

Non-linear Local Polynomial Regression Multiresolution Methods Using ℓ^1 -norm Minimization with Application to Signal Processing

Francesc Aràndiga¹, Pep Mulet¹, and Dionisio F. Yáñez^{2,3}(✉)

¹ Department of Matemática Aplicada, Universitat de València,
C/Doctor Moliner, 46100 Burjassot (Valencia), Spain
{arandiga,mulet}@uv.es

<http://gata.uv.es/~arandiga>, <http://gata.uv.es/~mulet>

² Campus Capacitas, Universidad Católica de Valencia, C/Sagrado Corazón, 5,
46110 Godella (Valencia), Spain

³ Departamento de Matemáticas, CC. NN. y CC. SS. aplicadas a la educación,
Universidad Católica de Valencia, C/Sagrado Corazón,
46110 Godella (Valencia), Spain
dionisiofelix.yanez@ucv.es
<http://www.uv.es/diofeya>

Abstract. Harten's Multiresolution has been developed and used for different applications such as fast algorithms for solving linear equations or compression, denoising and inpainting signals. These schemes are based on two principal operators: decimation and prediction. The goal of this paper is to construct an accurate prediction operator that approximates the real values of the signal by a polynomial and estimates the error using ℓ^1 -norm in each point. The result is a non-linear multiresolution method. The order of the operator is calculated. The stability of the schemes is ensured by using a special error control technique. Some numerical tests are performed comparing the new method with known linear and non-linear methods.

Keywords: Statistical Multiresolution · Multiscale decomposition · Signal processing · Generalized wavelets

1 Introduction

During the past few years multiresolution analysis (MR) has been widely used in several applications as signal processing (see, e.g., [16–18]). Harten in [24] designed a MR framework based on two operators: decimation and prediction defining them as the composition of two functions: discretization and reconstruction. The discretization represents the nature of the data. The study of

This research was partially supported by Spanish MCINN MTM 2011-22741 and MTM 2014-54388.

this function can be found in, for example, [9, 10, 20]. However, the extensive literature is dedicated to the construction of an accurate reconstruction operator using different techniques, linear (see, e.g., [11, 12]) and non-linear. In particular, Aràndiga and Donat in [7] use the Essential Non-Oscillatory (ENO) technique in order to obtain a more adapted prediction operator. Belda et al. extend it introducing Weighted ENO techniques (see [5]). Also, non-linear techniques based on Piecewise Polynomial Harmonic (PPH) are used obtaining interesting results for signal compression and adaptation to edges (see [2]). Recently, in order to design a new prediction operator statistical tools (see, e.g., [15, 19, 25, 26, 28]) have been considered as learning processes (see [6]) and approximation by local polynomial regression (LPR) (see, e.g., [12]). In this context, three variables determine the problem. First, the degree, r , of the chosen polynomials. Afterwards, the weight function which assigns a value to each point depending on the distance to the approximated point. Finally, the loss function, $L(x, y)$. Typically, the ℓ^2 -minimization norm is used (see [11]). In this context, we extend the method substituting $L(x, y)$ by the ℓ^1 -norm which is introduced in the formulation of the problem. Then, a non-linear problem is obtained. Therefore, the order and the stability have to be studied. The order is a measure of the accuracy of the prediction operator. The stability is crucial for MR compression processes since in order to reduce the number of elements to store, we have to truncate several values. Thus, a control on the final error obtained *a priori* between the original signal and the approximate signal is necessary, several papers in the literature have already dealt with this (see, e.g., [3, 4, 8, 18]).

It is not difficult to prove that the order does not depend on the loss function. However, the stability, in this case, is not ensured. In this paper, the error control strategy presented by Harten et al. (see, e.g., [1, 7, 24]) is used.

The paper is organized as follows: we review Harten's MR and local polynomial regression in Sects. 2 and 3. Subsequently, the error control method is presented in Sect. 4. Finally, some numerical results are displayed, some conclusions are presented and future research is described.

2 Harten's Interpolatory Framework

Harten in [23, 24] developed a MR scheme based on four operators: discretization, reconstruction, decimation and prediction. Let \mathcal{F} be a locally integrable function space and V^k a discrete vector space, being k the level of the resolution, then the discretization function, $\mathcal{D}_k : \mathcal{F} \rightarrow V^k$, represents the nature of data. Typically, in signal processing is used a point-value method which consists in considering a signal as the values of a function in a grid. Therefore, let us indicate a nested set of uniform dyadic grids on the interval $[0, 1]$, as $X^k = \{x_j^k\}$, $x_j^k = j \cdot h_k$, $j = 0, \dots, J_k$, $J_k \cdot h_k = 1$, with $J_k = 2^k J_0$ and where J_0 is some integer. Then $f^L = \{f_j^L\}_{j=0}^{J_L}$ with $f_j^k := (\mathcal{D}_k(f))_j = f(x_j^k)$. We need to take into account that, as $x_j^{k-1} = x_{2j}^k$, then $f_{2j}^k = f_j^{k-1}$. Thus, in this case, we define the decimation operator which combines two levels of the MR, $\mathcal{D}_k^{k-1} : V^k \rightarrow V^{k-1}$ as

$$(\mathcal{D}_k^{k-1} f^k)_j = f_j^{k-1} = f(x_j^{k-1}) = f(x_{2j}^k) = f_{2j}^k, \quad 0 \leq j \leq J_{k-1}.$$

It must be linear. The reconstruction operator, $\mathcal{R}_k : V^k \rightarrow \mathcal{F}$ can be viewed as an interpolating or approximating function. In the literature (see [7, 24]), this operator is usually defined using the following sequence: Let r be the degree of the polynomial, s the number of points chosen for the interpolation and

$$z(x) \in \Pi_1^r(\mathbb{R}) = \left\{ g \mid g(x) = \sum_{m=0}^r c_m x^m, c_m \in \mathbb{R}, \forall m \right\}$$

such that $z(x_{j+m}^{k-1}) = f_{j+m}^{k-1}$ for $m = -s, \dots, s-1$; with $r+1 = 2s$. Then, we define:

$$(\mathcal{R}_{k-1} f^{k-1})(x) = z(x).$$

Therefore, we define the prediction operator to obtain the values on the level k as $\mathcal{P}_{k-1}^k := \mathcal{D}_k \mathcal{R}_{k-1}$, then

$$(\mathcal{P}_{k-1}^k f^{k-1})_j = \mathcal{D}_k(\mathcal{R}_{k-1} f^{k-1})_j = (\mathcal{D}_k(z))_j = z(x_j^k). \quad (1)$$

These operators should be consistent following the equation: $\mathcal{D}_k^{k-1} \mathcal{P}_{k-1}^k = \mathcal{I}_{k-1}$, where \mathcal{I}_k is the identity function defined in the space V^k . When the discretization based on point-value is chosen then it is easy to prove that this property is satisfied replicating the odd-values from the low level, as a consequence

$$(\mathcal{P}_{k-1}^k f^{k-1})_{2j} = f_j^{k-1}. \quad (2)$$

From standard 1D interpolation results we obtain that:

$$z(x_{2j-1}^k) = \sum_{l=1}^s \beta_l (f_{j+l-1}^{k-1} + f_{j-l}^{k-1})$$

where the coefficients β_l are

$$\begin{cases} s = 1 \Rightarrow \beta_1 = \frac{1}{2}, \\ s = 2 \Rightarrow \beta_1 = \frac{9}{16}, \beta_2 = -\frac{1}{16}, \\ s = 3 \Rightarrow \beta_1 = \frac{150}{256}, \beta_2 = -\frac{25}{256}, \beta_3 = \frac{3}{256}. \end{cases} \quad (3)$$

Observe that $e_{2j}^k = f_{2j}^k - (\mathcal{P}_{k-1}^k f^{k-1})_{2j} = f_j^{k-1} - f_j^{k-1} = 0$. Then, if we define $d_j^k = f_{2j-1}^k - (\mathcal{P}_{k-1}^k f^{k-1})_{2j-1}$ we have that the two sets $\{f^k\}$ and $\{f^{k-1}, d^k\}$ which are equivalent: From $\{f^k\}$ we obtain $f_j^{k-1} = f_{2j}^k$ and $d_j^k = f_{2j-1}^k - (\mathcal{P}_{k-1}^k f^{k-1})_{2j-1}$. Also, from $\{f^{k-1}, d^k\}$ we obtain $f_{2j-1}^k = (\mathcal{P}_{k-1}^k f^{k-1})_{2j-1} + d_j^k$ and $f_{2j}^k = f_j^{k-1}$.

Then, the multiscale decomposition algorithm is:

Algorithm 1 ($f^L \rightarrow M f^L = (f^0, d^1, \dots, d^L)$)

$$\begin{cases} \text{for } k = L : 1 \\ \quad f_j^{k-1} = f_{2j}^k, & j = 0 : J_{k-1} \\ \quad d_j^k = f_{2j-1}^k - (\mathcal{P}_{k-1}^k f^{k-1})_{2j-1} & j = 1 : J_{k-1} \\ \text{end} \end{cases} \quad (4)$$

And the multiscale reconstruction algorithm is:

Algorithm 2 ($Mf^L = (f^0, d^1, \dots, d^L) \longrightarrow M^{-1}Mf^L$)

$$\left\{ \begin{array}{l} \mathbf{for} \quad k = 1 : L \\ \quad f_{2j-1}^k = (\mathcal{P}_{k-1}^k f^{k-1})_{2j-1} + d_j^k, \quad j = 1 : J_{k-1} \\ \quad f_{2j}^k = f_j^{k-1}, \quad j = 0 : J_{k-1} \\ \mathbf{end} \end{array} \right. \quad (5)$$

Thus, $\{f^L\} \leftrightarrow \{\{f^0\}, \{d^1\}, \dots, \{d^L\}\}$. The compression is obtained when the values d^0, \dots, d^L are close to zero or zero. The key of these schemes is to construct an accurate prediction operator. Then, when the decomposition algorithm is used, a truncation or quantization technique is applied to the result. It is important to control the *a priori* error (see Sect. 4). In the next section we extend the method developed in [11] using ℓ^1 -norm minimization.

3 Design of LPR MR Prediction Operator: Brief Review

In this section, a family of prediction operators is constructed using statistical tools. In particular, we will review the principal components marked in [11].

Therefore, we consider the values of a signal in a determined level of MR $f^{k-1} = \{f_j^{k-1}\}_{j=0}^{J_{k-1}}$ with $f_j^{k-1} = f(x_j^{k-1})$ and $x_j^{k-1} = jh_{k-1}$.

For a fitting point on level k , x_{2j-1}^k , we approximate the value f_{2j-1}^k with a curve based on the points included in an interval centered in x_{2j-1}^k with length fixed, $(2s-1)h_{k-1}$, with $s > 1$ constant, non-necessary integer. In order to assign a weight to each value f_l^{k-1} such that $x_l^{k-1} \in [x_{2j-1}^k - (2s-1)h_k, x_{2j-1}^k + (2s-1)h_k]$ we define a kern function that will be denoted by K_s , as:

$$K_s(x_{2j-1}^k, x) = \omega\left(\frac{x_{2j-1}^k - x}{(2s-1)\tilde{h}_k}\right) \quad (6)$$

where \tilde{h}_k is a slight perturbation of the value h_k . In [11] the value $\tilde{h}_k = (1 + \epsilon)h_k$, with $\epsilon = 2 \cdot 10^{-3}$ is taken to unify the notation with the interpolation method. And $\omega(x)$ is a non-negative symmetric function (see [26]). We show these functions in Table 1.

Table 1. Kernel functions

rect	$\omega(u) = 1, u \leq 1$
tria	$\omega(u) = 1 - u , u \leq 1$
epan	$\omega(u) = 1 - u^2, u \leq 1$
bisq	$\omega(u) = (1 - u^2)^2, u \leq 1$
tcub	$\omega(u) = (1 - u ^3)^3, u \leq 1$
trwt	$\omega(u) = (1 - u ^2)^3, u \leq 1$

Let $z(x)$ be a polynomial of degree r , $z(x) \in \Pi_1^r(\mathbb{R})$. We choose a loss-function $L(x, y)$ which measures the distance between the real values on the level $k-1$ and the approximation. In [11] the chosen function is $L(x, y) = (x - y)^2$. Then, the classical least-squares method is obtained. For this paper, we propose a robust LPR with

$$L(x, y) = |x - y|.$$

Hence, the problem is the following:

$$\begin{aligned} \hat{z}_{2j-1}^k(x) &= \arg \min_{z(x) \in \Pi_1^r(\mathbb{R})} \sum_{l=0}^{J_{k-1}} K_s(x_{2j-1}^k, x_l^{k-1}) L(f_l^{k-1}, z(x_l^{k-1})) \\ \mathbf{c}_{2j-1}^k &= \arg \min_{c_m \in \mathbb{R}, m=0, \dots, r} \sum_{l=0}^{J_{k-1}} K_s(x_{2j-1}^k, x_l^{k-1}) \left| f_l^{k-1} - \sum_{m=0}^r c_m (x_l^{k-1})^m \right| \end{aligned} \quad (7)$$

and we define $\mathcal{R}_{k-1}(x) = \hat{z}_{2j-1}^k(x)$. Consequently, by Eqs. (1) and (2) we have that:

$$\begin{cases} (\mathcal{P}_{k-1}^k f^{k-1})_{2j-1} = \mathcal{D}_k(\mathcal{R}_{k-1} f^{k-1})_{2j-1} = (\mathcal{D}_k(\hat{z}_{2j-1}^k))_{2j-1} = \hat{z}_{2j-1}^k(x_{2j-1}^k), \\ (\mathcal{P}_{k-1}^k f^{k-1})_{2j} = f_j^{k-1}. \end{cases}$$

Remark 1. We consider as \hat{z}_{2j-1}^k the polynomial used to calculate the approximation to f_{2j-1}^k . Notice that we need a different polynomial for each estimation. It is proved in [11] that the prediction operator is independent of the level k and of the point j . In Sect. 3.1 more details are explained.

Remark 2. If $r + 1 = 2\lfloor s \rfloor$ (where $\lfloor \cdot \rfloor$ is the function that rounds a number to the nearest integer less than or equal to it) then the obtained method is the method based on piecewise polynomial interpolation, Eq. (3), independently of the weight and the loss functions that have been chosen.

3.1 Optimization Problem and Non-linearity of the Operator

The choice of the loss function is crucial for the applications. The distance between the real values and the approximation is calculated with this operator. It is easy to prove that if we choose the ℓ^2 -norm, then the resulting problem is the classical least-square minimization. In this section, we explain the method using ℓ^1 -norm minimization.

By using weight functions such that $w(u) = 0, |u| > 1$, we have that

$$K_s(x_{2j-1}^k, x_{j+l}^{k-1}) \neq 0 \text{ if } -\lfloor s \rfloor \leq l \leq \lfloor s \rfloor - 1,$$

and our problem can be rewritten as:

$$\hat{z}_{2j-1}^k(x_{2j-1}^k) = \arg \min_{c_m \in \mathbb{R}, m=0, \dots, r} \sum_{l=-\lfloor s \rfloor}^{\lfloor s \rfloor - 1} K_s(x_{2j-1}^k, x_{j+l}^{k-1}) \left| f_{j+l}^{k-1} - \sum_{m=0}^r c_m (x_{j+l}^{k-1})^m \right|. \quad (8)$$

For fixed r, j and s , problem (8) can be written as follows:

Let $\bar{f}_j^{k-1} = (f_{j-\lfloor s \rfloor}^{k-1}, \dots, f_{j+\lfloor s \rfloor-1}^{k-1})^T$, the $2\lfloor s \rfloor \times (r+1)$ matrix

$$X = \begin{pmatrix} 1 & x_{j-\lfloor s \rfloor}^{k-1} & \dots & (x_{j-\lfloor s \rfloor}^{k-1})^r \\ 1 & x_{j-\lfloor s \rfloor+1}^{k-1} & \dots & (x_{j-\lfloor s \rfloor+1}^{k-1})^r \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{j+\lfloor s \rfloor-1}^{k-1} & \dots & (x_{j+\lfloor s \rfloor-1}^{k-1})^r \end{pmatrix}; \quad (9)$$

and the matrix $2\lfloor s \rfloor \times 2\lfloor s \rfloor$

$$W_{2j-1}^k = \begin{pmatrix} \omega\left(\frac{x_{2j-1}^k - x_{j-\lfloor s \rfloor}^{k-1}}{(2s-1)h_k}\right) & 0 & \dots & 0 \\ 0 & \omega\left(\frac{x_{2j-1}^k - x_{j-\lfloor s \rfloor+1}^{k-1}}{(2s-1)h_k}\right) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \omega\left(\frac{x_{2j-1}^k - x_{j+\lfloor s \rfloor-1}^{k-1}}{(2s-1)h_k}\right) \end{pmatrix}. \quad (10)$$

Then, our problem is to calculate

$$\mathbf{c}_{2j-1}^k = \arg \min_{(c_0, \dots, c_r) \in \mathbb{R}^{r+1}} \|W_{2j-1}^k(X(c_0, \dots, c_r)^T - \bar{f}_j^{k-1})\|_1. \quad (11)$$

Therefore, we have that $\hat{z}_{2j-1}^k(x_{2j-1}^k) = A_r(x_{2j-1}^k)^T \mathbf{c}_{2j-1}^k$, where $A_r(x) = (1, x, \dots, x^r)$ and consequently

$$\begin{cases} (\mathcal{P}_{k-1}^k f^{k-1})_{2j-1} = \mathbf{c}_{2j-1}^k A_r(x_{2j-1}^k)^T, \\ (\mathcal{P}_{k-1}^k f^{k-1})_{2j} = f_j^{k-1}. \end{cases}$$

In order to solve the ℓ^1 -norm approximation problem, Eq. (11), different methods can be used (see [14]). In this paper, we reformulate it as a linear program in a convex optimization context (see more details in [13]):

$$\begin{aligned} & \text{minimize} && (1, \dots, 1)^T t \\ & \text{s.t.} && -t \preceq W_{2j-1}^k(X(c_0, \dots, c_r)^T - \bar{f}_j^{k-1}) \preceq t. \end{aligned} \quad (12)$$

where \preceq is the componentwise inequality between two vectors.

In the numerical experiments (Sect. 5), we obtain the solutions using a Matlab [27] package, called `cvx`, designed by M. Grant et al. in [21, 22].

A measure of the prediction operator's accuracy is the order. It is defined in the following section.

3.2 Order of the Scheme Using ℓ^1 -LPR

Definition 1. Let be $p(x) \in \Pi_1^r(\mathbb{R})$ an arbitrary polynomial of degree less than or equal to r . Then, the order of the prediction operator is r if

$$\mathcal{P}_{k-1}^k(\mathcal{D}_{k-1}p) = \mathcal{D}_k p, \quad (13)$$

i.e., the prediction operator is exact for polynomials of degree less than or equal to r .

In [11] is proved that the order of the scheme for LPR MR using least squares is equal to the order of the polynomial taken to approximate the real values. Analogously, in this case with $L(x, y) = |x - y|$, we have the same consequence. Therefore, we prove the following theorem.

Theorem 1. *The order of the MR scheme using LPR of degree r is, independently of the weight and the loss functions chosen, r .*

The proof is a direct consequence of the Eq. (7). If the discrete signal f^k is the discretization of a polynomial of degree r then the polynomial that minimizes the functional is exactly the same. Also, the distance between the approximation and real values is 0.

4 Error Control Strategy and Data Compression

In applications as data compression, the details, $\{d_j^k\}_{j=1}^{J_k}$, $k = 1, \dots, L$ are reduced by means of truncating or quantizing to keep a lower number of values. Thus, if we denote as $\hat{d}_j^k = |d_j^k|_\varepsilon$ being

$$|d_j^k|_\varepsilon = \begin{cases} d_j^k, & \text{if } |d_j^k| \geq \varepsilon; \\ 0, & \text{if } |d_j^k| < \varepsilon. \end{cases} \quad (14)$$

with $j = 0, \dots, J_k$, $k = 0, \dots, L$ and ε the introduced threshold parameter, then the error between the approximation and the real values has to be controlled, i.e.

$$\|f^L - \hat{f}^L\|_p \leq C\varepsilon, \quad p = 1, 2, \infty \quad (15)$$

where \hat{f}^L is the signal obtained after applying the Algorithm 2 and after truncating the details, and C is a constant.

In order to satisfy this error, the linearity of the operator is an essential ingredient (see, e.g., [7, 9, 12]). In our case, a non-linear prediction operator is designed using ℓ^1 -norm minimization. Therefore, a modification of the encoding procedure is necessary to ensure stability. We use the strategy showed in [1, 7, 10] based on a change of the algorithm. The algorithmic description of the modified encoding is as follows:

Algorithm 3

$$\left\{ \begin{array}{l} \text{for } k = L : 1 \\ \quad f_j^{k-1} = f_{2j}^k, j = 1 : J_{k-1} \\ \quad \text{end} \\ \quad \text{Set } \hat{f}^0 = f^0 \\ \quad \text{for } k = 1 : L \\ \quad \quad \hat{f}_0^k = f_0^L \\ \quad \quad \text{for } j = 1 : J_{k-1} \\ \quad \quad \quad d_j^k = f_{2j-1}^k - (\mathcal{P}_{k-1}^k f^{k-1})_{2j-1}, \hat{d}_j^k = |d_j^k|_\varepsilon \\ \quad \quad \quad \hat{f}_{2j-1}^k = (\mathcal{P}_{k-1}^k f^{k-1})_{2j-1} + \hat{d}_j^k, \hat{f}_{2j}^k = \hat{f}_j^{k-1} \\ \quad \quad \quad \text{end} \\ \quad \quad \text{end} \end{array} \right. \quad (16)$$

With this modification it is not difficult to prove the following proposition.

Proposition 1. [Aràndiga and Donat [7]] *Given a discrete sequence f^L and a tolerance ε , if the truncation parameters ε_k in the modified encoding algorithm (Alg. 3) are chosen so that*

$$\varepsilon_k = \varepsilon$$

then the sequence $\hat{f}^L = M^{-1}\{f^0, \hat{d}^1, \dots, \hat{d}^L\}$, with M^{-1} defined in Eq. (16), satisfies

$$\|f^L - \hat{f}^L\|_p \leq \varepsilon, \quad p = 1, 2, \infty. \quad (17)$$

5 Numerical Experiments

We apply the new method to compress signals. The MR schemes in the point-value framework using reconstruction operators obtained from Lagrange polynomial interpolatory techniques are considered. Also, we compare our algorithm with the non-linear method designed by Amat et al. (see [2]) and with the scheme using ℓ^2 -norm presented in [11]. Each MR scheme is identified by an acronym. The equivalences are as follows:

PV: Point-value using interpolation techniques with four centered points, Eq. (3), $s = 2$. The predictor operator is:

$$\begin{aligned} (\mathcal{P}_{k-1}^k f^{k-1})_{2j} &= f_j^{k-1}, \\ (\mathcal{P}_{k-1}^k f^{k-1})_{2j-1} &= \frac{9}{16}(f_j^{k-1} + f_{j-1}^{k-1}) - \frac{1}{16}(f_{j-2}^{k-1} + f_{j+1}^{k-1}). \end{aligned}$$

PPH: This scheme introduced by Amat et al. [2] is a nonlinear stable algorithm which consists in modifying the PV scheme:

$$\begin{aligned} (\mathcal{P}_{k-1}^k f^{k-1})_{2j} &= f_j^{k-1}, \\ (\mathcal{P}_{k-1}^k f^{k-1})_{2j-1} &= \frac{1}{2}(f_j^{k-1} + f_{j-1}^{k-1}) - \frac{1}{8}pph(d^2 f_j^{k-1}, d^2 f_{j-1}^{k-1}). \end{aligned}$$

where $d^2 f_j^{k-1} = f_{j+1}^{k-1} - 2f_j^{k-1} + f_{j-1}^{k-1}$ and pph is the function

$$\begin{aligned} pph(x, y) &= \left(\frac{\text{sign}(x) + \text{sign}(y)}{2} \right) \frac{2|x||y|}{|x| + |y|}, \quad \forall x, y \in \mathbb{R} \setminus \{0\}; \\ pph(x, 0) &= 0, \forall x \in \mathbb{R}; \quad pph(0, y) = 0, \forall y \in \mathbb{R}. \end{aligned}$$

kern $_{r,s}^p$: LPR MR method, where kern is a weight function showed in Table 1; p is the loss function used, $p = 1, 2$; the degree of polynomial is r and the parameter s is the bandwidth of the LPR, i.e., the number of points used to construct the approximate polynomial.

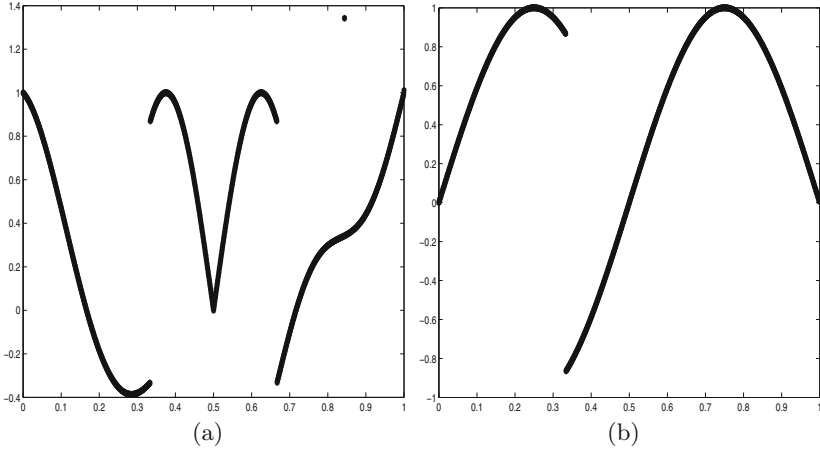


Fig. 1. Functions: (a) f_1 , (b) f_2

Table 2. TV of the original function, f_1 , and TV of the reconstructions using different weight functions and ℓ^p loss functions, with $p = 1, 2$

Method	f_1	PV	PPH	$\text{tria}_{1,3.5}^p$	$\text{epan}_{1,3.5}^p$	$\text{tcub}_{1,3.5}^p$	$\text{trwt}_{1,3.5}^p$
$p = 1$	9.433	10.157	9.350	9.360	15.288	9.360	9.360
$p = 2$				13.818	15.428	13.648	12.014

We use different functions to analyze the discrete data. It consists of the point-values of the functions: f_1 (Harten function, see [9,10]) and f_2 on the finest grid (see Fig. 1).

$$f_1(x) = \delta(x - x_{865}) + \begin{cases} -x \sin(\frac{3\pi}{2}x^2), & -1 < x \leq -\frac{1}{3}, \\ |\sin(2\pi x)|, & |x| < \frac{1}{3}, \\ 2x - 1 - \sin(3\pi x)/6, & \frac{1}{3} \leq x < 1. \end{cases}$$

$$f_2(x) = \begin{cases} \sin(2\pi x), & 0 \leq x \leq 1/3, \\ -\sin(2\pi x), & 1/3 \leq x \leq 1; \end{cases}$$

Firstly, we perform some experiments to check that the use of the ℓ^1 -norm reduces considerably the Gibbs phenomenon at the discontinuities. We take $J_0 = 256$ and $J_L = 1024$, i.e., $L = 3$ and we define a measure of oscillations, called Total Variation (TV) of a function as:

$$TV(f^k) = \sum_{j=1}^{J_k} |f_j^k - f_{j-1}^k|. \tag{18}$$

We compare the reconstructions obtained using different methods with the TV of the original function fixing the bandwidth for all the cases, $r = 1$, $s = 3.5$

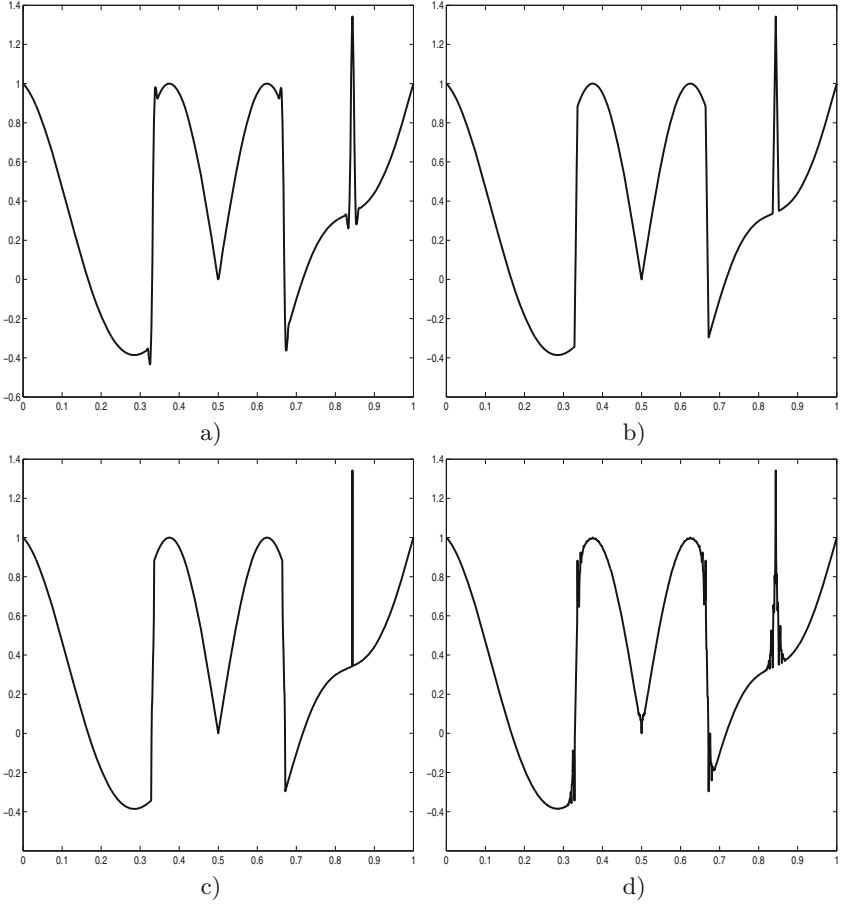


Fig. 2. Reconstructions obtained using different MR schemes: (a) PV, (b) PPH, (c) $\text{tcub}_{1,3.5}^1$ and (d) $\text{tcub}_{1,3.5}^2$

without keeping any details. In Table 2, we can see that when ℓ^1 -norm is used then the oscillations are reduced independently of the weight function chosen. The same rate than in the PPH case is obtained. However, when $p = 2$, the TV ratio is higher because of Gibbs phenomenon as we can observe in Fig. 2.

Afterwards, we present some tests to compress the discrete functions. We use the error control algorithm (Sect. 4) for each method.

In order to reduce the signals, the details are truncated following the strategy presented in the last section, i.e., $\hat{d}_j^k = |d_j^k|_\varepsilon$ with $j = 0, \dots, J_k$ and ε the constant introduced in Eq. (14). We measure the error in the ℓ^p discrete norm with $p = 1, 2, \infty$ defined by:

$$E_1 = \frac{1}{J_L} \sum_j |f_j^L - \hat{f}_j^L|, \quad E_2 = \sqrt{\frac{1}{J_L} \sum_j |f_j^L - \hat{f}_j^L|^2}, \quad E_\infty = \|f^L - \hat{f}^L\|_\infty.$$

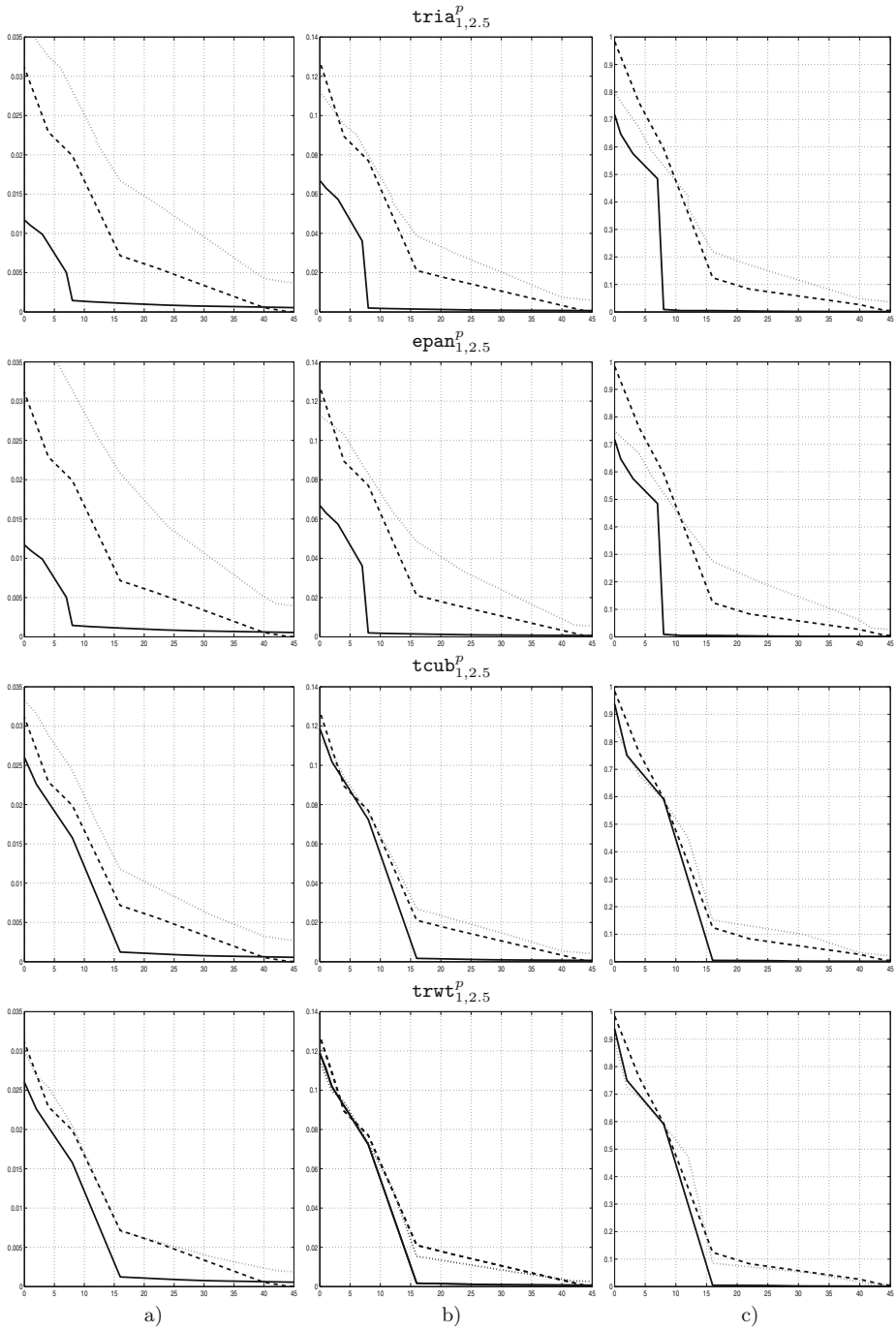


Fig. 3. Errors vs the number of details NNZ using MR schemes to compress the function f_1 with PV (dashed line) and $\text{kern}_{1,2.5}^p$ with $p = 1$ (solid line) and $p = 2$ (dotted line): (a) E_1 , (b) E_2 (c) E_∞

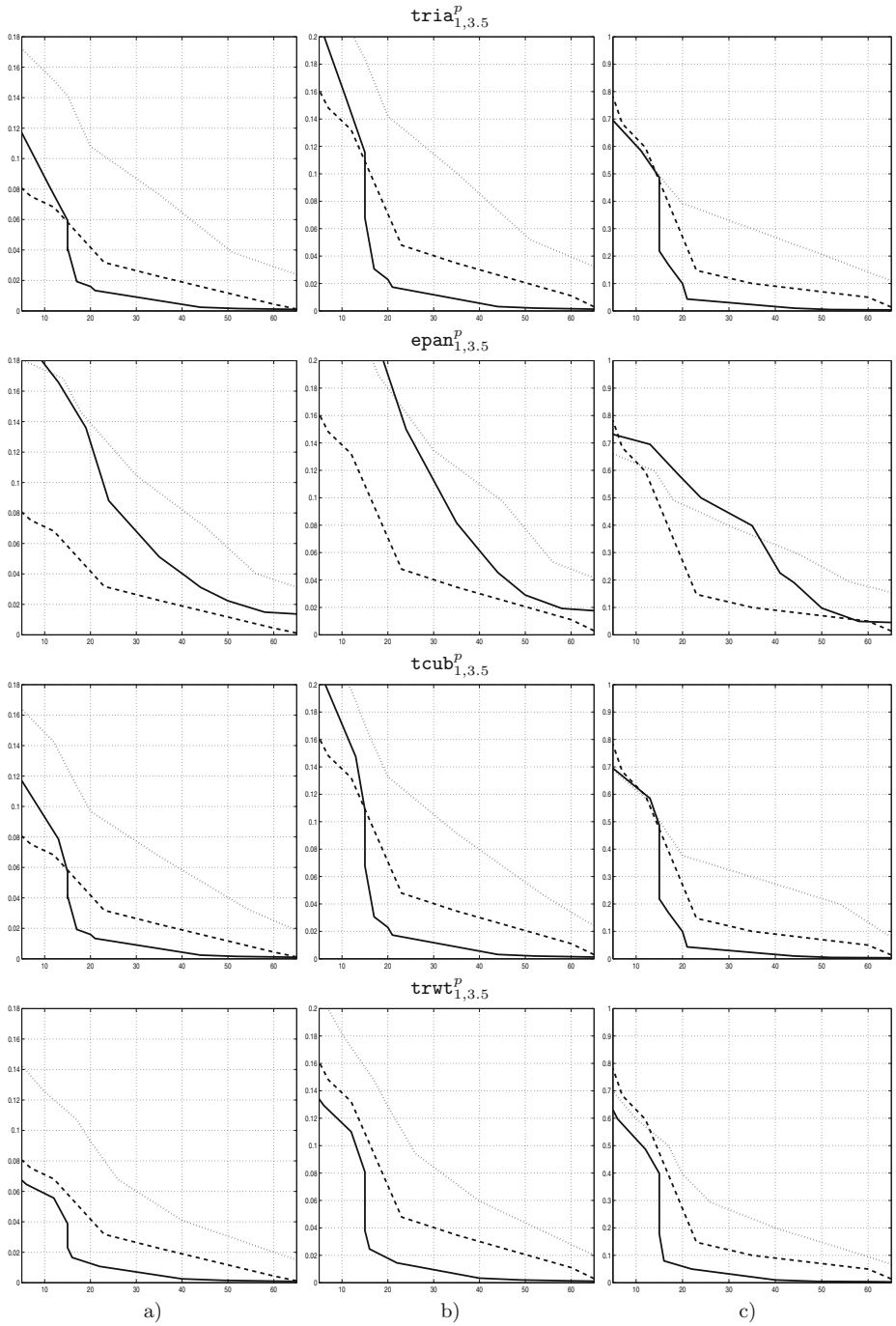
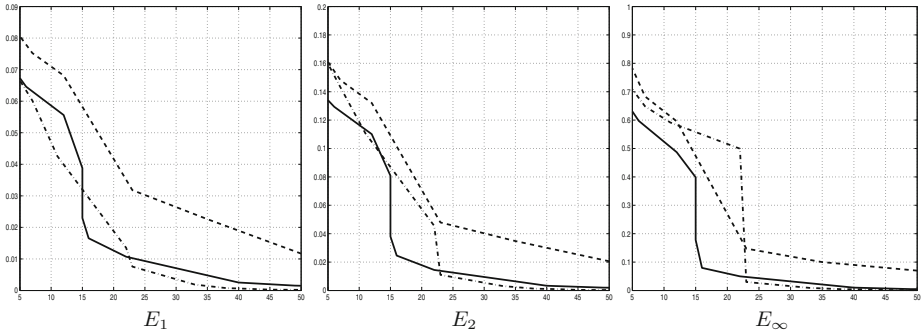


Fig. 4. Errors vs the number of details NNZ using MR schemes to compress the function f_1 with PV (dashed line) and $\text{kern}_{1,3.5}^p$ with $p = 1$ (solid line) and $p = 2$ (dotted line): (a) E_1 , (b) E_2 (c) E_∞

Table 3. Function f_1 . Errors and number of details obtained with various compression schemes.

	E_1	E_2	E_∞	NNZ	r_c
PV	$4.41 \cdot 10^{-3}$	$1.11 \cdot 10^{-2}$	$4.98 \cdot 10^{-2}$	60	0.940
PPH	$1.35 \cdot 10^{-2}$	$4.56 \cdot 10^{-2}$	$4.99 \cdot 10^{-1}$	22	0.978
$\text{tria}_{1,3.5}^1$	$1.34 \cdot 10^{-2}$	$1.74 \cdot 10^{-2}$	$4.32 \cdot 10^{-2}$	21	0.979
$\text{epan}_{1,3.5}^1$	$1.49 \cdot 10^{-2}$	$1.92 \cdot 10^{-2}$	$4.90 \cdot 10^{-2}$	58	0.942
$\text{tcub}_{1,3.5}^1$	$1.33 \cdot 10^{-2}$	$1.73 \cdot 10^{-2}$	$4.32 \cdot 10^{-2}$	21	0.979
$\text{trwt}_{1,3.5}^1$	$1.07 \cdot 10^{-2}$	$1.44 \cdot 10^{-2}$	$4.97 \cdot 10^{-2}$	22	0.978
$\text{tria}_{1,3.5}^2$	$9.29 \cdot 10^{-3}$	$1.29 \cdot 10^{-2}$	$4.97 \cdot 10^{-2}$	96	0.905
$\text{epan}_{1,3.5}^2$	$9.68 \cdot 10^{-3}$	$1.33 \cdot 10^{-2}$	$4.70 \cdot 10^{-2}$	99	0.902
$\text{tcub}_{1,3.5}^2$	$1.09 \cdot 10^{-2}$	$1.39 \cdot 10^{-2}$	$4.94 \cdot 10^{-2}$	73	0.928
$\text{trwt}_{1,3.5}^2$	$9.87 \cdot 10^{-3}$	$1.26 \cdot 10^{-2}$	$3.97 \cdot 10^{-2}$	70	0.931

**Fig. 5.** Errors vs the number of details NNZ using MR schemes to compress the function f_1 with PV (dashed line) and $\text{trwt}_{1,3.5}^1$ (solid line) and PPH (dotted line)**Table 4.** Function f_1 . Errors and number of details obtained with various compression schemes.

	E_1	E_2	E_∞	NNZ	r_c
PV	$4.41 \cdot 10^{-3}$	$1.11 \cdot 10^{-2}$	$4.98 \cdot 10^{-2}$	60	0.940
PPH	$1.35 \cdot 10^{-2}$	$4.56 \cdot 10^{-2}$	$4.99 \cdot 10^{-1}$	22	0.978
$\text{trwt}_{0,3.5}^1$	$1.13 \cdot 10^{-2}$	$1.54 \cdot 10^{-2}$	$4.89 \cdot 10^{-2}$	37	0.963
$\text{trwt}_{1,3.5}^1$	$1.07 \cdot 10^{-2}$	$1.44 \cdot 10^{-2}$	$4.97 \cdot 10^{-2}$	22	0.978
$\text{trwt}_{2,3.5}^1$	$3.51 \cdot 10^{-3}$	$8.05 \cdot 10^{-3}$	$5.00 \cdot 10^{-2}$	55	0.945
$\text{trwt}_{3,3.5}^1$	$4.41 \cdot 10^{-3}$	$1.11 \cdot 10^{-2}$	$4.98 \cdot 10^{-2}$	60	0.940
$\text{trwt}_{4,3.5}^1$	$5.11 \cdot 10^{-3}$	$7.82 \cdot 10^{-3}$	$3.20 \cdot 10^{-2}$	61	0.939
$\text{trwt}_{5,3.5}^1$	$5.25 \cdot 10^{-3}$	$8.00 \cdot 10^{-3}$	$3.15 \cdot 10^{-2}$	61	0.939

The number of details different to zero, NNZ, is displayed. Also, to measure the compression capabilities of each scheme, the following factor is defined:

$$r_c = \frac{J_L - J_0 - |\mathcal{D}_\varepsilon|}{J_L - J_0} \quad (19)$$

Table 5. Function f_2 . Errors and number of details obtained with various compression schemes.

	E_1	E_2	E_∞	NNZ	r_c
PV	$2.50 \cdot 10^{-3}$	$4.02 \cdot 10^{-3}$	$9.99 \cdot 10^{-3}$	461	0.543
PPH	$2.65 \cdot 10^{-3}$	$4.10 \cdot 10^{-3}$	$9.97 \cdot 10^{-3}$	425	0.578
tria _{1,3.5} ¹	$2.91 \cdot 10^{-3}$	$4.35 \cdot 10^{-3}$	$9.97 \cdot 10^{-3}$	385	0.618
epan _{1,3.5} ¹	$2.89 \cdot 10^{-3}$	$4.34 \cdot 10^{-3}$	$9.98 \cdot 10^{-3}$	384	0.619
tcub _{1,3.5} ¹	$2.93 \cdot 10^{-3}$	$4.37 \cdot 10^{-3}$	$9.97 \cdot 10^{-3}$	378	0.625
trwt _{1,3.5} ¹	$2.81 \cdot 10^{-3}$	$4.27 \cdot 10^{-3}$	$9.97 \cdot 10^{-3}$	398	0.605
tria _{1,3.5} ²	$2.81 \cdot 10^{-3}$	$4.22 \cdot 10^{-3}$	$9.98 \cdot 10^{-3}$	384	0.619
epan _{1,3.5} ²	$2.75 \cdot 10^{-3}$	$4.16 \cdot 10^{-3}$	$9.99 \cdot 10^{-3}$	387	0.616
tcub _{1,3.5} ²	$2.76 \cdot 10^{-3}$	$4.20 \cdot 10^{-3}$	$9.99 \cdot 10^{-3}$	387	0.616
trwt _{1,3.5} ²	$2.76 \cdot 10^{-3}$	$4.18 \cdot 10^{-3}$	$9.99 \cdot 10^{-3}$	393	0.610

where $\mathcal{D}_\varepsilon = \{(j, k) : |\hat{d}_j^k| = |d_j^k|_\varepsilon > 0\}$. If $|\mathcal{D}_\varepsilon| = 0$ then the compression is maximum and $r_c = 1$. When $r_c = 0$ then the algorithm does not produce any compression.

We set the number of points used to approximate the curve and analyze three aspects: the degree of the polynomial, the weight function (Table 1, [26], we only use the most representative one) and the chosen loss function, $p = 1, 2$.

In the next experiment, we take $J_0 = 64$ and $J_L = 1024$, thus $L = 4$. We compare the errors obtained using different weight function using four points (similar to PV), i.e., $s = 2.5$ and the number of details NNZ stored for each MR scheme. The degree chosen for this example is $r = 1$. In this case, Fig. 3, we can observe that the results obtained are similar when **tria** and **epan** kernel functions and **tcub** and **trwt** are used. It is interesting to remark that, when $p = 1$, the errors obtained with our method versus the NNZ are smaller than those obtained with every LPR methods.

In the following experiments, we take $J_0 = 16$ and $J_L = 1024$, thus $L = 6$. In order to analyze the advantages of increasing the number of point without increasing the degree of polynomial used, we fix the bandwidth $s = 3.5$. In Table 3, we can see that when we use the LPR MR with ℓ^1 -norm we reduce the number of values different to zero. Indeed, better results are obtained when **tria**, **tcub** and **trwt** are used, being the latter the one that produces a most accurate prediction operator (see Fig. 4). It is significant that PPH considerably reduces the non-zero elements obtaining similar results as LPR methods with ℓ^1 -norm. However, the value of E_2 is reduced with **trwt**_{1,3.5}¹ ($1.44 \cdot 10^{-2}$) in comparison with PPH ($4.56 \cdot 10^{-2}$). It is displayed in Fig. 5. When ℓ^2 -norm is introduced, the LPR algorithms do not improve the results of the linear methods, as we can see in Fig. 4.

In Table 4, we show the results when the degree of polynomial change, $r = 1, \dots, 5$. When the degree is 3 or bigger than 3, the results are similar to using the linear method. It is quite interesting that when a constant ($r = 0$) is used to approximate, the results are better than using PV method.

In the last experiment, with the function f_2 (Table 5), in order to make a more realistic example, we suppose our measuring is not correctly produced. Therefore, we introduce a Gaussian noise with standard deviation $\sigma = 10^{-2}$ in each value of f_2 . Taking into account that the signal is quite difficult to compress, with LPR MR we obtain a more accurate prediction operator. Therefore, the number of non-zero elements decreases considerably preserving E_p with $p = 1, 2, \infty$. In this case, the PPH method slightly improves the results obtained using the PV method.

In conclusion, the predictor operator using as loss function the ℓ^1 -norm improves the results in comparison with the use of other methods, obtaining a high compression rate. However, we have to calculate the approximation in each point. Therefore the runtime increases.

6 Conclusions and Future Research

In this work, we extend the method developed by Aràndiga et al. in [11] based on Harten's framework [24] replacing the minimization of the ℓ^2 - norm by a robust norm ℓ^1 . The linearity of the method is lost, therefore, the stability has been ensured by the modification of the direct algorithm. The order of the method has been calculated and it has been used to compress signals with some conclusions:

- i. The method obtains a high compression rate in signals with slight noise that are difficult to compress.
- ii. The prediction operator adapts point to point.
- iii. A bigger number of points can be used to approximate the real value without enlarging the order of the polynomial used.
- iv. The weight function is an essential variable in these examples.

Several possibilities can be studied as the stability (without using error control), the use of the different weight functions depending on the signal or the set of functions to approximate the curve. Also, this method can be used in image processing as compression or denoising.

References

1. Amat, S., Aràndiga, F., Cohen, A., Donat, R.: Tensor product multiresolution analysis with error control for compact representation. *Signal Process.* **82**, 587–608 (2002)
2. Amat, S., Donat, R., Liandrat, J., Trillo, J.C.: A fully adaptive PPH multiresolution scheme for image processing. *Math. Comput. Modelling* **46**, 2–11 (2007)
3. Amat, S., Liandrat, J.: On the stability of the pph nonlinear multiresolution. *Appl. Comput. Harmon. Anal.* **18**(2), 198–206 (2005)
4. Amat, S., Moncayo, M.: Exact error bound for the reconstruction processes using interpolating wavelets. *Math. Comput. Simul.* **79**, 3547–3555 (2009)
5. Aràndiga, F., Belda, A., Mulet, P.: Point-value WENO multiresolution applications to stable image compression. *J. Sci. Comput.* **43**(2), 158–182 (2010)

6. Aràndiga, F., Cohen, A., Yáñez, D.F.: Learning-based multiresolution schemes with application to compression of images. *Signal Process.* **93**, 2474–2484 (2013)
7. Aràndiga, F., Donat, R.: Nonlinear multiscale decompositions: the approach of A. Harten. *Numer. Algorithm.* **23**, 175–216 (2000)
8. Aràndiga, F., Donat, R.: Stability through synchronization in nonlinear multiscale transformation. *SIAM J. Sci. Comput.* **29**, 265–289 (2007)
9. Aràndiga, F., Donat, R., Harten, A.: Multiresolution based on weighted averages of the hat function I: linear reconstruction technique. *SIAM J. Numer. Anal.* **36**, 160–203 (1998)
10. Aràndiga, F., Donat, R., Harten, A.: Multiresolution based on weighted averages of the hat function II: Nonlinear reconstruction technique. *SIAM J. Numer. Anal.* **20**, 1053–1093 (1999)
11. Aràndiga, F., Yáñez, D.F.: Generalized wavelets design using kernel methods. Application to signal processing. *J. Comput. Appl. Math.* **250**, 1–15 (2013)
12. Aràndiga, F., Yáñez, D.F.: Cell-average multiresolution based on local polynomial regression. Application to image processing. *Appl. Math. Comput.* **245**, 1–16 (2014)
13. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, New York (2004)
14. Chan, T.F., Golub, G.H., Mulet, P.: A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Numer. Anal.* **20**, 1964–1977 (1997)
15. Cleveland, W.S., Devlin, S.J.: Locally weighted regression: an approach to regression analysis by local fitting. *J. Amer. Stat. Assoc.* **83**, 596–610 (1988)
16. Cohen, A.: *Numerical Analysis of Wavelet Methods*. Springer, New York (2003)
17. Cohen, A., Daubechies, I., Feauveau, J.: Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.* **45**, 485–560 (1992)
18. Dahmen, W.: Stability of multiscale transformations. *J. Fourier Anal. Appl.* **2**(4), 341–361 (1996)
19. Fan, J., Gijbels, I.: *Local Polynomials Modelling and its Applications*. Chapman and Hall, London (1996)
20. Getreuer, P., Meyer, F.: ENO multiresolution schemes with general discretizations. *SIAM J. Numer. Anal.* **46**, 2953–2977 (2008)
21. Grant, M., Boyd, S.: Graph implementations for nonsmooth convex programs. In: Blondel, V., Boyd, S., Kimura, H. (eds.) *Recent Advances in Learning and Control (a tribute to M. Vidyasagar)*. LNCSS, pp. 95–110. Springer, Heidelberg (2008)
22. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming (web page and software), (2009) <http://stanford.edu/~boyd/cvx>
23. Harten, A.: Discrete multiresolution analysis and generalized wavelets. *J. Appl. Numer. Math.* **12**, 153–192 (1993)
24. Harten, A.: Multiresolution representation of data: General framework. *SIAM J. Numer. Anal.* **33**, 1205–1256 (1996)
25. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*. Springer, New York (2009)
26. Loader, C.: *Local Regression and Likelihood*. Springer, New York (1999)
27. MATLAB. version 7.10.0 (R2010a). The MathWorks Inc., Natick (2010)
28. Wand, M.P., Jones, M.C.: *Kernel Smoothing*. Chapman and Hall, London (1995)