

An algebraic method for multi-dimensional derivative estimation

Samer Riachy, Yara Bachalany, Mamadou Mboup and Jean-Pierre Richard

Abstract—This communication revisits the algebra-based results for derivative estimation presented by Fliess and co-authors in 2005. It is proposed, here, to consider multi-dimensional functions, namely scalar or vector fields of several variables. Such fields are locally represented by a vector Taylor series expansion, and a computation technique is presented so to put successive partial derivatives (for instance, the gradient, the Hessian matrix...) as functions of iterated integrals of the measured quantities.

I. INTRODUCTION

Classical field theory [2] (not to be confused with algebraic field theory) is a well developed branch of mathematics, suitable to describe phenomena with distributed parameters such as fluid mechanics, heat transfer, econometrics [10], optimization [3] and many others.

Basic ingredients of the field theory are operators such as gradient vector, Jacobian, Laplacian, Hessian, divergence, curl, etc... Close looking to these operators, one can notice that they are combinations of successive partial derivatives of the field (scalar or vector) with respect to its coordinates.

As stated above, distributed parameters systems are modeled as fields (scalar or vector), and the above mentioned operators are manipulated in order to have a qualitative idea for a given problem. If one seeks quantitative description, numerical simulations have to be done.

One of the major problems in numerical simulation is the dilemma of accuracy and computational cost in estimating the above operators.

As an example, consider the problem of edge detection in image processing. Roughly speaking, an image is a scalar bidirectional field (x, y) , and the problem of edge detection can be solved by computing the Laplacian at each of the image pixels. The Laplacian operator is given by:

$$L = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}.$$

Thus estimating the Laplacian reduces to estimating second order partial derivatives with respect to x and y .

S. Riachy and Jean-Pierre Richard: Projet ALIEN, INRIA FUTURS and Equipe SyNeR, LAGIS UMR CNRS 8146, Ecole Centrale de Lille, P.O. Box: 48, Cité Scientifique, 59651 Villeneuve d'Ascq, France. samer.riachy@ec-lille.fr, jean-pierre.richard@ec-lille.fr

Y. Bachalany: LAGIS, Université des Sciences et Technologies de Lille, 59651 Villeneuve d'Ascq, France yara.bachalany@ed.univ-lille1.fr

M. Mboup: Projet ALIEN, INRIA FUTURS and UFR de Mathématiques et Informatiques, université René Descartes (Paris 5) 45 rues des saints-Pères, Paris cedex 06 France mboup@math-info.univ-paris5.fr

Usually those quantities are estimated with use of finite differences, resulting in poor estimation in the presence of noise.

In this paper, we propose a methodology whose purpose is to compute an estimate for successive partial derivatives of a given field.

Robust and fast time derivation of noisy signals is now possible, thanks to an algebra-based approach initiated in [7], and oriented toward signal derivation in [4], [5], [6], [8], [9]. Advanced results on algebra-based derivative estimation can be found in [9], which basic concepts well be recalled in section II.

Based on a truncated Taylor expansion, n^{th} order derivatives are obtained through iterative integrations on the signal.

In [4], estimation of the derivatives of a multidimensional signal was considered. The estimator developed in [4] was based on a 2^{nd} order bidimensional Taylor expansion, and was constituted of a linear combination of the signal and its (first) integral. But, using the original signal (without integration) may introduce perturbations in noisy cases. In this paper, we present a more general estimators based on a multidimensional N^{th} order Taylor expansion. We will see that the resulting estimators are linear combination of iterated integration on the multidimensional signal.

A sufficiently smooth field can be locally represented by its vector Taylor expansion to the N^{th} order. Consider a truncation to the N^{th} order of some C^r , $r \geq 2$, function $f(\mathbf{x}) : U \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ defined on some compact set U , with $\mathbf{x} = (x_1, \dots, x_n)^T$:

$$f(\mathbf{x} + \Delta\mathbf{x}) = f(\mathbf{x}) + \mathbf{G}\Delta\mathbf{x} + \Delta\mathbf{x}^T \mathbf{H} \Delta\mathbf{x} + \dots, \quad (1)$$

$$\mathbf{G} = [I_{x_1} \quad \dots \quad I_{x_n}],$$

$$\mathbf{H} = \begin{bmatrix} I_{x_1^2} & I_{x_1 x_2} & \dots & I_{x_1 x_n} \\ I_{x_2 x_1} & I_{x_2^2} & \dots & I_{x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ I_{x_n x_1} & I_{x_n x_2} & \dots & I_{x_n^2} \end{bmatrix}.$$

The I_{x_i} , $I_{x_i^2}$ $i = 1 \dots n$ are partial derivatives of $f(\mathbf{x})$ with respect x_i . On the one hand, they can be seen as the 1^{st} and 2^{nd} derivative estimation of a mono-dimensional noisy signal. Thus the derivative estimation technique from [9] can be used. Section (II) recalls its basic concepts. On the other hand, a vector Taylor expansion is used to compute those quantities. Section (III) presents some examples.

Generic forms can also be computed resulting in families of estimators. One of them is used for the estimation of the off-diagonal terms for the Hessian matrix. This is also presented in section (III).

In section (IV), a comparison with finite differences method is provided for the computation of successive partial derivatives on an academic example.

II. NUMERICAL DIFFERENTIATION: A RECALL

This part recalls the framework introduced in [9]. Let $y(t) = x(t) + n(t)$ be a noisy observation on a finite time interval of a real-valued signal $x(t)$, the derivatives of which are to be estimated. The signal $x(t)$ is assumed to be analytic. It is known that an analytic signal can be expanded into a Taylor series:

$$x(t) = \sum_{i \geq 0} x^{(i)}(0) \frac{t^i}{i!}.$$

On some finite interval of time, a truncation of the Taylor expansion to some order N is acceptable as a ‘good’ approximation:

$$x_N(t) = \sum_{i \geq 0}^N x^{(i)}(0) \frac{t^i}{i!}, \quad t \in [0, T]. \quad (2)$$

The successive derivatives estimation requires some mathematical manipulation on (2) such as time derivation and integration by parts. Doing these operations several times in a row is tedious and lengthy, while doing the same in the operational domain reduces computation time and provides more compact forms. For this reason, applying the Laplace transform to (2) gives:

$$s^{N+1} \hat{x}_N(s) = s^N x(0) + s^{N-1} \dot{x}(0) + \dots + x^{(N)}(0),$$

where $\hat{x}_N(s)$ is the operational analog to $x_N(t)$ on $[0, T]$.

To be able to calculate individual estimators for the n^{th} order derivative (see [9] for a detailed discussion on what is called simultaneous and individual estimators and the difference between), a properly chosen differential operator is needed to annihilate all the remaining coefficients $x^{(j)}(0)$, $j \in \{0, \dots, N\} - \{n\}$. One differential operator has the form:

$$\Theta_{\kappa}^{N,n} = \frac{d^{n+\kappa}}{ds^{n+\kappa}} \frac{1}{s} \frac{d^{N-n}}{ds^{N-n}} \quad \kappa \geq 0.$$

It yields the following estimator for $x^{(n)}(0)$:

$$\frac{x^{(n)}(0)}{s^{\nu+n+\kappa+1}} = \frac{(-1)^{n+\kappa}}{(n+\kappa)!(N-n)!} \frac{1}{s^{\nu}} \Theta_{\kappa}^{N,n}(s^{N+1}x),$$

which is strictly proper whenever ν is of the form $\nu = N + 1 + \mu$, $\mu \geq 0$. A family of strictly proper estimators parameterized by κ, μ and N is obtained. Back to time domain, one obtains:

$$\tilde{x}_N^{(n)}(0) = \frac{(\nu+n+\kappa)!(-1)^{n+\kappa}}{(n+\kappa)!(N-n)!T^{\nu+n+\kappa}} \int_0^T \Pi(\tau) \hat{x}(\tau) d\tau,$$

with:

$$\Pi(\tau) = \sum_{i=0}^{N-n} \binom{N-n}{i} \frac{(N+1)!}{(n+i+1)!} \sum_{j=0}^{N+\kappa} \binom{n+\kappa}{j} \frac{(n+1)!}{(1+j-\kappa)!} \frac{(T-\tau)^{\nu+\kappa-j-2} (-\tau)^{i+j}}{(\nu+\kappa-j-2)!}.$$

As a concrete final form for the estimator, one may obtain after normalization of the integral into $[0, 1]$ forms like the ones below:

$$I_t = \frac{-30}{T} \int_0^1 (3 - 32\tau + 90\tau^2 - 96\tau^3 + 35\tau^4) f(T\tau) d\tau,$$

for $n = 1, N = 2, \mu = 5, \kappa = 1$.

$$I_{t^2} = \frac{336}{T^2} \int_0^1 (4 - 75\tau + 360\tau^2 - 700\tau^3 + 600\tau^4 - 189\tau^5) f(T\tau) d\tau.$$

for $n = 2, N = 3, \mu = 6, \kappa = 1$.

III. ESTIMATORS BASED ON THE VECTOR TAYLOR EXPANSION

Consider the N^{th} order vector Taylor series expansion for the scalar field $f(\mathbf{x})$. Note that the development in (1) may be expressed in a more general form as:

$$f(\mathbf{x}) = \sum_{i=0}^N \left\{ \frac{1}{i!} \left(\sum_{k=1}^n x_k \frac{\partial}{\partial x_k} \right)^i f(\mathbf{x}) \right\}_{\mathbf{x}=0}, \quad (3)$$

with:

$$\left(\frac{\partial^{i_1}}{\partial x_1^{i_1}} \frac{\partial^{i_2}}{\partial x_2^{i_2}} \dots \frac{\partial^{i_n}}{\partial x_n^{i_n}} \right) f(\mathbf{x}) := I_{x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}} := I_{\Pi_{l=1}^n x_l^{i_l}},$$

$$i_1 + i_2 + \dots + i_l = i.$$

The multivariable Laplace transform is a linear operator on a scalar field $f(\mathbf{x})$ with $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}_+^n$, transforming it into a scalar field $\hat{f}(\mathbf{s})$ with $\mathbf{s} = (s_1, \dots, s_n)^T \in \mathbb{C}^n$ and defined as follows:

$$\hat{f}(\mathbf{s}) = \int_{\mathbb{R}_+^n} f(\mathbf{x}) \exp^{-\mathbf{s}^T \mathbf{x}} d\mathbf{x}. \quad (4)$$

We follow the same reasoning as in section (II). The idea is to try to put the desired partial derivative (to be estimated) as a function of iterated integrals of the observed quantities. Computation are far more complex than the one-dimension case in section (II). We start by some examples, to show how to proceed and, then, we present a generic form for the computation of the off-diagonal terms of the Hessian matrix.

A. Introductory examples

Consider a scalar vector field depending on two variables $z = f(x, y)$. Its Taylor truncation (for $N = 2$) around $(0, 0)$ is given by:

$$f_2(x, y) = I(0, 0) + I_x(0, 0)x + I_y(0, 0)y + \frac{1}{2}I_{x^2}(0, 0)x^2 + \frac{1}{2}I_{y^2}(0, 0)y^2 + I_{xy}(0, 0)xy.$$

In the operational domain, and after omitting $(0, 0)$ from the above equation, one obtains:

$$I(s, p) = \frac{I}{sp} + \frac{I_x}{s^2p} + \frac{I_y}{sp^2} + \frac{I_{x^2}}{s^3p} + \frac{I_{y^2}}{sp^3} + \frac{I_{xy}}{s^2p^2} \quad (5)$$

1) *Estimation of the 1st derivative using a bidimensional Taylor expansion:* To compute I_x , we try to manipulate (5) so to isolate I_x . Start by multiplying (5) by s^3p^3 and then derivate two times with respect to p and once with respect to s . Next multiply by $\frac{1}{sp}$ and differentiate once with respect to s . The right-hand side of (5) reduces to $\frac{-2I_x}{s^2p}$. Applying the same operations to the left-hand side of (5) we obtain:

$$18I(s, p) + 30s \frac{\partial I(s, p)}{\partial s} + 18p \frac{\partial I(s, p)}{\partial p} + 30sp \frac{\partial^2 I(s, p)}{\partial s \partial p} + 3p^2 \frac{\partial^2 I(s, p)}{\partial p^2} + 5sp^2 \frac{\partial^3 I(s, p)}{\partial s \partial^2 p} + 6s^2 \frac{\partial^2 I(s, p)}{\partial s^2} + 6s^2p \frac{\partial^3 I(s, p)}{\partial s^2 \partial p} + s^2p^2 \frac{\partial^4 I(s, p)}{\partial s^2 \partial p^2} = -2 \frac{I_x}{s^2p}$$

Note that multiplying by s (or p) corresponds to derivate with respect to x (or y) in the time domain, which is not desirable. For this reason, multiply (6) by $\frac{1}{s^3p^3}$ and then apply the inverse transform (see appendix) to return back to time the domain. The following form is obtained:

$$\begin{aligned} & \frac{18}{4} \int_0^X \int_0^Y (X-x)^2(Y-y)^2 I(x, y) dx dy + \\ & \frac{30}{2} \int_0^X \int_0^Y (X-x)(Y-y)^2(-x) I(x, y) dx dy + \\ & \frac{18}{2} \int_0^X \int_0^Y (X-x)^2(Y-y)(-y) I(x, y) dx dy + \\ & 30 \int_0^X \int_0^Y (X-x)(Y-y)(-x)(-y) I(x, y) dx dy + \\ & \frac{3}{2} \int_0^X \int_0^Y (X-x)^2(-y)^2 I(x, y) dx dy + \\ & 5 \int_0^X \int_0^Y (X-x)(-x)(-y)^2 I(x, y) dx dy + \\ & \frac{6}{2} \int_0^X \int_0^Y (Y-y)^2(-x)^2 I(x, y) dx dy + \\ & 6 \int_0^X \int_0^Y (Y-y)(-x)^2(-y) I(x, y) dx dy + \\ & \int_0^X \int_0^Y (-x)^2(-y)^2 I(x, y) dx dy + \\ & = -2I_x \frac{X^4}{4!} \frac{Y^3}{3!}. \end{aligned}$$

After an appropriate change of variable to normalize the integrals over the interval $[0, 1]$, the final form is obtained:

$$I_x = \frac{-36}{X} \int_0^1 \int_0^1 (-48x - 36y - 160xy^2 - 180x^2y + 9 + 30y^2 + 45x^2 + 150x^2y^2 + 192xy) I(Xx, Yy) dx dy \quad (6)$$

2) *Estimation of the I_{x^2} , I_{xy} derivatives using a bidimensional Taylor expansion:* Similarly, to compute an estimator for I_{x^2} , one has to manipulate (5) so to isolate I_{x^2} . One way is to successively apply $\frac{1}{s^3p^4} \frac{\partial^2}{\partial s^2} \frac{\partial^2}{\partial p^2} s^3p^3$. The final form in time domain and after normalization of the integrals over $[0, 1]$ is the following:

$$I_{x^2} = \frac{180}{X^2} \int_0^1 \int_0^1 (-18x - 12y - 60xy^2 - 72x^2y + 3 + 10y^2 + 18x^2 + 60x^2y^2 + 72xy) I(Xx, Yy) dx dy. \quad (7)$$

Similarly applying $\frac{1}{s^3p^3} \frac{\partial}{\partial p} \frac{\partial}{\partial s} \frac{1}{sp} \frac{\partial}{\partial p} \frac{\partial}{\partial s} s^3p^3$ on (5), we obtain:

$$I_{xy} = \frac{144}{XY} \int_0^1 \int_0^1 (-240x^2y - 48y - 48x - 240xy^2 + 9 + 45y^2 + 45x^2 + 225x^2y^2 + 256xy) I(Xx, Yy) dx dy. \quad (8)$$

Note that (8) is symmetric in x and y .

B. Generic form for $I_{x_i y_j}$

The equivalent equation to (3) in the operational domain is obtained through the Laplace transform (4):

$$\hat{f}_N(S) = \sum_{n=0}^N \left\{ \frac{1}{n!} \frac{n!}{\prod_{i=1}^k s_i^{n+1}} \left[\sum_{i_1=0}^n \sum_{i_2=n-i_1}^n \sum_{i_3=n-i_2}^n \dots \sum_{i_{k-1}=n-i_{k-2}}^n \prod_{f=0}^k s_f^{i_f} I_{\prod_{f=0}^k x_f^{n-i_f}} \right] \right\}$$

with $i_k = kN - \sum_{l=1}^{k-1} i_l$. The elementary Laplace transformation used to obtain the above equation are shown in appendix.

To estimate $I_{x_i x_j}$ $\{i, j\} = 1, \dots, n$, multiply by the differential operator:

$$\Xi_{\kappa_1, \kappa_2}^{N, n, i, j} = \Omega \prod_{l \in \{1, \dots, n\} - \{i, j\}} \left(\frac{\partial^N}{\partial s_l^N} \right),$$

with

$$\Omega = \frac{\partial^{\kappa_1+1}}{\partial s^{\kappa_1+1}} \frac{\partial^{\kappa_2+1}}{\partial s^{\kappa_2+1}} \frac{1}{s_i s_j} \frac{\partial^{N-1}}{\partial s_i^{N-1}} \frac{\partial^{N-1}}{\partial s_j^{N-1}}.$$

It yields the following estimator for $I_{x_i x_j}$ $\{i, j\} = 1, \dots, n$

$$\frac{I_{x_i x_j}}{s_i^{\kappa_1+2} s_j^{\kappa_2+2} \prod_{l=1}^n s_l^i} = \alpha \frac{1}{\prod_{l=1}^n s_l^i} \Xi_{\kappa_1, \kappa_2}^{N, n, i, j} \prod_{l=1}^n (s_l^{N+1}) f(S),$$

with

$$\alpha = \frac{(-1)^{\kappa_1 + \kappa_2 + 2}}{(\kappa_1 + 1)!(\kappa_2 + 1)!(N!)^{n-2}[(N-1)!]^2}.$$

Similar generic forms for all terms in (3) can also be computed.

IV. NUMERICAL SIMULATIONS

In order to show the efficiency of the above developed estimators, simulations are achieved on a bidimensional signal given by $f(x, y) = \sin(\frac{1}{2}x^2 + \frac{1}{4}y^2 + 3)\cos(2x + 1 - e^y)$. The obtained results are compared with finite differences methods from [1]. A step sampling of (0.001×0.001) is used. As it is clear, estimators are computed on some elementary (non-infinitesimal) surface over $[0, X] \times [0, Y]$ with $X = Y = 0.06$ (for noise free simulations). Such surface involves $7 \times 7 = 49$ samples.

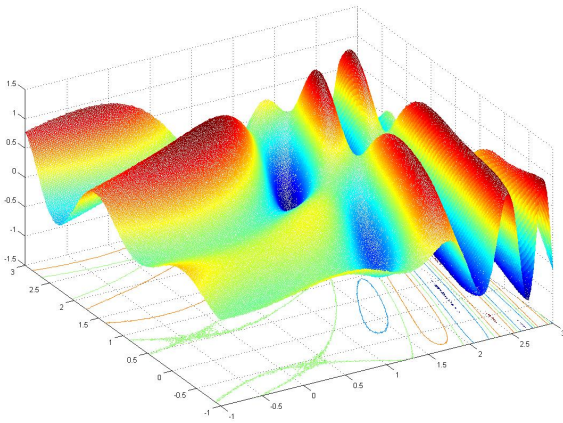


Fig. 1. 3-D plot of $f(x, y) = \sin(\frac{1}{2}x^2 + \frac{1}{4}y^2 + 3)\cos(2x + 1 - e^y)$

The derivatives are computed at a point (x_i, y_i) with $y_i = 2$ and x_i ranging from -2 to 7 . In fact, at each point of the line $(-2 \leq x_i \leq 7, y_i = 2)$, the elementary surface needed for the computation is taken around the particular point (x_i, y_i) .

Simulation results are shown in figures (2) and (3). The second order derivative (7) is plotted in a yellow thick line, while the finite differences ([1] section 25.3.23) is plotted in a red thin line in figure (2). Figure (3) shows the estimation of I_{xy} based on (8) in a thick yellow line and the finite differences from ([1] section 25.3.26) in a thin red line. In figures (2) and (3) the formal derivatives are in dashed green. It is easily seen that the finite differences lacks curvature in comparison to the algebra-based derivative.

Next, the surface is corrupted with noise $n(x, y)$. The noise level is measured by the signal to noise ratio in dB , i.e., $SNR = 10 \log_{10} \left(\frac{\sum_{i,j} |f(x_i, y_j)|^2}{\sum_{i,j} |n(x_i, y_j)|^2} \right)$. In the following simulation, the SNR is set to $SNR = 25 \text{ dB}$. Figure (4) shows a slice of the noisy surface in the plane $x_i = 2$ and for y_i ranging from -1 to 3 .

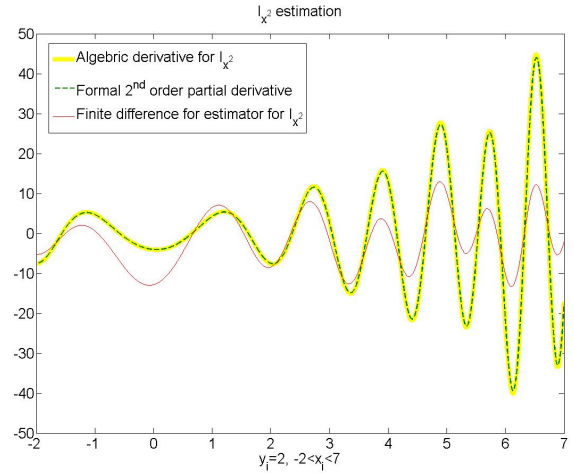


Fig. 2. Estimation of $I_{x,2}$

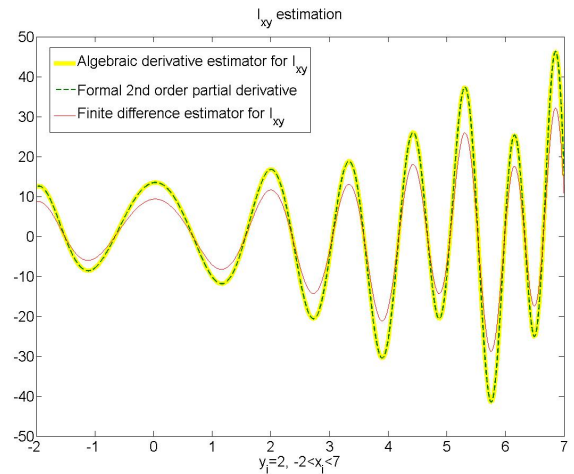


Fig. 3. Estimation of I_{xy}

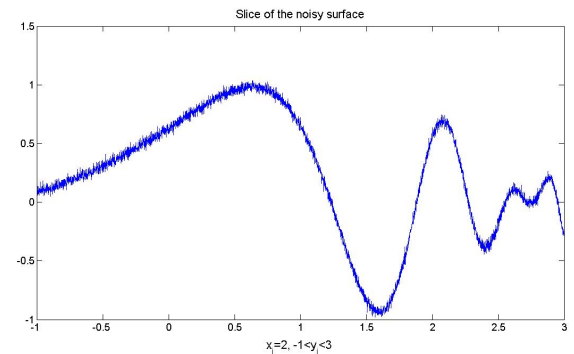


Fig. 4. A slice of the noisy surface at $x_i = 2$ and $-1 < y_i < 3$, 25 dB

A comparison between the algebra-based, and finite difference 1^{st} derivative estimation are compared in figures (5) and (6) respectively. The elementary surface needed for computation is of 80×80 samples. The same surface dimension (80×80) is used in both cases. The algebra-based approach shows robustness with respect to noise. Note that the estimation in (5) is subject to a delay. Following the same reasoning as in [9], an explanation through the Jacobi polynomials can be given. This is left for another work.

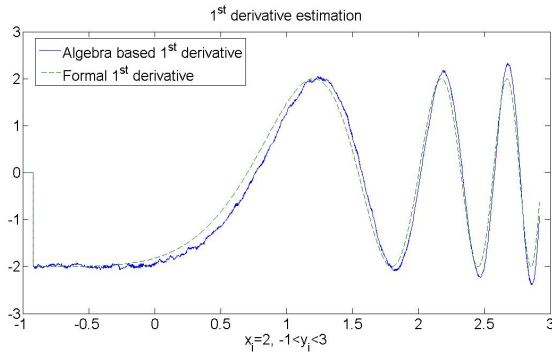


Fig. 5. Algebra-based estimation of the 1^{st} derivative, I_x

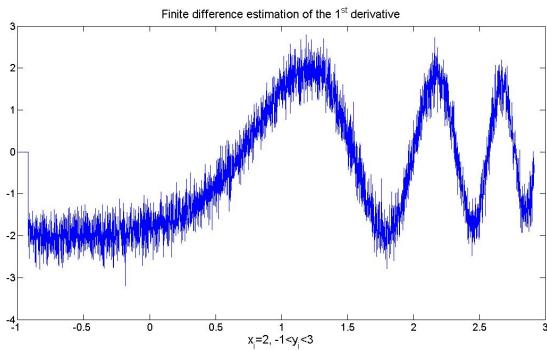


Fig. 6. Finite difference estimation of the 1^{st} derivative, I_x

We consider now the estimation of 2^{nd} order derivatives of the noisy surface, namely I_{x^2} and I_{xy} . The estimations obtained with an $SNR = 25 \text{ dB}$ for the algebra-based and finite difference estimators are affected by noise. When considering a higher signal to noise ratio ($SNR = 35 \text{ dB}$), one obtains the figures (7), (8) and (9). Figure (8) shows an estimation of I_{xy} . Computations are done on an elementary surface of 150×150 samples. The finite difference (in dashed red line) from ([1] section 25.3.26) shows robustness with respect to noise in comparison to the algebra-based estimation (in solid blue line), but less performance when the signal curvature increase. In figure (9), an estimation of I_{x^2} is plotted in a dash dotted blue line and compared to finite difference from ([1] section 25.3.24 in dashed red and 25.3.25 in dotted cyan line). The formal 2^{nd} derivative is shown in solid green, a square window of 80×80 is used in the simulation. Note that the algebra-based approach shows better performance. We hope to improve the estimation through the injection of a delay as in [9].

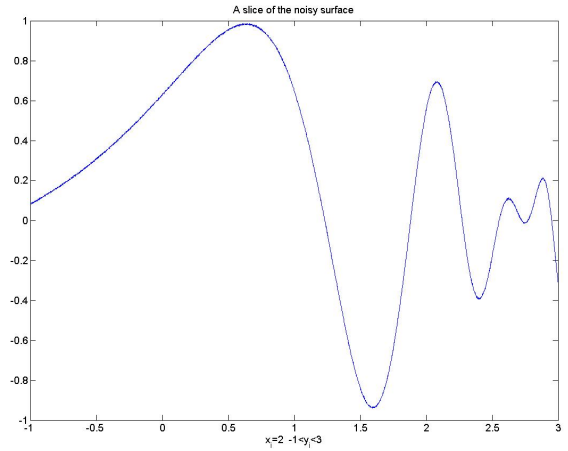


Fig. 7. A slice of the noisy surface at $x_i = 2$ and $-1 < y_i < 3$, 35 dB

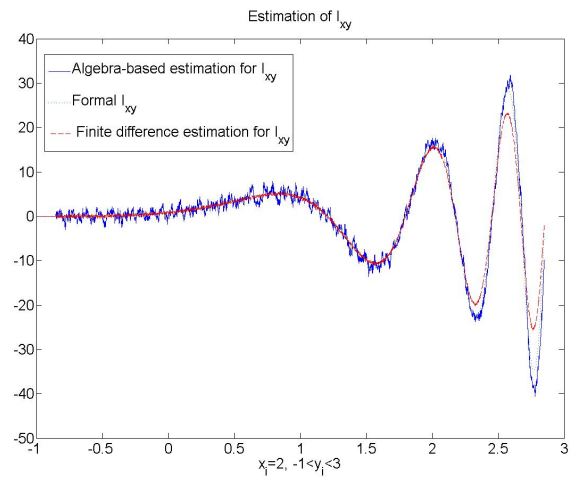


Fig. 8. Estimation of I_{xy}

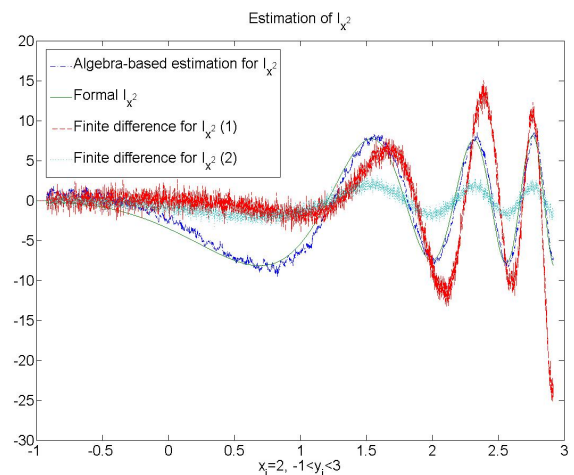


Fig. 9. Estimation of the 2^{nd} derivative, I_{x^2}

V. CONCLUSION

In this communication, we have presented an extension of the algebra-based derivative to multidimensional signals. Based on a vector Taylor expansion and the multivariable Laplace transform, estimators are computed. Those estimators are iterated integrals over the observed multidimensional signal. Our estimators are compared to finite differences methods from the literature. As demonstrated in [9], the introduction of a delay in the estimators improves considerably the quality of estimation and its robustness with respect to noise. A similar reasoning is hoped to be attached to the multidimensional estimators.

VI. APPENDIX

A. Laplace transform

In transforming from time domain into the operational domain, the following Laplace transform is used:

$$\mathcal{L}\left(\prod_{i=1}^k x_i^{n_i}\right) = \prod_{i=1}^k \frac{n_i!}{s^{n_i+1}} \quad (9)$$

To demonstrate this transform, start first with $k = 2$ which yields to:

$$\mathcal{L}\left(\frac{x^n y^m}{n! m!}\right) = \frac{1}{s^{n+1}} \frac{1}{p^{m+1}}.$$

Suppose $G(x, y) = \frac{x^n y^m}{n! m!}$ with $n, m \in \mathbb{Z}$. Then,

$$\begin{aligned} & \int_0^\infty e^{-py} \int_0^\infty e^{-sx} G(x, y) dx dy \\ &= \int_0^\infty e^{-py} \int_0^\infty e^{-sx} \frac{x^n y^m}{n! m!} dx dy \\ &= \int_0^\infty e^{-py} \frac{y^m}{m!} \int_0^\infty e^{-sx} \frac{x^n}{n!} dx dy \\ &= \int_0^\infty e^{-py} \frac{y^m}{m!} \frac{1}{s^{n+1}} dy \\ &= \frac{1}{s^{n+1}} \int_0^\infty e^{-py} \frac{y^m}{m!} dy \\ &= \frac{1}{s^{n+1}} \frac{1}{p^{m+1}}. \end{aligned}$$

The general formula (9) is easily deduced by recurrence on x_i .

B. Inverse transform

Back to time domain this inverse-transform is used:

$$\mathcal{L}^{-1}\left(\frac{1}{\prod_{i=1}^k s_i^{n_i}} \frac{d^{a_1+\dots+a_k} I}{\prod_{i=1}^k ds_i^{a_i}}\right) = \frac{1}{\prod_{i=1}^k (n_i - 1)!} \int_0^{X_1} \dots \int_0^{X_k} \prod_{i=1}^k (X_i - x_i)^{n_i-1} \prod_{i=1}^k (-x_i)^{a_i} I(x_1, x_2, \dots, x_k) dx_k \dots dx_1 \quad (10)$$

To demonstrate this inverse-transform, start first with $k = 2$, which yields:

$$\mathcal{L}^{-1}\left(\frac{1}{s^n} \frac{1}{p^m} \frac{\partial^{a+b} I}{\partial s^a \partial p^b}\right)$$

$$\text{Let } G = \frac{\partial^b I}{\partial p^b},$$

$$\begin{aligned} & \frac{1}{s^n p^m} \frac{\partial^a G}{\partial s^a} = \frac{1}{p^m} \frac{1}{s^n} \frac{\partial^a G}{\partial s^a} \\ &= \frac{1}{p^m} \int^{(n)} (-x)^a G dx = \int^{(n)} (-x)^a \frac{1}{p^m} \frac{d^b I}{dp^b} dx \\ &= \int^{(n)} (-x)^a \int^{(m)} (-y)^b I dy dx \\ &= \int^{(n)} \int^{(m)} (-x)^a (-y)^b I dy dx \\ &= \frac{1}{(n-1)!(m-1)!} \int_0^X \int_0^Y (X-x)^{n-1} (Y-y)^{m-1} (-x)^a (-y)^b I dy dx. \end{aligned}$$

The general formula (10) is easily deduced by recurrence on x_i .

REFERENCES

- [1] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions*, Dover, 1965.
- [2] D. E. Soper, *Classical field theory*, Wiley-Interscience publication, April 1976.
- [3] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge university press, (available at http://www.stanford.edu/boyd/cvxbook/bv_cvxbook.pdf).
- [4] M. Fliess, C. Join, M. Mboup and A. Sedoglavic, "Estimation des dérivées d'un signal multidimensionnel avec application aux images et aux vidéos", *Actes 20^e Col. GRETSI*, Louvain-la-neuve 2005 (available at <http://hal.inria.fr/inria-00001116>).
- [5] M. Fliess, C. Join, M. Mboup and H. Sira-Ramirez, "Compression différentielle de transitoire bruités", *C.R. Acad. Sci. Paris Ser. I*, vol.339, 2004, pp. 821-826.
- [6] M. Fliess, C. Join, M. Mboup and H. Sira-Ramirez, "Analyse et représentation de signaux transitoires: application à la compression, au débruitage et à la détection de ruptures", *Actes 20^e Coll. GRETSI*, Louvain-la-neuve, 2005 (available at <http://hal.inria.fr/inria-00001115>).
- [7] Michel Fliess and Hebertt Sira-Ramirez, "An algebraic framework for linear identification" *ESAIM: COCV* 9, 2003, pp. 151-168.
- [8] M. Mboup, "Parameter estimation via differential algebra and operational calculus", *in preparation*. (available at <http://hal.inria.fr>).
- [9] M. Mboup, C. Join and M. Fliess. "A revised look at numerical differentiation with an application to nonlinear feedback control". *15th Mediterranean conference on control and automation*, june 27-29, 2007. (available at <http://hal.inria.fr>).
- [10] Neudecker and Heinz and Magnus, *Matrix differential calculus with applications in statistics and econometrics*. New York: John Wiley and Sons, ISBN 978-0-471-91516-4, 1988, page 136.