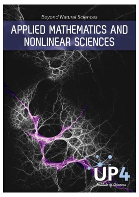
Applied Mathematics and Nonlinear Sciences 2(2) (2017) 367–384



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

$$|E - \hat{E}_K| \leq \varepsilon$$

where ε is a user-dependent predetermined accuracy, even when f is only piecewise smooth.

Notice that

$$|E_K - \hat{E}_K| = 2^{-K}\left|\left(\frac{1}{2}v_0^K + \sum_{i=1}^{2^K-1}\hat{v}_i^K + \frac{1}{2}v_{2^K}^K\right) - \left(\frac{1}{2}\hat{v}_0^K + \sum_{i=1}^{2^K-1}\hat{v}_i^K + \frac{1}{2}\hat{v}_{2^K}^K\right)\right| \leq \|v^K - \hat{v}^K\|_1 \leq \|v^K - \hat{v}^K\|_\infty,$$

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 $v^K = \big(f(\xi_i^K)\big)_{i=0}^{2^K}$.

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*:

$$d_i^k := \delta_{2i+1}^{k+1} = f(\xi_{2i+1}^{k+1}) - (P_k^{k+1}\mathscr{D}_k f)_{2i+1}, \qquad 0 \leq i < n_{k+1}/2 = n_k.$$

For simplicity, let us denote $v^k := \mathscr{D}_k f$. Then

$$d_i^k = v_{2i+1}^{k+1} - (P_k^{k+1}v^k)_{2i+1}, \qquad 0 \leq i < n_{k+1}/2 = n_k. \tag{5}$$

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 $(a_1;a_2;a_3;\ldots)$ the concatenation of many vectors. Note that (6) can be applied recursively to obtain the following equivalences

$$v^k \leftrightarrow (v^{k-1};d^{k-1}) \leftrightarrow (v^{k-2};d^{k-2};d^{k-1}) \leftrightarrow \cdots \leftrightarrow (v^0;d^0;d^1;\ldots;d^{k-1}) \in \mathbb{R}^{n_k+1}.$$

The *multiresolution decomposition of k levels*, M^k, (or multiresolution transform) is defined as

$$M^k v^k = (v^0;d^0;d^1;\ldots;d^{k-1}).$$

We denote its inverse, the *inverse multiresolution transform*, by

$$M^{-k}(v^0;d^0;d^1;\ldots;d^{k-1}) = v^k.$$

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

$$\|\mathscr{I}(\cdot;\xi^k,\mathscr{D}_k f) - f\|_\infty \le Ch^s_k, \quad \forall k \ge 0,$$

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:*

$$I(v^k_{i-1},v^k_i,v^k_{i+1},v^k_{i+2}) = \frac{1}{2}v^k_i + \frac{1}{2}v^k_{i+1} - \frac{1}{8}\Big(H(v^k_{i+1}-v^k_i, v^k_{i+2}-v^k_{i+1}) - H(v^k_i - v^k_{i-1}, v^k_{i+1}-v^k_i)\Big),$$

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

$$\|v^K - \hat{v}^K\|_\infty \overset{K\to\infty}{\longrightarrow} 0.$$

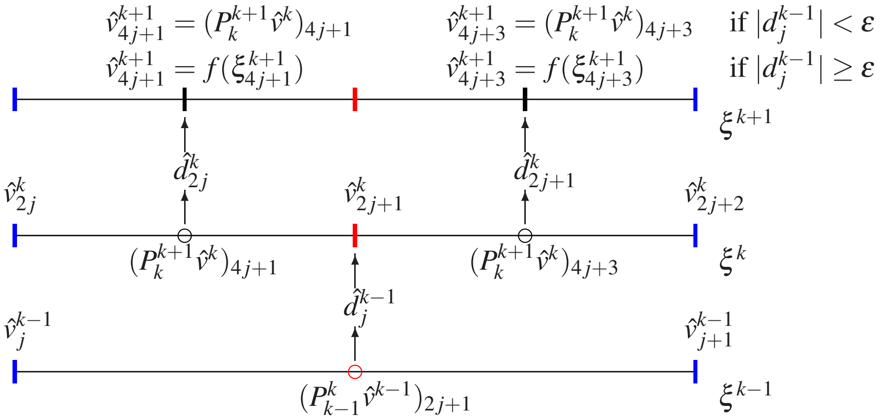

$\hat{v}^{k+1}_{4j+1} = (P^{k+1}_k \hat{v}^k)_{4j+1}$
$\hat{v}^{k+1}_{4j+3} = (P^{k+1}_k \hat{v}^k)_{4j+3}$
if $|d^{k-1}_j| < \varepsilon$
$\hat{v}^{k+1}_{4j+1} = f(\xi^{k+1}_{4j+1})$
$\hat{v}^{k+1}_{4j+3} = f(\xi^{k+1}_{4j+3})$
if $|d^{k-1}_j| \geq \varepsilon$
ξ^{k+1}
$\hat{d}^k_{2j}$
$\hat{d}^k_{2j+1}$
$\hat{v}^k_{2j}$
$\hat{v}^k_{2j+1}$
$\hat{v}^k_{2j+2}$
$(P^{k+1}_k \hat{v}^k)_{4j+1}$
$(P^{k+1}_k \hat{v}^k)_{4j+3}$
ξ^k
$\hat{d}^{k-1}_j$
$\hat{v}^{k-1}_j$
$\hat{v}^{k-1}_{j+1}$
$(P^k_{k-1} \hat{v}^{k-1})_{2j+1}$
ξ^{k-1}

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.

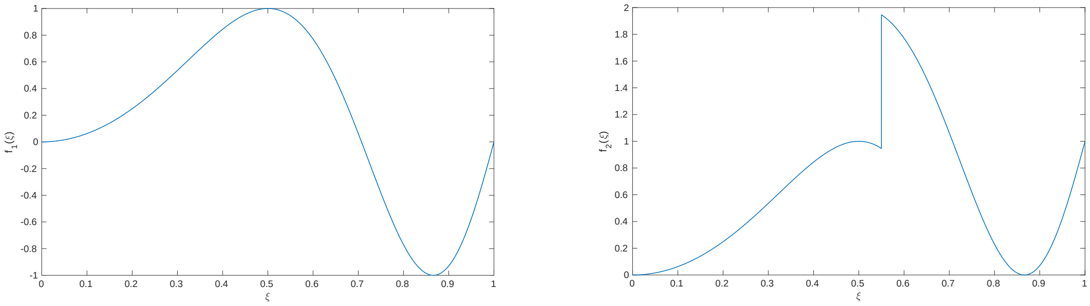


Fig. 2 Graphic representation of: left, f_1; right, f_2. Both functions are defined in (12).

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}	$\|v^K-\hat{v}^K\|_\infty$	$\|v^K-\hat{v}^K\|_1$	$\lvert E-\hat{E}_K\rvert$
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}	$\|v^K-\hat{v}^K\|_\infty$	$\|v^K-\hat{v}^K\|_1$	$\lvert E-\hat{E}_K\rvert$
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}	$\|v^K-\hat{v}^K\|_\infty$	$\|v^K-\hat{v}^K\|_1$	$\lvert E-\hat{E}_K\rvert$
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}.

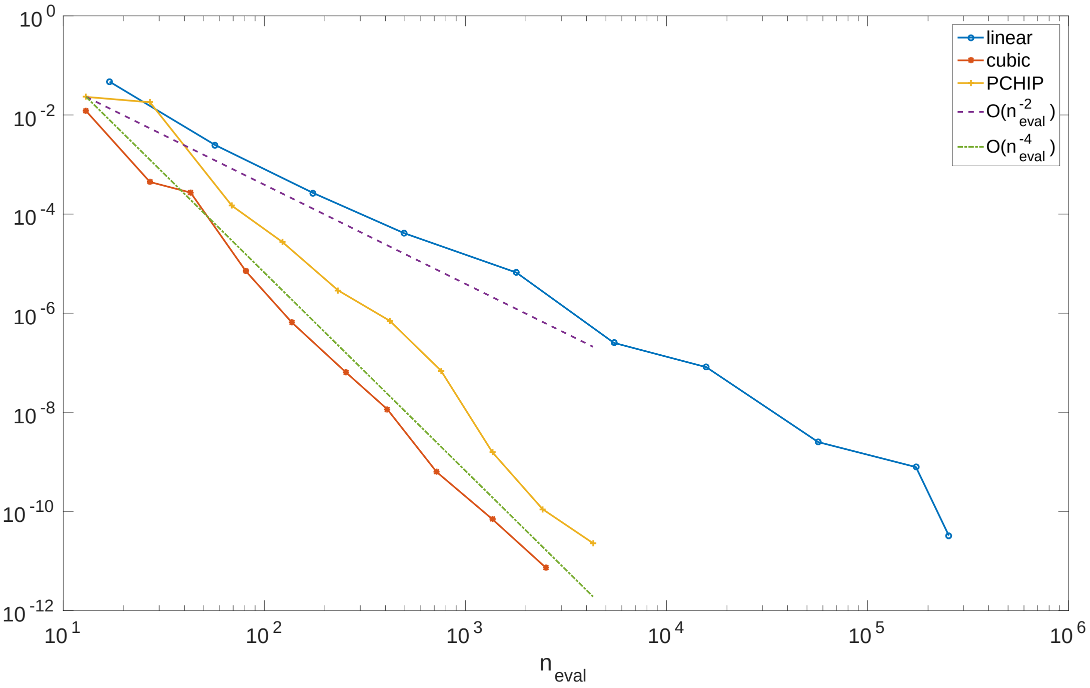


Fig. 3 In continuous line, the graphics of $\|v^{15} - \hat{v}^{15}\|_\infty$ as a function of n_{eval} for $\varepsilon^k = \varepsilon$ varying from 10^{-1} to 10^{-10}, applied to each local rule and f_1. For comparison, in discontinuous line are shown the growth rates $O(n_{eval}^{-2})$ and $O(n_{eval}^{-4})$.

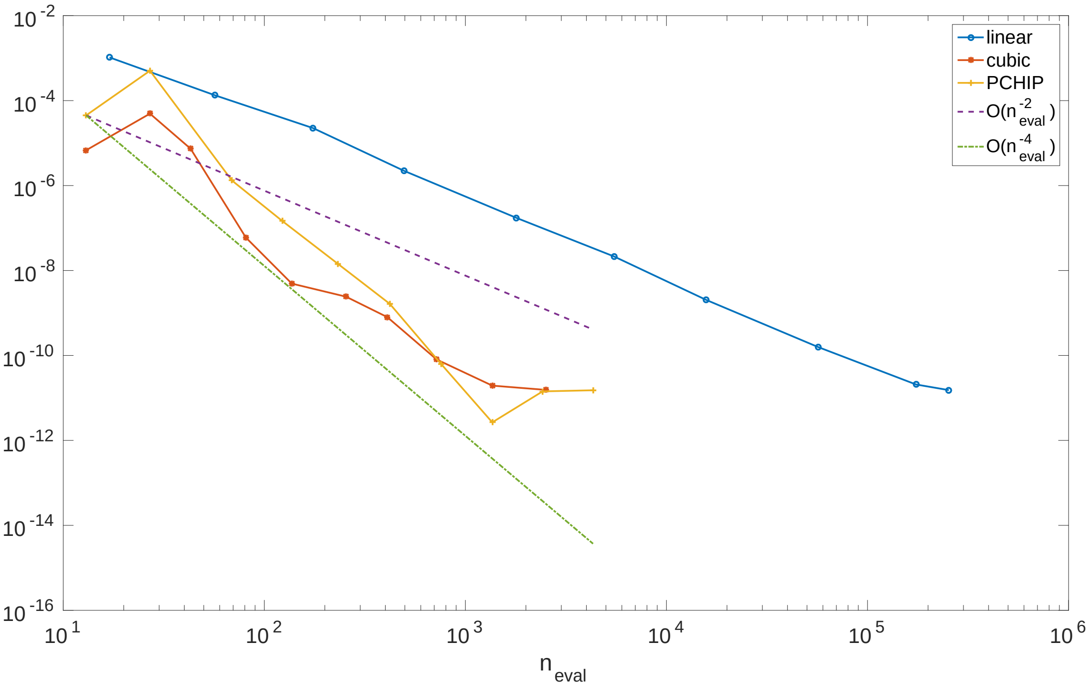


Fig. 4 In continuous line, the graphics of $|E - \hat{E}_{15}|$ as a function of n_{eval} for $\varepsilon^k = \varepsilon$ varying from 10^{-1} to 10^{-10}, applied to each local rule and f_1. For comparison, in discontinuous line are shown the growth rates $O(n_{eval}^{-2})$ and $O(n_{eval}^{-4})$.

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 7.8e-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 $|E-\hat{E}_K|\to$ 1.1e-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.

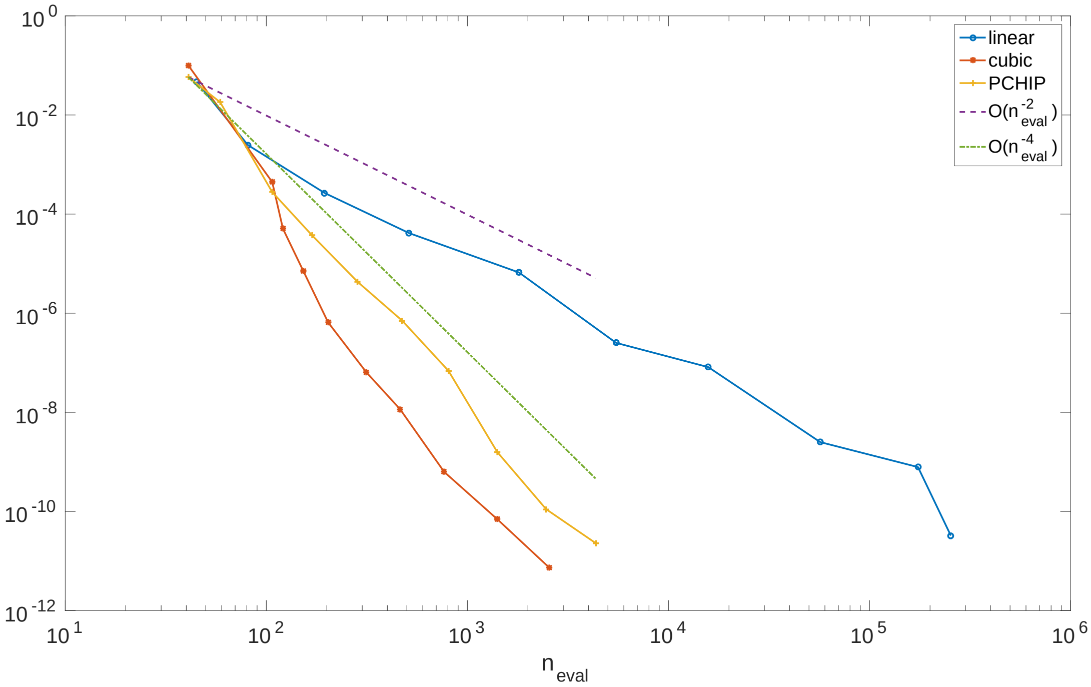


Fig. 5 In continuous line, the graphics of $\|v^{15} - \hat{v}^{15}\|_\infty$ as a function of n_{eval} for $\varepsilon^k = \varepsilon$ varying from 10^{-1} to 10^{-10}, applied to each local rule and f_2. For comparison, in discontinuous line are shown the growth rates $O(n_{eval}^{-2})$ and $O(n_{eval}^{-4})$.

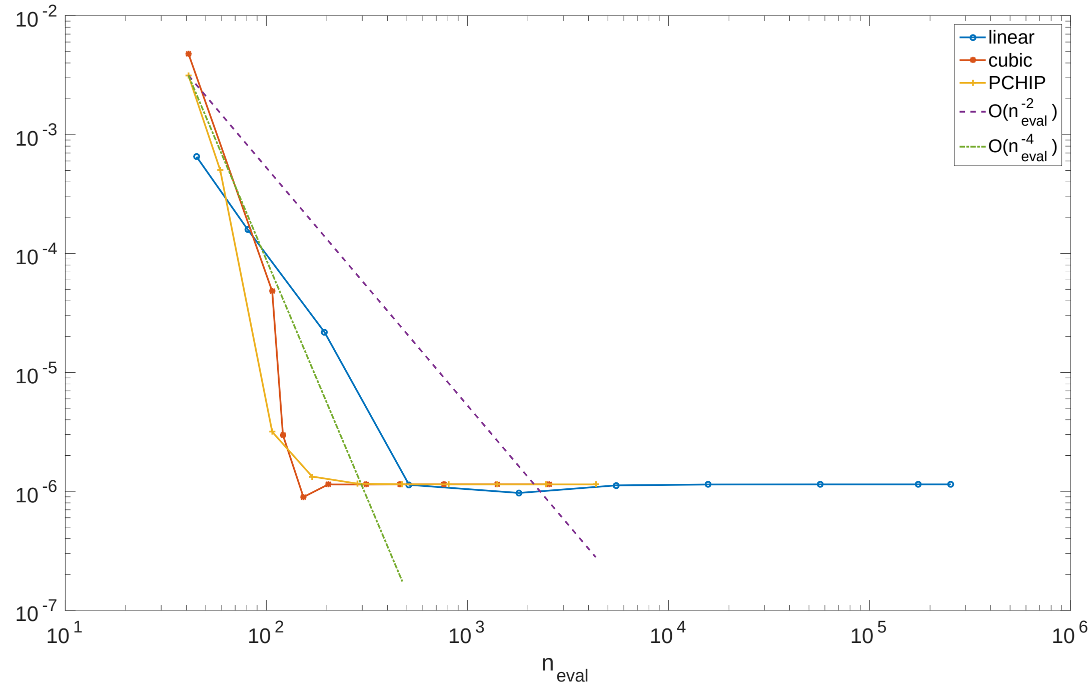


Fig. 6 In continuous line, the graphics of $|E - \hat{E}_{15}|$ as a function of n_{eval} for $\varepsilon^k = \varepsilon$ varying from 10^{-1} to 10^{-10}, applied to each local rule and f_2. For comparison, in discontinuous line are shown the growth rates $O(n_{eval}^{-2})$ and $O(n_{eval}^{-4})$.

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.