International Journal of Differential Equations and Applications
Volume 16 No. 1 2017, 93-122
ISSN: 1311-2872
url: http://www.ijpam.eu
doi: http://dx.doi.org/10.12732/ijdea.v16i1.4707
acadpubl.eu

IMPLICIT METHOD FOR NONLINEAR COMPLEX DIFFUSION WITH APPLICATIONS TO IMAGE DENOISING

Marlon Oliveira[1] §, Pedro Serranho[2]
[1]School of Computing
Dublin City University
Glasnevin, Dublin 9, IRELAND
[1,2]Department of Sciences and Technology
Universidade Aberta, Rua da Escola Politécnica 141-147
Lisboa, 1269-001, PORTUGAL

Abstract: In this paper we focus on the development and implementation of an implicit finite difference method for solving a complex diffusion differential equation with applications to noise filtering in images. We give a complete characterization of the implementation of the fully implicit method, including the characterization of the derivatives for Newton's method at each step. In order to show the feasibility of the method, we compare the implicit method with the semi-implicit and apply it to both *in vivo* images of the human retina by optical coherence tomography (OCT) and standard images (Lena and the Cameraman). We compute noise removal performance metrics to show the effect of several parameters in the performance of the method.

AMS Subject Classification: 65M06, 68U10
Key Words: denoising, despeckling, complex diffusion, finite difference methods, implicit method

1. Introduction

Complex diffusion is used to characterize several processes in science. They

Received: May 3, 2017

§Correspondence author

are usually governed by a partial differential equation, the Schödinger equation being probably one of the best known models. However, complex diffusion has several other applications, including image processing. Besides examples in inpainting, denoising and stereo vision [5, 8, 19, 20, 21], we are particularly interested in applications to image denoising and despeckling [4, 7, 11, 15], especially in the case where complex diffusion is used [4, 7, 14, 10]. Complex diffusion has been used in noise filtering in specific cases of medical imaging, such as optical coherence tomography (OCT), showing very good results [4, 9, 13, 12, 14].

The methods for diffusion processes in the image filtering context are usually nonlinear and based on the discretization of a nonlinear diffusion equation of the form

$$\frac{\partial u}{\partial t}(x,t) = \nabla \cdot (D(x,t,u)\nabla u(x,t)), \quad \text{in } Q \times [0,T], \tag{1}$$

where the solution $u(x,t)$ represents different stages of the filtered image at time t, x is the spatial coordinate defined in the square $Q = [1, N_1] \times [1, N_2]$ (the initial image has dimensions $N_1 \times N_2$), t is the time coordinate defined in the interval $[0,T]$ where T is the maximum diffusion time, and the diffusion coefficient D has to be properly defined in order to avoid blurring effects, e.g. diffusion across intensity edges in the image. [11] proposed to use a diffusion coefficient based on the gradient of the image, in order to distinguish between edges and constant regions. However, in the initial steps of the image where the noise level is high, the gradient is unstable. To overcome this problem, [7] suggested to consider an appropriate complex filter of the form

$$D = \frac{e^{i\vartheta}}{1 + \left(\frac{\Im(u)}{\kappa\vartheta}\right)^2}, \tag{2}$$

where $\vartheta \approx 0$ and κ is a positive coefficient. It was proven that this filter is appropriate, since

$$\lim_{\vartheta \to 0} \frac{\Im(u(.,t))}{t\vartheta} = G * \Delta I_0,$$

where I_0 is the initial image, G is a Gaussian and $*$ is the convolution operator. Therefore, with the evolution of time t the processed image $u(.,y)$ tends to a smoother version of the initial image. In [1, 2] rigorous stability results were proven for finite difference methods for Equation (1) with a complex diffusion coefficient. The convergence result was achieved in [3] as well.

Though implicit and semi-implicit finite difference methods are shown to be unconditionally stable, researchers tend to implement and use explicit methods

for the diffusion process. The explicit method is easier to implement, but the time step is limited by the stability condition, leading to the need for a higher number of time steps. However, since in the initial steps the image is not smooth (due to noise) it is not clear if a large step in time is advisable anyway, stressing the common use of explicit methods in the literature. On the other hand, implicit and semi-implicit methods need to solve a linear system at every time step, though the number of time steps can be considerably lower, since there are no stability constraints [1, 2].

In this work we will characterize the implicit method for the given nonlinear complex diffusion process (1) with diffusion coefficient (2), using Newton's method for its linearization. Moreover, we will numerically compare the performance of this approach with the performance of the semi-implicit method both for standard and OCT images in terms of image metrics, complementing the work in [14] where the explicit and semi-implicit methods were compared. In addition, for the semi-implicit case, we will consider both Dirichlet and Neumann boundary conditions, in order to illustrate if the boundary condition has an effect on the denoising of the image. We will illustrate how several parameters of the process (1)-(2) and the numerical discretization influence the filtering results as well. In this way we will illustrate whether the use of implicit methods is an advantage for image noise filtering from a performance and computational time perspective.

2. Mathematical Formulation

Let I_0 be the original (noisy) image of size $N_1 \times N_2$, which will define the domain $Q = [1, N_1] \times [1, N_2]$ for the complex diffusion Equation (1). In order to have a well posed problem, Equation (1) must be complemented with the initial condition of the form

$$u(x, 0) = I_0(x), \quad x \in Q, \tag{3}$$

defined by the original (noisy) image. Moreover, one needs boundary conditions, defined on the boundary Γ of the set Q. We will consider Neumann boundary conditions

$$\frac{\partial u}{\partial \nu}(x, t) = 0, \quad x \in \Gamma,\ t \in [0, T], \tag{4}$$

which means there is no normal intensity flux, where ν is the unit exterior normal vector.

For Dirichlet boundary conditions we consider an artificial domain given by $Q_0 = [0, N_1 + 1] \times [0, N_2 + 1]$ with boundary Γ_0 where the

$$u(x,t) = g(x,t), \quad x \in \Gamma_0,\, t \in [0,T], \tag{5}$$

which means that the boundaries of the image are defined by some a priori known function g. In image processing, the value of g is independent of time and defined in Γ_0 as the intensity of the original image in the boundary Γ. This is due to the fact that the original image is usually affected by noise and this artificial boundary allows to eliminate noise on the real boundary Γ.

3. Discretization

We consider an equally spaced mesh on Q_0, both for the Dirichlet and Neumann case. Since our context is image processing, we let $h_1 = h_2 = 1$ be the mesh intervals in the first and second spatial coordinate directions. The mesh is therefore defined by the set of points $x_{j,k} = (j,k), \quad j = 0, 1, 2, \ldots, N_1 + 1$, and $k = 0, 1, 2, \ldots, N_2 + 1$. Moreover, we consider the time step $h_{t,m}$ which defines the set of points

$$t_{m+1} = t_m + h_{t,m}, \quad m = 0, 1, \ldots, N_t.$$

such that $t_{N_t} = T$. Therefore we define a mesh Q_h defined by the set of points

$$(x_{j,k}, t_m), \quad j = 0, 1, 2, \ldots, N_1 + 1,\, k = 0, 1, 2, \ldots,$$

$$N_2 + 1,\, m = 0, 1, \ldots, N_t.$$

We consider a general finite difference method [1] for the differential Equation (1) given by

$$\begin{aligned} U_{j,k}^{m+1} =& U_{j,k}^{m} + \frac{h_{t,m}}{2}\Big[\left(D_{(j+1,k)}^{m,\theta,\mu} + D_{(j,k)}^{m,\theta,\mu} \right) U_{(j+1,k)}^{m+\theta} \\ &+ \left(D_{(j,k)}^{m,\theta,\mu} + D_{(j-1,k)}^{m,\theta,\mu} \right) U_{(j-1,k)}^{m+\theta} \\ &- \left(D_{(j+1,k)}^{m,\theta,\mu} + 2D_{(j,k)}^{m,\theta,\mu} + D_{(j-1,k)}^{m,\theta,\mu} \right) U_{(j,k)}^{m+\theta} \Big] \\ &+ \frac{h_{t,m}}{2}\Big[\left(D_{(j,k+1)}^{m,\theta,\mu} + D_{(j,k)}^{m,\theta,\mu} \right) U_{(j,k+1)}^{m+\theta} \\ &+ \left(D_{(j,k)}^{m,\theta,\mu} + D_{(j,k-1)}^{m,\theta,\mu} \right) U_{(j,k-1)}^{m+\theta} \\ &- \left(D_{(j,k+1)}^{m,\theta,\mu} + 2D_{(j,k)}^{m,\theta,\mu} + D_{(j,k-1)}^{m,\theta,\mu} \right) U_{(j,k)}^{m+\theta} \Big], \end{aligned} \tag{6}$$

for $j = 1, 2, \ldots, N_1$, $k = 1, 2, \ldots, N_2$, $m = 0, 1, \ldots, N_t - 1$ where $U_{j,k}^m$ is the approximation to the solution $u(x_{j,k}, t_m)$,

$$D_{(j,k)}^{m,\theta,\mu} = D\left(x_{j,k}, t^{m+\theta}, U_{j,k}^{m+\theta\mu}\right),$$

and $\mu \in \{0, 1\}$, $t^{m+\theta} = \theta t^{m+1} + (1-\theta)t^m$, $\theta \in [0, 1]$ and

$$V_{j,k}^{m+\theta} = \theta V_{j,k}^{m+1} + (1-\theta)V_{j,k}^m,$$

for $V = U, D$. We note that the semi-implicit method is defined by $\theta = 1$ and $\mu = 0$ and the fully implicit method is defined by $\mu = \theta = 1$.

3.1. Semi-Implicit Method

The semi-implicit method is known to be unconditionally stable [1], so one can choose a fixed step in time $h_t = T/N_t$. For the semi-implicit method, a linear system needs to be solved at each iteration, since the diffusion term $D_{j,k}^m := D\left(x_{j,k}, t^m\right)$ does not depended on the unknowns $U_{j,k}^{m+1}$ at each time step m.

Therefore, along with the initial condition

$$U_{j,k}^0 = I_0(j,k), \quad j = 1, 2, \ldots, N_1,\ k = 1, 2, \ldots, N_2. \tag{7}$$

and the Dirichlet boundary condition given

$$\begin{aligned} U_{j,0}^m = I_0(j,1),\ U_{j,N_2+1}^m = I_0(j,N_1), &\quad j = 1, 2, \ldots, N_1, \\ U_{0,k}^m = I_0(1,k),\ U_{N_1+1,k}^m = I_0(N_1,k), &\quad k = 1, 2, \ldots, N_2, \end{aligned} \tag{8}$$

for $m = 0, 1, \ldots, N_t$, or the Neumann boundary condition

$$\begin{aligned} U_{j,0}^m = U_{j,2}^m,\ U_{j,N_2+1}^m = U_{j,N_2-1}^m, &\quad j = 1, 2, \ldots, N_1, \\ U_{0,k}^m = U_{2,k}^m,\ U_{N_1+1,k}^m = U_{N_1-1,k}^m, &\quad k = 1, 2, \ldots, N_2, \end{aligned} \tag{9}$$

one gets a determined linear system at each time step, defined by a sparse matrix.

3.2. Implicit Method

In the case of the fully implicit method, the diffusion term depends on the unknowns $U_{j,k}^{m+1}$ at each time step m, then the system becomes nonlinear, namely

$$\left[1 + \frac{h_t}{2}\left(D_{(j+1,k)}^{m+1} + 2D_{(j,k)}^{m+1} + D_{(j-1,k)}^{m+1}\right)\right.$$

$$
\begin{aligned}
&+ \frac{h_t}{2}\left(D^{m+1}_{(j,k+1)} + 2D^{m+1}_{(j,k)} + D^{m+1}_{(j,k-1)}\right)\Big]U^{m+1}_{j,k} \\
&- \frac{h_t}{2}\Big[\left(D^{m+1}_{(j+1,k)} + D^{m+1}_{(j,k)}\right)U^{m+1}_{(j+1,k)} \\
&+ \left(D^{m+1}_{(j,k)} + D^{m+1}_{(j-1,k)}\right)U^{m+1}_{(j-1,k)}\Big] \\
&- \frac{h_t}{2}\Big[\left(D^{m+1}_{(j,k+1)} + D^{m+1}_{(j,k)}\right)U^{m+1}_{(j,k+1)} \\
&+ \left(D^{m+1}_{(j,k)} + D^{m+1}_{(j,k-1)}\right)U^{m+1}_{(j,k-1)}\Big] - U^{m}_{j,k} = 0,
\end{aligned} \tag{10}
$$

where

$$
D^{m+1}_{(j,k)} = \frac{e^{i\vartheta}}{1 + \left(\frac{U^{m+1}_{I(j,k)}}{\kappa\vartheta}\right)^2}, \tag{11}
$$

where $U^m_{I(j,k)}$ is the imaginary part of $U^m_{(j,k)}$. Therefore a linearization approach is needed, which can be done by applying Newton's method.

3.3. Newton's Method

In order to apply Newton's method, it is necessary to separate the real and imaginary parts of the solution, rewriting now Equation (10) in order to create the Equation (12)

$$
\begin{aligned}
F_{R(j,k)}\Big(&U^{m+1}_{R(j,k)}, U^{m+1}_{I(j,k)}, U^{m+1}_{R(j+1,k)}, U^{m+1}_{I(j+1,k)}, U^{m+1}_{R(j-1,k)}, U^{m+1}_{I(j-1,k)}, \\
&U^{m+1}_{R(j,k+1)}, U^{m+1}_{I(j,k+1)}, U^{m+1}_{R(j,k-1)}, U^{m+1}_{I(j,k-1)}\Big) = 0, \\
F_{I(j,k)}\Big(&U^{m+1}_{R(j,k)}, U^{m+1}_{I(j,k)}, U^{m+1}_{R(j+1,k)}, U^{m+1}_{I(j+1,k)}, U^{m+1}_{R(j-1,k)}, U^{m+1}_{I(j-1,k)}, \\
&U^{m+1}_{R(j,k+1)}, U^{m+1}_{I(j,k+1)}, U^{m+1}_{R(j,k-1)}, U^{m+1}_{I(j,k-1)}\Big) = 0
\end{aligned} \tag{12}
$$

where $F_{R(j,k)}$ is the real part of left hand side of (10) and $F_{I(j,k)}$ is its imaginary part.

Note that Equation (10) is around the point (j,k), leading to $N_1 \times N_2$ nonlinear complex equations.

$$
F(U) = \left[F_{R(1,1)}, F_{R(1,2)}, \ldots, F_{R(N_1,N_2)}, \ldots F_{I(1,1)}, F_{I(1,2)}, \ldots, F_{I(N_1,N_2)}\right]^T,
$$

to which we now apply Newton's method.

At each step of the finite difference method, we consider the initial iteration of Newton's method given by the solution in the current step m, i.e.,

$$
x_0 = [U^m_{R(1,1)}, U^m_{R(1,2)}, \ldots, U^m_{I(N_1,N_2)}, U^m_{I(1,1)}, U^m_{I(1,2)}, \ldots, U^m_{I(N_1,N_2)}].
$$

We now define the Fréchét derivative of F, which in this case is the Jacobian $2 \times N_1 \times N_2$ matrix of F given by

$$F' = \left[\begin{array}{ccc|ccc} & \vdots & & & \vdots & \\ \cdots & \dfrac{\partial F_{R(j,k)}}{\partial U^{m+1}_{R(n,p)}} & \cdots & \cdots & \dfrac{\partial F_{R(j,k)}}{\partial U^{m+1}_{I(n,p)}} & \cdots \\ & \vdots & & & \vdots & \\ \hline & \vdots & & & \vdots & \\ \cdots & \dfrac{\partial F_{I(j,k)}}{\partial U^{m+1}_{R(n,p)}} & \cdots & \cdots & \dfrac{\partial F_{I(j,k)}}{\partial U^{m+1}_{I(n,p)}} & \cdots \\ & \vdots & & & \vdots & \end{array}\right]$$

where each block is a matrix $N_1 \times N_2$ given by

$$\frac{\partial F_{(j,k)}}{\partial U^{m+1}_{(n,p)}} = \left[\begin{array}{ccc} \dfrac{\partial F_{(1,1)}}{\partial U^{m+1}_{(1,1)}} & \dfrac{\partial F_{(1,1)}}{\partial U^{m+1}_{(1,2)}} & \cdots & \dfrac{\partial F_{(1,1)}}{\partial U^{m+1}_{(N_1,N_2)}} \\ \dfrac{\partial F_{(1,2)}}{\partial U^{m+1}_{(1,1)}} & \dfrac{\partial F_{(1,2)}}{\partial U^{m+1}_{(1,2)}} & \cdots & \dfrac{\partial F_{(1,2)}}{\partial U^{m+1}_{(N_1,N_2)}} \\ \vdots & & \ddots & \vdots \\ \dfrac{\partial F_{(N_1,N_2)}}{\partial U^{m+1}_{(1,1)}} & \dfrac{\partial F_{(N_1,N_2)}}{\partial U^{m+1}_{(1,2)}} & \cdots & \dfrac{\partial F_{(N_1,N_2)}}{\partial U^{m+1}_{(N_1,N_2)}} \end{array}\right].$$

The characterization of each derivative is fully described in [10, pp.19–25]. We note that the Jacobian changes between two different steps of the finite difference method, since it depends on the approximation U^m at the step m.

We now iterate Newton's method by

$$\begin{cases} F'(x_n)h = F(x_n), \\ x_{n+1} = x_n + h. \end{cases}$$

As stopping criteria, we defined that the Euclidean norm between two different iterations should be less than some pre-defined tolerance δ, i.g.

$$\|x_{n+1} - x_n\|_2 \leq \delta.$$

4. Performance Metrics

In order to compare the performance of the methods, we considered different metrics. On one hand, noise needs to be removed, on the other hand, important features should be kept in the image. Therefore we will consider metrics between appropriate images, that take into account one or both of the previous.

In this way, we consider the Mean Squared Error (MSE) given by

$$\mathrm{MSE}\left(I, U^{N_t}\right) = \frac{1}{N_1 N_2} \sum_{j=1}^{N_1} \sum_{k=1}^{N_2} \left(I_{(j,k)} - U^{N_t}_{(j,k)}\right)^2,$$

that measures the difference between the images I and U^{N_t}. In addition, we consider the Peak Signal-to-Noise Ratio (PSNR) given by

$$\mathrm{PSNR}\left(I, U^{N_t}\right) = 10.\log_{10}\left(\frac{255^2}{MSE}\left(I, U^{N_t}\right)\right)$$

where 255 is the maximum possible value of the image. We note that if the MSE goes to zero, the PSNR goes to infinity. Therefore a higher PNSR is related to higher performance of the noise filter.

Moreover, we considered the Mean Structural Similarity (MSSIM) [18],[17] and [6] given by

$$\mathrm{MSSIM}\left(I, U^{N_t}\right) = \frac{1}{N_1 N_2} \sum_{j,k} SSIM\left(I^{(W_{i,j})}, U^{N_t}_{(W_{i,j})}\right)$$

where $I^{(W_{i,j})}$ (respectively, $U^{N_t}_{(W_{i,j})}$) is a window of the original image I (respectively, filtered image $U^{N_t}_{(W_{i,j})}$) around the pixel (i, j), and the Structural Similarity (SSIM) is a local measure given by

$$\mathrm{SSIM}\left(I_1, I_2\right) = \frac{(2\mu_1\mu_2 + C_1)(2\sigma_{12} + C_2)}{(\mu_1^2 + \mu_2^2 + C_1)(\sigma_1^2 + \sigma_2^2 + C_2)}$$

where μ_k and σ_k and σ_{12} are respectively the weighted mean, the weighted standard deviations of the values of I_k, $k = 1, 2$, and the weighted covariance, given by

$$\mu_k = \sum_i \sum_j w_{ij} I_k(i, j), \quad k = 1, 2$$

$$\sigma_k = \sum_i \sum_j w_{ij} (I_k(i, j) - \mu_k)^2, \quad k = 1, 2$$

$$\sigma_{1,2} = \sum_i \sum_j w_{ij}(I_1(i,j) - \mu_1)(I_2(i,j) - \mu_2)$$

with non-negative weights w_{ij} satisfying $\sum_i \sum_j w_{ij} = 1$. As in [18], we consider the weights as a gaussian of standard deviation of 1.5, in a window of 11×11 pixels. Moreover $C_k = (k_k L)^2, \quad k = 1, 2$ are constants to stabilize the division by denominators close to zero, where $L = 255$ is the dynamic range of the pixel-values, and $k_1 = 0.01$ and $k_2 = 0.03$ as in [18]. The MSSIM checks for the overall similarity between two images, assuming the value 1 if the images are equal.

For standard images (Lena and the Cameraman), we added noise with a uniform distribution. In this way we can compare the noiseless original image and the filtered image. In this way, we compute the MSE between the original (noiseless) image I_0 and the filtered image U^{N_t} and the end time T. If the MSE goes to zero, the method performs well, since the filtered image is close to the noiseless image. In a similar way, if the PNSR between I_0 and U^{N_t} is high, the method performs well. In the same way, the MSSIM between these two images measures the similarity between them, being results close to 1 related with a good performance of the filtering process without eliminating important features from the original image.

As for OCT images, no original noiseless image exists, since the acquired image is already noisy. In this case, we consider two approaches using: *a*) the OCT noisy image and the filtered image and *b*) a synthetic original noiseless image and added synthetic speckle OCT noise as described in [16] (giving rise to a initial noisy image), and the filtered image. For *a*) the meaning of the metrics is different since there is no ground truth (they are computed between the initial noisy image and the filtered image), while for *b*) one can follow the same procedure as previously mentioned for standard images.

5. Results

In this section we will illustrate the obtained results for both standard image used in image denoising performance testing (Lena and Cameraman) and OCT images, which are real medical images in need of image denoising.

For the discretization we considered a step in time $h_t = 0.025$ for standard images and $h_t = 0.05$ for OCT images, and different values for the final time T for both semi-implicit and implicit methods. We considered $\delta = 10^{-8}$ for the stopping criteria of Newton's method, which was chosen by trial and error. The experiments for the OCT images were made using a PC machine with Intel ®

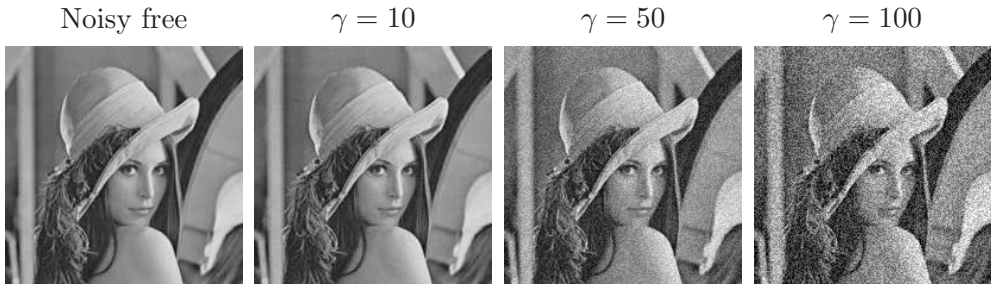

Figure 1: Standard images with different level of noise.

Core^{TM}i5 processor running Microsoft Windows 7 and for the standard images in a Intel ® Core^{TM}i7 running Microsoft Windows 10 both with 8GB RAM memory, 64 bits and Matlab ® R2012b.

5.1. Standard Images (Lena and Cameraman)

We considered 3 different sets of noise for images of size 200x200 pixels. The noise was added to the original image I to create a noisy image $I_{noise} = I + E$, where E is a 200x200 image with each pixel with random intensity with uniform distribution with zero mean and range $[-\gamma, \gamma]$, with $\gamma = 10, 50, 100$, corresponding to three different levels of noise. In this way, E is the noise component of the noisy image. In Figure 1 we present an example of the original (noiseless) image and noisy images with the three considered levels of noise.

The dataset of 2000x200 images is constituted by 10 noisy images of Lena for each noise level (in a total of 30 images) and 10 noisy images of the Cameraman for each level of noise generated independently. To study the influence of the image size and resolution, we considered noise images for different sizes as well, namely 100x100 and 300x300.

The images were filtered by the proposed methods and the results of the filtering processing will be documented in the following sections.

5.1.1. Influence of the Noise Level

Tables 1, 2 and 3 show metrics for images with the same size (200x200), same $\vartheta = \pi/180$ and $\kappa = 10$ and different levels of noise $\gamma = 10, 50, 100$ respectively.

These tables illustrate that the optimal final time T changes with the level of noise. In fact, for $\gamma = 10$ the optimal T is around 0.1, while for $\gamma = 50$ and $\gamma = 100$ it is around 0.3 and 0.5, respectively.

Table 1: Metrics for semi-implicit and implicit methods - $\gamma = 10$, $\vartheta = \pi/180$ and $\kappa = 10$, image size = 200x200

T	Scheme	Boundary Condition	Cameraman			Lena		
			MSE	PSNR	MSSIM	MSE	PSNR	MSSIM
0	-	-	8.320±0.034	38.964±0.018	0.945±0.000	8.313±0.036	38.967±0.019	0.956±0.000
0.1	semi-implicit	Neumann	16.244±0.101	36.057±0.027	0.965±0.000	9.803±0.085	38.251±0.038	0.970±0.001
		Dirichlet	17.631±0.106	35.701±0.026	0.965±0.000	10.639±0.102	37.896±0.042	0.971±0.001
	implicit	Dirichlet	14.870±0.097	36.441±0.028	0.965±0.000	9.457±0.083	38.407±0.038	0.971±0.000
0.2	semi-implicit	Neumann	33.052±0.142	32.972±0.019	0.965±0.000	17.683±0.113	35.688±0.028	0.967±0.000
		Dirichlet	34.920±0.159	32.733±0.020	0.966±0.000	19.335±0.139	35.301±0.031	0.968±0.001
	implicit	Dirichlet	30.577±0.138	33.310±0.020	0.966±0.000	16.945±0.111	35.874±0.029	0.968±0.000
0.3	semi-implicit	Neumann	51.482±0.171	31.047±0.014	0.958±0.000	27.124±0.131	33.830±0.021	0.957±0.001
		Dirichlet	52.654±0.197	30.949±0.016	0.960±0.001	29.193±0.163	33.511±0.024	0.958±0.000
	implicit	Dirichlet	48.431±0.167	31.312±0.015	0.959±0.000	26.169±0.128	33.986±0.021	0.958±0.000
0.5	semi-implicit	Neumann	88.908±0.211	28.674±0.010	0.937±0.000	46.664±0.150	31.474±0.014	0.933±0.001
		Dirichlet	86.363±0.249	28.800±0.013	0.941±0.000	48.622±0.195	31.295±0.017	0.935±0.000
	implicit	Dirichlet	85.358±0.209	28.851±0.011	0.938±0.000	45.524±0.146	31.581±0.014	0.933±0.001

Table 2: Metrics for semi-implicit and implicit methods - $\gamma = 50$, $\vartheta = \pi/180$ and $\kappa = 10$, image size = 200x200

T	Scheme	Boundary Condition	Cameraman			Lena		
			MSE	PSNR	MSSIM	MSE	PSNR	MSSIM
0	-	-	208.015±1.145	24.984±0.024	0.549±0.001	208.073±0.819	24.983±0.017	0.581±0.001
0.1	semi-implicit	Neumann	121.117±0.481	27.333±0.017	0.646±0.001	113.890±0.555	27.600±0.021	0.687±0.001
		Dirichlet	122.529±0.546	27.282±0.019	0.647±0.001	114.673±0.539	27.570±0.020	0.687±0.001
	implicit	Dirichlet	123.003±0.505	27.266±0.018	0.644±0.001	116.458±0.558	27.503±0.021	0.684±0.001
0.2	semi-implicit	Neumann	99.161±0.509	28.201±0.022	0.716±0.001	82.178±0.499	29.017±0.026	0.755±0.001
		Dirichlet	101.124±0.556	28.116±0.024	0.718±0.001	83.574±0.480	28.944±0.025	0.757±0.001
	implicit	Dirichlet	99.330±0.502	28.194±0.022	0.713±0.001	83.774±0.505	28.934±0.026	0.752±0.001
0.3	semi-implicit	Neumann	96.912±0.639	28.301±0.029	0.765±0.001	70.508±0.480	29.682±0.030	0.799±0.001
		Dirichlet	98.615±0.657	28.225±0.029	0.768±0.002	72.388±0.461	29.568±0.028	0.801±0.001
	implicit	Dirichlet	95.791±0.624	28.352±0.028	0.763±0.001	71.203±0.491	29.640±0.030	0.797±0.001
0.5	semi-implicit	Neumann	113.604±0.855	27.611±0.033	0.821±0.001	69.060±0.482	29.773±0.030	0.844±0.002
		Dirichlet	112.422±0.841	27.656±0.033	0.825±0.001	70.986±0.465	29.653±0.029	0.847±0.002
	implicit	Dirichlet	111.213±0.847	27.703±0.033	0.821±0.001	68.855±0.502	29.785±0.031	0.843±0.001

Table 3: Metrics for semi-implicit and implicit methods - $\gamma = 100$, $\vartheta = \pi/180$ and $\kappa = 10$, image size =200x200

T	Scheme	Boundary Condition	Cameraman			Lena		
			MSE	PSNR	MSSIM	MSE	PSNR	MSSIM
0	-	-	833.285±4.849	18.957±0.025	0.353±0.001	834.742±3.808	18.949±0.020	0.341±0.001
0.1	semi-implicit	Neumann	472.034±2.894	21.425±0.027	0.421±0.001	464.945±2.398	21.491±0.022	0.432±0.001
		Dirichlet	471.383±2.951	21.431±0.027	0.422±0.001	463.883±2.397	21.501±0.022	0.433±0.001
	implicit	Dirichlet	493.424±3.097	21.233±0.027	0.416±0.001	486.703±2.501	21.292±0.022	0.425±0.001
0.2	semi-implicit	Neumann	347.052±2.315	22.761±0.029	0.468±0.001	329.395±2.046	22.987±0.027	0.492±0.001
		Dirichlet	346.114±2.352	22.772±0.030	0.471±0.001	327.926±2.052	23.007±0.027	0.496±0.001
	implicit	Dirichlet	362.436±2.485	22.572±0.030	0.461±0.001	345.945±2.120	22.774±0.026	0.485±0.001
0.3	semi-implicit	Neumann	283.812±2.106	23.634±0.032	0.507±0.001	256.195±1.904	24.079±0.032	0.541±0.001
		Dirichlet	283.901±2.095	23.633±0.032	0.512±0.001	256.096±1.904	24.081±0.032	0.545±0.001
	implicit	Dirichlet	294.114±2.231	23.480±0.033	0.501±0.001	268.055±1.957	23.882±0.031	0.534±0.001
0.5	semi-implicit	Neumann	228.530±2.117	24.575±0.040	0.575±0.002	181.938±1.825	25.566±0.043	0.619±0.002
		Dirichlet	229.854±1.996	24.550±0.038	0.580±0.002	185.063±1.775	25.491±0.041	0.622±0.002
	implicit	Dirichlet	232.975±2.165	24.492±0.040	0.570±0.002	188.308±1.849	25.416±0.043	0.613±0.002

5.1.2. Influence of the Parameter ϑ

Tables 4, 2 and 5 show metrics for images with the same size (200x200), same $\kappa = 10$, same level of noise $\gamma = 50$ and different values of ϑ, namely the values $\vartheta = \pi/60, \pi/180, \pi/360$.

Results do not show any considerable change in quality with the variation of ϑ within the given values.

5.1.3. Influence of the Parameter κ

Tables 6, 2 and 7 show metrics for images with same size (200x200), same $\vartheta = \pi/180$, same level of noise $\gamma = 50$ and different values of $\kappa = 5, 10, 20$.

They show a slight change in quality of the denoised image according to κ. Therefore, the optimal value of T changes as well, for $\kappa = 5$ it seems that the optimal T is around 0.5 and for $\kappa = 20$ around 0.2. It seems to indicate that the higher the κ the less the diffusion time T should be. This is expected, since the increase in κ increases the absolute value of the diffusion coefficient (2)

5.1.4. Influence of the Image Size

Tables 8, 2 and 9 show metrics for images with same $\vartheta = \pi/180$, same level of noise $\gamma = 50$, same $\kappa = 10$ and different sizes of images 100x100, 200x200 and 300x300, respectively. It shows a slight change in quality according to size. Thus, the optimal value of T changes as well, e.g. for 100x100 T optimal is around 0.2 and for 300x300 it is around 0.3.

5.1.5. Computational Time

The computational time is presented in Table 10, where the mean times are presented.

In fact, all the experiments for each diffusion time T had the same computation times (up to the second). It is clear that the implicit method is slower, as expected, since at each step several iterations of Newton's method need to be computed. In addition, processing time seems not to change according to ϑ, κ or γ. However, and again as expected, the size of the image influences the computational time.

Table 4: Metrics for semi-implicit and implicit methods - $\gamma = 50$, $\vartheta = \pi/360$ and $\kappa = 10$, image size = 200x200

T	Scheme	Boundary Condition	Cameraman			Lena		
			MSE	PSNR	MSSIM	MSE	PSNR	MSSIM
0	-	-	208.015±1.145	24.984±0.024	0.549±0.001	208.073±0.819	24.983±0.017	0.581±0.001
0.1	semi-implicit	Neumann	121.120±0.482	27.333±0.017	0.646±0.001	113.890±0.554	27.600±0.021	0.687±0.001
		Dirichlet	122.532±0.546	27.282±0.020	0.647±0.001	114.674±0.539	27.570±0.020	0.687±0.001
	implicit	Dirichlet	123.005±0.505	27.266±0.018	0.644±0.001	116.458±0.558	27.503±0.021	0.684±0.001
0.2	semi-implicit	Neumann	99.170±0.509	28.201±0.022	0.716±0.001	82.184±0.499	29.017±0.026	0.755±0.001
		Dirichlet	101.135±0.556	28.116±0.024	0.718±0.001	83.580±0.480	28.944±0.025	0.757±0.001
	implicit	Dirichlet	99.339±0.501	28.194±0.022	0.713±0.001	83.779±0.505	28.934±0.026	0.752±0.001
0.3	semi-implicit	Neumann	96.926±0.639	28.301±0.029	0.765±0.001	70.516±0.480	29.682±0.029	0.799±0.001
		Dirichlet	98.629±0.657	28.225±0.029	0.768±0.002	72.397±0.461	29.568±0.028	0.801±0.001
	implicit	Dirichlet	95.804±0.624	28.351±0.028	0.763±0.001	71.211±0.491	29.639±0.030	0.797±0.001
0.5	semi-implicit	Neumann	113.622±0.855	27.610±0.033	0.821±0.001	69.070±0.482	29.772±0.030	0.844±0.002
		Dirichlet	112.442±0.841	27.656±0.033	0.825±0.001	70.997±0.465	29.653±0.028	0.847±0.002
	implicit	Dirichlet	111.231±0.847	27.703±0.033	0.821±0.002	68.865±0.502	29.785±0.032	0.843±0.001

Table 5: Metrics for semi-implicit and implicit methods - $\gamma = 50$, $\vartheta = \pi/60$ and $\kappa = 10$, image size = 200x200

T	Scheme	Boundary Condition	Cameraman			Lena		
			MSE	PSNR	MSSIM	MSE	PSNR	MSSIM
0	-	-	208.015±1.145	24.984±0.024	0.549±0.001	208.073±0.819	24.983±0.017	0.581±0.001
0.1	semi-implicit	Neumann	121.091±0.481	27.333±0.017	0.646±0.001	113.886±0.555	27.599±0.021	0.687±0.001
		Dirichlet	122.499±0.546	27.283±0.019	0.647±0.001	114.666±0.539	27.569±0.020	0.687±0.001
	implicit	Dirichlet	122.984±0.505	27.266±0.018	0.643±0.001	116.457±0.558	27.502±0.021	0.684±0.001
0.2	semi-implicit	Neumann	99.059±0.509	28.206±0.022	0.716±0.001	82.121±0.499	29.019±0.026	0.755±0.001
		Dirichlet	101.016±0.556	28.121±0.024	0.718±0.001	83.512±0.481	28.946±0.025	0.756±0.001
	implicit	Dirichlet	99.237±0.501	28.198±0.022	0.713±0.001	83.721±0.506	28.935±0.026	0.752±0.001
0.3	semi-implicit	Neumann	96.760±0.639	28.308±0.029	0.765±0.001	70.420±0.480	29.688±0.030	0.799±0.001
		Dirichlet	98.457±0.657	28.232±0.029	0.768±0.002	72.296±0.461	29.574±0.028	0.801±0.001
	implicit	Dirichlet	95.646±0.624	28.358±0.028	0.763±0.001	71.118±0.491	29.645±0.030	0.797±0.002
0.5	semi-implicit	Neumann	113.405±0.855	27.616±0.033	0.821±0.002	68.953±0.483	29.779±0.030	0.844±0.001
		Dirichlet	112.215±0.842	27.661±0.033	0.825±0.002	70.873±0.465	29.660±0.029	0.847±0.002
	implicit	Dirichlet	111.017±0.847	27.708±0.033	0.821±0.001	68.749±0.502	29.792±0.032	0.844±0.002

Table 6: Metrics for semi-implicit and implicit methods - $\gamma = 50$, $\vartheta = \pi/180$ and $\kappa = 5$, image size = 200x200

T	Scheme	Boundary Condition	Cameraman			Lena		
			MSE	PSNR	MSSIM	MSE	PSNR	MSSIM
0	-	-	208.015±1.145	24.984±0.024	0.549±0.001	208.073±0.819	24.983±0.017	0.581±0.001
0.1	semi-implicit	Neumann	125.698±0.517	27.171±0.018	0.638±0.001	119.903±0.565	27.377±0.020	0.678±0.001
		Dirichlet	125.891±0.595	27.165±0.021	0.639±0.001	119.906±0.555	27.376±0.020	0.679±0.001
	implicit	Dirichlet	130.081±0.595	27.023±0.020	0.631±0.001	125.073±0.575	27.193±0.020	0.671±0.001
0.2	semi-implicit	Neumann	102.691±0.454	28.049±0.019	0.693±0.001	90.915±0.504	28.578±0.024	0.733±0.001
		Dirichlet	104.760±7.534	27.972±0.288	0.691±0.018	90.867±0.492	28.581±0.024	0.736±0.001
	implicit	Dirichlet	107.428±8.018	27.863±0.298	0.681±0.017	94.423±0.513	28.414±0.023	0.726±0.001
0.3	semi-implicit	Neumann	94.393±0.514	28.416±0.024	0.734±0.002	77.148±0.477	29.291±0.027	0.771±0.001
		Dirichlet	94.331±3.062	28.420±0.136	0.734±0.013	77.244±0.462	29.286±0.026	0.775±0.001
	implicit	Dirichlet	96.051±3.320	28.342±0.144	0.724±0.013	79.313±0.486	29.171±0.026	0.766±0.001
0.5	semi-implicit	Neumann	94.719±0.699	28.401±0.032	0.792±0.002	67.187±0.473	29.892±0.030	0.820±0.001
		Dirichlet	91.307±1.105	28.560±0.052	0.790±0.018	67.110±0.435	29.897±0.028	0.824±0.001
	implicit	Dirichlet	93.771±0.925	28.444±0.043	0.782±0.019	67.813±0.484	29.852±0.031	0.818±0.001

Table 7: Metrics for semi-implicit and implicit methods - $\gamma = 50$, $\vartheta = \pi/180$ and $\kappa = 20$, image size = 200x200

T	Scheme	Boundary Condition	Cameraman			Lena		
			MSE	PSNR	MSSIM	MSE	PSNR	MSSIM
0	-	-	208.015±1.145	24.984±0.024	0.549±0.001	208.073±0.819	24.983±0.017	0.581±0.001
0.1	semi-implicit	Neumann	119.939±0.470	27.375±0.017	0.650±0.001	111.686±0.553	27.685±0.021	0.690±0.001
		Dirichlet	122.240±0.547	27.293±0.019	0.650±0.001	112.917±0.535	27.637±0.020	0.690±0.001
	implicit	Dirichlet	120.703±0.484	27.348±0.017	0.649±0.001	112.872±0.554	27.639±0.021	0.689±0.001
0.2	semi-implicit	Neumann	101.926±0.544	28.082±0.023	0.725±0.001	79.482±0.507	29.162±0.028	0.764±0.001
		Dirichlet	106.676±0.630	27.884±0.026	0.726±0.001	82.203±0.481	29.016±0.025	0.765±0.001
	implicit	Dirichlet	101.519±0.543	28.099±0.023	0.724±0.001	80.253±0.516	29.120±0.028	0.763±0.001
0.3	semi-implicit	Neumann	107.931±0.686	27.833±0.027	0.775±0.002	70.492±0.497	29.684±0.031	0.809±0.001
		Dirichlet	114.258±0.777	27.586±0.029	0.777±0.002	74.542±0.470	29.441±0.028	0.810±0.002
	implicit	Dirichlet	106.634±0.682	27.886±0.028	0.775±0.001	70.820±0.512	29.663±0.031	0.808±0.001
0.5	semi-implicit	Neumann	143.791±0.863	26.587±0.026	0.822±0.001	76.463±0.499	29.330±0.028	0.845±0.002
		Dirichlet	150.764±0.994	26.381±0.029	0.825±0.001	82.350±0.493	29.008±0.026	0.847±0.002
	implicit	Dirichlet	141.720±0.866	26.650±0.026	0.822±0.002	76.370±0.523	29.336±0.030	0.845±0.001

Table 8: Metrics for semi-implicit and implicit methods - $\gamma = 50$, $\vartheta = \pi/180$ and $\kappa = 10$, image size = 100x100

T	Scheme	Boundary Condition	Cameraman			Lena		
			MSE	PSNR	MSSIM	MSE	PSNR	MSSIM
0	-	-	208.441±1.827	24.975±0.038	0.641±0.002	209.195±1.481	24.959±0.031	0.715±0.002
0.1	semi-implicit	Neumann	132.064±1.695	26.957±0.056	0.719±0.003	119.617±1.018	27.387±0.037	0.795±0.002
		Dirichlet	135.710±1.818	26.839±0.058	0.719±0.003	121.727±1.269	27.311±0.045	0.796±0.001
	implicit	Dirichlet	133.343±1.659	26.915±0.054	0.717±0.003	122.455±0.992	27.285±0.035	0.793±0.001
0.2	semi-implicit	Neumann	122.436±2.206	27.286±0.078	0.769±0.003	95.262±1.033	28.376±0.047	0.836±0.001
		Dirichlet	128.635±2.440	27.072±0.082	0.771±0.003	98.804±1.356	28.217±0.060	0.838±0.001
	implicit	Dirichlet	121.229±2.182	27.329±0.078	0.768±0.003	96.652±0.977	28.313±0.044	0.835±0.001
0.3	semi-implicit	Neumann	131.340±2.639	26.982±0.087	0.803±0.003	90.860±1.122	28.581±0.054	0.857±0.001
		Dirichlet	138.571±2.909	26.749±0.091	0.805±0.003	95.224±1.435	28.378±0.066	0.860±0.001
	implicit	Dirichlet	128.480±2.642	27.077±0.089	0.802±0.003	90.987±1.049	28.575±0.050	0.857±0.001
0.5	semi-implicit	Neumann	168.248±3.283	25.906±0.084	0.835±0.003	103.374±1.326	28.021±0.056	0.868±0.001
		Dirichlet	173.529±3.549	25.771±0.089	0.839±0.003	107.418±1.592	27.854±0.064	0.873±0.001
	implicit	Dirichlet	163.796±3.338	26.022±0.088	0.836±0.003	102.065±1.246	28.076±0.053	0.869±0.001

Table 9: Metrics for semi-implicit and implicit methods - $\gamma = 50$, $\vartheta = \pi/180$ and $\kappa = 10$, image size =300x300

T	Scheme	Boundary Condition	Cameraman			Lena		
			MSE	PSNR	MSSIM	MSE	PSNR	MSSIM
0	-	-	208.378±0.808	24.976±0.017	0.499±0.000	208.540±0.684	24.973±0.014	0.831±0.005
0.1	semi-implicit	Neumann	115.111±0.401	27.554±0.015	0.608±0.001	111.961±0.434	27.674±0.017	0.633±0.001
		Dirichlet	115.902±0.454	27.524±0.017	0.609±0.001	112.442±0.416	27.655±0.016	0.633±0.001
	implicit	Dirichlet	117.391±0.426	27.468±0.016	0.604±0.001	114.456±0.450	27.578±0.017	0.629±0.001
0.2	semi-implicit	Neumann	85.295±0.330	28.856±0.017	0.688±0.001	76.856±0.364	29.308±0.021	0.714±0.001
		Dirichlet	86.168±0.444	28.811±0.023	0.690±0.001	77.662±0.349	29.262±0.020	0.715±0.001
	implicit	Dirichlet	86.368±0.348	28.801±0.018	0.684±0.001	78.549±0.379	29.213±0.021	0.710±0.001
0.3	semi-implicit	Neumann	75.940±0.343	29.360±0.019	0.747±0.001	61.966±0.340	30.243±0.024	0.770±0.001
		Dirichlet	76.389±0.492	29.335±0.028	0.749±0.001	63.095±0.341	30.165±0.023	0.772±0.001
	implicit	Dirichlet	75.948±0.354	29.360±0.020	0.744±0.001	62.855±0.355	30.181±0.025	0.767±0.001
0.5	semi-implicit	Neumann	79.863±0.403	29.141±0.022	0.817±0.001	54.560±0.345	30.796±0.027	0.833±0.001
		Dirichlet	77.810±0.586	29.254±0.033	0.821±0.001	55.744±0.385	30.703±0.030	0.835±0.001
	implicit	Dirichlet	78.770±0.411	29.201±0.023	0.816±0.001	54.652±0.355	30.789±0.028	0.832±0.001

Table 10: Times for semi-implicit and implicit methods Lena and Cameraman images.

T	Scheme	Boundary Condition	100x100		200x200		300x300		OCT
			Lena	Cameraman	Lena	Cameraman	Lena	Cameraman	
0.1	semi-implicit	Neumann	0:00:00	0:00:00	0:00:02	0:00:02	0:00:11	0:00:47	0:00:25
		Dirichlet	0:00:00	0:00:00	0:00:10	0:00:10	0:00:25	0:01:41	0:00:43
	implicit	Dirichlet	0:00:05	0:00:06	0:01:29	0:01:44	0:22:44	0:29:02	0:04:35
0.2	semi-implicit	Neumann	0:00:00	0:00:00	0:00:05	0:00:05	0:00:22	0:01:35	-
		Dirichlet	0:00:00	0:00:00	0:00:21	0:00:21	0:00:49	0:03:21	-
	implicit	Dirichlet	0:00:08	0:00:10	0:02:43	0:03:02	0:44:02	0:51:09	-
0.3	semi-implicit	Neumann	0:00:01	0:00:00	0:00:08	0:00:08	0:00:33	0:02:23	0:01:16
		Dirichlet	0:00:01	0:00:01	0:00:32	0:00:31	0:02:19	0:05:00	0:02:10
	implicit	Dirichlet	0:00:12	0:00:14	0:03:59	0:04:19	1:05:45	1:12:59	0:13:49
0.5	semi-implicit	Neumann	0:00:01	0:00:01	0:00:13	0:00:13	0:01:00	0:03:59	0:02:07
		Dirichlet	0:00:01	0:00:01	0:00:53	0:00:53	0:05:18	0:08:17	0:03:36
	implicit	Dirichlet	0:00:18	0:00:20	0:06:39	0:06:57	1:49:03	1:56:18	0:23:04

Table 11: Times for semi-implicit and implicit methods Lena and Cameraman images.

T	Scheme	Boundary Condition	100x100		200x200		300x300	
			Lena	Cameraman	Lena	Cameraman	Lena	Cameraman
0.1	semi-implicit	Neumann	0:00:00	0:00:00	0:00:02	0:00:02	0:00:11	0:00:47
		Dirichlet	0:00:00	0:00:00	0:00:10	0:00:10	0:00:25	0:01:41
	implicit	Dirichlet	0:00:05	0:00:06	0:01:29	0:01:44	0:22:44	0:29:02
0.2	semi-implicit	Neumann	0:00:00	0:00:00	0:00:05	0:00:05	0:00:22	0:01:35
		Dirichlet	0:00:00	0:00:00	0:00:21	0:00:21	0:00:49	0:03:21
	implicit	Dirichlet	0:00:08	0:00:10	0:02:43	0:03:02	0:44:02	0:51:09
0.3	semi-implicit	Neumann	0:00:01	0:00:00	0:00:08	0:00:08	0:00:33	0:02:23
		Dirichlet	0:00:01	0:00:01	0:00:32	0:00:31	0:02:19	0:05:00
	implicit	Dirichlet	0:00:12	0:00:14	0:03:59	0:04:19	1:05:45	1:12:59
0.5	semi-implicit	Neumann	0:00:01	0:00:01	0:00:13	0:00:13	0:01:00	0:03:59
		Dirichlet	0:00:01	0:00:01	0:00:53	0:00:53	0:05:18	0:08:17
	implicit	Dirichlet	0:00:18	0:00:20	0:06:39	0:06:57	1:49:03	1:56:18

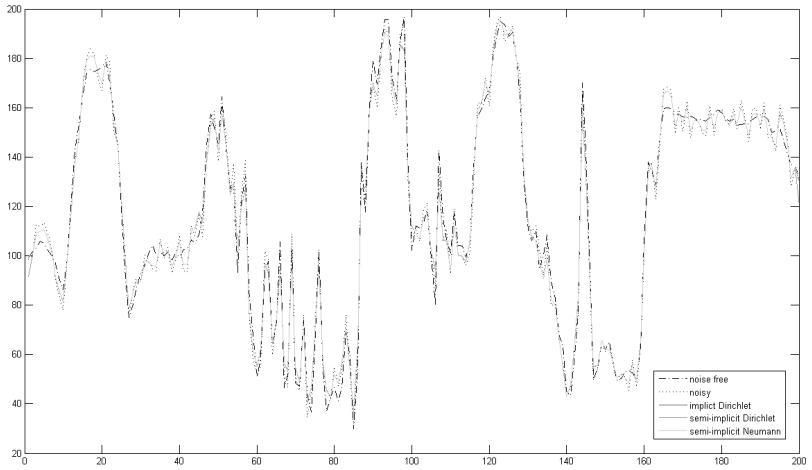

Figure 2: Profile for the middle line of the Lena image for $\gamma = 50$, comparing the three approaches for $T = 0.1$.

5.1.6. Summary

It is clear that in all cases, the methods are similar in performance. Furthermore, it is clear that in each case the optimal diffusion time T seems to depend on whether one wants to optimize MSE, PSNR or MSSIM. Therefore the choice of T must take that into account, and likewise the size of the image and the level of noise. The implicit method seems to slightly outperform the semi-implicit, while the Neumann condition seems to perform slightly better than Dirichlet. However these differences are very slight, and therefore they seem not to be worth while when taking into account the higher computational cost of the implicit method.

In Figure 3 we illustrate the filtering result for the implicit method and in Figure 4 for the semi-implicit method, both with Dirichlet boundary conditions. Moreover in Figure 2, we illustrate a profile of the images comparing the three approaches.

5.2. OCT Images

The dataset used to compute the metrics is composed by 4 different B-scan images from each of 8 OCT volumes, in a total of 32 images with dimension

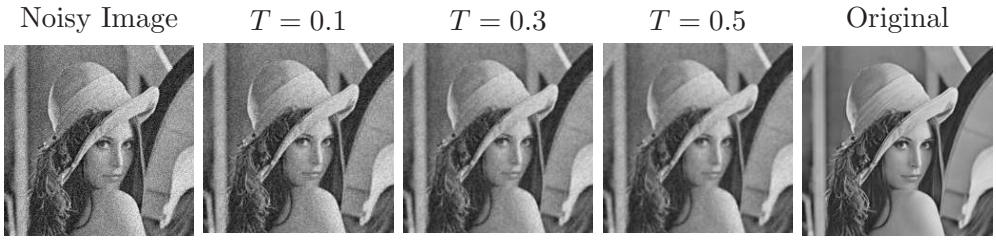

Figure 3: Result of filtering using the implicit method with Dirichlet boundary condition for Lena standard image with $\gamma = 50$.

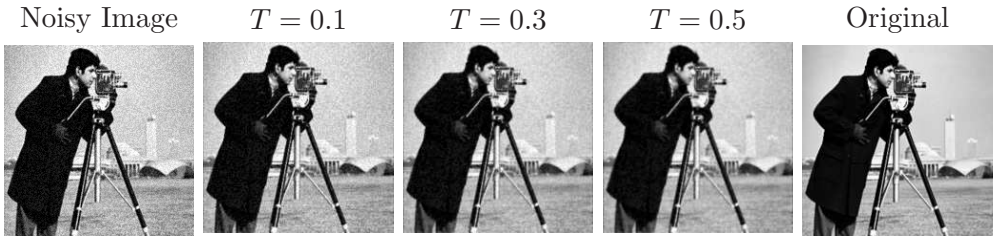

Figure 4: Result of filtering using the semi-implicit method with Dirichlet boundary condition for Camera standard image with $\gamma = 50$.

512x1024 each. Since no ground truth is known, the comparison of the metrics will be made considering the initial noisy image and the filtered image, therefore the interpretation of these metrics is slightly different than for the metrics for a original noiseless image and filtered image. For instance, in this case the MSE is expected to increase with the diffusion time, since the noise is eliminated.

In addition, we considered 4 B-scans from each of 8 synthetic OCT volumes, generated following the procedure in [16]. In this way, we have 32 synthetic images where the synthetic noiseless image is known, and to which synthetic OCT speckle noise is added to obtain a synthetic noisy OCT image, again following [16]. All the listed performance metrics can be used for these synthetic images, since the synthetic ground truth is known.

The filtering process for each image considered a constant $\kappa = 10$ and $\vartheta = \pi/180$ for the definition of the diffusion coefficient (2). The results are shown in Table 12.

We remind that for real OCT images, the performance metrics are computed between the initial (noisy) image and the filtered version, since no ground truth is known. It is clear that while MSE increases, the MSSIM does not change significantly, showing that mostly noise is being eliminate (the main features of the image remain). However, the MSSIM assumes always low values due to the

Table 12: Metrics for semi-implicit and implicit methods for OCT images.

T	k	Scheme	Boundary Condition	Synthetic OCT			Real OCT		
				MSE	PSNR	MSSIM	MSE	PSNR	MSSIM
0	-	-	-	116.94±17.44	27.52±0.62	0.55±0.03	-	-	-
0.1	constant	semi-implicit	Neumann	80.33±12.10	29.16±0.63	0.61±0.04	9.63±0.52	38.33±0.24	0.46±0.31
			Dirichlet	80.36±12.11	29.16±0.63	0.61±0.04	9.63±0.52	38.33±0.24	0.46±0.31
		implicit	Dirichlet	80.41±12.11	29.16±0.63	0.61±0.04	9.59±0.52	38.35±0.24	0.46±0.31
0.3	constant	semi-implicit	Neumann	47.04±7.17	31.49±0.63	0.69±0.03	47.58±2.56	31.40±0.23	0.47±0.28
			Dirichlet	47.15±7.19	31.48±0.63	0.69±0.03	47.63±2.57	31.39±0.24	0.47±0.28
		implicit	Dirichlet	47.15±7.19	31.48±0.63	0.69±0.03	47.44±2.55	31.41±0.23	0.47±0.28
0.5	constant	semi-implicit	Neumann	33.09±5.06	33.01±0.63	0.75±0.03	80.37±4.35	29.12±0.24	0.47±0.25
			Dirichlet	33.29±5.09	32.99±0.63	0.75±0.03	80.45±4.37	29.12±0.24	0.47±0.25
		implicit	Dirichlet	33.19±5.08	33.00±0.63	0.75±0.03	80.16±4.34	29.13±0.24	0.47±0.25

Table 13: Times for semi-implicit and implicit methods for OCT images.

T	Scheme	Boundary Condition	Times OCT
0.1	semi-implicit	Neumann	0:00:25
		Dirichlet	0:00:43
	implicit	Dirichlet	0:04:35
0.3	semi-implicit	Neumann	0:01:16
		Dirichlet	0:02:10
	implicit	Dirichlet	0:13:49
0.5	semi-implicit	Neumann	0:02:07
		Dirichlet	0:03:36
	implicit	Dirichlet	0:23:04

presence of noise in the original image that is being compared.

This performance is stressed in the synthetic OCT case, where the ground truth is known. The MSE decreases (and therefore the PNSR increases) while the MSSIM increases, showing in accordance with [4] that this is a suitable method for OCT noise filtering. However, it seems that in this case the optimal diffusion time T might be larger than $T = 0.5$, since the performance metrics continue to improve. Moreover, since the images are quite larger than in the previous section, the influence of the boundary condition seems to fade in the overall performance.

Computational times are displayed in Table 13.

We illustrate the effects of the filter in Figure 5 for real OCT and in Figure 6 for synthetic OCT.

6. Discussion and Conclusion

In this work, we proposed the development and implementation of an implicit finite difference method for solving a complex diffusion differential equation with applications to noise filtering in images of the human retina by OCT.We have made an extensive comparison of the performance of the semi-implicit and the implicit methods (ours) for complex diffusion, in the context of noise filtering. We were able to illustrate that several aspects (noise level, image size and method's parameters) influence the noise removal performance. In fact, all methods are similar in performance. However, in each case the optimal

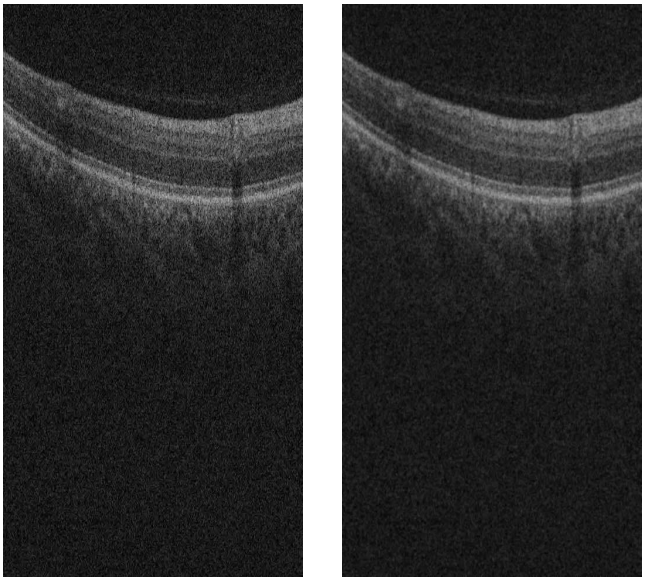

Figure 5: Real OCT image (left) and filtered image after $T = 0.5$ (right).

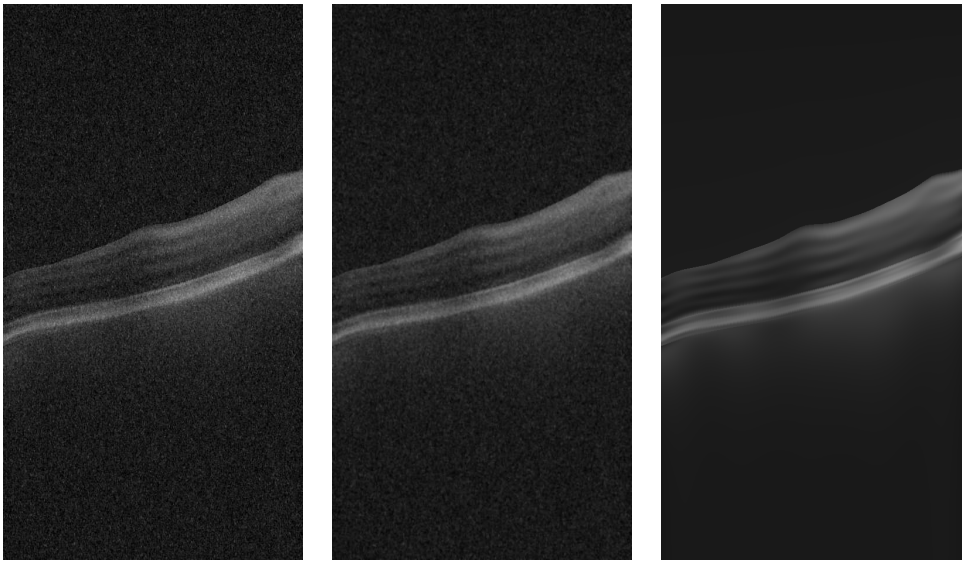

Figure 6: Synthetic Noisy OCT image (left), filtered image after $T = 0.5$ (center) and noiseless synthetic image (right).

diffusion time T seems to depend on whether one wants to optimize MSE, PSNR or MSSIM. Furthermore, the size of the image and the level of noise influence the choice of the optimal T. In addition, different values for the ϑ parameter did not show any considerable change in quality. However, the κ parameter showed a slight change in denoising quality, thus contributing to the optimal T.

As illustrated, though the implicit method has a few benefits in terms of noise removal, it is more arduous to implement and has more computational cost. As a by-result of this study, there is now available a complete characterization of the Jacobian associated with the linearization of the implicit step by the use of Newton's method [10, pp.19–25].

As illustrated in a previous work [14], it is clear that the boundary condition has little influence on the performance, though the Neumann condition seems to slightly outperform the Dirichlet case for relatively small images. For large images this effect fades out. Finally, this method is shown to be adequate for OCT filtering, and therefore it is expected to perform well for similar medical images as for instance, ultrasound.

In the future we will consider the tuning of parameters regarding the size of the image and noise level, since the optimal diffusion time seems to be dependent on that. In general, an adaptive proper choice of the parameters of the diffusion coefficient is still an open problem, though some advances have been made for OCT [4].

References

[1] A. Araújo, S. Barbeiro, and P. Serranho, Stability of finite difference schemes for complex diffusion processes, *SIAM J. Numer. Anal.*, 50 (3):1284–1296, 2012.

[2] A. Araújo, S. Barbeiro, and P. Serranho, Stability of finite difference schemes for nonlinear complex reaction-diffusion processes, *IMA Journal of Numerical Analysis*, 2014.

[3] A. Araújo, S. Barbeiro, and P. Serranho, Convergence of finite difference schemes for nonlinear complex reaction-diffusion processes, *SIAM J. Numer. Anal.*, to appear.

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

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *Computer Vision - ECCV 2004. Lecture Notes in Computer Science*, volume 3024, chapter High accuracy optical flow estimation based on a theory for warping, pages 25–36, Springer, Berlin, 2004.

[6] Richard Dosselmann and XueDong Yang, A comprehensive assessment of the structural similarity index, *Signal, Image and Video Processing*, 5(1):81–91, 2011.

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

[8] H. Grossauer and O. Scherzer, *Scale Space Methods in Computer Vision, Lecture Notes in Computer Science*, volume 2695, chapter Using the Complex Ginzburg-Landau Equation for Digital Inpainting in 2D and 3D, pages 225–236, Springer, 2003.

[9] C. Maduro, P. Serranho, T. Santos, P. Rodrigues, J. Cunha-Vaz, and R. Bernardes, Oct noise despeckling using 3d nonlinear complex diffusion filter, In Renato M. Natal Jorge, Joo Manuel R. S. Tavares, Marcos Pinotti Barbosa, and A.P. Slade, editors, *Technologies for Medical Sciences*, volume 1 of *Lecture Notes in Computational Vision and Biomechanics*, pages 141–157. Springer Netherlands, 2012.

[10] Marlon Oliveira, Método implícito para filtragem de ruído em imagem médica por difusão complexa, Master's thesis, Universidade Aberta, Portugal, 2014.

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

[12] P. Rodrigues and R. Bernardes, 3-d adaptive nonlinear complex-diffusion despeckling filter, *Medical Imaging, IEEE Transactions on*, 31(12):2205–2212, Dec 2012.

[13] P. Rodrigues, P. Serranho, and R. Bernardes, 3d nonlinear complex-diffusion filter on gpu, In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 110–113, Aug 2012.

[14] Pedro Rodrigues, Pedro Guimares, Adérito Araújo, Sílvia Barbeiro, Rui Bernardes, and Pedro Serranho, Explicit and semi-implicit complex-diffusion schemes for optical coherence tomography despeckling, In Mohamed Kamel and Aurélio Campilho, editors, *Image Analysis and Recognition*, volume 7950 of *Lecture Notes in Computer Science*, pages 282–289. Springer Berlin Heidelberg, 2013.

[15] H. Salinas and D. Fernández, Comparison of PDE-based nonlinear diffusion approaches for image enhancement and denoising in optical coherence tomography, *IEEE Trans. Med. Imaging*, 26 (6):761–771, 2007.

[16] Pedro Serranho, Cristina Maduro, Torcato Santos, José Cunha-Vaz, and Rui Bernardes, Synthetic oct data for image processing performance testing, In *International Conference in Image Processing (ICIP) 2011*, pages 401–404, 2011.

[17] Z. Wang and L. Lu and.C. Bovik, Video quality assessment based on structural distortion measurement, *Signal Processing: Image Communication,*, 19:1–9, 2004.

[18] Z. Wang, A.C. Bovik, H.R. Sheikh, and E. O. Simoncelli, Image quality assessment: From error visibility to structural similarity, *IEEE Transactions in Image Processing*, 13:600–612, 2004.

[19] J. Weickert, Anisotropic diffusion filters for image processing based quality control, *Proc. 7th Eur. Conf. Mathematics in Industry*, 1252:355–362, 1994.

[20] J. Weickert, A review of nonlinear diffusion filtering, *Scale-Space Theory in Computer Vision, ser. Lecture Notes in Computer Science*, 1252:3–28, 1997.

[21] H. Zimmer, A. Bruhn, L. Valgaerts, M. Breuss, J. Weickert, B. Rosenhahn, and H.-P. Seidel, *Vision, Modeling, and Visualization*, chapter PDE-based anisotropic disparity-driven stereo vision, pages 263–272, AKA Heidelberg, 2008.