SMOOTH DESIGN-ADAPTED WAVELETS FOR HALF-REGULAR DESIGNS IN TWO DIMENSIONS

V. DELOUILLE and R. von SACHS

Smooth Design-adapted Wavelets for Half-regular Designs in Two Dimensions

V. Delouille[1,3] R. von Sachs[2,3]

August 27, 2002

Abstract

In this paper, we treat nonparametric estimation of a regression function defined on a half-regular grid. By 'half-regular' grid, we mean a grid which is the Cartesian product of two irregular one dimensional grids. We use a wavelet-type estimator, based on a wavelet transform which is the tensor product of two one-dimensional design-adapted wavelet transforms. A denoising scheme is proposed, yielding a nonlinear wavelet estimator. We compare its performance with the one of a locally weighted regression procedure through a simulation study.

Keywords: nonparametric regression, tensor products, separable wavelets, design-adapted wavelet, denoising scheme.

1 Introduction

This paper extends the wavelet denoising method for stochastic design presented in [9] to the two-dimensional setting. In [9], the authors considered the model

$$Z_t = m(X_t) + \epsilon_t, t = 1, \ldots, n, \tag{1.1}$$

where the regressors X_t are random variables with common distribution $F_X(x)$, (X_t, Z_t) are independent pairs of observations and the errors ϵ_t are independent of the regressors with $E(\epsilon_t) = 0; \mathrm{Var}(\epsilon_t) < \infty$. Using the lifting scheme framework, biorthogonal wavelets in $\mathbf{L}_2(d\hat{F}_n)$ were constructed, where $\widehat{F}_n$ is the empirical distribution function of the regressors. These 'design-adapted' wavelets, combined with an appropriate thresholding method, provide a wavelet estimator for the regression function $m(x) = E(Z|X = x)$.

In the present paper, we generalize to a half-regular bidimensional grid the use of the method presented in [9]. We consider the model

$$Z_{s,t} = m(X_s, Y_t) + \epsilon_{s,t}, s = 1, \ldots, n_1,\ t = 1, \ldots, n_2,\ n_1 \cdot n_2 = n, \tag{1.2}$$

[1]Research Assistant of the National Fund for Scientific Research – Wallonie, Belgium (F.N.R.S.). Université catholique de Louvain, Institut de Statistique, Louvain-la-Neuve, Belgium

[2]Professor. Université catholique de Louvain, Institut de Statistique, Louvain-la-Neuve, Belgium

[3]Financial support from the contract 'Projet d'Actions de Recherche Concertées' nr. 98/03–217 from the Belgian government, and from the IAP research network nr. P5/24 of the Belgian State (Federal Office for Scientific, Technical and Cultural Affairs) is gratefully acknowledged.

where the $\{X_s\}$ and $\{Y_t\}$ are random variables having a distribution function $F_X(x)$ and $F_Y(y)$, respectively. Here also, the errors are supposed to be independent of the regressors, with $E(\epsilon_{s,t}) = 0$, $\mathrm{Var}(\epsilon_{s,t}) < \infty$. For the clarity of the presentation, we further assume that these errors are homoscedastic, that is $\mathrm{Var}(\epsilon_{s,t}) = \sigma_\epsilon^2$. However, we indicate how to generalise the method in case of heteroscedastic errors. Our target function is the conditional mean $m(x,y) = E[Z|X = x, Y = y]$.

This paper is organized as follows. In Section 2, we recall the basics of the univariate design-adapted wavelet transform developed in [9], and we explain how to obtain a wavelet estimator in this case. Section 3 introduces the separable multiresolution analysis of some weighted spaces, and gives the corresponding separable wavelet basis. A wavelet transform adapted to the half-regular grid of the model (1.2) is found by using tensor products of the one-dimensional (1D) design-adapted wavelet transform, as it is explained in Section 4. We specify some thresholding scheme in Section 5 in order to obtain a nonlinear wavelet estimator. Finally, Section 6 compare this estimator with a locally weighted regression procedure through a simulation study. Note that, in case the grid is completely irregular, another type of wavelet transform must be used, see [15, 8].

2 Preliminaries: design-adapted wavelet transform in one dimension

2.1 Multiresolution analysis and biorthogonal basis

Consider a *multiresolution analysis* (*MRA*) in a general weighted space $\mathbf{L}_2(\mu)$, that is, a strictly increasing and dense sequence $\mathcal{V} = \{V_j\}_{j \geq j_0}$ of closed subspaces of $\mathbf{L}_2(\mu)$,

$$V_j \subset V_{j+1}\,,\ j \in \mathbb{Z}\,, \quad \text{and} \quad \operatorname{clos} \bigcup_{j=j_0}^{\infty} V_j = \mathbf{L}_2(\mu) \ \text{ for any } \ j_0 \in \mathbb{N}\,.$$

Between successive spaces in $\mathcal{V}$, construct algebraic complements W_j so that

$$V_{j+1} = V_j \oplus W_j\ ,$$

where '$\oplus$' denotes the inner sum of disjoint linear spaces. The complement space W_j is not necessarily orthogonal to V_j. A fine-resolution space V_j can then be written as a telescopic decomposition into a coarser-resolution space and intermediate complement spaces,

$$V_j = V_{j_0} \oplus \bigoplus_{i=j_0}^{j-1} W_i\ . \tag{2.1}$$

With the notational convention that $W_{j_0-1} := V_{j_0}$, we call the sequence $\{W_j\}_{j \geq j_0-1}$ a *multiscale decomposition* (*MSD*).

In the one-dimensional setting, the spaces V_j and W_j are equipped with bases,

$$\begin{aligned} V_j &= \operatorname{clos}_{\mathbf{L}_2} \operatorname{span} \Phi_j, \quad \Phi_j := \{\, \varphi_{jk} \mid k \in \mathcal{K}_j \,\}\ , \\ W_j &= \operatorname{clos}_{\mathbf{L}_2} \operatorname{span} \Psi_j, \quad \Psi_j := \{\, \psi_{jm} \mid m \in \mathcal{M}_j \,\}\ , \end{aligned}$$

with $\mathcal{K}_j$ and $\mathcal{M}_j$ some index sets; on the interval we use ranges of integers. We refer to any basis Φ_j for the space V_j as a set of *scaling functions*, and any basis Ψ_j for any type of complement space W_j, $j \geq 0$, is called a set of *wavelets* at level j.

Due to the multiresolution structure of the spaces V_j and W_j, there exist refinement coefficients $\{h_{jlk}\}$ and $\{g_{jlm}\}$ such that

$$\varphi_{jk} = \sum_{l\in\mathcal{K}_{j+1}} h_{jlk}\,\varphi_{j+1,l}\ , \quad \psi_{jm} = \sum_{l\in\mathcal{K}_{j+1}} g_{jlm}\,\varphi_{j+1,l}\ . \tag{2.2}$$

It is convenient to rewrite the refinement relations (2.2) as matrix expressions

$$\Phi_j = \Phi_{j+1}H_j \quad \text{and} \quad \Psi_j = \Phi_{j+1}G_j\ , \tag{2.3}$$

where Φ_j is a row vector containing the functions $\{\varphi_{jk}, k \in \mathcal{K}_j\}$. If a_j is the vector of coefficients of an element of V_j in the basis Φ_j, then the vector of coefficients of the same element in the basis Φ_{j+1} is given by $a_{j+1} = H_j a_j$. When repeated, this is a (nonstationary) *subdivision scheme*, see [3, 5, 17].

Biorthogonal bases generalise orthogonal bases in that different bases are used for analysis and for synthesis of the signal: a primal MSD, generated by the basis $\Phi_{j_0} \cup \bigcup_{j=j_0}^{\infty} \Psi_j$ is used to reconstruct the signal, whereas a dual MSD, with spaces $\tilde{V}_{j_0}$ and $\tilde{W}_j$ spanned by $\tilde{\Phi}_{j_0}$ and $\tilde{\Psi}_j$, respectively, is used for decomposition. Both MSD's must be linked through biorthogonality relations, i.e.,

$$\begin{array}{ll} \langle \varphi_{jk}\,, \tilde{\varphi}_{jk'} \rangle = \delta_{kk'}\ \forall\, j\ , & \langle \varphi_{jk}\,, \tilde{\psi}_{jm'} \rangle = 0\ \forall\, j, k, m'\ , \\ \langle \psi_{jm}\,, \tilde{\varphi}_{jk'} \rangle = 0\ \forall\, j, m, k'\ , & \langle \psi_{jm}\,, \tilde{\psi}_{jm'} \rangle = \delta_{mm'}\ \forall\, j\ , \end{array} \tag{2.4}$$

where $\delta_{kk'}$ is the Kronecker symbol. If there exists a dual MSD satisfying (2.4), the resulting pair of biorthogonal wavelet bases allows the following decomposition of any function $m(.)$.

$$m(x) = \sum_k s_{j_0,k}\varphi_{j_0,k}(x) + \sum_{j,k} d_{j,k}\psi_{j,k}(x)\ , \tag{2.5}$$

where the scaling coefficients are $s_{j_0,k} = \langle m\,, \tilde{\varphi}_{j_0,k}\rangle$ and the wavelet coefficients are $d_{j,k} = \langle m\,, \tilde{\psi}_{jk}\rangle$.

The dual refinement operators $\tilde{H}_j$ and $\tilde{G}_j$ are defined analogously to (2.3):

$$\tilde{\Phi}_j = \tilde{\Phi}_{j+1}\tilde{H}_j \quad \text{and} \quad \tilde{\Psi}_j = \tilde{\Phi}_{j+1}\tilde{G}_j\ . \tag{2.6}$$

Filling in the refinement relations in (2.4) gives the conditions to have *biorthogonal filters*

$$\begin{array}{ll} \tilde{H}_j^* H_j = \mathbf{1}\ , & \tilde{G}_j^* H_j = 0\ , \\ \tilde{H}_j^* G_j = 0\ , & \tilde{G}_j^* G_j = \mathbf{1}\ , \end{array} \tag{2.7}$$

where $\mathbf{1}$ is the identity matrix.

The order of a MRA in the second-generation setting is defined as follows.

Definition 2.1. *The* (polynomial) order *of a univariate MRA is given by*

$$\tilde{N} := \max\{\, n \mid \exists j_\star \text{ such that } \forall j \geq j_\star : \Pi_{n-1} \subset V_j\, \}\ ,$$

where Π_{n-1} *is the space of polynomials of degree at most* $n-1$.

When the order of a MRA is $\tilde{N}$ and an associated dual MSD exists, the analysing wavelets $\tilde{\psi}_{jk}$ with $j \geq j_\star$ have $\tilde{N}$ vanishing moments, i.e., the space $\tilde{W}_j$ is orthogonal to $\Pi_{\tilde{N}-1}$ for $j \geq j_\star$. Symmetrically, the primal wavelets are said to have N vanishing moments if, for j greater than a given level j', W_j is orthogonal to Π_{N-1}. Note that, in order to have a sparser representation of a smooth function, only $\tilde{N}$ needs to be increased.

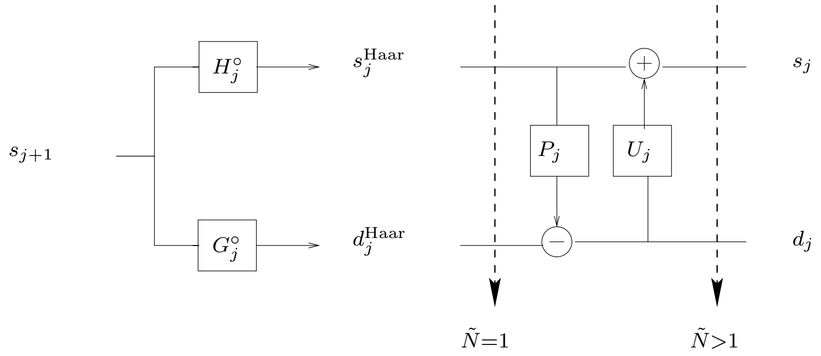


Figure 1: Filter bank diagram for the analysing design-adapted wavelet transform in one dimension. A Haar transform is performed first, producing detail and scaling coefficients corresponding to the initial wavelet basis of order $\tilde{N} = 1$. This basis is further improved by two lifting steps: the prediction operator P_j produces detail coefficients with better approximation properties, the update operator U_j improves the stability of the transform.

2.2 One-dimensional wavelet transform adapted to the design

As said before, one of the achievement of [9] is to construct biorthogonal filters which automatically adapt to whatever type of design (irregular, stochastic), and to small and non-dyadic sample sizes. Moreover, these filters furnish a wavelet transform that is adapted to the interval, that is, no particular treatment of the boundary is necessary.

We now recall the main features of the method presented in [9] to build biorthogonal filters in $\mathbf{L}_2(d\hat{F}_n)$

The construction starts from the unbalanced Haar wavelet basis of $\mathbf{L}_2(d\widehat{F}_n)$ [13, 7]. This basis is orthogonal in $\mathbf{L}_2(d\widehat{F}_n)$ and the corresponding Haar-like transform is equivalent to performing the classical Haar transform on the observations, ignoring thus the irregularities of the grid.

Next the *lifting scheme* [6, 22, 23] is used to provide a higher order for the primal MRA. In this process, orthogonality of the filters is relaxed to biorthogonality. Two lifting steps are used: a prediction and an update step. The prediction step increases the order $\tilde{N}$ of the primal MRA from 1 to $2d+1$, for a given natural d. This is achieved through an average-interpolation scheme [10, 21, 24] in $\mathbf{L}_2(d\widehat{F}_n)$. After the average-interpolating prediction step, the spaces V_j and W_j are no longer orthogonal. It is possible to reduce this nonorthogonality by performing an update step as described in [20]. This update step reduces the correlation between coefficients belonging to different levels and improves the stability of the resulting wavelet basis. A simulation study presented in [9] shows that this stabilisation induces a clear improvement of the performance of the resulting estimator. This improvement is apparent in particular near the boundary. We refer the reader to [9] for the implementation details of this biorthogonal wavelet transform in $\mathbf{L}_2(d\widehat{F}_n)$.

In the sequel, when talking about pairs of biorthogonal filters in one dimension, we will always refer to the filters obtained by a Haar transform followed by the above mentioned prediction and update steps.

2.3 Nonlinear threshold estimator in one dimension

The previous section provided us with a wavelet transform that automatically adapts to the irregular design at hand. We now indicate how to obtain a wavelet-based estimator for the model (1.1).

We define the empirical wavelet coefficients of the biorthognal basis obtained with the lifting scheme as follows.

$$\widehat{d}_{j,k} = \frac{1}{n}\sum_{i=1}^{n} Y_i \tilde{\psi}_{j,m}(X_i), \qquad \widehat{s}_{j,k} = \frac{1}{n}\sum_{i=1}^{n} Y_i \tilde{\varphi}_{j,k}(X_i). \tag{2.8}$$

At finest level $J = \lceil \log_2(n) \rceil$, the functions $\tilde{\varphi}_{J,k}(x)$ and $\varphi_{J,k}(x)$ are indicators functions, normalised in $\mathbf{L}_2(d\hat{F}_n)$. Hence the scaling coefficients $\{\hat{s}_{J,k}\}$ are just the normalised observations, $\hat{s}_{J,k} := Z_k/\sqrt{n}$. Next, the analysing filter pair $(\tilde{H}_j, \tilde{G}_j)$ decomposes the scaling coefficients $\hat{s}_{j+1}$ between

$$\hat{s}_j = \tilde{H}_j^* \hat{s}_{j+1} \quad \text{and} \quad \hat{d}_j = \tilde{G}_j^* \hat{s}_{j+1} \text{ for } j = J-1, \ldots, j_0, \tag{2.9}$$

where j_0 is the primary resolution level. The corresponding biorthogonal filter pair (H_j, G_j) recomposes the signal

$$\hat{s}_{j+1} = H_j \hat{s}_j + G_j \hat{d}_j, \qquad j = j_0, \ldots, J-1 \ . \tag{2.10}$$

Let $\mathcal{W}$ be the matrix of the analysing transform. The model (1.1) can be written in the vector form $Z = m(X) + \epsilon$, where $Z = [Z_1, \ldots, Z_n]^T$, $m(X) = [m(X_1), \ldots, m(X_n)]^T$, $\epsilon = [\epsilon_1, \ldots, \epsilon_n]^T$. In the wavelet domain, we have

$$\mathcal{W}Z = \mathcal{W}m(X) + \mathcal{W}\epsilon \ , \tag{2.11}$$

where $\mathcal{W}Z$ is the vector containing the estimated scaling and wavelet coefficients in Mallat ordering, $\mathcal{W}Z = [\hat{s}_{j_0}, \hat{d}_{j_0}, \ldots, \hat{d}_{J-1}]^T$. For thresholding, we do not need to consider the variance of $\mathcal{W}m(X)$ in (2.11), but only the variance of the noise component $\mathcal{W}\epsilon$. When the errors are i.i.d $N(0, \sigma_\epsilon^2)$, the variance of $\mathcal{W}\epsilon$ is

$$\mathrm{Var}(\mathcal{W}\epsilon) = \sigma_\epsilon^2 \mathcal{W}\mathcal{W}^T \ .$$

A natural way of estimating σ_ϵ^2 is to consider a modified version of the median absolute deviation from the median (MAD) estimator, similarly to [16]. Let ω_{J-1} be the elements of the diagonal of $\mathcal{W}\mathcal{W}^T$ corresponding to the wavelet coefficients d_{J-1}. The variance of the element of $\mathcal{W}\epsilon$ that corresponds to the detail coefficient $\widehat{d}_{jk}$ can be estimated as $\hat{\sigma}_{jk}^2$ where

$$\hat{\sigma}_{jk}^2 \quad = \quad \hat{\sigma}_{\mathrm{MAD}}^2 \, \mathrm{diag}(\mathcal{W}\mathcal{W}^T)_{jk} \ , \tag{2.12}$$

$$\hat{\sigma}_{\mathrm{MAD}} \quad = \quad \mathrm{MAD}\left(\frac{\hat{d}_{J-1}}{\sqrt{\omega_{J-1}}} \right) /0.6745 \ . \tag{2.13}$$

The direct generalization of the VisuShrink threshold [11] is given by $t_{jk} := \sigma_{jk}\sqrt{2\log n}$. This produces the nonlinear wavelet estimator:

$$\hat{m}(x) = \sum_k \hat{s}_{j_0,k} \varphi_{j_0,k}(x) + \sum_{j=j_0}^{J-1} \sum_k \eta_{(\cdot)}(\hat{d}_{jk}, t_{jk}) \psi_{jk}(x) \ , \tag{2.14}$$

where $\eta_{(\cdot)}(w, \lambda)$ denotes either 'hard' or 'soft' threshold nonlinearities:

$$\eta_H(w, \lambda) = w 1_{\{|w| > \lambda\}} \ ; \quad \eta_S(w, \lambda) = \mathrm{sign}(w)(|w| - \lambda)_+ \ . \tag{2.15}$$

3 Separable multiresolution analysis and biorthogonal wavelet basis

Consider now two one-dimensional multiresolution analyses of some weighted spaces, that is, let $\mathcal{V}^x = \{V_j^x\}$ and $\mathcal{V}^y = \{V_j^y\}$ be two MRA's of $\mathbf{L}_2(\mu_1)$ and $\mathbf{L}_2(\mu_2)$, respectively. The tensor product between the spaces V_j^x and V_j^y is defined by

$$\mathbf{V}_j = V_j^x \otimes V_j^y = \mathrm{clos}_{\mathbf{L}_2(\mu_1 \otimes \mu_2)} \left(\mathrm{span}\{a(x,y) = b(x)c(y), b \in V_j^x, c \in V_j^y\} \right) ,$$

where $\mu_1 \otimes \mu_2$ denotes the product measure of μ_1 and μ_2. The set $\mathcal{V} := \{\mathbf{V}_j\}$ forms a MRA of the space $\mathbf{L}_2(\mu_1 \otimes \mu_2)$, satisfying

$$\mathbf{V}_j \subset \mathbf{V}_{j+1} \, , \; j \in \mathbb{Z} \, , \quad \text{and} \quad \mathrm{clos} \bigcup_{j=j_0}^{\infty} \mathbf{V}_j = \mathbf{L}_2(\mu_1 \otimes \mu_2) \;\; \text{for any} \;\; j_0 \in \mathbb{N} \, .$$

It follows that, if $\Phi_j^x = \{\varphi_{jk}(x), \, k \in \mathcal{K}_j^x\}$ and $\Phi_j^y = \{\varphi_{jk}(y), \, k \in \mathcal{K}_j^y\}$ are bases of V_j^x and V_j^y, respectively, then the functions

$$\varphi_{j,k_1,k_2}(x,y) = \varphi_{j,k_1}(x)\varphi_{j,k_2}(y), k_1 \in \mathcal{K}_j^x, k_2 \in \mathcal{K}_j^y$$

provide a basis for $\mathbf{V}_j$. The sets $\mathcal{K}_j^x, \mathcal{M}_j^x$ (resp. $\mathcal{K}_j^y, \mathcal{M}_j^y$) are index sets on the $x-$axis (resp. $y-$axis).

We construct, for each j, algebraic complement spaces $\mathbf{W}_j$ so that

$$\mathbf{V}_{j+1} = \mathbf{V}_j \oplus \mathbf{W}_j \; .$$

By decomposition of the tensor product, we have [4, 18]

$$\begin{aligned} \mathbf{V}_{j+1} &= V_{j+1}^x \otimes V_{j+1}^y \\ &= \left(V_j^x \oplus W_j^x\right) \otimes \left(V_j^y \oplus W_j^y\right) \\ &= \left(V_j^x \otimes V_j^y\right) \oplus \left(V_j^x \otimes W_j^y\right) \oplus \left(W_j^x \otimes V_j^y\right) \oplus \left(W_j^x \otimes W_j^y\right) \\ &= \mathbf{V}_j \oplus \left(\mathbf{W}_j^h \oplus \mathbf{W}_j^v \oplus \mathbf{W}_j^d\right) \qquad (3.1) \\ &= \mathbf{V}_j \oplus \mathbf{W}_j \end{aligned}$$

Hence the space $\mathbf{W}_j$ consists of three parts, with bases given by

$$\begin{aligned} &\bullet \quad \psi_{j,k_1,k_2}^v(x,y) = \psi_{j,k_1}(x)\varphi_{j,k_2}(y), k_1 \in \mathcal{M}_j^x, k_2 \in \mathcal{K}_j^y \text{ for } W_j^x \otimes V_j^y \\ &\bullet \quad \psi_{j,k_1,k_2}^h(x,y) = \varphi_{j,k_1}(x)\psi_{j,k_2}(y), k_1 \in \mathcal{K}_j^x, k_2 \in \mathcal{M}_j^y \text{ for } V_j^x \otimes W_j^y \qquad (3.2) \\ &\bullet \quad \psi_{j,k_1,k_2}^d(x,y) = \psi_{j,k_1}(x)\psi_{j,k_2}(y), k_1 \in \mathcal{M}_j^x, k_2 \in \mathcal{M}_j^y \text{ for } W_j^x \otimes W_j^y \; . \end{aligned}$$

Then $\{\mathbf{\Psi}_j^\alpha, \alpha = h, v \text{ or } d\}$ is a basis for $\mathbf{W}_j$, where $\mathbf{\Psi}_j^\alpha$ is the matrix with elements $\psi_{j,k_1,k_2}^\alpha(x,y)$.

The superscripts h, v, d stand for 'horizontal', 'vertical', and 'diagonal', respectively. Indeed, the wavelet coefficients corresponding to $\mathbf{\Psi}_j^v(x,y)$ (resp. $\mathbf{\Psi}_j^h(x,y)$) indicate the fluctuations in the x (resp. y) direction, that is, they are particularly sensitive to a vertical (resp. horizontal) edge on the graph of a function to be decomposed. Next, if there are diagonal edges in the function $m(x,y)$, these will induce large absolute values of the detail coefficients associated to $\mathbf{\Psi}_j^d(x,y)$.

Suppose we have, in the one-dimensional setting, two MSD's which satisfy the biorthogonality relations in (2.4). To obtain separable biorthogonal bases of $\mathbf{L}_2(\mu_1 \otimes \mu_2)$,we repeat the decomposition of the spaces as in (3.1). This yields the following dual wavelets.

$$\begin{aligned}
\tilde{\psi}^v_{j,k_1,k_2}(x,y) &= \tilde{\psi}_{j,k_1}\tilde{\varphi}_{j,k_2}(y),\ k_1 \in \mathcal{M}^x_j, k_2 \in \mathcal{K}^y_j \\
\tilde{\psi}^h_{j,k_1,k_2}(x,y) &= \tilde{\varphi}_{j,k_1}(x)\tilde{\psi}_{j,k_2}(y), k_1 \in \mathcal{K}^x_j, k_2 \in \mathcal{M}^y_j \\
\tilde{\psi}^d_{j,k_1,k_2}(x,y) &= \tilde{\psi}_{j,k_1}(x)\tilde{\psi}_{j,k_2}(y), k_1 \in \mathcal{M}^x_j, k_2 \in \mathcal{M}^y_j\ .
\end{aligned}$$

Similarly to the one-dimensional case, it is possible to decompose any function $m \in \mathbf{L}_2(\mu_1 \otimes \mu_2)$ as follows.

$$\begin{aligned}
m(x,y) &= \sum_{k_1,k_2} \langle m\,, \tilde{\varphi}_{j_0,k_1,k_2}\rangle\, \varphi_{j_0,k_1,k_2}(x,y) \\
&+ \sum_{\alpha\in\{h,v,d\}} \sum_j \sum_{k_1,k_2} \left\langle m\,, \tilde{\psi}^\alpha_{j,k_1,k_2}\right\rangle \psi^\alpha_{j,k_1,k_2}(x,y)\ ,
\end{aligned}$$

where $\left\langle m\,, \tilde{\psi}^\alpha_{j,k_1,k_2}\right\rangle = \int m(x,y)\tilde{\psi}_{j,k_1,k_2}(x,y)d\mu_1(x)d\mu_2(y),\ \ \alpha \in \{v,h,d\}$ and similarly for $\langle m\,, \tilde{\varphi}_{j_0,k_1,k_2}\rangle$.

4 Two-dimensional wavelet transform adapted to the design

We now extend to two dimensions the design-adapted wavelet transform described by equations (2.9) and (2.10).

Daubechies [4] first proposes to treat images, that is, functions defined on a regular two-dimensional grid, by applying two times a one-dimensional wavelet transform: first on the 'rows' of the matrix of observations, then on the 'columns'. The extension presented in this paper completely parallels this two-dimensional transform of images [4, 18]. The only difference is that here *design-adapted* filters are used to process the 'rows' and the 'columns'.

In the model (1.2), let $J_1 := \lceil \log_2(n_1)\rceil$ (resp. $J_2 := \lceil \log_2(n_2)\rceil$) be the finest scale in the grid composed of the $\{X_s\}$ (resp. $\{Y_y\}$). Let $j_1 := J_1 - a, j_2 := J_2 - a, a = 1, 2, \ldots, M$, where a represents the depth in the layer of the MRA, and M is the maximum depth in the decomposition. Since we take the same value of a for j_1 and j_2, these two level indices represent the same depth in the MRA layer, just as in the classical algorithm used in image analysis [4, 18].

Remark 4.1. Let $n_1 = n_2$ in this note. In [19], Neumann and von Sachs considered the problem of estimating anisotropic functions with wavelets. When the function to be estimated has a different smoothness along the $x-$ and the $y-$axis, it is no more optimal to consider the wavelet basis given by the union of $\{\varphi_{j_0,k_1,k_2}(x,y)\}_{k_1,k_2}$ and of the expressions in (3.2), for $j = J-1, J-2, \ldots, j_0$ where $j_0 = J - M$ and $J := \lceil \log_2(n_1)\rceil$.

Instead, one should consider all the possible combinations $\psi_{j,j',k_1,k_2}(x,y) = \psi_{j,k_1}(x)\psi_{j',k_2}(y)$ for different j, j'. More precisely, the following basis should be used in case of anisotropic functions.

$$\begin{aligned}
&\{\varphi_{j_0,k_1}(x)\varphi_{j_0,k_2}(y)\}_{k_1,k_2} \cup \left(\cup_{j\geq j_0} \left\{\psi_{j,k_1}(x)\varphi_{j_0,k_2}(y)\right\}_{k_1,k_2}\right) \\
\cup\ &\left(\cup_{j'\geq j_0} \left\{\varphi_{j_0,k_1}(x)\psi_{j',k_2}(y)\right\}_{k_1,k_2}\right) \cup \left(\cup_{j,j'\geq j_0} \left\{\psi_{j,k_1}(x)\psi_{j',k_2}(y)\right\}_{k_1,k_2}\right)\ .
\end{aligned} \tag{4.1}$$

The estimator constructed with this basis attains the minimax rate of convergence for functions belonging to anisotropic Sobolev classes. An intuitive explanation is the following. With this last basis, the compression of the data will be better because this basis can better pick up the direction where there is some rapid change in the function, that is, where some wavelet coefficients are needed to represent the fluctuations of the function. Thus, if a function is flat in one direction, the basis (4.1) will concentrate all its effort on the other direction, and describe the function in the flat direction with the basis function $\varphi_{j_0,k_1}(x)\varphi_{j_0,k_2}(y)$ only. Note that we could also use a design-adapted basis of the form (4.1) for half-regular grids, but in this paper, we will concentrate on the other basis, given in (3.2).

Our bi-dimensional transform uses the filters $\left\{(\tilde{H}_{j_1},\tilde{G}_{j_1}),(H_{j_1},G_{j_1})\right\}_{j_1=J_1-1}^{J_1-M}$ that are adapted to the grid defined by the $\{X_s, s=1,\ldots,n_1\}$ for the decomposition along the $x-$axis, and the filters $\left\{(\tilde{H}_{j_2},\tilde{G}_{j_2}),(H_{j_2},G_{j_2})\right\}_{j_2=J_2-1}^{J_2-M}$ adapted to $\{Y_t,\ t=1,\ldots,n_2\}$ for the decomposition along the $y-$axis.

Consider first the noise-free case, and the decomposition of the matrix $\{m(X_t,Y_s)\}$. Since there are n_2 regressors $\{Y_t\}$ and n_1 regressors $\{X_s\}$, the observations matrix $\{m(X_s,Y_t)\}$ is of size $(n_2\times n_1)$. We found it convenient to denote the matrix of scaling coefficients by S_{J_2,J_1}, where J_2 (resp. J_1) gives an indication of the number of rows (resp. columns) at level (J_2,J_1). Similarly, we will use the notation $D^{\alpha}_{j_2,j_1}$, $\alpha=\{v,h,d\}$ for the matrices of detail coefficients at level (j_2,j_1).

Since the scaling functions at the finest level are normalised indicator functions, the matrix S_{J_2,J_1} is composed of the normalized observations. Then the analysing transform proceeds as follows. For $(j_2,j_1)=(J_2-a,J_1-a), a=0,\ldots,M-1$, process first the rows as

$$\begin{aligned} A_{j_2,j_1-1} &= S_{j_2,j_1}\tilde{H}_{j_1-1} \\ B_{j_2,j_1-1} &= S_{j_2,j_1}\tilde{G}_{j_1-1}. \end{aligned} \tag{4.2}$$

The matrices A_{j_2,j_1-1} and B_{j_2,j_1-1} are then further transformed along the $y-$axis.

$$\begin{aligned} S_{j_2-1,j_1-1} &= \tilde{H}^*_{j_2-1}A_{j_2,j_1-1} \\ D^v_{j_2-1,j_1-1} &= \tilde{H}^*_{j_2-1}B_{j_2,j_1-1} \\ D^h_{j_2-1,j_1-1} &= \tilde{G}^*_{j_2-1}A_{j_2,j_1-1} \\ D^d_{j_2-1,j_1-1} &= \tilde{G}^*_{j_2-1}B_{j_2,j_1-1} \end{aligned} \tag{4.3}$$

Figure 2 illustrates this decomposition. Once the analysis is completed, we have a set of detail matrices $\left\{D^{\alpha}_{J_2-a,J_1-a}, \alpha\in\{h,v,d\}, a=1,\ldots,M\right\}$. To reconstruct the original signal, we use the synthesis filters, see Figure 3.

$$\begin{aligned} E^{(1)}_{j_2,j_1-1} &= H_{j_2-1}S_{j_2-1,j_1-1}+G_{j_2-1}D^h_{j_2-1,j_1-1} \\ E^{(2)}_{j_2,j_1-1} &= H_{j_2-1}D^v_{j_2-1,j_1-1}+G_{j_2-1}D^d_{j_2-1,j_1-1} \\ S_{j_2,j_1} &= E^{(1)}_{j_2,j_1-1}H^*_{j_1-1}+E^{(2)}_{j_2,j_1-1}G^*_{j_1-1}\ . \end{aligned}$$

When the signal is observed with noise, S_{J_2,J_1} is estimated by the matrix $\hat{S}_{J_2,J_1}$, which contains the normalized noisy observations $Z_{s,t}$ of the model (1.2). The detail matrices obtained by processing $\hat{S}_{J_2,J_1}$ are denoted by $\left\{\hat{D}^{\alpha}_{J_2-a,J_1-a}, \alpha\in\{h,v,d\}, a=1,\ldots,M\right\}$. In order to denoise the data, we use the thresholding scheme presented in the next section.

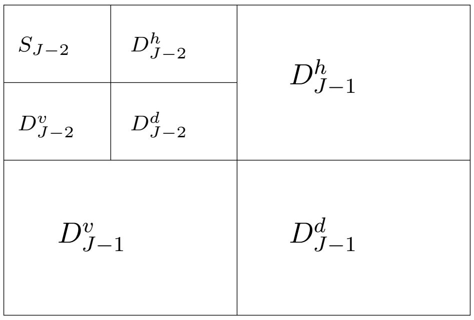


Figure 2: Schematic representation of the visualization of the two-dimensional wavelet transform. Here we have a decomposition of the original data into two layers. The notation of the level by j is a short-cut for (j_2, j_1), hence $J-1 \equiv (J_2-1, J_1-1)$.

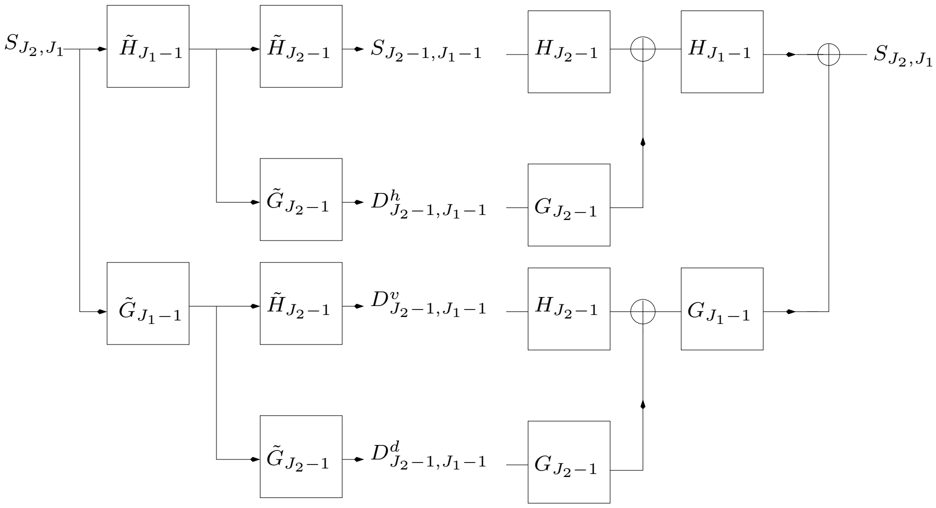


Figure 3: Two-dimensional filter bank diagram where the analysis and synthesis transform are represented.

5 Denoising process in two dimensions

5.1 Introduction

Having at our disposal a two-dimensional (2D) wavelet transform adapted to a half-regular grid, an estimator for the regression model (1.2) is found by thresholding the empirical detail coefficients, similarly to (2.14). Paralleling the 1D situation described in Section 2.3., we use a thresholding scheme based on a modified version of the VisuShrink threshold [11]. This scheme is fully determined once an estimate of the variance of each detail coefficient is found.

Looking at (2.12) and (2.13), we see that, in one dimension, two quantities intervene in the estimation of σ_{jk}: the diagonal elements of $\mathcal{W}\mathcal{W}^T$ and the fine detail coefficients $\hat{d}_{J-1}$ (to obtain $\hat{\sigma}_{\mathrm{MAD}}$). When $\sigma_\epsilon = 1$ in the model (1.1), the matrix $\mathcal{W}\mathcal{W}^T$ is the variance matrix of the empirical detail coefficients, and thus $\mathrm{diag}(\mathcal{W}\mathcal{W}^T)$ is a vector containing the variance of each detail coefficients. In two dimensions, the equivalent of $\mathcal{W}\mathcal{W}^T$ is a four dimensional array, denoted by $\mathbf{W}^4$, and the quantities of interest are the 'diagonal' elements of $\mathbf{W}^4$, that is, when $\sigma_\epsilon = 1$, the individual variance of each detail coefficients.

The amount of memory needed to store the whole array $\mathbf{W}^4$ is quite prohibitive. Hence we look for a good approximation of this structure under the constraint that the storage and computational cost should be small.

To compute the elements of $\mathbf{W}^4$, we use the following results of moments of multidimensional variates. If $\mathbf{x}, \mathbf{y}$ are random column vectors, a, b are constant vectors and E, F are constant matrices of an appropriate size, then

$$\begin{aligned} \mathrm{Cov}(a'\mathbf{x}, b'\mathbf{y}) = &\quad a' \,\mathrm{Cov}(\mathbf{x}, \mathbf{y}) b \\ \mathrm{Cov}(E\mathbf{x}, F\mathbf{y}) = &\quad E \,\mathrm{Cov}(\mathbf{x}, \mathbf{y}) F'. \end{aligned} \tag{5.1}$$

Section 5.2 shows how to compute the fine scale elements of $\mathbf{W}^4$, or, equivalently, how to compute the variance of each element in $\widehat{D}^\alpha_{J_2-1,J_1-1}$ when $\sigma_\epsilon = 1$. Section 5.3 indicates how to approximate the coarse-scale elements of $\mathbf{W}^4$. The wavelet estimator is given in Section 5.4, and we end this section by indicating a possible way to generalize the denoising method to deal with heteroscedastic errors.

5.2 Finest scale variance structure

We assume here that the errors in the model (1.2) are i.i.d. with $E(\epsilon_{s,t}) = 0$, $\mathrm{Var}(\epsilon_{s,t}) = 1$, and we give the formula for the variance of each elements in $\widehat{D}^\alpha_{J_2-1,J_1-1}$, $\alpha \in \{h, v, d\}$. Each row of $\hat{S}_{J_2,J_1}$ has a covariance matrix equal to $\mathbf{1}_{n_1,n_1}$ and each column of $\hat{S}_{J_2,J_1}$ has a covariance matrix equal to $\mathbf{1}_{n_2,n_2}$, where $\mathbf{1}_{n,m}$ indicates the $(n \times m)$ identity matrix. Let $\mathbf{M}_{(\cdot,i)}$ denote the $i-$th column of the matrix $\mathbf{M}$, whereas $\mathbf{M}_{(i,\cdot)}$ denotes the $i-$th row of $\mathbf{M}$. The variance structure of the $i-$th column of $A_{J_2,J_1-1} = \hat{S}_{J_2,J_1}\tilde{H}_{J_1-1}$ is computed using the relations (5.1). We have

$$\mathrm{Var}(A_{J_2,J_1-1,(\cdot,i)}) = \tilde{H}^*_{J_1-1(i,\cdot)}\tilde{H}_{J_1-1(\cdot,i)}\mathbf{1}_{n_2,n_2} \tag{5.2}$$

$$\mathrm{Cov}(A_{J_2,J_1-1,(\cdot,i)}, A_{J_2,J_1-1,(\cdot,l)}) = \tilde{H}^*_{J_1-1(i,\cdot)}\tilde{H}_{J_1-1(\cdot,l)}\mathbf{1}_{n_2,n_2} \tag{5.3}$$

$$\mathrm{Var}(B_{J_2,J_1-1,(\cdot,i)}) = \tilde{G}^*_{J_1-1(i,\cdot)}\tilde{G}_{J_1-1(\cdot,i)}\mathbf{1}_{n_2,n_2} \tag{5.4}$$

$$\mathrm{Cov}(B_{J_2,J_1-1,(\cdot,i)}, B_{J_2,J_1-1,(\cdot,l)}) = \tilde{G}^*_{J_1-1(i,\cdot)}\tilde{G}_{J_1-1(\cdot,l)}\mathbf{1}_{n_2,n_2}\ . \tag{5.5}$$

We now examine the effect of the transform on the columns (along the $y-$axis). First, we need to compute the variance matrix of each row of $\hat{S}_{J_2-1,J_1-1}$. This will allow us to compute the variance of the detail coefficient at the next coarser scale, see Section 5.3. The variance matrix of the $i-$th row of $\hat{S}_{J_2-1,J_1-1}$ is completely characterized by the following two equations, where $\mathbf{M}_{k,l}$ denotes the $(k,l)-$th element of the matrix $\mathbf{M}$.

$$\begin{aligned} \mathrm{Var}(\hat{S}_{J_2-1,J_1-1,(i,\cdot)})_{k,k} &= \tilde{H}^*_{J_2-1,(i,\cdot)} \mathrm{Var}(A_{J_2,J_1-1,(\cdot,k)})\tilde{H}_{J_2-1,(\cdot,i)} && (5.6)\\ \mathrm{Var}(\hat{S}_{J_2-1,J_1-1,(i,\cdot)})_{k,l} &= \tilde{H}^*_{J_2-1,(i,\cdot)} \mathrm{Cov}(A_{J_2,J_1-1,(\cdot,k)}, A_{J_2,J_1-1,(\cdot,l)})\tilde{H}_{J_2-1,(\cdot,i)}\ , && (5.7)\end{aligned}$$

where $\mathrm{Var}(A_{J_2,J_1-1,(\cdot,k)})$ and $\mathrm{Cov}(A_{J_2,J_1-1,(\cdot,k)}, A_{J_2,J_1-1,(\cdot,l)})$ are given by (5.2) and (5.3), respectively.

Second, we need to compute and store the diagonal elements of $\mathrm{Var}(\hat{D}^\alpha_{J_2-1,J_1-1,(.,i)})$, $\alpha \in \{v,h,d\}$. By the relations (4.3) and (5.1), it is easy to see that

$$\begin{aligned} \mathrm{Var}(\hat{D}^v_{J_2-1,J_1-1,(\cdot,i)}) &= \tilde{H}^*_{J_2-1} \mathrm{Var}(B_{J_2,J_1-1,(\cdot,i)})\tilde{H}_{J_2-1}\\ \mathrm{Var}(\hat{D}^h_{J_2-1,J_1-1,(\cdot,i)}) &= \tilde{G}^*_{J_2-1} \mathrm{Var}(A_{J_2,J_1-1,(\cdot,i)})\tilde{G}_{J_2-1}\\ \mathrm{Var}(\hat{D}^d_{J_2-1,J_1-1,(\cdot,i)}) &= \tilde{G}^*_{J_2-1} \mathrm{Var}(B_{J_2,J_1-1,(\cdot,i)})\tilde{G}_{J_2-1}, \end{aligned} \tag{5.8}$$

where $\mathrm{Var}(A_{J_2,J_1-1,(\cdot,i)})$ and $\mathrm{Var}(B_{J_2,J_1-1,(\cdot,i)})$ are given in (5.2) and (5.4), respectively. Note that, at level $J-1 \equiv (J_2-1, J_1-1)$, the variance of the detail coefficients are computed exactly; no approximation is made. We denote by $\Sigma^\alpha_{J_2-1,J_1-1}$, $\alpha \in \{h,v,d\}$ the matrices containing the variances given by (5.8).

5.3 Coarse scales variance structure

The diagonal elements of the variance of the $i-$th column of A_{J_2-1,J_1-2} and B_{J_2-1,J_1-2} are also computed exactly. We have:

$$\begin{aligned} \mathrm{Var}(A_{J_2-1,J_1-2,(\cdot,i)})_{k,k} &= \tilde{H}^*_{J_1-2,(i,\cdot)} \mathrm{Var}(\hat{S}_{J_2-1,J_1-1,(k,\cdot)})\tilde{H}_{J_1-2,(\cdot,i)}\\ \mathrm{Var}(B_{J_2-1,J_1-2,(\cdot,i)})_{k,k} &= \tilde{G}^*_{J_1-2,(i,\cdot)} \mathrm{Var}(\hat{S}_{J_2-1,J_1-1,(k,\cdot)})\tilde{G}_{J_1-2,(\cdot,i)}\ , \end{aligned}$$

where the variance of a given row of S_{J_2-1,J_1-1} is given by the expressions (5.6) and (5.7).

For the transform along the $y-$axis, the $k-$th element of $\mathrm{Var}(\hat{S}_{J_2-2,J_1-1,(i,\cdot)})$ is equal to

$$\mathrm{Var}(\hat{S}_{J_2-2,J_1-2,(i,\cdot)})_{(k,k)} = \tilde{H}^*_{J_2-2} \mathrm{Var}(A_{J_2-1,J_1-2,(\cdot,k)})\tilde{H}_{J_2-2} \tag{5.9}$$

In order to compute this variance in practice, we take into account only the diagonal elements of $\mathrm{Var}(A_{J_2-1,J_1-2,(\cdot,k)})$, that is, we set equal to zero all the correlations between the elements of the vector $A_{J_2-1,J_1-2,(\cdot,k)}$. To get an idea of the error made in this approximation, consider the ideal situation of an orthogonal transform. In that case, $\tilde{H}^*_{j_2}\tilde{H}_{j_2}$ and $\tilde{H}^*_{j_1}\tilde{H}_{j_1}$ are the identity matrix. In our biorthogonal transform, the matrices $\tilde{H}_{j_1}$ and $\tilde{H}_{j_2}$ are banded by construction. Hence only a few off-diagonal elements are non-zero in the products $\tilde{H}^*_{j_2}\tilde{H}_{j_2}$ and $\tilde{H}^*_{j_1}\tilde{H}_{j_1}$ which should be used to compute all the correlations.

By the equations in (4.3), the variance of the $i-$th column of $\hat{D}^v_{J_2-2,J_1-2}$ is equal to

$$\mathrm{Var}(\hat{D}^v_{J_2-2,J_1-2,(\cdot,i)}) = \tilde{H}^*_{J_2-2} \mathrm{Var}(B_{J_2-1,J_1-1,(\cdot,i)})\tilde{H}_{J_2-2}\ .$$

Again, to compute this variance matrix we make the approximation that $\mathrm{Var}(B_{J_2-1,J_1-1,(\cdot,i)})$ is a diagonal matrix. We record the diagonal elements of $\mathrm{Var}(\hat{D}^v_{J_2-2,J_1-2,(\cdot,i)})$ in the $i-$th column of the matrix $\Sigma^v_{J_2-2,J_1-2}$. The variance of the $i-$th column of $\hat{D}^h_{J_2-2,J_1-2}$ and of $\hat{D}^h_{J_2-2,J_1-2}$ are computed in a completely similar manner using the corresponding expressions in (4.3).

At the next coarser levels, all the variance matrices are approximated by diagonal matrices. For example, when computing

$$\mathrm{Var}(A_{J_2-2,J_1-3,(\cdot,i)})_{k,k} = \tilde{H}^*_{J_1-3,(i,\cdot)} \mathrm{Var}(\hat{S}_{J_2-2,J_1-2,(k,\cdot)})\tilde{H}_{J_1-3,(\cdot,i)} \ ,$$

we consider that the matrix $\mathrm{Var}(\hat{S}_{J_2-2,J_1-2,(k,\cdot)})$ is diagonal, and so on.

In this process, we also ignore the correlations between coefficients belonging to different levels (j_2, j_1) and (j_2+1, j_1+1), say. This approximation is reasonable since the stabilizing update reduces drastically the inter-level correlation between the coefficients, see [9].

5.4 Nonlinear wavelet estimator

We saw in the two previous sections how to obtain the matrices $\Sigma^\alpha_{j_2,j_1}$, $\alpha \in \{v,h,d\}$ which contain the approximations of the 'diagonal' elements of the four-dimensional array $\mathbf{W}^4$. These approximated values are positive by construction. Now, take elementwise the square root of the matrices $\Sigma^\alpha_{j_2,j_1}$ and call the resulting matrices $\Lambda^\alpha_{j_2,j_1}$, $\alpha \in \{v,h,d\}$.

In order to estimate the noise level σ_ϵ, we proceed as follows. For each type of detail coefficient (that is, for $\alpha \in \{v,h,d\}$), we divide the fine scale detail coefficients $\hat{D}^\alpha_{J_2-1,J_1-1}$ elementwise by the matrix $\Lambda^\alpha_{J_2-1,J_1-1}$. We take the median absolute deviation from the median over all the values of the resulting matrix and obtain:

$$\hat{\sigma}^\alpha_{\mathrm{MAD}} = \mathrm{MAD}\left(\hat{D}^\alpha_{J_2-1,J_1-1} ./ \Lambda^\alpha_{J_2-1,J_1-1}\right), \ \alpha \in \{v,h,d\} \ .$$

A level and location dependent threshold is given by the matrices

$$T^\alpha_{j_2,j_1} := \hat{\sigma}^\alpha_{\mathrm{MAD}}\Lambda^\alpha_{j_2,j_1}\sqrt{2\log n}, \ \alpha \in \{v,h,d\} \ , \tag{5.10}$$

where $\hat{\sigma}^\alpha_{\mathrm{MAD}}\Lambda^\alpha_{j_2,j_1}$ is the 2D-version of the standard deviation $\hat{\sigma}_{jk}$ given in (2.12).

Each detail coefficient is individually thresholded and the resulting nonlinear wavelet estimator is of the form

$$\begin{aligned}
\hat{m}(x,y) &= \sum_{k_1,k_2} \hat{S}_{J_2-M,J_1-M,(k_1,k_2)}\varphi_{j_0,(k_1,k_2)}(x,y) \\
&+ \sum_{a=1}^{M}\sum_{k_1,k_2} \tilde{D}^v_{J_2-a,J_1-a,(k_1,k_2)}\psi^v_{J_2-a,J_1-a,(k_1,k_2)}(x,y) \\
&+ \sum_{a=1}^{M}\sum_{k_1,k_2} \tilde{D}^h_{J_2-a,J_1-a,(k_1,k_2)}\psi^h_{J_2-a,J_1-a,(k_1,k_2)}(x,y) \\
&+ \sum_{a=1}^{M}\sum_{k_1,k_2} \tilde{D}^d_{J_2-a,J_1-a,(k_1,k_2)}\psi^d_{J_2-a,J_1-a,(k_1,k_2)}(x,y) \ ,
\end{aligned}$$

where $\tilde{D}^\alpha_{J_2-a,J_1-a,(k_1,k_2)}$ is the $(k_1,k_2)-$th thresholded element of the matrix $\hat{D}^\alpha_{J_2-a,J_1-a}$. The thresholding has been performed using either 'soft' or 'hard' threshold nonlinearities as given in (2.15).

5.5 Algorithm for heteroscedastic errors

In case the errors are independent, but their homoscedasticity is not guaranteed, a procedure to deal with heteroscedastic models of the form $Z_{s,t} = m(X_s, Y_t) + \sigma(X_s, Y_t)e_{s,t}$, $e_{s,t}$ i.i.d, is the following.

First, obtain a pilot estimator of $m(x, y)$ using a linear wavelet estimator, where all the detail coefficients above a cut-off scale (j_2, j_1) are set equal to zero. From this pilot estimator, denoted by $\hat{m}_0(x, y)$, take the residuals $r_{s,t} := Z_{s,t} - \hat{m}_0(X_s, Y_t)$ for $s = 1, \ldots, n_1$; $t = 1, \ldots, n_2$.

Second, estimate the variance function $\sigma^2(x, y)$ from the data set $\{(X_s, Y_t, r^2_{s,t})\}$, again using a linear wavelet estimator. Denote by $\{\hat{\sigma}^2(X_s, Y_t)\}$ the resulting estimator.

Finally, use the estimated values of $\sigma^2(x, y)$ at the design points to compute the variance of each detail coefficient. Let $\hat{\sigma}^2_{(k,\cdot)}$ be the $k-$th row of the matrix composed of the elements $\{\hat{\sigma}^2(X_s, Y_t)\}$. The covariance matrix of the $k-$th row of $\hat{S}_{J_2,J_1}$ is then equal (up to a normalisation factor) to the $(n_1 \times n_1)-$diagonal matrix with diagonal elements given by $\hat{\sigma}^2_{(k,\cdot)}$. The relations in (5.2)-(5.5) are replaced in this case by

$$\begin{aligned}
\mathrm{Cov}(A_{J_2,J_1-1,(\cdot,i)}, A_{J_2,J_1-1,(\cdot,l)})_{k,k} &= \tilde{H}^*_{J_1-1(i,\cdot)} \,\mathrm{diag}(\hat{\sigma}_{(k,\cdot)})\tilde{H}_{J_1-1(\cdot,l)}\mathbf{1}_{n_2,n_2} \\
\mathrm{Cov}(B_{J_2,J_1-1,(\cdot,i)}, B_{J_2,J_1-1,(\cdot,l)})_{k,k} &= \tilde{G}^*_{J_1-1(i,\cdot)} \,\mathrm{diag}(\hat{\sigma}^2_{(k,\cdot)})\tilde{G}_{J_1-1(\cdot,l)}\mathbf{1}_{n_2,n_2} \ ,
\end{aligned}$$

where $\mathrm{diag}(\hat{\sigma}^2_{(k,\cdot)})$ denotes the diagonal matrix with diagonal elements $\hat{\sigma}^2_{(k,\cdot)}$. The relations (5.6)-(5.7), as well as all the subsequent relations, can then be used as before.

6 Simulation study

6.1 Parameters of the simulation study

In this simulation study, the regressors $\{X_s\}_{s=1}^{n_1}$ and $\{Y_t\}_{t=1}^{n_2}$ are independent and uniformly distributed, $X_s \sim U[0, 1]$, $Y_t \sim U[0, 1]$. We consider three test functions.

1. A smooth quadratic function ('Quad' function).

$$m(x, y) = 6 - 12x + 12x^2 + 3y - 5y^2 \tag{6.1}$$

2. A piecewise quadratic function, where the size of the jump (or jump magnitude) is a function of x ('Jump' function).

$$m(x, y) = -0.5 - y + 3(x - 0.5)^2 + (1 + 0.2\sin(2\pi x))1_{\{y>-(0.5-x)^2+0.5\}} \tag{6.2}$$

3. A piecewise quadratic function where the jump curve is a circle ('Hat' function).

$$m(x, y) = -2(x - 0.5)^2 - 2(y - 0.5)^2 + 1_{\{(x-0.5)^2+(y-0.5)^2\leq(0.25)^2\}} \tag{6.3}$$

The three test functions are displayed on Figure 4. The functions (6.2) and (6.3) have been studied in [14] in the context of automatic detection of the jump location in nonparametric regression. We consider data sets of total sample size $n = 3000$ with $n_1 = 50$, $n_2 = 60$. The maximal depth in the decomposition is taken equal to $M = 3$ (this corresponds to a primary level $(j_{0_1}, j_{0_2}) = (2, 2)$).

In our chosen error model, $\epsilon_{s,t} \sim N(0, \sigma_\epsilon^2)$. In order to specify the variance σ_ϵ^2, we define a signal-to-noise ratio as SNR $:= \mathrm{sd}(m)/\sigma$, where

$$\mathrm{sd}(m)^2 = \mathrm{Var}(m) = \frac{1}{n-1}\sum_{s=1}^{n_1}\sum_{t=1}^{n_2}(m(x_s, y_t) - \overline{m})^2, \tag{6.4}$$

for a realisation $\{x_s\}_{s=1}^{n_1}$ of the random variable $X \sim U[0,1]$, and a realisation $\{y_t\}_{t=1}^{n_2}$ of $Y \sim U[0,1]$. The quantity $\overline{m}$ is equal to the mean value of the $m(x_s, y_t)$, $s = 1, \ldots, n_1$, $t = 1, \ldots, n_2$. We set the SNR equal to two, which allows us to specify σ_ϵ. In Figure 4, we show for the three test functions a typical realisation of a data set generated with a SNR = 2.

Let z be a realisation of the observations, $z = \{z_{s,t}\}$. To evaluate the performance of a method on one given data set, we use an efficiency measure defined as

$$\mathrm{Eff}(\hat{m}) = \log_{10}\left(\frac{\mathrm{Var}(m)}{\mathrm{Var}(\hat{m} - m)}\right) / \log_{10}\left(\frac{\mathrm{Var}(m)}{\mathrm{Var}(z - m)}\right) , \tag{6.5}$$

where $\mathrm{Var}(m)$ is defined in (6.4). Naturally, a good estimator will have a large efficiency Eff > 1. On the contrary, if Eff ≤ 1, this means that the estimator is not better than the original observation from a mean square error point of view. In the simulation, we computed equation (6.5) for a large number B of data sets and we summarized the information given by the B runs by recording the median, first and third quartiles of the B efficiencies Eff. For all the situations, $B = 500$ runs were performed.

Our proposed wavelet estimator is based on a weighted average-interpolation scheme followed by a stabilizing update step (hereafter denoted by WAI). The computational complexity of the described two-dimensional wavelet transform is $O(n)$, [9]. We use a hard threshold rule with threshold values given in (5.10).

We compare this method with a locally weighted regression procedure [1, 2], which is implemented in the Splus package under the LOESS function. In this procedure, the predicted value at $\mathbf{x_0}$ is obtained as follows. Find the k nearest neighbors of $\mathbf{x_0} = (x_0, y_0)$. Each of these neighbors is then weighted according to its distance from $\mathbf{x_0}$: points close to $\mathbf{x_0}$ have large weight and points far from $\mathbf{x_0}$ have small weight. In practice, this distance is introduced in a weight function, chosen to be the tricube function $W(u) : (1-u^3)^3$ for $0 \leq u < 1$ and 0 otherwise. A local quadratic function is fitted using weighted least squares. The value of this fitted function at (x_0, y_0) gives $\hat{m}(x_0, y_0)$, and the operation is repeated for all the points in the grid.

The ratio of the number of neighbors k to the total number of points is called the *span* and must be specified by the user. This value of the *span* greatly influences the performance of the estimator: large value will result in a smooth estimator, whereas a small value will produce a rough estimator. Since two of the test functions are discontinuous, we choose the value of the *span* parameter equal to 0.1 and 0.25.

Note that, in contrast, once a threshold rule is chosen (like the modified VisuShrink rule), the only effective tuning parameter that needs to be specified in our wavelet procedure is the maximal depth (or, equivalently, the primary resolution level). In the literature, the primary level is usually set equal to two. Moreover, if the threshold is location and scale dependent, this is only due to the biorthogonality of the transform (and, possibly, to the heteroscedasticity of the data). Unlike the choice of a local bandwidth in kernel or local polynomial smoother, the determination of a data–driven threshold t_{jk} does not involve the estimation of the smoothness of the underlying function.

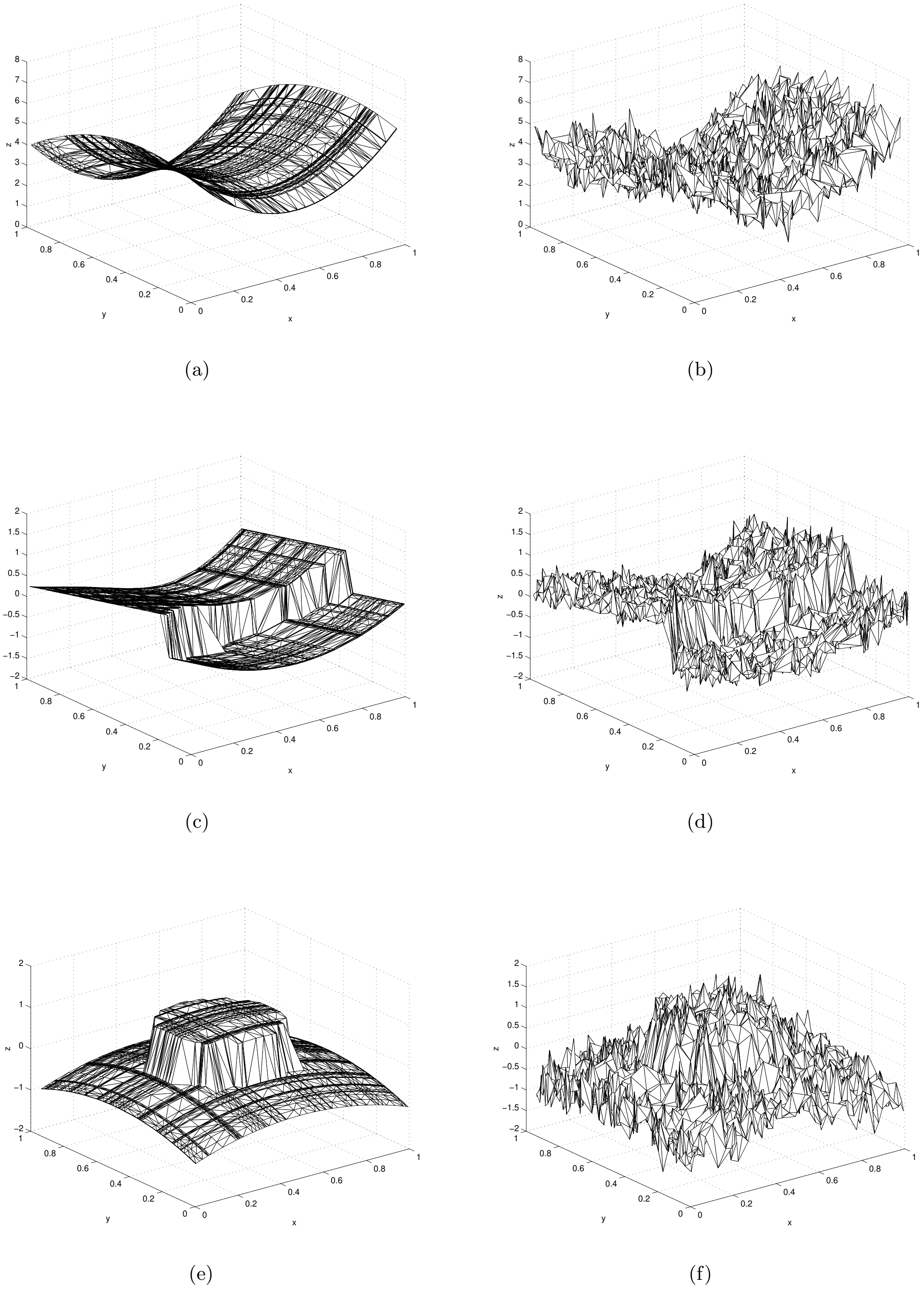


Figure 4: Test functions together with a typical realisation of a noisy data set with SNR = 2: (a) Quadratic smooth function (Quad) (b) Noisy ‘Quad’ function (c) Quadratic function with a jump (Jump) (d) Noisy ‘Jump’ function (e) Quadratic function with a jump in the form of a circle (Hat) (f) Noisy ‘Hat’ function

6.2 Simulation results

Table 1 shows the result of the simulation study for the wavelet denoising method (WAI) and the LOESS algorithm, where the *span* parameter has been chosen equal to 0.1 and 0.25. We now comment and illustrate these results. The estimated functions represented in the graphs have an efficiency close to the corresponding median efficiency reported in Table 1.

Table 1: Results of the simulation study for the WAI estimator and the LOESS estimator with *span*= 0.1 and 0.25 on the three test functions. For each function, the first line gives the median efficiency Eff over the $B = 500$ runs. The numbers between brackets represent the inter-quartile intervals of the efficiency Eff. The sample size was $n_1 = 50, n_2 = 60$ and the SNR used to generate the data sets is taken equal to 2.

	WAI	LOESS	
		span =0.1	*span* =0.25
Quad	3.855 [3.741; 3.957]	3.945 [3.853; 4.028]	4.538 [4.405; 4.682]
Jump	2.195 [2.132; 2.266]	2.020 [1.951; 2.095]	1.703 [1.637; 1.796]
Hat	2.191 [2.126; 2.260]	2.056 [1.980; 2.150]	1.753 [1.686; 1.843]

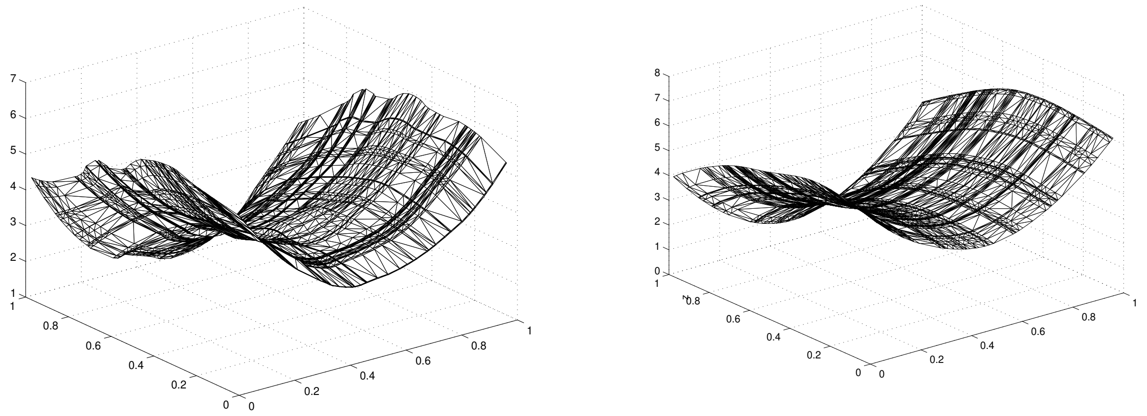
Looking at the values in Table 1, we see that for the smooth Quad function (6.1), the difference in performance between the wavelet estimator and the LOESS estimator with *span* = 0.1 is not significant, since the interquartile intervals of efficiency overlap each other. With a larger value of the smoothing parameter (i.e. *span* = 0.25), this Quad function is perfectly well estimated by the LOESS estimator, see Figure 5(b). When comparing Figure 5(a) and 5(b), we see that the wavelet estimator produces reasonable results for this smooth Quad function.

For the Jump function (6.2), the improvement achieved when using wavelets rather than the LOESS procedure is clearly significant. Indeed, the LOESS estimator oversmoothes the discontinuity, and this leads to a poor efficiency, see Figure 6(b). On the contrary, if one takes a value of *span* inferior to 0.1, the estimator will become rough, or undersmoothed.

In case of the Hat function (6.3), there is a small overlap between the interquartile intervals of the WAI estimator and of the LOESS estimator with *span*=0.1. Figure 8 represents the two estimators of this function. The LOESS algorithm provides a smoother estimate than the wavelet estimator, but the latter one shows more clearly the presence of a discontinuity.

To summarize, these results show that the proposed wavelet method performs reasonably well for smooth functions and provides the best estimation for functions with discontinuities. This is no surprise, since wavelet denoising methods are specially designed to estimate relatively well functions of inhomogeneous smoothness thanks to the nonlinear thresholding procedure. In contrast, the locally weighted regression smoothing has a *global* smoothing parameter, which does not allow to estimate precisely sudden changes in the smoothness of the function.

Figure 6 illustrates this comparison: it shows the estimates of the Jump function when using the wavelet estimator on one hand and the LOESS estimator with *span* equal to 0.1



(a) (b)

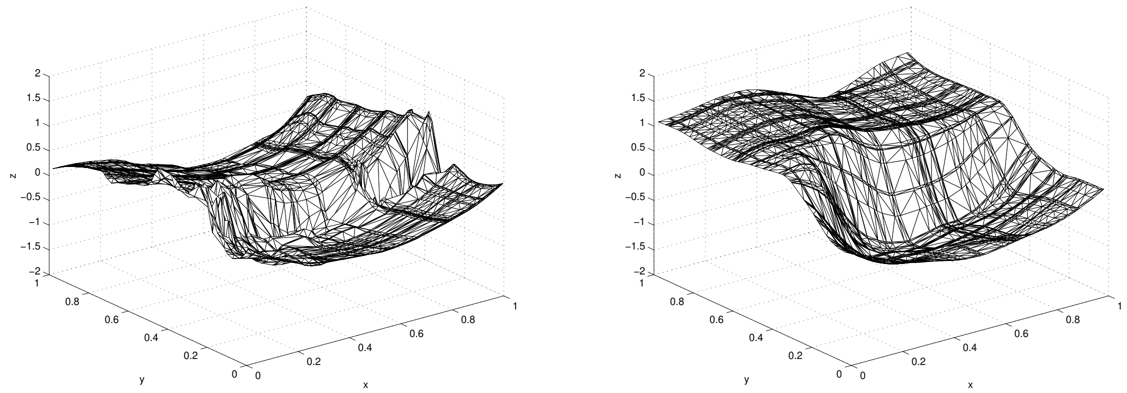
Figure 5: Estimation of the quadratic function defined in (6.1) with (a) the design-adapted wavelet estimator, (b) the locally weighted regression estimator, where the global smoothing parameter is equal to : *span =0.25*. The efficiency Eff is equal to (a) 3.855 and (b) 4.538.

on the other hand. The wavelet procedure gives a good estimation of the function near the discontinuity as well as away from the discontinuity. The LOESS estimator produces an oversmoothed estimator, which is quite far from the actual underlying function. This becomes even clearer if we look at the estimator of the univariate function $m_{x_0}(y) := m(x_0, y) = E[Z|X = x_0, Y = y]$ for different values of x_0, see Figure 7. Figure 7(a) represents the WAI estimation of $m_{x_0}(y)$ for $x_0 = 0.07, x_0 = 0.52$ and $x_0 = 0.92$. It is striking to see how the wavelet estimator manage to localize the jump. Also the jump magnitude (i.e, the size of the jump) is fairly well estimated. However, some wiggles do appear around the change point, but this is inherent to the nature of a wavelet estimator. Observe also how close are the wavelet estimate and the underlying function away from the discontinuity. The result obtained with the LOESS estimator in Figure 7(b) is totally different: here the jump is not so well localized and the jump magnitude is largely over-estimated.

7 Conclusion

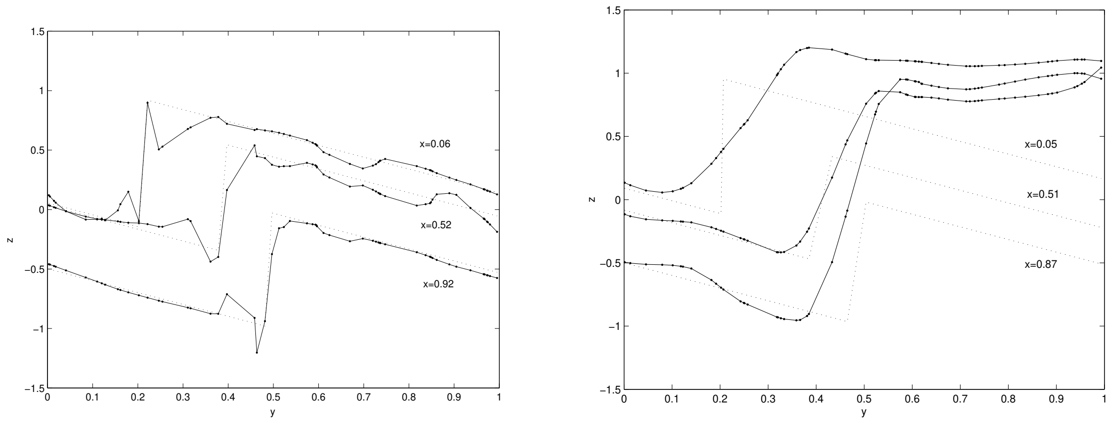
In this paper, we proposed a two-dimensional wavelet estimator for functions that live on so-called half-regular grid. This estimator is based on a tensor product of one-dimensional design-adapted wavelets introduced in [9]. The corresponding algorithm is fast and the wavelet estimator automatically adapts to the sample size and to the presence of a discontinuity (jump curve).

We used a simple thresholding rule which is the direct generalization of the Universal Threshold [11]. Special attention however has to be paid to the estimation of the variance of the detail coefficients. We compared this estimator with a locally weighted regression procedure, called LOESS and implemented in the Splus package. This procedure involves the choice of a global smoothing parameter, which is crucial for the good performance of the estimator. In our simulation study, we choose it by hand, although some data-driven choice,



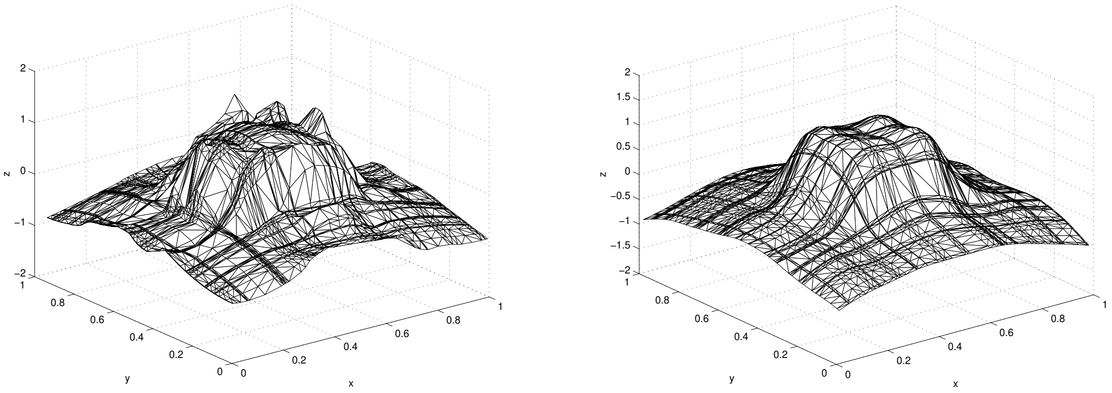
(a) (b)

Figure 6: Estimation of the quadratic function with a jump curve defined in (6.2) with (a) the design-adapted wavelet estimator, (b) the locally weighted regression estimator, where the global smoothing parameter is equal to : *span =0.1.* The efficiency Eff is equal to (a) 2.196 and (b) 2.009.



(a) (b)

Figure 7: In (a), three 1D-views of the wavelet estimation of Figure 6(a) are shown: the estimation of $m(x_0, \cdot)$ at $x_0 = 0.06, x_0 = 0.51$ and $x_0 = 0.92$. Similarly, in (b), three 1D-views of the LOESS estimation of Figure 6(b), for$x_0 = 0.05, x_0 = 0.52$ and $x_0 = 0.87$, are shown. In both cases, the dotted lines represent the underlying signal and the dots represent the fitted values at the given design-points.



(a) (b)

Figure 8: Estimation of the 'Hat' function defined in (6.3) with (a) the design-adapted wavelet estimator, (b) the locally weighted regression estimator, where the global smoothing parameter is equal to: *span =0.1*. The efficiency Eff is equal to (a) 2.192 and (b) 2.056.

e.g. by cross-validation, is possible [12].

With a global smoothing parameter, the LOESS procedure tends to oversmooth the discontinuities. Indeed, it is a well-known fact that for such kernel or local-polynomial type estimator, a local choice of the smoothing parameter must be implemented in order to be able to adapt to the (possible varying) smoothness of the underlying function [12].

In contrast, with our design-adapted wavelet estimator, the thresholding rule allows to easily handle the presence of jumps, while providing a smooth estimation in the regions without discontinuities. Moreover, no complex data-driven choice of the smoothing parameter is necessary.

References

[1] W.S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *J. Amer. Statist. Assoc.*, 74:829–836, 1979.

[2] W.S. Cleveland and S.J. Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *J. Amer. Statist. Assoc.*, 83:596–610, 1988.

[3] W. Dahmen. Some remarks on multiscale transformations, stability and orthogonality. In P. J. Laurent, A. Le Méhauté, and L. L. Schumaker, editors, *Curves and Surfaces II*, pages 1–32. A. K. Peters, Boston, 1991.

[4] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, Philadelphia, 1992.

[5] I. Daubechies, I. Guskov, and W. Sweldens. Regularity of irregular subdivision. *Constructive Approximation*, 15(3):381–426, 1999.

[6] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.

[7] V. Delouille, J. Franke, and R. von Sachs. Nonparametric stochastic regression with design-adapted wavelets. *Sankhya: The Indian Journal of Statistics. Special Issue on Wavelets, Series A*, 63(3):328–366, 2001.

[8] V. Delouille, M. Jansen, and R. von Sachs. Second generation wavelet denoising for irregularly spaced data in two dimensions. In preparation, 2002.

[9] V. Delouille, J. Simoens, and R. von Sachs. Smooth design-adapted wavelets for non-parametric stochastic regression. Discussion Paper 0117, Institut de Statistique, UCL, Belgium, 2001.

[10] D.L. Donoho. Smooth wavelet decompositions with blocky coefficient kernels. In L. Schumaker and G. Webb, editors, *Recent Advances in Wavelet Analysis*, pages 259–308. Academic Press, 1993.

[11] D.L. Donoho. De-noising via soft-thresholding. *IEEE Transactions on Information Theory*, 41:613–627, 1995.

[12] J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*. Chapman and Hall, 1996.

[13] M. Girardi and W. Sweldens. A new class of unbalanced Haar wavelets that form an unconditional basis for L_p on general measure spaces. *J. Fourier Anal. Appl.*, 3(4):457–474, 1997.

[14] A.C. Goderniaux. *Automatic detection of change-points in nonparametric regression*. PhD thesis, Institut de Statistique, UCL, Belgium, 2001.

[15] M. Jansen, G. Nason, and B. Silverman. Scattered data smoothing by empirical Bayesian shrinkage of second generation wavelet coefficients. In M. Unser and A. Aldroubi, editors, *Wavelet Applications in Signal and Image Processing*, volume 4478, pages 87–97. IX Proceedings of SPIE, 2001.

[16] A. Kovac. *Wavelet Thresholding for Unequally Spaced Data*. PhD thesis, University of Bristol, March 1998.

[17] D. Levin. Using Laurent polynomial representation for the analysis of non-uniform binary subdivision schemes. *Advances in Comp. Math.*, 11:41–54, 1999.

[18] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, second edition, 1998.

[19] M. Neumann and R. von Sachs. Wavelet thresholding in anisotropic function classes and application to adaptive estimation of evolutionary spectra. *Annals of Statistics*, 25(1):38–76, 1997.

[20] J. Simoens and S. Vandewalle. A stabilized lifting construction of wavelets on irregular meshes on the interval. *SIAM J. Sci. Comput.*, 2002. To appear.

[21] W. Sweldens. *Construction and Applications of Wavelets in Numerical Analysis*. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, 1994.

[22] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.

[23] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.

[24] W. Sweldens and P. Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, pages 15–87. ACM SIGGRAPH Course notes, 1996.