (3, 1)$,

Table 6 Same as Table 5 for $[a, b] = [-1, -0.3]$

l	$S_{3,3}$	S_{H_2}	S_{H_3}	$\mathrm{SWH}_{1,1}$	$\mathrm{SWH}_{2,1}$	$\mathrm{SWH}_{2,2}$	$\mathrm{SWH}_{3,1}$	$\mathrm{SWH}_{3,2}$
0	3.1e-6	6.1e-4	5.9e-4	1.0e-4	1.5e-5	8.9e-6	1.5e-5	3.3e-6
1	5.1e-8	7.7e-5	7.6e-5	7.3e-6	5.3e-7	2.1e-7	5.3e-7	5.0e-8
2	8.1e-10	9.7e-6	9.6e-6	4.6e-7	1.7e-8	5.7e-9	1.6e-8	7.4e-10
3	1.3e-11	1.2e-6	1.2e-6	2.9e-8	5.3e-10	1.5e-10	5.3e-10	1.1e-11
r_n	5.97	2.99	2.98	3.95	4.95	5.26	4.95	6.07
r_t	6	*	*	4	5	5	5	6

Table 7 Approximation order for Tangent data: $E_{S,[a,b]}(h_l)$, $[a, b] = [0.1, 0.3]$, $h_l = 2^{-l}h_0$ and several schemes

l	$S_{3,3}$	S_{H_2}	S_{H_3}	$\mathrm{SWH}_{1,1}$	$\mathrm{SWH}_{2,1}$	$\mathrm{SWH}_{2,2}$	$\mathrm{SWH}_{3,1}$	$\mathrm{SWH}_{3,2}$
2	1.6e-5	7.3e-6	1.4e-4	3.3e-4	9.0e-5	2.8e-5	1.8e-5	1.7e-5
3	2.8e-7	4.8e-7	1.1e-5	2.0e-5	2.3e-6	4.5e-7	2.1e-6	2.8e-7
4	4.7e-9	3.1e-8	7.5e-7	1.3e-6	6.9e-8	7.6e-9	6.6e-8	4.7e-9
5	7.7e-11	1.9e-9	5.0e-8	1.3e-7	2.1e-9	1.2e-10	2.0e-9	7.7e-11
r_n	5.88	3.96	3.80	4.09	5.11	5.94	5.02	5.93

(3, 2) we find that the 'theoretical' approximation order after one iteration is 5 and 6, respectively, also in the non-convex region.

We also remark that, in all the tables displayed, the magnitude of the errors corresponding to the 6-point nonlinear schemes whose order of accuracy is 4, 5 or 6 are similar to those of $S_{3,3}$ and better than that of the Power_p schemes.

5.3.2 Tangent data

We repeat the previous study for $F(x) = \tan(\pi x)$. In this case, the function is convex in the interval $[0.1, 0.3]$, and changes convexity at $[-0.25, 0.25]$. We consider $h_0 = 0.1$. The conclusions are similar. In particular, the magnitude of the errors is similar to those of $S_{3,3}$, and better than those obtained with the Power_p schemes, for $\mathrm{SWH}_{p,q}$ schemes whose order of accuracy is 4, 5 or 6. We also remark that order of approximation of the scheme coincides with r_t (not displayed in the tables). Notice that for $F(x) = \tan(\pi x)$

$$F^{IV}(x)/F''(x) = 8\pi^6(\cos(2\pi x) - 5)^2 \tan(\pi x) \sec^6(\pi x)$$

which is a bounded function around $x = 0$. Hence, according to Eq. 60, the order of approximation after one iteration of $\mathrm{SWH}_{p,q}dd$ is 6 also in the non-convex region.

Table 8 Same as Table 7 for $[a, b] = [-0.25, 0.25]$

l	$S_{3,3}$	S_{H_2}	S_{H_3}	$\mathrm{SWH}_{1,1}$	$\mathrm{SWH}_{2,1}$	$\mathrm{SWH}_{2,2}$	$\mathrm{SWH}_{3,1}$	$\mathrm{SWH}_{3,2}$
2	3.5e-6	6.2e-5	6.2e-5	2.1e-4	2.2e-5	6.1e-6	7.1e-5	3.7e-6
3	6.0e-8	7.8e-6	7.8e-6	1.2e-5	6.1e-7	9.9e-8	5.6e-7	6.1e-8
4	1.0e-9	9.7e-7	9.7e-7	7.2e-7	1.8e-8	1.6e-9	1.8e-8	1.0e-9
5	1.6e-11	1.2e-7	1.2e-7	4.4e-8	5.6e-10	2.6e-11	5.5e-10	1.6e-11
r_n	5.90	3.00	3.00	4.07	5.08	5.93	5.01	5.93

6 Conclusions

We have constructed two families of non-oscillatory subdivision schemes that can be considered nonlinear versions of the 6-point Deslauries-Dubuc interpolatory subdivision scheme. We have studied their convergence by exploiting the (piecewise) smoothness properties of the functions that define these subdivision schemes, following a novel technique developed in [19]. The stability of these schemes turns out to be more difficult to study with the techniques employed in [22] and we have explored this issue computationally. The numerical results reveal indeed that the techniques based on finding appropriate bounds for the Generalized Jacobian of the second difference scheme, as in [22], have no chance to succeed, except for $(p, q) = (1, 1)$ (where contractivity might be proven for $L = 2$) and $(p, q) = (2, 1)$ (for $L = 4$).

We have also performed several numerical experiments that suggest that the approximation properties of the new schemes can be as good as those of the 6-point linear scheme when reconstructing smooth functions. In addition, numerical experiments show that the smoothness of the limit functions obtained from convex data may be larger than the smoothness of the limit functions obtained with Power_p schemes.

Acknowledgments The authors acknowledge support from Project MTM2014-54388 (MINECO, Spain) and the FPU14/02216 grant (MECD, Spain).

Appendix A. Generalized Gradients and Generalized Jacobians

The schemes considered in this paper involve nonlinear functions that are continuous but only piecewise differentiable. As shown in [22], the theory developed by Oswald and Harizanov in [19] can be used to analyze the stability of such schemes, provided that the functions that define them belong to a special class of piecewise smooth functions, for which uniformly bounded *Generalized Gradients* can be defined. For the sake of completeness, and ease of future reference, we include here the relevant theoretical results, including proofs, in a framework broad enough to cover the schemes considered in [19, 22] and in this paper. We also remark that the theory of Generalized Gradients has been developed in greater generality by Clarke in a series of papers (see [7, 21]).

The $C^1_{pw}(\mathbb{R}^m)$ class of functions: Generalized Gradients

The $C^1_{pw}(\mathbb{R}^m)$ class of functions was defined in [19]. Functions $\psi : \mathbb{R}^m \to \mathbb{R}$ in this class are continuous, piecewise smooth and have directional derivatives except (maybe) at $0 \in \mathbb{R}^m$ and *across* certain hyperplanes separating regions of C^1 smoothness. Directional derivatives *along* the separating hyperplanes do, also, exist.

We shall assume that there exist a finite number of hyperplanes $\{\mathcal{H}_i\}_i$, $0 \in \mathcal{H}_i$, such that ψ is continuously differentiable $\forall x \in \mathbb{R}^m \setminus \bigcup_i \mathcal{H}_i$ and $\psi|_{\mathcal{H}_i}$ is continuously differentiable (as a function of $m-1$ variables) except maybe at $x = 0$. In this context $\mathbb{R}^m \setminus \bigcup_i \mathcal{H}_i$ is the union of a finite number of disjoint open convex *sectors* in $\mathbb{R}^m$,

which we shall denote by Ω_j. Therefore $\mathbb{R}^m = (\bigcup_j \Omega_j) \cup (\bigcup_i \mathcal{H}_i)$, the sets $\bigcup_j \Omega_j$ and $\bigcup_i \mathcal{H}_i$ are disjoint, and $\partial\Omega_j$, the boundary of Ω_j, is always included in the union of the separating hyperplanes.

In addition, functions in $C^1_{pw}(\mathbb{R}^m)$ have uniformly bounded gradients in smooth regions, i.e.

$$C_{||\cdot||} := \sup_j \sup_{p\in\Omega_j} ||\nabla\psi(p)|| < \infty, \tag{A.1}$$

and the smooth gradients satisfy the following compatibility condition on the separating hyperplanes: Let $0 \neq x \in \partial\Omega \subset \mathcal{H}$, where Ω is a smoothness region for ψ and $\mathcal{H}$ is one of the hyperplanes separating the regions of smoothness of ψ. Then

$$(\lim_{x\leftarrow p\in\Omega} \nabla\psi(p)) \cdot w = D_w\psi(x), \qquad \forall 0 \neq w \in \mathcal{H} \tag{A.2}$$

where $D_w\psi(x)$ is the derivative of ψ at x in the direction of w.

Functions in $C^1_{pw}(\mathbb{R}^m)$ admit *Generalized Gradients*. As the standard gradient of a smooth function, the main property of a Generalized Gradient is that the associated linear map recovers all directional derivatives that *'make sense'* at a given point. Conditions (A.1)-(A.2) ensure that

$$D\psi(x) := \begin{cases} \nabla\psi(x) & \text{if } x \in \Omega \text{ (smoothness region)} \\ \lim_{x\leftarrow p\in\Omega} \nabla\psi(p) & \text{if } x \in \partial\Omega \end{cases} \qquad \forall 0 \neq x \in \mathbb{R}^m, \quad D\psi(0) = 0 \tag{A.3}$$

provides an adequate definition of a Generalized Gradient of $\psi \in C^1_{pw}(\mathbb{R}^m)$, since for each $0 \neq x \in \mathbb{R}^m$, $D\psi(x)$ defines a linear map that satisfies

$$D_v\psi(x) = D\psi(x) \cdot v \tag{A.4}$$

for any $\vec{0} \neq v \in \mathbb{R}^m$ when x belongs to a smoothness region, and also for any $0 \neq v \in \mathcal{H}$ when $0 \neq x \in \mathcal{H}$.

Notice that $D\psi(x)$ in Eq. A.3 might not be uniquely defined when x belongs to a hyperplane separating two or more regions of smoothness, if the limit in Eq. A.3 is different for different smoothness regions with a common boundary. However the compatibility condition (A.2) ensures (A.4) for all directional derivatives that *make sense*, independently of the chosen definition for the vector $D\psi(x)$.

The following results generalize some of the properties satisfied by the gradient of a smooth function.

Lemma 18 *(Generalized MVT) Let $\psi \in \mathbb{C}^1_{pw}(\mathbb{R}^m)$, $x, y \in \mathbb{R}^m$, $x \neq y$ and $\Gamma := \{tx + (1-t)y, \quad t \in (0,1)\}$. Then, if $\Gamma \subset \Omega$ (smoothness region for ψ) or $\Gamma \subset \mathcal{H} - \{0\}$, then $\exists\, \hat{t} \in (0,1)$ such that*

$$\psi(x) - \psi(y) = D\psi(\xi)(x-y), \qquad \xi = \hat{t}x + (1-\hat{t})y,$$

where $D\psi$ is a generalized gradient of ψ.

Proof Define $\gamma : [0,1] \to \mathbb{R}^m$, $g : [0,1] \to \mathbb{R}$

$$\gamma(t) := tx + (1-t)y = y + t(x-y), \qquad g(t) := \psi(\gamma(t)). \tag{A.5}$$

Then,

$$g(t+h) - g(t) = \psi(\gamma(t) + h(x-y)) - \psi(\gamma(t)) \qquad \forall h \in \mathbb{R} \tag{A.6}$$

hence, under the hypothesis of the Lemma, $\gamma((0,1)) =: \Gamma \subset \Omega$ or $\Gamma \subset \mathcal{H} - \{0\}$, and

$$g'(t) = \lim_{h\to 0} \frac{g(t+h) - g(t)}{h} = D_{x-y}\psi(\gamma(t)) = D\psi(\gamma(t))(x-y) \tag{A.7}$$

for any Generalized Gradient $D\psi$ and for any $t \in (0,1)$ (notice that if $x, y \in \mathcal{H}$, a separating hyperplane, then $x - y \in \mathcal{H}$). Since $g(t)$ is differentiable in $(0,1)$, by the classical Mean Value Theorem (MTV)

$$\exists \hat{t} \in (0,1): \quad \psi(x) - \psi(y) = g(1) - g(0) = g'(\hat{t}) = D\psi(\xi)(x-y), \tag{A.8}$$

with $\xi = \gamma(\hat{t}) = \hat{t}x + (1-\hat{t})y$. □

Corollary 19 *Let $\psi \in C^1_{pw}(\mathbb{R}^m)$. Then $\forall x, y \in \mathbb{R}^m$*

$$|\psi(x) - \psi(y)| \le C_1 ||x-y|| \qquad C_1 = \sup_j \sup_{p\in\Omega_j} ||\nabla\psi(p)||_1 \tag{A.9}$$

Proof We consider again the straight line $\gamma(t) = y + t(x-y)$ and the function $g(t) = \psi(\gamma(t))$ in Eq. A.5. Notice that $\gamma(t)$ can either cut the separating hyperplanes at a finite number of points, or belong entirely to one of them. We prove (A.9) in each case.

Let us assume that that $0 \le t_1 < \ldots < t_k \le 1$ are such that $\gamma(t_k)$ are the cutting points with the (finite number of) hyperplanes separating the smoothness regions of ψ. We consider $t_0 = 0$ and $t_{k+1} = 1$. Without loss of generality, we may assume $t_0 < t_1 < \ldots < t_{k+1}$, hence $\Gamma_l := \{\gamma(t),\ t \in (t_l, t_{l+1})\}$ is included in one of the smoothness regions of ψ, for $l = 0, \cdots, k$. By the previous lemma,

$$g(t_{l+1}) - g(t_l) = g'(\hat{t}_l) = D\psi(\gamma(\hat{t}_l))(\gamma(t_{l+1}) - \gamma(t_l)) = D\psi(\gamma(\hat{t}_l))(t_{l+1} - t_l)(x-y), \quad l = 0, \ldots, k.$$

Thus, using Lemma 18 and considering the Generalized Gradient $D\psi$ in Eq. A.3, we can write

$$\begin{aligned} |\psi(x) - \psi(y)| &= |\sum_{l=0}^{k} g(t_{l+1}) - g(t_l)| = |\sum_{l=0}^{k} D\psi(\hat{t}_l)(t_{l+1} - t_l)(x-y)| \le \sum_{l=0}^{k} (t_{l+1} - t_l)|D\psi(\hat{t}_l)(x-y)| \\ &\le \sum_{l=0}^{k} (t_{l+1} - t_l)||D\psi(\hat{t}_l)||_1 ||x-y||_\infty \le C_1 ||x-y||_\infty \sum_{l=0}^{k} (t_{l+1} - t_l) = C_1 ||x-y||_\infty. \end{aligned}$$

since (A.1) leads to $||D\psi(x)||_1 \le C_1$, $\forall\, x \in \mathbb{R}^m$, for $D\psi$ in Eq. A.3.

Let us assume now that $\{\gamma(t), t \in \mathbb{R}\} \subset \mathcal{H}$. Then, either $\Gamma \in \mathcal{H} - \{0\}$ or $\exists \bar{t} \in (0,1)$ such that $\gamma(\bar{t}) = 0$. In both cases, the result follows easily from Lemma 18, using the same arguments as before. □

The following result establishes that the chain rule holds for the composition of $C^1_{pw}(\mathbb{R}^m)$ functions with Lipschitz curves.

Theorem 20 *Let $\gamma : [a, b] \to \mathbb{R}^m$ be a Lipschitz curve, $\psi : \mathbb{R}^m \to \mathbb{R}$ a function in $C^1_{pw}(\mathbb{R}^m)$ and $D\psi$ a generalized gradient of ψ. Then $\tilde{\gamma} = \psi \circ \gamma : [a, b] \to \mathbb{R}$ is also a Lipschitz curve, and*

$$\tilde{\gamma}'(t) = D\psi(\gamma(t))\gamma'(t) \qquad \text{a.e. in } (a, b). \tag{A.10}$$

Proof The curve $\tilde{\gamma}$ is Lipschitz because both γ and ψ are Lipschitz. Let A_γ be the set of points where γ is not differentiable and $A_{\tilde{\gamma}}$ the corresponding set for $\tilde{\gamma}$. Notice that both sets have zero measure, since Lipschitz functions are a.e. differentiable.

Since $\mathbb{R}^m = (\bigcup_i \mathcal{H}_i) \cup (\bigcup_j \Omega_j)$ it follows that $[a, b] = (\bigcup_i \tilde{\mathcal{H}}_i) \cup (\bigcup_j \tilde{\Omega}_j) \cup O$, where

$$\tilde{\mathcal{H}}_i := \{t \in [a, b], 0 \neq \gamma(t) \in \mathcal{H}_i\}, \qquad \tilde{\Omega}_j := \{t \in [a, b] : \gamma(t) \in \Omega_j\}, \qquad O = \{t \in [a, b] : \gamma(t) = 0\}.$$

Let us denote by E_i the set of isolated points in $\tilde{\mathcal{H}}_i$ and F the set of isolated points in O. These sets are countable[3], hence $(\bigcup_i E_i) \cup F$ is also countable. Therefore $B = A_\gamma \cup A_{\tilde{\gamma}} \cup (\bigcup_i E_i) \cup F$ is also a set of zero-measure. We shall check that (A.10) holds $\forall t \in (a, b)\backslash B$. Notice that $\forall t \in (a, b)\backslash B$ both $\tilde{\gamma}'(t)$ and $\gamma'(t)$ exist, since $t \notin A_{\tilde{\gamma}} \cap A_\gamma$, and can be computed as

$$\tilde{\gamma}'(t) = \lim_{n\to\infty} \frac{\tilde{\gamma}(t_n) - \tilde{\gamma}(t)}{t_n - t}, \qquad \gamma'(t) = \lim_{n\to\infty} \frac{\tilde{\gamma}(t_n) - \tilde{\gamma}(t)}{t_n - t} \tag{A.11}$$

for any sequence $\{t_n\}_n$ such that $t_n \to t$.

Let us assume that $t \in \tilde{\Omega}_j\backslash B$ and let $\{t_n\}$ be a sequence such that $t_n \to t$. Since γ is continuous, $\gamma(t_n) \to t$. Moreover, $\gamma(t_n) \in \Omega_j$ for n large enough, because Ω_j is an open set. Since Ω_j is convex, $\Gamma_n \subset \Omega_j$, where Γ_n is the segment joining $\gamma(t_n)$ and $\gamma(t)$. Hence, for n large enough, using Lemma 18 we can write

$$\tilde{\gamma}(t_n) - \tilde{\gamma}(t) = D\psi(\xi_n)(\gamma(t_n) - \gamma(t)), \qquad \xi_n \in \Gamma_n. \tag{A.12}$$

Since $\gamma(t_n) \to \gamma(t)$, we have that $\xi_n \to \gamma(t)$. In smooth regions, any generalized gradient must be uniquely defined as $D\psi(x) = \nabla\psi(x)$. Since $\psi|_{\Omega_j} \in C^1(\Omega_j)$, we have that

$$\lim_{n\to\infty} D\psi(\xi_n) = D\psi(\gamma(t)). \tag{A.13}$$

Hence,

$$\tilde{\gamma}'(t) = \lim_{n\to\infty} \frac{\tilde{\gamma}(t_n) - \tilde{\gamma}(t)}{t_n - t} = \lim_{n\to\infty} D\psi(\xi_n) \frac{\gamma(t_n) - \gamma(t)}{t_n - t} = D\psi(\gamma(t))\gamma'(t). \tag{A.14}$$

Let us assume now that $t \in \tilde{\mathcal{H}}_i\backslash B$, with $\mathcal{H}_i$ one of the separating hyperplanes. Since $t \notin E_i$, $\exists t_n \to t$, $t_n \in \tilde{\mathcal{H}}_i$. Since $\gamma(t)$ is continuous and $\gamma(t) \neq 0$, we can also

[3] If $x \in E$ is an isolated point, $\exists V_x$ open, such that $V_x \cap E = \{x\}$, and $\mathbb{R}$ is a second-countable space.

assume that $\Gamma_n \subset \mathcal{H}_i \backslash \{0\}$ and, by Lemma 18, we get (A.12). Notice that Eq. A.13 also holds, because of the requirement that $\psi|_{\mathcal{H}}$ is continuously differentiable (as a function of $\mathbb{R}^{m-1}$ variables) in $\mathcal{H}\backslash\{0\}$. Hence, Eq. A.14 also follows in this case.

Finally, let us assume that $t \in O \backslash B$. Since $t \in O \backslash F$, $\exists t_n \to t$, $\gamma(t_n) = 0$. Then

$$\tilde{\gamma}'(t) = \lim_{n\to\infty} \frac{\tilde{\gamma}(t_n) - \tilde{\gamma}(t)}{t_n - t} = 0 = D\psi(\gamma(t)) \lim_{n\to\infty} \frac{\gamma(t_n) - \gamma(t)}{t_n - t}$$

since $\tilde{\gamma}(t_n) = \tilde{\gamma}(t)$ and $\gamma(t_n) = \gamma(t)$. □

The chain rule (A.10) turns out to be the key property for our purposes. The next result shows that this property is also satisfied by a possibly wider class of functions.

Corollary 21 *Let $\psi \in C^1(\mathbb{R}^m)$, $M : \mathbb{R}^p \to \mathbb{R}^m$ a linear map and $\gamma : [a, b] \to \mathbb{R}^p$ a Lipschitz curve. Then the curve $\tilde{\gamma} := \psi \circ M \circ \gamma$ is also Lipschitz and satisfies*

$$\tilde{\gamma}'(t) = D\psi(M\gamma(t))M\gamma'(t) \quad a.e.\ in\ (a, b)$$

Proof Note that $\gamma_M := M \circ \gamma : [a, b] \to \mathbb{R}^m$ is continuous and $\gamma'_M(t) = M\gamma'(t)$ a.e in (a, b). Hence, the result follows from applying Theorem 20 to ψ and γ_M. □

This result allows us to write $D(\psi \circ M)(x) = D\psi(Mx)M$ as a *Generalized Gradient* of $\psi \circ M$ when $\psi \in C^1_{pw}(\mathbb{R}^m)$. Notice that $||D(\psi \circ M)(x)||$ is also uniformly bounded. Hence, we may also associate the notion of *Generalized Gradients* to certain functions, related (by composition) to $C^1_{pw}(\mathbb{R}^m)$ functions.

Generalized Jacobians

The notion of *Generalized Gradient* leads, in a rather natural way, to that of *Generalized Jacobian* for functions $\psi : \mathbb{R}^m \to \mathbb{R}^p$, $\psi = (\psi_1, \ldots, \psi_p)$, $\psi_i \in \mathbb{C}^1_{pw}(\mathbb{R}^m)$. As in the smooth case,

$$D\psi(x) := \begin{pmatrix} D\psi_1(x) \\ \vdots \\ D\psi_p(x) \end{pmatrix}$$

provides the definition of a *Generalized Jacobian* of ψ at $x \in \mathbb{R}^m$. Obviously, the definition may not be unique, but, as stated below, the linear maps associated to such matrices also satisfy the chain rule for the composition with Lipschitz curves.

Theorem 22 *Let $\gamma : [a, b] \to \mathbb{R}^m$ be a Lipschitz curve, $\psi : \mathbb{R}^m \to \mathbb{R}^p$, $\psi = (\psi_1, \cdots, \psi_p)$, a function such that $\psi_i \in C^1_{pw}(\mathbb{R}^m)$ and $D\psi$ a generalized Jacobian of ψ. Then $\tilde{\gamma} = \psi \circ \gamma$ is also Lipschitz and*

$$\tilde{\gamma}'(t) = D\psi(\gamma(t))\gamma'(t), \qquad a.e.\ in\ (a, b). \tag{A.15}$$

Proof Since $\tilde{\gamma}_i = \psi_i \circ \gamma$, it is Lipschitz and Eq. A.15 holds a.e. for each $1 \leq i \leq p$. Hence

$$\tilde{\gamma}_i'(t) = D\psi_i(\gamma(t))\gamma'(t), \quad \forall t \in [a,b]\backslash I_i \quad \rightarrow \quad \tilde{\gamma}'(t) = D\psi(\gamma(t))\gamma'(t), \quad \forall t \in [a,b]\backslash \bigcup_i I_i$$

since $\bigcup_i I_i$ is a null set. Hence the chain rule is valid a.e. in (a,b). □

In [19], the authors extend the notion of *Generalized Jacobian* to any scheme S that is defined by functions in $C^1_{pw}(\mathbb{R}^m)$. For binary schemes

$$\begin{cases} (Sf)_{2n} & = \psi_0(f_{n-p}, \ldots, f_{n+p}) \\ (Sf)_{2n+1} & = \psi_1(f_{n-p}, \ldots, f_{n+p}) \end{cases} \tag{A.16}$$

such that $\psi_k \in C^1_{pw}(\mathbb{R}^{2p+1})$, $k = 0, 1$, a *Generalized Jacobian* of S at $f \in l_\infty(\mathbb{Z})$, $DS(f)$, is defined as the linear operator associated to the bi-infinite matrix whose rows have the following non-zero components

$$(DS(f))_{[2n+k,n-p:n+p]} = D\psi_k(f_{n-p}, \ldots, f_{n+p}), \quad k = 0, 1, \quad j \in \mathbb{Z}, \tag{A.17}$$

where $D\psi_k$ is a generalized gradient of the function ψ_k. We notice that $\forall\, f, g \in l_\infty(\mathbb{Z})$, $k = 0, 1$

$$|(DS(f)g)_{2n+k}| = |D\psi_k(f_{n-p}, \ldots, f_{n+p}) \cdot (g_{n-p}, \ldots, g_{n+p})| \leq ||D\psi_k(f_{n-p}, \ldots, f_{n+p})||_1 ||g||_\infty$$

hence

$$||DS(f)||_\infty = \sup_{g \neq 0} \frac{||DS(f)g||_\infty}{||g||_\infty} \leq \max\{C_1^{\psi_0}, C_1^{\psi_1}\}$$

with $C_1^{\psi_k}$ in (A.1). The following result generalizes Theorem 22.

Theorem 23 *Let S be a scheme of the form* (A.16)*, $\psi_k \in C^1_{pw}(\mathbb{R}^{2p+1})$, and let $\gamma : [a,b] \to l_\infty(\mathbb{Z})$ be a Lipschitz curve*[4]*. Then $\tilde{\gamma} = S \circ \gamma : [a,b] \to l_\infty(\mathbb{Z})$ is also a Lipschitz curve, and*

$$\tilde{\gamma}'(t) = DS(\gamma(t))\gamma'(t) \qquad \text{a.e. on } (a,b). \tag{A.18}$$

Proof In each coordinate we have

$$\tilde{\gamma}_{2j+k} = (S\gamma)_{2j+k} = \psi_k \circ \chi_{j-p,j+p} \circ \gamma,$$

with $\chi_{n,m} f = (f_n, \ldots, f_m)$, $n < m$. Since ψ_k is Lipschitz, we have that

$$\begin{aligned} \|\tilde{\gamma}_{2j+k}(s) - \tilde{\gamma}_{2j+k}(t)\|_\infty &\leq L_{\psi_k} \|(\chi_{j-p,j+p} \circ \gamma)(s) - (\chi_{j-p,j+p} \circ \gamma)(t)\|_\infty \\ &\leq L_{\psi_k} \|\gamma(s) - \gamma(t)\|_\infty \leq L_{\psi_k} L_\gamma |s - t|, \end{aligned}$$

[4] $\gamma = \{\gamma_i\}_{i \in \mathbb{Z}}$ with $\gamma_i : \mathbb{R} \to \mathbb{R}$ and $|\gamma_i(x) - \gamma_i(y)| \leq L_\gamma |x - y|$, $\forall x, y \in \mathbb{R}$, $\forall i \in \mathbb{Z}$.

where L_* denotes the Lipschitz constant of each function involved. Therefore, each component of $\tilde{\gamma}$ is a Lipschitz curve and

$$\|\tilde{\gamma}(s) - \tilde{\gamma}(t)\|_\infty = \sup_{i\in\mathbb{Z}} |\tilde{\gamma}_i(s) - \tilde{\gamma}_i(t)| \leq \max\{L_{\psi_0}, L_{\psi_1}\} L_\gamma |s-t|.$$

Since the countable union of sets of zero measure has zero measure, Eq. A.18 follows as in Theorem 22. □

Obviously, the last two theorems apply as long as the functions ψ_i admit uniformly bounded Generalized Gradients, $D\psi_i$, satisfying the chain rule (A.10).

The study of contractivity via Generalized Jacobians

Theorem 23 allows to study the contractivity properties of the powers of subdivision schemes that are defined by functions in $C^1_{pw}(\mathbb{R}^m)$ (or, in general, by functions that admit uniformly bounded Generalized Gradients satisfying the chain rule (A.10)) by the following argument, sketched in [19]:
Given $f, g \in l_\infty(\mathbb{Z})$, define recursively $\gamma^j : [0,1] \to l_\infty(\mathbb{Z})$ as follows

$$\gamma^0(t) := tf + (1-t)g, \qquad \gamma^j(t) := S \circ \gamma^{j-1}(t), \quad j > 0.$$

Notice that $(\gamma^0)'(t) = f - g$, hence γ^j is also Lipschitz and

$$\begin{aligned}(\gamma^j)'(t) &= DS(\gamma^{j-1}(t))(\gamma^{j-1}(t))' = \dots \\ &= DS(\gamma^{j-1}(t))DS(\gamma^{j-2})(t)\cdots DS(\gamma^0(t))(\gamma^0)'(t), \quad \text{a.e. in } (0,1).\end{aligned}$$

Hence, since $\gamma^j(1) = S^j f$, $\gamma^j(0) = S^j g$, $\forall\, j \geq 0$ we can write

$$S^j f - S^j g = \gamma^j(1) - \gamma^j(0) = \int_0^1 DS(\gamma^{j-1}(t))DS(\gamma^{j-2})(t)\cdots DS(\gamma^0(t))(f-g)dt, \quad \text{(A.19)}$$

$$||S^j f - S^j g||_\infty \leq \left(\int_0^1 ||\Pi_{k=0}^{j-1} DS(\gamma^k(t))||_\infty dt\right) ||f-g||_\infty. \quad \text{(A.20)}$$

From Eq. A.20, we easily deduce the following contractivity result, which has been used in [22] to prove the stability of a monotone nonlinear scheme.

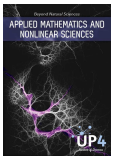
Corollary 24 *Let us assume that* $\exists L \geq 1$, $0 < \mu < 1$ *s. t.* $\forall f_0, \dots, f_L \in l_\infty(\mathbb{Z})$, $||\Pi_{k=0}^{L-1} DS(f_k)||_\infty \leq \mu$, *then* S^L *is contractive.*

References

1. Amat, S., Dadourian, K., Liandrat, J.: Analysis of a class of nonlinear subdivision schemes and associated multiresolution transforms. Adv. Comput. Math. **34**(3), 253–277 (2011)
2. Amat, S., Donat, R., Liandrat, J., Trillo, J.C.: Analysis of a new nonlinear subdivision scheme. Applications in image processing, Applications in Image Processing. Found Comput. Math., 193–225 (2006)
3. Dadourian, K., Liandrat, J.: Analysis of some bivariate non-linear interpolatory subdivision schemes. Numer. Algorithm. **48**, 261–278 (2008)

4. Amat, S., Donat, R., Liandrat, J., Trillo, J.C.: A fully adaptive multiresolution scheme for image processing. Math. Comput. Modell. **46**, 2–11 (2007)
5. Kuijt, F.: Convexity preserving interpolation. Stationary nonlinear subdivision and splines. PhD thesis, University of Twente, The Netherlands (1998)
6. Deslauriers, G., Dubuc, S.: Interpolation dyadique: fractals, dimension non entieres et application. Masson Paris, 44–45 (1987)
7. Clarke, F.H., Ledyae, Y.S.: Mean value inequalities. Proc. Am. Math. Soc. **122**, 1075–1083 (1994)
8. Kuijt, F., Van Damme, R.: Monotonicity preserving interpolatory subdivision scheme. J. Comput. Appl. Math. **101**, 203–229 (1999)
9. Aràndiga, F., Donat, R., Harten, A.: Multiresolution based on weighted averages of the hat function i: linear reconstruction techniques. SIAM J. Numer. Anal. **36**, 160–203 (1999)
10. Harten, A.: Multiresolution representation of data: a general framework. SIAM J. Numer. Anal. **33**, 1205–1256 (1996)
11. Aràndiga, F., Donat, R.: Nonlinear multi-scale decomposition: the approach of A. harten. Numer. Algorithm. **23**, 175–216 (2000)
12. Daubechies, I., Runborg, O., Sweldens, W.: Normal multiresolution approximation of curves. Constr. Approx. **20**, 399–463 (2003)
13. Burden, R.L., Douglas Faires, J.: Numerical analysis. Thomson Brooks/Cole (2005)
14. Aràndiga, F., Belda, A.M., Mulet, P.: Point-value WENO multiresolution applications to stable image compression. J. Sci. Comput. **43**, 158–182 (2010)
15. Marquina, A., Serna, S.: Power ENO methods: a fifth-order accurate weighted power ENO method. J. Comput. Phys. **194**(2), 632–658 (2004)
16. Cohen, A., Dyn, N., Matei, B.: Quasilinear subdivision schemes with applications to ENO interpolation. Appl. Comput. Harmon. Anal. **15**, 89–116 (2003)
17. Dadourian, K.: Schémas de Subdivision, Analyses Multirésolutions non-linéaires, Applications, PhD thesis, Université de Provence (2008)
18. Xie, G., Yu, T.P.-Y.: Smoothness analysis of nonlinear subdivision schemes of homogeneous and affine invariant type. Constr. Approx. **22**, 219–254 (2005)
19. Harizanov, S., Oswald, P.: Stability of nonlinear subdivision and multiscale transforms. Constr. Approx. **31**(3), 359–393 (2010)
20. Dyn, N.: Subdivision schemes in computer-aided geometric design, Advances in numerical analysis, Vol. II (Lancaster, 1990), pp. 36–104. Oxford Science of Publication, Oxford University Press, New York (1992)
21. Imbert, C.: Support functions of the Clarke generalized Jacobian and of its plenary hull. Nonlinear Anal. **49**, 1111–1125 (2002)
22. Aràndiga, F., Donat, R., Santágueda, M.: The PCHIP subdivision scheme. J. Comput. Appl. Math. **272**, 28–40 (2016)
23. Aràndiga, F., Donat, R., Santágueda, M.: Weighted-Power$_p$ nonlinear subdivision schemes. Curv. Surf. **6920**, 109–129 (2012)

Applied Mathematics and Nonlinear Sciences 2(2) (2017) 367–384



Applied Mathematics and Nonlinear Sciences
http://journals.up4sciences.org

High-accuracy approximation of piecewise smooth functions using the Truncation and Encode approach

Rosa Donat, Sergio López-Ureña [†].

Departament de Matemàtiques. Universitat de València. Doctor Moliner Street 50, 46100 Burjassot, Valencia Spain

Submission Info

Communicated by Juan L.G. Guirao
Received Day 7th April 2017
Accepted Day 6th September 2017
Available online 6th September 2017

Abstract

In the present work, we analyze a technique designed by Geraci et al. in [1, 11] named the Truncate and Encode (TE) strategy. It was presented as a non-intrusive method for steady and non-steady Partial Differential Equations (PDEs) in Uncertainty Quantification (UQ), and as a weakly intrusive method in the unsteady case.
We analyze the TE algorithm applied to the approximation of functions, and in particular its performance for piecewise smooth functions. We carry out some numerical experiments, comparing the performance of the algorithm when using different linear and non-linear interpolation techniques and provide some recommendations that we find useful in order to achieve a high performance of the algorithm.

Keywords: Truncate and Encode, Harten's Multiresolution Framework, Approximation, Uncertainty Quantification, Piecewise smooth functions
AMS 2010 codes: 41-XX, 65-XX.

1 Introduction

Uncertainty Quantification (UQ) is the science that studies the quantification and the reduction of uncertainties in real applications with an intensive computational component. An example would be the computation of the fuel consumption of a car. Suppose we know how to compute the consumption as a function of some parameters (car and wind velocity, wheel condition, car weight, etc.), but we do not know the exact value of these parameters because they are random variables. Hence, the fuel consumption is also a random variable. UQ studies how to infer the model of the fuel consumption random variable by assuming models for the parameters. To carry out such study, many simulations of the consumption, changing the parameters among simulations,

[†]Corresponding author.
Email address: sergio.lopez-urena@uv.es

ISSN 2444-8656 doi:10.21042/AMNS.2017.2.00030

maybe required. These computations tend to be very expensive, because they may involve numerically solving systems of partial differential equations.

One of the aims of UQ is to find the statistical moments of the random variable that provides the solution to a given problem. In [1, 11], the authors propose a new multi-scale technique, the *Truncation and Encode* approach (TE henceforth) to be applied to lower the cost of the computation (by quadrature rules) of the moments of the solution of the following stochastic PDE

$$\partial_t u(x,t,\xi) + \partial_x f(x,t,\xi,u(x,t,\xi)) = 0,$$

where $\xi \in \mathbb{R}^N$ is a random vector, with known probabilistic distribution. Notice that for each ξ, a PDE must be solved to compute $u_\xi(x,t) = u(x,t,\xi)$, namely

$$\partial_t u_\xi(x,t) + \partial_x f_\xi(x,t,u_\xi(x,t)) = 0.$$

Consequently, the calculation of these moments is a computationally intensive task, since function evaluations for given parameter values usually require a complex numerical simulation. Many numerical methods have been specially developed for this kind of UQ problems (for instance, methods generalizing or improving the classical Monte Carlo method, or those based on the Polynomial Chaos representation [14]) and all of them try to use as few function evaluations as possible.

The TE technique is specifically designed to meet a target accuracy in the computation of the desired integral, by ensuring a predetermined accuracy in the required function evaluations. It is specially suited for the integration of functions which are only piecewise smooth, since in this case other alternatives such as the Polynomial Chaos method may be highly inefficient.

In this paper we analyze a basic ingredient of the TE technique designed in [1, 11]: the influence of the approximation method for piecewise smooth functions in the global efficiency of the TE technique. For this, we consider a simplified framework in which we seek to compute integrals of the type

$$E = \int_0^1 f(\xi) d\xi$$

by means of a quadrature rule using point values on an equally spaced grid. For the sake of simplicity, let us assume that we use the trapezoidal rule and a uniform mesh of grid-size $h_K = 2^{-K}$, i.e.

$$E_K = 2^{-K}\left(\frac{1}{2}v_0^K + \sum_{i=0}^{2^K} v_i^K + \frac{1}{2}v_{2^K}^K\right), \qquad v_i^K = f(i2^{-K}).$$

By standard results, we know that when f is sufficiently smooth

$$|E - E_K| = O(2^{-2K})$$

while

$$|E - E_K| = O(2^{-K})$$

when f is only piecewise smooth.

The TE algorithm seeks to replace as many as possible v_i^K values, which require function evaluations of f, with *modified* $\hat{v}_i^K$ values, so that

$$\hat{E}_K = 2^{-K}\left(\frac{1}{2}\hat{v}_0^K + \sum_{i=0}^{2^K} \hat{v}_i^K + \frac{1}{2}\hat{v}_{2^K}^K\right)$$

satisfies

$$|E - \hat{E}_K| \le \varepsilon$$

where ε is a user-dependent predetermined accuracy, even when f is only piecewise smooth.

Notice that

$$|E_K - \hat{E}_K| = 2^{-K}\left|(\frac{1}{2}v_0^K + \sum_{i=1}^{2^K-1}\hat{v}_i^K + \frac{1}{2}v_{2^K}^K) - (\frac{1}{2}\hat{v}_0^K + \sum_{i=1}^{2^K-1}\hat{v}_i^K + \frac{1}{2}\hat{v}_{2^K}^K)\right| \leq \|v^K - \hat{v}^K\|_1 \leq \|v^K - \hat{v}^K\|_\infty,$$

where

$$\|v^K - \hat{v}^K\|_1 = 2^{-K}\sum_{i=0}^{2^K}|v_i^K - \hat{v}_i^K|, \qquad \|v^K - \hat{v}^K\|_\infty = \sup_{0\leq i\leq 2^K}|v_i^K - \hat{v}_i^K|.$$

Hence,

$$|E - \hat{E}_K| \leq |E - E_K| + ||\hat{v}^K - v^K||_1. \tag{1}$$

Thus, for a given (piecewise smooth) f, it is possible to ensure a target accuracy by choosing appropriately the mesh (i.e. K) and the target accuracy ε so that $||\hat{v}^K - v^K||_\infty \leq \varepsilon$.

The TE algorithm in [1, 11] is based on Harten's Multiresolution Framework (MRF), which provides a set of tools to manage data in a multi-scale setting. In the last two decades, the MRF has been successfully applied in various contexts, and in particular in the design of adaptive schemes for the numerical solution of conservation laws and systems [6–8, 13]. The TE algorithm follows a strategy similar to that used in the cost-effective schemes described in [7] and used in [6, 8, 13]. Both algorithms use the ideas in Harten's MRF to compute a sequence $\hat{v}^K$ such that $||\hat{v}^K - v^K|| \leq \varepsilon$ for a specified norm, with $v^K = \left(f(\xi_i^K)\right)_{i=0}^{2^K}$.

These strategies proceed from coarse to fine resolution levels as follows: At each resolution level $k \leq K$, (associated to a uniform grid with spacing 2^{-k}), a vector $\hat{v}^k$ is obtained, whose components either coincide with or approximate the values of f on the corresponding grid, i.e. $\hat{v}_i^k \approx f(i2^{-k})$. Approximations are computed in [1, 7, 11] via an interpolatory reconstruction. The TE authors also consider a multilevel thresholding strategy $(\varepsilon^k)_{k=0}^K$, which should be properly chosen to guarantee certain precision in the output $\hat{v}^K$:

$$\|v^K - \hat{v}^K\|_\infty \lessapprox \varepsilon.$$

In this work we show that the simple strategy $\varepsilon^k = \varepsilon$, also considered in [7], is more effective than the one proposed in [1, 11] for the current purpose, $\varepsilon^k = 2^{K-k}\varepsilon$. Another advantage of taking $\varepsilon^k = \varepsilon$ is that K can be taken as large as needed without increasing the number of function evaluations. Thus, $|E - E_K|$ can be made arbitrary small in (1), which implies $|E - \hat{E}_K| \lessapprox \varepsilon$.

In this work we also analyze the use of high accuracy interpolation techniques, and we establish the growth rate of the number of evaluations (n_{eval}) respect to the precision ($\|v^K - \hat{v}^K\|_\infty$):

$$n_{eval} = O(\|v^K - \hat{v}^K\|_\infty^{-1/s}) \quad \equiv \quad \|v^K - \hat{v}^K\|_\infty = O(n_{eval}^{-s}),$$

where s is the approximation order of the interpolation technique. This result is independent of the smoothness of f. That is, the number of function evaluations increases slowly even if f has some discontinuities. Hence the TE algorithm is very convenient in such situations.

A comparison between linear and non-linear interpolation techniques is also carried out. Our numerical results will show that non-linear techniques are dispensable in the TE strategy. Since linear ones are faster, computationally speaking, they are preferable. In particular, we make the comparison between piecewise polynomial Lagrange interpolation of degrees 1 and 3, and the nonlinear PCHIP interpolation developed in [5]. In [1, 11], the ENO interpolation was chosen instead of PCHIP, but PCHIP interpolation is monotone, non-oscillatory, *stable* as a subdivision process and faster than the ENO technique.

The paper is organized as follows: In section 2 we recall Harten's MRF and define the specific prediction schemes to be considered for our study. In section 3 we recall the Truncation and Encode strategy described in [1, 11] and in section 4 we perform a series of numerical experiments that confirm our observations. We close the paper with some conclusions and perspectives for future work.

2 Harten's Multiresolution Framework

2.1 Presentation

The TE technique is based on Harten's Multiresolution Framework (MRF), which relies on two basic elements: *discretization* and (compatible) *reconstruction* operators. In this paper we only consider the interpolatory framework for functions defined in the unit interval. The reader is referred to [3, 12] for a more complete description of Harten's MRF.

In the interpolatory framework, the *discretization operators* $(\mathscr{D}_k)_{k\geq 0}$:

$$\mathscr{D}_k : \mathscr{C}([0,1]) \longrightarrow \mathbb{R}^{n_k+1}, \qquad \mathscr{D}_k f = (f(\xi_i^k))_{i=0}^{n_k} \in \mathbb{R}^{n_k+1}$$

obtain the point values of a function f on a sequence of *nested grids* $(\xi^k)_{k\geq 0}$, i.e.

$$\xi^k = (\xi_i^k)_{i=0}^{n_k} \in \mathbb{R}^{n_k+1}, \quad \xi_0^k = 0, \xi_{n_k}^k = 1, \quad \xi_i^k < \xi_{i+1}^k, \quad \xi_i^k = \xi_{2i}^{k+1}, \quad n_{k+1} = 2n_k.$$

Together with the discretization operators, a sequence of *interpolatory reconstruction* operators $(\mathscr{R}_k)_{k\geq 0}$ is considered, which verify

$$\mathscr{R}_k : \mathbb{R}^{n_k+1} \longrightarrow \mathscr{C}([0,1]), \quad \mathscr{R}_k v^k \in \mathscr{C}([0,1]), \quad \mathscr{R}_k v^k(\xi_i^k) = v_i^k, \quad i = 0,1,\ldots,n_k,$$

with $\mathscr{C}([0,1]) = \{f : [0,1] \longrightarrow \mathbb{R},\ f \text{ continuous}\}$.

Since $\mathscr{R}_k$ is interpolatory, the *compatibility condition*

$$\mathscr{D}_k \mathscr{R}_k = \mathbb{I}_k \tag{2}$$

is fulfilled, where $\mathbb{I}_k : \mathbb{R}^{n_k+1} \longrightarrow \mathbb{R}^{n_k+1}$ is the identity operator.

Note 1. *An example of these concepts is the following. On the one hand, let be $\xi_i^k = i2^{-k}$, $n_k = 2^k$. Then*

$$\mathscr{D}_k f = (f(i2^{-k}))_{i=0}^{2^k} \in \mathbb{R}^{2^k+1}.$$

On the other hand, we consider the reconstruction operators

$$\mathscr{R}_k v^k \in \mathscr{C}([0,1]), \quad (\mathscr{R}_k v^k)(\xi) = \mathscr{I}_1(\xi; \xi^k, v^k),$$

where $\mathscr{I}_1$ is the 1st degree polynomial interpolation:

$$\mathscr{I}_1(\xi; \xi^k, v^k) = v_{i+1}^k \frac{\xi - \xi_i^k}{\xi_{i+1}^k - \xi_i^k} - v_i^k \frac{\xi - \xi_{i+1}^k}{\xi_{i+1}^k - \xi_i^k}, \qquad \xi \in [\xi_i^k, \xi_{i+1}^k].$$

Actually, $\mathscr{R}_k v^k$ is the unique polygonal satisfying $(\mathscr{R}_k v^k)(i2^{-k}) = v_i^k$, for all $0 \leq i \leq n_k$.

$\mathscr{D}_k f$ is the *representation* of f at the *resolution level* k, and will be denoted by v^k. In Lemma 3.1 of [12] it is proved that the operator that sends $v^{k+1} = \mathscr{D}_{k+1} f$ to $v^k = \mathscr{D}_k f$ is well-defined and linear. It is called the *decimation* operator and it is written D_{k+1}^k. Moreover, it can always be expressed as

$$D_{k+1}^k = \mathscr{D}_k \mathscr{R}_{k+1}, \tag{3}$$

although $\mathscr{R}_{k+1}$ could be non-linear. In particular, for the point-valued discretization $\mathscr{D}_k$, the decimation operator is

$$D_{k+1}^k v^{k+1} = (v_{2i}^{k+1})_{i=0}^{n_{k+1}/2}. \tag{4}$$

Inversely, the *prediction* operator P_k^{k+1} gives an approximation of $\mathscr{D}_{k+1}f$ from $\mathscr{D}_k f$:

$$P_k^{k+1} : \mathbb{R}^{n_k+1} \longrightarrow \mathbb{R}^{n_{k+1}+1}, \quad P_k^{k+1} := \mathscr{D}_{k+1}\mathscr{R}_k.$$

Notice the similarity with (3). The error of P_k^{k+1} is measured as

$$\delta^{k+1} := \mathscr{D}_{k+1}f - P_k^{k+1}\mathscr{D}_k f.$$

By (4), (3) and (2), in this order,

$$(\delta_{2i}^{k+1})_{i=0}^{n_{k+1}/2} = D_{k+1}^k \delta^{k+1} = \mathscr{D}_k\mathscr{R}_{k+1}\mathscr{D}_{k+1}f - \mathscr{D}_k\mathscr{R}_{k+1}\mathscr{D}_{k+1}\mathscr{R}_k\mathscr{D}_k f = \mathscr{D}_k f - \mathscr{D}_k\mathscr{R}_k\mathscr{D}_k f = \mathscr{D}_k f - \mathscr{D}_k f = 0.$$

So the even values of δ^{k+1} are 0. Hence, we will pay attention exclusively to the odd values of δ^{k+1}, which are named the *detail coefficients*:

$$d_i^k := \delta_{2i+1}^{k+1} = f(\xi_{2i+1}^{k+1}) - (P_k^{k+1}\mathscr{D}_k f)_{2i+1}, \qquad 0 \le i < n_{k+1}/2 = n_k.$$

For simplicity, let us denote $v^k := \mathscr{D}_k f$. Then

$$d_i^k = v_{2i+1}^{k+1} - (P_k^{k+1}v^k)_{2i+1}, \qquad 0 \le i < n_{k+1}/2 = n_k. \tag{5}$$

Let be $a \in \mathbb{R}^\alpha$ and $b \in \mathbb{R}^\beta$. Let us denote by $(a;b) \in \mathbb{R}^{\alpha+\beta}$ the concatenation of a and b. Since $v^k \in \mathbb{R}^{n_k+1} = \mathbb{R}^{n_{k+1}/2+1}$ and $d^k \in \mathbb{R}^{n_{k+1}/2}$, then $(v^k;d^k) \in \mathbb{R}^{n_{k+1}+1}$. In fact, there exists a bijection between $(v^k;d^k)$ and $v^{k+1} \in \mathbb{R}^{n_{k+1}+1}$:

$$v^{k+1} = P_k^{k+1}v^k + \delta^{k+1} \longleftrightarrow \begin{cases} v^k = D_{k+1}^k v^{k+1} \\ \delta^{k+1} = v^{k+1} - P_k^{k+1}v^k \end{cases}.$$

That is

$$\begin{cases} v_{2i}^{k+1} = v_i^k \\ v_{2i+1}^{k+1} = (P_k^{k+1}v^k)_{2i+1} + d_i^k \end{cases} \longleftrightarrow \begin{cases} v^k = D_{k+1}^k v^{k+1} \\ d_i^k = v_{2i+1}^{k+1} - (P_k^{k+1}v^k)_{2i+1} \end{cases}. \tag{6}$$

Abusing of notation, we denote by $(a_1;a_2;a_3;\dots)$ the concatenation of many vectors. Note that (6) can be applied recursively to obtain the following equivalences

$$v^k \leftrightarrow (v^{k-1};d^{k-1}) \leftrightarrow (v^{k-2};d^{k-2};d^{k-1}) \leftrightarrow \cdots \leftrightarrow (v^0;d^0;d^1;\dots;d^{k-1}) \in \mathbb{R}^{n_k+1}.$$

The *multiresolution decomposition of k levels*, M^k, (or multiresolution transform) is defined as

$$M^k v^k = (v^0;d^0;d^1;\dots;d^{k-1}).$$

We denote its inverse, the *inverse multiresolution transform*, by

$$M^{-k}(v^0;d^0;d^1;\dots;d^{k-1}) = v^k.$$

2.2 Interpolatory prediction operators

A classical approach to design prediction operators consists in using piecewise polynomial interpolation techniques, $\mathscr{I}(\xi;\xi^k,v^k)$. See for instance Remark 1. Given some values v^k on a grid ξ^k, they provide a function of ξ satisfying

$$\mathscr{I}(\xi_i^k;\xi^k,v^k) = v_i^k.$$

The corresponding prediction operators are obtained by evaluating $\mathscr{I}(\cdot;\xi^k,v^k)$ on the finer grid ξ^{k+1}:

$$P_k^{k+1}v^k = (\mathscr{I}(\xi_i^{k+1};\xi^k,v^k))_{i=0}^{n_{k+1}}.$$

For equally-spaced grids, that is $\xi^k_{i+1} - \xi^k_i = h_k$, the reconstruction technique can usually be expressed in terms of a *local rule* I, as follows:

$$\mathscr{I}(\xi^{k+1}_{2i}; \xi^k, v^k) = v^k_i, \qquad \mathscr{I}(\xi^{k+1}_{2i+1}; \xi^k, v^k) = I(v^k_{i-l+1}, \dots, v^k_i, \dots, v^k_{i+r-1}),$$

for some $l, r > 0$. In such case,

$$(P^{k+1}_k v^k)_{2i} = v^k_i, \qquad (P^{k+1}_k v^k)_{2i+1} = I(v^k_{i-l+1}, \dots, v^k_i, \dots, v^k_{i+r-1}).$$

The *accuracy* of the interpolation technique is an important aspect to be taken into account. $\mathscr{I}$ has order of approximation s if

$$\|\mathscr{I}(\cdot; \xi^k, \mathscr{D}_k f) - f\|_\infty \le C h^s_k, \quad \forall k \ge 0,$$

for any f sufficiently smooth. It can be written in terms of the local rule as

$$I(f(\xi^k_{i-l+1}), \dots, f(\xi^k_i), \dots, f(\xi^k_{i+r-1})) = f(\xi^{k+1}_{2i+1}) + O(h^s_k), \quad \forall i, k. \tag{7}$$

Note 2. *Some examples of local rules are*

- *Polygonal rule or 1st degree polynomial rule:*

$$I(v^k_i, v^k_{i+1}) = \frac{1}{2} v^k_i + \frac{1}{2} v^k_{i+1}.$$

- *Cubic rule or 3rd degree polynomial rule:*

$$I(v^k_{i-1}, v^k_i, v^k_{i+1}, v^k_{i+2}) = -\frac{1}{16} v^k_{i-1} + \frac{9}{16} v^k_i + \frac{9}{16} v^k_{i+1} - \frac{1}{16} v^k_{i+2}.$$

- *PCHIP rule:*

$$I(v^k_{i-1}, v^k_i, v^k_{i+1}, v^k_{i+2}) = \frac{1}{2} v^k_i + \frac{1}{2} v^k_{i+1} - \frac{1}{8} \left(H(v^k_{i+1} - v^k_i, v^k_{i+2} - v^k_{i+1}) - H(v^k_i - v^k_{i-1}, v^k_{i+1} - v^k_i) \right),$$

  where $H(x,y) = \frac{2xy}{x+y}$ *if* x, y *have the same sign,* $H(x,y) = 0$ *otherwise.*

Notice that the polygonal and cubic rules are linear, while the PCHIP rule is nonlinear. It can be proven that the polygonal rule has approximation order 2, the cubic rule has order 4, and the PCHIP rule has order 4 if $(v^k_{i-1}, v^k_i, v^k_{i+1}, v^k_{i+2})$ *is strictly monotone but order 2 in general.*

Prediction operators in Harten's MRF define *subdivision schemes*. These are iterative process where denser and denser sets of data are generated using recursively the local rules. A subdivision scheme is convergent if this process converges to a continuous function. See for instance [4, 9, 10].

In [5] the subdivision scheme defined from the PCHIP rule was studied. Its authors found some advantages respect to other subdivision schemes, such as the ENO scheme [2]. They proved that PCHIP is monotonicity preserving, which means that it converges to a monotone function if monotone data is used. But also it is Lipschitz stable (see [5] for specific definitions) which makes it more suitable as prediction operators in Harten's MRF than the non-linear Essentially Non Oscillatory (ENO) reconstruction techniques used in [1, 11]. The monotonicity preserving property is particularly interesting when data coming from discontinuous functions needs to be handled, because it avoids oscillatory behavior in the reconstructed data around jumps and steep gradients (see [5]), just as the ENO reconstructions.

If f is sufficiently smooth, by (7)

$$d^k_i = v^{k+1}_{2i+1} - (P^{k+1}_k v^k)_{2i+1} = f(\xi^{k+1}_{2i+1}) - I(f(\xi^k_{i-l+1}), \dots, f(\xi^k_{i+r})) = O(h^s_k).$$

So

$$d_{2i}^{k+1} \approx (\frac{h_{k+1}}{h_k})^s d_i^k \approx d_{2i+1}^{k+1}.$$

We are assuming that $(\xi^k)_{k\geq 0}$ are nested and equal-spaced. Thus

$$O(h_k^s) = O(2^{-ks}), \qquad (\frac{h_{k+1}}{h_k})^s = 2^{-s}.$$

If f has a discontinuity in its s_0 derivative, $0 \leq s_0 \leq s$, then

$$d_i^k = O(2^{-ks_0}), \qquad d_{2i}^{k+1} \approx 2^{-s_0} d_i^k \approx d_{2i+1}^{k+1}.$$

3 The Truncation and Encode approach

In this section, we describe the TE method restricted to the approximation of functions.

Given a target error $\varepsilon > 0$, a piecewise smooth function $f : [0,1] \longrightarrow \mathbb{R}$ and a suitable finest resolution level K, the goal is to obtain some $\hat{v}^K \in \mathbb{R}^{n_K+1}$ such that

$$\|v^K - \hat{v}^K\|_\infty \lessapprox \varepsilon, \qquad v^K = \mathscr{D}_K f, \tag{8}$$

using as few evaluations of f as we can.

The algorithm computes recursively $\hat{v}^k$, from the coarsest level $k = 0$ to the finest one $k = K$. The construction of $(\hat{v}^k)_{k=0}^K$ begins by evaluating f at ξ^0 and ξ^1: $\hat{v}^0 = \mathscr{D}_0 f$ and $\hat{v}^1 = \mathscr{D}_1 f$. The details are computed using (5):

$$\hat{d}_j^0 = \hat{v}_{2j+1}^1 - (P_0^1 \hat{v}^0)_{2j+1} = d_j^0 = v_{2j+1}^1 - (P_0^1 v^0)_{2j+1}.$$

We compute $\hat{v}^k$, $k > 1$, iteratively: For each $0 \leq i \leq n_k$, and for each $1 \leq k \leq K$ (K pre-fixed), we set $\hat{v}_{2i}^{k+1} = \hat{v}_i^k$ and

$$\hat{v}_{2i+1}^{k+1} = \begin{cases} (P_k^{k+1}\hat{v}^k)_{2i+1}, & \text{if } |\hat{d}_j^{k-1}| < \varepsilon^k \\ f(\xi_{2i+1}^{k+1}), & \text{if } |\hat{d}_j^{k-1}| \geq \varepsilon^k \end{cases}, \qquad \forall i \in \{2j, 2j+1\}, \tag{9}$$

where

$$\hat{d}_j^{k-1} = \hat{v}_{2j+1}^k - (P_{k-1}^k \hat{v}^{k-1})_{2j+1}. \tag{10}$$

Figure 1 shows a sketch of recurrence (9). It has a simple interpretation: On one hand, if $|\hat{d}_j^{k-1}| < \varepsilon^k$, then a 'small' prediction error indicates that P_{k-1}^k approximates well the function at ξ_{2j+1}^k; thus P_k^{k+1} may also provide a good approximation at ξ_{2i+1}^{k+1}, $i \in \{2j, 2j+1\}$, and we do not need to evaluate f at these points.

On the other hand, $|\hat{d}_j^{k-1}| \geq \varepsilon^k$ means that P_{k-1}^k did not provide a sufficiently 'good' approximation to the function values, hence we prefer to evaluate f to obtain $\hat{v}_{2i+1}^{k+1}$ as the exact value of v_{2i+1}^{k+1}. It should be noted that if $|\hat{d}_j^{k-1}| < \varepsilon^k$ then $\hat{d}_i^k = 0$, and if $|\hat{d}_j^{k-1}| \geq \varepsilon^k$ then $\hat{d}_i^k = f(\xi_{2i+1}^{k+1}) - (P_k^{k+1}\hat{v}^k)_{2i+1}$, $i \in \{2j, 2j+1\}$. Also, $|\hat{d}_j^{k-1}| \geq \varepsilon^k$ implies that v_{4j+1}^{k+1} ($i = 2j$) and v_{4j+3}^{k+1} ($i = 2j+1$) must be computed, so f is evaluated twice.

In the end of the recurrence, we obtain the vectors $\hat{v}^K$ and $(\hat{d}^k)_{k=0}^{K-1}$. Moreover, we know the MR decomposition of $\hat{v}^k$ because of (10):

$$M^k \hat{v}^k = (\hat{v}^0; \hat{d}^0; \ldots; \hat{d}^{k-1}) = (v^0; d^0; \hat{d}^1; \ldots; \hat{d}^{k-1}).$$

In [1] the authors arrive to the conclusion that (8) if fulfilled if $\varepsilon^k = 2^{K-k}\varepsilon$. However, we notice in the numerical experiments of [1, 11] that this thresholding strategy leads to

$$\|v^K - \hat{v}^K\|_\infty \overset{K\to\infty}{\longrightarrow} 0.$$

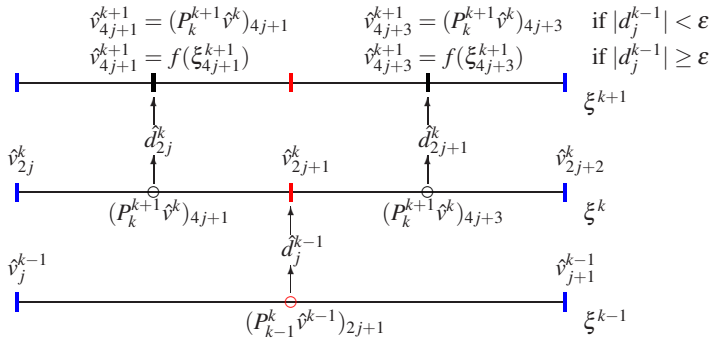


Fig. 1 Sketch of recurrence (9). The value of $\hat{v}^{k+1}_{2i+1}$, $i \in \{2j, 2j+1\}$, depends on $\hat{d}^{k-1}_j = \hat{v}^k_{2j+1} - (P^k_{k-1}\hat{v}^{k-1})_{2j+1}$.

Thus, the real precision of $\hat{v}^K$ depends on ε and K. Moreover, if a large K is needed, because a very fine mesh is demanded, then it turns out that $\|v^K - \hat{v}^K\|_\infty$ is unnecessarily small and the technique looses efficiency. We will see that the thresholding criterion $\varepsilon^k = 2^{K-k}\varepsilon$ proposed in [1, 11] can be relaxed to $\varepsilon^k = \varepsilon$, to ensure (8) with a precision that stays close to ε. Our strategy reduces the number of evaluations. In addition, the non-dependency of K allows us to take this value as large as necessary to ensure the desired accuracy in the numerical integration of f.

Another advantage of using $\varepsilon^k = \varepsilon$ is that it is not mandatory to prefix the finest level K. So we can select another criterion to stop the recurrence, for example, that certain amount of evaluations has been reached. On the opposite, taking $\varepsilon = 2^{K-k}\varepsilon$ demands knowing K before the execution starts.

4 Numerical experiments

The purpose of the present section is to examine the efficiency of the TE algorithm under the various strategies considered in the previous sections. In particular, we show that the threshold strategy $\varepsilon^k = \varepsilon$, $k \geq 0$, leads to the desired target accuracy in a much more efficient manner than $\varepsilon^k = 2^{K-k}\varepsilon$, as proposed in [1, 11].

In this paper, the efficiency is a measure of the number of evaluations required to reach a target error in the computation of $\hat{v}^K$. We show in this section that the efficiency of the TE strategy relies on the interpolation technique used in the prediction operator. In addition, we show numerically that the strategy $\varepsilon^k = \varepsilon$ ensures

$$2^{-s}\varepsilon \lessapprox \|v^k - \hat{v}^k\|_\infty \lessapprox \varepsilon,$$

where s is the order of approximation of the local rule, but also we show that the number of evaluations increases slowly respect to the precision:

$$n_{eval} = O(\|v^k - \hat{v}^k\|_\infty^{-1/s}) = O(\varepsilon^{-1/s}).$$

Hence, highly accurate interpolation techniques improves the efficiency.

On the other hand, we will see that non-linear interpolation techniques does not provide, in general, any advantage respect to the linear ones. Thus, linear prediction operators are preferable, because they are faster to compute.

The TE authors consider the following efficiency measures. The first one is

$$\mu_{cr} = \frac{n_K + 1}{n_w + n_0 + 1}, \qquad n_w = \#\{|\hat{d}^k_i| \geq \varepsilon\},$$

n_{eval}: Number of function evaluations done by TE to compute $\hat{v}^K$.
$n_K+1=2^K+1$: Number of function evaluations needed to compute v^K.
$\tau=\frac{n_K+1}{n_{eval}}$.
$e_K=\|v^K-\hat{v}^K\|_\infty=\sup_i\|v_i^K-\hat{v}_i^K\|$.
$\|v^K-\hat{v}^K\|_1=n_K^{-1}\sum_{i=0}^{n_K}\|v_i^K-\hat{v}_i^K\|$.
$E=\int_0^1 f(x)dx$.
$\hat{E}_K$ is obtained applying the trapezoidal rule to $\hat{v}^K$.

Table 1 Parameters for the numerical experiments.

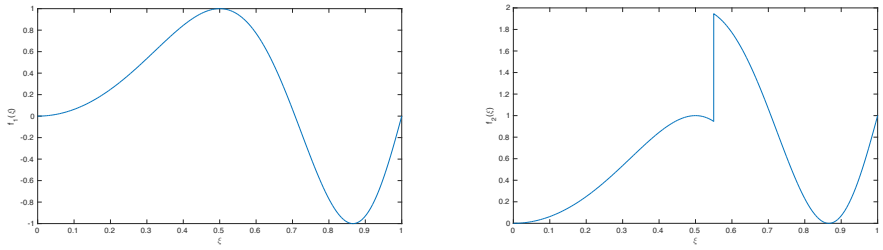


Fig. 2 Graphic representation of: left, f_1; right, f_2. Both functions are defined in (12).

where n_w is the number of detail coefficients greater than ε. The second one is

$$\tau=\frac{n_K+1}{n_{eval}},$$

where n_{eval} is the total amount of function evaluations used to obtain $\hat{v}^K$. $\tau=1$ if the function was evaluated everywhere in ξ^K, and the larger τ is, the more v_i^k were interpolated instead of evaluated.

In the TE method, each $|\hat{d}_i^{k-1}|\geq\varepsilon$ implies two new function evaluations at the resolution level $k+1$. So n_{eval} can be written as

$$n_{eval}=2n_w+n_0. \tag{11}$$

Using (11), τ can be expressed as

$$\tau=\frac{n_K+1}{2n_w+n_0}\quad\Longrightarrow\quad\mu_{cr}\approx 2\tau,$$

as shown in Tables 1 and 2 of [1, 11]. In this paper, we will only consider τ.

In Tables 2 to 7, we carry out the same experiment as in section 5.1 of [11]. Let us consider the parameters of Table 1 and the following functions, which are shown in Figure 2:

$$f_1(x)=\sin(2\pi x^2),\qquad f_2(x)=\begin{cases}\sin(2\pi x^2), & x\leq 11/20\\ \sin(2\pi x^2)+1, & x>11/20\end{cases},\qquad x\in[0,1]. \tag{12}$$

In Tables 2 to 4 we apply the TE strategy with $\varepsilon^k=10^{-1}$, $k\geq 0$, to the smooth f_1 function in Figure 2 and display the parameters of Table 1. We use the prediction operators, P_k^{k+1} described in Remark 2, i.e. the polygonal, cubic and PCHIP rules, while in [11] it was used the polygonal, cubic and ENO rules. As mentioned previously, we consider that the PCHIP prediction is more convenient because of its stability. As in the original papers, we set $n_0=1$, so $\xi^0=\{0,1\}$. From these experiments we extract the following conclusions.

K	n_{eval}	n_K+1	τ	$\|v^K-\hat{v}^K\|_\infty$	$\|v^K-\hat{v}^K\|_1$	$\|E-\hat{E}_K\|$
3	1.3000e+01	1.7000e+01	1.3077e+00	4.6488e-02	5.5437e-03	4.2125e-03
4	1.7000e+01	3.3000e+01	1.9412e+00	4.6488e-02	9.2544e-03	1.0395e-03
5	1.7000e+01	6.5000e+01	3.8235e+00	4.6488e-02	1.1582e-02	1.0395e-03
6	1.7000e+01	1.2900e+02	7.5882e+00	4.7016e-02	1.2225e-02	1.0395e-03
7	1.7000e+01	2.5700e+02	1.5118e+01	4.7016e-02	1.2411e-02	1.0395e-03
8	1.7000e+01	5.1300e+02	3.0176e+01	4.7026e-02	1.2470e-02	1.0395e-03
9	1.7000e+01	1.0250e+03	6.0294e+01	4.7033e-02	1.2491e-02	1.0395e-03
10	1.7000e+01	2.0490e+03	1.2053e+02	4.7033e-02	1.2499e-02	1.0395e-03
11	1.7000e+01	4.0970e+03	2.4100e+02	4.7034e-02	1.2503e-02	1.0395e-03

Table 2 Parameters as specified in Table 1 for f_1, $\varepsilon=\varepsilon^k=10^{-1}$ and the polygonal rule.

K	n_{eval}	n_K+1	τ	$\|v^K-\hat{v}^K\|_\infty$	$\|v^K-\hat{v}^K\|_1$	$\|E-\hat{E}_K\|$
3	1.3000e+01	1.7000e+01	1.3077e+00	5.6435e-03	7.8277e-04	4.4742e-03
4	1.3000e+01	3.3000e+01	2.5385e+00	1.1245e-02	1.7642e-03	1.1215e-03
5	1.3000e+01	6.5000e+01	5.0000e+00	1.1245e-02	1.9634e-03	2.7580e-04
6	1.3000e+01	1.2900e+02	9.9231e+00	1.2010e-02	2.0193e-03	6.3898e-05
7	1.3000e+01	2.5700e+02	1.9769e+01	1.2010e-02	2.0373e-03	1.0892e-05
8	1.3000e+01	5.1300e+02	3.9462e+01	1.2010e-02	2.0439e-03	2.3614e-06
9	1.3000e+01	1.0250e+03	7.8846e+01	1.2010e-02	2.0465e-03	5.6749e-06
10	1.3000e+01	2.0490e+03	1.5762e+02	1.2012e-02	2.0477e-03	6.5032e-06
11	1.3000e+01	4.0970e+03	3.1515e+02	1.2012e-02	2.0482e-03	6.7103e-06

Table 3 Parameters as specified in Table 1 for f_1, $\varepsilon=\varepsilon^k=10^{-1}$ and the cubic rule.

As observed in Table 2, the polygonal rule uses 13 evaluations for $K=3$, but 17 for $K\geq 4$. The cubic and PCHIP rules only uses 13 evaluation for any $K\geq 3$, as shown in Tables 3 and 4. From this fact, we deduce that there is a level k_0 such that for all $k\geq k_0$ the TE algorithm does not evaluate f_1 anymore. Actually, this property may hold for a general smooth function and a given threshold $\varepsilon>0$.

Notice also that $e_K=\|v^K-\hat{v}^K\|_\infty$ is always smaller than ε, but it stays close to ε, a clear advantage of using $\varepsilon^k=\varepsilon$. Indeed, it can be observed in Tables 1 and 2 of [1, 11] that e_K tends to zero when K increases, when the strategy $\varepsilon^k=2^{K-k}\varepsilon$ was selected. Since our target accuracy is $e_K\lessapprox\varepsilon$ with as few evaluations as possible, if the TE strategy uses just the necessary evaluations, then e_K should be close to ε. Thus, the fact that e_K tends to zero with the criterion $\varepsilon^k=2^{K-k}\varepsilon$ means that the function is being evaluated more than it is actually needed.

As exposed in Section 1, the next condition is fulfilled:

$$|E-\hat{E}_K|\leq|E-E_K|+\|v^K-\hat{v}^K\|_1\leq|E-E_K|+\|v^K-\hat{v}^K\|_\infty. \tag{13}$$

Since f_1 is smooth, then $|E-E_K|=O(2^{-2K})$. Thus, for large values of K, (13) becomes

$$|E-\hat{E}_K|\lessapprox\|v^K-\hat{v}^K\|_1\leq\|v^K-\hat{v}^K\|_\infty\lessapprox\varepsilon, \tag{14}$$

which can also be checked in Tables 2 to 4 for any $K>3$.

The smallest e_K is achieved with the cubic rule, and the largest with the polygonal rule. Between them is the PCHIP rule, with less e_K and less evaluations than the polygonal rule, but larger e_K with the same evaluations

K	n_{eval}	n_K+1	τ	$\|v^K-\hat{v}^K\|_\infty$	$\|v^K-\hat{v}^K\|_1$	$\|E-\hat{E}_K\|$
3	1.3000e+01	1.7000e+01	1.3077e+00	1.0606e-02	1.5196e-03	4.2125e-03
4	1.3000e+01	3.3000e+01	2.5385e+00	2.3126e-02	3.8381e-03	1.0814e-03
5	1.3000e+01	6.5000e+01	5.0000e+00	2.3126e-02	4.3212e-03	2.3874e-04
6	1.3000e+01	1.2900e+02	9.9231e+00	2.3126e-02	4.4675e-03	2.6119e-05
7	1.3000e+01	2.5700e+02	1.9769e+01	2.3126e-02	4.5131e-03	2.7146e-05
8	1.3000e+01	5.1300e+02	3.9462e+01	2.3144e-02	4.5297e-03	4.0470e-05
9	1.3000e+01	1.0250e+03	7.8846e+01	2.3177e-02	4.5360e-03	4.3801e-05
10	1.3000e+01	2.0490e+03	1.5762e+02	2.3177e-02	4.5387e-03	4.4634e-05
11	1.3000e+01	4.0970e+03	3.1515e+02	2.3177e-02	4.5399e-03	4.4842e-05

Table 4 Parameters as specified in Table 1 for f_1, $\varepsilon=\varepsilon^k=10^{-1}$ and the PCHIP rule.

K	n_{eval}	n_K+1	τ	$\|v^K-\hat{v}^K\|_\infty$	$\|v^K-\hat{v}^K\|_1$	$\|E-\hat{E}_K\|$
3	1.3000e+01	1.7000e+01	1.3077e+00	4.6488e-02	5.5437e-03	2.2962e-02
4	1.9000e+01	3.3000e+01	1.7368e+00	4.6488e-02	7.8089e-03	3.5761e-03
5	2.1000e+01	6.5000e+01	3.0952e+00	4.6488e-02	9.9419e-03	4.0614e-03
6	2.3000e+01	1.2900e+02	5.6087e+00	4.7016e-02	1.0549e-02	1.3192e-04
7	2.5000e+01	2.5700e+02	1.0280e+01	4.7016e-02	1.0726e-02	1.8242e-03
8	2.7000e+01	5.1300e+02	1.9000e+01	4.7026e-02	1.0781e-02	8.4798e-04
9	2.9000e+01	1.0250e+03	3.5345e+01	4.7033e-02	1.0800e-02	3.5975e-04
10	3.1000e+01	2.0490e+03	6.6097e+01	4.7033e-02	1.0808e-02	6.0389e-04
11	3.3000e+01	4.0970e+03	1.2415e+02	4.7034e-02	1.0811e-02	7.2597e-04

Table 5 Parameters as specified in Table 1 for f_2, $\varepsilon=\varepsilon^k=10^{-1}$ and the polygonal rule.

K	n_{eval}	n_K+1	τ	$\|v^K-\hat{v}^K\|_\infty$	$\|v^K-\hat{v}^K\|_1$	$\|E-\hat{E}_K\|$
3	1.3000e+01	1.7000e+01	1.3077e+00	6.1742e-02	4.3700e-03	1.9318e-02
4	1.5000e+01	3.3000e+01	2.2000e+00	9.7336e-02	7.6400e-03	1.6578e-03
5	1.7000e+01	6.5000e+01	3.8235e+00	9.7336e-02	9.0579e-03	9.4329e-03
6	1.9000e+01	1.2900e+02	6.7895e+00	9.7601e-02	9.8133e-03	5.2198e-03
7	2.1000e+01	2.5700e+02	1.2238e+01	9.8885e-02	1.0169e-02	3.5400e-03
8	2.3000e+01	5.1300e+02	2.2304e+01	9.8885e-02	1.0359e-02	4.6595e-03
9	2.5000e+01	1.0250e+03	4.1000e+01	9.9210e-02	1.0448e-02	5.0960e-03
10	2.7000e+01	2.0490e+03	7.5889e+01	9.9210e-02	1.0496e-02	4.8203e-03
11	2.9000e+01	4.0970e+03	1.4128e+02	9.9272e-02	1.0518e-02	4.7122e-03

Table 6 Parameters as specified in Table 1 for f_2, $\varepsilon=\varepsilon^k=10^{-1}$ and the cubic rule.

than the cubic rule. Hence the cubic rule here seems to be better than PCHIP rule for smooth functions, which is probably a consequence of the better approximation properties of the cubic rule at all smooth regions, while the PCHIP formula its as accurate as the cubic rule only at monotone regions.

In Tables 5 to 7 we repeat the same experiment for the function f_2, which has a discontinuity jump in $\xi=11/20$. Once again, we use $\varepsilon^k=\varepsilon=10^{-1}$, and we explore the effect of increasing K.

For $K>4$ in Table 5 and $K>3$ in Tables 6 and 7, the TE algorithm evaluates f_2 only two more times in each new level. This happens because there is only one detail coefficient satisfying $|\hat{d}_j^{k-1}|\geq\varepsilon$, which implies two new evaluations at level $k+1$. We refer to Figure 1 for the sake of clarity. This d_j^{k-1} is located around the discontinuity, which is easily seen in Fig. 5 of [11].

In spite of the discontinuity of f_2, $\|v^K-\hat{v}^K\|_\infty$ is smaller than ε, and also is near ε, which supports the criterion $\varepsilon^k=\varepsilon$. Now $|E-E_K|=O(2^{-K})$, thus (14) is satisfied for $K>5$.

Note that the polygonal rule uses more f_2 evaluations, but also its e_K are smaller than the other rules. e_K has similar magnitudes in the PCHIP and cubic rules and both use the same number of evaluations.

Once again, e_K converges to some value close to ε for $K\to\infty$, in contrast to observed in Tables 1 and 2 of [1, 11], where $e_K\to 0$, which confirms that the thresholding strategy $\varepsilon^k=\varepsilon$, instead of $\varepsilon^k=2^{K-k}\varepsilon$, is the most appropriate to attain a predetermined target accuracy in the computation.

In order to validate the error control property

$$2^{-s}\varepsilon\lessapprox\|v^K-\hat{v}^K\|_\infty\lessapprox\varepsilon, \tag{15}$$

K	n_{eval}	n_K+1	τ	$\|v^K-\hat{v}^K\|_\infty$	$\|v^K-\hat{v}^K\|_1$	$\|E-\hat{E}_K\|$
3	1.3000e+01	1.7000e+01	1.3077e+00	5.2400e-02	3.9780e-03	2.0350e-02
4	1.5000e+01	3.3000e+01	2.2000e+00	5.3922e-02	7.1545e-03	8.9677e-04
5	1.7000e+01	6.5000e+01	3.8235e+00	5.7556e-02	7.8826e-03	7.5834e-03
6	1.9000e+01	1.2900e+02	6.7895e+00	5.7616e-02	8.1229e-03	3.8474e-03
7	2.1000e+01	2.5700e+02	1.2238e+01	5.7920e-02	8.2001e-03	1.9526e-03
8	2.3000e+01	5.1300e+02	2.2304e+01	5.7920e-02	8.2293e-03	2.9445e-03
9	2.5000e+01	1.0250e+03	4.1000e+01	5.7920e-02	8.2408e-03	3.4357e-03
10	2.7000e+01	2.0490e+03	7.5889e+01	5.7922e-02	8.2458e-03	3.1923e-03
11	2.9000e+01	4.0970e+03	1.4128e+02	5.7922e-02	8.2481e-03	3.0704e-03

Table 7 Parameters as specified in Table 1 for f_2, $\varepsilon=\varepsilon^k=10^{-1}$ and the PCHIP rule.

ε	$2^{-2}\varepsilon$	n_{eval}	$\|v^K - \hat{v}^K\|_\infty$	$\|v^K - \hat{v}^K\|_1$	$\|E - \hat{E}_K\|$
1e-01	2.5e-02	1.7000e+01	4.7034e-02	1.2506e-02	1.0395e-03
1e-02	2.5e-03	5.7000e+01	2.4734e-03	9.4241e-04	1.3285e-04
1e-03	2.5e-04	1.7500e+02	2.6667e-04	8.3589e-05	2.2476e-05
1e-04	2.5e-05	4.9500e+02	4.1356e-05	1.1371e-05	2.2269e-06
1e-05	2.5e-06	1.7930e+03	6.6194e-06	8.9509e-07	1.7367e-07
1e-06	2.5e-07	5.4810e+03	2.5200e-07	8.5716e-08	2.1467e-08
1e-07	2.5e-08	1.5791e+04	8.1472e-08	1.1234e-08	2.0251e-09
1e-08	2.5e-09	5.6865e+04	2.5045e-09	8.1255e-10	1.5742e-10
1e-09	2.5e-10	1.7420e+05	7.8479e-10	4.0906e-11	2.0754e-11
1e-10	2.5e-11	2.5368e+05	3.2548e-11	4.7299e-13	1.5278e-11

Table 8 Parameters as specified in Table 1 for f_1, $K = 15$ and the polygonal rule. $\varepsilon^k = \varepsilon$ varies between 10^{-1} and 10^{-10}.

ε	$2^{-4}\varepsilon$	n_{eval}	$\|v^K - \hat{v}^K\|_\infty$	$\|v^K - \hat{v}^K\|_1$	$\|E - \hat{E}_K\|$
1e-01	6.25e-03	1.3000e+01	1.2012e-02	2.0488e-03	6.7794e-06
1e-02	6.25e-04	2.7000e+01	4.4450e-04	9.2550e-05	4.9354e-05
1e-03	6.25e-05	4.3000e+01	2.7188e-04	2.2672e-05	7.3722e-06
1e-04	6.25e-06	8.1000e+01	7.0349e-06	9.1132e-07	5.9591e-08
1e-05	6.25e-07	1.3700e+02	6.5275e-07	1.1135e-07	4.9200e-09
1e-06	6.25e-08	2.5500e+02	6.3729e-08	1.0246e-08	2.4199e-09
1e-07	6.25e-09	4.0900e+02	1.1524e-08	1.4623e-09	7.9961e-10
1e-08	6.25e-10	7.1900e+02	6.2685e-10	1.1760e-10	8.0592e-11
1e-09	6.25e-11	1.3610e+03	6.9934e-11	9.8732e-12	1.9488e-11
1e-10	6.25e-12	2.5150e+03	7.3016e-12	9.4366e-13	1.5464e-11

Table 9 Parameters as specified in Table 1 for f_1, $K = 15$ and the cubic rule. $\varepsilon^k = \varepsilon$ varies between 10^{-1} and 10^{-10}.

ε	$2^{-4}\varepsilon$	n_{eval}	$\|v^K - \hat{v}^K\|_\infty$	$\|v^K - \hat{v}^K\|_1$	$\|E - \hat{E}_K\|$
1e-01	6.25e-03	1.3000e+01	2.3177e-02	4.5411e-03	4.4912e-05
1e-02	6.25e-04	2.7000e+01	1.8042e-02	1.1650e-03	5.0535e-04
1e-03	6.25e-05	6.9000e+01	1.4751e-04	1.3376e-05	1.3180e-06
1e-04	6.25e-06	1.2300e+02	2.7379e-05	1.4956e-06	1.4601e-07
1e-05	6.25e-07	2.3300e+02	2.8833e-06	2.0876e-07	1.4208e-08
1e-06	6.25e-08	4.2100e+02	7.0168e-07	1.1231e-08	1.6390e-09
1e-07	6.25e-09	7.5700e+02	6.8925e-08	1.3645e-09	6.3232e-11
1e-08	6.25e-10	1.3610e+03	1.5733e-09	1.1174e-10	2.6616e-12
1e-09	6.25e-11	2.4230e+03	1.1084e-10	1.1353e-11	1.4272e-11
1e-10	6.25e-12	4.3150e+03	2.2858e-11	1.1252e-12	1.5153e-11

Table 10 Parameters as specified in Table 1 for f_1, $K = 15$ and the PCHIP rule. $\varepsilon^k = \varepsilon$ varies between 10^{-1} and 10^{-10}.

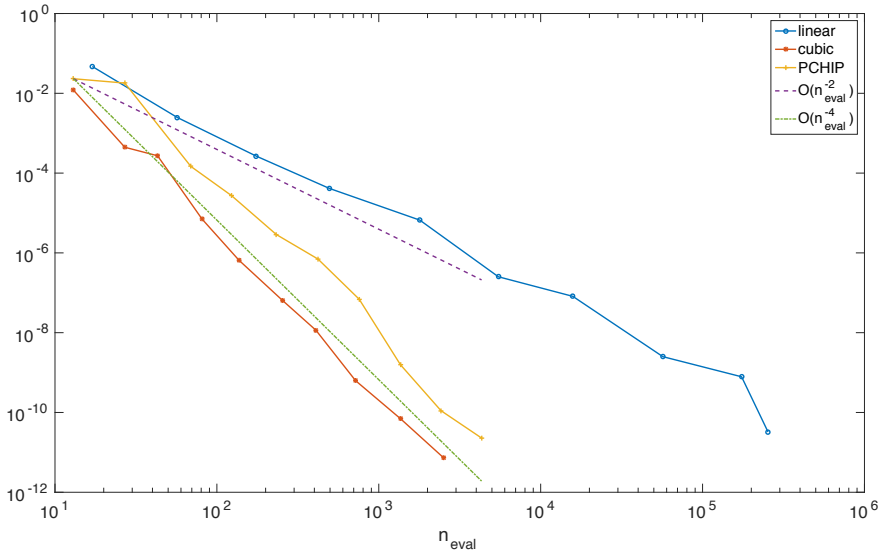


Fig. 3 In continuous line, the graphics of $\|v^{15} - \hat{v}^{15}\|_\infty$ as a function of n_{eval} for $\varepsilon^k = \varepsilon$ varying from 10^{-1} to 10^{-10}, applied to each local rule and f_1. For comparison, in discontinuous line are shown the growth rates $O(n_{eval}^{-2})$ and $O(n_{eval}^{-4})$.

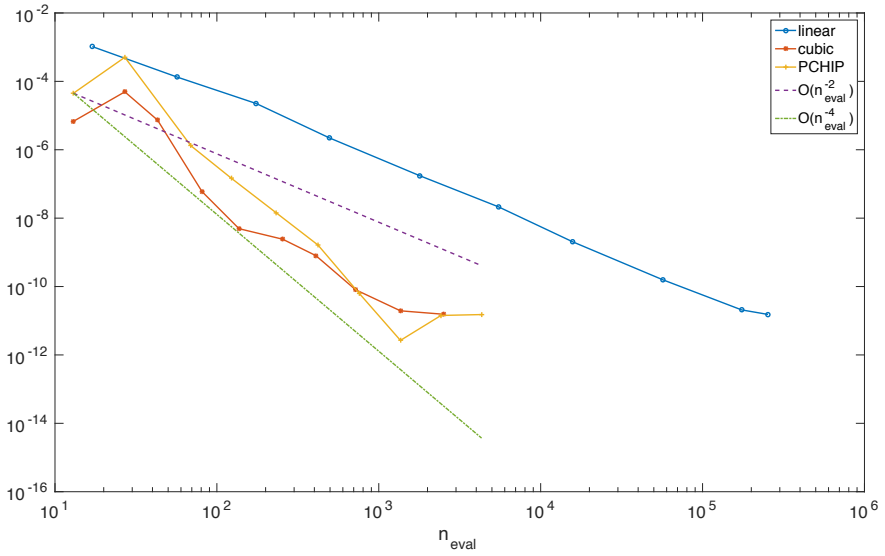


Fig. 4 In continuous line, the graphics of $|E - \hat{E}_{15}|$ as a function of n_{eval} for $\varepsilon^k = \varepsilon$ varying from 10^{-1} to 10^{-10}, applied to each local rule and f_1. For comparison, in discontinuous line are shown the growth rates $O(n_{eval}^{-2})$ and $O(n_{eval}^{-4})$.

ε	$2^{-2}\varepsilon$	n_{eval}	$\|v^K - \hat{v}^K\|_\infty$	$\|v^K - \hat{v}^K\|_1$	$\|E - \hat{E}_K\|$
1e-01	2.5e-02	4.5000e+01	4.7034e-02	1.0814e-02	6.5158e-04
1e-02	2.5e-03	8.1000e+01	2.4734e-03	9.1579e-04	1.5833e-04
1e-03	2.5e-04	1.9500e+02	2.6667e-04	8.3169e-05	2.1751e-05
1e-04	2.5e-05	5.1300e+02	4.1356e-05	1.1319e-05	1.1348e-06
1e-05	2.5e-06	1.8070e+03	6.6194e-06	8.9427e-07	9.6992e-07
1e-06	2.5e-07	5.4910e+03	2.5200e-07	8.5703e-08	1.1229e-06
1e-07	2.5e-08	1.5799e+04	8.1472e-08	1.1232e-08	1.1424e-06
1e-08	2.5e-09	5.6869e+04	2.5045e-09	8.1253e-10	1.1443e-06
1e-09	2.5e-10	1.7420e+05	7.8479e-10	4.0906e-11	1.1444e-06
1e-10	2.5e-11	2.5368e+05	3.2548e-11	4.7299e-13	1.1444e-06

Table 11 Parameters as specified in Table 1 for f_2, $K = 15$ and the polygonal rule. $\varepsilon^k = \varepsilon$ varies between 10^{-1} and 10^{-10}.

we repeat the same experiment for a fixed value of K, $K = 15$, and several ε values. We do not consider τ in Tables 8 to 13, because if K is fixed, then $\tau = O(n_{eval}^{-1})$. In the following, we study the approximation of the smooth function f_1, which is shown in Tables 8 to 10.

The first feature we observe is that (15) is satisfied, which provides a strong control on the error $e_K = \|v^K - \hat{v}^K\|_\infty$. We also notice that the growth rate of the number of evaluations, n_{eval}, is larger in the polygonal rule. We can easily see it plotting the results of the tables, shown in Figure 3 and 4. From such figures is deduced that the growth rate is $n_{eval} = O(e_K^{-1/2})$ for the polygonal rule and $n_{eval} = O(e_K^{-1/4})$ for the cubic and PCHIP rules. It can be equivalently formulated as $e_K = O((n_{eval})^{-2})$ and $e_K = O((n_{eval})^{-4})$. Remember that 2 and 4 are the highest orders of approximation of the considered local rules: PCHIP has only second order of approximation, in general, but forth order in monotone regions. Despite this, PCHIP verifies $e_K = O((n_{eval})^{-4})$ because, in the TE algorithm, the rule is applied in small monotone regions along the whole interval [0,1], where PCHIP has order 4. In addition, we clearly observe that cubic is a bit better than PCHIP, but much better than polygonal.

Another issue we can observe in Tables 8 to 10 is that $|E - \hat{E}_K|$ decrees together with ε, if $\varepsilon \geq 10^{-8}$, but for smaller ε, the precision of the integral gets stuck around 10^{-11}. It can be explained using the triangle inequality:

$$|E - \hat{E}_K| \geq |E - E_K| - |E_K - \hat{E}_K| \geq |E - E_K| - \|v^K - v^K\|_1 \geq |E - E_K| - \varepsilon.$$

Since f_1 is an smooth function, then

$$|E - E_{15}| = \frac{2^{-30}}{12}|f''(\eta)| \cong 7.8\text{e-}11|f''(\eta)|, \qquad \eta \in [0,1]. \tag{16}$$

Thus,

$$|E - \hat{E}_K| \geq 7.8\text{e-}11|f''(\eta)| - \varepsilon.$$

This says that, if $\varepsilon \to 0$, then $|E - \hat{E}_K|$ is lower bounded by $7.8\text{e-}11|f''(\eta)|$, which should be close to the number 1.5e-11, shown in Tables 8 to 10.

Comparing Figure 4 in this paper with Fig. 4 of [11], we note that our plot displays less oscillations. This may indicate that the dependency between e_K and the number of evaluations is stronger and smoother when the criterion $\varepsilon^k = \varepsilon$ is selected.

We repeat the experiment for the function f_2, which has a discontinuity jump in $\xi = 11/20$, in Tables 5 to 7.

Notice that (15) is satisfied and the growth of n_{eval} is faster in the polygonal rule. Observing the Figures 5 and 6, corresponding to these tables, we observe that the growth rates are $n_{eval} = O(e_K^{-1/2})$ for polygonal rule and $n_{eval} = O(e_K^{-1/4})$ for cubic and PCHIP rules. As shown in Figure 5, cubic rule is better than PCHIP rule to approximate v^K. However, PCHIP rule is now better than cubic rule to compute the integral. At least, it is true for $\varepsilon > 10^{-4}$, before $|E - \hat{E}_K|$ gets stuck.

A similar argument to (16) can be used to explain that $|E - \hat{E}_K| \to 1.1\text{e-}06$ in Tables 5 to 7. Since f_2 is not continuous, the error of the trapezoidal rule decreases as $O(2^{-K})$. Thus

$$|E - \hat{E}_K| \geq O(2^{-K}) - \varepsilon.$$

ε	$2^{-4}\varepsilon$	n_{eval}	$\|v^K - \hat{v}^K\|_\infty$	$\|v^K - \hat{v}^K\|_1$	$\|E - \hat{E}_K\|$
1e-01	6.25e-03	4.1000e+01	9.9324e-02	1.0541e-02	4.7904e-03
1e-02	6.25e-04	1.0700e+02	4.4450e-04	7.5032e-05	4.8158e-05
1e-03	6.25e-05	1.2100e+02	5.0725e-05	7.7497e-06	2.9716e-06
1e-04	6.25e-06	1.5300e+02	7.0349e-06	7.2132e-07	8.9567e-07
1e-05	6.25e-07	2.0300e+02	6.5275e-07	1.0494e-07	1.1429e-06
1e-06	6.25e-08	3.1500e+02	6.3729e-08	1.0056e-08	1.1418e-06
1e-07	6.25e-09	4.6300e+02	1.1524e-08	1.4565e-09	1.1436e-06
1e-08	6.25e-10	7.6700e+02	6.2685e-10	1.1741e-10	1.1443e-06
1e-09	6.25e-11	1.4070e+03	6.9934e-11	9.7495e-12	1.1444e-06
1e-10	6.25e-12	2.5570e+03	7.3016e-12	9.3789e-13	1.1444e-06

Table 12 Parameters as specified in Table 1 for f_2, $K = 15$ and the cubic rule. $\varepsilon^k = \varepsilon$ varies between 10^{-1} and 10^{-10}.

ε	$2^{-4}\varepsilon$	n_{eval}	$\|v^K - \hat{v}^K\|_\infty$	$\|v^K - \hat{v}^K\|_1$	$\|E - \hat{E}_K\|$
1e-01	6.25e-03	4.1000e+01	5.7922e-02	8.2502e-03	3.1449e-03
1e-02	6.25e-04	5.9000e+01	1.8042e-02	1.1149e-03	5.0165e-04
1e-03	6.25e-05	1.0700e+02	2.7833e-04	1.2775e-05	3.1659e-06
1e-04	6.25e-06	1.6900e+02	3.7144e-05	1.4538e-06	1.3326e-06
1e-05	6.25e-07	2.8500e+02	4.2689e-06	2.0742e-07	1.1600e-06
1e-06	6.25e-08	4.7500e+02	7.0168e-07	1.1186e-08	1.1461e-06
1e-07	6.25e-09	8.0500e+02	6.8925e-08	1.3631e-09	1.1445e-06
1e-08	6.25e-10	1.4090e+03	1.5733e-09	1.1040e-10	1.1444e-06
1e-09	6.25e-11	2.4650e+03	1.1084e-10	1.1313e-11	1.1444e-06
1e-10	6.25e-12	4.3510e+03	2.2858e-11	1.1239e-12	1.1444e-06

Table 13 Parameters as specified in Table 1 for f_2, $K = 15$ and the PCHIP rule. $\varepsilon^k = \varepsilon$ varies between 10^{-1} and 10^{-10}.

Again, a comparison between our Figure 4 and Fig. 6 of [11] seems to indicate that the strategy $\varepsilon^k = \varepsilon$ makes the TE approach more robust than $\varepsilon^k = 2^{K-k}\varepsilon$.

We conclude that PCHIP only provides a better performance respect to the usual cubic interpolation when the integral of a discontinuous function is computed. However, the cubic rule was better in all the other cases: integration of a continuous functions and approximation of continuous or even discontinuous functions. The TE authors also arrived to the conclusion that no advantages were seen using the ENO interpolation for f_1, but ENO was better for f_2 in terms of accuracy and amount of evaluations.

The results included in the present paper indicate that non-linear interpolatory reconstructions may not lead to substantial advantages over linear ones within the TE framework. We have observed that, in general, the cubic interpolation lead to better results than the PCHIP interpolation, while both always fulfill $n_{eval} = O(e_K^{-1/4})$. On the other hand, the usual reason of using ENO or PCHIP interpolatory techniques is the capability of approximating jumps in the data without spurious oscillations. But in TE strategy this is not needed, because the algorithm itself has a mechanism to detect the problematic zones of the function. In addition, linear interpolations are always faster than nonlinear ones.

We wish to remark again that the choice $\varepsilon^k = \varepsilon$ confers a high control over the error. In contrast, the original selection $\varepsilon^k = 2^{K-k}\varepsilon$ done by the TE authors only guarantees that ε is an upper bound of the error. Observe that in [1, 11] all the numerical experiments were done with $\varepsilon = 10^{-1}$, but the error converges to zero as soon as $K \to \infty$. In our case, the error converges to some number under ε, but greater than $2^{-s}\varepsilon$, where s is the order of the interpolation technique. This is a practical advantage, because K determines the resolution of the approximation $\hat{v}^K$, and we can set it as large as we need, but the number of evaluations and the error depends exclusively on ε.

5 Conclusions

In this paper we have examined some features of the Truncation and Encode strategy described in [1, 11], in particular its ability to compute integrals by quadrature rules to a given target accuracy.

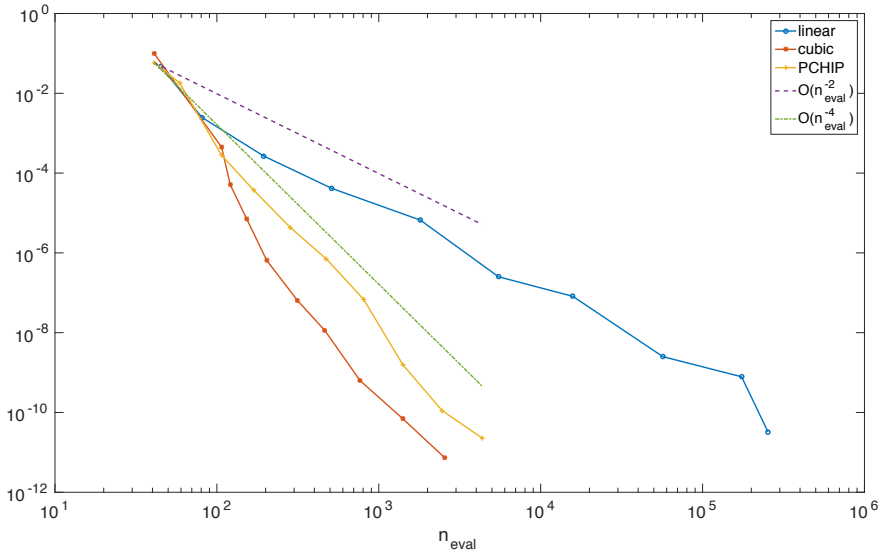


Fig. 5 In continuous line, the graphics of $\|v^{15} - \hat{v}^{15}\|_\infty$ as a function of n_{eval} for $\varepsilon^k = \varepsilon$ varying from 10^{-1} to 10^{-10}, applied to each local rule and f_2. For comparison, in discontinuous line are shown the growth rates $O(n_{eval}^{-2})$ and $O(n_{eval}^{-4})$.

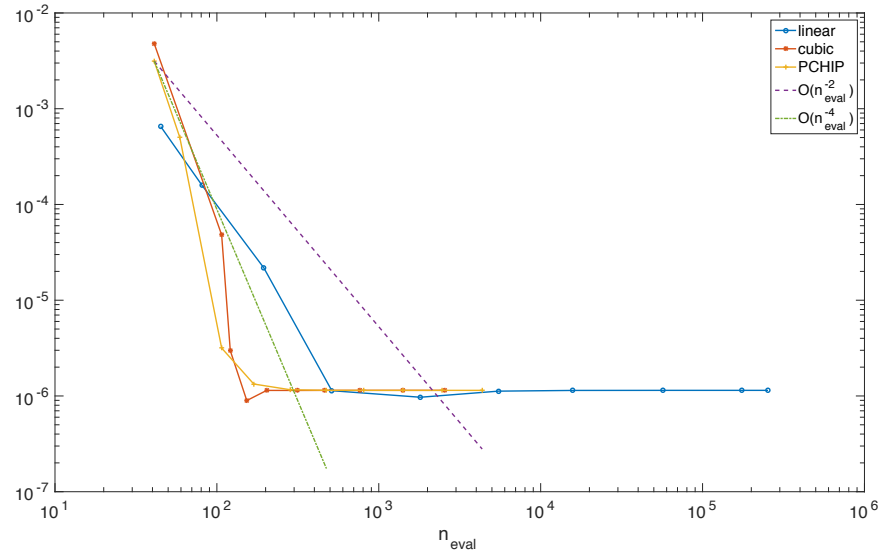


Fig. 6 In continuous line, the graphics of $|E - \hat{E}_{15}|$ as a function of n_{eval} for $\varepsilon^k = \varepsilon$ varying from 10^{-1} to 10^{-10}, applied to each local rule and f_2. For comparison, in discontinuous line are shown the growth rates $O(n_{eval}^{-2})$ and $O(n_{eval}^{-4})$.

In the TE strategy, the values used in the quadrature rule are computed by combining function evaluations and suitably interpolated values in a multilevel fashion, following the basic guidelines of Harten's MRF.

Our theoretical observations and numerical experiments allow us to conclude that the TE technique becomes more efficient if the thresholding strategy proposed in [1, 11] is relaxed to $\varepsilon^k = \varepsilon$ for all resolution levels. In addition, higher order prediction schemes lead to improved the performance, but it does not seem necessary to resort to nonlinear prediction schemes.

More research should be done to support these conclusions at a theoretical level.

Acknowledgements

The authors acknowledge support from Project MTM2014-54388 (MINECO, Spain) and the FPU14/02216 grant (MECD, Spain).

References

[1] R. Abgrall, P.M. Congedo, and G. Geraci. A one-time truncate and encode multiresolution stochastic framework. *Journal of Computational Physics*, 257, Part A:19 – 56, 2014. URL: http://www.sciencedirect.com/science/article/pii/S0021999113005342, doi:https://doi.org/10.1016/j.jcp.2013.08.006.

[2] F. Aràndiga, A. M. Belda, and P. Mulet. Point-value weno multiresolution applications to stable image compression. *Journal of Scientific Computing*, 43(2):158–182, 2010. URL: http://dx.doi.org/10.1007/s10915-010-9351-8, doi:10.1007/s10915-010-9351-8.

[3] Francesc Aràndiga and Rosa Donat. Nonlinear multiscale decompositions: The approach of a. harten. *Numerical Algorithms*, 23(2):175–216, 2000. URL: http://dx.doi.org/10.1023/A:1019104118012, doi:10.1023/A:1019104118012.

[4] Francesc Aràndiga, Rosa Donat, and Maria Santágueda. *Weighted-Power p Nonlinear Subdivision Schemes*, pages 109–129. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. URL: http://dx.doi.org/10.1007/978-3-642-27413-8_7, doi:10.1007/978-3-642-27413-8_7.

[5] F. Ar[ndiga, R. Donat, and M. Sant"gueda. The PCHIP subdivision scheme. *Applied Mathematics and Computation*, 272, Part 1:28 – 40, 2016. Subdivision, Geometric and Algebraic Methods, Isogeometric Analysis and Refinability. URL: http://www.sciencedirect.com/science/article/pii/S009630031500990X, doi:http://doi.org/10.1016/j.amc.2015.07.071.

[6] O. Boiron, G. Chiavassa, and R. Donat. A high-resolution penalization method for large mach number flows in the presence of obstacles. *Computers & Fluids*, 38(3):703 – 714, 2009. URL: http://www.sciencedirect.com/science/article/pii/S0045793008001424, doi:http://dx.doi.org/10.1016/j.compfluid.2008.07.003.

[7] Guillaume Chiavassa and Rosa Donat. Point value multiscale algorithms for 2d compressible flows. *SIAM Journal on Scientific Computing*, 23(3):805–823, 2001. URL: http://dx.doi.org/10.1137/S1064827599363988, arXiv:http://dx.doi.org/10.1137/S1064827599363988, doi:10.1137/S1064827599363988.

[8] Chiavassa, Guillaume, Donat, Rosa, and Martinez-Gavara, Anna. Cost-effective multiresolution schemes for shock computations. *ESAIM: Proc.*, 29:8–27, 2009. URL: https://doi.org/10.1051/proc/2009052, doi:10.1051/proc/2009052.

[9] Rosa Donat, Sergio López-Ureña, and Maria Santágueda. A family of non-oscillatory 6-point interpolatory subdivision schemes. *Advances in Computational Mathematics*, pages 1–35, 2017. URL: http://dx.doi.org/10.1007/s10444-016-9509-5, doi:10.1007/s10444-016-9509-5.

[10] Nira Dyn. Subdivision schemes in cagd. In *Advances in Numerical Analysis*, pages 36–104. Univ. Press, 1992.

[11] Gianluca Geraci, Pietro Marco Congedo, Rémi Abgrall, and Gianluca Iaccarino. A novel weakly-intrusive nonlinear multiresolution framework for uncertainty quantification in hyperbolic partial differential equations. *Journal of Scientific Computing*, 66(1):358–405, 2016. URL: http://dx.doi.org/10.1007/s10915-015-0026-3, doi:10.1007/s10915-015-0026-3.

[12] Ami Harten. Multiresolution representation of data: A general framework. *SIAM Journal on Numerical Analysis*, 33(3):1205–1256, 1996. URL: http://dx.doi.org/10.1137/0733060, arXiv:http://dx.doi.org/10.1137/0733060, doi:10.1137/0733060.

[13] Audrey Rault, Guillaume Chiavassa, and Rosa Donat. Shock-vortex interactions at high mach numbers. *Journal of Scientific Computing*, 19(1):347–371, 2003. URL: http://dx.doi.org/10.1023/A:1025316311633,

doi:10.1023/A:1025316311633.
[14] Norbert Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938. URL: http://www.jstor.org/stable/2371268.

A Novel Multi-Scale Strategy for Multi-Parametric Optimization

Rosa Donat, Sergio López-Ureña, and Marc Menec

Abstract The motion of a sailing yacht is the result of an equilibrium between the aerodynamic forces, generated by the sails, and the hydrodynamic forces, generated by the hull(s) and the appendages (such as the keels, the rudders, the foils, etc.), which may be fixed or movable and not only compensate the aero-forces, but are also used to drive the boat. In most of the design, the 3D shape of an appendage is the combination of a plan form (2D side shape) and a planar section(s) perpendicular to it, whose design depends on the function of the appendage. We often need a section which generates a certain quantity of lift to fulfill its function, but the lift comes with a penalty which is the drag. The efficiency, equilibrium and speed of a sailing boat depend on the appendage hence on the planar section. We describe a multi-scale strategy to optimize the shape of a section in a multi-parametric setting by embedding the problem within a discrete multiresolution framework. We show that our strategy can be easily implemented and, combined with appropriate optimization techniques, provides a fast algorithm to obtain an 'optimal' perturbation of the original shape.

1 Introduction

Appendages of a sailing yacht such as keels and rudders are designed from planar sections [1, 4]. The shape of these sections determines the drag and lift generated by the appendage and, hence, the boat's efficiency and performance. The search for an 'optimal' shape, which minimizes the drag generated by the section, is thus an important problem in yacht design.

Sections are closed planar curves, which may be parametrized as $\alpha(t) = (x(t), y(t))$, $t \in [0, 1]$. We shall assume that the first coordinate, x, runs along the

R. Donat • S. López-Ureña (✉)
Universitat de València, 50th Doctor Moliner street, 46100 València, Burjassot, Spain
e-mail: donat@uv.es; sergio.lopez-urena@uv.es

M. Menec
IS3DE ENG., 5th Mare Nostrum avenue, València, Spain
e-mail: marc.menec@is3de.com

P. Quintela et al. (eds.), *Progress in Industrial Mathematics at ECMI 2016*,
Mathematics in Industry 26, DOI 10.1007/978-3-319-63082-3_91

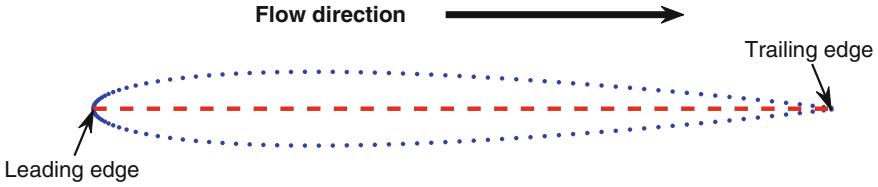


Fig. 1 An example of section: NACA0010 with $N = 129$ points

fluid direction and the second coordinate, y, is perpendicular to the direction of x. We also assume that $\alpha(0) = \alpha(1)$ corresponds to the trailing edge (Fig. 1).

Our goal is to perturb the shape of a given section so that the associated *drag coefficient*, computed by the CFD code `xfoil`,[1] is 'optimally' low. Codes like `xfoil` work in a discrete framework where the closed profile α is simply given by a finite number of points in the plane $\alpha = (\alpha_i)$, $\alpha_i = (x_i, y_i)$, $1 \le i \le N$. If the shape of the section is given by the parametrization $\alpha(t)$, $t \in [0, 1]$, then

$$\alpha_i = \alpha(t_i), \quad t_i < t_{i+1}, \quad t_i \in [0, 1]. \tag{1}$$

Hence, we may assume that we have at our disposal an underlying mesh on $[0, 1]$, given by a finite number of nodes t_i, $1 \le i \le N$, such that α_i is associated t_i.

We seek an 'optimal shape' by considering perturbations of the original shape $\alpha = (\alpha_i)$ of the form $\alpha^\varepsilon := (\alpha_i + \varepsilon_i)_i$, $\varepsilon_i \in \mathbb{R}^2$. Our goal is to find

$$\varepsilon_* \in \mathbb{R}^{2\times N} : \mathscr{D}(\alpha^{\varepsilon_*}) = \min_{\varepsilon\in\mathbb{R}^{2\times N}} \mathscr{D}(\alpha^\varepsilon), \tag{2}$$

where $\mathscr{D}(\beta)$ is the drag coefficient associated to a section $\beta = (\beta_i)$. The structural limitations of the appendage have to be taken into account in the optimization process. For example, for a keel section we would like to preserve the inertia[2] and the chord length of the initial shape. Hence (2) is, in general, a multi-parametric constrained optimization problem.

For a 'typical' section shape ($N = 129$) finding the solution of (2) may be (very) slow due to the large number of free parameters. The purpose of this paper is to describe a *multi-scale optimization* (MSO) strategy, which reduces the computational cost of solving the minimization problem (2).

The paper is organized as follows: In Sect. 2 we briefly recall the main ingredients of Harten's Multiresolution Framework (H-MRF), and in particular its connection with subdivision schemes, which is the main tool used in our MSO strategy. Section 3 provides the description of our technique and in Sect. 4 we carry out several examples of application of our strategy. We demonstrate the efficiency of

[1] XFOIL is an interactive program for the design and analysis of subsonic isolated airfoils.

[2] Inertia is the resistance of any physical object to any change in its state of motion.

our MSO strategy in an academic example. An example involving a more realistic situation is also shown.

2 Harten's Framework for Multiresolution

A multiresolution (MR) decomposition of a discrete data set α^L is an equivalent representation that encodes the discrete information contained in α^L as a *coarse* representation of this data set, α^0, together with a sequence of details at each resolution level (d^j)

$$\alpha^L \equiv M\alpha^L := (\alpha^0, d^0, d^1, \ldots, d^{L-1}), \tag{3}$$

where L represents the finest resolution level and 0 the coarsest one. In H-MRF the levels of resolution are induced from a hierarchy of *nested*[3] meshes $(\mathcal{G}^j)_{j=0}^L$ on an underlying spatial domain. In H-MRF, the representation in (3) is obtained after a repeated application of the operators which connect two consecutive resolution levels.

- *Decimation* (from fine to coarse): $\alpha^{j-1} = D_j^{j-1}\alpha^j$. D_j^{j-1} are linear operators that define a coarser representation α^{j-1} from a finer representation α^j.
- *Prediction* (from coarse to fine): $\tilde{\alpha}^j = P_{j-1}^j\alpha^{j-1}$. P_{j-1}^j are (possibly nonlinear) operators that define new data at a finer level, from discrete data at a coarser resolution level.

The multiresolution transform, and its inverse, are obtained after a recursive application of the 2-level transform

$$\begin{cases} \alpha^{j-1} & = D_j^{j-1}\alpha^j \\ e^j & = (I - P_{j-1}^j D_j^{j-1})\alpha^j \end{cases} \quad \leftrightarrow \quad \begin{cases} \alpha^j & = P_j^{j-1}\alpha^{j-1} + e^j. \end{cases} \tag{4}$$

An essential property in H-MRF is *consistency*: $D_j^{j-1}P_{j-1}^j = I$, that is, decimation after prediction must preserve the original data set. Prediction operators are often obtained by using subdivision schemes [3] which are *consistent* with the decimation. The detail coefficients (d^j) represent the non-redundant information in the prediction error (e^j), and $\alpha^j \equiv (\alpha^{j-1}, d^{j-1})$, $\forall j$, and both sequences have exactly the same number of components.

For further details on Harten's MRF, we refer the interested reader to [5] and references therein.

[3] $\{\mathcal{G}^j\}_j$ is nested if $\mathcal{G}^j \subset \mathcal{G}^{j+1}$. An example of nested grids on $[0, 1]$ is $\mathcal{G}^j = (i2^{-j})_{i=0}^{2^j}$.

3 The Multi-Scale Optimization (MSO) Technique

As mentioned before, using a standard optimization tool to solve problem (2) might be very costly, or even unfeasible, when N is large. This is mainly due to the large number of parameters involved, and the cost of evaluating the functional $\mathscr{D}$.

In this section we describe a feasible mechanism to solve problem (2), under appropriate conditions. Our MSO technique is based on an embedding of the problem within a conveniently chosen MRF, specified by given decimation and prediction operators. Here, we only provide a description of the final algorithm, which can be seen as a multi-scale 'parameter-reduction' approach to the original minimization problem. We refer the interested reader to [2] for specific details on the design of the proposed technique, and only mention here that, although the design of our MSO technique is based on (3), the decimation operator does not play a direct role in the practical application of the technique.

We assume that we are given an initial (discrete) shape, $\alpha = (\alpha_i)_{i=1}^N$ and that the underlying grid $(t_i)_{i=1}^N$ in (1) defines the finest resolution level in our chosen MRF. Then, starting at the coarsest level, we solve a sequence of optimization problems that involve a number of free parameters which increases when climbing up the MR ladder. More specifically, at the coarsest level we compute

$$\varepsilon_*^0 = \operatorname{argmin}_{\varepsilon^0 \in \mathbb{R}^{2\times N_0}} \mathscr{D}(\alpha^L + \Pi_{j=1}^L P_{j-1}^j \varepsilon^0), \quad \text{initial guess } \varepsilon_0^0 = 0. \tag{5}$$

Notice that, if N_0 is sufficiently small, (5) may be solved very fast. Then, for $k = 1, 2, \ldots, L$, we compute

$$\varepsilon_*^k = \operatorname{argmin}_{\varepsilon^k \in \mathbb{R}^{2\times N_k}} \mathscr{D}(\alpha^L + \Pi_{j=k+1}^L P_{j-1}^j \varepsilon^k), \quad \text{initial guess } \varepsilon_0^k = P_{k-1}^k \varepsilon_*^{k-1}. \tag{6}$$

We notice that the k-level optimal set of parameters, ε_*^k serves to construct a (discrete) curve at the highest resolution level, $\alpha^{L,k} := \alpha^L + \Pi_{j=k+1}^L P_{j-1}^j \varepsilon_*^k$, which represent the *best* kth-*level approximation* to the solution of problem (2). For $j = L$, $\alpha^{L,L}$ is the 'optimal' curve which solves problem (2).

It is expected that $\alpha^{L,k}$ gets closer to the 'optimal' curve when k increases. Hence, even though the successive optimization problems in (6) involve more parameters, when k increases, the initial guess is closer to the final solution, which reduces the total cost. The performance of the proposed MSO technique will be illustrated in the following section.

4 Numerical Experiments

The setting in the numerical experiments considered in this section is as follows:

- We seek to minimize a given functional with a specific minimization tool (in this work we use those available in Matlab) and using an initial guess provided by the user.
- As a stopping criteria we consider the max-norm of the difference between two consecutive iterates and the absolute value of the corresponding functional evaluations. We stop the minimization process when both quantities are less than a specified tolerance (*tol*). In addition, we stop the process (and consider it unfeasible) if the number of evaluations of the functional reaches 10^5.
- The prediction operator is always given by a B-spline subdivision scheme of order 5 [3] to ensure smooth perturbations of the initial shapes throughout the process.

4.1 An Academic Example

We start with an academic example that shows the advantages of using our proposed MSO technique, as opposed to using directly the minimization code to solve the problem in a large parameter space. The setting here is one-dimensional, and a bit simpler than that considered in (2): We seek to find the minimum of the functional

$$F(z) := \|z_i - \cos(2\pi t_i)\|_2^2, \qquad (t_i)_{i=0}^{2^L} = (i2^{-L})_{i=0}^{2^L},$$

starting from the initial shape $\bar{z} = (\lambda \cos(2\pi t_i))_{i=1}^{N}$. The parameter λ controls how far the initial guess is from the minimum of the functional, $z_{\min} = (\cos(2\pi t_i))_i$. The MATLAB `fminsearch` minimization tool has been selected to solve

$$\text{Find } \varepsilon_* \in \mathbb{R}^N : \ F(z^{\varepsilon *}) = \min_{\varepsilon \in \mathbb{R}^N} F(z^{\varepsilon}), \ \ z^{\varepsilon} := z + \varepsilon, \ \ N = 2^L + 1, \, L = 7, \qquad (7)$$

using the following minimization strategies ($tol = 10^{-4}$):

- DM: a Direct use of the chosen minimization tool on $\mathbb{R}^N$.
- MSO: using the same minimization code within our MSO strategy.

In Fig. 2, we show the output of the MSO strategy (always on top of the true solution, $z_{\min}$). In Fig. 2b, c the output of the DM strategy after 10^5 iterations is displayed. Notice that the DM strategy may be unfeasible when the initial guess is not sufficiently close to the desired solution.

We consider next the computational cost, measured by the number of functional evaluations (in more realistic cases, the remaining operations have a negligible cost

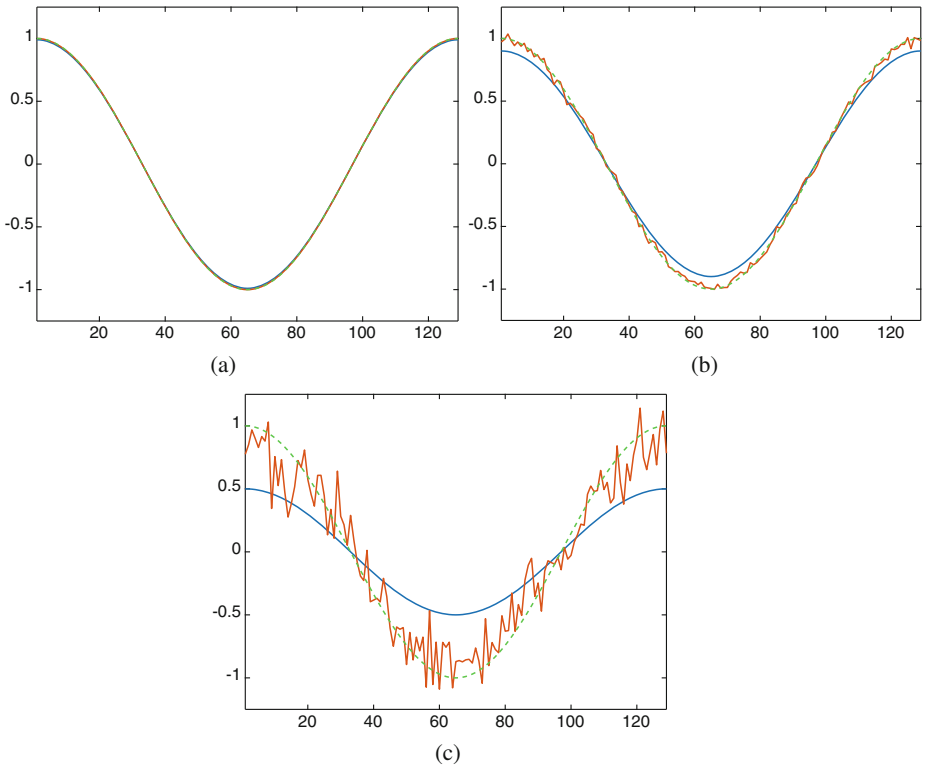


Fig. 2 Initial guess (*blue line*), output of DM algorithm (*red line*) and optimal solution/output of MSO algorithm (*green dashed line*) for (**a**) $\lambda = 0.999$, (**b**) $\lambda = 0.9$, (**c**) $\lambda = 0.5$

Table 1 F-eval= total number of function evaluations. $F_{min} = F(z_{last})$, $z_{last} = z + \epsilon_{last}$, ϵ_{last} last value computed

	MSO		DM	
λ	F-eval.	F_{min}	F-eval.	F_{min}
0.999	1758	9.48e−11	11,957	3.27e−9
0.9	4248	1.81e−10	*	6.81e−4
0.5	11,343	3.85e−11	*	3.58e−2

The symbol * means that the maximum allowed value of 10^5 has been reached

compared with the evaluation of the functional). The results, for both strategies and for various values of λ are compiled in Table 1. According to Table 1, only for $\lambda = 0.999$ the initial guess is close enough to $z_{\min}$ so that the DM strategy is feasible. For the other values of λ, only the MSO leads to a feasible algorithm. In addition, the MSO strategy leads to a very large computational gain for $\lambda = 0.999$. The results in Table 1 also show the value of $F(z_{last})$, which, for this problem, is also a measure (in the 2-norm) of how close this iterate is with respect to $z_{\min}$.

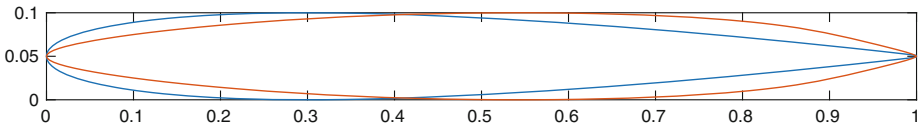


Fig. 3 Initial profile: *blue line*. Output of the MSO of the drag coefficient while preserving chord length and max height: *red line*. The drag coefficient is reduced from $9.21 \cdot 10^{-3}$ to $4.47 \cdot 10^{-3}$

4.2 A Realistic Case

In this section we consider a more realistic experiment, for which a DM strategy is unfeasible. We seek to solve problem (2) where the functional $\mathscr{D}$ provides the drag coefficient of a section, computed with `xfoil`, under a value of the Reynolds number equal to 10^6. In this section $tol = 10^{-5}$, $L = 5$, $N_0 = 2^2 + 1$.

The initial profile, provided by the user is a discrete version of the NACA0010-profile, with $N = 129$ points, shown in Fig. 1. The new sections α^ε generated during the minimization process are required to meet some structural conditions and the selected minimization tool is the MATLAB function `patternsearch`, which allows the specification of linear as well as nonlinear constraints.

In Fig. 3 we display the minimizing shape (in red) when the imposed constraints are that the chord length and maximum height of the initial profile (in blue) should be maintained.

5 Conclusions and Perspectives

We have described a novel multi-scale strategy to solve optimization problems in large parameter spaces. The technique only relies on the definition of an appropriate subdivision scheme, which depends on the type of application under consideration.

By considering an academic toy-problem we have shown that the technique computes the desired minimal shape with a rather small computational effort, compared with a direct strategy that uses the same underlying minimization tool. For realistic scenarios, only the multi-scale strategy is able to provide results.

More analytic work is necessary in order to determine the range of problems for which the technique would provide the desired optimal shape, as well as the influence of the underlying minimization code or the specification of the particular constraints.

Acknowledgements The authors acknowledge support from Project MTM2014-54388 (MINECO, Spain) and the FPU14/02216 grant (MECD, Spain)

References

1. Abbot, I.H., Von Doenhoff, A.E.: Theory of Wing Sections. Dover Publication, New York (1959)
2. Donat, R., López-Ureña, S.: A novel multi-scale strategy for multi-parametric optimization. (in preparation)
3. Dyn, N.: Subdivision Schemes in Computer-Aided Geometric Design. Advances in Numerical Analysis, vol. II. Lancaster (1990), pp. 36–104. Oxford Science Publication, Oxford University Press, New York (1992)
4. Fossati, F.: Aero-hydrodynamics and the performance of sailing Yachts: the science behind sailboats and their design. A&C Black, London (2009)
5. Harten, A.: Multiresolution representation of data: a general framework. SIAM J. Numer. Anal. **33**(3), 1205–1256 (1996)

Journal of Chromatography A, 1534 (2018) 32–42

Contents lists available at ScienceDirect

Journal of Chromatography A

journal homepage: www.elsevier.com/locate/chroma

Gradient design for liquid chromatography using multi-scale optimization☆

S. López-Ureña[a], J.R. Torres-Lapasió[b,*], R. Donat[a], M.C. García-Alvarez-Coque[b]

[a] Department of Mathematics, Faculty of Mathematics, Universitat de València, c/Dr. Moliner 50, 46100, Burjassot, Spain
[b] Department of Analytical Chemistry, Faculty of Chemistry, Universitat de València, c/Dr. Moliner 50, 46100, Burjassot, Spain

ARTICLE INFO

Article history:
Received 12 September 2017
Received in revised form
12 December 2017
Accepted 15 December 2017
Available online 19 December 2017

Keywords:
Liquid chromatography
Multi-linear gradients
Cubic splines
Resolution
Multi-scale optimization

ABSTRACT

In reversed phase-liquid chromatography, the usual solution to the "general elution problem" is the application of gradient elution with programmed changes of organic solvent (or other properties). A correct quantification of chromatographic peaks in liquid chromatography requires well resolved signals in a proper analysis time. When the complexity of the sample is high, the gradient program should be accommodated to the local resolution needs of each analyte. This makes the optimization of such situations rather troublesome, since enhancing the resolution for a given analyte may imply a collateral worsening of the resolution of other analytes. The aim of this work is to design multi-linear gradients that maximize the resolution, while fulfilling some restrictions: all peaks should be eluted before a given maximal time, the gradient should be flat or increasing, and sudden changes close to eluting peaks are penalized. Consequently, an equilibrated baseline resolution for all compounds is sought. This goal is achieved by splitting the optimization problem in a multi-scale framework. In each scale κ, an optimization problem is solved with $N_\kappa \approx 2^\kappa$ variables that are used to build the gradients. The N_κ variables define cubic splines written in terms of a B-spline basis. This allows expressing gradients as polygonals of M points approximating the splines. The cubic splines are built using subdivision schemes, a technique of fast generation of smooth curves, compatible with the multi-scale framework. Owing to the nature of the problem and the presence of multiple local maxima, the algorithm used in the optimization problem of each scale κ should be "global", such as the pattern-search algorithm. The multi-scale optimization approach is successfully applied to find the best multi-linear gradient for resolving a mixture of amino acid derivatives.

1. Introduction

Reversed-phase liquid chromatography (RPLC) with programmed changes (i.e., gradients) in the amount of organic modifier is the technique of choice for the separation of complex mixtures of compounds exhibiting a wide range of polarities, which cannot be resolved in an acceptable analysis time with a mobile phase at fixed composition [1–6]. Gradients speed up the elution of the strongly retained compounds, and keep large enough the retention of those poorly retained, while getting good resolution. For this purpose, the elution strength of the mobile phase should be initially low, becoming increasingly stronger by increasing the concentration of modifier as the separation progresses. To the effect of decreased analysis time, the benefits of peak compression for increasing gradients should be added.

In practice, nowadays instruments generate gradients by performing consecutive small isocratic steps at increasing modifier concentration. An accurate sloped gradient requires a large number of these small steps. A simpler solution is to approximate gradients to a few large isocratic steps with a prefixed increment in solvent concentration (i.e., stepwise or multi-isocratic gradients) [7–9]. Usually, the stepwise variation pattern is designed to increase the modifier concentration. One of these approaches was developed by Cela et al. [7,8]. It assumes a certain number of equally spaced modifier concentration steps in the domain covered by the experimental design. The duration of these steps is varied to optimize the separation.

Linear gradients, and especially those where more than one segment of different slope is defined (multi-linear gradients), are good alternatives to multi-isocratic gradients [10–12]. In order to speed up the elution, the gradient slopes should be always positive, but occasionally, one or more isocratic steps can be introduced to

☆ Selected paper from 45th International Symposium on High Performance Liquid Phase Separations and Related Techniques (HPLC 2017), 18–22 June 2017, Prague, Czechia.

* Corresponding author.
E-mail address: jrtorres@uv.es (J.R. Torres-Lapasió).

https://doi.org/10.1016/j.chroma.2017.12.040
0021-9673/

offer more separation time for closely eluting late compounds. The optimization of linear and multi-linear gradients can be made by trial-and-error. There are RPLC simulation packages that facilitate this approach [13,14]. Trial-and-error may be satisfactory in many situations, but inappropriate for complex separations, or complex gradient programs.

Multi-linear gradients are usually defined by setting the coordinates (time and modifier concentration) of the transition points between consecutive linear segments, namely, the gradient nodes. Years ago, pumps were too limited to generate highly precise multi-linear gradients, but the situation nowadays is completely different: accurate complex gradients can be easily set up. The accuracy of the search can be modulated by the way the nodes are encoded; that is, by discretizing the possible values to wider or narrower distance between consecutive levels. Owing to the multi-parametric nature of multi-linear gradients, the efficiency of interpretive optimizations will depend on such codification. Along the years, several proposals have been formulated to make the search faster and more reliable, but paying a price: any discretization implies losing accuracy in the solution. This can be illustrated by the results found in previous work [15], where we developed a numerical procedure where the number of nodes was predefined and each node could vary only in a certain time range, so that the distribution of nodes tended to be uniform. Once codified, the position of the nodes allowed being moved in discrete steps. The procedure began with a reduced number of nodes (e.g., 5 nodes), and was executed in successive runs. In each run, the number of nodes could be increased or decreased attending to the previous results. The gradient time and the initial and final concentrations in the gradient could be identically adapted. Other variants of this outline can be proposed, such as dividing the time domain in other ways, or discretizing only one variable (the time or the concentration).

Another type of multi-linear gradient optimization is the so-called "one-segment-per-group-of-components" strategy [16]. In this strategy, the slope is adjusted after the elution of each individual component (or group of components) in the sample, letting the retention properties of the different analytes auto-guide the course of the gradient profile. The analysis time is significantly reduced compared to the best single linear gradient, enabling faster searches than the traditional multi-linear gradient methods. However, it could be inappropriate for complex situations, where a single linear gradient fails, because the fact of setting an early node might negatively impact the separation of later eluting compounds. For this reason, the one-segment-per-group of components method requires an iterative outline: when a bad resolution for a later eluting compound is encountered, the search is stopped and repeated from the start (using a different initial slope). In this way, the optimal solution is more likely to be found in spite of a bad choice of the initial gradient slope. Owing to the sequential construction of the gradient (which implies discarding part of the search space in each choice), in practice, the search is self-constrained and cannot deal well with very complex resolution surfaces. Other consequences of the sequential nature of the search are the inability to cope well with non-related gradients and the slowness of the search speed for complex problems.

In this work, we propose a different approach to optimize multi-linear gradients using a non-conventional outline based on cubic splines and a multi-scale optimization technique [17]. This confers two valuable advantages to the optimization: (i) the search has a more global nature, and (ii) a fine-tuning of the separation is made to satisfy the local resolution requirements of each solute. The new approach is able to locate complex solutions that are easily overlooked by other search techniques.

2. Theory

2.1. Measurement of chromatographic resolution

As commented, the context of this work is RPLC. In general, a gradient $g(t)$ can be outlined as a function defined between 0 and t_G (gradient time) that specifies the amount of organic solvent present in the mobile phase at each time t. Gradients can contain isocratic steps and are usually increasing, that is $g(t_1) \leq g(t_2)$ whenever $t_1 < t_2$. The gradient program should be preferably developed in the solvent region covered by the training experiments. More restrictions and modifications can be added, as will be discussed later.

The implementation of a gradient g in the instrument gives rise to a chromatographic signal, $c(g)$, which in case the sample contains p analytes, will be composed of a maximal number of p peaks. For the next treatment, we will assume that $c(g)$ is a noise-free signal and lacks of baseline. The individual signal of each peak i will be denoted by $c_i(g)$, with $i = 1, 2, \ldots, p$. The sum of the $c_i(g)$ signals constitutes the global signal (Fig. 1):

$$c(g) = c_1(g) + c_2(g) + \ldots + c_p(g) \quad (1)$$

Each analyte peak $c_i(g)$ in the chromatogram has an associated resolution, namely $r_i(g)$, which can be computed according to different criteria. The peak purity, which has been demonstrated to offer the best performance, is used in this work as resolution criterion [18–20]. This is an intuitive normalized measurement, which is evaluated from computer predicted chromatograms, not from experimental ones. Peak purities are calculated from the expected individual signals of all constituents in a sample. Each signal has been previously calculated with the assistance of retention and peak profile models, fitted from standards and following a certain experimental design.

The retention time under gradient elution ($t_{g,i}$) was obtained by solving numerically the following integral equation:

$$t_0 = \int_0^{t_{g,i}-t_0} \frac{dt}{k_i(\varphi(t))} \quad (2)$$

where $\varphi(t)$ is the gradient program describing the change in organic solvent content in the mobile phase, t_0 is the dead time, and $k_i(\varphi)$ is the retention factor for each solute, described by:

$$\log k_i = d_{0,i} + d_{1,i}\,\varphi + d_{2,i}\,\varphi^2 \quad (3)$$

The peak profile was described by:

$$A_i = a_0 + a_1\,t_{R,i} + a_2\,t_{R,i}^2 \quad (4)$$

$$B_i = b_0 + b_1\,t_{R,i} + b_2\,t_{R,i}^2 \quad (5)$$

A_i and B_i being the left and right half-widths, respectively, of peak c_i at 10% peak height for each solute, and $t_{R,i}$ the retention time. The coefficients $d_{0\text{-}2,i}$, $a_{0\text{-}2}$ and $b_{0\text{-}2}$ are obtained by regression. In contrast to Eq. (3), which should be fitted for each solute, Eqs. (4) and (5) were fitted for all solutes eluted at several φ values. The half-width models allowed predicting both peak width and asymmetry. More details can be found in Ref. [15].

The peak purity criterion facilitates the combination of elementary resolution values into a single global measurement, and the combination with other quality criteria. It is also very realistic since it considers the full signal (peak profile and size), and finally, it qualifies individual peaks instead of peak pairs (as is the case of the R_S criterion), so there is no ambiguous relationships between the identities of the peaks and the numerical resolution values.

The goal of the optimization process is maximizing the global resolution by tuning the gradient g. The global resolution $R(g)$ asso-

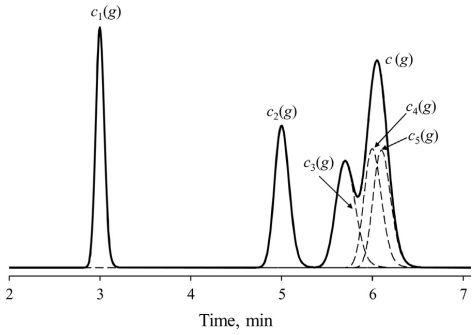


Fig. 1. Individual contributions, $c_i(g)$ (dashed lines), and global chromatogram, $c(g)$ (full line), for a mixture of five compounds.

ciated to a given gradient can be measured in several ways [18]. The definition applied in this work is:

$$R(g) = \prod_{i=1}^{p} (\varepsilon + r_i(g)^{\gamma}) \tag{6}$$

where γ and ε are weighting parameters with $\gamma \geq 1$ and $\varepsilon > 0$. Whenever $R(g) \approx 1$, the chromatogram is well resolved, and for those gradients with $R(g) \approx 0$, significant overlapping occurs at least between two peaks. Parameter γ emphasizes the weight of incompletely resolved peaks in the final measurement, whereas ε prevents that an unresolved peak j dominates $R(g)$, which may hide incidental improvements of resolution for the remaining peaks. In other words, ε prevents null values for $R(g)$ when at least one $r_i(g)$ value is null.

The use of the product as a combination criterion of elementary resolutions is justified, because it prioritizes the maximization of the resolution of critical peaks; the larger the value of γ, the higher such priority. Meanwhile, the addition of a small positive offset ε guarantees that if any r_i is increased, then R is increased, even when for a given peak j, $r_j = 0$, which is often the case.

2.2. Addition of new constraints to the objective function

In general terms, an optimization problem consists of finding a set of parameters that can be arranged as a vector, namely the optimal vector, which maximizes (or minimizes) an objective function quantifying the success in resolving the problem. In our case, the objective function is the resolution, $R(g)$ (Eq. (6)), which should be maximized. Several constraints must be imposed to the problem to guarantee that the solution found is adequate for our needs. Some of these constraints do not depend on the associated chromatograms (e.g., positive gradients with or without isocratic steps), whereas others require examining the associated simulation (e.g., peaks not exceeding a certain asymmetry level, or an analysis time below a given value). In this case, it is mandatory to simulate the peak positions to check whether the gradient is suitable or not. In this work, the following constraints were implemented:

(i) *All peaks in the chromatogram should appear before a certain target analysis time,* t_{max}. Longer analysis times would benefit the separation of all peaks in the chromatogram. However, under a practical standpoint, times should be as short as possible, provided enough resolution is obtained. With this purpose, late appearing peaks were penalized by multiplying their individual resolution $r_i(g)$ by:

$$\Phi_i(g) = \frac{1}{1 + \beta_1 \ \max\{0, t_{R,i} - t_{max}\}} \tag{7}$$

where $t_{R,i}$ is the position of peak i and $\beta_1 > 0$. Note that $\Phi_i(g) = 1$ when the peaks appear before the target analysis time ($t_{R,i} \leq t_{max}$), and will be in the range $0 < \Phi_i(g) < 1$ if the peak appears too late ($t_{R,i} > t_{max}$). The larger the value of β_1, the smaller $\Phi_i(g)$, which is translated in a more important penalization. In our simulations, we have explored the consequences of penalizing long elution in all peaks, using $\beta_1 = 1$. Alternatively, the penalization can only be applied to the last peak.

(ii) *The slope of the gradient should not be too steep close to peak appearance*, otherwise the peak may be distorted. It should be noted that the consequences of any change in the solvent composition in the mixer needs a certain time to reach the solute band, which includes the dwell time, t_{dw} (delay time that the eluent needs to cross the distance between mixer and column inlet), and the time the solvent front needs to reach the solute band from the column inlet.

To penalize steep slopes, $r_i(g)$ is multiplied by a function that depends on the maximal slope of the gradient around peak $c_i(g)$. The significant neighborhood of the peak was arbitrarily taken as the range $-3A_i < t < +3B_i$, where the half-widths A_i, $B_i > 0$. Considering this, the penalization function is:

$$\Psi_i(g) = \frac{1}{1 + (\beta_2^{-1} \ \max(g'(t_{R,i} + t)))^2} \tag{8}$$

where $\beta_2 > 0$ and g' is the local slope in the neighborhood of peak i. As indicated above, only gradients with positive or null slopes are considered ($g' \geq 0$). In Eq. (8), the maximal slope around the peak is divided by β_2 (which represents the maximal admissible slope). Special attention should be paid to the squared power in the denominator of Eq. (8). This emphasizes the limit of admissible slopes: if the slope around the peak is smaller than β_2 (region of admissible slopes) then the squared term in the denominator becomes closer to zero (and $\Psi_i(g) \approx 1$). On the contrary, if the slope is greater than β_2, then this term becomes well above 1 ($\Psi_i(g) \approx 0$).

Along the optimization, a value of $\beta_2 = 10(\varphi_{max} - \varphi_{min})/t_{max}$ was arbitrarily taken as maximal admissible slope, where φ_{max} and φ_{min} delimit the solvent region covered by the experimental design ($\varphi_{min} \leq g(t) \leq \varphi_{max}$). The ratio $(\varphi_{max} - \varphi_{min})/t_{max}$ corresponds to the slope of a linear gradient going from φ_{min} to φ_{max} in the target analysis time.

Altogether, the above comments lead to the following objective function, suitable to govern the optimization process (see Eqs. (6)–(8)):

$$F(g) = \prod_{i=1}^{p} (\varepsilon + r_i(g)^{\gamma} \ \Phi_i(g)\, \Psi_i(g)) \tag{9}$$

2.3. Using cubic splines for modeling gradient elution

The procedure proposed in this work to build the gradients is described in this and the next sections. To understand the approach, we should describe how gradients are obtained from cubic splines. As indicated above, the input of the objective function is a gradient, g, which is defined as an increasing function.

First, we should explain what is meant by a cubic spline, which in this work represents approximately the gradient (the concentration of organic solvent as a function of time). A cubic spline, denoted by s, is a function with continuous second-order derivative, which is piecewise composed of polynomials of third degree (Fig. 2a). In our outline, a cubic spline s is defined by $N+1$ equally spaced nodes which are located at times $(t_n)_{n=0}^{N}$, where the right sub- and supra-indexes indicate that index n varies between 0 and N. The time location of the nodes is a multiple of the spacing between the nodes: $t_n = nh$, where $h = t_{max}/N$ is the distance between consecu-

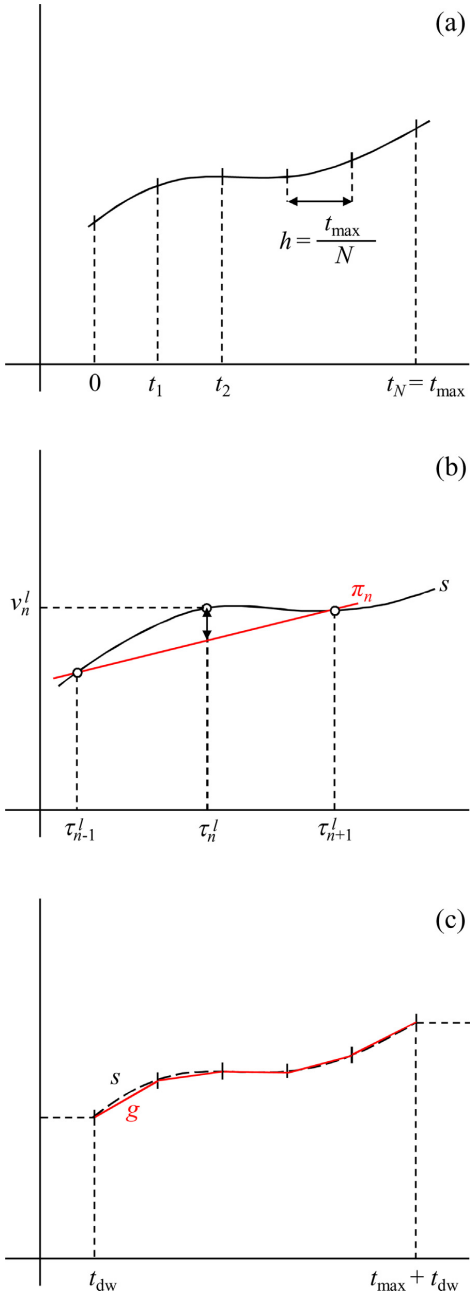


Fig. 2. Use of cubic splines to build multi-linear gradients: (a) cubic spline with 6 equally-spaced nodes ($N = 5$, h represents the spacing between nodes), (b) cubic spline (black) and straight-line (red) joining the two neighboring points of v_n^l, and (c) cubic spline (black, dashed line) and multi-linear gradient approximation (red, full line) obtained from that spline using the P_M operator. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

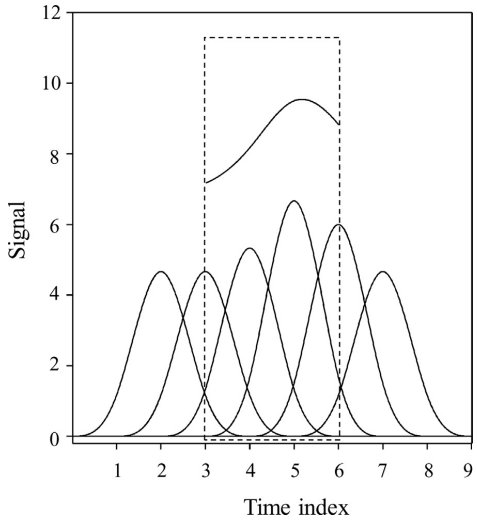


Fig. 3. Description of a cubic spline using the B-Spline basis,*b*. The drawn cubic spline (top) is a linear combination of some translations of *b*, as described in Eq. (10).

tive nodes in the cubic spline. Any spline $s(t)$ can be written as a linear combination of the B-spline basis, b:

$$s(t) = \sum_{n=-1}^{N+1} f_n \; b\left(\frac{t - t_n}{h}\right) \tag{10}$$

where coefficients $f_{-1}, f_0, \ldots, f_{N+1}$ are real numbers, and $0 \le t \le t_{max}$. Fig. 3 depicts how a cubic spline can be described using the B-spline basis. The coefficients f_n determine the heights of a series of bell-shaped functions, all of them with null values outside the interval (t_{n-2}, t_{n+2}). The linear combination of the bell-shaped functions gives rise to the spline. For more information, readers should consult Ref. [21].

For the computation of cubic splines, "subdivision schemes" are recommended and will be used in this work. This powerful technique is usually applied in Computed Aided Geometric Design for the fast generation of curves and surfaces [21]. Roughly speaking, a *subdivision scheme* is a recursive process where sets of points are computed iteratively using simple rules. In this work, the number of points is duplicated in each iteration. For any initial set of points, the process converges to a function, in our case, a cubic spline. With the assistance of *subdivision schemes*, cubic splines can be readily generated.

2.4. Taking advantage of the properties of B-splines for constraining gradients

B-splines have two interesting properties that are useful to build gradients:

(i) *Bounding*, which forces the concentration of organic solvent in the spline at time t, $s(t)$, to be between the minimal (φ_{min}) and maximal concentrations (φ_{max}) covered by the experimental design. This requirement is fulfilled if the coefficients f_n in Eq. (10) are bracketed between φ_{min} and φ_{max}. In this way, if $\varphi_{min} \le f_n \le \varphi_{max}$ for all $n = -1, 0, 1, \ldots, N+1$, then $\varphi_{min} \le s(t) \le \varphi_{max}$ for all times t in the range $0 \le t \le t_{max}$.

(ii) *Monotonicity*, which guarantees that the generated spline will be increasing when the coefficients f_n form an increasing sequence: if $f_{n-1} \leq f_n$ for all n, then s is increasing.

Both properties (bounding and monotonicity) allow the control of important aspects in the spline concentration function, $s(t)$, and will influence the properties of the generated gradient $g(t)$. This will be described below.

2.5. Simplifying cubic splines for defining multi-linear gradients

Our final aim is optimizing multi-linear gradients. The gradients will be isocratic before reaching the dwell time, t_{dw}, and beyond $t_{max} + t_{dw}$. In the range $t_{dw} \leq t \leq t_{max} + t_{dw}$, the gradient will be a multi-linear function with M nodes, approximating the cubic spline $s(t)$. In this work, we first considered $M = 15$, but as demonstrated in Section 4.4, a much smaller number of nodes can give satisfactory results.

We will explain next how to proceed to obtain the simplified multi-linear gradient (with a reduced number of nodes, M), starting from another multi-linear gradient (with a large number of nodes, L), which approximates the spline s. The number of nodes is reduced in successive iterations. In each iteration, the difference between the current multi-linear function and the one obtained after eliminating each node is computed. That node for which the smallest difference is obtained is the one finally eliminated in that iteration, since its contribution to the description of the spline is less significant. The process continues up to only M nodes remain.

The operations carried out to simplify the original spline to a multi-linear gradient $g(t)$ will be denoted by:

$$g = P_M(s) \tag{11}$$

where $P_M(s)$ is the operator used for the simplification of the spline to a multi-linear function of M nodes, which is executed as follows:

Step 1: Consider a large enough integer number, L (for instance, $L = 300$). Define the sequence of equally spaced times τ_n^L for the nodes of the multi-linear gradient (Fig. 2b), as follows:

$$\tau_n^L = \frac{n-1}{L-1} t_{max} \tag{12}$$

where the index n ranges from 1 to L; hence, τ_n^L ranges from 0 to t_{max}. Obtain the concentration values in spline s at each time τ_n^L, namely $v_n^L = s(\tau_n^L)$, that forms the sequence v^L.

Step 2: Set $l = L$, where l is the current number of nodes, and reduce the number of nodes along the iterations as explained next. In this process, the initial and final nodes are preserved.

(a) For each $n = 2$ to $l - 1$, consider the straight-line π_n, whose values at τ_{n-1}^l and τ_{n+1}^l match those of the cubic spline (Fig. 2b): $v_{n-1}^l = \pi_n(\tau_{n-1}^l)$ and $v_{n+1}^l = \pi_n(\tau_{n+1}^l)$. The difference between the intermediate value, $\pi_n(\tau_n^l)$, and the corresponding spline value is calculated:

$$\Delta_n^l = \left| \pi_n(\tau_n^l) - v_n^l \right| \tag{13}$$

This operation is carried out with all possible subsequent triads of nodes (τ_{n-1}^l, τ_n^l and τ_{n+1}^l).

(b) Find the node n^* giving the minimal difference, Δ_{n*}^l.

(c) Define the next sequence of values τ^{l-1} and v^{l-1} involving $l - 1$ values, by eliminating the n^* value from sequences τ^l and v^l, and set $l = l - 1$.

As far as $l > M$, proceed again following the (a–c) steps. The process is repeated until the predetermined number of nodes, M, is obtained so that the multi-linear function agrees the best with the target cubic spline.

Step 3: The multi-linear function satisfying $g(0 + t_{dw}) = v_1^M = s(0)$, $g(t_{max} + t_{dw}) = v_M^M = s(t_G + t_{dw})$, and $g(\tau_n^M + t_{dw}) = v_n^M$, for any $n = 1, 2, \ldots, M$, is built.

Note that outside the gradient region ($t < t_{dw}$ and $t > t_G + t_{dw}$), the elution is isocratic.

An example of a multi-linear gradient defined from a cubic spline according to the P_M operator is illustrated in Fig. 2c. It should be here remarked that $g(t)$ exhibits the same properties as $s(t)$ (i.e., bounding and monotonicity).

2.6. General equation describing the optimization problem

It should be here reminded that from any f_n values, and using the B-spline basis (Fig. 3), we obtain a cubic spline, which is further simplified to a multi-linear gradient with the wished number of nodes (Fig. 2c). To this point, we have described the different elements that constitute our optimization problem. The final expression that summarizes the process is the following:

$$\text{maximize} \quad F\left[P_M\left(\sum_{n=-1}^{N+1} f_n \; b\left(\frac{t - t_n}{h}\right)\right)\right] \tag{14}$$

$$\text{subjected to } \varphi_{min} \leq f_n \leq \varphi_{max} (n = -1, 2, \ldots, N+1) \tag{15}$$

$$\text{and } f_{n-1} \leq f_n (n = 0, 1, 2, \ldots, N+1) \tag{16}$$

In Eq. (14), three equations are nested (Eqs. (9)–(11)): the description of a cubic spline (Eq. (10)), its approximation to a multi-linear gradient $P_M(s)$ (Eq. (11)), and the objective function quantifying the quality of the separation, F (Eq. (9)). We denote by $\hat{f}_n$ the optimal values that determine the expression of the best multi-linear gradient, $\hat{g}$:

$$\hat{g} = P_M\left(\sum_{n=-1}^{N+1} \hat{f}_n \, b\left(\frac{t - t_n}{h}\right)\right) \tag{17}$$

which is calculated through Eq. (14) using a maximization algorithm, able to deal with the constraints (15) and (16). Alternatively, Eq. (14) can be modified to yield an unconstrained optimization problem, which is the approach we followed in this work. This enables the use of any maximization algorithm.

The unconstrained optimization problem is based on the use of a "projection operator" Q, which given a sequence $f = (f_n)_{n=-1}^{N+1}$, returns a second sequence $(Q(f))_{n=-1}^{N+1}$ fulfilling the constraints (15) and (16). In this way, instead of performing the optimization subject to the restrictions, an unrestricted search is carried out, where the Q operator is applied to guarantee that the solutions satisfy the restrictions. The "projection operator" will not modify f (i.e., $Q(f) = f$) if f already satisfies the inequalities (15) and (16).

In detail, Q is defined as follows: First, any $f = (f_n)_{n=-1}^{N+1}$ is transformed into another vector $Q_1(f)$ fulfilling $\varphi_{min} \leq (Q_1(f))_n \leq \varphi_{max}$. For this purpose, the following operation is carried out:

$$\frac{f_n - \varphi_{min}}{\varphi_{max} - \varphi_{min}} = q_n + \frac{r_n}{\varphi_{max} - \varphi_{min}}, \; 0 \leq r_n < \varphi_{max} - \varphi_{min} \tag{18}$$

where q_n and r_n are the quotient (an integer) and remainder (a real number) of the division. If q_n is even, then we define $(Q_1(f))_n = \varphi_{min} + r_n$; if q_n is odd, then $(Q_1(f))_n = \varphi_{max} - r_n$. The effect of this operation, which depicts a zig-zag pattern, is shown in the Supplementary material (Fig. 1S). Finally, we define $Q(f)$ as the increasing sorted sequence of $Q_1(f)$.

Since $Q(f)$ fulfills the constraints (15) and (16), the next unconstrained optimization problem can be solved instead of Eq. (14):

$$\text{maximize} \quad F\left[P_M\left(\sum_{n=-1}^{N+1} (Q(f))_n \, b\left(\frac{t - t_n}{h}\right)\right)\right] \tag{19}$$

The optimal multi-linear gradient is (see Eq. (17)):

$$\hat{g} = P_M\left(\sum_{n=-1}^{N+1} (Q(\hat{f}))_n \, b\left(\frac{t - t_n}{h}\right)\right) \tag{20}$$

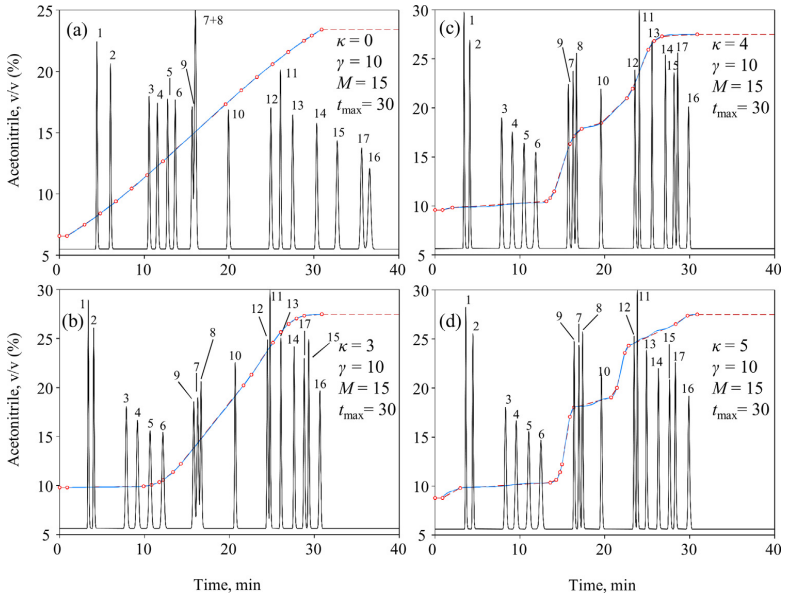


Fig. 4. Outline of the separation problem described in Section 4.1. The multi-scale strategy searches the optimal gradient in several steps: starting from an initial gradient ($\kappa = 0$), the algorithm recursively finds more detailed gradients ($\kappa = 3$, 4 and 5, respectively), until it arrives to the optimal one ($\kappa = 7$, see Fig. 5a). The results for iterations $\kappa = 1$, 2 and 6 are not shown, since the changes are less significant. The running parameters are overlaid. The penalization function Ψ was included in the optimization.

2.7. The multi-scale optimization technique

Maximization algorithms are usually iterative processes that starting from an initial estimate, arrive recursively to the maximum of the objective function. These algorithms can be classified in two categories: local and global. Global algorithms are more suitable for our complex problem, since chromatographic optimizations present many local maxima where local algorithms could get stuck, without finding the global one. In a previous report [17], two authors of this work found similar issues in an optimization problem belonging to other field. In that report, the aim was improving the performance of a sailing yacht by modifying the geometry of some elements in its design. The optimization problem consisted in reducing the drag with the water, while fulfilling some constraints. The goal was achieved using a *multi-scale optimization technique*, which is a novel approach that consists in outlining the optimization problem in several scales, solving a reduced optimization problem in each scale. A good point of such technique is that it is compatible with any maximization (or minimization) algorithm needed to solve each reduced optimization problem.

In the context of the optimization of gradients in HPLC, the *multi-scale optimization technique* can be outlined as follows: The process is started (iteration $\kappa = 0$) by searching the sub-optimal multi-linear gradient of M nodes, which is built from a cubic spline of $N_0 + 1$ nodes (N_0 being a small arbitrary value). The multi-linear gradient maximizes the objective function F (Eq. (14) or better Eq. (19), which adds constraints to Eq. (14)). From now on, we will use exclusively Eq. (19). Thus, Eq. (19) is solved for $N = N_0$. We denote by $\hat{g}^0$ the best gradient found by the maximization algorithm in this iteration. In iteration κ, the search space is restricted to the multi-linear gradients of M nodes obtained from splines with $N_\kappa + 1$ nodes, where $N_\kappa = 2N_{\kappa-1} = 2^\kappa N_0$. This means that the number of nodes of the previous iteration is duplicated in the current one. Therefore, Eq. (19) is solved for $N = N_\kappa$ in order to find the sub-optimal gradient, $\hat{g}^k$.

In iteration κ, the B-spline width is reduced to the half value of the former iteration. This increases the detail in the definition of the gradient program. The refinability property of the B-spline basis indicates that the B-splines of iteration $\kappa - 1$ can be expressed as a sum of the B-splines of iteration κ (for more information see Ref. [21]):

$$b(t/h_{\kappa-1}) = \frac{1}{8}\left[b(t/h_\kappa - 2) + 4b(t/h_\kappa - 1) + 6b(t/h_\kappa) + 4b(t/h_\kappa + 1) + b(t/h_\kappa + 2)\right] \tag{21}$$

where $h_\kappa = t_{\max}/N_\kappa$. Hence, the search space of iteration κ contains the optimal gradient of the previous iteration, $\hat{g}^{k-1}$. Therefore, we will start the maximization process of this iteration from $\hat{g}^{k-1}$. This nested outline speeds up remarkably the optimization process. Another consequence is that $\hat{g}^k$ is improved with regard to $\hat{g}^{k-1}$ (i.e., it is never worsened).

Accordingly, the initial values for iteration κ, $(f_n^k)_{n=-1}^{N_\kappa+1}$, are computed from the solution found in iteration $\kappa - 1$, $(\hat{f}_n^{k-1})_{n=-1}^{N_{\kappa-1}+1}$. The computation is carried out using the following subdivision scheme formula, which can be deduced from Eq. (21) (see also Ref. [21]):

$$f_{2n}^\kappa = \frac{1}{8}(Q(\hat{f}^{k-1}))_{n-1} + \frac{3}{4}(Q(\hat{f}^{k-1}))_n + \frac{1}{8}(Q(\hat{f}^{k-1}))_{n+1} \tag{22}$$

$$f_{2n+1}^\kappa = \frac{1}{2}(Q(\hat{f}^{k-1}))_n + \frac{1}{2}(Q(\hat{f}^{k-1}))_{n+1} \tag{23}$$

The final iteration, namely $\kappa = K$, which should be established before running the algorithm, provides an optimal gradient $\hat{g}^K$ better than all previous gradients. The separation problem described in Eq. (19) is thus solved for $N_K = 2^K N_0$. We should remark that this optimization process is faster than resolving directly Eq. (19) for $N = N_K$, since the initial estimate for iteration K is the solution found in the former iteration $K - 1$.

As commented, resolving directly Eq. (19) is prone to get stuck in local maxima. The multi-scale optimization usually avoids local solutions because the search is started with a small number of variables (N_0), and for the successive iterations the initial estimates get closer to the optimal. Also, with regard to classical multi-linear functions, a cubic spline has much better approximation properties. This makes the proposed algorithm faster.

The architecture of the *multi-scale optimization technique* is next summarized for gradient optimization:

Step 1: Set $\kappa = 0$, $N_0 \geq 2$, $K \geq 0$, an initial estimate $(f_n^0)_{n=-1}^{N_0+1}$ and a maximization algorithm.

Step 2: Solve Eq. (19) starting from $(f_n^k)_{n=-1}^{N_\kappa+1}$ using the maximization algorithm. The optimal values $(\hat{f}_n^k)_{n=-1}^{N_\kappa+1}$, and consequently the sub-optimal gradient $\hat{g}^k$ using Eq. (20), are obtained.

Step 3: The process is finished when $\kappa = K$. Otherwise:

Step 4: Compute the initial estimate for iteration $\kappa+1$ $(f_n^{k+1})_{n=-1}^{N_{\kappa+1}+1}$ from the current optimal values $(\hat{f}_n^k)_{n=-1}^{N_\kappa+1}$, using Eqs. (22) and (23).

Step 5: Set $\kappa = \kappa + 1$, $N_\kappa = 2^\kappa N_0$, and return to Step 2.

In our numerical experiments, the MATLAB *patternsearch* function (2016b version, The MathWorks Inc., Natick, MA, USA) was used for maximization. Remember that Eq. (14) can be solved instead of Eq. (19) if a maximization algorithm that accepts constraints (15) and (16) is used.

3. Experimental

The aim of this work is to report a new optimization strategy. To develop it, we have used information taken from our laboratory database, corresponding to the separation of amino acid derivatives, which we found adequate to assay the approach. Other examples would be equally good to test the new optimization strategy, obtaining similar conclusions. The particular final behavior of our sample is of no concern.

Mixtures of the following L-amino acids were considered in this work: (1) Aspartic acid, (2) glutamic acid, (3) asparagine, (4) serine, (5) glutamine, (6) histidine, (7) glycine, (8) arginine, (9) threonine, (10) alanine, (11) cysteine, (12) tyrosine, (13) valine, (14) methionine, (15) isoleucine, (16) tryptophan, and (17) lysine. The amino acids were derivatized with *o*-phthalaldehyde (OPA) and *N*-acetylcysteine (NAC), in the presence of boric/borate buffer at pH 9.5. The amino acid derivatives were eluted in the isocratic mode to get a training set of chromatographic data. Hydro-organic mobile phases with acetonitrile (Scharlab, HPLC grade) were prepared, buffered at pH 6.5 with 5×10^{-3} M dihydrated trisodium citrate (Scharlab) and NaOH (AnalaR, Poole, UK). The separation was carried out with a 250 mm × 4.6 mm i.d. Inertsil ODS3 column with 5 μm particle size (Análisis Vínicos, Tomelloso, Spain).

The information obtained for each particular amino acid was used to model the chromatographic behavior and obtain the best separation conditions through the multi-scale optimization algorithm described in this work. Peak position under gradient elution was predicted by numerical integration [15] (Eq. (2)). The experimental work to get the coefficients in Eqs. (3)–(5) was carried out from isocratic measurements, owing to their higher intrinsic accuracy, although it would have been simpler to use two scouting gradients to predict the isocratic retention. The isocratic training set used to build the models that describe the peak retention and half-widths covered the 5.0-27.5% acetonitrile range. The gradients covered the same concentration range. In the Supplementary material, the solvent domain in the experimental design (Fig. 2S) and the range of retention times for each solute (Fig. 3S) are illustrated. For each amino acid, two to five mobile phase compositions were assayed. Duplicated injections were carried out at each composition.

A modular Agilent chromatograph (Model HP 1100, Waldbronn, Germany), consisting of quaternary pump, autosampler, thermostated column compartment, and UV–vis detector of variable wavelength, were used to acquire the chromatographic signals. The detection was performed at 335 nm and the injection volume was 20 μl. In all cases, the mobile phase flow rate was kept constant at a value of 1 ml/min. The column temperature was 25 °C. Other details on the procedure to analyze the amino acid derivatives are given elsewhere [22].

4. Results and discussion

In this section, the application of the algorithm for gradient design proposed in this work is examined. As commented above, a mixture of 17 OPA-NAC amino acid derivatives was used to check its performance. Although the methodology is valid for predicting optimal conditions for samples containing analytes in specific concentrations at different levels, unitary areas were set for all peaks in order to get a more general solution. The good quality of the predictive models used in the optimization of the separation of the mixture of amino acids, based on Eqs. (2)–(5), can be observed in Figs. 4S–6S in the Supplementary material, where experimental and predicted chromatograms for three gradient configurations are compared.

4.1. Performance of the multi-scale optimization technique

The progress of the algorithm described in Section 2.7 is here shown when applied to find the best separation conditions for the mixture of 17 amino acid derivatives. With this purpose, the optimization problem was solved using the following working parameters: $\gamma = 10$, $t_{max} = 30$, $\varphi_{min} = 5.0\%$, $\varphi_{max} = 27.5\%$, $\beta_1 = 1$, $\beta_2 = 10(\varphi_{max} - \varphi_{min})/t_{max} = 8.33 \times 10^{-2}$, $\varepsilon = 10^{-6}$, and $M = 15$. For the initial iteration, $N_0 = 1$, and the values of the B-spline coefficients were $f_{-1}^0 = f_0^0 = 0.05$, and $f_1^0 = f_2^0 = 0.275$. The number of iterations was $K = 7$. It should be noted that in the chromatograms shown in Figs. 4–7, the splines are depicted together with the optimized multi-linear gradients.

Fig. 4 depicts the chromatograms and associated gradients at the start of the process ($\kappa = 0$) (Fig. 4a), and the optimal values after iterations $\kappa = 3$, 4 and 5 (Fig. 4b–d). The last iteration, $\kappa = K = 7$ (i.e., the optimal solution) is shown in Fig. 5a. As observed, the amino acids were almost fully resolved in the established $t_G = 30$ min.

The calculation time for finding the optimal multi-linear gradient is around one hour. This is not detrimental because the algorithm is only run once to get the best separation. The found solution is global, which means that it can be considered the true optimal solution for the problem.

4.2. Penalization of steep slopes

The proposed optimization algorithm includes a penalization term to avoid steep slopes close to peak appearance (Eq. (8)). It should be noted that the optimal gradient is nearly flat in the regions where groups of compounds are eluting (Fig. 5a). Hence, the major changes in mobile phase composition occur in empty regions. This is an advantage considering the stability of the solution.

Fig. 5b shows the result of eliminating the Ψ_i term (the penalization function for steep slopes) from Eq. (9). As observed, the resolution is not compromised; it is even improved although with a change in the elution order of peaks 7–9. The most outstanding difference between the gradients in Fig. 5a and b is found in the last segment around 22 min, which takes place while compounds 11 and 12 are eluting. However, apparently, there are no significant

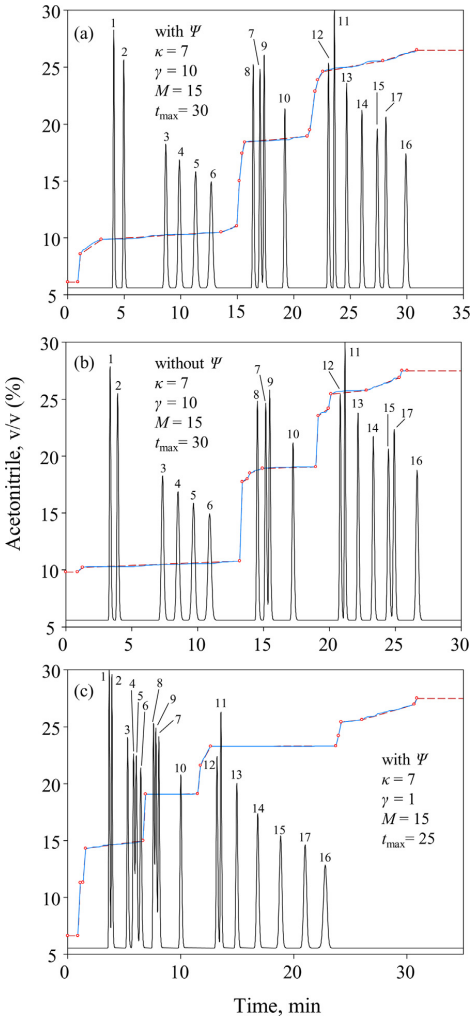


Fig. 5. Optimal gradient for a target gradient time of 30 min, including the penalization of the steepness term Ψ in Eq. (9) (a, c), and eliminating Ψ (b). The value of the γ coefficient, which prioritizes the maximization of the resolution of critical peaks, was $\gamma = 10$ in (a) and (b), and $\gamma = 1$ in (c). Other running parameters are overlaid.

consequences in the elution of the group of compounds 11–17. For other samples, the role of the penalization function can be more dramatic.

4.3. Influence of the weighting parameter γ

The parameter γ used in Eq. (9) emphasizes the weight of incomplete resolution in the F function. In order to illustrate the benefits associated to parameter γ in the optimization, the multi-scale algorithm was again run with the same parameters as in Section 4.1, but with $\gamma = 1$ instead of $\gamma = 10$. As shown in Fig. 5c, the resolution found as optimal is poorer with respect to the chromatogram shown in Fig. 5a. The reason is that smaller γ values reduce the importance of poor resolution. On the other hand, the analysis time was shorter than the allowed maximal value. Therefore, we are not

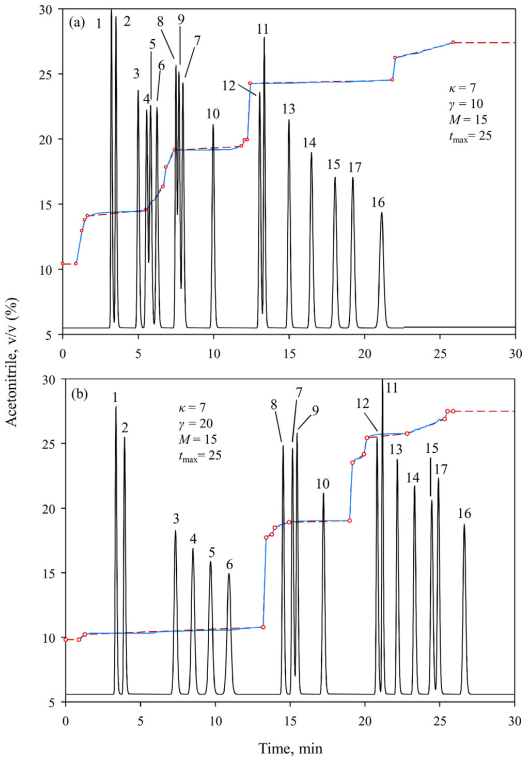


Fig. 6. Optimal gradient for a target analysis time of 25 min, including the penalization of the steepness term Ψ in Eq. (5); $\gamma = 10$ in (a), and $\gamma = 20$ in (b). Other running parameters are overlaid.

taking advantage of the available separation space properly, which arrives to 30 min. The inconveniently short analysis time gives rise to larger overlapping. This problem was satisfactorily solved by setting $\gamma = 10$ (see Fig. 5a). Note that a worst value of individual resolution $r_i = 0.95$ would yield with $\gamma = 10$, $r_i^{\gamma} = 0.95^{10} \approx 0.6$ (see Eq. (6)). This is translated in a stronger penalization for critical peaks of low resolution.

If the maximal allowed analysis time were reduced to a smaller value, such as 25 min, and the value $\gamma = 10$ is kept, we would find similar issues to those in Fig. 5c (see Fig. 6a). Nevertheless, simply by setting $\gamma = 20$, the critical peaks get adequate resolution (Fig. 6b). As observed, an adequate balance of the penalization and weighting parameters guide the search to the type of desired features for the solution, in terms of resolution and analysis time.

4.4. Reduction of the number of nodes

The results above were obtained with $M = 15$. Naturally, this is an inconveniently high value and the results should be considered as ideal or exploratory. The point is finding out similar resolution with a more reduced number of nodes, M. Fig. 7 shows the optimal separation found for $M = 7$ and $M = 5$, using an analysis time of 30 min. As observed in Fig. 7a, the peaks are still resolved for $M = 7$, in spite of the strong reduction in the number of nodes. This is not the case for $M = 5$ (Fig. 7b), where the overlaps become significant. As observed, the difference between the splines and the

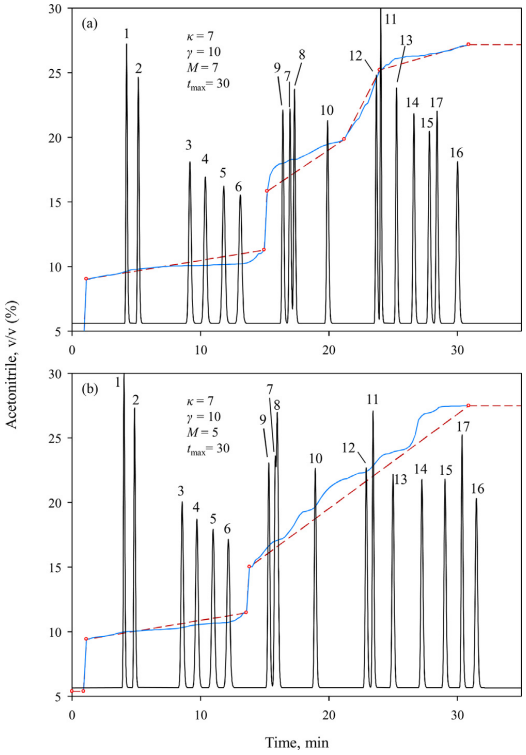


Fig. 7. Optimal gradients for a target analysis time of 30 min, obtained with a smaller number of nodes describing the multi-linear gradient (compare with Fig. 6a). Number of nodes: (a) $M = 7$, and (b) $M = 5$. Other running parameters are overlaid.

multi-linear gradients is noticeable. Note that this difference was negligible in Figs. 5 and 6.

5. Conclusions

The use of linear gradients is the simplest solution for attenuating the consequences of large differences in hydrophobicity of analytes in RPLC. However, maximal flexibility is obtained using multi-linear gradients, where g is a piecewise function composed by first degree polynomials. In any case, the gradients must be feasible to be implemented in a chromatographic system. We should indicate that although the framework is RPLC, the multi-scale optimization is valid for other chromatographic modes requiring gradient elution. Also, it may be useful to optimize other types of gradients, such as temperature, pH and salt concentration gradients.

This work reports the development of a mathematical procedure to calculate the optimal multi-linear gradient under a global perspective. In contrast to previous methods, here the level of detail in the solution is increased along the search using subdivision schemes. The solution found can be considered as the best possible attending the constraints set for guiding the search. We have checked that independent optimizations starting from different initial estimates converge into the same solution. Also, cubic splines can be adapted to any kind of function, and therefore, they are better intermediaries to define any arbitrary solvent variation function, in comparison to the straightforward use of a polygonal function. In any case, the resolution expectancies are calculated always from the polygonal approximations.

Some years ago, we developed a stepwise search using genetic algorithms for resolving the same problem [10]. That approach was applied to get the chromatograms shown in Figs. 4S–6S in the Supplementary material. In that report, the complexity of the gradients was increased gradually and the search domain modified considering the results found with less complex gradients, favoring an optimal profit of the search. The codification system for the time domain promoted an even distribution of nodes, and a good adaptation level to the local complexity requirements of the different solutes. The new approach reported in this work establishes the optimal configuration in an automatic way and leads to a finer tuning of the nodes position, due to the use of the multi-scale technique. The implementation of such gradients has the interest of accommodating the elution to shorter times. Moreover, it is able to make a better profit of the results found in previous less complex gradients. The practical consequence of all these features is the capability of finding the really best optimal gradient with high accuracy.

Several improvements reported in this work can be used in other search strategies for chromatographic optimization: (i) the global resolution function in Eq. (6); (ii) the formulation of the constraints that modify the suitability of the solutions (Eqs. (7) and (8)); (iii) the use of cubic splines for defining gradients, together with the methodology for simplifying them to define multi-linear gradients (Section 2.5); and particularly, (iv) the use of the multi-scale optimization technique (Section 2.7), which have been applied in this work for the first time in chromatography.

Some straightforward modifications can be proposed. Thus, the pattern-search algorithm can be replaced by other maximization function, such as simulated annealing or genetic algorithms. Also, the B-splines, which were used to model the gradients owing to their bonding and monotonicity properties, can be replaced by other techniques, such as Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) interpolation that present the same good properties. Finally, it is possible to replace the way of combining the constraints with other multi-criteria decision-making functions, or carry out the simultaneous optimization of the chromatographic resolution and analysis time (and other secondary aims), using Pareto plots or other tools.

It should be observed that the application of the multi-scale optimization strategy to the resolution of highly complex samples, requiring a very specific gradient, would be limited by the accuracy obtained in the predictions of simulated chromatograms. Very complex gradients make more visible the uncertainties associated to the retention models. This also may compromise the possibility to fully validate chromatographic methods using such complex gradient profiles, since it raises concerns about repeatability and reproducibility. Therefore, for highly complex samples and complex gradients, the modeling of retention times and peak profiles may become the limiting factor. In contrast, for simple samples, there will be enough separation space for trying alternative solutions, perhaps less critical. Indeed, the most important application of the approach may be the reduction in terms of analysis time that can be achieved for relatively simple samples. Nevertheless, the merit of the proposed approach of finding the truly optimal solution, independently of the starting point and gradient complexity, should not be neglected.

Another issue that deserves be mentioned is the feasibility of the implementation of the prescribed gradients in the HPLC equipment. Such problem is extensively discussed in Ref. [4]. Very complex gradients have the drawback of the sensitivity of the resolution level to small changes in the gradient program, often due to small errors in the setting of the gradient in the instruments. Also, very

complex gradients cannot be accurately set in old instruments, due to gradient rounding at the node changes.

To sum up, this work recommends an optimization strategy to find the truly optimal solution, provided the retention models are accurate. If this is not the case, more perfect models would give response to this insufficiency. In other words, the possible discrepancies between the results found by the search procedure (which is the aim of this work) are not produced by the optimization strategy, but should be ascribed to the quality of the predictions, which can be improved independently.

Finally, the methodology described is general and valid for any kind of sample (i.e., there is no restriction concerning the sample nature). Naturally, the more complex the sample, the more the need for a powerful optimization strategy, like that reported in this work. The optimization of a multi-linear gradient is a main solution to find good selectivity by itself, or in combination with other separation strategies, such as the use of serially-coupled columns [15,22,23], or two-dimensional liquid chromatography [24].

Acknowledgements

This work was supported by Projects CTQ2016–75644-P and MTM2014–54388 (Ministry of Economy, Industry and Competitiveness, MINECO, Spain, and FEDER funds), and PROMETEO/2016/128 (Direcció General d'Universitat, Investigació i Ciència, Generalitat Valenciana, Spain). Sergio López-Ureña thanks the Ministry of Education and Culture and Sports, MECD of Spain for the FPU14/02216 grant.

Appendix A. Supplementary data

Supplementary material related to this article can be found, in the online version, at doi:https://doi.org/10.1016/j.chroma.2017.12.040.

References

[1] P. Jandera, J. Churáček, Gradient Elution in Column Liquid Chromatography, vol. 31, Elsevier Science, Amsterdam, 1985.
[2] L.R. Snyder, J.J. Kirkland, J.L. Glajch, Practical HPLC Method Development, 2nd ed., John Wiley & Sons, 1997.
[3] P. Jandera, Can the theory of gradient liquid chromatography be useful in solving practical problems? J. Chromatogr. A 1126 (2006) 195–218.
[4] L.R. Snyder, J.W. Dolan, High-Performance Gradient Elution, John Wiley and Sons, Hoboken, NJ, 2007.
[5] J.W. Dolan, L.R. Snyder, Theory and practice of gradient elution, in: S. Fanali, P. Haddad, C.F. Poole, P.J. Schoenmakers, D. Lloyd (Eds.), Liquid Chromatography: Fundamentals and Instrumentation, Elsevier, Amsterdam, 2013, pp. 269–282.
[6] J.J. Baeza-Baeza, M.C. García-Alvarez-Coque, Some insights on the description of gradient elution in reversed phase-liquid chromatography, J. Sep. Sci. 37 (2014) 2269–2277.
[7] R. Cela, M. Lores, PREOPT-W: a simulation program for off-line optimization of binary gradient separations in HPLC. I. Fundamentals and overview, Comput. Chem. 20 (1996) 175–191.
[8] J. García-Lavandeira, P. Oliveri, J.A. Martínez-Pontevedra, M.H. Bollaín, M. Forina, R. Cela, Computer-assisted modeling and optimization of reversed-phase high-temperature liquid chromatographic (RP-HTLC) separations, Anal. Bioanal. Chem. 399 (2011) 1951–1964.
[9] P. Nikitas, A. Pappa-Louisi, K. Papachristos, Optimisation technique for stepwise gradient elution in reversed-phase liquid chromatography, J. Chromatogr. A 1033 (2004) 283–289.
[10] V. Concha-Herrera, G. Vivó-Truyols, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Limits of multi-linear gradient optimisation in reversed-phase liquid chromatography, J. Chromatogr. A 1063 (2005) 79–88.
[11] P. Nikitas, A. Pappa-Louisi, New approaches to linear gradient elution used for optimization in reversed-phase liquid chromatography, J. Liq. Chromatogr. Relat. Technol. 32 (2009) 1527–1576.
[12] P. Nikitas, A. Pappa-Louisi, Ch. Zisi, Multilinear gradient elution optimization in liquid chromatography, Adv. Chromatogr. 52 (2015) 79–116.
[13] L.R. Snyder, J.W. Dolan, D.C. Lommen, Drylab® computer simulation for high-performance liquid chromatographic method development: II. Gradient elution, J. Chromatogr. 485 (1989) 91–112.
[14] J.W. Dolan, Gradient elution, Part I: Intuition, LC GC North America 31 (2013) 204, 206, 208, 209.
[15] C. Ortiz-Bolsico, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Optimization of gradient elution with serially-coupled columns. Part II: multi-linear gradients, J. Chromatogr. A 1373 (2014) 51–60.
[16] E. Tyteca, K. Vanderlinden, M. Favier, D. Clicq, D. Cabooter, G. Desmet, Enhanced selectivity and search speed for method development using one-segment-per-component optimization strategies, J. Chromatogr. A 1358 (2014) 145–154.
[17] R. Donat, S. López-Ureña, M. Menec, A novel multi-scale strategy for multi-parametric optimization, in: P. Quintela, P. Barral, D. Gómez, F.J. Pena, J. Rodríguez, P. Salgado, M.E. Vázquez-Méndez (Eds.), Progress in Industrial Mathematics at ECMI 2016, Springer, Berlin, 2017.
[18] S. Carda-Broch, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Evaluation of several global resolution functions for liquid chromatography, Anal. Chim. Acta 396 (1999) 61–74.
[19] T. Alvarez-Segura, A. Gómez-Díaz, C. Ortiz-Bolsico, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, A chromatographic objective function to characterise chromatograms with unknown compounds or without standards available, J. Chromatogr. A 1409 (2015) 79–88.
[20] J.A. Navarro-Huerta, T. Alvarez-Segura, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Study of the performance of a resolution criterion to characterise complex chromatograms with unknowns or without standards, Anal. Methods 9 (2017) 4293–4303.
[21] N. Dyn, Subdivision schemes in computer-aided geometric design (CAGD), in: W. Light (Ed.), Advances in Numerical Analysis, Vol. II, Wavelets, Subdivision Algorithms and Radial Basis Functions, Clarendon Press, Oxford, 1992, pp. 36–104.
[22] T. Alvarez-Segura, C. Camacho-Molinero, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Analysis of amino acids using serially-coupled columns, J. Sep. Sci. 40 (2017) 2741–2751.
[23] T. Alvarez-Segura, J.R. Torres-Lapasió, C. Ortiz-Bolsico, M.C. García-Alvarez-Coque, Stationary phase modulation in liquid chromatography through the serial coupling of columns, Anal. Chim. Acta 923 (2016) 1–23.
[24] P. Jandera, Comprehensive two-dimensional liquid chromatography: practical impacts of theoretical considerations, Cent. Eur. J. Chem. 10 (2012) 844–875.

Glossary

β_1: Parameter used in the Φ penalization function to prevent long analysis times in $R(g)$
β_2: Parameter used in the Ψ penalization function to prevent steep slopes in $R(g)$ close to peak elution
γ: Weighting parameter that penalizes incompletely resolved peaks in $R(g)$
Δ_n^l: Difference between an intermediate value in the polygonal approximation and the corresponding spline value v_n^l (Eq. (13))
ε: Weighting parameter that prevents null values in $R(g)$
κ: Current scale inside the multi-scale optimization process
K: Last iteration of the multi-scale optimization technique
v_n^l: Concentration value of the spline s at time τ_n^l
π_n: Straight-line that connects the points in the spline at times τ_{n-1}^l and τ_{n+1}^l
τ_n^l: Inside the operator $P_M(s)$, the n-th time coordinate of the multi-linear gradient with l nodes
φ: Organic solvent concentration in the mobile phase
$\varphi(t)$: Gradient program
φ_{max}: Maximal concentration of organic solvent in the experimental design
φ_{min}: Minimal concentration of organic solvent in the experimental design
$\Phi i(g)$: Penalization function preventing long analysis times, governed by β_1
$\Psi i(g)$: Penalization function preventing steep slopes close to peak elution, governed by β_2
A_i: Left half-widths
b: B-Spline basis
B_i: Right half-widths
$c(g)$: Global chromatographic signal
$c_i(g)$: Individual chromatographic signal associated to solute i
$F(g)$: Objective function used in the optimization
f_n: Coefficients determining the heights of a series of bell-shaped functions used to define the spline
$\hat{f}_n$: Optimal values that determine the expression of the best multi-linear gradient $\hat{g}$
$g(t)$ or g: Multi-linear gradient
$\hat{g}$: Best multi-linear gradient
g': Slope or derivative of g
h: Distance between nodes in the cubic spline
$k_i(\varphi)$: Solute retention factor at composition φ
I: Inside the operator $P_M(s)$, the current number of nodes of the multi-linear gradient
L: Inside the operator $P_M(s)$, the starting number of nodes of the multi-linear gradient
M: Number of nodes in the multi-linear gradient
n: Dummy index variable
n^*: Node giving the minimal difference Δ_{n*}^l
N: Number of spline nodes
N_0: Small arbitrary number of nodes to start the iterations in the multi-scale optimization
N_κ: Number of nodes for scale κ

NAC: *N*-Acetylcysteine
OPA: *o*-Phthalaldehyde
p: Number of analytes
$P_M(s)$*:* Operator used to simplify the spline to a multi-linear function of *M* nodes
PCHIP: Piecewise Cubic Hermite Interpolating Polynomial
Q: Projection operator
$(Q(f))_{n=-1}^{N+1}$*:* Sequence fulfilling the constraints in the multi-scale optimization
$r_i(g)$*:* Individual resolution associated to solute *i*
R(g): Global resolution for gradient *g*
R_S*:* Snyder resolution
RPLC: Reversed-phase liquid chromatography
s(t) or s: Cubic spline
t: Time
t_0*:* Dead time
t_G*:* Gradient time
$t_{g,i}$*:* Retention time of solute *i* under gradient elution
$t_{R,i}$*:* Retention time for solute *i*
t_{max}*:* Maximal analysis time
t_{dw}*:* Dwell time
t_n*:* Time location of spline nodes

Journal of Chromatography A, 1600 (2019) 137–147

Contents lists available at ScienceDirect

Journal of Chromatography A

journal homepage: www.elsevier.com/locate/chroma

Enhancement in the computation of gradient retention times in liquid chromatography using root-finding methods

S. López-Ureña [a], J.R. Torres-Lapasió [b,*], M.C. García-Alvarez-Coque [b]

[a] Department of Mathematics, Faculty of Mathematics, Universitat de València, c/Dr. Moliner 50, 46100 Burjassot, Spain
[b] Department of Analytical Chemistry, Faculty of Chemistry, Universitat de València, c/Dr. Moliner 50, 46100 Burjassot, Spain

ARTICLE INFO

Article history:
Received 28 February 2019
Received in revised form 10 April 2019
Accepted 11 April 2019
Available online 25 April 2019

Keywords:
Reversed-phase liquid chromatography
Multi-linear gradients
Fundamental equation for gradient elution
Integration
Root-finding methods

ABSTRACT

Gradient elution may provide adequate separations within acceptably short times in a single run, by gradually increasing the elution speed. Similarly to isocratic elution, chromatograms can be predicted under any experimental condition, through strategies based on retention models. The most usual approach implies solving an integral equation (i.e., the fundamental equation of gradient elution), which has an analytical solution only for certain combinations of retention model and gradient programme. This limitation can be overcome by using numerical integration, which is a universal approach although at the cost of longer computation times. In this work, several alternatives to improve the performance in the resolution of the integral equation are explored, which can be especially useful with multi-linear gradients. For this purpose, the application of several root-finding methods that include the Newton's and bisection searches is explored in three frameworks: isolated predictions, regression modelling problems using gradient training sets, and optimisation of multi-linear gradients. Significant reductions of computation times were obtained. The substitution of non-integrable retention models by Tchebyshev polynomial approximations, which are pre-calculated before solving the integral equation in optimisation problems, is also investigated.

1. Introduction

In reversed-phase liquid chromatography, isocratic elution is often inadequate to satisfactorily resolve complex samples. If the most retained solutes were eluted at sufficiently short retention times, the least retained would give peaks poorly resolved, or even lost in the solvent front. On the contrary, if the least retained solutes are intended to be well resolved, then the most retained would be eluted at excessively long times. The usual solution to this problem is the application of gradient elution with programmed changes in the elution conditions, mostly the concentration of organic modifier [1–4]. Gradient elution may provide adequate resolution within acceptably short times during a single run, by increasing the retention of the poorly retained solutes and speeding up the elution of those strongly retained. For this purpose, the elution strength of the mobile phase should be initially low, and become gradually stronger by increasing the concentration of the modifier as the separation progresses.

In either isocratic or gradient elution, finding the best separation conditions for complex mixtures is not easy, especially when trial-and-error strategies are used. Also, there are many situations that can only be tackled through the use of the so-called interpretive strategies, which are based on the accurate description of the retention behaviour. By using models, it is possible to forecast how chromatograms will be like under any experimental condition within a reasonable time domain. Repeating the prediction process for a number of conditions, the chromatographer can arbitrarily select which are the most promising conditions to separate the compounds in a sample, or alternatively use optimisation software to explore more rigorously the conditions providing maximal resolution.

Solving the problem in isocratic elution is rather simple, but this is not the case in gradient elution, where the complexity is much higher. The most usual approach implies solving an integral equation, which has an analytical solution only for some combinations of retention model and gradient programme. Frequently, the integral even cannot be solved. These limitations can be overcome using numerical integration, which is a universal approach [5,6].

When multi-linear gradients are applied, obtaining the gradient elution time by an algebraic expression is not possible [7,8]. The piece-wise nature of these gradients leads to piece-wise expres-

* Corresponding author.
E-mail address: jrtorres@uv.es (J.R. Torres-Lapasió).

https://doi.org/10.1016/j.chroma.2019.04.030
0021-9673/

sions, where the solution of the integral at a given segment depends on the solution of the previous one. This type of gradient is of maximal practical interest, since the retention times are considerably shortened, while the resolution of intermediate compounds is maintained, outperforming simple linear gradients.

The selection of the best separation conditions using multi-linear gradients implies massive calculations, which are exponentially increased with the complexity of the gradient programme. For complex gradients, even simple numerical integration methods involve long computation times. Highly efficient optimisation procedures based on natural computation (e.g., genetic algorithms [9]) are unable to yield acceptable times, except if a comprehensive exploration level is avoided. Complex separations require local adaptations of the gradient to the needs of each solute. Maximal exploitation of a gradient instrument thus requires being able to accommodate the elution programme to the features of the sample, which usually implies a high number of segments along the gradient programme. Due to the limitations involved in the optimisation of gradients, chromatographers usually accept basic designs as valid solutions, although these are not able to fully exploit the separation capabilities of their instrument. It should be noted that the alteration of a segment in a multi-linear gradient affects, in a complex fashion, the separation of compounds eluting later, which makes any sequential optimisation impossible, except in simple cases [10]. Global searching strategies have shown better exploration capability, which gives more opportunities to find a good separation [11].

Throughout the years, several authors have been working in the field of chromatographic optimisation, taking benefit from the constant improvements in the computation speed [12–24]. This has made possible to tackle problems that few years ago were out of the question. In addition to the improvement facilitated by modern computers, it is also possible to improve the performance of optimisations by using more efficient search algorithms, or by increasing the speed of the most critical step (i.e., that one involving most of the calculation volume).

In previous work [11], the optimisation of gradient programmes to improve the resolution of chromatographic peaks was carried out by applying a multi-scale strategy, based on subdivision schemes. This is a very common technique applied in Computer Aid Geometric Design, due to its efficiency and easy implementation. During an optimisation, multiple simulations of retention times are required (several thousands of times). It is, thus, very important to implement fast simulation codes.

We have being considering other alternatives to reduce the computation of retention times in gradient elution, especially for multi-linear programmes. Initially, we considered strategies based on root-finding algorithms [25], for retention models that have primitive. For those situations where a primitive cannot be found, we propose the use of approximations of the retention models, obtaining thus a universal method to estimate retention times, useful for all situations.

2. Theory

2.1. Gradient programmes and calculation of retention times

Let us denote a gradient programme of organic solvent as a function of time, $\varphi(t)$, which is valid for any time $t \geq 0$. If $\varphi(t)$ is a constant function, then the chromatographic run is called isocratic, while if $\varphi(t)$ is a straight-line, then it is a linear gradient. More complex programmes can be used as well, such as multi-linear gradients, where $\varphi(t)$ is continuous and piecewise-defined with polynomials of degree one. The theory here presented is valid for a general $\varphi(t)$ programme, but later we will focus on the use of multi-linear gradients. For any gradient, the delay from the mixer to the column inlet (i.e., the dwell time) must be considered; henceforth, any gradient programme starts with an isocratic step in an initial region $[0, t_{dwell}]$ (see Fig. 1 for the meaning of the variables used in this work).

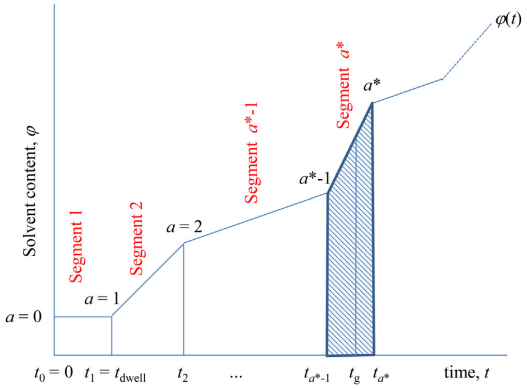


Fig. 1. Graphical definition of the variables involved in a multi-linear gradient, composed of several consecutive segments between the a nodes. The solute leaves the column inside the shaded region.

In order to model the migration process under gradient conditions, it is convenient to define the retention factor in differential form [26]:

$$k = \frac{dt_s}{dt_m} \tag{1}$$

where k is the retention factor in each elementary segment in which the column has been imaginarily split, t_s is the residence time in the stationary phase, and t_m, the time spent by the solute in the mobile phase. Rearranging Eq. (1), the following is obtained:

$$dt_m = \frac{dt_s}{k} \tag{2}$$

This equation can be integrated, leading to the so-called "fundamental equation for gradient elution":

$$\int_0^{t_0} dt_m = \int_0^{t_g - t_0} \frac{dt_s}{k(t)} \tag{3}$$

where t_0 is the dead time and t_g is the retention time in gradient conditions. Taking into account that the residence time outside the column (t_{ext}) does not affect the intra-column migration process (it should be subtracted from t_0), and that the retention factor depends on the solvent content in gradient elution, the following expression is finally obtained:

$$t_0 - t_{ext} = \int_0^{t_g - t_0} \frac{1}{k(\varphi(t))} dt \tag{4}$$

The correction of the extra-column contributions requires redefining the retention factor as:

$$k = \frac{t_R - t_0}{t_0 - t_{ext}} \tag{5}$$

t_R being the retention time in isocratic elution. Observe that t_0 and k depend on the organic solvent content during the gradient programme, $\varphi(t)$. In Eq. (4), the dead time can be considered constant:

$$t_0(\varphi) = c_0 \tag{6}$$

or be a function of φ, owing to the changes in viscosity induced by the gradient programme:

$$t_0(\varphi) = c_0 + c_1\varphi + c_2\varphi^2 \tag{7}$$

Some examples of equations relating k with the solvent content φ are the logarithmic-quadratic model (Eq. (8)) [27], the equation proposed by Nikitas (Eq. (9)) [28], and the Neue-Kuss model (Eq. (10)) [29]:

$$k(\varphi) = \exp(d_1 + d_2\varphi + d_3\varphi^2) \tag{8}$$

$$k(\varphi) = \exp\left(d_1 - \frac{d_2\varphi}{1 + d_3\varphi} + d_4\varphi\right) \tag{9}$$

$$k(\varphi) = d_1(1 + d_2\varphi)^2 \exp\left(-\frac{d_3\varphi}{1 + d_2\varphi}\right) \tag{10}$$

where d_i are model parameters, which are characteristic of each solute, and should be fitted using an adequate training set of isocratic or gradient data, following an appropriate experimental design [30]. Provided the gradient programme is linear, the inclusion of Eq. (8) in Eq. (4) can be solved analytically, although the solution depends on the error function [31]. If Eq. (9) is used instead, the resulting integral in Eq. (4) cannot be computed analytically, whereas with Eq. (10) it can be. In all these cases (Eqs. (8)–(10)), it is not possible to get a closed expression for the retention time. Hence, a numerical approach is needed [5].

A universal integration method for Eq. (4) consists of discretising the gradient in a high number of isocratic steps, and computing the cumulative sum $\Delta t/k(\varphi(t))$ up to matching $t_0 - t_{ext}$. At this point, the solute leaves the column. The time value along the gradient programme when this condition is fulfilled corresponds to t_g.

2.2. Use of root-finding methods

As commented, the calculation of t_g implies working out this variable from Eq. (4), which is only possible in very limited cases. However, the universal integration method explained above has the inconvenience of being slow, particularly when an accurate solution is required. In this work, we check faster alternatives necessary for massive calculations, such as those carried out during the optimisation of multi-linear gradients [7,8,18,32,33].

Eq. (4) can be rearranged, so that the search value t_g is obtained as the root of the following function:

$$F(t) = t_0(\varphi(t)) - t_{ext} - \int_0^{t - t_0(\varphi(t))} \frac{1}{k(\varphi(t))} dt \tag{11}$$

where t is the integration variable, which will become $t = t_g$ exactly when the solute leaves the column ($F(t_g) = 0$). In Eq. (11), the dependence of the dead time with viscosity (Eq. (7)) is considered. The calculation of gradient retention times (t_g) can be implemented as the zero value of Eq. (11), which can be found by applying any root-finding algorithm. This outline allows as many ways to obtain t_g as root-finding algorithms exist. The next paragraphs show the adaptation of four of these algorithms for solving the fundamental equation for gradient elution (Eq. (4)):

(i) The first approach, which is included only for comparison purposes, is quite straightforward and consists in evaluating $F(t)$ successively, with a fixed step of size h: $F(0)$, $F(h)$, $F(2h)$, . . . The $F(t)$ values that constitute this sequence are initially positive, and decrease with the number of included h terms. Beyond a certain value, $F(nh)$ becomes negative. Accordingly, $F(t) > 0$ when the solute is still inside the column, $F(t) < 0$ if it has been eluted, and $F(t) = 0$ if it is right at the column outlet. The first negative value will be referred as n^*h. The search consists in finding this first n^* value, where $F(n^*h) \leq 0$. This means that the solution will be between the last positive value and the first negative one, $(n^* - 1)h < t_g \leq n^*h$ (Bolzano's theorem). Note that this algorithm can be particularly slow when an accurate estimation of t_g is wished, since h must be as small as the desired accuracy. It is convenient to start the discussion with this method because it has already been used in the interpretive optimisation of complex gradients [6,20,23], and it is conceptually the most intuitive.

(ii) The second approach makes use of the bisection method. This is much faster than the direct method, and convergence to the solution is always guaranteed.

(iii) The third approach searches the root of Eq. (11) by applying the Newton's method, with some adaptations for safe computation. This root-finding method is even faster than the bisection method, but demands the fulfilment of some conditions for properly work, which are no trivial and can make the search slow or even fail.

(iv) The fourth approach combines both the Newton's and bisection methods to achieve a fast and always-convergent method.

2.3. Considerations for the application of root-finding methods

In order to find t_g, the $F(t)$ function (Eq. (11)) will be evaluated multiple times, until reaching the solution with the desired accuracy level. Each guess to get t_g, together with all the operations that must be carried out to define the new guess, constitute an iteration. We will denote the guess at the iteration n (being $n \geq 0$) by t^n (n is a superindex which indicates the number of steps or iterations in the methods explained below; it does not mean the operator power). In each iteration, the value $F(t^n)$ is computed to check whether t^n is the solution (or close enough to it). Hence, the definition of $F(t)$ should be carefully analysed. In our case, it should be noted that $t - t_0(\varphi(t)) \geq 0$ is required at the upper limit of the integral in Eq. (11). For the sake of simplicity, let us denote the upper limit of the integral by $A(t) = \max\{t - t_0(\varphi(t)), 0\}$ (the lower limit will be set as $A(0) = 0$). Thus, $F(t)$ can be redefined as:

$$\begin{aligned} F(t) &= t_0(\varphi(t)) - t_{ext} - \int_0^{\max\{t - t_0(\varphi(t)), 0\}} \frac{1}{k(\varphi(t))} dt \\ &= t_0(\varphi(t)) - t_{ext} - \int_{A(0)}^{A(t)} \frac{1}{k(\varphi(t))} dt \end{aligned} \tag{12}$$

The dead time should be properly calculated in order to get an invertible $A(t)$ function. This treatment allows computing $F(t)$ safely, for any $t \geq 0$. This trick (i.e., using a maximal value as upper limit of the integral, so that it will be always zero or positive) does not affect negatively our purpose, because for sure the retention time for any solute is always larger or equal to the dead time ($t_g - t_0(\varphi(t_g)) \geq 0$).

If there is a primitive function, $P(t)$, associated to the integration of $1/k(\varphi(t))$, the $1/k(\varphi(t))$ function is qualified as integrable. This integral will be easily computed using the Barrow's rule:

$$F(t) = t_0(\varphi(t)) - t_{ext} - [P(A(t)) - P(A(0))] \tag{13}$$

First, we will describe approaches that require integrable functions, but later their application to non-integrable functions will be considered (Section 4). Observe that the integrability strongly depends on the mathematical definition of the gradient programme (i.e., the changes in $\varphi(t)$). If the run is isocratic, $1/k(\varphi(t))$ will be always integrable, because it is a constant function. However, if $\varphi(t)$ varies linearly with time ($\varphi(t) = \alpha + \beta\, t$, which is defined as a simple linear gradient), its integration will depend on the combination of the retention model and the linear gradient (i.e., the nested function $k(\varphi(t))$).

Thus, for the Neue-Kuss equation (Eq. (10)), which is integrable, the following primitives are obtained:

$$P(t) = \int \frac{1}{k(\varphi(t))} dt = \begin{cases} \dfrac{\exp\left(\dfrac{d_3(\alpha+\beta t)}{1+d_2(\alpha+\beta t)}\right)}{\beta d_1 d_3} & \text{for} \quad \beta \neq 0 \\ \dfrac{t \exp\left(\dfrac{\alpha d_3}{1+\alpha d_2}\right)}{d_1(1+\alpha d_2)^2} & \text{for} \quad \beta = 0 \end{cases} \tag{14}$$

2.4. Case of multi-linear gradients

We will now focus on the general case of multi-linear gradients. Taking this into account, the methodology also allows the calculation for either isocratic or simple linear gradients by simplification, or gradients of larger complexity by assimilation of the gradient to a complex curve with a large number of consecutive linear segments. Some methods, such as the Newton's method, may have problems with multi-linear and assimilable gradients because $F(t)$ should be differentiable, but there is no derivative at the junction of two consecutive linear segments. To avoid this potential problem, a pre-processing step can be applied just before applying the root-finding algorithm. This strategy is strongly recommended for the Newton-based algorithms and described next.

Let us suppose that the multi-linear gradient have N segments, each of them defined by $\varphi(t) = \alpha + \beta\, t$. The nodes a (from 0 to N) define the boundaries between consecutive segments, being $a = 0$ the node at $t = 0$, and $[\tau_a,\ \tau_{a+1}[$ the time domain of each segment (the bracket at the right indicates that the upper extreme point, τ_{a+1}, belongs to the next segment $a+1$). Along this work, and for a more general treatment, the isocratic step associated to the dwell volume will be included as the first segment in the gradient (Fig. 1). From now on, we will consider the times t_a, such that $A(t_a) = \tau_a$ (in other words, $t_a = A^{-1}(\tau_a)$).

The calculation of the gradient retention time computes $F(t_a)$ with $a = 1, 2, 3, \ldots$, up to the first $a*$ node where $F(t_{a*}) < 0$ (Fig. 1). The algorithm starts by computing $F(t)$ at the end of the first segment ($A(t_1) = \tau_1 = t_{\text{dwell}}$). Usually, solutes with enough retention will elute after the dwell time, and accordingly $F(t_1) > 0$. However, very scarcely retained or no retained solutes may elute in the isocratic step associated to the dwell time. In this case, $t_g < t_1$, and $F(t_1) < 0$.

When the solute leaves the column beyond the dwell time, and according to the Bolzano's theorem, the retention time must be in the interval $[t_{a^*-1},\ t_{a^*}[$. For any node a, $F(t_a)$ is obtained as follows:

$$\begin{aligned} \mathrm{F}(t_a) &= t_0(\varphi(t_a)) - t_{\text{ext}} - \int_0^{\tau_a} \frac{1}{k(\varphi(t))} dt \\ &= t_0(\varphi(t_a)) - t_{\text{ext}} - \sum_{i=1}^{a} \int_{\tau_{i-1}}^{\tau_i} \frac{1}{k(\alpha_i + \beta_i t)} dt \\ &= t_0(\varphi(t_a)) - t_{\text{ext}} - \sum_{i=1}^{a} [P_i(\tau_i) - P_i(\tau_{i-1})] \end{aligned} \tag{15}$$

where P_i is the primitive of $1/k(\alpha_i + \beta_i t)$. Observe that the split-by-nodes treatment allows making the calculation of each step to depend on the results found in the previous step:

$$F(t_a) = F(t_{a-1}) + [t_0(\varphi(t_a)) - t_0(\varphi(t_{a-1}))] - [P_a(\tau_a) - P_a(\tau_{a-1})] \tag{16}$$

Taking into account that the solute leaves the column between t_{a*-1} and t_{a^*}, at a time t (i.e., the solution of the equation), such that $t_{a^*-1} \le t < t_{a^*}$, the final expression is:

$$\begin{aligned} F(t) = t_0(\varphi(t)) - t_{\text{ext}} - \int_0^{\tau_{a*-1}} \frac{1}{k(\varphi(t))} dt - \int_{\tau_{a*-1}}^{A(t)} \frac{1}{k(\alpha_{a*} + \beta_{a*} t)} dt = \\ t_0(\alpha_{a*} + \beta_{a*} t) - t_{\text{ext}} - \sum_{i=1}^{a*-1} [P_i(\tau_i) - P_i(\tau_{i-1})] \\ -[P_{a*}(A(t)) - P_{a*}(\tau_{a*-1})] \end{aligned} \tag{17}$$

Table 1
Root-finding algorithms to resolve the fundamental equation for gradient elution.

Algorithm 1: Pre-processing step
1. $a = 0$ // Interval index
2. $I = 0$ // Value of the integral
3. Do while $t_0(\varphi(t_a)) - t_{\text{ext}} - I > 0$
4. $a = a + 1$
5. $I = I + P_a(\tau_a) - P_a(\tau_{a-1})$
   Update the integral value
6. Loop
7. $I = I - P_a(\tau_a)$ // We found $a* = a$
8. Apply root-finding method to $F(t) = t_0(\alpha_a + \beta_a\, t) - t_{\text{ext}} - P_a(A(t)) - I$

Algorithm 2: Direct method
1. $n = 0$
2. $\hat{t}^0 = t_{a*} - 1$
3. Do while $F(\hat{t}^n) \ge 0$
4. $n = n + 1$
5. $\hat{t}^n = \hat{t}^{n-1} + h$
6. Loop

Algorithm 3: Bisection method
1. $n = 0$
2. $t_{\text{left}} = t_{a*-1}$, $t_{\text{right}} = t_{a*}$
3. Do while $(t_{\text{right}} - t_{\text{left}})/2 > h$
4. $n = n + 1$
5. $\hat{t}^n = (t_{\text{left}} + t_{\text{right}})/2$
6. If $F(\hat{t}^n) \ge 0$, $t_{\text{left}} = \hat{t}^n$, otherwise $t_{\text{right}} = \hat{t}^n$
7. Loop

Algorithm 4: Newton's method
1. $n = 0$
2. $\hat{t}^0 = (t_{a*-1} + t_{a*})/2$
3. Do
4. ratio $= \frac{F(\hat{t}^n)}{F'(\hat{t}^n)}$
5. $n = n + 1$
6. $\hat{t}^n = \hat{t}^{n-1} -$ ratio
7. Loop while $|\text{ratio}| > h$

Algorithm 5: Newton-bisection method
1. $n = 0$
2. $t_{\text{left}} = t_{a*-1}$, $t_{\text{right}} = t_{a}*$
3. $t^0 = (t_{a*-1} + t_{a*})/2$
4. Do
5. ratio $= \frac{F(\hat{t}^n)}{F'(\hat{t}^n)}$
6. $n = n + 1$
7. $\hat{t}^n = \hat{t}^{n-1} -$ ratio
8. If $\hat{t}^n < t_{\text{left}}$ or $\hat{t}^n > t_{\text{right}}$ or $F'(\hat{t}^n) = 0$, $\hat{t}^n = (t_{\text{left}} + t_{\text{right}})/2$
9. If $F(\hat{t}^n) > 0$, $t_{\text{left}} = \hat{t}^n$, otherwise $t_{\text{right}} = \hat{t}^n$
10. Loop while $|\text{ratio}| > h$

Observe that the terms associated to the former nodes (i.e., those in the summation term) do not depend on t, but only on the coordinates of the nodes. Therefore, this sum is a constant, which was already computed when the former node was inspected. Once the interval $[t_{a^*-1},\ t_{a^*}[$ that contains the retention time is known, t_g can be found by applying a proper root-finding algorithm to $F(t)$. Therefore, the search of t_g for a multi-linear gradient is reduced to the linear gradient in the segment where the solute leaves the column. The pre-processing step is summarised in Table 1 (Algorithm 1).

3. Implementation of the root-finding methods

3.1. Direct method

Since $F(t_{a^*-1}) \ge 0$, the first strategy that we present to compute t_g consists in finding the time $\hat{t}^n = t_{a*-1} + nh$ where $F(\hat{t}^n) < 0$ is fulfilled. This would mean that t_g belongs to the interval $[\hat{t}^{n-1}, \hat{t}^n[$, which provides an approximation to t_g. Observe that the smaller the h value, the greater the accuracy in the computation of t_g. This method can be considerably slow, because it requires around $(t_g - t_{a^*-1})/h$ iterations. The advantage is that it will always grant convergence to the solution. The corresponding code of a simple algorithm to compute the retention time in gradient elution, t_g, is listed in Table 1 (Algorithm 2).

3.2. Bisection method

This method requires setting two times for starting, t_{left} and t_{right} (which are smaller and larger than the solution), such that $F(t_{\text{left}}) \ge 0$ and $F(t_{\text{right}}) < 0$. In our case, $t_{\text{left}} = t_{a^*-1}$ and $t_{\text{right}} = t_{a^*}$. According to the Bolzano's theorem, $F(t)$ must be zero at a certain time inside $[t_{\text{left}}, t_{\text{right}}[$. We can start by checking if the root value of $F(t)$ is in the middle of the interval ($\hat{t}^n = (t_{\text{left}} + t_{\text{right}})/2$). If $F(\hat{t}^n) \ge 0$,

then the root must be in $[\hat{t}^n, t_{right}[$, otherwise it is in $[t_{left}, \hat{t}_n]$. In any case, the next guess will be the middle point of the interval: $\hat{t}^{n+1} = (\hat{t}^n + t_{right})/2$ or $\hat{t}^{n+1} = (t_{left} + \hat{t}^n)/2$, respectively. The algorithm stops when the length of the next interval is less than the established accuracy, h.

The bisection-based algorithm to compute the retention time t_g is outlined in Table 1 (Algorithm 3). This method is much faster than the direct method, because here the number of iterations is around $\log_2((t_{right} - t_{left})/h)$, and in addition convergence is always guaranteed.

3.3. Newton's method

The Newton's method consists in computing iteratively the following recursive equation:

$$\hat{t}^{n+1} = \hat{t}^n - \frac{F(\hat{t}^n)}{F'(\hat{t}^n)} \tag{18}$$

giving rise to the sequence $(\hat{t}^n)_{n=1}^{\infty}$, F' being the first derivative of the F function. The computation of the retention time using the Newton's method is given in Table 1 (Algorithm 4). The ending criterion is that the distance between two consecutive time guesses, t^n and t^{n+1}, is less than h. The sequence of times $\hat{t}^n$ will converge with a quadratic rate if a certain set of conditions is satisfied, such as: the second derivative F'' exists and is continuous, and $F'(\hat{t}^n) \neq 0$ for $n \geq 0$. This means that convergence is not always guaranteed.

Observe that the derivative of $F(t)$ can be easily obtained from Eq. (11):

$$\begin{aligned} F'(t) &= t'_0(\varphi(t))\varphi'(t) - \frac{1}{k(\varphi(t - t_0(\varphi(t))))}(1 - t'_0(\varphi(t))\varphi'(t)) \\ &= t'_0(\varphi(t))\varphi'(t) + \frac{t'_0(\varphi(t))\varphi'(t) - 1}{k(\varphi(t - t_0(\varphi(t))))} \end{aligned} \tag{19}$$

Since usually $t'_0(\varphi(t)) < 0$ and $k > 0$, then $\varphi'(t) \geq 0$ will imply $F'(t) < 0$. That is, if the gradient is increasing or constant, then $F(t)$ is decreasing. If the dead time is constant, then $F'(t)$ is non-differentiable, thus the condition for the quadratic convergence rate fails. Meanwhile, if the dead time is not constant, $F'(t)$ does not exists for $t = \tau_a$. As a consequence, the full potential of the Newton's method to find quickly the solution is only possible in some regions of the gradient programme. Thus, its theoretical superiority with regard to the bisection method is not so straightforward, and should be examined in the numerical experiments. Even worse, we have verified that the Newton's method is not safe enough to be used directly in gradient problems and requires modifications to grant a successful convergence. This will be the subject of the next section.

3.4. Newton-bisection method

The always-convergent bisection method and the fast Newton's method can be conveniently combined to solve their mutual limitations. This is carried out by applying the bisection method, considering instead of the middle point $(t_{left} + t_{right})/2$ the value given by Eq. (18), in case it is inside the interval $[t_{left}, t_{right}]$. Of course, there are situations where the middle point should be considered instead, such as when $F'(\hat{t}^n) = 0$. The algorithm describing the computation of t_g using the Newton-bisection method is described in Table 1 (Algorithm 5).

As will be shown in Section 6, the Newton-bisection and bisection methods have a fairly similar performance in the problems discussed in this work, owing to the partial fulfilment of the Newton's requirements.

4. Application to non-integrable retention models

Some retention models $k(\varphi)$, such as the log-quadratic equation (Eq. (8)) and Nikitas (Eq. (9)) models, formally lack of an explicit expression for the primitive function. For Eq. (8), there is a solution, but it depends on the error function. Without explicit primitives, the root-finding methods described above cannot be applied directly. However, we show here that for non-integrable models, $1/k(\varphi(t))$ can be safely replaced by another mathematical function $\pi(\varphi(t))$ having primitive, $\hat{P}$, where $\varphi(t)$ corresponds to a linear segment.

Suppose we wish to compute t_g with an error lower than ε. In practical situations, where the retention time is between 1 and 100 min, $\varepsilon = 10^{-3}$ is small enough. If the reciprocal of the retention model, $1/k(\varphi)$, is approximated by the $\pi(\varphi)$ function, the $F(t)$ function in Eq. (11) would be transformed as follows (see also Eqs. (12) and (13)):

$$\begin{aligned} \hat{F}(t) &= t_0(\varphi(t)) - t_{ext} - \int_0^{A(t)} \pi(\varphi(t))dt = t_0(\varphi(t)) - t_{ext} \\ &\quad -[\hat{P}(A(t)) - \hat{P}(0)] \end{aligned} \tag{20}$$

The approximation $\hat{t}_g$ to the retention time will be found as the zero of Eq. (20). This value is likely $\hat{t}_g \neq t_g$, but if $\pi(\varphi(t))$ is properly defined, a very close solution can be reached ($|t_g - \hat{t}_g| \leq \varepsilon$). Therefore, we propose to replace $F(t)$ by $\hat{F}(t)$ in Algorithms 1–5, for non-integrable models. The way $\pi(\varphi(t))$ is defined to this goal is next described. Observe that:

$$0 = F(t_g) - \hat{F}(\hat{t}_g) = F(t_g) - F(\hat{t}_g) + F(\hat{t}_g) - \hat{F}(\hat{t}_g) \tag{21}$$

By applying the Taylor series expansion, a value ξ exists belonging to the interval $[t_g, \hat{t}_g]$, such that:

$$F(t_g) - F(\hat{t}_g) = F'(\xi)(t_g - \hat{t}_g) \tag{22}$$

From Eqs. (21) and (22):

$$t_g - \hat{t}_g = \frac{F(\hat{t}_g) - \hat{F}(\hat{t}_g)}{F'(\xi)} \tag{23}$$

Let us denote a time T larger than the elution time of all solutes. Hence $t_g, \hat{t}_g \in [0,T]$. Then,

$$|t_g - \hat{t}_g| \leq \frac{\max_{t \in [0,T]} \left|F(t) - \hat{F}(t)\right|}{\min_{t \in [0,T]} \left|F'(t)\right|} \tag{24}$$

The minimum $m = \min_{t \in [0,T]} |F'(t)|$ should be positive, which is true whenever the gradient is increasing, or the elution is isocratic. Taking into account that $t'_0 < 0$, $k > 0$ and $\varphi' \geq 0$, from Eq. (19) it is deduced that $\left|F'(t)\right| \geq \min(1/k(\varphi))$, where $\varphi \in [c_{min}, c_{max}]$, being c_{min} and c_{max} the extreme concentrations in the gradient. This minimum takes place at c_{min}. Hence, $m \geq 1/k(c_{min})$. On the other hand:

$$\begin{aligned} |F(t) - \hat{F}(t)| &= \left| \int_0^{A(t)} \left(\pi(\varphi(t)) - \frac{1}{k(\varphi(t))} \right) dt \right| \\ &\leq T \max \left| \pi(\varphi(t)) - \frac{1}{k(\varphi(t))} \right| \end{aligned} \tag{25}$$

In conclusion, the precision in the computation ($|t_g - \hat{t}_g| \leq \varepsilon$) is guaranteed by defining $\pi(\varphi(t))$, such that $\left|\pi(\varphi(t)) - 1/k(\varphi(t))\right| \leq \varepsilon m/T$, provided that φ belongs to the interval $[c_{min}, c_{max}]$. Thus, in order to get the desired precision ε, it is enough that:

$$\left| \pi(\varphi(t)) - \frac{1}{k(\varphi(t))} \right| \leq \frac{\varepsilon}{Tk(c_{min})} \tag{26}$$

The above method can be applied whenever $\pi(\alpha+\beta\ t)$ (the approximated function within a linear ramp in the gradient) has primitive. We recommend the use of a polynomial for approximating $1/k(\varphi)$. If Tchebyshev nodes were used for the interpolation, the polynomial degree would be reduced. Given the polynomial degree n, Tchebyshev nodes are just:

$$\varphi_i = \cos\left(\frac{2i+1}{2n+2}\pi\right) \quad \text{for} \quad i = 0,\ 1, \ldots, n. \tag{27}$$

5. Experimental

For this study, 14 sulphonamides were considered as probe compounds (ordered according to their retention times): (1) sulphaguanidine, (2) sulphanilamide, (3) sulphadiazine, (4) sulphathiazole, (5) sulphapyridine, (6) sulphamerazine, (7) sulphamethazine, (8) sulphamethizole, (9) sulphamonomethoxine, (10) sulphachloropyridazine, (11) sulphamethoxazole, (12) sulphisoxazole, (13) sulphadimethoxine, and (14) sulphaquinoxaline, all from Sigma (Roedermark, Germany). The training set consisted of five isocratic experiments at 10, 13, 15, 20 and 25% (v/v) acetonitrile. Other details can be found elsewhere [34].

6. Results and discussion

In previous sections, some root finding methods were proposed to calculate efficiently retention times in reversed-phase liquid chromatography using gradient elution. Here a numerical study of the performance of these methods is carried out. Diverse situations, related to regression modelling and optimisation of chromatographic resolution, are addressed. Note that the computation time varies depending on the computer being used and on the background processes running simultaneously along the calculation. Thus, for these evaluations, both the computation time (t_F) and the number of times that the fundamental equation for gradient elution has to be solved (n_F) are given (Table 2). For those cases where the total time estimation is unpractical, a straightforward multiplication provides an estimation of the time the problem would require.

Henceforth we will consider that the fundamental equation for gradient elution (Eq. (4)) has an "analytical solution" when the calculation of the primitive can be done through algebraic expressions. Some situations involving analytical solutions, such as the combination of the linear solvent strength model [35] (or Eqs. (8) and (10)) with simple linear gradients, allow obtaining explicit expressions that provide the retention times. Otherwise, more complex expressions are obtained, which gives rise to a branched structure depending on the segment where the solute leaves the column. This is less attractive under a practical standpoint.

Next, the performance of several root-finding methods is evaluated in three frameworks: (i) isolated predictions, (ii) regression modelling problems using gradient training sets, and (iii) optimisation of multi-linear gradients. For these studies, the Neue-Kuss model (Eq. (10)) was selected.

6.1. Performance of the root-finding methods in isolated predictions

The following root-finding methods were considered: numerical integration in a direct search (Num), and the bisection (B), Newton's (N), and Newton-bisection (NB) methods. The time required to compute the retention for one thousand random multi-linear gradients of 10 nodes was evaluated. The same set of gradient programmes was used to evaluate all methods. The concentration of organic solvent in the gradients was increased in a segmented way between 10 and 25% acetonitrile in 60 min. The random generation granted a more representative population of retention times that could participate along a search, owing to the larger variability in solute retention, associated to the differences in slope between the different sections in the gradient programmes.

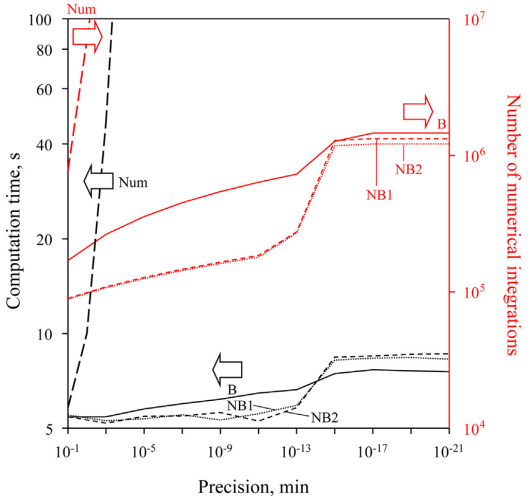


Fig. 2. Number of times (right axis) the fundamental equation for gradient elution has been solved, and computation time (left axis) required for calculating all retention times of the 14 sulphonamides, in one thousand random multi-linear gradients of 10 nodes. Numerical integration (Num), bisection (B) method alone, and combinations of Newton's and bisection methods (NB1 and NB2, see text) were investigated.

In this section, only the calculation time strictly associated to each root-finding method will be inspected. In later sections, the root-finding methods will be evaluated under other objectives that require additional calculations. Also, in this and the next section, the numerical integration was extended only to the last linear segment where the solute leaves the column. Note that, in situations where the combination of the retention model and gradient programme lacks of primitive, the calculation time would be much longer because the numerical treatment should be extended to the whole time axis. This will be illustrated in Section 6.3.

From the first studied examples, it was observed that the analytical integration using the Newton's method does not offer guarantees of success, so this method is not safe enough to be used without the cooperation of other methods. Thus, we have considered two calculation schemes instead, both including the bisection method. In the first scheme (NB1), when the Newton's method fails, the bisection method is applied until the process is finished for the current solute. In the second scheme (NB2), the bisection method is applied once the failure is detected in the current iteration, and the Newton's method is retried in the next iteration. Incidental further failures would lead to new applications of the bisection method. The NB2 scheme corresponds to Algorithm 5. Whenever the Newton's method is valid, the NB1 scheme operates according to Algorithm 4. When the Newton's method fails, the NB1 scheme operates according to Algorithm 3.

Fig. 2 shows the results found when each root-finding method was applied at different precision levels. As expected, the numerical integration (Num) requires calculation times that quickly become prohibitive when the precision is increased, although as mentioned, it was only applied to the last sector of the multi-linear gradient. In general, the numerical integration is three orders of magnitude slower than the analytical counterparts.

Table 2
Training set used for testing the fitting performance of gradient retention times to the Neue-Kuss retention model (see Fig. 3).

	Training set			
	Gradient 1	Gradient 2	Gradient 3	Gradient 4
Acetonitrile range, v/v[a]	10–25%	13–20%	15–25%	20–25%
Starting and ending times[b]	10–60 min	10–60 min	10–60 min	10–40 min
Compound	Retention time, min			
Sulphaguanidine	2.300	2.050	1.947	1.788
Sulphanilamide	2.887	2.529	2.385	2.174
Sulphadiazine	6.681	4.896	4.208	3.246
Sulphathiazole	8.924	5.878	4.777	3.338
Sulphapyridine	9.287	6.443	5.290	3.645
Sulphamerazine	10.498	7.088	5.830	4.149
Sulphamethazine	15.431	10.052	7.906	5.127
Sulphamethizole	18.062	11.982	9.123	5.343
Sulphamonomethoxine	24.906	18.172	13.758	7.512
Sulphachloropyridazine	25.483	19.370	15.142	8.837
Sulphamethoxazole	30.464	24.744	19.323	11.321
Sulphisoxazole	36.094	31.321	24.020	13.814
Sulphadimethoxine	48.090	48.356	35.961	21.368
Sulphaquinoxaline	50.117	52.125	38.449	22.645

[a] Starting and ending acetonitrile concentrations (φ_{start} and φ_{end}, respectively) for the four linear gradient experiments.
[b] Starting times of the linear ramps (t_{start}) and times at which the target acetonitrile concentration was reached (t_G). Before t_{start} and beyond t_G, the acetonitrile concentration was kept constant at φ_{start} and φ_{end}, respectively (the dwell time, t_{dwell} = 1.1625 min, has been included in the provided time coordinates).

It can be seen that the three other root-finding algorithms (B, NB1 and NB2) offer similar performance. Interestingly, the performance of the bisection method is somewhat inferior, but comparable to the schemes that include the Newton's method. It is true that the Newton's method is able to progress more quickly towards the solution, but each evaluation takes more computation time than the bisection method (associated to the derivatives). For this reason, the bisection method offers a surprisingly acceptable performance.

Finally, it should be indicated that the process of estimation of retention times can be considered acceptable until a precision level of 10^{-14} min (Fig. 2). Beyond this precision, the results are too unsecured, owing to the accumulation of uncertainties throughout the whole calculation process. Since even in the calculation of numerical derivatives, it is not necessary in practice to go beyond a precision of 10^{-14} min, the most advisable method is the NB1 scheme. In addition, we have observed that once the Newton's method has failed, it will fail systematically in the next iterations. This is another reason that makes the second scheme of the Newton-bisection (NB2) method less advisable. Nevertheless, both schemes (NB1 and NB2) perform similarly in practice.

From the cumulative n_F and t_F values in Table 3 (using for the fittings the values in Table 2 as training data), it can be calculated that the numerical integration needs 9×10^{-4} to 2×10^{-3} seconds for each time step. This means that, for each minute a solute takes to leave the column from the time of the node a^*-1 (see Fig. 1), the method needs around (10^{-3} min/integration step) seconds for the computation. For example, for a 10^{-5} min precision, for each minute of delay ($t_g - t_{a^*-1}$), 100 seconds of computation is required.

6.2. Analytical versus numerical integration in retention modelling using gradient data

The performance in retention modelling (regression) of root-finding methods using gradient data is next studied. With this purpose, a training set consisting of four linear gradients was selected. Table 2 indicates the retention times for each solute and the parameters defining the gradient programmes used as training set. Fig. 3 shows the training gradient profiles and associated chromatograms.

To perform the regressions, both analytical and numerical integration methods were considered. In the case of numerical integration, the time axis was explored at two levels of precision (0.01 and 0.001 min), which means that the gradient retention times will be obtained with these (or larger) uncertainty levels. Here one must be aware that an accuracy of 0.01 min may be insufficient when the gradients show strong variations [34]. The analytical integration for the prediction of gradient retention times was carried out with the Newton-bisection method NB1 (see Section 6.1), up to a 10^{-6} min precision level. For the analytical integration, two precision levels were considered: (i) the experimental uncertainty of each solute found in the isocratic mode (expressed as standard deviation in the prediction or pure experimental error, s_{pe}) and (ii) the machine precision in the manipulation of floating point numbers. The pure experimental error was calculated as:

$$s_{pe} = \sqrt{\frac{1}{n-p}\left(\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right)} \quad (28)$$

where for a given solute, n and p are the number of training experiments and parameters in the solute retention model, and y_i and $\hat{y}_i$ the experimental and predicted retention times corresponding to each i experiment.

Table 3 shows the results of both integration methods, expressed as the number n_F of evaluations of Eq. (11) to carry out the numerical integration, and the time needed (t_F, measured in seconds) until the local method (Powell) [36]) converged. The results should be taken with caution because local searches do not follow equivalent paths with independent searches, when the precision level changes. Thus, the number of iterations required by a shorter integration step is usually smaller than that required by a longer step, because greater precision (i.e., shorter integration step) is translated in a less erratic evolution. However, occasionally, the search with 0.01 min worked better than with 0.001 min. It should be noted that, for sulphaguanidine, convergence was obtained above the pure experimental error.

There is another consideration in the analysis of the results in Table 3. Usually, the default initial values $d_1 = 800$, $d_2 = 4$, and $d_3 = 50$ were adequate to fit the Neue-Kuss model (Eq. (10)) for typical solutes, using a local search method. However, in some configurations the fitting failed. When this happened (indicated in Table 3

Table 3
Number of times (n_F) the root function (Eq. (11)) had to be calculated up to convergence, and corresponding computing time (t_F), for fitting each solute to the Neue-Kuss model (Eq. (10)), using the gradient data shown in Table 2 as training set, and both numerical and analytical integrations.

Compound	Pure experimental error[a]	Numerical integration[b]				Analytical integration[c]			
		0.01 min		0.001 min		Experimental uncertainty[d]		Machine precision[e]	
	s_{pe}, min	n_F	t_F, s	n_F	t_F, s	n_F	t_F, s	n_F	t_F, s
Sulphaguanidine	0.0196	5180	288.1[f]	2380	121.2[f]	–	–	790	26.4
Sulphanilamide	0.0463	4083	184.6[f]	2709	134.3[f]	971	34.4	1373	48.1
Sulphadiazine	0.1199	4047	231.2[f]	449	66.7	734	20.3	5311	146.1
Sulphathiazole	0.1176	3896	216.1[f]	1838	279.0	366	10.1	1134	31.5
Sulphapyridine	0.0465	1709	91.8	1267	202.2	395	10.8	1755	49.2
Sulphamerazine	0.2308	2502	136.5[f]	1909	297.9	111	2.9	1465	40.5
Sulphamethazine	0.2838	1253	67.8	1325	204.1	124	3.2	350	9.1
Sulphamethizole	0.0517	2712	240.8[f]	1205	187.0	103	2.7	2224	60.7
Sulphamonomethoxine	0.1808	3073	170.8[f]	1513	238.5	123	3.4	544	15.6
Sulphachloropyridazine	0.0869	4052	224.2[f]	1092	172.7	892	26.9	1392	41.6
Sulphamethoxazole	0.1930	840	46.4	266	40.9	906	25.1	977	27.2
Sulphisoxazole	0.2417	762	42.4	1468	232.5	150	4.1	4059	112.2
Sulphadimethoxine	0.5821	2767	151.7	1794	270.5	252	7.1	1134	31.5
Sulphaquinoxaline	0.3955	3282	181.7	1428	221.8	109	2.9	2464	67.4
Cumulative n_F and t_F (min)[g]	–	40158	37.9	20643	44.5	6450	3.2	24628	11.6

[a] Magnitude of the pure experimental error (s_{pe}), expressed as the standard deviation around the regression curve fitted to experimental isocratic retention times, using the Neue-Kuss model (s_{pe} for gradients should be smaller).
[b] Numerical integration was applied only to the last gradient segment along which the solute is leaving the column.
[c] The Newton-bisection method (NB1) was used for this study.
[d] Non-linear fittings were carried out by setting the s_{pe} of each solute ([a]) as termination accuracy.
[e] Non-linear fittings were carried out leaving the fitting process to follow till no improvements were found (i.e., termination accuracy conditioned by the machine precision).
[f] The local method failed using the default parameters. An additional scan of initial values was needed (see text).
[g] Cumulative number of function evaluations (i.e., n_F, number of times the fundamental equation for gradient elution was solved), and calculation time (t_F) in minutes for the fitting of the 14 solutes.

with the superindex f), a coarse search of initial values was performed using a genetic algorithm, applying then again the local method with the best set of values found by the genetic algorithm. In the cases where the local method failed in the first run, the number of required numerical integrations was larger for a 0.01 min integration step compared to a 0.001 min step. The calculation time was around 30–40 min for the numerical integration method. The analytical integration offered better precision and less computation time, so this type of integration is the preferable option.

6.3. Analytical versus numerical integration in the optimisation of multi-linear gradients

The mixture of 14 sulphonamides was used to evaluate the performance in the optimisation of the resolution using multi-linear gradients of different complexity. Each gradient had two fixed nodes delimiting the search space with the coordinates (t_{dwell}, φ_0) and (t_G, φ_{end}), and n additional nodes, so that the total number of linear sectors was $n+3$, the first and last of which were isocratic (Fig. 4). Four series of experiments were considered (Table 4), which were solved using genetic algorithms with the following configuration: population of 150 individuals that evolves with a probability of crossover of 100%, probability of mutation of 3.3%, and probability of introduction of the best individual of 5%.

In the first series, the number of nodes was varied between $n=1$ and 15 nodes, using analytical integration with the same precision (10^{-6} min). In the second series, the number of nodes was kept constant ($n=5$) and the precision was varied between 10^{-3} and 10^{-18} min. The third and fourth series used numerical integration. The third series varied the number of nodes, and used a constant precision of 0.01 min due to the slower calculation. Finally, in the fourth series, the effect of increasing the precision level in the numerical integration was explored in an optimisation of a multi-linear gradient with 5 nodes. In this series, the precision ranged between 0.1 and 0.0001 min, because the calculation time increases exponentially with larger precision. For 0.0001 min precision, the calculation time (4.5 h) was obtained by extrapolation considering that the evolution of 6 generations needed 114.9 min.

Table 4 indicates the number of generations needed to get convergence. To obtain the total computation time, the number of generations should be multiplied by the time used for each generation. The random nature of the genetic algorithms leads to a certain variability in the convergence pattern. For this reason, the number of generations necessary to reach the end conditions oscillated between 14 and 24 generations. It should be indicated that the resolution of the 14 sulphonamides cannot be considered a too difficult separation problem. These compounds can be easily separated independently of the number of nodes. The global resolution, measured as peak purity [30], always exceeded the value $R_{max}=0.995$ ($R=1.000$ denotes full resolution). Due to the variability of convergence, the performance should not be evaluated once the method converges, but after a constant number of generations. Since all optimised configurations required at least 14 generations for convergence, the results found in the 14th generation were taken as reference. Thus, Table 4 provides, on the one hand, the time necessary to operate 14 generations, and on the other, the time associated to the calculation of a single generation.

The time needed to evolve 14 generations illustrates the convenience of using the analytical integration. However, for low precision searches, the numerical integration can be still acceptable. However, precision levels poorer than 0.01 min can be inappropriate in some situations, such as isocratic experiments that include pulses of organic solvent [34]. With a larger number of nodes, there is no significant increase in the calculation time needed to find the optimal gradient, using the analytical integration. In spite of this result, it should be taken into account that a harder sample whose optimal is more sensitive to small variations would require more generations to reach convergence, when the complexity of the gradient increases. This difference is irrelevant when the problem has no great difficulty and complete resolution can be reached. Numerical integration shows also minor dependence with the number of nodes, although with a much poorer preci-

Table 4
Performance of the numerical and analytical integration in the optimisation of resolution for the separation of the 14 sulphonamides listed in Table 2.

Type of integration[a]	Optimised nodes (n)	Precision, min	Generations until convergence[b]	R_{max}[c]	Calculation time up to the 14th generation, s	Time by generation, s[d]
Analytical	1	10^{-6}	15	0.9972	23.3	1.67
Analytical	2	10^{-6}	15	0.9975	23.6	1.68
Analytical	5	10^{-6}	15	0.9973	25.0	1.78
Analytical	10	10^{-6}	17	0.9975	24.3	1.75
Analytical	15	10^{-6}	18	0.9980	25.1	1.78
Series 2						
Analytical	5	10^{-3}	14	0.9971	23.4	1.67
Analytical	5	10^{-6}	15	0.9973	25.0	1.78
Analytical	5	10^{-9}	15	0.9976	24.1	1.72
Analytical	5	10^{-12}	15	0.9972	24.1	1.84
Analytical	5	10^{-15}	17	0.9972	28.4	2.08
Analytical	5	10^{-18}	15	0.9971	30.7	2.19
Series 3						
Numerical	1	0.01	18	0.9957	109.7	7.82
Numerical	2	0.01	16	0.9969	110.4	7.87
Numerical	5	0.01	19	0.9968	113.2	8.05
Numerical	10	0.01	19	0.9970	114.8	8.16
Numerical	15	0.01	24	0.9973	117.2	8.31
Series 4						
Numerical	5	0.1	20	0.9983	63.9	4.61
Numerical	5	0.01	19	0.9968	113.2	8.05
Numerical	5	0.001	15	0.9970	1135 (19 min)	81.1
Numerical	5	0.0001	–[e]	–[e]	16089 (4.47 h)[e]	1149[e]

[a] The Newton-bisection method (NB1) was selected for the analytical integration. For numerical integration, the whole time axis was stepped according to the indicated precision (i.e., integration step).
[b] The number of function evaluations (i.e., number of times the fundamental equation for gradient elution was solved) is 150 (population size) × 14 (solutes) × number of generations.
[c] Maximal global peak purity found after convergence.
[d] Time expressed as seconds needed for calculating and evolving a generation constituted by 150 encoded gradients, and obtained by diving the total calculation time up to convergence by the number of generations.
[e] Total calculation time exceeded 4 h (estimations are extrapolated from 6 generations, which need 114.9 min).

Table 5
Performance of the Tchebyshev interpolation to approximate the Neue-Kuss model (Eq. (10)) at a predefined precision level of 10^{-6} min. The interpolation error threshold that the polynomials may achieve and the true error are given. The needed polynomial degree for reaching the threshold is also indicated, together with the prediction error measured as the absolute difference between the retention times according to the polynomial model with respect to the Neue-Kuss model, obtained for a linear gradient where the acetonitrile concentration was increased from 10% to 25% in 60 min.

Compound	Threshold	True error	Polynomial degree	Prediction error
Sulphaguanidine	2.53×10^{-7}	1.17×10^{-7}	8	5.74×10^{-9}
Sulphanilamide	4.41×10^{-8}	2.31×10^{-8}	9	1.11×10^{-8}
Sulphadiazine	1.26×10^{-6}	7.45×10^{-7}	9	5.51×10^{-10}
Sulphathiazole	7.99×10^{-9}	9.13×10^{-10}	10	8.19×10^{-10}
Sulphapyridine	7.56×10^{-8}	1.27×10^{-8}	10	2.07×10^{-10}
Sulphamerazine	1.21×10^{-7}	1.10×10^{-7}	9	6.02×10^{-9}
Sulphamethazine	1.51×10^{-7}	1.43×10^{-7}	8	6.78×10^{-9}
Sulphamethizole	3.47×10^{-7}	1.28×10^{-7}	8	1.68×10^{-9}
Sulphamonomethoxine	2.53×10^{-7}	1.50×10^{-8}	9	7.14×10^{-10}
Sulphachloropyridazine	1.28×10^{-8}	3.66×10^{-9}	10	5.67×10^{-10}
Sulphamethoxazole	6.56×10^{-8}	4.72×10^{-8}	8	1.68×10^{-8}
Sulphisoxazole	1.47×10^{-6}	9.00×10^{-8}	8	1.02×10^{-9}
Sulphadimethoxine	6.27×10^{-8}	4.40×10^{-9}	10	6.14×10^{-10}
Sulphaquinoxaline	5.06×10^{-7}	1.57×10^{-7}	8	1.58×10^{-9}

sion (0.01 and 10^{-6} min in the numerical and analytical integration, respectively), the calculation time is 5 times larger.

The chromatogram in Fig. 4 was obtained with a gradient of 5 optimised nodes. The inclusion of more nodes reduces the analysis time considerably, although the resolution in the example will be similar. Note that the analytical integration is less sensitive to the gradient complexity. A precision up to 10^{-12} min did not affect the calculation time appreciably, which agrees with the results shown in Fig. 2. When the established precision level is too demanding (10^{-15} min, or even larger), the computation time tends to increase, although in a minor extent. This strongly contrasts with the performance of the numerical integration, which quickly becomes unfeasible with a precision equal or better than 0.001 min.

6.4. Tchebyshev interpolation

When the combination of a retention model and a multi-linear gradient, expressed in Eq. (11) by $k(\varphi(t))$, leads to expressions that lack of primitive, the reciprocal of the retention model ($1/k(\varphi)$) can be replaced by interpolating polynomials ($\pi(\varphi(t))$) whose primitive is easy to compute (Eq. (20)). With this aim, we propose the use of polynomials based on Tchebyshev nodes, which give rise to highly accurate approximations. These polynomials constitute an interesting choice in situations where the models have been previously established, such as optimisation problems. The substitution by a polynomial does not suppose any significant increment in the computation time with regard to the original models, since

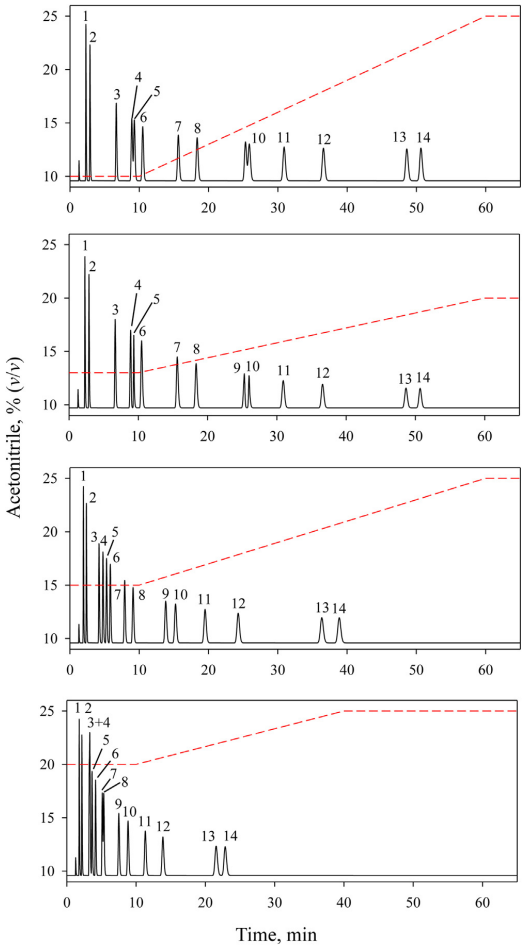


Fig. 3. Chromatograms and associated linear gradients used as training set for modelling the retention of 14 sulphonamides. See Section 5 for compound identity.

the coefficients of the polynomial that approximates the original function are established only once, before starting the optimisation.

Section 4 establishes that for reaching a target precision (ε), the interpolation of $1/k(\varphi)$ must be done with an error smaller than the target interpolation error (threshold) calculated as $\varepsilon/(Tk(c_{min}))$ (Eq. (26)). For validating the approach, the retention models of the 14 sulphonamides were approximated by Tchebyshev interpolating polynomials. Table 5 shows the results for $\varepsilon = 10^{-6}$ min, consisting of: (i) threshold, (ii) true interpolation error (maximal difference between $\pi(\varphi)$ and $1/k(\varphi)$), (iii) polynomial degree necessary for achieving the threshold, and (iv) prediction error, calculated as the difference between the retention times obtained with the polynomial models with regard to the original ones. For these predictions, a linear gradient increasing the acetonitrile concentration between 10% and 25% in 60 min was selected.

The observation of the results given in Table 5 allows concluding that, on the one hand, the true interpolation error is always smaller than the threshold, and on the other, the prediction error is below 10^{-6} min. Even more, the error is three orders of magnitude smaller. The reason behind this low error may be the large time set for these

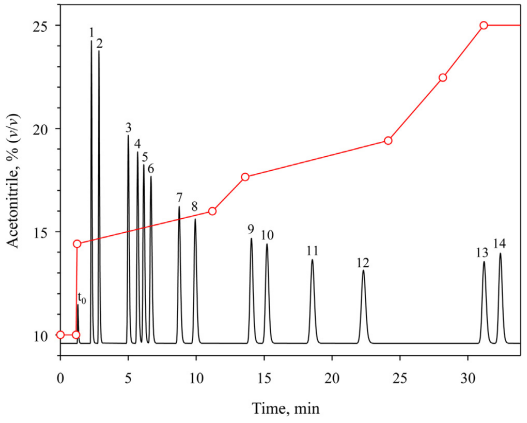


Fig. 4. Optimal chromatogram and associated multi-linear gradient. The total number of nodes is 7: the 5 optimised nodes and two fixed nodes defining the search space (non-optimised). See Section 5 for compound identity.

calculations ($T = 100$ min, which is larger than the retention times of all solutes). Meanwhile, from the theoretical outline in Section 4, it can be deduced that the largest error is associated to an isocratic elution at the minimal modifier concentration in the linear gradient. Finally, Table 5 indicates that the maximal polynomial degree to achieve the desired accuracy is 10.

7. Conclusions

Gradient elution requires considering continuous changes in the solvent conditions along the migration. This can be implemented in a differential fashion leading to the estimation of peak position and profile [37], and in an integrated fashion, which leads to algebraic expressions, faster to be calculated. This work enhances this second perspective.

The conventional resolution of the fundamental equation of gradient elution implies working out the upper limit of an integral. Instead, we propose outlining the problem as a root search. In this work, several alternatives to solve efficiently the fundamental equation are studied, which are valid for multi-linear gradients. The substitution of non-integrable retention models by Tchebyshev polynomial approximations, which are pre-calculated before the resolution of the integral equation, was also investigated.

It has been found that the calculation of retention times in gradient elution using the root-finding Newton's method did not yield results as good as expected, in terms of progression towards the solution. This can be attributed to the sudden changes in slope along the consecutive sectors in the multi-linear gradients, where the conditions required for a full performance of the Newton's method are not fulfilled. Furthermore, this lack of fulfilment gives rise to convergence problems when the Newton's method is run alone. As a consequence, occasional failures may happen for some solutes and gradients, especially in the presence of sudden changes in the gradient programme. This situation is found for instance along an optimisation, when the initial concentration level of organic modifier is inadequate for the requirements of the analytes. The best solution to overcome this problem is the combination of the Newton's and bisection methods. The bisection method should be preferably applied after the first detection of a failure (scheme NB1), along the iterations. It was also found that the bisection method by itself has, surprisingly, as good performance in finding the gradient retention time as the Newton's method and its combinations with the bisection method. In general, the studied methods yielded sat-

isfactory results at least up to 10^{-14} min, although precisions in the 10^{-6}–10^{-9} min range are sufficient for most calculations.

In modelling problems, which imply regression procedures, the numerical integration (Num) offered poorer performance than the schemes based on the analytical integration (NB1, NB2 and B). The computation times for the numerical integration were notably longer with regard to all methods based on analytical integration. This happened even in favourable configurations, like that studied in the shown regression example, where the numerical integration was only applied to the last gradient segment where the solute was leaving the column.

In the optimisation of resolution, when the number of nodes or the calculation precision was varied, the performance of all analytical methods was comparable. The time required for the numerical integration was more susceptible to the number of nodes of the gradient programme and the required precision, becoming quickly unfeasible for precisions better than 10^{-3} min, with an exponential increase in the computation time. Despite the poorer performance, the numerical integration is still a valid option in situations where high precision is not required, and there is no primitive function. This is the case of gradient optimisation, where precisions in the 10^{-2}–10^{-3} min range are enough.

Finally, the analytical integration using root-finding methods can be applied to non-integrable retention models through Tchebyshev polynomial approximations. These approximations give rise to mathematical expressions with terms having primitive. Thus, as long as the gradient programme is applied divided in consecutive linear segments, the use of such polynomial-based models gives rise to a universal method to calculate retention times with gradients as complex as desired, valid for non-integrable retention models.

Acknowledgements

This work was supported by Projects CTQ2016-75644-P and MTM2017-83942-P (Ministry of Science, Innovation and Universities, Spain, and FEDER funds), and PROMETEO/2016/128 (Direcció General d'Universitat, Investigació i Ciència, Generalitat Valenciana, Spain). Sergio López-Ureña thanks the Ministry of Education, Culture and Sports, MECD of Spain for the FPU14/02216 grant.

References

[1] J.E. Haky, D.A. Teifer, Gradient elution, in: J. Cazes (Ed.), Encyclopedia of Chromatography, Taylor & Francis, New York, 2006, pp. 393–396.
[2] L.R. Snyder, J.W. Dolan, High-Performance Gradient Elution, John Wiley & Sons, Inc., Hoboken, NJ, 2007.
[3] J.W. Dolan, L.R. Snyder, Gradient elution chromatography, in: R.A. Meyers (Ed.), Encyclopedia of Analytical Chemistry, John Wiley & Sons, New York, 2012.
[4] J.J. Baeza-Baeza, C. Ortiz-Bolsico, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Approaches to model the retention and peak profile in linear gradient reversed-phase liquid chromatography, J. Chromatogr. A 1284 (2013) 28–35.
[5] S.A. Tomellini, R.A. Hartwick, H.B. Woodruff, Computer-based numerical integration for the calculation of retention times in gradient high-performance liquid chromatography, Anal. Chem. 57 (1985) 811–816.
[6] P. Nikitas, A. Pappa-Louisi, Expressions of the fundamental equation of gradient elution and a numerical solution of these equations under any gradient profile, Anal. Chem. 77 (2005) 5670–5677.
[7] V. Concha-Herrera, G. Vivó-Truyols, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Limits of multi-linear gradient optimisation in reversed-phase liquid chromatography, J. Chromatogr. A 1063 (2005) 79–88.
[8] C. Ortiz-Bolsico, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Optimisation of gradient elution with serially-coupled columns. Part II: Multi-linear gradients, J. Chromatogr. A 1373 (2014) 51–60.
[9] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, 1998.
[10] E. Tyteca, S.H. Park, R.A. Shellie, P.R. Haddad, G. Desmet, Computer-assisted multi-segment gradient optimization in ion chromatography, J. Chromatogr. A 1381 (2015) 101–109.
[11] S. López-Ureña, J.R. Torres-Lapasió, R. Donat, M.C. García-Alvarez-Coque, Gradient design for liquid chromatography using multi-scale optimization, J. Chromatogr. A 1534 (2018) 32–42.
[12] S. Heinisch, E. Lesellier, C. Podevin, J.L. Rocca, A. Tchapla, Computerized optimization of RP-HPLC separation with nonaqueous or partially aqueous mobile phases, Chromatographia 44 (1997) 529–537.
[13] R.G. Wolcott, J.W. Dolan, L.R. Snyder, Computer simulation for the convenient optimization of isocratic reversed-phase liquid chromatographic separations by varying temperature and mobile phase strength, J. Chromatogr. A 869 (2000) 3–25.
[14] J.R. Torres-Lapasió, M. Rosés, E. Bosch, M.C. García-Alvarez-Coque, Interpretive optimisation strategy applied to the isocratic separation of phenols by reversed-phase liquid chromatography with acetonitrile–water and methanol–water mobile phases, J. Chromatogr. A 886 (2000) 31–46.
[15] W.D. Beinert, R. Jack, V. Eckert, S. Galushko, V. Tanchuck, I. Shishkina, A program for automated HPLC method development, Am. Lab. 33 (2001) 14–15.
[16] G. Vivó-Truyols, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Towards the optimization of complementary systems in reversed-phase liquid chromatography, Chromatographia 56 (2002) 699–707.
[17] I. Molnar, Computerized design of separation strategies by reversed-phase liquid chromatography: development of DryLab software, J. Chromatogr. A 965 (2002) 175–194.
[18] P. Nikitas, A. Pappa-Louisi, A. Papageorgiou, Simple algorithms for fitting and optimisation for multilinear gradient elution in reversed-phase liquid chromatography, J. Chromatogr. A 1157 (2007) 178–186.
[19] S. Pous-Torres, J.R. Torres-Lapasió, M.J. Ruiz-Angel, M.C. García-Alvarez-Coque, Interpretive optimisation of organic solvent content and flow-rate in the separation of β-blockers with a Chromolith RP-18e column, J. Sep. Sci. 32 (2009) 2793–2803.
[20] C. Ortiz-Bolsico, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Simultaneous optimization of mobile phase composition, column nature and length to analyse complex samples using serially coupled columns, J. Chromatogr. A 1317 (2013) 39–48.
[21] S. Fasoula, C. Zisi, P. Nikitas, A. Pappa-Louisi, Retention prediction and separation optimization of ionizable analytes in reversed-phase liquid chromatography by organic modifier gradients in different eluent pHs, J. Chromatogr. A 1305 (2013) 131–138.
[22] L.R. Snyder, J.W. Dolan, Optimizing selectivity during reversed-phase high performance liquid chromatography method development: prioritizing experimental conditions, J. Chromatogr. A 1302 (2013) 45–54.
[23] J.R. Torres-Lapasió, S. Pous-Torres, C. Ortiz-Bolsico, M.C. García-Alvarez-Coque, Optimisation of chromatographic resolution using objective functions including both time and spectral information, J. Chromatogr. A 1377 (2015) 75–84.
[24] N. Rácz, I. Molnár, A. Zöldhegyi, H.J. Rieger, R. Kormány, Simultaneous optimization of mobile phase composition and pH using retention modeling and experimental design, J. Pharm. Biomed. Anal. 160 (2018) 336–343.
[25] R.L. Burden, J.D. Faires, Numerical Analysis, 3rd ed., Prindle, Weber & Schmidt, Boston, MA, 1985.
[26] U.D. Neue, Theory of peak capacity in gradient elution, J. Chromatogr. A 1079 (2005) 153–161.
[27] P.J. Schoenmakers, H.A.H. Billiet, L. de Galan, Description of solute retention over the full range of mobile phase compositions in reversed-phase liquid chromatography, J. Chromatogr. 282 (1983) 107–121.
[28] P. Nikitas, A. Pappa-Louisi, Retention models for isocratic and gradient elution in reversed-phase liquid chromatography, J. Chromatogr. A 1216 (2009) 1737–1755.
[29] U.D. Neue, H.J. Kuss, Improved reversed-phase gradient retention modeling, J. Chromatogr. A 1217 (2010) 3794–3803.
[30] M.C. García-Alvarez-Coque, J.R. Torres-Lapasió, J.J. Baeza-Baeza, Models and objective functions for the optimisation of selectivity in reversed-phase liquid chromatography, Anal. Chim. Acta 579 (2006) 125–145.
[31] P.J. Schoenmakers, H.A.H. Billiet, R. Tijssen, L. de Galan, Gradient selection in reversed-phase liquid chromatography, J. Chromatogr. 149 (1978) 519–537.
[32] Y. Shan, W. Zhang, A. Seidel-Morgenstern, Z. Ruihuan, Z. Yukui, Multi-segment linear gradient optimization strategy based on resolution map in HPLC, Sci. China Series B: Chem. 49 (2006) 315–325.
[33] A. Pappa-Louisi, P. Nikitas, A. Papageorgiou, Optimisation of multilinear gradient elutions in reversed-phase liquid chromatography using ternary solvent mixtures, J. Chromatogr. A 1166 (2007) 126–134.
[34] J.A. Navarro-Huerta, A. Gisbert-Alonso, J.R. Torres-Lapasió, M.C. García-Alvarez-Coque, Benefits of solvent concentration pulses in liquid chromatography modelling, unpublished results.
[35] K. Valkó, L.R. Snyder, J.L. Glajch, Retention in reversed-phase liquid chromatography as a function of mobile-phase composition, J. Chromatogr. A 656 (1993) 501–520.
[36] M.J.D. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives, Computer J. 7 (1964) 155–162.
[37] K. Lan, J.W. Jorgenson, Theoretical investigation of the spatial progression of temporal statistical moments in linear chromatography, Anal. Chem. 72 (2000) 1555–1563.