



## Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers ParisTech researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <http://sam.ensam.eu>  
Handle ID: <http://hdl.handle.net/10985/8626>

### To cite this version :

Raphaël MOULART, René ROTINAT - On the use of a penalized least squares method to process kinematic full-field measurements - Measurement Science and Technology - Vol. 25, p.1-15 - 2014

Any correspondence concerning this service should be sent to the repository

Administrator : [archiveouverte@ensam.eu](mailto:archiveouverte@ensam.eu)

# On the use of a penalized least squares method to process kinematic full-field measurements

Raphaël Moulart<sup>1</sup> and René Rotinat<sup>1</sup>

<sup>1</sup> Mechanics, Surfaces and Materials Processing (MSMP), Arts et Métiers ParisTech, Rue Saint-Dominique B.P. 508, 51006 Châlons-en-Champagne, FRANCE

**Abstract.** This work is aimed at exploring the performances of an alternative procedure to smooth and differentiate full-field displacement measurements. After recalling the strategies currently used by the experimental mechanics community, a short overview of the available smoothing algorithms is drawn up and the requirements that such an algorithm has to fulfil to be applicable to process kinematic measurements are listed. A comparative study of the chosen algorithm is performed including the 2D penalized least squares method and two other commonly implemented strategies. The results obtained by penalized least squares are comparable in terms of quality to those produced by the two other algorithms, while the penalized least squares method appears to be the fastest and the most flexible. Unlike both other considered methods, it is possible with penalized least squares to automatically choose the parameter governing the amount of smoothing to apply. Unfortunately, it appears that this automation is not suitable for the proposed application since it does not lead to optimal strain maps. Finally, it is possible with this technique to perform the derivation to obtain strain maps before smoothing them (while the smoothing is normally applied to displacement maps before the differentiation) which can lead in some cases to a more effective reconstruction of the strain fields.

*Keywords:* Full-field displacements maps; smoothing algorithms; penalized least squares; 2D differentiation; generalized cross-validation.

## 1. Introduction

Most of the full-field kinematic measurement methods (digital image correlation, electronic speckle pattern interferometry, grid method, . . .) provide displacement maps. However, in solid mechanics, the useful information is normally given by strain fields (for identification of mechanical properties for instance). Consequently, the raw displacement data usually have to be numerically differentiated. In dynamics, acceleration and strain rate maps can also be significant but they need derivations of the displacements maps as well (respectively temporal derivation and spatio-temporal derivation). However, it is well known that a numerical differentiation strongly amplifies the level of noise that affects a signal (*i.e.* the experimental measurements). An intermediary step of smoothing is thus often necessary before proceeding to the numerical derivation.

Only a few smoothing techniques are currently used by the experimental mechanics community [1]:

- Local data filtering by a kernel: this 2D-filtering is directly inspired by image processing algorithms. A convolution with a 2D-weighting kernel (for instance, a gaussian kernel) is performed onto the data [2]. This technique provides a fast approach but does not efficiently reconstruct the data near the boundaries of their domain.
- Global regression techniques: the experimental data are assumed to follow a specific global form (a 2D polynomial function for instance) and are reconstructed by regression thanks to a least squares algorithm [3]. This approach can be efficient for very low spatial frequency signals that do not present strong localization phenomena but can induce reconstruction artefacts on the borders and parasitic oscillations within the reconstructed field when the number of parameters of the function increases (for instance, the degree of the polynomial) [4].
- Global regression based on a “finite elements-like” approach [5, 6]. This technique can be quite efficient in terms of computation time but the quality of the reconstruction is strongly related to the meshing of the region of interest.
- Local polynomial regression utilizing a moving weighted least squares algorithm is the approach underlying the so-called “diffuse approximation” [5, 6]. This technique provides very satisfactory results (especially efficient for reconstructing the data at the boundary) but can be very time-consuming when the span of the moving least squares algorithm increases.

In order to improve the currently applied methodologies, a fast bibliographical review in statistics and data analysis literature shows that several competitive alternatives do exist. This article is aimed at adapting such a technique to the processing of full-field kinematic measurements.

After having drawn up a bibliographical review of available methods, a scope statement of the smoothing algorithm of displacement maps is established. In the next section, the principle of the chosen algorithm, the penalized least squares, is explained before exploring the performances of this technique compared to two other popular methods. A final section concludes the paper and gives some prospects of the chosen method that could easily be extended to fit new requirements induced by the current evolution of full-field kinematic measurements (especially, the 3D tomographic measurements that are becoming more and more popular).

## **2. State of the art: different methods of smoothing**

As underlined in the previous section, the critical point to perform a valid numerical derivation is to smooth the data in such a way that the effect of the noise does not completely invalidate the derivative values. But, at the same time, the smoothed values must not diverge too much from the original raw data.

For some authors, the notion of “smoothing” itself implies this compromise between regression techniques for which the data are assumed to follow a specific functional form (parametric approach), which is a strong assumption, and the direct use of raw data (purely nonparametric approach) [7, 8]. In smoothing techniques, the only assumption that is made *a priori* is that the true values follow a smooth progression (which excludes, *a priori*, the occurrence of discontinuities such as in cracking phenomena).

Among, the available techniques, four main approaches can be distinguished. They are overviewed in the next paragraphs and mathematically explained for a simple 1D data set (*i.e.* a function  $y = f(x)$  for which noisy observations are recorded, depending on a single parameter  $x$ ).

The simplest method is kernel smoothing. This approach was introduced for density estimation by Rozenbalt [9] and Parzen [10] and has since been widely employed as the method is intuitive and easily set up. The reader may find a general presentation of kernel smoothing and references in [11].

The smoothed value  $\hat{y}_i$  at a specific point  $x_i$  is defined as a weighted average of the raw values in a window defined around the specified point  $x_i$ . Generally, the window is centered on the considered point and thus the reconstruction algorithm can be written as:

$$\hat{y}_i = \frac{\sum_{j=1}^n W\left(\frac{x_j - x_i}{h}\right) y_j}{\sum_{k=1}^n W\left(\frac{x_k - x_i}{h}\right)}, \quad (1)$$

where

- $\hat{y}_i$  is the reconstructed smooth value of the data at the point  $x_i$ ;
- $h$  is a parameter known as the bandwidth, such as the window is defined as  $x \in [x - h; x + h]$ ;
- $n$  is the number of  $x$  values in the window;
- $W(\cdot)$  is the weight function constructed such as to be maximal near  $x_i$  and decrease more or less progressively toward zero when moving away from  $x_i$  (such as gaussian or polynomial functions for instance);
- $y_j$  is the raw recorded value of the data at the point  $x_j$ .

Applying this algorithm for all the available  $x_i$  leads to a complete smooth curve. This approach is widely used, due to its relatively simple principle. It is particularly implemented for image processing as “convolution” to remove high frequency noise from a picture.

However, this kernel approach presents a number of weaknesses: especially the boundary bias. Near the boundaries of the domain of the data, the sum  $\sum_{j=1}^n W\left(\frac{x_j - x_i}{h}\right) y_j$  is truncated leading to border effects with erroneous results (under or

overestimation of the values). To prevent this bias, modified versions of kernel smoothers have been proposed in the literature with a more or less efficient correction effect [11, 12].

The second available approach listed is the locally weighted polynomial regression. This method was mainly introduced by Cleveland as the LOWESS (LOcally WEighted Scatterplot Smoothing) or LOESS (LOcal regrESSion) method [13, 14, 15]. The “diffuse approximation” previously mentioned is based on the same approach and can be seen as a 2D variation on Cleveland’s LOESS. More information about it can be found in [12, 16].

With this technique, the reconstructed smooth value of the data  $\hat{y}_i$  is considered to be the image of  $x_i$  by a low degree polynomial, only valid in the neighbourhood of  $x_i$  (for  $x \in [x - h; x + h]$ ). The  $\hat{y}_i$  value is computed thanks to a locally weighted least squares approach, that is to say by minimizing the following expression 2 (for a quadratic approximation):

$$\sum_{j=1}^n W \left( \frac{x_j - x_i}{h} \right) (y_j - (a_0 + a_1 (x_j - x_i) + a_2 (x_j - x_i)^2))^2, \quad (2)$$

where the terms are the same as defined for kernel smoothing in equation 1.

As the least squares method has an analytical solution, one can write:

$$\begin{Bmatrix} \hat{a}_0 \\ \hat{a}_1 \\ \hat{a}_2 \end{Bmatrix}_i = \left( [X_i]^T [W_i] [X_i] \right)^{-1} [X_i]^T [W_i] \{Y_i\}, \quad (3)$$

with  $[X_i]$ , the following matrix:

$$[X_i] = \begin{bmatrix} 1 & x_1 - x_i & (x_1 - x_i)^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n - x_i & (x_n - x_i)^2 \end{bmatrix},$$

$[W_i]$ , a diagonal matrix containing the different  $W \left( \frac{x_j - x_i}{h} \right)$  values and  $\{Y_i\} = \langle y_1 \cdots y_n \rangle^T$ .

Finally,  $\hat{y}_i$  is equal to the polynomial value at  $x_i$ , *i.e.*,  $(\hat{a}_0)_i$  (and its local first derivative is equal to  $(\hat{a}_1)_i$ ).

Compared to kernel smoothing, this method is less sensitive to border effects which allows to increase the bandwidth parameter and thus the level of smoothing without impairing the results.

At the crossroads of the two previously mentioned approaches, the popular Savitzky-Golay smoothing filter is a kernel based estimation of the local polynomial regression [17]. This method combines the simplicity and velocity of kernel smoothers and the efficiency of a local polynomial regression (all the more by using a modified version dealing with borders [18]). It can also directly compute the derivatives of the parameter [18].

Orthogonal series methods represent the data with respect to an orthogonal set of basis functions. This is the case of Fourier and wavelet transforms. Eventually, only the low frequency terms are retained to reconstruct the smoothed data [19, 20]. These methods have been widely used to model time series.

Finally, the last classical smoothing method is the use of splines combined with a penalized likelihood algorithm. This one, which is the chosen method in this study, is explained in details in next section.

Apart from these “classical” smoothing approaches, one can also mention other methods based especially on dimension reduction such as projection pursuit [21], regression trees [22], generalized additive models [23] and neural networks [24].

After this short review of available smoothing methods, among which a few are already implemented to process kinematic fields as stated in the introduction, it can be seen that various solution exists. To be suitable for the processing of kinematic full-field measurements, the chosen method must fulfill a few obvious requirements:

- (i) be able to deal with 2D data (measured data depending on two spatial parameters  $(x, y)$ ) or even 3D data (results from tomographic measurements);
- (ii) be able to deal with irregularly disposed set of values and missing values: due to the shapes of samples that can present some curvature, holes, etc.;

In addition, it is preferable that the method fulfills the two following conditions:

- (i) to allow a local reconstruction: in the most general cases, kinematic full-field measurements are used to visualize and study heterogeneous strain fields with potentially high gradients. The application of overly broad methods such as a polynomial regression will involve an increase of the order of the polynomial that will induce parasitic effects as previously mentioned [4];
- (ii) to be quick and robust: this point is critical as the smoothing procedure can be time consuming when large data sets have to be processed.

If all the reviewed techniques fulfil more or less most of these requirements, it has been said that kernel smoothing is not really suited to deal with the second one (irregular shapes and holes due to boundary effects) whereas local polynomial regression can be overly time consuming. It is shown in the two next sections that the penalized least squares technique illustrated in this study satisfies *a priori* all the listed conditions.

### 3. The penalized least squares method

The explanations of the principles and developments of the penalized least squares method are taken mostly from [25] and [26]. The reader may find further details in both these references.

#### 3.1. Basic expression

The origin of the penalized least squares method dates from 1899 with Bohlmann’s work [27]. However, this work did not receive much attention at this time (maybe due to the fact that the paper was written in German) and the original idea is often attributed to Whittaker in 1923 [28], who spoke about graduation of the observations

to refer to the smoothing operation. This work was completed by Whittaker [29] and Henderson [30, 31, 32], which is why the method is sometimes called “Whittaker-Henderson smoothing”.

Four decades later, Tikhonov proposed a well-known regularization scheme within the context of ill-posed problems [33]. The underlying equation is very similar to Whittaker’s formulation: *Penalized least squares* method, *Tikhonov regularization* and *ridge regression* are different names for the same approach.

The modern formulation as spline smoothing within the context of nonparametric regression was given by Schoenberg [34] and Reinsch [35] in the 60’s.

The basic principle of the penalized least squares method is to explicitly and mathematically express the need of a compromise between fidelity to the raw data and smoothness of the reconstruction. To do this, an objective function to minimize is constructed as a combination of a quantity parameterizing the deviation from the original data and another quantity parameterizing the roughness of the reconstructed data. A relative weight is given to each of these quantities: generally, the weight of the “Deviation” is set to 1 and relative weight of the “Roughness” is set to a real value  $\lambda$  which is called the smoothing parameter (or sometimes “ridge parameter”).

To illustrate this principle, let us consider a 1D data set consisting in  $n$  observed values  $y_i$  depending on a single parameter  $x$  whose values  $x_i$  are equally spaced. The following equation defines the  $Q$  term that has to be minimized to get smoothed values  $\hat{y}_i$ :

$$Q = \underbrace{\sum_{i=1}^n (y_i - \hat{y}_i)^2}_{\text{Deviation from data}} + \lambda \underbrace{\sum_{i=1}^{n-d} (\Delta^d \hat{y}_i)^2}_{\text{Roughness}}. \quad (4)$$

The term  $\Delta^d \hat{y}_i$  defines the  $d^{\text{th}}$  difference of the  $\hat{y}_i$  values such as:

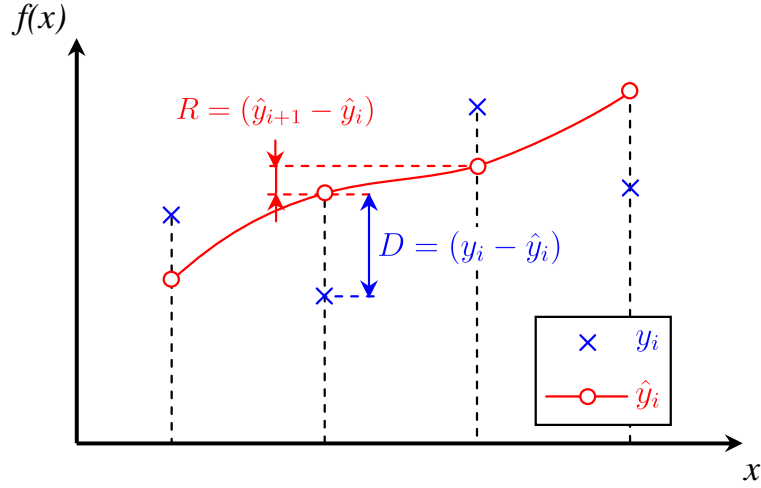
$$\begin{aligned} \Delta^1 \hat{y}_i &= \Delta \hat{y}_i &= \hat{y}_{i+1} - \hat{y}_i \\ \Delta^2 \hat{y}_i &= \Delta (\Delta \hat{y}_i) &= \hat{y}_{i+2} - 2 \times \hat{y}_{i+1} + \hat{y}_i \\ &\dots \end{aligned} \quad (5)$$

Figure 1 illustrates the values  $(y_i - \hat{y}_i)$  and  $(\hat{y}_{i+1} - \hat{y}_i)$  (respectively named  $D$  for “deviation” and  $R$  for “roughness”) for a single point  $(x_i, y_i)$  (and its reconstruction  $(x_i, \hat{y}_i)$ ) for  $d = 1$ .

Rewriting equation 4 with a matrix form leads to:

$$\begin{aligned} Q &= \langle y - \hat{y} \rangle \cdot \{y - \hat{y}\} + \lambda ([D_d] \{\hat{y}\})^T \cdot ([D_d] \{\hat{y}\}) \\ &= \langle y - \hat{y} \rangle \cdot \{y - \hat{y}\} + \lambda \langle \hat{y} \rangle [D_d]^T \cdot [D_d] \{\hat{y}\} \end{aligned}, \quad (6)$$

where  $\langle \cdot \rangle$  denotes a line vector,  $\{ \cdot \}$ , a column vector and  $[ \cdot ]$  a matrix. The  $[D_d]$  denotes the  $(n - d)$  by  $n$   $d^{\text{th}}$  difference matrix; for instance  $D_1$  is the  $(n - 1)$  by  $n$  matrix defined



**Figure 1.** Illustration on a simple 1D case of the “deviation” and “roughness” terms of the penalized least squares method.

as:

$$D_1 = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}. \quad (7)$$

To minimize  $Q$ , its derivatives with respect to the  $\hat{y}_i$  have to be canceled:

$$\begin{cases} \frac{\partial Q}{\partial \hat{y}_i} \end{cases} = 0 = -2 \times \{y - \hat{y}\} + 2 \times \lambda [D_d]^T [D_d] \{\hat{y}\}. \quad (8)$$

It leads to:

$$\begin{cases} \{\hat{y}\} + \lambda [D_d]^T [D_d] \{\hat{y}\} \\ ([I] + \lambda [D_d]^T [D_d]) \{\hat{y}\} \end{cases} = \{y\}. \quad (9)$$

Assuming that the operator  $([I] + \lambda [D_d]^T [D_d])$  is invertible, the final optimal values of  $\{\hat{y}\}$  are:

$$\{\hat{y}\} = \left( [I] + \lambda [D_d]^T [D_d] \right)^{-1} \{y\}. \quad (10)$$

This way to smooth data is very simple and very fast since the problem has a close-form solution (due to the use of quadratic terms for both deviation and roughness terms). It can be easily implemented on high level languages such as Matlab<sup>®</sup> or equivalent free softwares (Octave, Scilab). Computation time can be further improved by using sparse matrices and Cholesky factorization [25].

If the smoothing parameter  $\lambda$  is set to be equal to 0, the reconstructed data  $\hat{y}_i$  will be equal to the original data  $y_i$ . On the contrary, if  $\lambda$  tends to infinity, the reconstructed



$\hat{y}_i$  will be such as  $\hat{y}_i = P(x_i)$ , with  $P(x_i)$ , the polynomial of degree  $(d-1)$  that is obtained with a simple least square regression applied on the whole set of values [34, 25].

### 3.2. Dealing with missing data

To deal with missing data (that can be represented as NaN's in the Matlab<sup>®</sup> environment), a weighted version of the algorithm is set up. The missing values are affected with a null weight  $w_i = 0$  whereas the valid values are weighted by  $w_i = 1$ . The  $Q$  term to minimize thus becomes:

$$Q = \sum_{i=1}^n w_i (y_i - \hat{y}_i)^2 + \lambda \sum_{i=1}^{n-d} (\Delta^d \hat{y}_i)^2, \quad (11)$$

or, in vectorial form:

$$Q = \langle y - \hat{y} \rangle [W] \{y - \hat{y}\} + \lambda \langle \hat{y} \rangle [D_d]^T \cdot [D_d] \{\hat{y}\}, \quad (12)$$

with  $[W]$ , the diagonal matrix containing the different  $w_i$  values. Minimizing this equation, following the same previous steps, leads to a vector of the reconstructed values  $\hat{y}_i$  such as:

$$\{\hat{y}\} = \left( [W] + \lambda [D_d]^T [D_d] \right)^{-1} [W] \{y\} \quad (13)$$

### 3.3. Dealing with 2D data

To deal with 2D data sets, a simple way is to smooth each column first with the 1D-algorithm and then each row. But other more subtle possibilities do exist [36, 26].

For example, the  $[D_d]$  matrix can be decomposed to let a discrete cosine transform appear [37, 26].

For that, let us consider the  $[D_2]$  matrix slightly modified by adding two rows (at the beginning and the end of the matrix, in boldface in the following expression), leading to a square matrix  $[D_{2m}]$ :

$$[D_{2m}] = \begin{bmatrix} \mathbf{-1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 1 & -2 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -2 & 1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{-1} \end{bmatrix}. \quad (14)$$

This modification corresponds to the expression of reflective boundary conditions [37].

As this new matrix is square, it can be eigendecomposed [38]:

$$[D_{2m}] = [U] [\Omega] [U]^{-1} = [U] [\Omega] [U]^T, \quad (15)$$

with  $[\Omega]$ , a diagonal matrix containing the eigenvalues of  $[D_{2m}]$  and  $[U]$ , the concatenation of the eigenvectors of  $[D_{2m}]$  in a square matrix.

Replacing the  $[D_d]$  matrices in equation 10, it leads to:

$$\{\hat{y}\} = [U] ([I] + \lambda [\Omega]^2)^{-1} [U]^T \{y\} = [U] [\Gamma] [U]^T \{y\}. \quad (16)$$

$[U]$  and  $[U]^T$  are respectively the type-2 discrete cosine transform (DCT) and inverse discrete cosine transform (IDCT) matrices [39]. Equation 16 thus becomes:

$$\{\hat{y}\} = [U] [\Gamma] [U]^T \{y\} = \text{IDCT}([\Gamma] \text{DCT}(\{y\})). \quad (17)$$

Due to the properties of the discrete cosine transform, the same kind of approach can be applied to 2D equally spaced data. For a 2D grid of values, equation 17 becomes:

$$[\hat{y}] = \text{IDCT2}([\Gamma^{II}] \circ \text{DCT2}([y])), \quad (18)$$

with DCT2 and IDCT2 respectively denoting the 2D type-2 discrete cosine transform and inverse discrete cosine transform,  $\circ$ , the Hadamard or Schur product (elementwise product) and  $[\Gamma^{II}]$  a square matrix depending on  $\lambda$  and the eigenvalues of  $[D_{2m}]$  in 2D.

#### 3.4. Automatic choice of the smoothing parameter

One strength of the penalized least squares is the fact that the smoothing parameter can be chosen automatically using the method of generalized cross-validation (GCV) [40, 41, 42].

The hypothesis underlying this method is that the measured values of  $\{y\}$  are supposed to be affected by a gaussian noise, such as:

$$\{y\}_{\text{measured}} = \{y\}_{\text{actual}} + \{\varepsilon\}, \quad (19)$$

with  $\{\varepsilon\}$ , a gaussian noise with mean zero and unknown variance. Within the framework of the smoothing procedure,  $\{y\}_{\text{actual}}$  is supposed to be the smoothed value:

$$\{y\} = \{\hat{y}\} + \{\varepsilon\}. \quad (20)$$

Assuming this hypothesis, the following linear system is considered:

$$\{\hat{y}\} = [H(\lambda)] \{y\}. \quad (21)$$

$[H(\lambda)]$  is the so-called ‘‘hat matrix’’ that can be identified in equation 10 as  $([I] + \lambda [D_d]^T [D_d])^{-1}$ .  $\lambda$  is chosen to minimize the GCV score:

$$\lambda = \text{argmin}(\text{GCV}), \quad (22)$$

with:

$$\text{GCV}(\lambda) = \frac{\text{RSS}/n}{(1 - \text{Tr}([H(\lambda)])/n)^2}. \quad (23)$$

RSS denotes the residual sum of squares  $\sum_{i=1}^n$  of the matrix  $[H(\lambda)]$ . It can be seen that the function  $\text{GCV}(\lambda)$  does not have a closed-form expression. Consequently, its minimization needs an iterative optimization procedure than can be potentially time consuming. Using the decomposition defined in equation 15, the problem is simplified as:

$$\text{Tr}([H(\lambda)]) = \sum_{i=1}^n \frac{1}{1 + \lambda \omega_i^2}. \quad (24)$$

This approach can be extended to process 2D data. The reader may find more details about the exact implementation of GCV minimization in [26].

#### 4. Smoothing of kinematic maps: the penalized least squares algorithm

To study the ability of the penalized least squares algorithm (denoted “PLS” hereafter) to deal with kinematic maps, it has been applied successively to:

- a map of a 2D analytical function on which noise has been added ;
- noised maps resulting from a finite-element simulation of a mechanical test;
- experimental maps.

For this purpose, the algorithm described in [26] has been used and reprogrammed in the Matlab<sup>®</sup> environment.

The results produced by PLS have been compared to those produced by two other popular smoothing procedures:

- the diffuse approximation (denoted “DA” hereafter) that has been qualified as a 2D variation of the LOWESS/LOESS approach in section 2 and that is often set up by the experimental mechanics community;
- a modified version of the Savitzky-Golay smoothing filter (denoted “SG” hereafter) as described in [18].

For both these approaches, a local 2D second order polynomial reconstruction has been considered.

As previously mentioned, most of full-field kinematic measurement methods provide displacement maps when the useful information is given by strain fields. Thus, there is a spatial derivation operation (usually, done in a second time) that is superimposed to the smoothing one. Consequently, the criteria that have been considered to qualify the performance of the different algorithms has been applied on strain maps (the spatial derivatives) and not directly on smoothed displacement maps.

##### 4.1. Smoothing of a noised map of a 2D analytical function

The performances of the chosen algorithm have been first evaluated on a noised analytical function. The advantage of using an analytical function is that one has access

to the analytical derivatives which allows a comparison of the reconstructed derivatives to the actual ones getting rid off every source of error on these last values.

The analytical function that has been chosen is the “peaks” function of Matlab<sup>®</sup> that produces a square matrix of  $(n \times n)$  values taken by the following  $z(x, y)$  function:

$$z(x, y) = 3(1-x)^2 \cdot e^{-x^2-(1+y)^2} - 10 \left( \frac{x}{5} - x^3 - y^5 \right) \cdot e^{-x^2-y^2} - \frac{1}{3} e^{-(1+x)^2-y^2}, \quad (25)$$

for  $-3 \leq x \leq +3$  and  $-3 \leq y \leq +3$  and for a pitch of  $\frac{6}{n-1}$ . The analytical expressions of both partial derivatives of  $z(x, y)$  (with respect to  $x$  and to  $y$ ) are the following ones:

$$\begin{aligned} \frac{\partial z}{\partial x} = & -6(1-x)(1+x-x^2) \cdot e^{-x^2-(1+y)^2} \\ & -2(1-17x^2+10x^4+10xy^5) \cdot e^{-x^2-y^2}, \quad (26) \\ & +\frac{2}{3}(1+x) \cdot e^{-(1+x)^2-y^2} \end{aligned}$$

$$\begin{aligned} \frac{\partial z}{\partial y} = & -6(1-x)^2(1+y) \cdot e^{-x^2-(1+y)^2} \\ & +2y(2x-10x^3+25y^3-10y^5) \cdot e^{-x^2-y^2}. \quad (27) \\ & +\frac{2}{3}y \cdot e^{-(1+x)^2-y^2} \end{aligned}$$

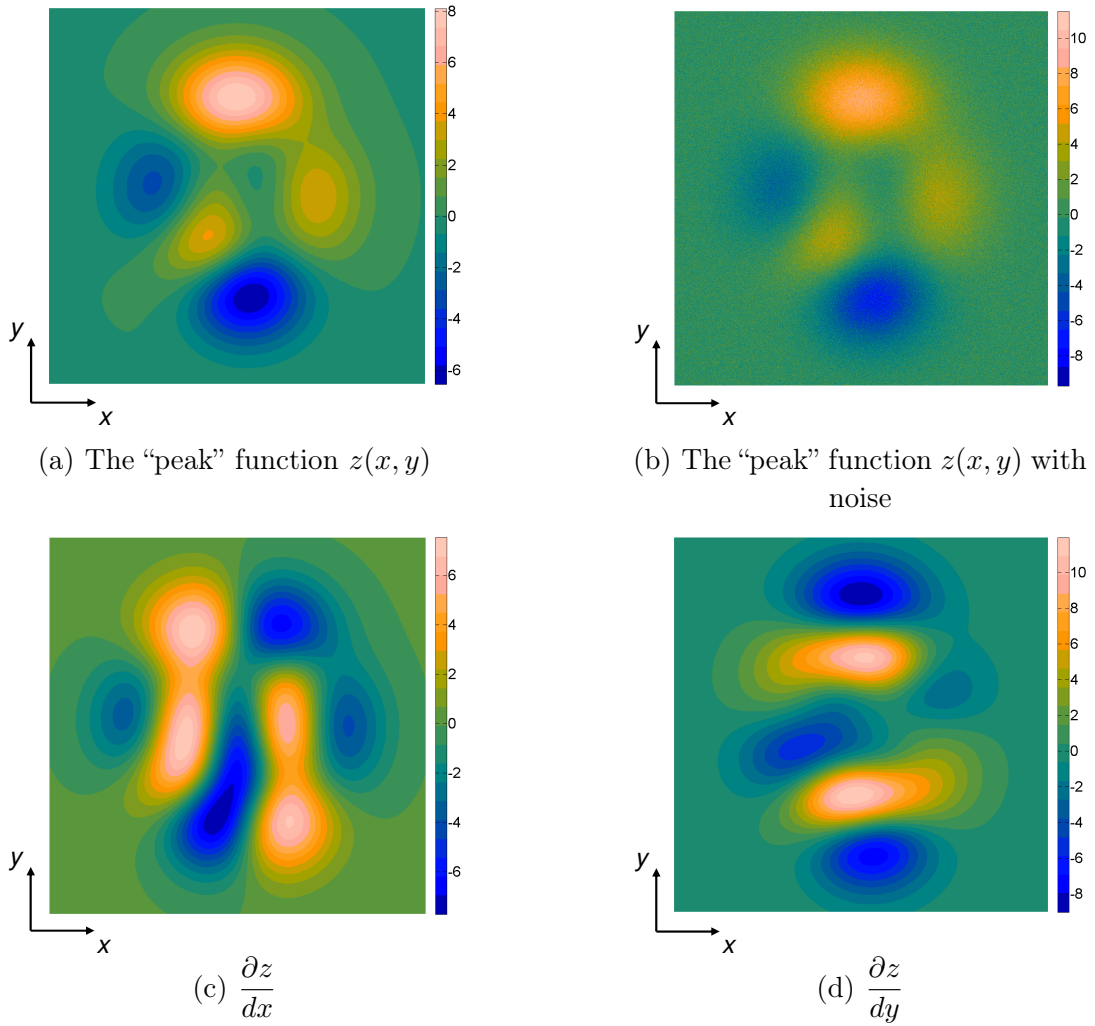
Here, a  $(1000 \times 1000)$  matrix has been generated (pitch equal to  $\frac{2}{333}$ ). A gaussian noise with mean zero and standard deviation one has been added to the map which leads to a typical high resolution noised displacement maps both in terms of spatial localizations and of signal to noise ratio. Views of the raw and noised functions and its analytical derivatives are shown in figure 2.

First of all, the time spent to perform the reconstruction has been looked at. As previously mentioned, the time needed to process large maps with DA can become very long and discouraging.

The computation has been made as if the noised  $z(x, y)$  function map was both  $u_x$  and  $u_y$  maps (leading to three in-plane strain components  $\varepsilon_{xx}$ ,  $\varepsilon_{yy}$  and  $\varepsilon_{xy}$ ). It has been performed using Matlab<sup>®</sup> on a 64 bits computer equipped with an Intel<sup>®</sup> Core<sup>™</sup> i7-2760QM (2.4 GHz) CPU and a 8 GB DDR3 1333 MHz RAM. The computation time against the smoothing parameters for DA, SG and PLS algorithms are shown in figure 3.

It can be seen that:

- for both DA and SG, the computation time is strongly dependent of the span of the window on which the reconstruction is performed (called  $h$  in both cases): the more  $h$ , the longer the processing;
- if the computation time remains reasonable for SG (inferior to half a minute for  $h = 85$  pixels), it becomes very large for DA (about 6 hours for  $h = 85$  pixels);



**Figure 2.** The “peak” function, the noised function and its derivatives.

- the computation time is very low (about 0.78 seconds) and independent of the smoothing parameter  $\lambda$  for PLS (for an automatic choice thanks to the GCV minimization, the computation time becomes 1.2 seconds).

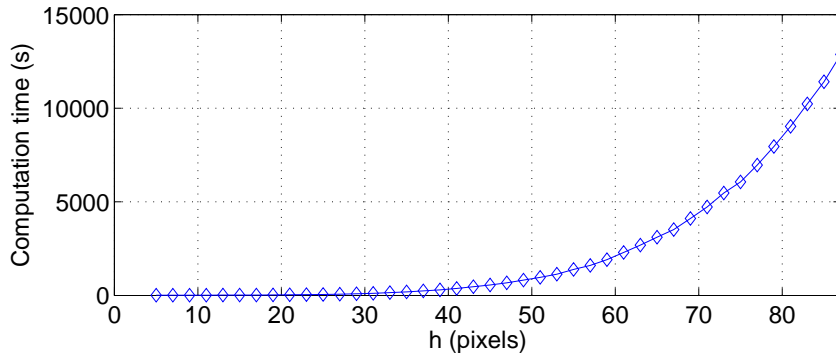
From these results, it is clear that PLS is far quicker than both DA and SG. Moreover, the computation time is independent of the amount of smoothing.

In a second time, the accuracy of the reconstruction has been investigated. For that, an approach similar to the one developed in [5, 6] has been set up. The criterion to qualify the reconstruction is the average distance between the reconstructed derivatives and the exact ones defined as:

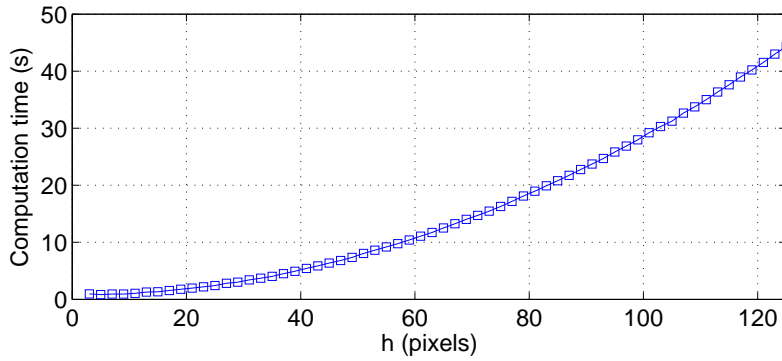
$$d_{\partial} = \left\langle \sqrt{\left( \frac{\partial z^{\text{rec.}}}{\partial x} - \frac{\partial z^{\text{exact}}}{\partial x} \right)^2 + \left( \frac{\partial z^{\text{rec.}}}{\partial y} - \frac{\partial z^{\text{exact}}}{\partial y} \right)^2} \right\rangle, \quad (28)$$

where  $\langle \cdot \rangle$  denotes here a spatial averaging of the data all over the map.

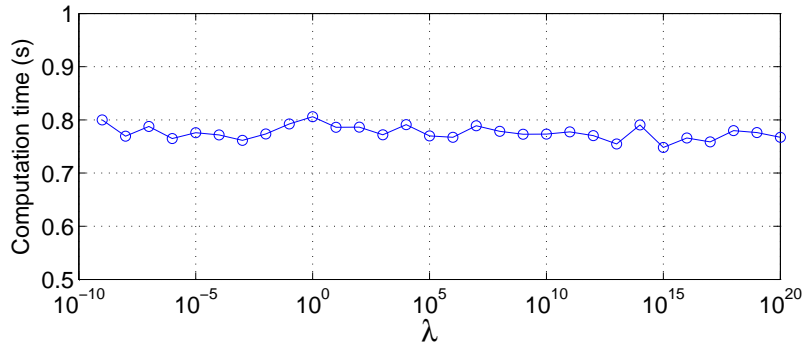
The optimal results for each method (*i.e.* the smoothing parameter leading to the lowest value of  $d_{\partial}$  and this last value) are given in table 1.



(a) Computation time against the span parameter for DA



(b) Computation time against the span parameter for SG



(c) Computation time against the smoothing parameter for PLS

**Figure 3.** Computation time against the smoothing parameters for the different considered algorithms.

It can be seen that, for an optimal value of the smoothing parameter, PLS conducts to a lower value of this average deviation than DA and SG (0.12 for PLS to be compared to 0.22 for DA and 0.23 to SG).

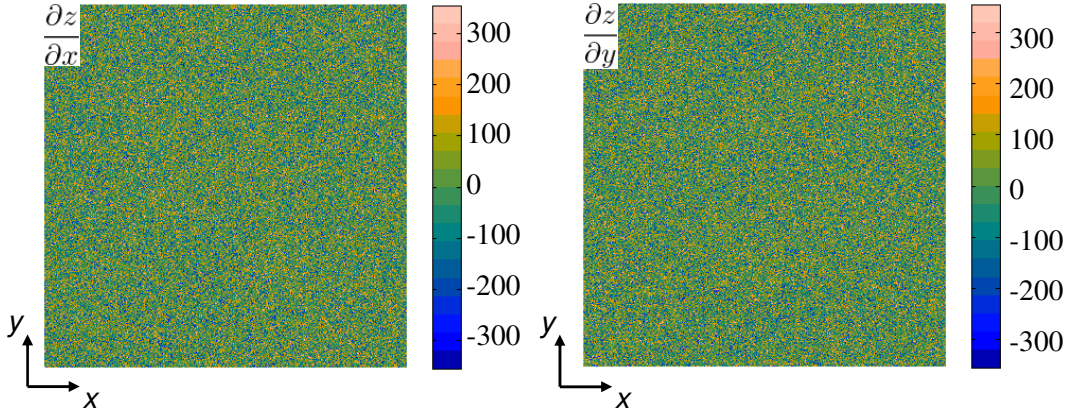
Considering the automatic choice of  $\lambda$  by the GCV minimization, one notes that it does not exactly lead to the value minimizing the average deviation defined in equation 28. However, the value is not far from it ( $\lambda \simeq 5.02 \cdot 10^4$  leading to an average deviation of 0.14 to be compared to  $\lambda \simeq 1.26 \cdot 10^5$  leading to an average deviation of 0.12).

**Table 1.** Performance results for the different algorithms with optimized smoothing parameters for the noised analytical map.

Algorithm	DA	SG	PLS	PLS with GCV
Smoothing parameter	$h = 105$ pixels	$h = 87$ pixels	$\lambda = 1.26 \cdot 10^5$	$\lambda = 5.02 \cdot 10^4$
$d_{\partial}$	0.22	0.23	0.12	0.14

Algorithm	PLS on raw derivatives	PLS on raw derivatives + GCV
Smoothing parameter	$\lambda(\partial Z/\partial x) = 1.58 \cdot 10^5$ ; $\lambda(\partial Z/\partial y) = 1.26 \cdot 10^5$	$\lambda(\partial Z/\partial x) = 4.87 \cdot 10^6$ ; $\lambda(\partial Z/\partial y) = 3.62 \cdot 10^6$
$d_{\partial}$	0.20	0.86

In addition, thanks to the nature of the  $\lambda$  parameter (an adimensional real number) compared to the span  $h$  for DA and SG (an integer denoting a number of pixels), there is a wider range of flexibility. Consequently, one can imagine the validity of applying PLS after the spatial derivation and not before (which is not the case for DA and SG). Moreover,  $\lambda$  does not directly represent a physical quantity (when  $h$  is related to the spatial resolution), so there is no reason not to apply different values of  $\lambda$  to smooth  $\frac{\partial z}{\partial x}$  and  $\frac{\partial z}{\partial y}$ .



**Figure 4.** Derivatives values obtained by simple finite difference applied onto the raw noised  $z(x, y)$  map.

The raw derivatives (obtained by point-to-point difference of the noised maps) are shown in figure 4. To the naked eye, no structure can be observed; the maps seem to be composed only of noise. However, PLS manages to extract reconstructed derivatives from these highly noised maps (average computation time  $\simeq 1.01s$ ) with an accuracy comparable to the one of DA or SG (reported in table 1). On the contrary, the automatic choice of  $\lambda$  by GCV (computation time  $\simeq 1.40s$ ) does not produced optimal results. This can be explained easily by the fact that the minimization of the GCV is based on a minimization of an average weighted quadratic distance between the reconstructed data and the raw noised data [40, 42]. This approach is no longer applicable when the

amount of noise is greater than the signal (*i.e.* the signal to noise ratio becomes inferior to 1).

All these results are confirmed by figure 5 that shows the reconstructed  $\frac{\partial z}{\partial x}$  maps as well as the corresponding error maps  $(\frac{\partial z^{\text{rec.}}}{\partial x} - \frac{\partial z^{\text{exact}}}{\partial x})$  for each applied method. The error maps especially contain interesting information: it appears that error maps corresponding to the application of PLS are less spatially correlated than for both DA and SG. Therefore, one can maintain that PLS presents a better reconstruction of the noised  $z(x, y)$  function than DA and SG. It can also be observed, that the reconstruction is more disturbed on the edges (vertical ones for  $\frac{\partial z}{\partial x}$ ) by applying the differentiation before PLS smoothing. Finally, the minimization of GCV when PLS is applied after differentiation leads to error maps showing large patterns of deviation from the exact values.

#### 4.2. Smoothing of kinematic maps from a finite elements simulation

The algorithm has been tested a second time on simulated data. For that, a finite element model has been built. The mechanical test was a tensile test on a double notched specimen such as the one used in [43]. Figure 6 shows a view of the geometry, the region of interest and the boundary conditions of this model.

The finite element model consisted of a plane-stress simulation using standard linear 3-node triangular elements with an isotropic material. The sample was meshed by 74664 elements and 37587 nodes (65554 elements and 33204 nodes within the region of interest where the mesh size was thinner so that the finite element displacement fields could be reasonably considered as the exact ones). To create the synthetic displacement maps, the fields were evaluated on a regular grid of  $240 \times 160$  pixels by interpolating linearly the nodal displacements (*i.e.* using the shape functions of the 3-node triangular element). In these conditions, the data grid spacing is equal to 0.1 mm (which is typically the thinner grid spacing of results that can be obtained from white-light optical measurements with the “grid method” [44, 45]). Then, a gaussian noise with mean zero and standard deviation  $10^{-8}$  (in m *i.e.* 10 nm) was added to the displacement maps (leading to typical noised displacement maps in terms of signal to noise ratio). Views of these maps and of the strain maps obtained by finite difference derivation are shown in figure 7.

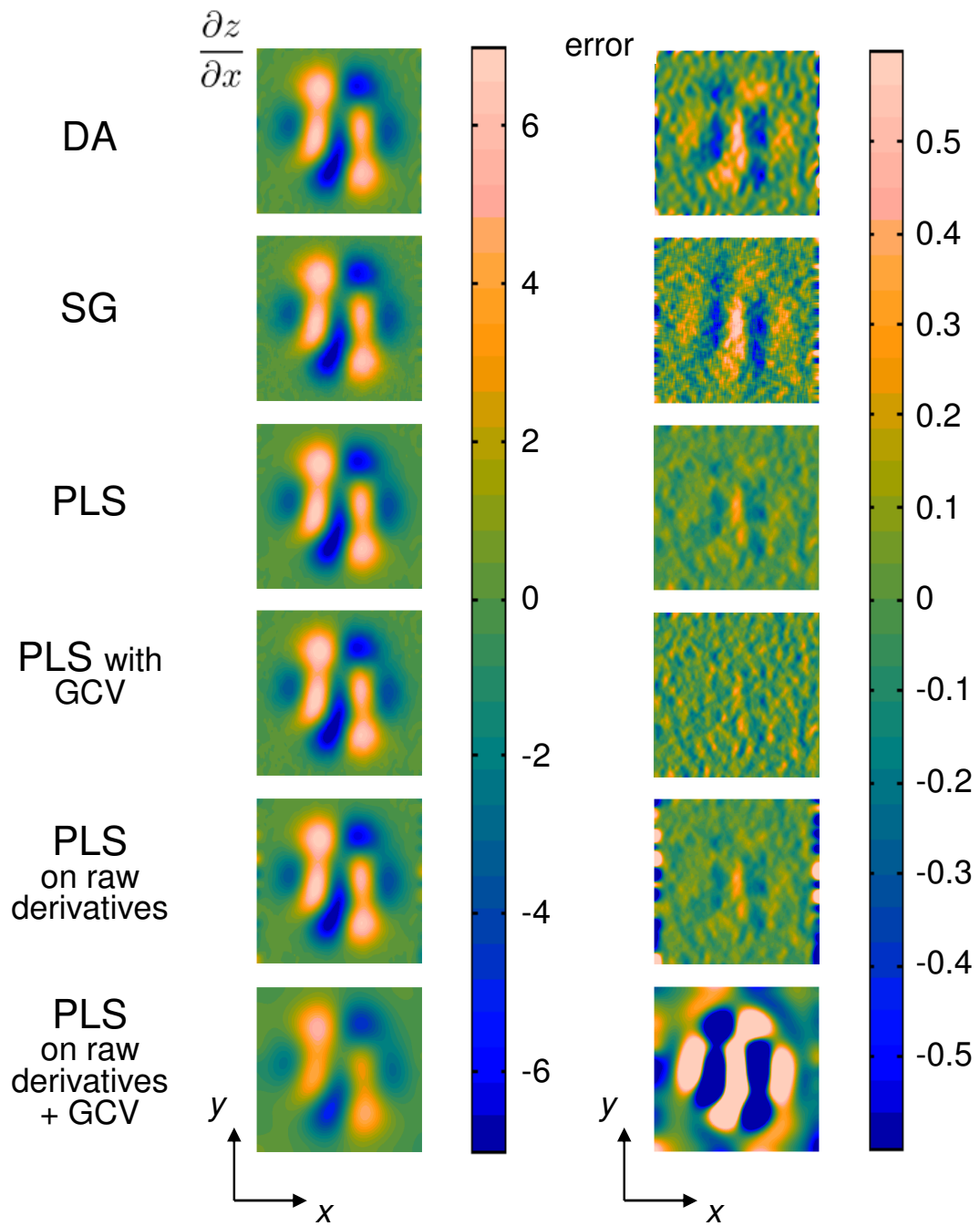
The quality of the reconstruction has been investigated using the same kind of criterion as previously: the average distance between the reconstructed strains and the exact ones defined as:

$$d_\varepsilon = \left\langle \sqrt{(\varepsilon_{xx}^{\text{rec.}} - \varepsilon_{xx}^{\text{exact}})^2 + 2(\varepsilon_{xy}^{\text{rec.}} - \varepsilon_{xy}^{\text{exact}})^2 + (\varepsilon_{yy}^{\text{rec.}} - \varepsilon_{yy}^{\text{exact}})^2} \right\rangle, \quad (29)$$

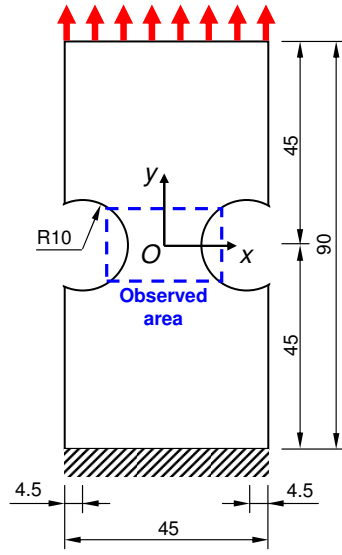
where  $\langle \cdot \rangle$  always denotes a spatial averaging of the data all over the map.

The results obtained are summarized in table 2 and  $\varepsilon_{xx}$  maps as well as the corresponding error maps  $(\varepsilon_{xx}^{\text{rec.}} - \varepsilon_{xx}^{\text{exact}})$  are shown in figure 7.

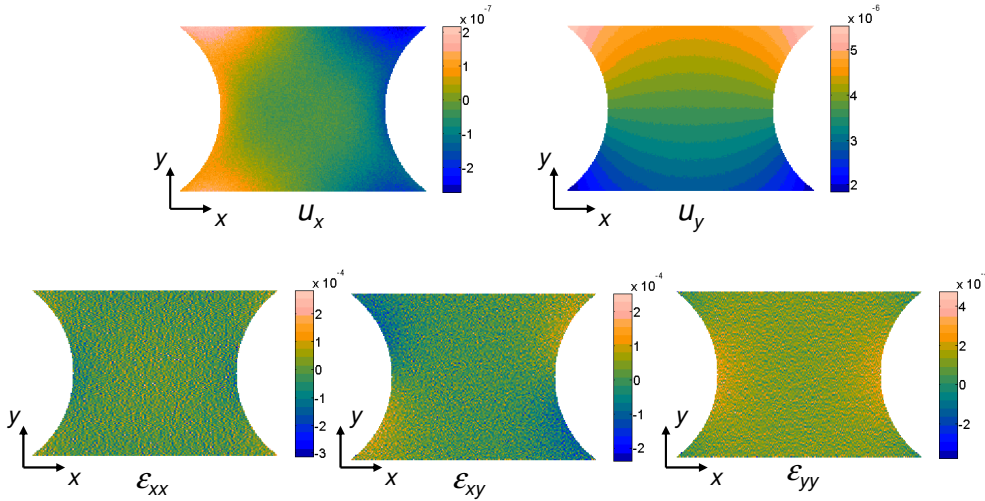




**Figure 5.** Reconstructed  $\frac{\partial z}{\partial x}$  and corresponding error maps for the different smoothing methods.



**Figure 6.** Model of the tensile sample used to simulate a mechanical test (through a finite elements code).



**Figure 7.** Noised displacements maps and raw strain maps for the simulated tensile test.

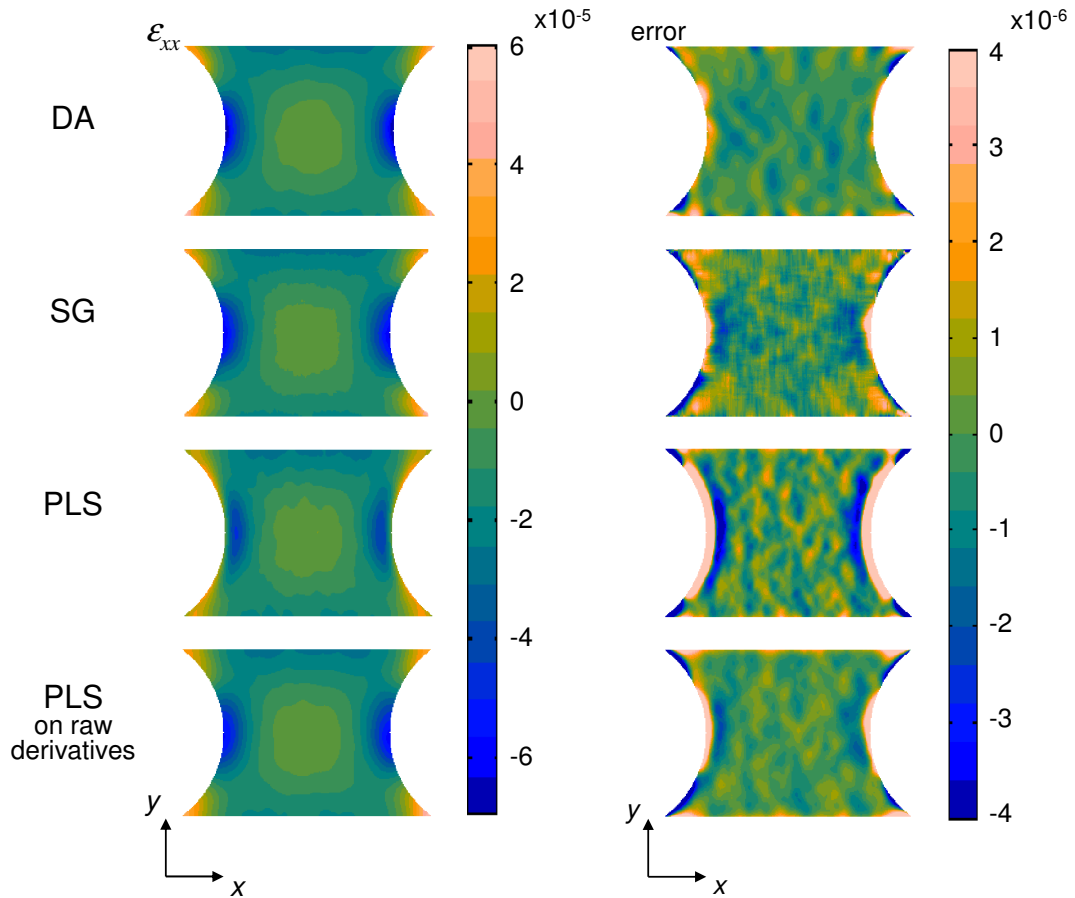
It can be observed that, for an optimal value of the smoothing parameter, PLS works worse than both DA and SG. Indeed, the obtained average deviation in these conditions ( $1.64 \times 10^{-5}$ ) is an order of magnitude higher than the ones obtained for the two others algorithms ( $1.47 \times 10^{-6}$  for DA and  $2.03 \times 10^{-6}$  for SG). This counterbalances the conclusions of the previous section concerning the best performances of PLS. It appears especially that PLS does not manage to reconstruct correctly the edges of the circular notches, where the absolute deviation from the exact value of the strain is maximum.

An automatic choice of  $\lambda$  by minimization of GCV produces worse results: it leads to smoothing parameters  $\lambda = 34.46$  for the smoothing of  $u_x$  and  $\lambda = 0.0282$  for the smoothing of  $u_y$  and to an average deviation of the strains equal to  $5.83 \times 10^{-5}$ . The

**Table 2.** Performance results for the different algorithms with optimized smoothing parameters for the noised maps from a FEM simulation.

Algorithm	DA	SG	PLS	PLS with GCV
Smoothing parameter	$h = 37$ pixels	$h = 31$ pixels	$\lambda(u_x) = 354.8;$ $\lambda(u_y) = 89.1$	$\lambda(u_x) = 34.5;$ $\lambda(u_x) = 0.028$
$d_\varepsilon$	$1.47 \cdot 10^{-6}$	$2.03 \cdot 10^{-6}$	$1.64 \cdot 10^{-5}$	$5.83 \cdot 10^{-5}$

Algorithm	PLS on raw derivatives	PLS on raw derivatives + GCV
Smoothing parameter	$\lambda(\varepsilon_{xx}) = 1000;$ $\lambda(\varepsilon_{yy}) = 1000;$ $\lambda(\varepsilon_{xy}) = 501.2$	$\lambda(\varepsilon_{xx}) = 11440;$ $\lambda(\varepsilon_{yy}) = 6986;$ $\lambda(\varepsilon_{xy}) = 4917$
$d_\varepsilon$	$2.13 \cdot 10^{-6}$	$3.66 \cdot 10^{-6}$



**Figure 8.** Reconstructed  $\varepsilon_{xx}$  and corresponding error maps for the different smoothing methods.

most probable explanation of this fact is that the GCV minimization optimizes the reconstruction of the smoothed data. In the present case, this is not directly smoothed values that are looked at but their spatial derivatives. To work efficiently, an automatic choice of the smoothing parameter should consider minimizing a criterion based on the derivatives (which is not really applicable since the derivation always increases dramatically the influence of noise). It can thus be concluded that automatic choice of the smoothing parameter is not suitable to the smoothing and derivation of full-field kinematic measurements.

However, when smoothing directly the raw derivatives as previously done for the noised analytical function, the reconstruction by PLS works more efficiently. Indeed, for an optimal set of  $\lambda$ 's, the average deviation obtained is of the same range as the one obtained with the two other algorithms ( $2.13 \times 10^{-6}$ ).

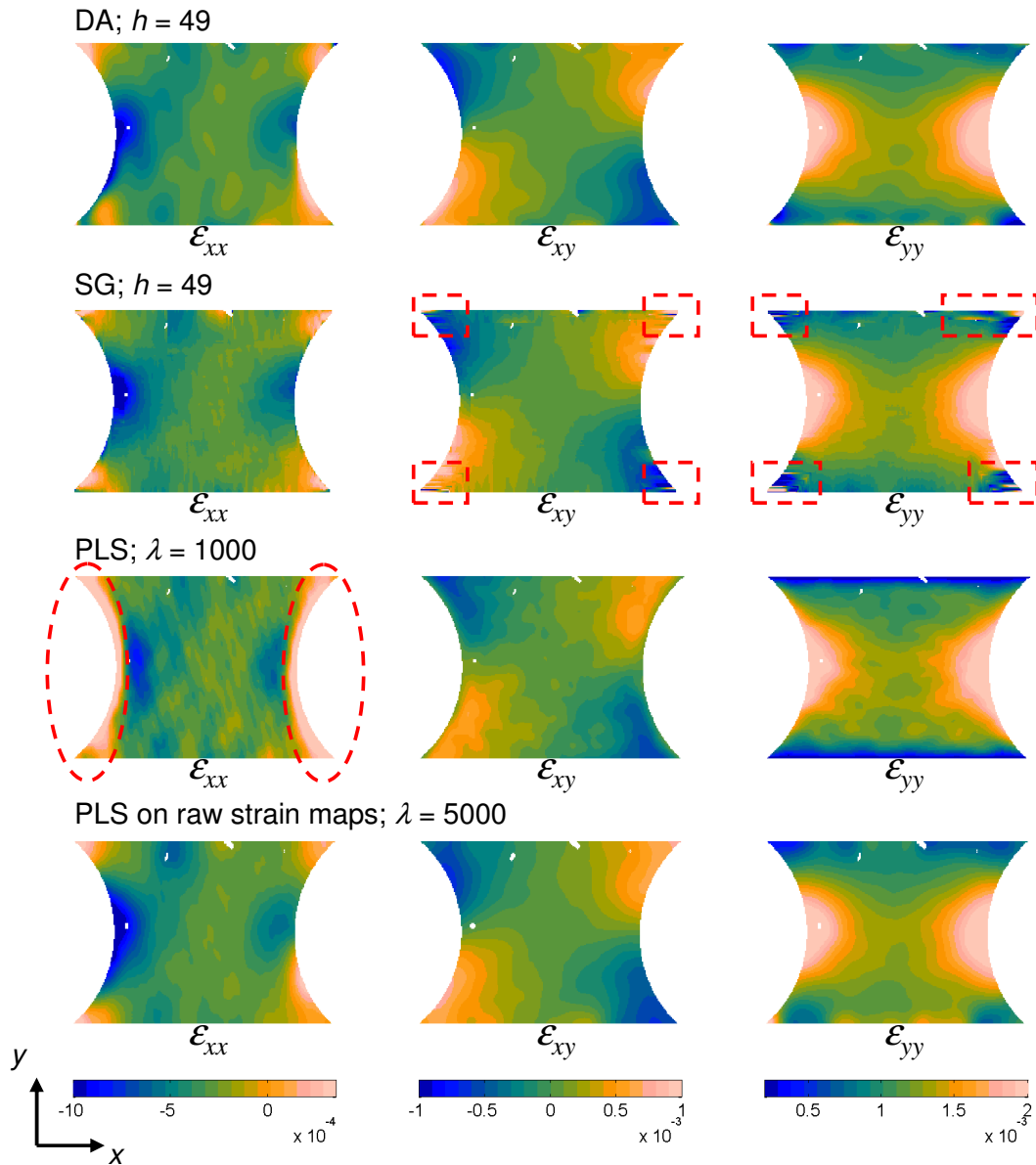
In conclusion, it can be said that, if PLS is not the best procedure for this problem (possibly because the highest gradients are located on the edges of the maps which was not the case previously, with the analytical function), it shows results comparable with the ones produced by DA or SG (especially when performing the derivation before the smoothing) and remains far faster.

### *4.3. Smoothing of experimental kinematic maps*

Finally, the smoothing procedure by PLS has been tested onto true experimental kinematic maps. To illustrate this final step, two mechanical test are considered.

*4.3.1. Smoothing of experimental kinematic maps of a notched specimen under tensile test* The first one is the experimental set up of the tensile test on a double notched specimen considered in the previous section. The displacement maps were obtained by the grid method. This one consists in computing  $u_x$  and  $u_y$ , the two components of the in-plane displacement field, from a periodical pattern deposited onto the surface to study by a phase-stepping algorithm [44, 45]. The grid here had a pitch of  $100 \mu\text{m}$  leading to kinematic maps of  $159 \times 231$  pixels<sup>2</sup> (a measurement point corresponds to a period).

The raw displacements fields have been processed by the different algorithms leading to strain maps represented in figure 9 (the few missing points within the region of interest correspond to small zones where the grid was not transferred properly onto the specimen). In this figure, it can be clearly observed that both DA and PLS directly applied on raw strain maps lead to accurate maps that can be compared to the ones obtained on simulated data even if the experimental results show an asymmetry between the left and right sides due to some misalignment between the axis of the sample and the one of the testing device during the test (figure 8). On the contrary, SG and PLS applied before derivating displacement maps do not work. Strain maps resulting from SG show a dithering effect probably due to the important influence of the edges (zones surrounded by dashed rectangles in figure 9). For the same reason, PLS applied before



**Figure 9.** Experimental in-plane strain maps obtained on the double-notched tensile sample by applying different smoothing procedures (size of the region of interest:  $16 \times 23 \text{ mm}^2$ ).

derivating displacement maps does not provide good results. This is especially the case for  $\varepsilon_{xx}$  (zones surrounded by dashed ellipses in figure 9) that is here the most difficult map to reconstruct as the signal to noise ratio is the lowest (the  $x$  direction is not highly solicited in this test). So, for such a case, with high gradients on the edges and potentially low signal to noise ratio, it seems preferable to apply a point-to-point differentiation before smoothing by PLS rather than applying the smoothing and then differentiating the smoothed displacement maps: this produces results comparable with the ones of DA.

The most probable reason for this is that the probability distribution of the

noise of those experimental maps is not normal. But, when applying a point-to-point differentiation, if it leads to an increase in the level of noise as already mentioned, this also modifies the probability distribution to get closer to a “bell-shaped distribution”. Indeed, due to the central limit theorem, the application of a  $n^{\text{th}}$  differentiation operation onto random variables following the same probability distribution converges to a normal distribution when  $n$  increases. Practically, in most experimental cases, a first differentiation is enough to assume a “normalization” of the distribution.

To illustrate this effect, three different non-normal distributions have been studied:

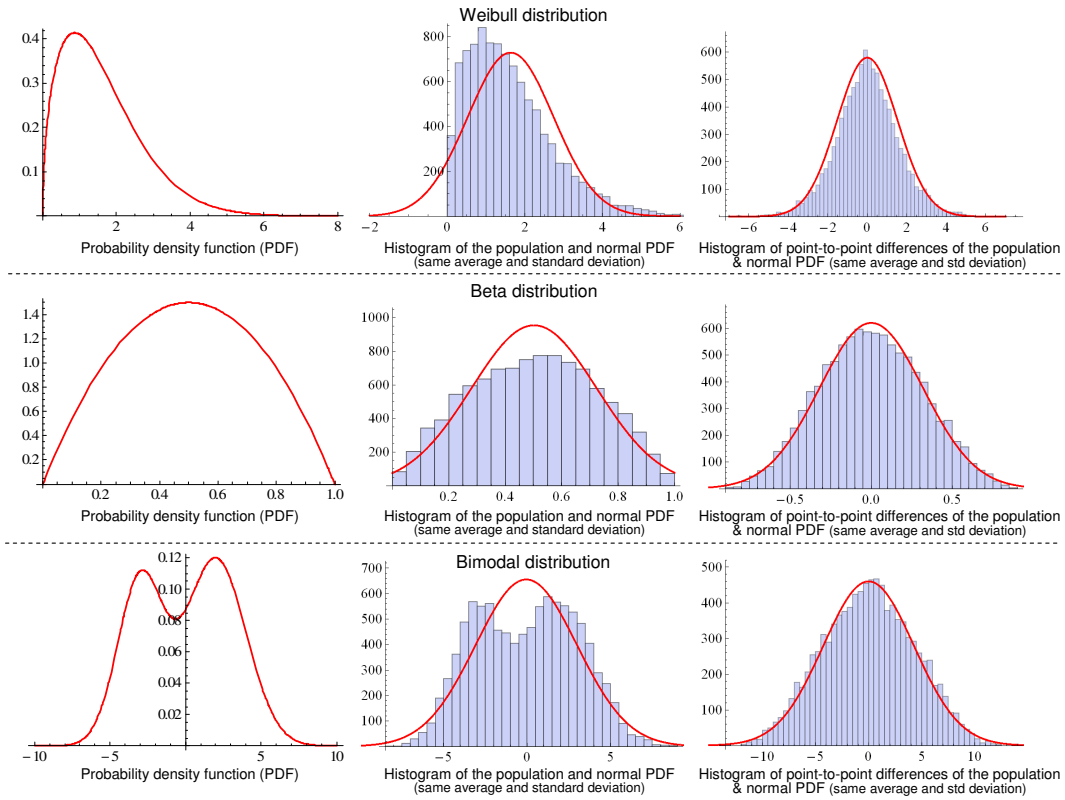
- a Weibull distribution (scale parameter = 1.5; shape parameter = 1.8);
- a beta distribution (shape parameters:  $\alpha = 2$ ;  $\beta = 2$ );
- a bimodal distribution, mixture of two normal distributions (contribution for 40 % of a normal law with mean of -3 and standard deviation of 1.5 and 60 % of a normal law with mean of 2 and standard deviation of 2).

For each of them, a population of 10,000 numbers following these distributions have been generated. Figure 10 shows for each the probability density function, a histogram of the population with a normal probability density function having the same average and standard deviation and a histogram of the point-to-point differences of the population with a normal probability density function having the same average and standard deviation of this new population. This figure clearly indicates that the differentiation leads to a more “normalized” distribution of the noise even if it impairs the signal to noise ratio.

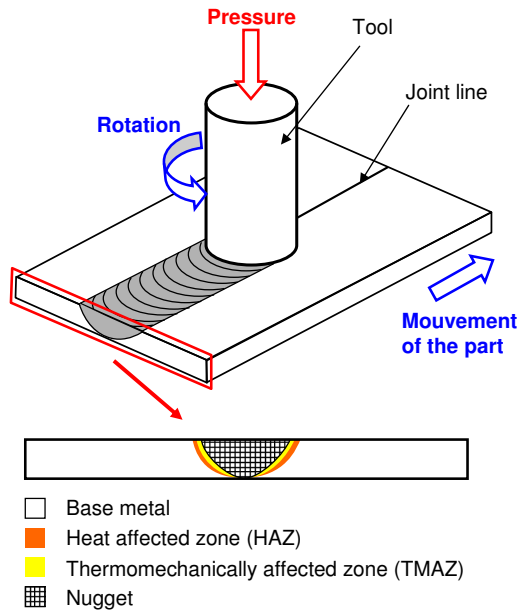
So, as mentioned in section 3.4, the underlying hypothesis behind PLS is that the noise is gaussian. This must be the reason why, on these experimental displacement maps, applying the derivation before PLS works better than doing it in the reverse order.

*4.3.2. Smoothing of experimental kinematic maps of a specimen with a FSW joint under tensile test* The second experimental data came from a study of the elasto-plastic behaviour of a magnesium sample presenting a friction-stir welding joint [46]. The principle of friction-stir welding (FSW) is to use a tool whose speed generates heat, softening locally the material. This tool mechanically intermixes the two pieces of metal at the place of the joint (Figure 11(a)). This thermomechanical process leads to different regions within the material (base metal, heat affected zone, thermomechanically affected zone, nugget) that present different mechanical properties. A mechanical test performed onto such a sample produces heterogeneous strain maps with high gradients zones (especially, within the elasto-plastic domain).

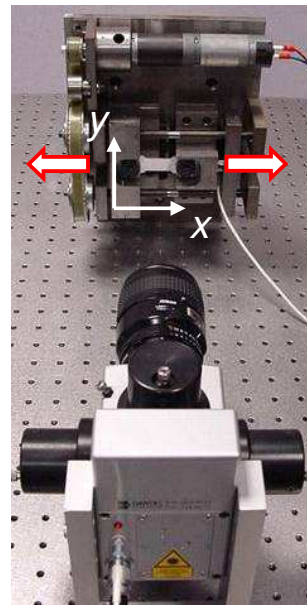
The sample has been tested on a small home-made micro tensile machine [47]. Full-field measurements have been done through the thickness of the sample thanks to an Electronic speckle pattern interferometric system (ESPI) [48]. The spatial resolution is



**Figure 10.** Effect of the point-to-point differentiation on several statistical populations following different probability distributions.



(a) Principle of the friction-stir welding

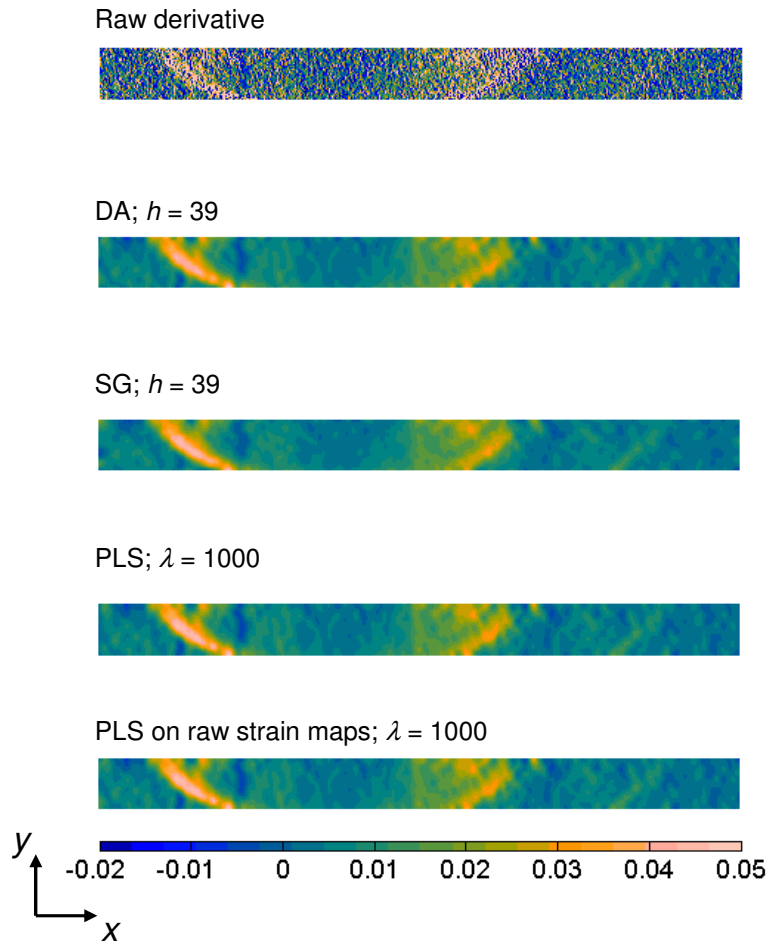


(b) Experimental setup

**Figure 11.** Tensile test of a friction-stir welded sample (FSW) and full-field measurement by Electronic Speckle Pattern Interferometry (ESPI).

16  $\mu\text{m}$  for a size of sample of  $1.8 \times 22 \text{ mm}^2$  ( $109 \times 1361 \text{ pixels}^2$ ). A view of the setup is shown in figure 11(b).

Figure 12 shows the raw  $\varepsilon_{xx}$  map and the reconstructed  $\varepsilon_{xx}$  maps obtained for DA, SG and PLS applied before and after differentiation (the smoothing parameters indicated in the figure). In this case, less severe than the tensile test on the double-notched specimen, the four procedures give very close maps. So, it can be concluded that PLS manages to reveal strain localizations due to the FSW joint tested in its plastic domain.



**Figure 12.**  $\varepsilon_{xx}$  maps obtained on the FSW tensile sample by applying different smoothing procedures (size of the sample:  $1.8 \times 22 \text{ mm}^2$ ).

## 5. Conclusion and prospects

In this paper, the possibility to use the penalized least squares smoothing procedure (PLS) as an alternative way to process kinematic full-field measurements has been addressed. At the end of this study, it can be asserted that PLS manages to produce valid strain maps from displacements measurements, with an accuracy comparable to



the techniques currently set up such as diffuse approximation (DA) or Savitzky-Golay smoothing filter (SG). Nevertheless, a few precautions have to be taken.

- For kinematic measurements presenting high gradients on the edges of the studied region and showing low signal to noise ratio, it is highly preferable to apply PLS on raw strain fields, after a spatial derivation of displacement fields thanks to a simple finite differences procedure.
- The automatic choice of the smoothing parameter  $\lambda$  by minimizing the generalized cross validation (GCV) does not produce optimal results. It is thus advised to choose “manually”  $\lambda$  taking into account engineering considerations.

Compared to other procedure such as DA or SG, PLS presents two main advantages:

- the computation time of the algorithm is independent of the value of the smoothing parameter and highly lower than both DA and SG;
- The  $h$  span parameter of DA and SG is an odd integer that gives only a discrete control of the amount of smoothing to apply; on the contrary, the smoothing parameter  $\lambda$  of PLS is a real continuous value.

However, a few drawbacks can be underlined:

- $\lambda$ , the real smoothing parameter, has no direct physical meaning (whereas  $h$  for DA and SG is directly related to the spatial resolution);
- due to the global nature of the procedure, the amount of smoothing cannot be adapted locally to match local gradients (which can be imagined with DA even if it is hardly applied).

Finally, PLS appears to be a robust alternative to currently applied smoothing techniques that can be preferentially selected whenever one has to process large maps that need a large amount of smoothing due to its high computation speed.

Several prospects of this work are envisaged:

- the procedure can easily be extended to 3D problems which would allow one to smooth a whole set of 2D maps both spatially and temporally or to process 3D tomographic measurements that have become more and more popular in experimental mechanics applications;
- a criterion based on considerations of signal to noise ratio and of an empirical computation rather than GCV minimization could be defined.

At last, a long term prospect could be to use another regularization method. Replacing equation 4 by this expression:

$$Q = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{i=1}^{n-d} |\Delta^d \hat{y}_i|, \quad (30)$$

one get access to the so called “total variation” scheme that was introduced within the context of image processing in [49]. This method, using the absolute value of the  $d^{th}$

order difference of reconstructed values as a penalization term instead of the its square value, leads to a more accurate determination of discontinuities occurring within the data. The drawback is that the minimization of  $Q$  defined in equation 30 has no more close-form solution and needs a stage of numerical optimization (which is more time consuming). This approach has already been successfully set up within the numerical differentiation context for 1D non-missing data [50].

## Acknowledgements

The authors would like to thank Pr. Paul Eilers for some useful exchanges on the principles and developments of the penalized least squares algorithm.

They also would like to thank Dr. Marco Rossi for providing the 2D Savitzky-Golay algorithm used in this study.

Finally, they would like to thank Mr. Alain Prévot for his advice on statistical issues, Ms. Virginie Jamar and Mr. Jeremy Blanks for proofreading and English corrections; and Dr. Laurence Fouilland-Paillé for proofreading.

## References

- [1] P. Feissel. From displacement to strain (Chap. 7). In *Full-field measurements and identification in solid mechanics*, pages 191–221. John Wiley & Sons, 2013.
- [2] R.R. Cordero, J. Molimard, F. Labbé, and A. Martínez. Strain maps obtained by phase-shifting interferometry: an uncertainty analysis. *Optics communications*, 281(8):2195–2206, 2008.
- [3] R Moulart, S. Avril, and F. Pierron. Identification of the through-thickness rigidities of a thick laminated composite tube. *Composites Part A*, 37(2):326–336, 2006.
- [4] F. Pierron, B. Green, and M.R. Wisnom. Full-field assessment of the damage process of laminated composite open-hole tensile specimens. Part I: Methodology. *Composites Part A*, 38:2321–2332, 2007.
- [5] S. Avril, P. Feissel, F. Pierron, and P. Villon. Estimation of the strain field from full-field displacement noisy data. Comparing finite elements global least squares and polynomial diffuse approximation. *European Journal of Computational Mechanics*, 17(5-7):857–868, 2008.
- [6] S. Avril, P. Feissel, F. Pierron, and P. Villon. Comparison of two approaches for differentiating full-field data in solid mechanics. *Measurement Science & Technology*, 21(1):015703, 2010.
- [7] C. Loader. Smoothing: Local regression techniques. In *Handbook of Computational Statistics*, pages 539–563. Springer, 2004.
- [8] J.S. Simonoff. *Smoothing Methods in Statistics*. Springer-Verlag, 1996.
- [9] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.
- [10] E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.
- [11] M.P. Wand and M.C. Jones. *Kernel smoothing*. Chapman & Hall, 1995.
- [12] J. Fan and I. Gijbels. *Local polynomial modelling and its applications*. Chapman & Hall, 1996.
- [13] W.S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.
- [14] W.S. Cleveland. LOWESS: A program for smoothing scatterplots by robust locally weighted regression. *The American Statistician*, 35(1):54, 1981.
- [15] W.S. Cleveland and S.J. Devlin. Locally-Weighted Regression: An Approach to Regression

- Analysis by Local Fitting. *Journal of the American Statistical Association*, 83(403):593–610, 1988.
- [16] C. Loader. *Local regression and likelihood*. New York: Springer-Verlag, 1999.
- [17] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36:1627–1639, 1964.
- [18] P. A. Gorry. General least-squares smoothing and differentiation by the convolution (Savitzky-Golay) method. *Analytical Chemistry*, 62:570–573, 1990.
- [19] S. Efromovich. *Nonparametric curve estimation*. Springer, 1999.
- [20] D.L. Donoho and Johnstone I.M. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
- [21] J.H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76(376):817–823, 1981.
- [22] L. Breiman, J.H. Friedman, R. Olshen, and C. Stone. *Classification and regression trees*. Wadsworth, 1984.
- [23] T. Hastie and R. Tibshirani. Generalized additive models. *Statistical Science*, 1:297–310, 1986.
- [24] B. Cheng and D.M. Titterton. Neural networks: a review from a statistical perspective. *Statistical Science*, 9(1):2–54, 1994.
- [25] P. H. C. Eilers. A perfect smoother. *Analytical Chemistry*, 75:3631–3636, 2003.
- [26] D. Garcia. Robust smoothing of gridded data in one and higher dimensions with missing values. *Computational Statistics & Data Analysis*, 54(4):1167–1178, 2010.
- [27] G. Bohlmann. Ein Ausgleichungsproblem. In *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, pages 260–271. 1899. In German.
- [28] E.T. Whittaker. On a new method of graduation. *Proceedings of the Edinburgh Mathematical Society*, 41:63–75, 1923.
- [29] E.T. Whittaker. On the theory of graduation. *Proceedings of the Edinburgh Mathematical Society*, 44:77–83, 1924.
- [30] R. Henderson. A new method of graduation. *Transactions of the Actuarial Society of America*, 25:29–40, 1924.
- [31] R. Henderson. Further remarks on graduation. *Transactions of the Actuarial Society of America*, 26:52–57, 1925.
- [32] R. Henderson. *Mathematical Theory of Graduation*. Actuarial Society of America, 1938.
- [33] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Doklady Akademii Nauk SSSR*, 151:501–504, 1963. in Russian; translated in *Soviet Mathematics*, 4:1035–1038,.
- [34] I.J. Schoenberg. Spline functions and the problem of graduation. *Proceedings of the National Academy of Sciences of the United States of America*, 52:947–950, 1964.
- [35] C.H. Reinsch. Smoothing by spline functions. *Numerische Mathematik*, 10:177–183, 1967.
- [36] P.H.C. Eilers, I.D. Currie, and M. Durbán. Fast and compact smoothing on large multidimensional grids. *Computational Statistics & Data Analysis*, 50(1):61–76, 2006.
- [37] M. J. Buckley. Fast computation of discretized thin-plate smoothing spline for image data. *Biometrika*, 81(2):247–258, 1994.
- [38] W.C. Yueh. Eigenvalues of several tridiagonal matrices. *Applied mathematics e-notes*, 5:66–74, 2005.
- [39] G. Strang. The discrete cosine transform. *SIAM Review*, 41:135–147, 1999.
- [40] P. Craven and G. Wahba. Smoothing noisy data with spline functions. Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numerische Mathematik*, 31:377–403, 1979.
- [41] G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.
- [42] G. Wahba. *Spline models for observational data*. Society for Industrial Mathematics, 1990.
- [43] S. Avril, F. Pierron, Y. Pannier, and R. Rotinat. Stress reconstruction and constitutive parameter

- identification in plane-stress elasto-plastic problems using surface measurements of deformation fields. *Experimental Mechanics*, 48(4):403–419, 2008.
- [44] Y. Surrel. Moiré and grid methods: a signal-processing approach. In *SPIE 2342, Interferometry '94: Photomechanics*, pages 213–220. 1994.
- [45] Y. Surrel. Fringe analysis. In P.K. Rastogi, editor, *Photomechanics*, pages 55–102. Springer, 2000.
- [46] L. Commin, M. Dumont, R. Rotinat, F. Pierron, J.-E. Masse, and L. Barrallier. Influence of the microstructural changes and induced residual stresses on tensile properties of wrought magnesium alloy friction stir welds. *Materials Science and Engineering: A*, 551:288–292, 2012.
- [47] R. Moulart, R. Rotinat, and F. Pierron. Full-field evaluation of the onset of microplasticity in a steel specimen. *Mechanics of Materials*, 41:1207–1222, 2009.
- [48] R. Jones and C. Wykes. *Holographic and Speckle Interferometry: A discussion of the theory, practice and application of the techniques*. Cambridge University Press, 1990.
- [49] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60(1-4):259–268, 1992.
- [50] R. Chartrand. Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics*, 2011.