Pré-Publicações do Departamento de Matemática
Universidade de Coimbra
Preprint Number 19–24

IMPROVED PDE MODELS FOR IMAGE RESTORATION THROUGH BACKPROPAGATION

SÍLVIA BARBEIRO AND DIOGO LOBO

Abstract: In this paper we focus on learning optimized partial differential equation (PDE) models for image filtering. In this approach, the grey-scaled images are represented by a vector field of two real-valued functions and the image restoration problem is modelled by an evolutionary process such that the restored image at any time satisfies an initial-boundary-value problem of cross-diffusion with reaction type. The coupled evolution of the two components of the image is determined by a nondiagonal matrix that depends on those components. A critical question when designing a good-performing filter lies in the selection of the optimal coefficients and influence functions which define the cross-diffusion matrix. We propose the use of deep learning techniques in order to optimize the parameters of the model. In particular, we use a back propagation technique in order to minimize a cost function related to the quality of the denoising processe, while we ensure stability during the learning procedures. Consequently, we obtain improved image restoration models with solid mathematical foundations. The learning framework and resulting models are presented along with related numerical results and image comparisons.

Keywords: Nonlinear cross-diffusion, image denoising, back propagation, learning optimal parameters, stability.

1. Introduction

Nonlinear diffusion processes are well-known and widely used for image noise removal. Roughly speaking, the idea is to combine an effective noise reduction by diffusion with the preservation of the edges and other important image features. Amongst the better known approaches are those where the diffusion coefficient depends on the gradient with an inverse proportion, [14, 17].

The use of nonlinear complex diffusion filters (NCDF) where the image is represented by a complex function and the process of filtering is governed by a diffusion equation with a complex-valued diffusion coefficient, was investigated in [11]. Those filters, which bring the advantage of using the imaginary part of the solution as an edge detector avoiding the computation of the gradient to control the diffusion coefficient, can be successfully applied for denoising in particular for medical imaging despeckling [6, 15].

Received June 28, 2019.

A complex diffusion equation can be written as a cross-diffusion system for the real and imaginary parts. The use of general cross-diffusion systems for image processing, which encompasses the complex-diffusion equations as particular cases, was investigated in [2]. The theoretical studies of the correspondent initial valued boundary problems were presented in [1] and [2] for the linear and nonlinear cases, respectively, where well-posedness and some scale-space representation properties were derived under hypothesis on the cross-diffusion coefficient matrix. However, besides these conditions, there is not much insight on the form that these coefficients should take in practical applications. In [3], the same authors applied the cross-diffusion model to several examples related to the problem of reducing the speckle noise in optical coherence tomography (OCT) images. The conclusion is that the performance of the model is greatly influenced by the choice of the cross-diffusion coefficient matrix.

All those methods based on diffusion filtering are highly dependent on some crucial parameters (*e.g.* [2, 16]). The parameters that lead to the most effective methods vary depending on the acquisition methods, the nature of the images and the associated noise profile. Furthermore, diffusion filters require the specification of a stopping time. To circumvent this issue, a reaction term can be introduced which keeps the steady-state solution close to the original image [17].

The challenge lies in the formulation of robust models with optimal parameters corresponding to each architecture. The use of neural networks for image processing tasks has set the pace of current research in this area. However, the general intrinsic "black box" nature of such algorithms is a drawback, particularly in the context of medical applications. In [9] the authors use a learning framework inspired in nonlinear reaction diffusion processes where the filters and the influence functions are simultaneously learned from training data. The results are auspicious. There are, however, some important questions remaining: the numerical solution is not related with the original reaction diffusion problem, and more importantly, the qualitative properties of the computed solution are not derived. Moreover, the method does not cover promising models as the one based on cross-diffusion.

We propose the use of deep learning techniques such as backpropagation [12] and Adam algorithm [13] to optimize the parameters of a system of PDEs for a specific task. In this work, we focus on nonlinear cross-diffusion with reaction filters in order to optimize the reaction parameter and the influence functions of the cross-diffusion matrix for the removal of gaussian noise.

The optimization process we suggest includes the stability of the inherent numerical method, which is crucial for obtaining successful results. We drive our attention first to synthetic images in order to develop our learning strategy. The methodology consists in the parametrization of the functions that govern the cross-diffusion system, the computation of the gradient of a cost function with respect to these parameters and the application of the backpropagation technique, widely used in neural network supervised learning, to learn the optimal parameters. Experiments in real noisy images show that the training process improves significantly the performance of the filters.

The article is organized as follows. In Section 2 the fully discrete cross-diffusion with reaction model is presented. Next, the stability conditions are derived. Section 4 is devoted to the learning model. We present the numerical experiments in Section 5. We end the paper with a section dedicated to conclusions.

2. Cross-diffusion reaction model

In this section we present a fully discrete cross-diffusion reaction model for image restoration. The image is represented by a two-component vector field, $\mathbf{w} = (u, v)^\top$, and the restoration process is governed by the nonlinear cross-diffusion reaction system

$$\begin{cases} u_t = \nabla \cdot (d_1(\mathbf{w})\nabla u + d_2(\mathbf{w})\nabla v) - \lambda(u - u^0) \text{ in } \Omega \times \mathbb{R}^+, \\ v_t = \nabla \cdot (d_3(\mathbf{w})\nabla u + d_4(\mathbf{w})\nabla v) \text{ in } \Omega \times \mathbb{R}^+, \\ u(\mathbf{x}, 0) = u^0(\mathbf{x}),\ v(\mathbf{x}, 0) = v^0(\mathbf{x}) \text{ in } \Omega, \\ u_\eta = 0,\ v_\eta = 0 \text{ on } \Gamma \times \mathbb{R}^+, \end{cases} \tag{1}$$

where $\Omega = (a_1, b_1) \times (a_2, b_2) \subset \mathbb{R}^2$ is the domain of interest, u^0 and v^0 are the given initial conditions for u and v and η denotes the outward normal vector to the boundary $\Gamma = \partial\Omega$. The cross-diffusion matrix of the model is given by

$$D(\mathbf{w}) = \begin{pmatrix} d_1(\mathbf{w}) & d_2(\mathbf{w}) \\ d_3(\mathbf{w}) & d_4(\mathbf{w}) \end{pmatrix}. \tag{2}$$

λ is a time dependent non-negative parameter.

The domain $\overline{\Omega} = \Omega \cup \Gamma$ is discretized by the points $\mathbf{x_j} = (x_{j_1}, x_{j_2})$, where

$$x_{j_1} = a_1 + h_1 j_1,\ x_{j_2} = a_2 + h_2 j_2, \quad j_k = 0, 1, \ldots, N_k, \quad h_k = \frac{b_k - a_k}{N_k},\ k = 1, 2,$$

for two given integers $N_1, N_2 \geq 1$, $\mathbf{j} = (j_1, j_2)$ and $\mathbf{h} = (h_1, h_2)$. This spatial mesh on $\overline{\Omega}$ is denoted by $\overline{\Omega}_\mathbf{h}$ and $\Gamma_\mathbf{h} = \Gamma \cap \overline{\Omega}_\mathbf{h}$. Points halfway between two adjacent grid points are denoted by $\mathbf{x}_{\mathbf{j}\pm(1/2)\mathbf{e}_k} = \mathbf{x_j} \pm \frac{h_k}{2}\mathbf{e}_k$, $k = 1, 2$, where $\{\mathbf{e}_1, \mathbf{e}_2\}$ is the canonical basis, that is, $\mathbf{e}_k$ is the standard basis unit vector in the kth direction.

For the discretization in time, we consider a mesh with time step Δt, $0 = t^0 < t^1 < t^2 < \ldots$, where $t^{m+1} - t^m = \Delta t$.

We denote by $Z_\mathbf{j}^m$ the value of a mesh function Z at the point $(\mathbf{x_j}, t^m)$. For the formulation of the finite difference approximations, we use the centered finite difference quotients in the kth spatial direction, for $k = 1, 2$,

$$\delta_k Z_\mathbf{j} = \frac{Z_{\mathbf{j}+(1/2)\mathbf{e}_k} - Z_{\mathbf{j}-(1/2)\mathbf{e}_k}}{h_k}, \quad \delta_k Z_{\mathbf{j}+(1/2)\mathbf{e}_k} = \frac{Z_{\mathbf{j}+\mathbf{e}_k} - Z_\mathbf{j}}{h_k}.$$

In order to formulate the discrete cross-diffusion restoration problem, let $u^0 : \Omega_\mathbf{h} \to \mathbb{R}$ be a discrete real-valued function standing for the grey level values on $\Omega_\mathbf{h}$ of the noisy image to be restored. From u^0, an initial distribution $\mathbf{W}^0 = (U^0, V^0)$, for the cross-diffusion, is required. This is given by two real-valued functions $U^0, V^0 : \Omega_\mathbf{h} \to \mathbb{R}$, that can be selected following different criteria. A simple choice, that will be considered in the experiments we present later in this paper, consists of taking $U^0(\mathbf{x_j}) = u^0(\mathbf{x_j})$ and $V^0(\mathbf{x_j}) = 0$, $\mathbf{x_j} \in \overline{\Omega}_\mathbf{h}$. A more detailed discussion on the initial data can be seen in [1, 2].

Let $\mathbf{W}_\mathbf{j}^m = (U_\mathbf{j}^m, V_\mathbf{j}^m)^\top$, such that $\mathbf{x_j} \in \overline{\Omega}_\mathbf{h}$. Given the initial solution $\mathbf{W}_\mathbf{j}^0 = (U_\mathbf{j}^0, V_\mathbf{j}^0)$, the numerical solution of (1) at the time t^{m+1} is obtained considering the following finite difference scheme

$$\begin{aligned} \frac{U_\mathbf{j}^{m+1} - U_\mathbf{j}^m}{\Delta t} &= \sum_{k=1}^{2} \delta_k \left(d_1(\mathbf{W}^m)_\mathbf{j} \delta_k U_\mathbf{j}^{m+\theta} + d_2(\mathbf{W}^m)_\mathbf{j} \delta_k V_\mathbf{j}^{m+\theta} \right) \\ &\quad -\lambda^{m+\theta}(U_\mathbf{j}^{m+\theta} - U_\mathbf{j}^0), \end{aligned} \tag{3}$$

$$\frac{V_\mathbf{j}^{m+1} - V_\mathbf{j}^m}{\Delta t} = \sum_{k=1}^{2} \delta_k \left(d_3(\mathbf{W}^m)_\mathbf{j} \delta_k U_\mathbf{j}^{m+\theta} + d_4(\mathbf{W}^m)_\mathbf{j} \delta_k V_\mathbf{j}^{m+\theta} \right), \tag{4}$$

where

$$D(\mathbf{W}^m)_{\mathbf{j}+(1/2)\mathbf{e}_k} = \frac{D(\mathbf{W}^m_{\mathbf{j}}) + D(\mathbf{W}^m_{\mathbf{j}+\mathbf{e}_k})}{2}$$

and $\theta \in \{0,1\}$. If $\theta = 0$ then (3) is an explicit method. If $\theta = 1$ then method (3) is semi-implicit and its solution is obtained by solving a system of linear equations. In the next section we will derive the stability properties of both, explicit and semi-implicit, methods. Later, in sections 4.2 and 4.3, we will discuss their role in our learning strategy: to optimize the cross-diffusion model we will use the explicit scheme ; to denoise the images we will use the semi-implicit discretization of the optimized model.

Equations (3) and (4), for $\mathbf{j}$ such that $\mathbf{x_j} \in \Gamma_{\mathbf{h}}$, are defined using points of the form $\mathbf{x_j} + \mathbf{e}_k$ and $\mathbf{x_j} - \mathbf{e}_k$, which don't belong to $\Omega_{\mathbf{h}}$ and are not yet defined. For those points $\mathbf{x_j} + \mathbf{e}_k$ and $\mathbf{x_j} - \mathbf{e}_k$, we consider the solution in $\mathbf{x_j} - \mathbf{e}_k$ and $\mathbf{x_j} + \mathbf{e}_k$, respectively. This corresponds to the usual discretization of the homogeneous Neumann boundary conditions on $\Gamma_{\mathbf{h}}$ with central finite differences.

3. Stability of the numerical scheme

We will now investigate the stability of the finite difference scheme (3)-(4). The approach generalizes the strategies in [4] used to study the stability of the particular case of complex-diffusion equations written as a cross-diffusion system.

For each $x_{\mathbf{j}} = (x_{j_1}, x_{j_2}) \in \bar{\Omega}_{\mathbf{h}}$, we define the rectangle $\square_{\mathbf{j}} = (x_{j_1}, x_{j_1+1}) \times (x_{j_2}, x_{j_2+1})$ and denote by $|\square_{\mathbf{j}}|$ the measure of $\square_{\mathbf{j}}$. We consider the discrete L^2 inner products

$$\begin{aligned}(U,V)_h &= \sum_{\square_{\mathbf{j}}\subset\Omega} \frac{|\square_{\mathbf{j}}|}{4} \left(U_{j_1,j_2}V_{j_1,j_2} + U_{j_1+1,j_2}V_{j_1+1,j_2}\right. \\ &\qquad\qquad \left.+U_{j_1,j_2+1}V_{j_1,j_2+1} + U_{j_1+1,j_2+1}V_{j_1+1,j_2+1}\right),\end{aligned}$$

$$(U,V)_{h_1^*} = \sum_{\square_{\mathbf{j}}\subset\Omega} \frac{|\square_{\mathbf{j}}|}{2} \left(U_{j_1+1/2,j_2}V_{j_1+1/2,j_2} + U_{j_1+1/2,j_2+1}V_{j_1+1/2,j_2+1}\right),$$

and

$$(U,V)_{h_2^*} = \sum_{\square_{\mathbf{j}}\subset\Omega} \frac{|\square_{\mathbf{j}}|}{2} \left(U_{j_1,j_2+1/2}V_{j_1,j_2+1/2} + U_{j_1+1,j_2+1/2}V_{j_1+1,j_2+1/2}\right).$$

Their correspondent norms are denoted by $\|.\|_h$, $\|.\|_{h_1^*}$ and $\|.\|_{h_2^*}$, respectively. For $\mathbf{W} = (U, V)^\top$ we define $\|\mathbf{W}\|_h^2 = \|U\|_h^2 + \|V\|_h^2$.

To simplify the notation and where it is clear from the context, we write in what follows d_ℓ instead of $d_\ell(\mathbf{W}^m)$, or $d_\ell(\mathbf{W}^m)_{\mathbf{j}}$, $\ell = 1, 2, 3, 4$.

Multiplying both members of equations (3) and (4) by $U^{m+\theta}$ and $V^{m+\theta}$, respectively, according to the discrete inner product $(\cdot,\cdot)_h$, and using summation by parts we get

$$\begin{aligned}
&\left(\frac{U^{m+1}-U^m}{\Delta t}, U^{m+\theta}\right)_h + \left(\frac{V^{m+1}-V^m}{\Delta t}, V^{m+\theta}\right)_h + \sum_{k=1}^{2}\Big(||(d_1)^{\frac{1}{2}}\delta_k U^{m+\theta}||_{h_k^*}^2 \\
&+ ||(d_4)^{\frac{1}{2}}\delta_k V^{m+\theta}||_{h_k^*}^2 + (d_2\delta_k V^{m+\theta}, \delta_k U^{m+\theta})_{h_k^*} + (d_3\delta_k U^{m+\theta}, \delta_k V^{m+\theta})_{h_k^*}\Big) \\
&= -\lambda^{m+\theta}(U^{m+\theta} - U^0, U^{m+\theta})_h.
\end{aligned}$$

We can write

$$\mathbf{W}^{m+\theta} = \frac{\mathbf{W}^{m+1}+\mathbf{W}^m}{2} + (\theta - \frac{1}{2})\Delta t \frac{\mathbf{W}^{m+1}-\mathbf{W}^m}{\Delta t}$$

and then

$$\begin{aligned}
&\frac{||\mathbf{W}^{m+1}||_h^2 - ||\mathbf{W}^m||_h^2}{2\Delta t} + (\theta-\frac{1}{2})\Delta t\left|\left|\frac{\mathbf{W}^{m+1}-\mathbf{W}^m}{\Delta t}\right|\right|_h^2 \\
&+ \sum_{k=1}^{2}\Big(||(d_1)^{\frac{1}{2}}\delta_k U^{m+\theta}||_{h_k^*}^2 + ||(d_4)^{\frac{1}{2}}\delta_k V^{m+\theta}||_{h_k^*}^2 + (d_2\delta_k V^{m+\theta}, \delta_k U^{m+\theta})_{h_k^*} \qquad (5) \\
&+ (d_3\delta_k U^{m+\theta}, \delta_k V^{m+\theta})_{h_k^*}\Big) \le \lambda^{m+\theta}\|U^{m+\theta} - U^0\|_h\|U^{m+\theta}\|_h.
\end{aligned}$$

For any $\epsilon > 0$ we have that

$$\begin{aligned}
\lambda^{m+\theta}\|U^{m+\theta} - U^0\|_h\|U^{m+\theta}\|_h &\le \lambda^{m+\theta}(\|U^{m+\theta}\|_h + \|U^0\|_h)\|U^{m+\theta}\|_h \\
&\le \lambda^{m+\theta}\|U^{m+\theta}\|_h^2 + \frac{(\lambda^{m+\theta})^2}{4\epsilon}\|U^0\|_h^2 + \epsilon\|U^{m+\theta}\|_h^2. \qquad (6)
\end{aligned}$$

3.1. Stability of the semi-implicit scheme. We start by considering the case where $\theta = 1$. With the assumption that

$$\begin{aligned}\sum_{k=1}^{2}\Big(&||(d_1)^{\frac{1}{2}}\delta_k U^{m+1}||_{h_k^*}^2 + ||(d_4)^{\frac{1}{2}}\delta_k V^{m+1}||_{h_k^*}^2 \\ &+ (d_2\delta_k V^{m+1}, \delta_k U^{m+1})_{h_k^*} + (d_3\delta_k U^{m+1}, \delta_k V^{m+1})_{h_k^*}\Big) > 0,\end{aligned} \tag{7}$$

which corresponds to the natural assumption of the diffusion matrix D to be uniformly positive definite, from (5) and (6) we get

$$(1 - 2\Delta t\lambda^{m+1} - 2\Delta t\epsilon)||\mathbf{W}^{m+1}||_h^2 \le ||\mathbf{W}^m||_h^2 + 2\Delta t\frac{(\lambda^{m+1})^2}{4\epsilon}\|U^0\|_h^2.$$

Assuming that

$$\lambda^{m+1} \le \lambda_{\max}, \quad 0 < \zeta < 1 - 2\Delta t\lambda^{m+1} - 2\Delta t\epsilon, \quad m = 1, 2, \dots \tag{8}$$

for some $\lambda_{\max}, \zeta \in \mathbb{R}^+$ where ϵ is a constant arbitrarily chosen, we get

$$||\mathbf{W}^{m+1}||_h^2 \le (1 + 2\Delta t(\lambda + \epsilon)\zeta^{-1})||\mathbf{W}^m||_h^2 + \Delta t\frac{\lambda_{\max}^2}{2\epsilon\zeta}\|U^0\|_h^2.$$

If (8) holds, using the Duhamel's principle ([8], Lemma 4.1 in Appendix B) we get

$$||\mathbf{W}^{m+1}||_h^2 \le e^{(1+2(\lambda_{\max}+\epsilon)\zeta^{-1})t^{m+1}}\Big(1 + t^{m+1}\frac{\lambda_{\max}^2}{2\epsilon\zeta}\Big)||\mathbf{W}^0||_h^2$$

and we conclude that the method is stable.

3.2. Stability of the explicit scheme. We now consider the case where $\theta = 0$. Applying the $|| \cdot ||_h$ norm on both sides of equations (3) and (4), and using the inequalities $(a \pm b)^2 \le 2a^2 + 2b^2$ and $(a \pm b)^2 \le (1+\eta)a^2 + (1+\eta^{-1})b^2$ for $\eta > 0$, we obtain, respectively,

$$\begin{aligned}\left|\left|\frac{U^{m+1} - U^m}{\Delta t}\right|\right|_h^2 \le &\sum_{k=1}^{2}\frac{8(1+\eta)}{h_k^2}\Big(||d_1\delta_k U^m||_{h_k^*}^2 + ||d_2\delta_k V^m||_{h_k^*}^2 \\ &+ 2(d_1\delta_k U^m, d_2\delta_k V^m)_{h_k^*}\Big) \\ &+ 2(1+\eta^{-1})(\lambda^m)^2(\|U^m\|_h^2 + \|U^0\|_h^2),\end{aligned} \tag{9}$$

for any $\eta > 0$ and

$$\left|\left|\frac{V^{m+1}-V^m}{\Delta t}\right|\right|_h^2 \leq \sum_{k=1}^{2}\frac{8}{h_k^2}\Big(||d_3\delta_k U^m||_{h_k^*}^2 + ||d_4\delta_k V^m||_{h_k^*}^2 + 2(d_3\delta_k U^m, d_4\delta_k V^m)_{h_k^*}\Big). \tag{10}$$

Combining the inequalities (6), (9) and (10) with (5) leads to

$$\begin{aligned}
&\frac{||\mathbf{W}^{m+1}||_h^2 - ||\mathbf{W}^m||_h^2}{2\Delta t} + \sum_{k=1}^{2}\Big(||(d_1)^{\frac{1}{2}}\delta_k U^m||_{h_k^*}^2 + ||(d_4)^{\frac{1}{2}}\delta_k V^m||_{h_k^*}^2\\
&+ (d_2\delta_k V^m, \delta_k U^m)_{h_k^*} + (d_3\delta_k U^m, \delta_k V^m)_{h_k^*}\\
&- \frac{4\Delta t}{h_k^2}\big((1+\eta_1)(||d_1\delta_k U^m||_{h_k^*}^2 + ||d_2\delta_k V^m||_{h_k^*}^2 + 2(d_1\delta_k U^m, d_2\delta_k V^m)_{h_k^*})\\
&+ ||d_3\delta_k U^m||_{h_k^*}^2 + ||d_4\delta_k V^m||_{h_k^*}^2 + 2(d_3\delta_k U^m, d_4\delta_k V^m)_{h_k^*})\Big)\\
&\leq \Big(\lambda^m + \epsilon + \Delta t(1+\eta^{-1})(\lambda^m)^2\Big)\|U^m\|_h^2\\
&+ \Big(\frac{(\lambda^m)^2}{4\epsilon} + \Delta t(1+\eta^{-1})(\lambda^m)^2\Big)\|U^0\|_h^2.
\end{aligned}$$

Let $\eta = \epsilon$. In order to obtain a stable scheme we impose that

$$\begin{aligned}
&\sum_{k=1}^{2}\Big(||(d_1)^{\frac{1}{2}}\delta_k U^m||_{h_k^*}^2 + ||(d_4)^{\frac{1}{2}}\delta_k V^m||_{h_k^*}^2\\
&+ (d_2\delta_k V^m, \delta_k U^m)_{h_k^*} + (d_3\delta_k U^m, \delta_k V^m)_{h_k^*}\\
&- \frac{4\Delta t}{h_k^2}\big((1+\epsilon)(||d_1\delta_k U^m||_{h_k^*}^2 + ||d_2\delta_k V^m||_{h_k^*}^2 + 2(d_1\delta_k U^m, d_2\delta_k V^m)_{h_k^*})\\
&+ ||d_3\delta_k U^m||_{h_k^*}^2 + ||d_4\delta_k V^m||_{h_k^*}^2 + 2(d_3\delta_k U^m, d_4\delta_k V^m)_{h_k^*})\Big) \geq 0,
\end{aligned} \tag{11}$$

for some $\epsilon > 0$.

If (11) holds, then

$$\begin{aligned}
||\mathbf{W}^{m+1}||_h^2 \leq &\|\mathbf{W}^m\|_h^2 + 2\Delta t\Big(\lambda^m + \epsilon + \Delta t(1+\epsilon^{-1})(\lambda^m)^2\Big)\|U^m\|_h^2\\
&+ 2\Delta t\Big(\frac{(\lambda^m)^2}{4\epsilon} + \Delta t(1+\epsilon^{-1})(\lambda^m)^2\Big)\|U^0\|_h^2.
\end{aligned} \tag{12}$$

We now take $a_\epsilon = 2\Big(\lambda^m + \epsilon + \Delta t(1+\epsilon^{-1})(\lambda^m)^2\Big)$ and $b_\epsilon = 2\Big(\frac{(\lambda^m)^2}{4\epsilon} + \Delta t(1 + \epsilon^{-1})(\lambda^m)^2\Big)$. From (12) and using the Duhamel's principle we get

$$||\mathbf{W}^{m+1}||_h^2 \leq e^{a_\epsilon t^{m+1}}\Big(1 + t^{m+1}b_\epsilon\Big)||\mathbf{W}^0||_h^2$$

and we conclude that the method is stable.

Note that we can rewrite (11) as

$$\sum_{k=1}^{2}(\delta_k W)^{m\top} M \delta_k W^m \geq 0, \tag{13}$$

where M is a square matrix of dimension $2N_1N_2 \times 2N_1N_2$. In order (13) to hold we require M to be semi-positive definite, that is, the eigenvalues of M are all non-negative. Using the Gershgorin Theorem, a sufficient condition for the method to be stable is to impose that the influence functions satisfy, for some $\epsilon > 0$,

$$\begin{aligned} d_1 - \frac{4\Delta t}{h_k^2}((1+\epsilon)d_1^2 + d_3^2) &\geq |\frac{1}{2}(d_2+d_3) - \frac{4\Delta t}{h_k^2}((1+\epsilon)d_1d_2 + d_3d_4)|, \\ d_4 - \frac{4\Delta t}{h_k^2}((1+\epsilon)d_2^2 + d_4^2) &\geq |\frac{1}{2}(d_2+d_3) - \frac{4\Delta t}{h_k^2}((1+\epsilon)d_1d_2 + d_3d_4)|. \end{aligned} \tag{14}$$

4. Learning model

Our goal is to optimize the parameters of our model using deep learning techniques. In order to adapt the cross-diffusion scheme (3)-(4) to a neural network architecture, we concatenate U and V into a single column vector w with $2N_1N_2$ entries:

$$w = (u, v)^\top, \quad u = \begin{bmatrix} U_{1,1} \\ U_{1,2} \\ \vdots \\ U_{N_1,N_2} \end{bmatrix}, \quad v = \begin{bmatrix} V_{1,1} \\ V_{1,2} \\ \vdots \\ V_{N_1,N_2} \end{bmatrix}.$$

The m-th iteration of the scheme (3)-(4) can now be written in the vector formulation

$$\begin{aligned}\frac{w^{m+1}-w^m}{\Delta t} \\ &= (K_l^x D_L^{x,m} K_{rL}^x + K_l^y D_L^{y,m} K_{rL}^y + K_l^x D_R^{x,m} K_{rR}^x + K_l^y D_R^{y,m} K_{rR}^y) w^{m+\theta} \\ &\quad -\lambda^{m+\theta}(u^{m+\theta}-u^0), \qquad (15)\end{aligned}$$

with $\lambda = e^\beta$, $\beta \in \mathbb{R}$, where the first difference matrices (K_{r*}^*) are block matrices:

$$K_{rL}^x = \begin{bmatrix} k_r^x & 0 \\ 0 & k_r^x \end{bmatrix}, \quad K_{rR}^x = \begin{bmatrix} 0 & k_r^x \\ k_r^x & 0 \end{bmatrix}, \quad K_{rL}^y = \begin{bmatrix} k_r^y & 0 \\ 0 & k_r^y \end{bmatrix}, \quad K_{rR}^y = \begin{bmatrix} 0 & k_r^y \\ k_r^y & 0 \end{bmatrix},$$

k_r^x and k_r^y denote the backward difference operators with respect to x and y, respectively. The matrices $D_L^{x,m}, D_L^{y,m}, D_R^{x,m}$ and $D_R^{y,m}$ are $2N_1N_2 \times 2N_1N_2$ diagonal matrices. Each entry (j,j) of $D_L^{x,m}$ and $D_L^{y,m}$ depend on the function d_1 for $j = 1, \ldots, N_1N_2$ and on the function d_4 for $j = N_1N_2+1, \ldots, 2N_1N_2$, at the time t^m. The entries of $D_R^{x,m}$ and $D_R^{y,m}$ follow the same pattern, replacing d_1 and d_4 by d_2 e d_3, respectively. It remains to define the difference matrices K_l^x and K_l^y:

$$K_l^x = \begin{bmatrix} k_l^x & 0 \\ 0 & k_l^x \end{bmatrix}, \qquad K_l^y = \begin{bmatrix} k_l^y & 0 \\ 0 & k_l^y \end{bmatrix},$$

where k_l^x and k_l^y denote the forward difference operators with respect to x and y, respectively.

We reiterate that the scheme (15) is equivalent to the cross-diffusion scheme (3)-(4).

4.1. Parameterization of the influence functions. An important aspect of the methodology of our learning strategy consists in the parametrization of arbitrary influence functions. We parameterize the influence functions of the cross-diffusion matrix (2) through standard gaussian radial basis functions. Here we consider that they only depend on the second component of the vector $w = (u,v)^\top$, *i.e.*, they depend on the component which plays the role of edge detector. Each of these functions has the expression

$$d_\ell(v) = \sum_{i=1}^{P} \delta_{\ell,i} \phi\left(\frac{|v-\mu_i|}{2\nu}\right), \quad \phi(z) = e^{-||z||^2}, \qquad (16)$$

where μ_i, for $i = 1, ..., P$, are equidistant points in the set A of edge-detector range of values.

4.2. Constrained optimization. In order to optimize/learn the influence functions of the cross diffusion matrix d_1, d_2, d_3 and d_4 and also the function λ which characterizes the reaction term, a training cost function and a training dataset are required. We will use a set of B gray-scale images which will serve as basis for our training set. We fix the time-step Δt and a number of steps M, which will result in a cross-diffusion stopping time of $T = M\Delta t$. In the learning viewpoint, we will have a M-layered convolutional neural network. We aim to minimize the loss function L,

$$L(\Theta, w_1^m, \ldots, w_B^m, w_1^{nl}, \ldots, w_B^{nl}) = \sum_{i=1}^{B} l(\Theta, w_i^m, w_i^{nl}) = \sum_{i=1}^{B} \frac{1}{2}||u_i^m - u_i^{nl}||^2, \tag{17}$$

with $w_i^m = (u_i^m, v_i^m)^\top$, $w_i^{nl} = (u_i^{nl}, v_i^{nl})^\top$, where w_i^m and w_i^{nl} are, respectively, the m- th iteration of (15) of the i-th corrupted image and the non-corrupted image, respectively, B is the learning batch size, and Θ is the set of parameters we want to optimize. As such, the set of parameters Θ is

$$\Theta = \{\lambda, \delta_{1,1}, \ldots, \delta_{1,P}, \delta_{4,1}, \ldots, \delta_{4,P}, \delta_{2,1}, \ldots, \delta_{2,P}, \delta_{3,1}, \ldots, \delta_{3,P}\},$$

and we must now obtain $\frac{\partial L}{\partial \Theta}(\Theta, w_1^M, \ldots, w_B^M, w_1^{nl}, \ldots, w_B^{nl})$ to solve the minimization problem.

We need to guarantee that the learning procedure ends up with a stable scheme. For that, we will make use of the stability conditions derived in Section 3. Although the use of the explicit method (15) with $\theta = 0$ can bring some advantages in terms of computational effort, the stability conditions are much more restrictive when compared to the semi-implicit method (15) with $\theta = 1$. Moreover, since we need to impose the stability conditions over the values of Θ, the nonlinearities in the stability condition (14) corresponding to the explicit scheme carry out some issues that we want to avoid here. For that reason our goal is to learn a semi-implicit scheme for image denoising.

To obtain a stable semi-implicit scheme, instead of minimizing the loss function (17) we seek the solution of the constrained optimization problem

$$\min_{\Theta} \quad L(\Theta, w_1^M, \ldots, w_B^M, w_1^{nl}, \ldots, w_B^{nl}) \tag{18a}$$

$$\text{s.t.} \quad d_1(x) \geq \tfrac{1}{2}|d_2(x) + d_3(x)|, \quad \forall x \in A, \tag{18b}$$

$$d_4(x) \geq \tfrac{1}{2}|d_2(x) + d_3(x)|, \quad \forall x \in A. \tag{18c}$$

We first note that constraints (18b)-(18c) are not in the standard formulation over the values of Θ. As the radial basis functions (16) are strictly positive, we replace (18b)-(18c) with

$$c_{1,i}(\Theta) = \delta_{1,i} - \tfrac{1}{2}(\delta_{2,i} + \delta_{3,i}) \geq 0, \quad i = 1, \ldots, P, \tag{19a}$$

$$c_{2,i}(\Theta) = \delta_{1,i} + \tfrac{1}{2}(\delta_{2,i} + \delta_{3,i}) \geq 0, \quad i = 1, \ldots, P, \tag{19b}$$

$$c_{3,i}(\Theta) = \delta_{4,i} - \tfrac{1}{2}(\delta_{2,i} + \delta_{3,i}) \geq 0, \quad i = 1, \ldots, P, \tag{19c}$$

$$c_{4,i}(\Theta) = \delta_{4,i} + \tfrac{1}{2}(\delta_{2,i} + \delta_{3,i}) \geq 0, \quad i = 1, \ldots, P, \tag{19d}$$

using the fact that (19a)-(19d) implies (18b)-(18c). To solve the inequality constrained problem we define the augmented Lagrangian $\mathcal{L}$ as

$$\mathcal{L}(\Theta, \mu, \rho) = L(\Theta) + \frac{\rho}{2} \sum_{i=1}^{P} \sum_{\ell=1}^{4} \max\left(0, \frac{\mu_{\ell,i}}{\rho} - c_{\ell,i}(\Theta)\right)^2, \tag{20}$$

which is the usual Powell-Hestenes-Rockafellar function applied to problem (18a) with constraints (19a)-(19d). The vector μ is the set of Lagrange multipliers associated with constraints (19a)-(19d) and ρ is the penalty parameter.

We follow the main model algorithm in [7] with some minor modifications. The learning procedure will iterate through $\mathcal{L}(\Theta_k, \mu_k, \rho_k)$. At the iteration k of the minimization algorithm, the Lagrange multipliers are updated through the formula

$$\mu_{k+1} = \min\{\max\{0, \mu_k - \rho_k c(\Theta_k)\}, \bar{\mu}\} \tag{21}$$

where $\bar{\mu} > 0$ is a multiplier cap, and the penalty parameter is increased or decreased if the infeasibility increases or decreases, respectively. The measure of infeasibility at iteration k is given by the quantity

$$I_k = \min\{c(\Theta_k), \mu_k/\rho_k\}. \tag{22}$$

The penalty parameter is updated in the following way: for some predefined $\tau \in (0, 1)$ and $\gamma > 1$, if $I_k \leq \tau I_{k-1}$ then $\rho_{k+1} = \rho_k/\gamma$, otherwise $\rho_{k+1} = \gamma\rho_k$.

Remark 1. *The stability condition (8) is not very limitative and for that reason we don't include it in the formulation of the optimization problem (18). Even so, we verify if the condition is satisfied by the optimal solution. In our numerical experiments, presented later in the paper, that was always the case.*

4.3. Deriving the gradients. A central step to solve the optimization problem is to compute the gradients of the loss function with respect to the training parameters. We note that the direction of maximum growth of L can be obtained via back-propagation:

$$\frac{\partial l(\Theta, w^M, w^{nl})}{\partial \Theta} = \sum_{m=1}^{M} \frac{\partial l(\Theta, w^M, w^{nl})}{\partial w^M} \frac{\partial w^M}{\partial w^{M-1}} \cdots \frac{\partial w^m}{\partial \Theta}. \tag{23}$$

Although the objective is optimize a semi-implicit cross-diffusion with reaction scheme, we use the explicit scheme (15) with $\theta = 0$ for the learning steps.

We can readily determine

$$\frac{\partial l(\Theta, w^M, w^{nl})}{\partial w^M} = \begin{bmatrix} (u^M - u^{nl})^\top & 0 \end{bmatrix}.$$

We will obtain $\frac{\partial w^m}{\partial \Theta}$ and $\frac{\partial w^{m+1}}{\partial w^m}$ using (15). We start by noticing that

$$\frac{\partial w^m}{\partial \lambda} = -\Delta t \begin{bmatrix} (u^{m-1} - u^0) \\ 0 \end{bmatrix}.$$

We observe that

$$K_l^x D_L^{x,m} K_{rL}^x w^m = K_l^x \mathrm{diag}(K_{rL}^x w^m) g_L^{x,m}$$

where $g_L^{x,m}$ is the vector with the entries of $D_L^{x,m}$. From (16), we can write $g_L^{x,m} = G^{x,m}\Lambda_{1,4}$, where $\Lambda_{1,4}$ is a $2P$ sized vector with the parameters $\delta_{1,i}$, $i = 1, \ldots, P$ and $\delta_{4,i}$, $i = 1, \ldots, P$ of d_1 and d_4, respectively, and

$$G^{x,m} = \frac{1}{2} \begin{bmatrix} S^x \Phi^{x,m} & 0 \\ 0 & S^x \Phi^{x,m} \end{bmatrix},$$

where $S^x = |k_r^x|$ and with $\Phi^{x,m}$ being a $N_1 N_2 \times P$ matrix

$$\Phi^{x,m} = \begin{bmatrix} \phi_1(v_1^m) & \cdots & \phi_P(v_1^m) \\ \vdots & & \vdots \\ \phi_1(v_{N_1N_2}^m) & \cdots & \phi_P(v_{N_1N_2}^m) \end{bmatrix}.$$

Analogously, we write $g_L^{y,m} = G^{y,m}\Lambda_{1,4}$, $g_R^{x,m} = G^{x,m}\Lambda_{2,3}$, $g_R^{y,m} = G^{y,m}\Lambda_{2,3}$, where $S^y = |k_r^y|$, and obtain

$$\frac{\partial w^m}{\partial \Lambda_{1,4}} = \Delta t K_l^x \mathrm{diag}(K_{rL}^x w^{m-1}) G^{x,m-1} + \Delta t K_l^y \mathrm{diag}(K_{rL}^y w^{m-1}) G^{y,m-1},$$

$$\frac{\partial w^m}{\partial \Lambda_{2,3}} = \Delta t K_l^x \mathrm{diag}(K_{rR}^x w^{m-1}) G^{x,m-1} + \Delta t K_l^y \mathrm{diag}(K_{rR}^y w^{m-1}) G^{y,m-1}.$$

Consequently,

$$\frac{\partial w^m}{\partial \Theta} = \begin{bmatrix} \dfrac{\partial w^m}{\partial \lambda} & \dfrac{\partial w^m}{\partial \Lambda_{1,4}} & \dfrac{\partial w^m}{\partial \Lambda_{2,3}} \end{bmatrix}. \tag{24}$$

To derive the expression of $\dfrac{\partial w^{m+1}}{\partial w^m}$, we consider (15). We have

$$\frac{\partial w^{m+1}}{\partial w^m} = I + \left.\frac{\partial w^{m+1}}{\partial w^m}\right]_L^x + \left.\frac{\partial w^{m+1}}{\partial w^m}\right]_R^x + \left.\frac{\partial w^{m+1}}{\partial w^m}\right]_L^y + \left.\frac{\partial w^{m+1}}{\partial w^m}\right]_R^y, \tag{25}$$

with

$$\left.\frac{\partial w^{m+1}}{\partial w^m}\right]_L^x = \Delta t K_l^x D_L^{x,m} K_{rL}^x + \frac{\Delta t}{2} K_l^x \mathrm{diag}(K_{rL}^x w^m) \begin{bmatrix} 0 & S^x d_1'(v^m) \\ 0 & S^x d_4'(v^m) \end{bmatrix} \Bigg),$$

$$\left.\frac{\partial w^{m+1}}{\partial w^m}\right]_R^x = \Delta t K_l^x D_R^{x,m} K_{rR}^x + \frac{\Delta t}{2} K_l^x \mathrm{diag}(K_{rR}^x w^m) \begin{bmatrix} 0 & S^x d_2'(v^m) \\ 0 & S^x d_3'(v^m) \end{bmatrix} \Bigg),$$

$$\left.\frac{\partial w^{m+1}}{\partial w^m}\right]_L^y = \Delta t K_l^y D_L^{y,m} K_{rL}^y + \frac{\Delta t}{2} K_l^y \mathrm{diag}(K_{rL}^y w^m) \begin{bmatrix} 0 & S^y d_1'(v^m) \\ 0 & S^y d_4'(v^m) \end{bmatrix} \Bigg),$$

$$\left.\frac{\partial w^{m+1}}{\partial w^m}\right]_R^y = \Delta t K_l^y D_R^{y,m} K_{rR}^y + \frac{\Delta t}{2} K_l^y \mathrm{diag}(K_{rR}^y w^m) \begin{bmatrix} 0 & S^x d_2'(v^m) \\ 0 & S^x d_3'(v^m) \end{bmatrix} \Bigg),$$

where $d_\ell'(v^m)$ is a vector of dimension $N_1 N_2$ with entries $[d_\ell'(v^m)]_j = d_\ell'(v_j^m)$, being v_j^m is the second component of the vector $w_j^m = (u_j^m, v_j^m)^\top$. We obtain these derivatives numerically with a centered difference scheme.

4.4. Back-propagation algorithm. We now describe how the unknown parameters are learned via back-propagation. We store B gray-scale images to form the basis for our training set. In each iteration, we crop a random $N_1 \times N_2$ section of each image and add the desired synthetic random noise to each section (for example noise of Gaussian type with zero mean and standard deviation σ. This set of freshly cut sections with noise will make up the batch for the iteration.

Algorithm 1 Algorithm for NCDF learning

```
Input: $B$ gray-scale images, $\sigma > 0$ (s.d. of gaussian noise);
Input: $\triangle t > 0$, $(N_1, N_2, M) \in \mathbb{N}^3$, $\Theta_1 \in \mathbb{R}^{4P+1}$;
Input: $K_{\max} > 0$, $\bar{\mu} > 0$, $\mu_1 \in [0, \bar{\mu}]^{4P}$, $\rho_1 > 0$, $\tau \in ]0, 1]$, $\gamma > 1$;
Input: Initialization of Adam parameters.
Set $k = 1$;
while $k \leq K_{\max}$ do
    Extract $B$ random $N_1 \times N_2$ sections from the training set (batch);
    Add random noise to the batch;
    Compute $\frac{\partial \mathcal{L}}{\partial \Theta_k}$ through backpropagation;
    Update $\Theta_k$ via Adam algorithm as in [13];
    $\mu_{k+1} = \min\{\max\{0, \mu_k - \rho_k c(\Theta_k)\}, \bar{\mu}\}$;
    if $I_k \leq \tau I_{k-1}$ then
        $\rho_{k+1} = \rho_k / \gamma$;
    else
        $\rho_{k+1} = \gamma \rho_k$;
    end if
    $k = k + 1$;
end while
Output: $\Theta_{K_{\max}}$
```

There is also the need to fix the step time Δt and the number of layers M before solving the minimization problem. This is a crucial choice, as $T = M\Delta t$ becomes the diffusion time for which the procedure will be optimized. The functions d_1, d_2, d_3 and d_4 are initialized as in the nonlinear complex diffusion case [2], while λ is initialized with the value that provides the best average peak to signal noise ratio (PSNR) for the nonlinear complex diffusion process over the training set.

Given these particular considerations, we present the learning algorithm model (Algorithm 1), which is the Adam algorithm [13] applied to our particular problem.

5. Experimental results

Having previously established a learning framework for the optimization of a cross-diffusion process, we now try it out with some examples. We set $B = 50$ and store 50 gray-scaled images (Figure 1) that will serve as training set. The crop size for the batch extraction was fixed for all tests as $N_1 = N_2 = 100$ and the influence functions were initialized as in the

Figure 1. Images used as training set for the numerical experiments.

Figure 2. Images used as test set for the numerical experiments.

non-linear complex diffusion case [4, 5, 11],

$$d_1(x) = 0.99g(x),\ d_2(x) = -0.1g(x),\ d_3(x) = 0.1g(x),\ d_4(x) = 0.99g(x),$$

	$\triangle t$	M	T		$\triangle t$	M	T
	0.05	10	0.5		0.1	10	1
	0.05	15	0.75		0.1	15	1.5
$\sigma = 10$	0.1	10	1	$\sigma = 20$	0.1	20	2
	0.125	10	1.25		0.125	20	2.5
	0.125	12	1.5		0.2	30	3

TABLE 1. Time-step $\triangle t$ and number of iterations M fixed for the numerical experiments. Different choices lead to different stopping times.

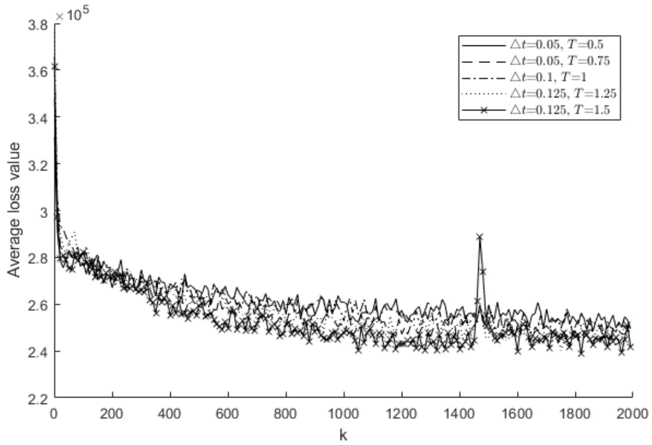


FIGURE 3. Loss value (17) in training for gaussian denoising with $\sigma = 10$, $K_{\max} = 2000$ and different stopping times. An initial slope gives rise to a steady decline that seems to converge to an asymptotic value shared by all stopping times.

with

$$g(x) = \frac{1}{1 + x^2}.$$

We experimented with different values of $\triangle t$ and T to see the effect of changing stopping times in this learning framework. The values chosen are summarized in (Table 1). For the RBF interpolation (16), we chose $A = [-20, 20]$, $P = 151$ and $\nu = 0.2$. For the augmented Lagrangian parameters, we fixed for all tests $\bar{\mu} = 2$, $\rho_1 = 6 \times 10^5$, $\tau = 0.5$ and $\gamma = 2$. Finally, we set $K_{\max} = 2000$ for the number of iterations of the minimization problem.

In Figure 3 we average the values of (17) over groups of 10 algorithm updates for all the training procedures carried out with $\sigma = 10$. The results

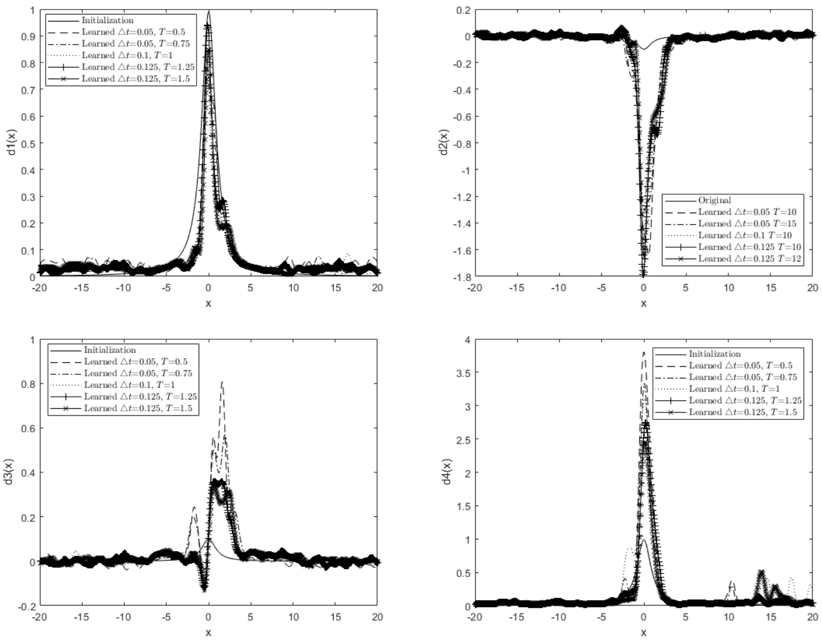


FIGURE 4. Influence functions learning results for gaussian denoising with $\sigma = 10$ and different stopping times. From left to right, top to bottom: d_1, d_2, d_3 and d_4

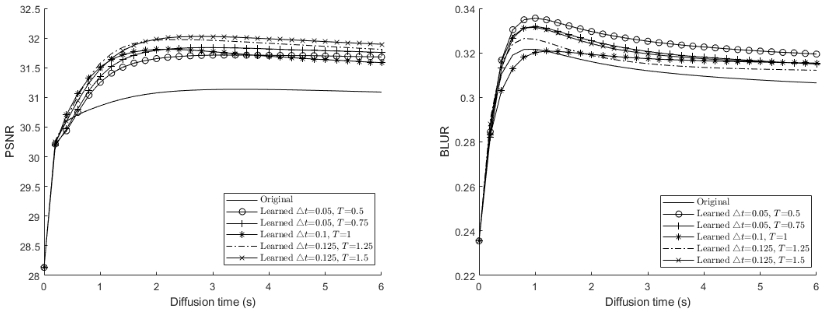
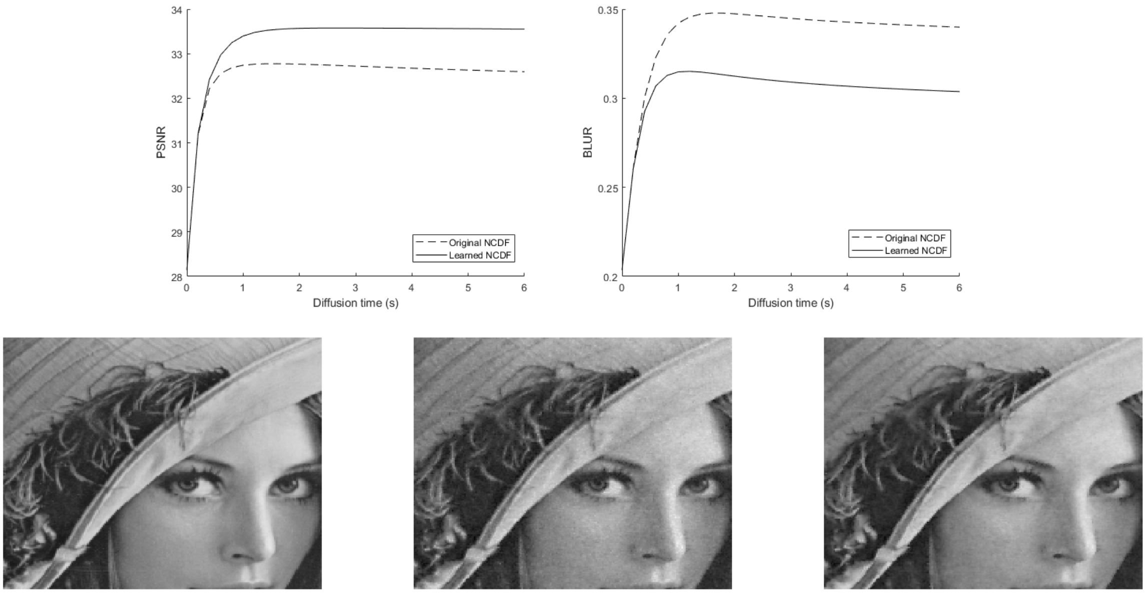


FIGURE 5. Average of the values of PSNR (left) and Blur (right) obtained over the test set of the learning results for gaussian denoising with $\sigma = 10$ and different stopping times. We obtain a considerable improvement in PSNR through training and no significant alterations in Blur.

show a similar pattern for all combination of $\triangle t$ and M. The algorithm reaches a significant lower loss value relatively fast (in 50 to 100 updates) and proceeds to improve (although in a slower pace) throughout the rest of the optimization procedure.



Original NCDF Learned

FIGURE 6. Learned NCDF results for gaussian denoising with $\sigma = 20$. Top left: PSNR values of Lena denoising for original and learned NCDF. Top right: Blur values of Lena denoising for original and learned NCDF. Bottom: Lena detail of original image (left), three seconds diffusion via original NCDF (middle) and three seconds diffusion via learned NCDF (right). Considerable improvements in both metrics.

We describe the learned influence functions in figures 4 and 7 for $\sigma = 10$ and $\sigma = 20$, respectively. We notice that the learning translates into the determination of the appropriate scale of the original function in the cases of influence functions d_1, d_2 and d_4, and into the radical reshaping of d_3. The results are consistent across the different stopping times and both levels of noise.

In figures 5 and 8 we test the learned parameters against the non-linear complex diffusion case and with the value of λ that provides the best PSNR in the training set. The comparisons are performed in the test set (figure 2), which is made by images that were not integrated in the training set. We see in both cases a significant increase in PSNR and a slight increase in blur ([10]). The combinations of $\triangle t$ and M that achieve better performances both for $\sigma = 10$ and $\sigma = 20$ are those that combine into a greater value of T, that is, a later stopping time. The reason for this is the introduction of the

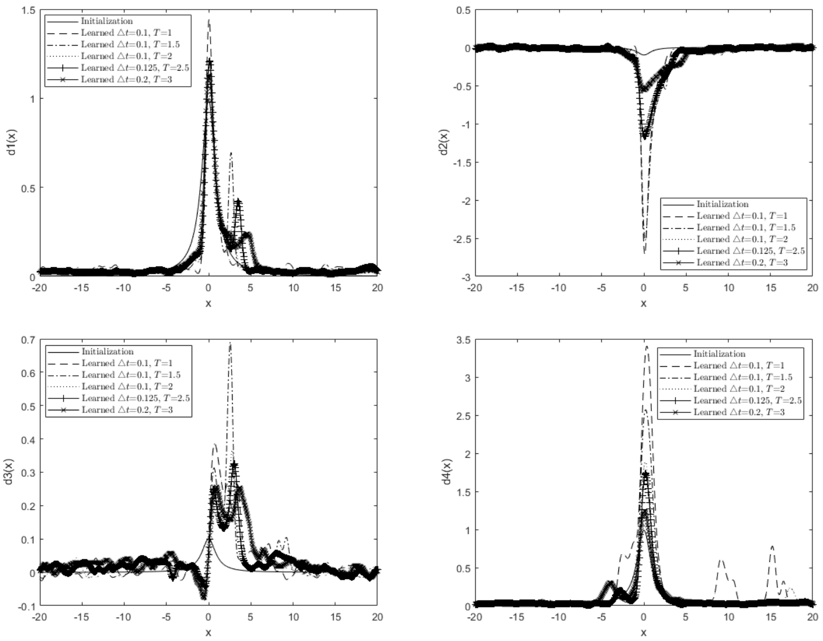


Figure 7. Influence functions learning results for gaussian denoising with $\sigma = 20$ and different stopping times. From left to right, top to bottom: d_1, d_2, d_3 and d_4

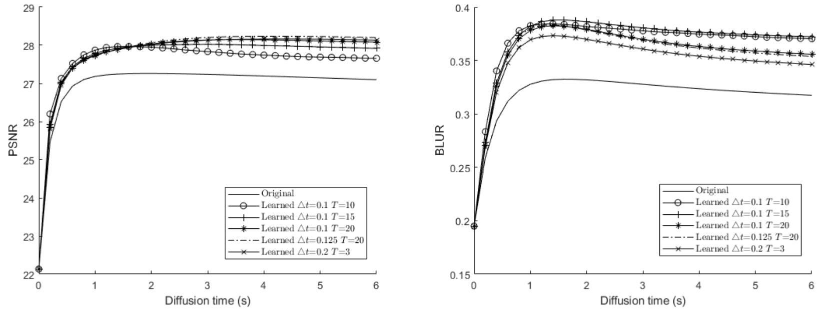
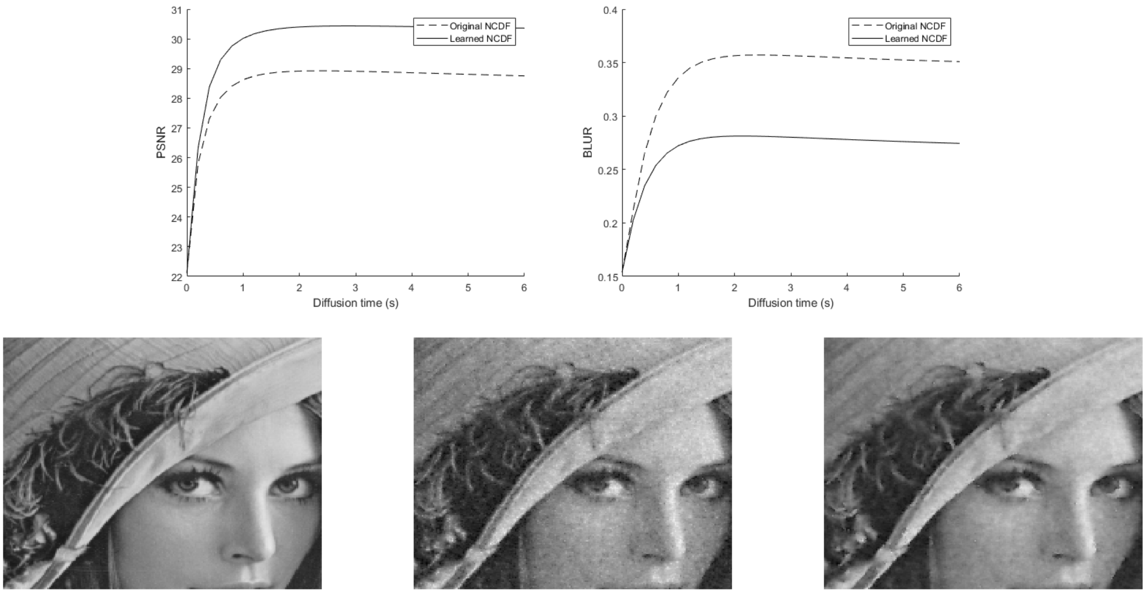


Figure 8. Average of the values of PSNR (left) and Blur (right) obtained over the test set of the learning results for gaussian denoising with $\sigma = 20$ and different stopping times. We obtain an improvement in PSNR through training and a slight increase in Blur.

reaction term, which shifts the steady-state solution into a non-trivial one (when compared to diffusion filters without such reaction), and consequently the denoising process benefits from a longer and more controlled diffusion.



Original NCDF Learned

FIGURE 9. Learned NCDF results for gaussian denoising with $\sigma = 20$. Top left: PSNR values of Lena denoising for original and learned NCDF. Top right: Blur values of Lena denoising for original and learned NCDF. Bottom: Lena detail of original image (left), three seconds diffusion via original NCDF (middle) and three seconds diffusion via learned NCDF (right). Considerable improvements in both metrics.

Finally, we compare the action of the nonlinear complex filter against the best learned parameters (that is, the ones trained with a larger T) on the widely used Lena image. The results are in figures 6 and 9 for $\sigma = 10$ and $\sigma = 20$, respectively. In both cases, we achieved significantly improved PSNR and blur values.

6. Conclusions

We have successfully adapted a cross-diffusion model for image restoration into a learning framework, obtaining a way to automatically parametrize the cross-diffusion matrix for different images and levels of noise. By making the parallelism between neural networks and the parametrization of PDEs, we believe that this work can be transferred to a broad scope of related problems.

References

[1] A. Araújo, S. Barbeiro, E. Cuesta, and A. Durán. Cross-diffusion systems for image processing: I. the linear case. *Journal of Mathematical Imaging and Vision*, 58:447–467, 2017.

[2] A. Araújo, S. Barbeiro, E. Cuesta, and A. Durán. Cross-diffusion systems for image processing: Ii. the nonlinear case. *Journal of Mathematical Imaging and Vision*, 58(3):427–446, Jul 2017.

[3] A. Araújo, S. Barbeiro, E. Cuesta, and A. Durán. A discrete cross-diffusion model for image restoration. In P. Quintela, P. Barral, D. Gómez, F. J. Pena, J. Rodríguez, P. Salgado, and M. E. Vázquez-Méndez, editors, *Progress in Industrial Mathematics at ECMI 2016*, pages 401–408, Cham, 2017. Springer International Publishing.

[4] A. Araújo, S. Barbeiro, and P. Serranho. Stability of finite difference schemes for complex diffusion processes. *SIAM Journal on Numerical Analysis*, 50(3):1284–1296, 2012.

[5] A. Araújo, S. Barbeiro, and P. Serranho. Stability of finite difference schemes for nonlinear complex reaction–diffusion processes. *IMA Journal of Numerical Analysis*, 35:1381–1401, 01 2015.

[6] R. Bernardes, C. Maduro, P. Serranho, A. Araújo, S. Barbeiro, and J. Cunha-Vaz. Improved adaptive complex diffusion despeckling filter. *Opt. Express*, 18(23):24048–24059, Nov 2010.

[7] Ernesto G. Birgin and J. M. Martínez. *Practical augmented Lagrangian methodsPractical Augmented Lagrangian Methods*, pages 3013–3023. Springer US, Boston, MA, 2009.

[8] Tony F. Chan and Longjun Shen. Stability analysis of difference schemes for variable coefficient schrodinger type equations. *SIAM Journal on Numerical Analysis*, 24(2):336–349, 1987.

[9] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1256–1272, June 2017.

[10] Frédérique Crété-Roffet, Thierry Dolmière, Patricia Ladret, and Marina Nicolas. The blur effect: Perception and estimation with a new no-reference perceptual blur metric. *Human Vision and Electronic Imaging*, 12, 03 2007.

[11] G. Gilboa, N. Sochen, and Y. Y. Zeevi. Image enhancement and denoising by complex diffusion processes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(8):1020–1036, August 2004.

[12] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* The MIT Press, 2016.

[13] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[14] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(7):629–639, July 1990.

[15] H. M. Salinas and D. C. Fernandez. Comparison of pde-based nonlinear diffusion approaches for image enhancement and denoising in optical coherence tomography. *IEEE Transactions on Medical Imaging*, 26(6):761–771, June 2007.

[16] C. Tsiotsios and M. Petrou. On the choice of the parameters for anisotropic diffusion in image processing. *Pattern Recognition*, 46:1369–1381, 05 2013.

[17] J. Weickert. A review of nonlinear diffusion filtering. In Bart ter Haar Romeny, Luc Florack, Jan Koenderink, and Max Viergever, editors, *Scale-Space Theory in Computer Vision*, pages 1–28, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

Acknowledgements

This work was partially supported by the Centre for Mathematics of the University of Coimbra – UID/MAT/00324/2019, funded by the Portuguese Government through FCT/MEC and co-funded by the European Regional Development Fund through the Partnership Agreement PT2020. The second author was supported by the FCT grant PD/BD/142956/2018.

Sílvia Barbeiro
CMUC, Department of Mathematics, University of Coimbra, Apartado 3008, EC Santa Cruz, 3001 - 501 Coimbra, Portugal
E-mail address: silvia@mat.uc.pt
URL: http://www.mat.uc.pt/~silvia

Diogo Lobo
CMUC, Department of Mathematics, University of Coimbra, Apartado 3008, EC Santa Cruz, 3001 - 501 Coimbra, Portugal
E-mail address: diogo.lobo@mat.uc.pt