



## Some solutions for non-homogeneous smoothing

Rutger A. Biezemans, Arsene Brice Zotsa Ngoufack, Chiara Cordier, Ayoub Belhachmi

### ► To cite this version:

Rutger A. Biezemans, Arsene Brice Zotsa Ngoufack, Chiara Cordier, Ayoub Belhachmi. Some solutions for non-homogeneous smoothing. [Research Report] IRMAR, University of Rennes 1. 2022. hal-03672087v2

**HAL Id: hal-03672087**

**<https://hal.science/hal-03672087v2>**

Submitted on 20 Jun 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

SEMAINE D'ETUDE MATHÉMATIQUES - ENTREPRISES

---

## Some solutions for non-homogeneous smoothing

---

Rutger A. BIEZEMANS  
CERMICS, École Nationale des Pont et Chaussées,  
MATERIALS project-team, Inria Paris

Arsene Brice ZOTSA NGOUFACK  
ALEA, Aix-Marseille Université

Chiara CORDIER  
LAREMA, Université d'Angers  
Unité Omiques et *Data Science*, ICO

Ayoub BELHACHMI  
AROMATH project-team,  
Inria Université Côte d'Azur

Rennes, 2 - 6 May 2022

## INTRODUCTION

When a physical signal is received and used, it is most of the time noisy and not smooth. In order to manipulate it digitally, it is necessary to convert it into a smooth signal, or regularize it. A classical way to do so is to convolve the signal with a Gaussian of fixed variance, which determines the scale at which the signal can be considered smooth.

In many situations, signals are not uniformly noisy, and some parts show important local defects, while the global signal is acceptable. Similarly, industrial constraints may require only local control of the smoothing in some regions, without having a globally smooth signal. All these applications call for non-uniform smoothing. Additionally, for industrial applications, the smoothing operation must be as fast as possible. From a practical point of view, the goal is to devise a regularization method that takes into account a certain non-uniform metric linked either to industrial constraints or to the initial regularity of the signal.

In Section I of this report, we present the theoretical mathematical tools that the regularization techniques used later are based upon. Section II presents a study of non-homogeneous smoothing of a signal in 1D based on discrete differential operators that are easy to construct. For the two-dimensional setting, we present non-homogeneous smoothing techniques based on convolution operations in Section III and in Section IV, the resolution of a partial differential equation with non-uniform coefficients is the key to local control of the regularization method.

## I. MATHEMATICAL APPROACHES

### I.1. Convolution.

**Definition 1.** A function is regular if it is at least of class  $\mathcal{C}^1$ .

**Definition 2.** The convolution product “ $\star$ ” between two functions  $f$  and  $\phi$  on  $\mathbb{R}^2$  is defined by

$$\phi \star f(x) = \int_{\mathbb{R}^2} \phi(y) f(x - y) dy.$$

**Theorem 3.** If  $f$  is a regular function on  $\mathbb{R}^2$  and  $\phi$  is any function on  $\mathbb{R}^2$ , then  $\phi \star f$  is also a regular function on  $\mathbb{R}^2$ .

The regular function that is used to convolve with is also called the convolution kernel. Consider the family of Gaussian convolution kernels

$$(1) \quad f_t(x) = \frac{1}{2\pi t} \exp\left(-\frac{\|x\|^2}{2t}\right),$$

indexed by a parameter  $t \in \mathbb{R}_+^*$ . The notation  $\|x\|$  is used for the standard euclidean norm of  $x \in \mathbb{R}^2$ .

For any function  $\phi$  on  $\mathbb{R}^2$ , the convolution  $\phi \star f_t$  is a uniform smoothing of  $\phi$ . Indeed, around each  $x \in \mathbb{R}^2$ , the convolution kernel decays with the same characteristic width.

For industrial applications, we are interested in a non-uniform smoothing of  $\phi$ . One way to do so is the following generalization of the Gaussian smoothing. We propose to compute a regularized function  $\tilde{\phi}$  by

$$(2) \quad \tilde{\phi}(x) = \int_{\mathbb{R}^2} \phi(y) \frac{1}{2\pi t(y)} \exp\left(-\frac{\|x - y\|^2}{2t(y)}\right) dy$$

where  $t(\cdot)$  is now a parameter function on smoothing. This can, in general, no longer be written as the convolution of  $\phi$  with another function.

Efficient numerical methods exist that evaluate convolutions, e.g. using discrete Fourier transforms. In order to avoid the evaluation of the integrals in (2) for each point  $x$  individually, and to benefit from efficient implementations of convolutions, we propose to choose for  $t(\cdot)$  a piecewise constant function. We can then use the following lemma to obtain an expression for  $\tilde{\phi}$  as the sum of convolution products.

**Lemma 4.** There exists a family  $(K_i)_i \subset \mathbb{R}^2$  such that for all  $i$ ,  $K_i \neq \emptyset$ ,  $\overset{\circ}{K}_i \cap \overset{\circ}{K}_j = \emptyset$  if  $i \neq j$  and  $\mathbb{R}^2 = \cup_i K_i$ .

With the notation of the foregoing lemma, let  $\chi_{K_i}$  be the characteristic function of  $K_i$  that is equal to 1 on  $K_i$  and vanishes outside  $K_i$ . Define the piecewise constant function  $t(\cdot)$  by  $t(y) = \sum_i t_i \chi_{K_i}$ . We obtain by linearity of the convolution:

$$\tilde{\phi}(x) = \int_{\mathbb{R}^2} \phi(y) \frac{1}{2\pi t(y)} \exp\left(-\frac{\|x-y\|^2}{2t(y)}\right) dy = \sum_i ((\phi \chi_{K_i} \star f_{t_i}))(x).$$

This expression is easy to calculate. Indeed we can use the FFTCONV function of the Python library SCIPY [4], which calculates a discrete convolution product automatically by an appropriate method.

**I.2. Link with the heat equation.** Another idea to obtain a non-uniform smoothing of  $\phi$  follows from interpreting the convolution  $\phi \star f_t$ , with  $f_t$  as defined in (1), as the solution to the so-called heat equation. More precisely, we have the following theorem.

**Theorem 5.** Let  $U : \mathbb{R}^2 \times \mathbb{R}_+ \rightarrow \mathbb{R}$ ,  $(x, t) \mapsto U(x, t)$  such that

$$\begin{cases} \partial_t U(x, t) = \Delta U(x, t), \\ U(x, 0) = \phi(x). \end{cases}$$

Here,  $\Delta = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2}$  denotes the Laplace operator with respect to the space variable  $x = (x_1, x_2)$ . Then

$$\phi \star f_t(x) = U(x, t).$$

In words, solving the heat equation with initial condition  $\phi$  yields as solution a smoothing of  $\phi$ . We propose to consider a non-homogeneous smoothing by solving the heat equation with a non-uniform diffusion coefficient  $C(x)$ . Possible definitions of this function are given in Section IV. Let  $\tilde{U}$  be the solution to the partial differential equation (PDE)

$$(3) \quad \begin{cases} \partial_t \tilde{U}(x, t) = \operatorname{div} \left( C(x) \nabla \tilde{U}(x, t) \right), \\ \tilde{U}(x, 0) = \phi(x), \end{cases}$$

where  $\nabla$  denotes the gradient with respect to  $x = (x_1, x_2)$  and  $\operatorname{div} = \frac{\partial}{\partial x_1} + \frac{\partial}{\partial x_2}$ . The PDE approach to non-homogeneous smoothing amounts to setting  $\tilde{\phi}(x) = \tilde{U}(x, t)$  for a value of  $t$  that determines, together with the choice of the coefficient  $C(\cdot)$ , the extent to which the original data  $\phi$  is regularized. For more details on the relation between PDEs and image processing, we refer e.g. to [5].

## II. SMOOTHING 1D SIGNALS

**II.1. Analyzing signals and detecting edges.** One of the most used techniques in image processing is the use of a discrete differential operator to recover the original signal and capture edges and its main patterns. We propose to use basic discrete differential operators here: the gradient and Laplacian. The matrix representations resulting from a finite difference discretization on a uniform 1D grid in 1D of the gradient ( $\nabla$ ) and Laplacian ( $\Delta$ ) are as follows:

$$\nabla = \begin{pmatrix} -1 & 1 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}, \quad \Delta = \begin{pmatrix} 1 & -2 & 1 & \cdots & 0 & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & -2 & 1 \end{pmatrix}.$$

These operators will be applied in the sequel to the 1D signal  $S(t)$  represented in Figure 1. The corresponding signals  $\nabla S(t)$  and  $\Delta S(t)$  are shown in Figure 2.

The gradient of  $S(t)$  captures the regions where the signal shows large variations, corresponding to edges. The Laplacian also successfully captures these regions. We propose to use the

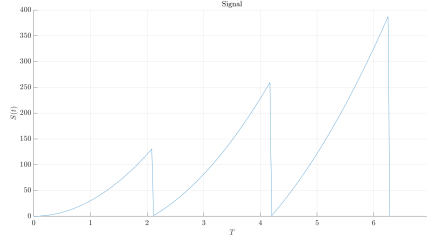


FIGURE 1. Original signal  $S(t)$  in 1D displaying sharp jumps.

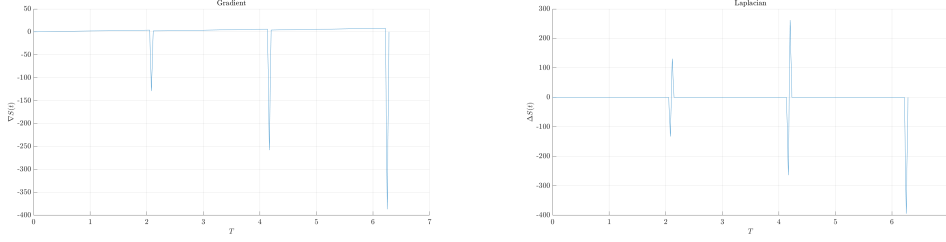


FIGURE 2. Visualization of  $\nabla S(t)$  (left) and  $\Delta S(t)$  (right) in 1D.

gradient of the signal to smooth while conserving the edges. We can see that in the regions where the gradient is large, we want to keep these features and in those regions where the gradient is small, we want to smooth and remove noise present in the signal. We notice that smoothing is inversely related to the variations of the gradient. For this purpose, we try to find a good smoothing function  $G$  establishing such a relationship.

In image processing, non-homogeneous smoothing functions derived from solving a non-homogeneous heat equation is widely used and various well-known possibilities exist for the choice of the function  $G$ . Different functions  $G$  have been tested, giving different results:  $G(t) = \exp(-|\nabla u(t)|^n)$  for  $n = 1, 2, \dots$ . The value  $n = 1$  is used in Figure 3 and for all subsequent results for simplicity.

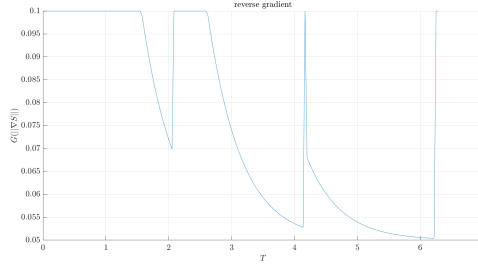


FIGURE 3. Inverse gradient of the signal  $S(t)$  computed by the function  $G(t)$ .

The function  $G(t)$  corresponds to the profile of a smoothing function we want to construct. On the other hand, strong discontinuities are present and this will in turn lead to discontinuities when the signal  $S(t)$  is smoothed. This problem is also present in the 2D study of Section III and can be observed in Figure 8. A solution we propose here to avoid these discontinuities is to smooth the function  $G$  once more using splines, penalized by a minimum curvature constraint with different smoothing weights. We show the results for three different weights in Figure 4.

The function  $G$  in the middle is a good candidate, sufficiently smooth and preserving the same profile of reversed gradient. We choose it as a smoothing function and present below the results.

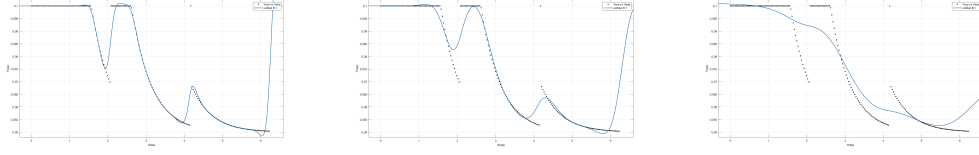
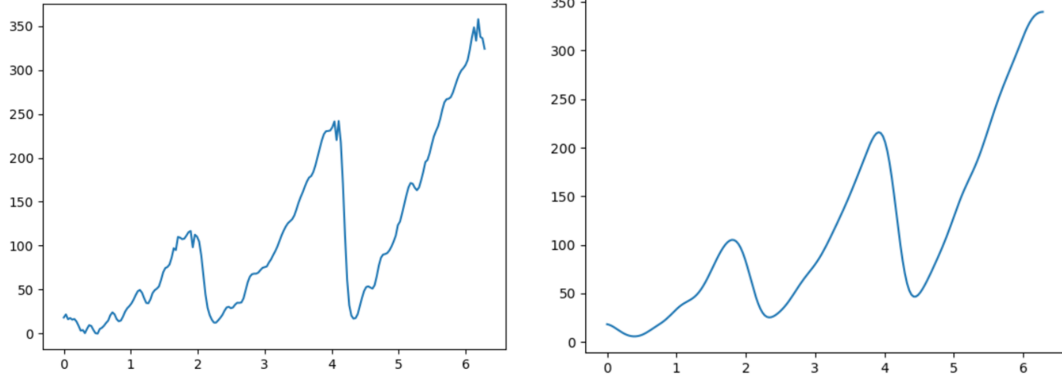
FIGURE 4. Smoothed  $G$  using 3 different weights in 1D.

FIGURE 5. Noisy (left) and smoothed (right) signal in 1D.

We can see in Figure 5 that we remove noise in between the edges of the original signal. The smooth function  $G(t)$  also avoids the large discontinuity jumps in the smoothed signal that would have been obtained without the smoothed variant.

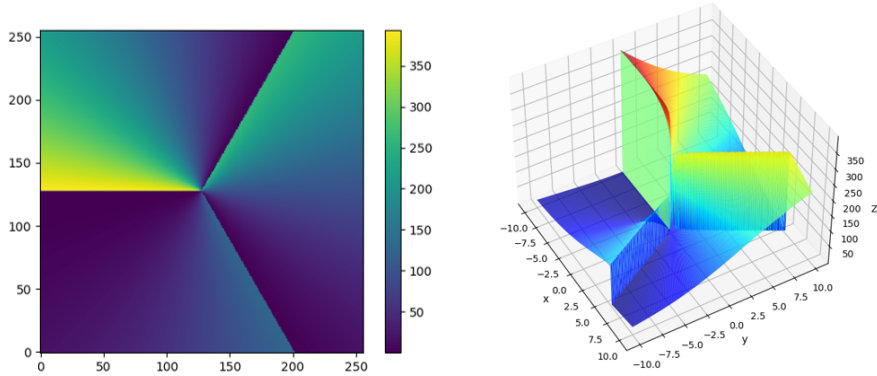
**II.2. Generalization.** Our method gives promising results. The smoothing effect and noise removal is satisfactory. What we developed in 1D is based on really simple linear differential operators discretized on the same structure of the signal, regular grids. Its extension to 2D and 3D problems using finite difference methods on regular grids is straightforward.

### III. NON-HOMOGENEOUS CONVOLUTION IN 2D.

**III.1. Case study description.** Here we shall apply the ideas of Section I.1 to the following initial data proposed by Cailabs:

$$\phi(r, \theta) = S\theta \bmod(M\theta),$$

where  $S = 10$  and  $M = 3$ . The tuple  $(r, \theta)$  represents the polar coordinates of a point in  $\mathbb{R}^2$ . We can visualize this function in Figure 6.

FIGURE 6. Vizualisation of  $\phi$  (2D and 3D).

The function displays jumps each time the product  $M\theta$  equals a multiple of  $2\pi$ . The integer  $M$  is the number of jumps in the image, depending on the angle. The signal could also be polluted by noise, but this is not considered here.

A well-known method to rapidly smooth a function is to convolve it with a Gaussian thanks to Fourier transformations, which was described in Section I.1. This is shown in Figure 7.

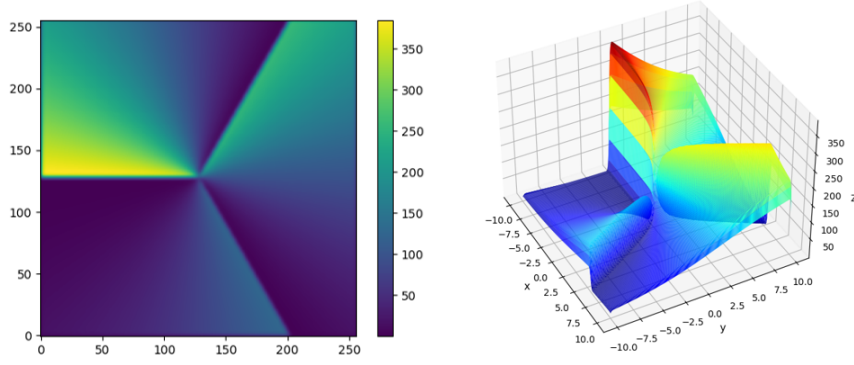


FIGURE 7. Visualization of homogeneously smoothed  $\phi$ .

Specifically, the larger the variance of the Gaussian used, the smoother the phi function becomes after convolution.

The industrial application that we have in mind may contain noise in a non-uniform way and some in some regions it may be important to smooth the signal  $\phi$ , while in other areas this may be less important. For this purpose, Cailabs seeks a method to non homogeneously smooth the signal in a fast way.

### III.2. Non-homogeneous smoothing.

III.2.1. *Idea.* In Section I.1, we have proposed a first approach which consists in convolving  $\phi$  with a Gaussian, the variance of which varies in a piecewise constant manner. To put this idea into practice on the case study of Section III.1, we will :

- (1) Partition the image according to the smoothing we want to perform on it.
- (2) Perform several smoothings of the whole image by convolution with different Gaussians (with high variance if high smoothing desired, and vice versa).
- (3) With each smoothed image obtained, build a heterogeneously smoothed image by recovering the areas we are interested in.

III.2.2. *Application to the case study.* We consider the case where we want to regularize strongly in the center of the domain – more precisely, inside the ball of radius  $r$  around the origin that we denote  $B(0, r)$  – and to obtain only a small smoothing effect outside this ball. To this end, we perform two convolutions: one with a large and one with a small variance, corresponding to a high and a low degree of smoothing, respectively. We recover and join the parts  $B(0, r)$  of the strongly smoothed image and  $B(0, r)^C$  of the less smoothed image. Figure 8 shows the result of this procedure.

We notice that this technique induces a discontinuity at the junction of the two parts of the domain where we want to smooth differently. This discontinuity is a numerical artifact of the smoothing procedure and may be troublesome for industrial applications. In the next section, we discuss some possible solutions to this problem.

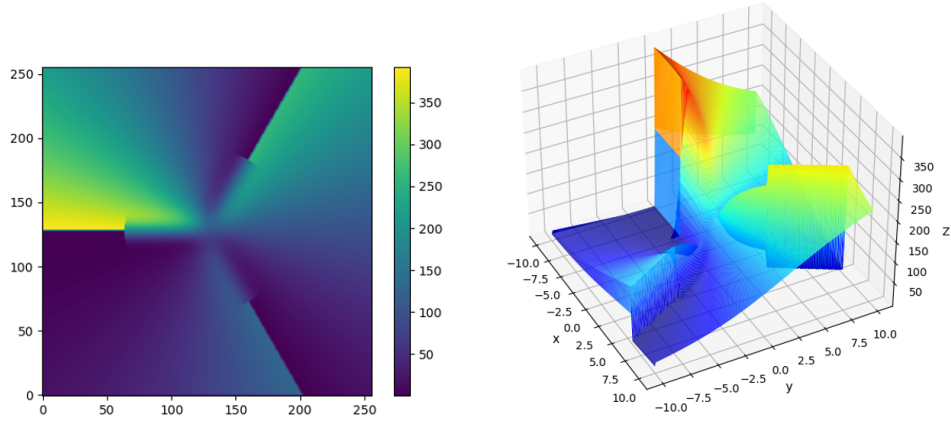


FIGURE 8. Visualization of non homogeneously smoothed  $\phi$  by means of a variance that takes two different values on the ball  $B(0, r)$  and on its complement.

III.2.3. *Some solutions.* Several remedies to the discontinuity induced by the method of Section III.2.2 can be proposed.

**A.** The first one is to use a large number of partitions around the junction and smoothly decrease the variance of the convolution kernel on each partition as one moves away from the center. Instead of having one large jump at the junction of the two regions considered in Section III.2.2, this technique induces multiple jumps that are all small. It results in an almost continuous signal, as shown in Figure 9.

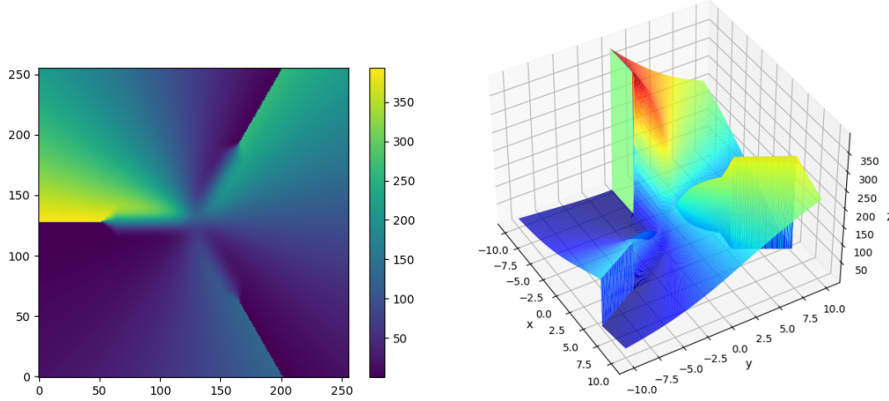


FIGURE 9. Visualization of non homogeneously smoothed  $\phi$  without inducing a strong discontinuity thanks to a smoothly varying variance on several small domains around the junction region.

**B.** A second way to avoid the large jump on the junction of two regions of interest is to add a final fine smoothing on the whole image. This solution allows to smooth this jump without impacting too much the global signal. The result is displayed in Figure 10.

**C.** A last solution we wish to invoke is the use of an interpolation. Around the junction, we can use (for example) a linear interpolation between the two functions that are obtained after smoothing by convolutions kernels with two different variances. This can be done to obtain a continuous signal. In Figure 11 we provide an example in 1D.

#### IV. PDE APPROACH TO NON-HOMOGENEOUS SMOOTHING

In this section, we will consider two concrete cases of the PDE approach to obtain a smoothing of an original signal  $\phi$ . These will be illustrated by numerical examples.



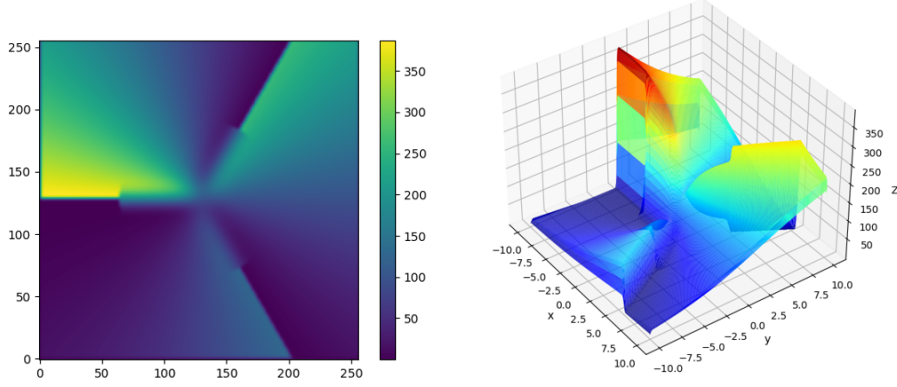


FIGURE 10. Visualization of non-homogeneously smoothed  $\phi$  without inducing a strong discontinuity thanks to fine smoothing.

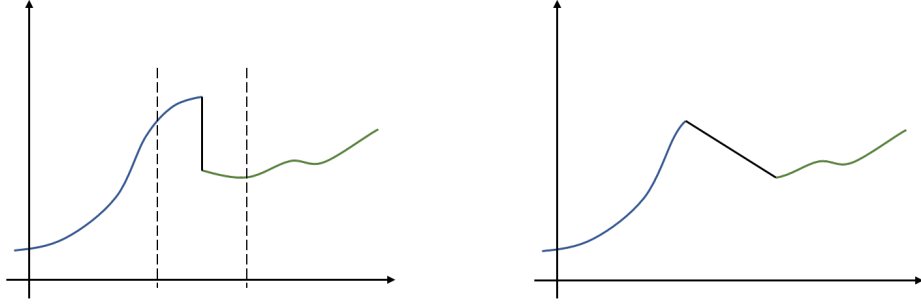


FIGURE 11. Illustration on the removal of a jump by interpolation in 1D.

IV.1. **Case study description.** In this section, we shall focus on a problem with exponential wells that represent relevant features for Cailabs. Let us use Cartesian coordinates  $(x, y)$  to represent any point of  $\mathbb{R}^2$ . The initial signal is given by

$$\phi(x, y) = \exp \left[ - \left( \frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2} \right) \right] + \sum_{y_k \in \{-\frac{1}{2}, -\frac{1}{4}, 0, \frac{1}{4}, \frac{1}{2}\}} \frac{1}{2} \exp \left[ - \frac{x^2 + (y - y_k)^2}{2\sigma_x^2} \right] + w(x, y),$$

with  $\sigma_x = 0.05$  and  $\sigma_y = 0.7$  and where  $w$  represents some noise that is detailed below. The signal is given on the square domain  $[-1, 1]^2$  in  $\mathbb{R}^2$ . We discretize the square as a uniform triangulation that is obtained from dividing the square in  $300 \times 300$  small squares and dividing each small square into two triangles. The noise  $w$  is piecewise constant on the triangular mesh and is drawn according to a uniform distribution on the interval  $[-0.1, 0.1]$ . The entire discretization procedure, as well as the numerical resolution of the problems described below, is carried out with FREEFEM++ [2]. The profile obtained for the signal  $\phi$  is depicted in Figure 12.

IV.2. **Smoothing around the origin.** We consider here a problem with the same motivation as the case study of Section III: one wishes to obtain a smooth signal around the origin, while smoothing may be less pronounced further away from the origin. To do so, we solve (3) with the variable diffusion coefficient  $C$  given by

$$(4) \quad C(x, y) = \exp [-\|(x, y)\|/\delta].$$

The strength of the smoothing effect decays faster as we move away from the origin when the value of the parameter  $\delta$  is smaller.

In order to solve (3), we use an implicit Euler scheme for a single time step  $\Delta t = 10^{-3}$ . This amounts to solving the PDE

$$\tilde{\phi}(x, y) - \Delta t \operatorname{div} \left( C(x, y) \nabla \tilde{\phi}(x, y) \right) = \phi(x, y) \quad \text{in } (-1, 1)^2,$$

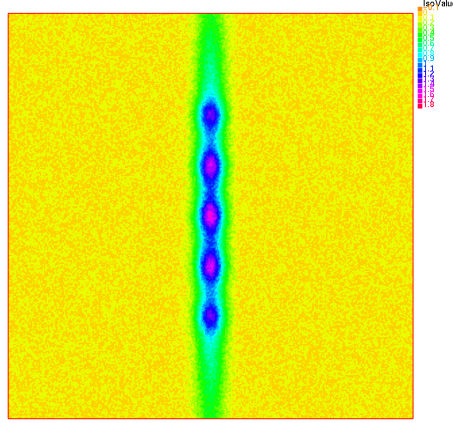


FIGURE 12. Visualization of the exponential wells signal.

for the regularized signal  $\tilde{\phi}$ . For a well-defined problem, we need to equip the PDE with boundary conditions. We use the values of  $\phi$  without noise as boundary conditions. This may not be available in industrial applications, in which case alternative boundary conditions may need to be used that are suitable for the problem at hand. The entire problem is discretized using  $\mathbb{P}_1$  finite elements (see, e.g., [1]).

The results of this smoothing procedure are depicted in Figure 13 for three different values of  $\delta$ . All visualizations are on the same color scale as the one used in Figure 12. It can clearly be observed that the profile of the wells becomes less steep as  $\delta$  increases. Also the smoothing of the noise towards the boundaries of the domain can be seen to be more present for larger values of  $\delta$ . This illustrates how different regions for smoothing can easily be selected thanks to the resolution of the heat equation with a variable diffusion coefficient.

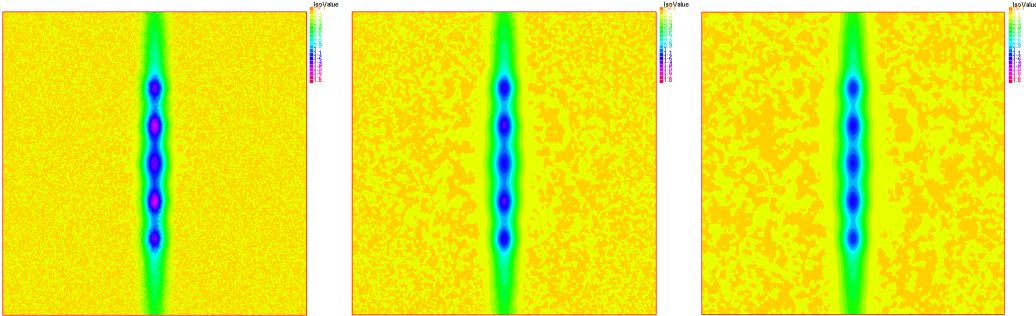


FIGURE 13. Regularization by solving the heat equation for the exponentially decaying coefficient (4). From left to right:  $\delta = 0.05$ ,  $\delta = 0.25$ ,  $\delta = 0.5$ .

We note that the smoothing procedure applied here also leads to the peaks of the exponential wells being lower. When certain characteristics of the original signal are to be preserved for the application at hand, one may apply post-processing steps in order to preserve e.g. the mass or the contrast of the original signal.

**IV.3. Smoothing based on the original signal.** With the PDE approach, it is relatively easy to adapt to regularization of  $\phi$  to the signal  $\phi$  itself. Indeed, one can use a coefficient  $C$  in (3) that depends on  $\phi$ . In order to enrich the exploration of non-homogeneous smoothing techniques, we briefly mention one such example that is used for image processing in the context of edge detection. In this case, one wishes to denoise an initial image while preserving strong gradients, which characterize the contours of the image.

Concretely, for the problem of edge detection, one wishes to reduce the smoothing effect when the gradient of  $\phi$  is large. This is done by choosing a coefficient  $C$  in (3) that decays with

the norm of  $\nabla\phi$ . Two examples used in [3] are

$$C(x, y) = e^{-(\|\nabla\phi\|/K)^2} \quad \text{and} \quad C(x, y) = \frac{1}{1 + (\|\nabla\phi\|/K)^2}.$$

The parameter  $K$  has to be calibrated in view of the application. We refer to [3] for more details.

## CONCLUSION AND PERSPECTIVES

We have recalled two classical regularization approaches: the convolution and the resolution of the heat equation. For both approaches, we have proposed adaptations that might provide a first step for the construction of non-homogeneous smoothing solutions in industry.

The non-homogeneous convolution approach consists in performing multiple convolutions with different variance parameters of the convolution kernel. A naive approach with abrupt variations in the variance leads to artificial discontinuity jumps in the final result. These may be removed by taking a finer partition that allows for a gradual change of the variance, by a second smoothing step on the discontinuous result, or by a simple interpolation procedure. The solution to be preferred depends on the industrial requirements on the final result.

In terms of computational time, the convolutions that are to be performed for each value of the variance can be performed in parallel because they are completely independent from one another. Moreover, since the convolution is (at the level of numerical implementation) a localized operation, each convolution need not be calculated globally, even though this option was not investigated here. This might also be exploited to obtain an efficient numerical algorithm.

A regularization approach based on differential operators was studied in 1D with basic finite difference discretizations. We have seen that the discrete gradient, after regularization, can be used to efficiently remove noise from signals while preserving other important features such as edges. The discrete differential operators based on finite difference methods are easily implemented in higher dimensions.

The PDE approach in 2D allows to define, in a very flexible way, a smoothing effect that varies throughout the domain of interest. In particular, any regularization measure can be defined point-wise, but the smoothing procedure does not require the point-by-point computation of a convolution. Only one global PDE is to be solved to regularize a signal in the entire domain. Efficient PDE solvers exist and may make this approach appealing for industrial applications.

We note that all approaches introduce various parameters to control the degree of smoothing locally. In order to tune these parameters, precise requirements on the final result should be formulated. Then it can be investigated if and how non-homogeneous smoothing approaches can allow to meet these requirements.

## REFERENCES

- [1] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer New York, New York, NY, 2008.
- [2] F. Hecht. New development in FreeFem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.
- [3] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. en. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(7):629–639, 1990.
- [4] SciPy. <https://docs.scipy.org/doc/scipy/>, version 1.8.0.
- [5] J. Weickert. *Anisotropic Diffusion in Image Processing*. ECMI Series. Stuttgart, Germany, 1998.