Numer Algor (2009) 51:85–114
DOI 10.1007/s11075-009-9272-5

ORIGINAL PAPER

Parallel Schwarz alternating methods for anisotropic diffusion of speckled medical images

M. Chau · C. Tauber · P. Spiteri

Received: 23 June 2008 / Accepted: 2 February 2009 /
Published online: 28 February 2009

Abstract This paper deals with ultrasound medical image processing, particularly to filter the noise while preserving the edges and structures of information. The mathematical processing consists in solving by a numerical way a nonlinear evolutive boundary value problem. Several numerical semi-implicit time marching schemes are considered and analyzed. At each time step, parallel synchronous or asynchronous Schwarz alternating methods are used to solve the linear system and its convergence is studied. Lastly, the results of sequential and parallel simulations are presented.

M. Chau
ASA - Advanced Solutions Accelerator, 199 rue de l'Oppidum,
34 170 Castelnau le Lez, France
e-mail: mchau@advancedsolutionsaccelerator.com

C. Tauber
Institut Pasteur, URA CNRS 2582, 25 rue du docteur Roux,
75015 Paris Cedex 15, France

P. Spiteri (✉)
IRIT - ENSEEIHT, UMR CNRS 5505, 2 rue Camichel,
BP 7122, 31 071 Toulouse, France
e-mail: Pierre.Spiteri@enseeiht.fr

Present Address:
C. Tauber
UMRS INSERM U 930 - CNRS FRE 2448,
Université François Rabelais, 37 044 Tours, France
e-mail: clovis.tauber@univ-tours.fr

Keywords Parallel asynchronous algorithm · Schwarz alternating method · Subdomain method · Boundary value problem · Large scale system · Anisotropic diffusion · Optical coherence tomography · Ultrasound imaging

Mathematics Subject Classifications (2000) 68U10 · 65F10 · 65Y05 · 65N22 · 65C20 · 65Y20 · 65M12

1 Introduction

The goal of the proposed study is to compute accurate and fast anisotropic diffusion of images affected by a specific noise, called speckle. This multiplicative and locally correlated noise is due to destructive interference of signals reflected from scatters within a single resolution cell. It is common in ultrasound and optical coherence tomography medical images. This speckle reduces their contrast and the quantity of information perceived, and can lead to inaccurate image processing and medical decisions. Most existing anisotropic diffusion techniques to filter noises [26] suffer one or several of these limitations: ineffectivity to filter multiplicative noise, insufficient noise attenuation, insufficient edge preservation, instability or slowness of the method.

In a previous work [22], we proposed a robust, speckle reducing anisotropic diffusion that addresses the first three limitations, i.e. insufficient noise attenuation, edge preservation and that was specifically tailored to filter multiplicative noises.

In order to determine the restored intensity function of the image u, the present paper deals with all the mentioned previous limitations by solving a boundary values problem of diffusion (see [2]) in a bounded domain Ω. In order to prevent energy transport outside of the domain Ω, the boundary conditions are the Neumann ones. In the considered problem, the positive coefficient of diffusion $c = c(x, y, t, u)$ depends of the intensity u and consequently this boundary values problem to solve is nonlinear.

Explicit time marching numerical scheme suffers stability limitations and small time step size are necessary to avoid this drawback. Moreover, for the specific study and parti-cularly considering the definition of the coefficient of diffusion c, the use of a fully implicit time marching scheme is excluded. It is the reason why, in this original study, we propose and analyze several appropriate time marching semi-implicit discretization schemes for medical images processing. In particular, the consistency and the stability of the global semi-implicit scheme are studied.

At each time step, it is necessary to solve a large linear algebraic system. Since the size of this system is large, iterative methods are well suited to obtain the numerical solution of the algebraic system, at each time step. We therefore propose to use and analyze the Schwarz alternating method (see [13, 14]) which is well suited for very large medical images and allows to decrease the elapsed time by using parallel facilities. Finally, we prove that at each time step the matrix to invert is an M-matrix. Thus, in a parallel computation framework, it is possible to use synchronous Schwarz alternating method; and more generally, it is also possible to suppress synchronization and to use asynchronous Schwarz alternating method with or without flexible communications (see [5, 8, 11, 15–17, 21]). For both previous parallel synchronous or asynchronous methods, we analyse in an unified framework the convergence of these iterative algorithms. Furthermore, in image processing, the spatial discretization step size has a constant value defined by the distance between two consecutive pixels. Usually, the value of the spatial discretization step size is fixed to one (see [19] and [24]). Consequently, this last property involves very high convergence speed of the considered parallel domain decomposition method and leads to very efficient iterative parallel methods.

The present paper is organized as follows. Section 2 presents the anisotropic diffusion model problem. In Section 3 the discretization schemes are presented and analyzed for both the stationary problem and the evolution problem. In particular, the consistency and the stability of the explicit and semi-implicit time marching schemes are analyzed. Section 4 presents briefly synchronous or more generally asynchronous Schwarz alternating method with or without flexible communications and is also devoted to the study of the behavior of such algorithms. Lastly, experimental sequential and parallel results concerning synthetic and real ultrasound medical images are presented in Section 5.

2 Anisotropic diffusion of ultrasound images

2.1 Ultrasound image processing

Ultrasound imaging is a widely used method for medical applications. It is non-invasive, real-time and relatively inexpensive. However, compared to other medical imaging methods, it suffers from low signal to noise ratio, low contrast and the presence of speckle, which appears as a strong granularity within regions that should be homogeneous. This phenomenon, also encountered in optical coherence tomography degrades the resolution of images and makes difficult the detection of their features. Thus, the development of effective noise reduction filters for these kinds of medical images is of primary importance in order to i) enhance the image allowing better diagnosis and ii) facilitate higher level processing such as image segmentation, which aims at locating objects and boundaries.

Image filtering methods aim at reducing the noise while enhancing the perception of relevant data. Among them, anisotropic diffusion was proposed by Perona and Malik [19]. It is derived from the heat diffusion equation, whose solution is equivalent to Gaussian convolution

$$\frac{\partial u}{\partial t} - \Delta u = 0, \text{ everywhere in } \Omega, \tag{1}$$

where u represents the intensity function of the image. Perona and Malik [19] have introduced the following diffusion equation

$$\frac{\partial u}{\partial t} = \operatorname{div}(c(|\nabla u(x, y; t)|) \cdot \nabla u(x, y; t)), \tag{2}$$

where the function c, called the coefficient of diffusion, controls the diffusion process by letting the diffusion being isotropic in homogeneous regions, and slowing down the diffusion across image contours.

Anisotropic diffusion has been largely studied for additive noise filtering [1, 4, 6, 7, 24]. For example, Catté et al. [7] proposed to reduce the noise participation in the computation of the coefficient of diffusion. In their diffusion process, the argument of the coefficient of diffusion is the norm of the gradient of the Gaussian smoothed image. But most of the models proposed in the literature have inherent limitations, particularly their tendency to produce a totally flat image after a high number of iterations.

The proposed anisotropic diffusion model, called γ-diffusion [22], is specifically tailored to filter images affected by multiplicative noise, like speckle. It is based on the coefficient of variation to differentiate pixels in homogeneous regions from pixels on edges. Moreover, the speckle encountered in ultrasound images is a multiplicative noise that cannot be treated by filter developped for additive noise.

2.2 Coefficient of variation

There are two types of coefficients of variation: the global coefficient of variation, denoted in the sequel γ_g and the local coefficient of variation, denoted γ. The global coefficient of variation is defined as follows

$$\gamma_g^2(u, t) = \frac{var(u)}{\overline{u}^2}, \tag{3}$$

where $var(u)$ and $\overline{u}$ are the variance and the mean of the intensity of an area having a homogeneous reflectivity. The reflectivity being the intensity that should be perceived if the image was not corrupted by noise. The local coefficient of variation γ is a local version of the global coefficient of variation, defined as follows

$$\gamma^2(x, y, t, u) = \frac{1}{|\eta_s|} \sum_{P \in \eta_s} \frac{(u(P) - \overline{u_s})^2}{\overline{u_s}^2}, \tag{4}$$

where η_s is a neighborhood of a central pixel s. $\overline{u_s}$ is the mean intensity on η_s.

It was proved that (see [22]), in images affected by speckle, the global coefficient of variation does not depend on the reflectivity of the scene; it only depends on the multiplicative noise. Hence, the coefficient of variation characterizes the speckle affecting the image.

Moreover, it was also established that the local coefficient of variation can be written as the sum of the global coefficient of variation and a positive term that increases with the variance of the real reflectivity of the scene. This shows that γ is close to γ_g within homogeneous regions, and largely greater than γ_g on edges. Therefore, the coefficient of variation can be used as an effective edge detector in images affected by speckle.

Note that edge detectors, like the gradient of intensity, that can be effective in images affected by additive noise, fail in images with multiplicative noise. This is because in the latter, the variance of intensity increases with the brightness of the region, which is taken into account with the local coefficient of variation.

2.3 The γ-diffusion partial derivative equation

The partial derivative equation of the γ-diffusion can be written as follows on a bounded domain $\Omega \subset \mathbb{R}^2$

$$\frac{\partial u(x,y,t)}{\partial t} - \operatorname{div}(c((x,y,t,u))\nabla u(x,y,t)) = 0, \text{ everywhere in } \Omega, \tag{5}$$

and, in order to ensure the energy conservation, the boundary conditions correspond to the Neumann ones.

The considered coefficient of diffusion is derived from the weight function of the M-estimator of Tukey [23], applied to the local coefficient of variation [4]. This function assigns zero weights to outliers having a magnitude above a certain threshold. This allows stopping completely the diffusion in the directions corresponding to high values of the local coefficient of variation. As such, the intra regions intensity is conserved if the local coefficient of variation is higher enough than the global coefficient of variation on the borders $\partial\Omega$ of the region. The coefficient of variation acts here as an edge detector more robust to speckle than the gradient of the intensity, leading to a more precise filtering.

Let $\gamma(x,y,t,u)$ be the local coefficient of variation at the point $M = (x,y)$. The coefficient of γ-diffusion is

$$c(x,y,t,u) = \begin{cases} \left[1 - \dfrac{\gamma^2(x,y,t,u)}{5\gamma_g^2(u,t)}\right]^2 & \text{if } \gamma(x,y,t,u) \leq \sqrt{5}\gamma_g(u,t), \\ 0 & \text{elsewhere.} \end{cases} \tag{6}$$

2.4 Estimation of the global coefficient of variation

As stated earlier, the global coefficient of variation must be calculated in a homogeneous area. The selection of such an area requires an undesirable interaction. We thus proposed an automatic estimation of the global coefficient that avoids any interaction and allows a fine assessment of the level of speckle at each iteration. It is reasonable to consider that there are far less pixels associated with edges than pixels in homogeneous regions. The values of the local coefficient of variation corresponding to edges can thus be considered as outliers. In order to diagnose these outliers, we first robustly normalize the observations [20]:

$$\bar{\delta}_{x,y} = \frac{\gamma(x,y,t,u) - \underset{p\in\Omega}{\mathrm{med}}(\gamma(p))}{1.4826\,\underset{q\in\Omega}{\mathrm{med}}(|\gamma(q) - \underset{r\in\Omega}{\mathrm{med}}(\gamma(r))|)} \tag{7}$$

where $\underset{p\in\Omega}{\mathrm{med}}(\gamma(p))$ is the median of γ at the current time iteration.

γ_g is the value of γ above which the pixel are considered on countours. It thus corresponds to the rejection rule $\bar{\delta}_{x,y} = 1$, established by Rousseeuw [20]:

$$\gamma_g = 1.4826\,\underset{p\in\Omega}{\mathrm{med}}\left(\left|\gamma(p) - \underset{q\in\Omega}{\mathrm{med}}(\gamma(q))\right|\right) + \underset{r\in\Omega}{\mathrm{med}}(\gamma(r)). \tag{8}$$

3 Discretization schemes

Consider the following model for anisotropic diffusion of speckled images

$$\begin{cases} \dfrac{\partial u}{\partial t} - div(c.grad(u)) = 0, \text{ everywhere in } \Omega, 0 < t \leq T, \\ \dfrac{\partial u(x,y,t)}{\partial n}_{|\partial\Omega} = 0, \quad \forall t \in [0,T] \text{ (Boundary Conditions)}, \\ u(x,y,0) = u_0(x,y), \text{ (Initial Conditions)}, \end{cases} \tag{9}$$

where $\Omega = [0,d] \times [0,b]$ is a rectangular domain, $\partial\Omega$ is the boundary of the domain Ω, n is the outward vector normal to $\partial\Omega$, $u = u(x,y,t)$ is the intensity, T is a strictly positive real number and the nonnegative coefficient of diffusion $c = c(x,y,t,u)$ depends of the intensity u. Note that, for this particular application, we have

$$0 \leq c(x,y,t,u) \leq 1, \text{ everywhere in } \Omega, 0 < t \leq T. \tag{10}$$

Note also, that the boundary value problem (9) is nonlinear and that the boundary conditions correspond to the Neumann ones. In order to solve by a numerical way the problem (9), we will construct and analyze in the sequel appropriate discretization schemes.

3.1 Spatial discretization scheme background

In this section, we first consider the one-dimensional Poisson problem to obtain accurate discretization schemes. We then extend the proposed schemes for the two-dimensional Poisson problem. The linear one-dimensional Poisson problem defined in $[0, d]$ with homogeneous Neumann boundary condition can be written as follows

$$\begin{cases} -\dfrac{d}{dx}\left(c(x)\dfrac{du(x)}{dx}\right) + \delta.u(x) = f(x), x \in]0, d[, \delta > 0, \\ \dfrac{du(0)}{dx} = \dfrac{du(d)}{dx} = 0. \end{cases} \tag{11}$$

Consider a discretization of the domain $[0, d]$ by equidistant nodes such that $x_i = i.h, i = 0, \ldots, n+1$ where n is a positive integer and $h = \frac{d}{n+1}$ is the spatial discretization step-size. Assume that $u \in \mathscr{C}^4([0, d])$, and set $v(x) = c(x)\frac{du(x)}{dx}$. To obtain a discretization scheme of the problem $-v^{'}(x_i) + \delta.u(x_i) = f(x_i), 1 \leq i \leq n$, consider first a forward-difference formula such that

$$-\frac{v(x_{i+1}) - v(x_i)}{h} + \delta.u(x_i) + \mathscr{O}(h) = f(x_i), 1 \leq i \leq n;$$

in order to obtain a discretization scheme around the node x_i, consider now the following backward-difference formula

$$v(x_i) = c(x_i)\left(\frac{u(x_i) - u(x_{i-1})}{h} + \mathscr{O}(h)\right),$$

and

$$v(x_{i+1}) = c(x_{i+1})\left(\frac{u(x_{i+1}) - u(x_i)}{h} + \mathscr{O}(h)\right);$$

then, we obtain the following discretization forward-backward scheme

$$\begin{cases} \dfrac{-c(x_i)u_{i-1} + (c(x_i) + c(x_{i+1}))u_i - c(x_{i+1})u_{i+1}}{h^2} + \delta.u_i = f_i, 1 \leq i \leq n \\ u_0 = u_1 \text{ and } u_n = u_{n+1}. \end{cases} \tag{12}$$

Note that, for the inner nodes, the local truncation error of the previous scheme defined by

$$E_i^f = \frac{-c(x_i)u(x_{i-1}) + (c(x_i) + c(x_{i+1}))u(x_i) - c(x_{i+1})u(x_{i+1})}{h^2} + \delta.u(x_i) - f(x_i);$$

assume that the diffusion coefficient c and the intensity u are continuous and sufficiently differentiable; so, after very simple calculations, the local truncation error is given by

$$E_i^f = -\frac{h}{2}\left(\frac{dc(x_i)}{dx}.\frac{d^2u(x_i)}{dx^2} + \frac{d^2c(x_i)}{dx^2}.\frac{du(x_i)}{dx}\right) + \mathscr{O}(h^2). \tag{13}$$

Thus the forward-backward scheme is consistent.

The discrete scheme defined by (12) can be written as a linear system in which, due to the first and the last discrete equation, the discretization matrix is not symmetric. Nevertheless, taking into account the relations $u_0 = u_1$

and $u_n = u_{n+1}$, and injecting the two previous equalities in the corresponding scheme for the index $i = 1$ and $i = n$, the discrete scheme is now given by

$$\begin{cases} \dfrac{c(x_2)u_1 - c(x_2)u_2}{h^2} + \delta.u_1 = f_1 \\ \dfrac{-c(x_i)u_{i-1} + (c(x_i) + c(x_{i+1}))u_i - c(x_{i+1})u_{i+1}}{h^2} + \delta.u_i = f_i, 2 \leq i \leq n-1 \\ \dfrac{c(x_n)u_n - c(x_n)u_{n-1}}{h^2} + \delta.u_n = f_n, \end{cases} \tag{14}$$

and then, the discretization matrix of dimension n (instead of $(n+2)$ previously) is now symmetric, strictly diagonally dominant with strictly positive diagonal entries and negative or null off-diagonal entries; thus, since $\delta > 0$, the global discretization matrix is symmetric positive definite (see [3]). In the general case where the coefficient of diffusion can vanish for some values of x_i, we cannot conclude to the irreducibility of the matrix, contrary to the discrete schemes studied in [25]. Moreover the local truncation error of the scheme is $E_i^f = \mathcal{O}(h)$ and the scheme is of order one.

To obtain an approximation of the derivative of the function v at the node x_i, we can also use a backward-difference formula such that

$$-\frac{v(x_i) - v(x_{i-1})}{h} + \delta.u(x_i) + \mathcal{O}(h) = f(x_i), 1 \leq i \leq n,$$

and consider for the values $v(x_i)$ and $v(x_{i-1})$ the following forward-difference formula

$$v(x_i) = c(x_i)\left(\frac{u(x_{i+1}) - u(x_i)}{h} + \mathcal{O}(h)\right),$$

and

$$v(x_{i-1}) = c(x_{i-1})\left(\frac{u(x_i) - u(x_{i-1})}{h} + \mathcal{O}(h)\right);$$

then, by a similar way, we obtain the following discretization backward-forward scheme

$$\begin{cases} \dfrac{-c(x_{i-1})u_{i-1} + (c(x_{i-1}) + c(x_i))u_i - c(x_i)u_{i+1}}{h^2} + \delta.u_i = f_i, 1 \leq i \leq n, \\ u_0 = u_1 \text{ and } u_n = u_{n+1}. \end{cases} \tag{15}$$

In this case, for the inner nodes, with the same assumptions than the ones considered in (13), the local truncation error is defined by

$$E_i^b = \frac{-c(x_{i-1})u(x_{i-1}) + (c(x_{i-1}) + c(x_i))u(x_i) - c(x_i)u(x_{i+1})}{h^2} + \delta.u(x_i) - f(x_i),$$

and, after very simple calculations, is given by

$$E_i^b = \frac{h}{2}\left(\frac{dc(x_i)}{dx}.\frac{d^2u(x_i)}{dx^2} + \frac{d^2c(x_i)}{dx^2}.\frac{du(x_i)}{dx}\right) + \mathcal{O}(h^2). \tag{16}$$

Thus the backward-forward scheme is consistent. Here again we obtain a local truncation error $E_i^b = \mathcal{O}(h)$ and the scheme is of order one; moreover, the cancelation of the unknowns u_0 and u_{n+1} from the scheme defined by (15) leads to the following scheme

$$\begin{cases} \dfrac{c(x_1)u_1 - c(x_1)u_2}{h^2} + \delta.u_1 = f_1 \\ \dfrac{-c(x_{i-1})u_{i-1} + (c(x_{i-1}) + c(x_i))u_i - c(x_i)u_{i+1}}{h^2} + \delta.u_i = f_i, 2 \le i \le n-1 \\ \dfrac{c(x_{n-1})u_n - c(x_{n-1})u_{n-1}}{h^2} + \delta.u_n = f_n, \end{cases} \tag{17}$$

and, then the discretization matrix of dimension n (instead of $(n+2)$ previously) is now symmetric and strictly diagonally dominant with strictly positive diagonal entries and negative or null off-diagonal entries; thus, since $\delta > 0$, the global discretization matrix is symmetric positive definite (see [3]). For the same reasons than the forward-backward scheme when the coefficient of diffusion is locally null, the discretization matrix is not irreducible.

The combination of the previous schemes (14) and (17) can lead to a more accurate discretization scheme. Indeed, by taking the mean of both previous schemes, we note that the local truncation error for the inner nodes is now given by $E_i = \frac{1}{2}.(E_i^b + E_i^f) = \mathcal{O}(h^2)$ and the scheme is of order two; moreover the resulting scheme is now

$$\begin{cases} \dfrac{(c(x_1) + c(x_2))u_1 - (c(x_1) + c(x_2))u_2}{2.h^2} + \delta.u_1 = f_1 \\ \dfrac{-(c(x_{i-1}) + c(x_i))u_{i-1} + (c(x_{i-1}) + 2.c(x_i) + c(x_{i+1}))u_i - (c(x_i) + c(x_{i+1}))u_{i+1}}{2.h^2} \\ +\delta.u_i = f_i, \quad \text{for } 2 \le i \le n-1 \\ \dfrac{(c(x_{n-1}) + c(x_n))u_n - (c(x_{n-1}) + c(x_n))u_{n-1}}{2h^2} + \delta.u_n = f_n, \end{cases} \tag{18}$$

and, obviously, the global discretization matrix is symmetric positive definite.

Finally, for the two-dimensional elliptic problem,

$$\begin{cases} -\dfrac{\partial}{\partial x}\left(c(x,y)\dfrac{\partial u}{\partial x}\right) - \dfrac{\partial}{\partial y}\left(c(x,y)\dfrac{\partial u}{\partial y}\right) + \delta.u = f(x,y), \text{ in }]0,d[\times]0,b[, d>0, \\ \dfrac{\partial u(x,y)}{\partial n}_{|\partial\Omega} = 0, \end{cases} \tag{19}$$

we can obtain similar discretization formula for the second order derivative with respect to y. For sake of simplicity, denote in the sequel by $c_{i,j} = c(x_i, y_j)$, for $i = 1, \ldots, n+2$ and $j = 1, \ldots, m+2$, where $n+2$ and $m+2$, respectively, denotes the number of pixels on $[0, d]$ and $[0, b]$, according to the size of the image to process. Note also that, in image processing, the spatial discretization step size along each direction can be fixed, so that $h = h_x = h_y$. Then, by combining the final scheme (18) obtained in each direction, we can set in a similar way the following second order scheme for the inner grid points (i, j)

$$\begin{cases} \dfrac{-(c_{i,j}+c_{i-1,j})u_{i-1,j}+(c_{i-1,j}+2.c_{i,j}+c_{i+1,j})u_{i,j}-(c_{i,j}+c_{i+1,j})u_{i+1,j}}{2.h^2} \\ +\dfrac{-(c_{i,j}+c_{i,j-1})u_{i,j-1}+(c_{i,j-1}+2.c_{i,j}+c_{i,j+1})u_{i,j}-(c_{i,j}+c_{i,j+1})u_{i,j+1}}{2.h^2} \\ +\delta.u_{i,j} = f_{i,j},\ 1 \le i \le n,\ 1 \le j \le m, \\ u_{0,j} = u_{1,j} \text{ and } u_{n,j} = u_{n+1,j},\ 1 \le j \le m, \\ u_{i,0} = u_{i,1} \text{ and } u_{i,m} = u_{i,m+1},\ 1 \le i \le n. \end{cases}$$

Note also that, due to the Neumann boundary conditions, for the grid points near the boundary $\partial\Omega$ the above scheme can be simplified by cancelling from the corresponding equations the values of the intensity at the grid nodes which belong to $\partial\Omega$ in a similar way than the one used in the discretization of the previous one-dimensional problem. Similarly to the one-dimensional problem, since $\delta > 0$, the global discretization matrix $\frac{1}{2h^2}.A + \delta.Id$ is symmetric and strictly diagonally dominant with strictly positive diagonal entries; thus the matrix $\frac{1}{2h^2}.A + \delta.Id$ is symmetric positive definite. Moreover, since the off-diagonal entries of matrix A are nonpositive, then $\frac{1}{2h^2}.A + \delta.Id$ is a Stieltjes matrix (or symmetric M-matrix). Finally the discretization scheme is of order two and consequently is consistent.

3.2 Explicit scheme

For the solution of the evolutive problem (9), it is necessary to adapt the classical time marching schemes used for the solution of the heat conduction equation. Note that a first possible scheme is the explicit one. Let us write now the problem (9) for $t_k = k\tau$, $k \in \mathbb{N}$, where τ denotes the time step size; then, we can write

$$\frac{\partial u(x, y, t_k)}{\partial t} - \frac{\partial}{\partial x}\left(c\left(x, y, t_k, u^{(k)}\right)\frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(c\left(x, y, t_k, u^{(k)}\right)\frac{\partial u}{\partial y}\right) = 0,$$

where $u^{(k)}$ denotes the value of u when $t = t_k$. Denote $\alpha = \frac{\tau}{2h^2}$; note that, for the particular application concerning the study of image processing, the spatial discretization step size h is here mentioned for memory; in fact in image processing, h is defined implicitly by the distance between two pixels and can be fixed to the value one (see [19] and [24]); in the sequel, we will see that the value of h has no effect with respect to the behavior of the studied algorithms. Denote also by $U^{(k)}$ the vector of components $u_{i,j}^{(k)}$ that approximates $u(x_i, y_j, t_k)$; in the sequel $U^{(0)}$ is given by the components of

$u_0(x_i, y_j)$ of the initial condition arising in (9). Then, classically, the explicit scheme is given by

$$U^{(k+1)} = \left(Id - \alpha A^{(k)}\right).U^{(k)}, k = 0, 1, \ldots, \tag{20}$$

where Id is the identity matrix and the matrix $A^{(k)}$ is the discretization matrix defined at step k which entries are defined in a similar way than in Section 3.1 and where, in the present situation, the coefficients $c_{i,j}$ are replaced by $c_{i,j}^{(k)}$ with $c_{i,j}^{(k)} = c\left(x_i, y_j, t_k, u_{i,j}^{(k)}\right)$. In fact the explicit scheme (20) has the drawback to be instable if the time step size τ is too large. Using very simple and classical arguments, particularly since the modulus of the eigenvalues of the matrix $(Id - \alpha A^{(k)})$ must be less than one, it can be proved that the explicit time marching scheme is stable if the time step size τ is bounded as follows

$$\tau < \frac{4h^2}{]|A^{(k)}|[}$$

where $]|A^{(k)}|[$ denotes any norm of the matrix $A^{(k)}$. The more interesting and practical use of matrix norm concerns the case where the norm of the matrix $A^{(k)}$ is induced by the uniform norm or the l_1-norm, respectively. If we denote by $]|A^{(k)}|[_\infty$ or $]|A^{(k)}|[_1$, respectively, the corresponding particular matrix norm, then the previous condition leads here to

$$\tau < \frac{4h^2}{\max\limits_{1\leq i\leq n.m}\left(\sum\limits_{j=1}^{n.m}\left|a_{i,j}^{(k)}\right|\right)} = \frac{4h^2}{\max\limits_{1\leq j\leq n.m}\left(\sum\limits_{i=1}^{n.m}\left|a_{i,j}^{(k)}\right|\right)}.$$

3.3 Semi-implicit discretization scheme for the anisotropic diffusion problem

In the general case, it seems to be difficult to draw up an implicit scheme, since a Newton method is very hard to use due to the difficulty to compute the gradient of the local coefficient of variation. Thus, in many contributions about image processing, semi-implicit schemes are considered. Such a scheme, corresponding to a time marching scheme with two levels k and $k+1$, is defined by

$$\left(Id + \alpha A^{(k)}\right).U^{(k+1)} = U^{(k)}, k = 0, 1, \ldots, \tag{21}$$

instead of the fully-implicit scheme defined by

$$\left(Id + \alpha A^{(k+1)}\right).U^{(k+1)} = U^{(k)}, k = 0, 1, \ldots.$$

Due to the nonlinearity, the previous system can be hard to solve. For the semi-implicit scheme we have the following results

Proposition 1 *Assume that the diffusion coefficient c and the intensity u are continuous and sufficiently differentiable. Then, the semi-implicit scheme* (21) *is consistent and the local truncation error of the scheme denoted by* $E_{i,j}^{(k)}$ *verifies* $\left|E_{i,j}^{(k)}\right| \leq \mathscr{C}.(\tau + h^2)$, *where* $\mathscr{C}$ *is a constant.*

Proof Indeed, classically, the implicit scheme (20) is of order one with respect to the time and of order two with respect to the space. Then, the proof follows obviously. □

Remark 1 Note that, except for edges where the diffusion coefficient vanish, images are mostly constituted of homogeneous regions where the coefficient of variation is relatively smooth. So, the previous discretization scheme is based on realistic framework.

Proposition 2 *The semi-implicit scheme* (21) *is unconditionally stable.*

Proof Denote by $\lambda_l^{(k)}, l = 1, \ldots, m.n$ the eigenvalues of the discretization matrix $A^{(k)}$ at each time step k. Since, at each time step k, the discretization matrix $\left(I + \alpha A^{(k)}\right)$ is symmetric positive definite, then the eigenvalues $\nu_l^{(k)} = 1 + \alpha\lambda_l^{(k)}$ of $\left(I + \alpha A^{(k)}\right)$ are real and strictly positive. Moreover the matrix $\left(I + \alpha A^{(k)}\right)$ being symmetric, it is also diagonalizable; thus, if we write the system (21) in the base of the eigenvectors of the matrix $\left(I + \alpha A^{(k)}\right)$, we obtain $u_i^{(k+1)} = \frac{u_i^{(k)}}{1+\alpha\lambda_i^{(k)}} = \frac{u_i^{(0)}}{\prod_{j=0}^{k}(1 + \alpha\lambda_i^{(j)})}, \forall i = 1, \ldots, n.m$ and then $u_i^{(k+1)}$ is bounded. Thus, the scheme is unconditionally stable, which achieves the proof. □

Remark 2 Clearly, it follows from the result of Proposition 2 that on the contrary to the explicit scheme (20), the semi-implicit scheme (21) does not suffer from any time step size restriction and can be fully adapted to the desired accuracy without the need to choose small time steps for stability reasons.

3.4 Semi-implicit Gear discretization scheme for the anisotropic diffusion problem

In this subsection, in order to obtain better accuracy, we adapt the classical Gear scheme called also backward scheme (see [10]) to the studied situation. The classical Gear scheme is an implicit scheme defined with three levels. For the initialization of this backward scheme, it is necessary to use either a semi-implicit or an explicit scheme. In our situation, a fully implicit scheme is not practical to use. Then, we draw up a semi-implicit backward scheme as follows

$$\left(\frac{3}{2}Id + \alpha A^{(k)}\right).U^{(k+1)} = 2.U^{(k)} - \frac{1}{2}.U^{(k-1)}, k = 1, \ldots. \tag{22}$$

For the semi-implicit Gear scheme we have the following result

Proposition 3 *Assume that the diffusion coefficient and the intensity are continuous and sufficiently differentiable. Then, the semi-implicit Gear scheme* (22) *is consistent and the local truncation error of the scheme denoted by* $E_{i,j}^{(k)}$ *verifies* $\left|E_{i,j}^{(k)}\right| \leq \mathscr{C}(\tau^2+h^2)$, *where* $\mathscr{C}$ *is a constant.*

Proof Indeed, by Taylor's formula, we have

$$u(x,t) = u(x,t+\tau) - \tau.\frac{\partial u(x,t+\tau)}{\partial t} + \frac{\tau^2}{2}.\frac{\partial^2 u(x,t+\tau)}{\partial t^2} - \frac{\tau^3}{6}.\frac{\partial^3 u(x,t+\tau)}{\partial t^3} + \mathscr{O}\left(\tau^4\right)$$

and

$$u(x,t-\tau) = u(x,t+\tau) - 2.\tau.\frac{\partial u(x,t+\tau)}{\partial t} + 2.\tau^2.\frac{\partial^2 u(x,t+\tau)}{\partial t^2} - \frac{4.\tau^3}{3}.\frac{\partial^3 u(x,t+\tau)}{\partial t^3} + \mathscr{O}\left(\tau^4\right).$$

The following combination $\frac{1}{2}u(x,t-\tau) - 2u(x,t)$ leads to

$$\frac{\partial u(x,t+\tau)}{\partial t} = \frac{1}{\tau}\left(\frac{3}{2}u(x,t+\tau) - 2u(x,t) + \frac{1}{2}u(x,t-\tau)\right) - \frac{\tau^2}{3}.\frac{\partial^3 u(x,t+\tau)}{\partial t^3} + \mathscr{O}\left(\tau^3\right),$$

and, taking into account the local spatial truncation error obtained in Section 3.1, we have finally

$$E_{i,j}^{(k)} = -\frac{\tau^2}{3}.\frac{\partial^3 u(x,t+\tau)}{\partial t^3} - \frac{h^2}{12}M_4,$$

where M_4 is the sum of the partial derivative of order 4 with respect to x and y; which achieves the proof. □

Note that the Remark 1 holds again here.

Concerning, the stability of the semi-implicit Gear scheme we have the following result

Proposition 4 *The semi-implicit Gear scheme* (22) *is unconditionally stable.*

Proof Using the same notations than in Proposition 2, consider the semi-implicit Gear scheme and also the obvious equality $U^k = U^k$; then using a matrix notation we can write the two previous systems as follows

$$\begin{bmatrix} \left(\frac{3}{2}Id + \alpha A^{(k)}\right) & 0 \\ 0 & Id \end{bmatrix} \begin{bmatrix} U^{(k+1)} \\ U^{(k)} \end{bmatrix} = \begin{bmatrix} 2Id & -\frac{1}{2}.Id \\ Id & 0 \end{bmatrix} \begin{bmatrix} U^{(k)} \\ U^{(k-1)} \end{bmatrix}.$$

Now, we can write this system in the eigenvector base; if $\Lambda^{(k)}$ denotes the diagonal matrix which entries are the eigenvalues of the matrix $A^{(k)}$, *i.e.* $\lambda_i^{(k)}$, $i = 1, \ldots, n.m$, and $\mathscr{U}^{(k)} = \left(\bar{u}_i^{(k)}\right)$ denotes the vector $U^{(k)}$ written in the base of eigenvectors of the matrix $A^{(k)}$, then we have

$$\begin{bmatrix} \left(\frac{3}{2} Id + \alpha\Lambda^{(k)}\right) & 0 \\ 0 & Id \end{bmatrix} \begin{bmatrix} \mathscr{U}^{(k+1)} \\ \mathscr{U}^{(k)} \end{bmatrix} = \begin{bmatrix} 2Id & -\frac{1}{2}.Id \\ Id & 0 \end{bmatrix} \begin{bmatrix} \mathscr{U}^{(k)} \\ \mathscr{U}^{(k-1)} \end{bmatrix}.$$

By writing the previous system component by component, we obtain

$$\begin{bmatrix} \bar{u}_i^{(k+1)} \\ \bar{u}_i^{(k)} \end{bmatrix} = \begin{bmatrix} \frac{2}{\frac{3}{2}+\alpha\lambda_i^{(k)}} & -\frac{1}{2\left(\frac{3}{2}+\alpha\lambda_i^{(k)}\right)} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \bar{u}_i^{(k)} \\ \bar{u}_i^{(k-1)} \end{bmatrix};$$

the eigenvalues θ of the 2×2 matrix satisfy the algebraic equation

$$\left(\frac{3}{2} + \alpha\lambda^{(k)}\right)\theta^2 - 2\theta + \frac{1}{2} = 0;$$

let $\mu = 4 - 2(\frac{3}{2} + \alpha\lambda^{(k)})$ the discriminant of the previous second order algebraic equation. If μ is negative then

$$|\theta_\pm|^2 = \frac{1}{3 + 2\alpha\lambda^{(k)}} < 1;$$

If μ is positive then

$$\max(|\theta_+|, |\theta_-|) = \frac{1 + \sqrt{\frac{1}{4} - \frac{\alpha\lambda^{(k)}}{2}}}{\frac{3}{2} + \alpha\lambda^{(k)}} < 1,$$

and, by a straightforward way, in both cases $|\theta| < 1$. Thus, the studied scheme is stable, which achieves the proof. □

4 Parallel asynchronous Schwarz alternating method

Domain decomposition methods, such as the Schwarz algorithm introduced by Lions (see for example [14]) are well suited for the parallel solution of boundary values problems (see [13]).

4.1 The sequential Schwarz alternating method

Let us first recall the Schwarz alternating method in a very simple one-dimensional context. For this purpose, consider the Poisson equation with homogeneous Dirichlet boundary condition and defined in the domain $\Omega = [0, 1] \subset \mathbb{R}$

$$\begin{cases} -\dfrac{d^2u}{dx^2} = 0, \text{ everywhere in } [0, 1], \\ u(0) = u(1) = 0. \end{cases} \tag{23}$$

We consider here the sequential context and assume that Ω is splitted into two overlapping subdomains Ω_1 and Ω_2, where $\Omega_1 = [0, \gamma_1^2], 0 < \gamma_1^2 < 1$ and $\Omega_2 = [\gamma_2^1, 1], 0 < \gamma_2^1 < \gamma_1^2 < 1$, where γ_1^2 is the right boundary of Ω_1 and γ_2^1 is the left boundary of Ω_2; note that $\Omega = \Omega_1 \bigcup \Omega_2$ and $\Omega_1 \bigcap \Omega_2 \neq \emptyset$; consider accordingly the decomposition of u into two subvectors u_1 and u_2. In order to solve equation (23) by the sequential Schwarz alternating method, let us define an initial guess $u^0 = (u_1^0, u_2^0)$; then the first component u_1 is computed on the subdomain Ω_1 using the boundary conditions $u_1(0) = 0$ and $u_1(\gamma_1^2) = \tilde{u_2}$, where $\tilde{u_2}$ is the restriction at γ_1^2 of the subvector u_2 computed on the other subdomain Ω_2. On the other hand, the component u_2 is computed symmetrically on the subdomain Ω_2 using the boundary conditions $u_2(1) = 0$ and $u_2(\gamma_2^1) = \tilde{u_1}$, where $\tilde{u_1}$ is the restriction at γ_2^1 of the subvector u_1 computed on the subdomain Ω_1. Then this iterative process is repeated alternatively. In the sequential context, for choosing the values of $\tilde{u_1}$ and $\tilde{u_2}$ various strategies can be considered.

In the sequel let us denote by r the label of the Schwarz iteration. The case where at each step $(r+1)$, we have $\tilde{u_1} = u_1^r(\gamma_2^1)$ and $\tilde{u_2} = u_2^r(\gamma_1^2)$ corresponds to a Jacobi like method, similar to an additive Schwarz alternating method (see Fig. 1a). On the other hand for the computation of u^{r+1}, we can also consider that $\tilde{u_1} = u_1^{r+1}(\gamma_2^1)$ and $\tilde{u_2} = u_2^r(\gamma_1^2)$, algorithm corresponding to a Gauss-Seidel like method, similar to a multiplicative Schwarz alternating method (see Fig. 1b).

The implementation of Schwarz alternating method is very easy since we have to

- compute u_i by solving a numerically a boundary value problem on each subdomain $\Omega_i, i = 1, 2$,
- on the subdomain Ω_i, exchange the values of the subvectors u_j at the points belonging to the boundaries $\gamma_j^i, i \neq j$.

When, instead of Dirichlet boundary conditions, the Neumann ones are specified on $x = 0$ and $x = 1$ the computation process can be adapted by a straightforward way; in fact, on each subdomain $\Omega_i, i = 1, 2$ it is necessary

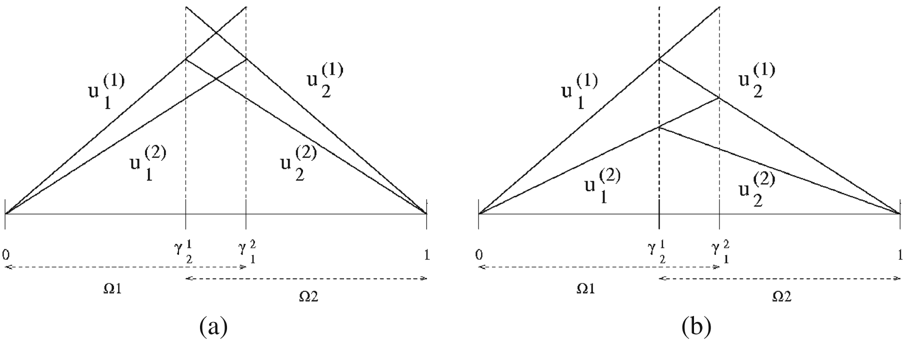


Fig. 1 Behavior of additive (**a**) and multiplicative (**b**) Schwarz alternating method

to take into account these latter boundary conditions at $x = 0$ and $x = 1$ for respectively the first and the last process. Moreover, in the case of nonhomogeneous coefficient of diffusion, the subdomain method can be adapted by a straightforward way.

More generally, this decomposition of the domain Ω can be extended to the 2D case or 3D case. Moreover, as presented in the sequel, the processing on each subdomain can be handled by a parallel way, the exchanges between the processors being synchronous or asynchronous by various ways.

4.2 Intuitive presentation of parallel asynchronous Schwarz alternating method

In order to parallelize the computation, the domain $\Omega \subset \mathbb{R}^2$ of a partial differential equation to solve is splitted into rectangular subdomains with vertical overlapping strips according to Fig. 2; note that a decomposition into rectangular blocks does not facilitate the implementation. Moreover, since speckle noise is local, the way for decomposing the studied images does not have any influence on the noise filtering. Let us denote by γ_i^j, $j = 1, 2$ the left ($j = 1$) and right ($j = 2$) boundaries between the overlapping subdomains Ω_i. Sequences of smaller subproblems are solved on each processor of a parallel computer in order to compute a solution of the global problem. In practice, since the sub-problems to solve have a smaller size, more accuracy is obtained if the systems are ill-conditioned.

Δ being any operator, consider a boundary value problem $\Delta.u = f$ on the domain Ω with boundary condition $\mathscr{B}.u = g$ on $\partial\Omega$. For the sake of simplicity, we consider first a decomposition in two subdomains Ω_1 and Ω_2. Parallel synchronous or asynchronous Schwarz algorithms for two processors consist in solving at each iteration

$$\begin{cases} \Delta_1.u_1^{r+1} = f_1 \text{ on } \Omega_1 \\ \mathscr{B}_1.u_1^{r+1} = g_1 \text{ on } \partial\Omega \cap \Omega_1 \\ u_1^{r+1} = \tilde{u}_2^r \text{ on } \partial\Omega_1 \cap \Omega_2 \end{cases} \text{and} \begin{cases} \Delta_2.u_2^{r+1} = f_2 \text{ on } \Omega_2 \\ \mathscr{B}_2.u_2^{r+1} = g_2 \text{ on } \partial\Omega \cap \Omega_2 \\ u_2^{r+1} = \tilde{u}_1^r \text{ on } \partial\Omega_2 \cap \Omega_1 \end{cases} \tag{24}$$

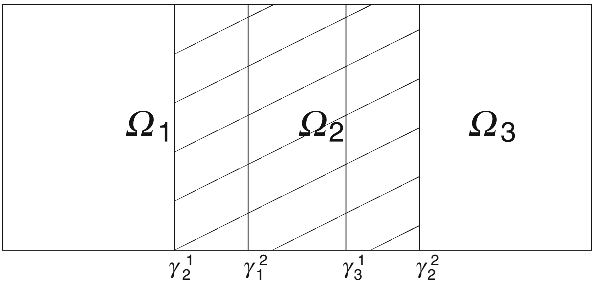


Fig. 2 Decomposition of domain Ω

where $\tilde{u}_1^r$ and $\tilde{u}_2^r$ denote again the available values of the components of the iterate vector (u_1, u_2) at the current iteration, values restricted to the overlapping boundaries.

In the synchronous algorithm, corresponding to the additive Schwarz alternating method $\tilde{u}_1^r = u_1^r$ and $\tilde{u}_2^r = u_2^r$. It is also possible to chose a multiplicative strategy of the Schwarz alternating method; nevertheless, in this last case the idle times due to necessary synchronizations increase. As previously said, each processor updates the components of the iterate vector associated with its subdomains and computes the residual norm corresponding to the subdomains in order to participate to the convergence detection. Each subproblem in (24) can be solved by any sequential algorithm; a point or a block relaxation method, for example, the Gauss-Seidel method for solving each discretized subproblem on each subdomain is a suitable algorithm. This kind of method allows more flexible communications between the processors.

Parallel asynchronous Schwarz alternating methods (see [5, 11, 15] are a general class of parallel iterative methods where iterations are carried out in parallel by several processors without any order nor synchronization; computations can be made using the current available values of each component of the iterate vector. The main feature of this class of parallel iterative methods is to allow more flexible and efficient data exchanges between the processors; as a consequence, the coupling between communication and computation is improved. The values of the components of the iterate vector which are used in an updating phase may come from updates which are in progress. Thus, the implementation principle of the parallel asynchronous Schwarz alternating method needs a particular effort concerning the combination of the solution of very large scale algebraic systems with an efficient communication strategy. When the size of the subsystems is large, note that the asynchronous communications compared to the synchronous ones suppress more efficiently the idle times (see [9]).

According to the type of communication between the processors, we can consider two distinct kinds of asynchronous Schwarz alternating method : the classical one and the asynchronous Schwarz alternating method with flexible communications.

In the classical asynchronous algorithm (see [8] and [15]), the available components may be delayed as follows : $\tilde{u}_1^r = u_1^{\rho_1(r)}$ and $\tilde{u}_2^r = u_2^{\rho_2(r)}$, with $\rho_i(r) \geq 0, i = 1, 2,$ where $\rho_i(r)$ refer to a component number; note that for such classical asynchronous method, computations are performed in parallel by using values of the block component of the iterate vector produced at the end of each updating phase. Note that when $\rho_i(r) = r$, this last kind of method generalizes also the classical parallel synchronous scheme such as parallel Jacobi method.

In the case of Schwarz alternating method with flexible communications (see [11, 16]) $\tilde{u}_i^{(r)}$ are not necessarily associated to components that are labelled by an outer iteration number as data exchanges may occur at any time, using the last values of the components just computed and available. Then, in this class of method, partial updates, i.e. the current value of each component of the iterate vector, can be used at any time in the computation; thus available values of the components of the iterate vector which are used in an updating phase may come from updates which are still in progress and communications may occur at any time; so flexible data exchanges between processors are allowed. Note that, this general asynchronous Schwarz alternating method with flexible communications extend the classical asynchronous one.

In the sequel, we will focus on cases where Δ is a linear operator. The discretization matrix of the operator Δ will be denoted by $\mathscr{A}$, the right-hand side of the linear system derived from discretization by $\mathscr{F}$ and its associated discrete solution by $\mathscr{U}$.

4.3 Numerical behavior of the parallel Schwarz alternating algorithms

In this section, we formalize the algorithm just described in Section 4.2. We have shown that the semi-linearization and the discretization of the anisotropic diffusion problem (9) by a semi-implicit scheme, leads to the solution of a large scale linear algebraic system; in the sequel we use asynchronous parallel Schwarz alternating method for the computation at each time step $k \in \mathbb{N}$ of the value $U^{(k)}$ of the semi-implicit time marching scheme. Furthermore it will be verified that at each time step $k \in \mathbb{N}$ the matrix arising in this linear system has interesting properties.

Proposition 5 *The matrix defined by* $\mathscr{A} = (\theta.Id + \alpha.A^{(k)})$, $\forall k \in \mathbb{N}$, *where* θ *is such that*

$$\theta = \begin{cases} 1 \textit{ for the semi implicit scheme} \\ \dfrac{3}{2} \textit{ for the Gear semi implicit scheme} \end{cases} \tag{25}$$

is an M-matrix.

Proof Indeed, due to the positive value of θ, the matrix $\mathscr{A}$ is strictly diagonally dominant; furthermore the diagonal entries of $\mathscr{A}$ are strictly positive and the off-diagonal entries are non positive. Thus $\mathscr{A}$ is an M-matrix □

Remark 3 Note that due to the fact that the diffusion coefficient can vanish locally, all the coefficients of the same row (or eventually of several rows) of the matrix $A^{(k)}$ can be equal to zero. Then, the matrix $\mathscr{A}$ is not necessarily irreducible. Nevertheless, due to the strict diagonal dominance, $\mathscr{A}$ is invertible.

According to previous defined notations, we consider the following system of algebraic equations

$$\mathscr{A}.\mathscr{U} = \mathscr{F} \tag{26}$$

where $\mathscr{A} \in \mathscr{L}(\mathbb{R}^{n.m})$, $\mathscr{F} \in \mathbb{R}^{n.m}$ and $\mathscr{U} \in \mathbb{R}^{n.m}$ where $\mathscr{U}$ denotes the value of $U^{(k)}$, $\forall k \in \mathbb{N}$. Furthermore, assume that

$$\mathscr{A} \text{ is an M-matrix.} \tag{27}$$

The numerical solution of (26) by the Schwarz alternating method is equivalent to the solution of the following system

$$\tilde{\mathscr{A}}.\tilde{\mathscr{U}} = \tilde{\mathscr{F}}, \tag{28}$$

where $\tilde{\mathscr{A}}$, $\tilde{\mathscr{U}}$ and $\tilde{\mathscr{F}}$ are derived from the expansion process of the Schwarz alternating method [12]. Actually in (28) the values of the unknows belonging to the overlapping area are stored and computed twice. Then we can rewrite the discretized system (26) in a new formulation (28), called augmented system, which has the same solution than the initial system (26) and which takes into account a parallel formulation applied to the initial and splitted system such that between the subsystems there are repeated parts of the unknowns; this expansion process is a theoretical model that represents the parallel solution of the algebraic system (26) by a Schwarz domain decomposition method. Then the augmented matrix $\tilde{\mathscr{A}}$ is derived from the initial one by appropriate, but not identical, repetitions of some rows. In the implementation, the system (28) is not really produced, since $\tilde{\mathscr{A}}$ and $\tilde{\mathscr{F}}$ are not explicitly computed; in fact (28) corresponds to a classical and formal notation related to the formulation of the Schwarz alternating method.

According to a result of Evans and Deren [12], the matrix $\tilde{\mathscr{A}}$ is also an M-matrix. So, the system (28), derived from the expansion process, has the same property than the one of the initial algebraic system (26).

Let $p \in \mathbb{N}$ be a positive integer and consider now the following block decomposition of problem (28) into p subproblems

$$\sum_{j=1}^{p} \tilde{\mathscr{A}_{ij}}.\tilde{\mathscr{U}_j} = \tilde{\mathscr{F}_i}, \forall i \in \{1, ..., p\} \tag{29}$$

where $\tilde{\mathscr{U}_i} \in \mathbb{R}^{(n.m)_i}$ and $\tilde{\mathscr{F}_i} \in \mathbb{R}^{(n.m)_i}$, where $(n.m)_i$ denotes the size of the i-th block of the previous vectors and $\tilde{\mathscr{A}} = (\tilde{\mathscr{A}_{ij}})_{1\leq i, j, \leq p}$, according to the associated block decomposition.

Consider now the solution of subproblems (29) by the asynchronous parallel iteration which can be classically written as follows

$$\begin{cases} \tilde{\mathscr{A}_{ii}}.\tilde{\mathscr{U}_i}^{r+1} = \tilde{\mathscr{F}_i} - \sum\limits_{j \neq i} \tilde{\mathscr{A}_{ij}}.\tilde{\mathscr{W}_j}, \text{ if } i \in s(r), \\ \tilde{\mathscr{U}_i}^{r+1} = \tilde{\mathscr{U}_i}^{r}, \text{ if } i \notin s(r), \end{cases} \tag{30}$$

where $\{\tilde{\mathscr{W}_1}, \tilde{\mathscr{W}_2}, \ldots, \tilde{\mathscr{W}_p}\}$ are the available values of the components $(\tilde{\mathscr{U}_j})_{j \neq i}$, which will be specified more precisely in the sequel, and $\mathscr{S} = \{s(r)\}_{r \in \mathbb{N}}$ is a sequence of non empty subsets of $\{1, 2, .., p\}$. In other words, $s(r)$ is the subset of the subscripts of the components updated at the r-th iteration. In the sequel, we also consider the vector

$$\mathcal{R} = \{\rho_1(r), .., \rho_p(r)\}_{r\in\mathbb{N}},$$

denoting a sequence of integer vectors from $\mathbb{N}^p$. In order to take into account the asynchronism between the processors, $\mathcal{R}$ models the delays between the parallel updates of each component, at the r-th iteration. Furthermore $\mathcal{S}$ and $\mathcal{R}$ verify the following assumptions

$$\forall i \in \{1, 2, ..., p\},\ \text{the set}\ \{r \in \mathbb{N} | i \in s(r)\}\ \text{is unbounded,}$$
$$\forall i \in \{1, 2, ..., p\}, \forall r \in \mathbb{N}, 0 \leq \rho_i(r) \leq r,$$
$$\forall i \in \{1, 2, ..., p\},\ \lim_{r\to\infty} \rho_i(r) = +\infty.$$

Remark 4 As previously said, the synchronous Schwarz alternating method is a particular case of the asynchronous one corresponding to $\rho_i(r) = r$ for all $i \in \{1, \ldots, p\}$ and for all $r \in \mathbb{N}$; if furthermore $s(r) = \{1, \ldots, p\}$ then the corresponding synchronous Schwarz alternating method is the additive one and if $s(r) = r.mod(p) + 1$ then the corresponding synchronous Schwarz alternating method is the multiplicative one (see [8]).

For the solution of system (26), the parallel asynchronous Schwarz alternating method with flexible communications corresponds to the more general model of parallel asynchronous iterations. In such a case, the access to the available values of the iterate vector $\tilde{\mathcal{W}}_j$ is obtained by the following norm constraint (see [11])

$$\left\| \tilde{\mathcal{W}}_j - \mathcal{U}_j^\star \right\|_{j,\infty} \leq \left\| W^{\rho(r)} - \mathcal{U}^\star \right\|_\infty, \forall j \in \{1, \ldots, p\} \tag{31}$$

where $W^{\rho(r)} = \left(W_1^{\rho_1(r)}, \ldots, W_p^{\rho_p(r)}\right)$ denotes the available values of the components just computed of the iterate vector, $\mathcal{U}^\star$ is the exact solution of the linear system to solve, $\| \,.\, \|_\infty$ denotes a suitable weighted uniform norm [11] and $\| \,.\, \|_{j,\infty}$ an analogous weighted uniform norm defined in $\mathbb{R}^{(nm)_j}$, $j = 1, \ldots, p$. More precisely, in the present context, the values of the components of the iterate vector $W^{\rho(r)}$ generated by the other process, can be accessed while the computations are still in progress. Thus, we have the following result derived from [11].

Proposition 6 *Assume that* (27) *is verified. Then,* $\tilde{\mathcal{W}} = \{\tilde{\mathcal{W}}_1, \ldots, \tilde{\mathcal{W}}_p\}$ *being defined according to* (31)*, the numerical solution of problem* (26) *by the parallel asynchronous Schwarz alternating method with flexible communications associated to* (30) *and starting from any initial guess* $\mathcal{U}^0$*, converges to the solution of* $\mathcal{A}.\mathcal{U} = \mathcal{F}$.

Proof As previously said, since $\mathcal{A}$ is an M-matrix, the augmented matrix $\tilde{\mathcal{A}}$, derived from the augmentation process of the Schwarz alternating method, has the important property to be an M-matrix. Thus, the convergence of the parallel asynchronous Schwarz alternating method with flexible communications, obtained by contraction techniques, is derived from the result presented, in the linear case, in [11]. □

Remark 5 The classical asynchronous Schwarz alternating method is a particular case of formulation of the asynchronous Schwarz alternating method with flexible communications (31) in which the updates are performed at the end of any relaxation and defined by $\tilde{\mathscr{W}}_j = \tilde{\mathscr{U}}_j^{\rho_j(r)}$ (see [5, 15]). This case corresponds to the classical parallel asynchronous iterations as defined in [5, 15]. Then, since $\mathscr{A}$ is an M-matrix, according to a result of [17], it can be proved, that, for any subdomain decomposition, the classical parallel asynchronous Schwarz alternating method defined by (30) and starting from any initial guess $\mathscr{U}^0$ converge to the solution of $\mathscr{A}.\mathscr{U} = \mathscr{F}$.

From the previous results, we can infer the following obvious corollary :

Corollary 1 *For each time step $k \in \mathbb{N}$, for both semi-implicit schemes, the sequential and the parallel synchronous and asynchronous Schwarz alternating method with or without flexible communications starting from any initial guess $\mathscr{U}^0$ converge to the solution of each considered discretized boundary value problem $\mathscr{A}.\mathscr{U} = \mathscr{F}$ associated with the anisotopic diffusion problem.*

Remark 6 In order to estimate the efficiency of the considered parallel asynchronous iterative method, let us give a bound of the spectral radius ρ of the point Jacobi matrix. Let us denote by $a_{i,j}^{(k)}$ the entries of the matrix $A^{(k)}$. For the evaluation of ρ, since the off-diagonal entries of the Jacobi matrix are non-negative, we have

$$\rho \leq \max_{1\leq i\leq n.m}\left(\frac{\alpha\sum\limits_{j\neq i} a_{i,j}^{(k)}}{\left(\theta + \alpha.a_{i,i}^{(k)}\right)}\right);$$

since for all i, $\sum\limits_{j\neq i} a_{i,j}^{(k)} \leq a_{i,i}^{(k)}, \forall k \in \mathbb{N}$, then

$$\rho \leq \max_{1\leq i\leq n.m}\left(\frac{\alpha a_{i,i}^{(k)}}{\theta + \alpha.a_{i,i}^{(k)}}\right).$$

Then, the maximum value of the diagonal entries of the matrix $A^{(k)}$ being bounded by (8) and since the mapping $\xi \rightarrow \frac{\alpha.\xi}{\theta+\alpha.\xi}$ is obviously increasing, then

$$\rho \leq \bar{\rho} = \frac{8.\alpha}{\theta + 8.\alpha} = \frac{4.\tau}{\theta + 4.\tau}. \tag{32}$$

Table 1 Upper-bound of the spectral radius of the Jacobi matrix

τ	Semi-implicit method	Semi-implicit Gear method
$\frac{1}{5}$	0.44444	0.34782
$\frac{1}{7}$	0.36363	0.27586
$\frac{1}{10}$	0.28571	0.21052
$\frac{1}{20}$	0.16666	0.11764

Thus, using several values of τ we have in Table 1 the following corresponding upper-bound for ρ and we can observe that the values of ρ are small and decrease with τ. Consequently, the use of relaxation method is suitable in such situation.

Remark 7 Thanks to the previous upper bound (32) of the spectral radius of the Jacobi matrix, one can obtain some interesting information concerning the asymptotic rate of convergence of the asynchronous Schwarz alternating method. Indeed, as previously said, the Schwarz alternating method leads to solve here a linear algebraic system, in which the matrix $\tilde{\mathscr{A}}$ and the right hand side $\tilde{\mathscr{F}}$ are derived from the augmentation process. In the case of classical asynchronous iterations, where the updates are performed at the end of any relaxation, if we consider the point to point decomposition, then the Jacobi matrix $\tilde{\mathscr{J}}$ associated to the augmented matrix $\tilde{\mathscr{A}}$ is a contraction matrix [17]. According to [17], when the space is normed by an appropriate weighted uniform norm, the spectral radius $\tilde{\rho}$ of $\tilde{\mathscr{J}}$ allows to obtain an estimate of the classical asymptotic rate of convergence of the asynchronous iteration defined by $R_\infty = -Log(\tilde{\rho})$. Since the augmented system to solve is derived from the initial system, then it follows from previous considerations that the spectral radius of the contraction matrix $\tilde{\mathscr{J}}$ can be upper bounded by

$$\tilde{\rho} \leq \bar{\rho} = \frac{4.\tau}{\theta + 4.\tau}. \tag{33}$$

5 Numerical simulation results

5.1 Image processing results

The general goal of an image restoration algorithm should be to reduce the variance in homogeneous regions while preserving the edges (i.e. maintaining some contrast while filtering the noise). The γ-diffusion, discretized with the proposed schemes, obtains a low variance and a relatively high contrast. Moreover, it leads to images with good edge detection and preservation.

Figure 3a presents one of the image used during the experimentation. It is an echocardiographic image featuring the left ventricle and auricle.

The first result in Fig. 3b corresponds to the diffusion over a time $T = 1$, over 10 timesteps ($\tau = 0.1$). Our filter progressively reduces the speckle in

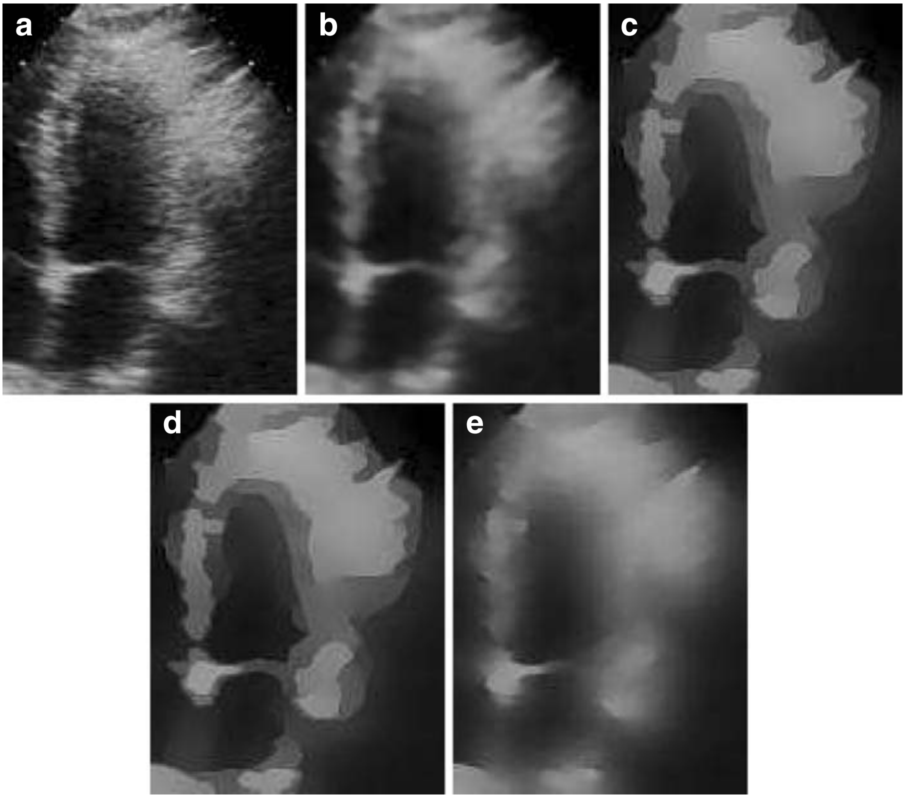


Fig. 3 Image processing results for several schemes and parameters (**a**–**e**)

homogeneous regions, while enhancing the perception of the boundaries between the different parts of the heart.

Figure 3c presents the result after a diffusion time of $T = 25$, over 1000 timesteps ($\tau = 0.0025$). The γ-diffusion leads to an almost piecewise constant image, preserving important contours of the initial image. Note that processing the diffusion for longer time, even infinite, would not change the result much. This is because our coefficient of diffusion progressively reduce and then totally stop the diffusion across edges. This is an important feature of our filter, as it preserves the total intensity within each region.

One improvement obtained with the proposed discretization scheme is presented in Fig. 3d. This result corresponds again to a diffusion over a time $T = 25$, over only 25 time steps are used ($\tau = 1$). It confirms the good stability and convergence properties, as the same quality of filtering is obtained for a very larger τ.

Finally, Fig. 3e is a result obtained when the time step used ($\tau = 1$) is too large for the discretization scheme, and the process diverges. Note that this result was not obtained with the semi-implicit scheme but with an explicit discretization scheme.

5.2 Principle of parallel implementation of the Schwarz method

Implementation and numerical experiments of parallel Schwarz alternating algorithms have been carried out on distributed memory machines. In this subsection, both synchronous and asynchronous parallel implementations of Schwarz alternating method are briefly presented. The solution of the specific evolution problem considered consists in solving one stationary problem at each time step by using the Schwarz alternating method. Parallelization is performed with domain decomposition method. Note that, at each new time step, synchronizations are mandatory, excepted inside the solution of algebraic system when asynchronous Schwarz alternating method is used.

The domain Ω, of the anisotopic diffusion problem is splitted into rectangular subdomains with minimal overlapping corresponding to two mesh points (see Fig. 2). Thus minimal values exchanges are communicated between the processors and consequently minimal computational overhead is ensured. One subdomain is assigned to each processor. Thus, the behavior of the Schwarz alternating method is additive.

The implementation of the asynchronous solver is more complex than the synchronous one due to the termination procedure, which requires the evaluation of a global state of the processes, without synchronization. Recall that parallel asynchronous iterative algorithms are general iterative methods where iterations are carried out in parallel by several processors without any order nor synchronization (see for example [11]). Using the global residual as a criterion for termination would introduce synchronizations in the asynchronous algorithm. Thus, parallel computing would suffer from idle times caused by termination procedure. Simple asynchronous termination procedures exist. One of them is presented in this section.

Various strategies of data exchange can be implemented (see [9]). Experiments show that the more data is exchanged, the faster the convergence is because the behaviour tends to multiplicative scheme. However, note that the efficiency of parallel algorithms strongly depends on the communication frequency within the computations as communications increase the overhead. It is the reason why we have chosen a strategy based on middle frequency communications between the processors; then the overhead is well reduced. Point to point communications between two processes have been implemented using persistent communications request in MPI facilities (Message Passing Interface). Message exchanges with the same argument list is repeatedly executed; it corresponds to data transmission of successive values of the components of the iterate vector associated with subdomain boundaries. That is the reason why persistent communications request has been used. A persistent communication request can be thought of as a one way channel. This approach permits one to reduce the communication overhead between the process and the communication controller.

In the considered implementation, each processor solves its subsystem with Gauss-Seidel algorithm. At the beginning of every inner iteration in the PDE solver, an attempt of receiving a newer boundary value is made. Each time a message is received, the local solver restarts with a new right hand side and uses the latest iterate vector as the initial guess. After that, the PDE solver computes the local solution. Each time a subdomain solver converges, a new set of values is communicated to other processors. Practically, local convergence detection is based on the norm of the local residual. This parallel process is performed until global convergence. Our global termination procedure is based on two rules:

1. when a processor reaches convergence on its subdomain, the set of values that need to be sent must be sent only once;
2. restart is required only when local convergence is violated after a boundary update.

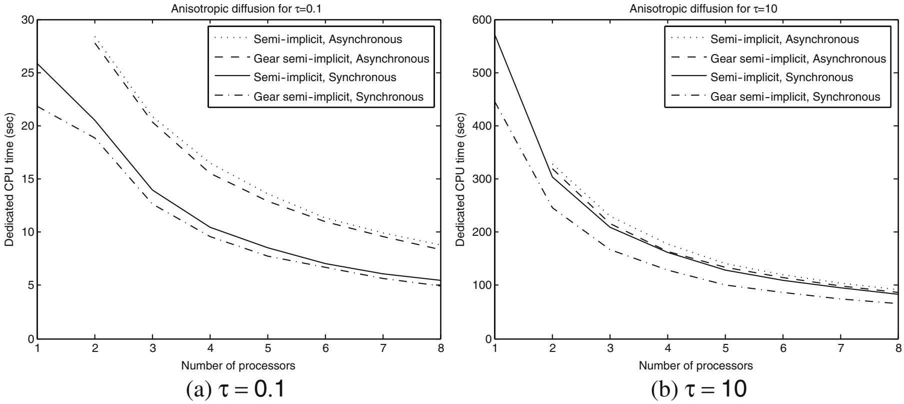
To summarize, each processor must count how many messages are sent and received as Schwarz iterations progress. These messages accounts can be collected by a master processor with an asynchronous gathering protocol. The convergence of asynchronous Schwarz algorithm will lead to the following situation: no new message is sent thanks to the former rules. Finally, global termination occurs when sent messages count equals received messages count and all processors have reached local convergence.

If global convergence is detected, then computations can be terminated and resources can be freed. All persistent communication requests are cancelled. Note that cancellation of send requests must occur before cancellation of receive requests; otherwise data exchange based on rendezvous mechanism may fail. For more details on the implementation of asynchronous iterative schemes of computation, the reader is referred to [9]. From a mathematical point of view, this implementation of the stopping criterion of the parallel asynchronous iterative process is based on the study presented in [18]. Indeed, in the context of perturbed computations by round-off errors, this previous study uses the fact that the components of the successive iterates belong to nested sets centered on the solution; then the parallel asynchronous iterative process stops when the diameter of the nested sets is smaller than a given tolerance.

Implementation of parallel synchronous iterative schemes of computation was based on the blocking reception of boundary values. The termination order of communications requests is totally handled with MPI facilities. It is not necessary to provide additional information about synchronous Schwarz alternating method, since its implementation and message passing issues between the processors are straightforward in this case. Reference is made to [9] for implementation details concerning parallel synchronous iterative algorithms.

5.3 Schwarz simulation results

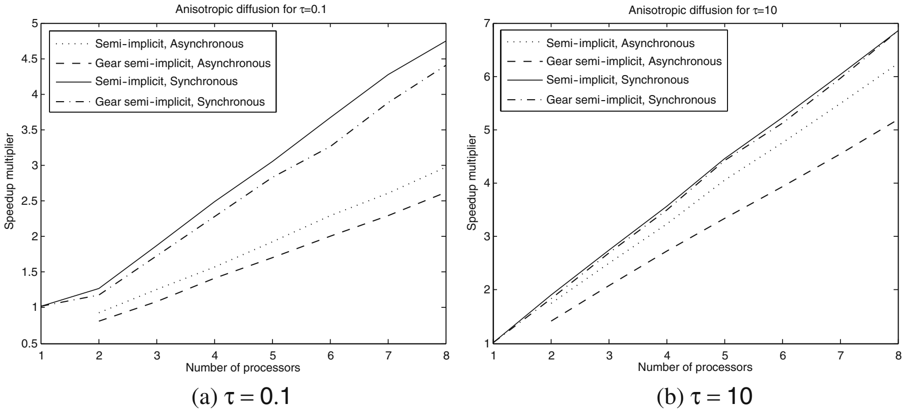
Parallel simulations have been performed on a cluster located at the IRIT - ENSEEIHT laboratory in Toulouse; this cluster is constituted by eight Sun-Blade 1500 equipped with 1.5 Ghz Sparc III processors connected via



(a) $\tau = 0.1$ (b) $\tau = 10$

Fig. 4 Elapsed time of parallel experiments for two values of the time step (**a**, **b**)

a 100 Mbits/s ethernet network. The measure of performances has been performed for the processing of a synthetic ultrasound image, that admits 1024 × 1024 pixels; then the size of the algebraic system to invert is 1044484. In the parallel simulations the number of processors used is limited to eight. Note that the simulations are performed twice; once using a small time step $\tau = 0.1$ and once using a larger one equal to $\tau = 10$. Figure 4a and b show the computation time when several processors are used for respectively the small and the large time step. Similarly, Fig. 5a and b show the speed-up according to the number of processors used, for the two considered time step. The results of parallel simulation when $\tau = 0.1$, are summarized in Tables 2 to 4; Table 2 gives the elapsed time necessary for convergence for sequential, parallel synchronous and asynchronous Schwarz alternating methods; Table 3 gives the speed-up of the synchronous and asynchronous methods and in the



(a) $\tau = 0.1$ (b) $\tau = 10$

Fig. 5 Speed up of parallel experiments for two values of the time step (**a**, **b**)

Table 2 Dedicated C.P.U. time (sec.) with $\tau = 0.1$ for computation of 1024 × 1024 synthetic image

Nb proc.	Semi-implicit		Gear semi-implicit	
	Asynchronous	Synchronous	Asynchronous	Synchronous
1	–	25.86 s.	–	21.79 s.
2	28.38 s.	20.52 s.	27.74 s.	18.83 s.
3	20.84 s.	13.92 s.	20.37 s.	12.68 s.
4	16.53 s.	10.42 s.	15.55 s.	9.59 s.
5	13.56 s.	8.51 s.	12.88 s.	7.75 s.
6	11.34 s.	7.07 s.	10.95 s.	6.71 s.
7	9.97 s.	6.07 s.	9.57 s.	5.65 s.
8	8.75 s.	5.47 s.	8.35 s.	4.96 s.

same framework, Table 4 gives the efficiencies of the implemented algorithms. Recall that the speed - up and the efficiency are defined as follows

$$\text{Speed - up} = \frac{\text{Sequential time}}{\text{Parallel Time}} \text{ and Efficiency} = \frac{\text{Speed - up}}{\text{Number of Processors}}$$

The parallelization of the Schwarz alternating method globally leads to good efficiencies, particularly in the synchronous context when the time step size is small. Indeed, when the number of processors increases, the elapsed time of computation signifiantly decreases, which proves that the parallelization of the Schwarz alternating method is efficient. For example, when eight processors are used, the elapsed time is divided by a factor between four and five. The elapsed time obtained for the synchronous Schwarz alternating method is about one third less than the asynchronous one. This is mainly due to the fact that only additive asynchronous Schwarz alternating method is implemented here. Experimental results also show that the use of parallel synchronous and asynchronous Schwarz alternating method is interesting for the process of large images.

As previously said, we have tested several variants of the Schwarz alternating methods with various values of the time step in order to observe the behaviors of synchronous and asynchronous algorithms, according to the spectral radius of the contraction matrix given by (33). According to Remark 7, when the time step size is small, the spectral radius of the contraction matrix

Table 3 Speed - up with $\tau = 0.1$ for computation of 1024 × 1024 synthetic image

Nb proc.	Semi-implicit		Gear semi-implicit	
	Asynchronous	Synchronous	Asynchronous	Synchronous
1	–	1	–	1
2	0.91	1.26	0.79	1.16
3	1.24	1.86	1.07	1.72
4	1.56	2.48	1.40	2.27
5	1.91	3.04	1.69	2.81
6	2.28	3.66	1.99	3.25
7	2.59	4.26	2.28	3.86
8	2.96	4.73	2.61	4.39

Table 4 Efficiency with $\tau = 0.1$ for computation of 1024×1024 synthetic image

Nb proc.	Semi-implicit		Gear semi-implicit	
	Asynchronous	Synchronous	Asynchronous	Synchronous
1	–	1	–	1
2	0.46	0.63	0.39	0.58
3	0.41	0.62	0.36	0.57
4	0.39	0.62	0.35	0.57
5	0.38	0.61	0.34	0.56
6	0.38	0.61	0.33	0.54
7	0.37	0.61	0.33	0.55
8	0.37	0.59	0.33	0.55

given by (33) is small. In this case all sequential and parallel variants of the Schwarz alternating method converge with a small number of iterations. In such a case, due to fast convergence, the overhead to restart the processors damages the performances of the parallel asynchronous algorithms and the concept of asynchronism loses interest in the case of the considered additive implementation.

When an artificially large value of the time step τ is chosen, due to unconditional stability of the semi-implicit scheme and of the Gear semi-implicit scheme, the medical image is restored in a satisfactory way. Nevertheless, in such case, the spectral radius of the contraction matrix is large, and the parallel synchronous or asynchronous Schwarz alternating methods converge with a larger number of iterations. In this context, the impact of the overhead decreases and synchronous or asynchronous Schwarz alternating methods have comparable performances, even if the synchronous variant is slightly more efficient.

In addition, Gear time discretization scheme leads to matrix with better spectral radius of the contraction matrix than the semi-implicit scheme. The algorithms also converge in fewer iterations. From a general point of view, experiments show that asynchronous algorithms behave more efficiently when the spectral radius of the contraction matrix is worse. In other words, fast convergence does not favour asynchronous iterations, since fewer iterations means less overhead due to synchronization, restart and termination.

Note that, in the case of image processing, the domain Ω is included in the two dimensional space; then each communication or reception consists in transmitting about one thousand real numbers, i.e. eight thousand bytes when the computation are performed in double precision. If the domain is included in the three dimensional space, then the use of asynchronous Schwarz alternating method could reduce more significantly the idle times due to synchronizations and improve the performance of asynchronous Schwarz alternating methods. Consequently, in the two dimensional case the small size of messages is an additional explanation of the cut of performance of the parallel asynchronous Schwarz alternating method compared to the synchronous one (see [9]).

In all cases, at each time step, it is necessary to compute the median of the local coefficient of variation. Due to the overlapping between the subdomains, a sequential handling, particularly a sequential sort, is necessary to keep a global coherency of the informations. It is important to note that the sequential part included in the parallel processing is irreducible and that it affects the performances of the parallel implementation.

For both semi-implicit schemes, particularly when the time step is small, sequential runs need few iterations to converge. Indeed, when $\tau = 0.1$, the sequential Schwarz alternating method needs about 8 iterations at each time step to converge. Few iterations are also necessary to achieve convergence in the parallel runs of the considered subdomain method. Note that the parallel asynchronous iterative implementation of the Schwarz alternating method leads to slightly more iterations than the synchronous implementation. When $\tau = 0.1$, the parallel synchronous Schwarz alternating method needs 10 to 12 iterations at each time step size, while the asynchronous Schwarz alternating method needs 14 to 18 iterations. This experimental result confirms that the parallel asynchronous iterative implementation of the Schwarz alternating method leads to slightly more iterations than the synchronous one. The same behavior can be observed when the time step is large; however, in this case, the number of iterations is larger.

Note also that when the number of processors increases, the efficiency is practically constant in both cases of semi-implicit scheme and Gear semi-implicit scheme.

Finally, note that when the number of processors increases, the efficiencies of synchronous and asynchronous iterative methods have the same variations.

References

1. Alvarez, L., Lions, P.L., Morel, J.M.: Image selective smoothing and edge detection by non linear diffusion. II. SIAM J. Numer. Anal. **29**(3), 845–866 (1992)
2. Aubert, G., Kornprobst, F.: Mathematical Problems in Image Processing—Partial Differential Equations and the Calculus of Variations. Applied Mathematical Sciences, vol. 147, 2nd edn. Springer, New York (2006)
3. Axelson, O.: Iterative Solution Methods. Cambridge University Press, Cambridge (1996)
4. Black, M.J., Sapiro, G., Marimont, D.H., Heeger, D.: Robust anisotropic diffusion. IEEE Trans. Image Proc. **7**(3), 421–432 (1998)
5. Baudet, G.M.: Asynchronous iterative methods for multiprocessor. J. Assoc. Comput. Mach. **2**, 226–244 (1978)
6. Blanc-Feraud, L., Charbonnier, P., Aubert, G., Barlaud, M.: Nonlinear image processing: modeling and fast algorithm for regularization with edge detection. Proc. Int. Conf. Image Process. **1**, 474–477 (1995)
7. Catte, F., Lions, P.L., Morel, J.M., Coll, T.: Image selective smoothing and edge detection by nonlinear diffusion. SIAM J. Numer. Anal. **29**(1), 182–193 (1992)
8. Chazan, D., Miranker, W.: Chaotic relaxation. Linear Algebra Appl. **2**, 199–222 (1969)
9. Chau, M., El Baz, D., Guivarch, R., Spiteri, P.: MPI implementation of parallel subdomain methods for linear and nonlinear convection-diffusion problems. J. Parallel Distrib. Comput. **67**, 581–591 (2007)
10. Dautray, R., Lions, J.L.: Analyse mathématique et calcul numérique pour les sciences et les techniques. tome 9, Masson (1988)

11. El Baz, D., Frommer, A.A., Spiteri, P.: Asynchronous iterations with flexible communications: contracting operators. J. Comput. Appl. Math. **176**, 91–103 (2005)
12. Evans, D.J., Deren, W.: An asynchronous parallel algorithm for solving a class of nonlinear simultaneous equations. Parallel Comput. **17**, 165–180 (1991)
13. Hoffman, K.H., Zou, J.: Parallel efficiency of domain decomposition methods. Parallel Comput. **19**, 1375–1391 (1993)
14. Lions, P.L.: On the schwarz alternating method I. In: Glowinski, R., Golub, G.H., Meurant, G.A., Periaux, J. (eds.) First International Symposium on Domain Decomposition Methods for Partial Differential Equations, pp. 1–42. SIAM, Philadelphia (1988)
15. Miellou, J.C.: Algorithmes de relaxation chaotique à retards. RAIRO **1**, 55–82 (1975)
16. Miellou, J.C., El Baz, D., Spiteri, P.: A new class of iterative algorithms with order intervals. Math. Comput. **67**, 237–255 (1998)
17. Miellou, J.C., Spiteri, P.: Un critère de convergence pour des méthodes générales de point fixe. M2AN **19**(4), 645–669 (1985)
18. Miellou, J.C., Spiteri, P., El Baz, D.: A new stopping criterion for linear perturbed asynchronous iterations. J. Comput. Appl. Math. **219**(2), 471–483 (2008)
19. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. Mach. Intell. **12**(7), 629–639 (1990)
20. Rousseeuw, P.J., Leroy, A.M.: Regression and Outlier Detection. Wiley, New York (1987)
21. Spiteri, P., Miellou, J.C., El Baz, D.: Parallel asynchronous Schwarz and multisplitting methods for a non linear diffusion problem. Numer. Algorithm **33**, 461–474 (2003)
22. Tauber, C., Batatia, H., Ayache, A.: A Robust Speckle Reducing Anisotropic Diffusion. IEEE ICIP **1**, 247–250 (2004)
23. Tukey, J.W.: Explortary Data Analysis. Addison-Wesley, Reading (1977)
24. Weickert, J.: Anisotropic diffusion in image processing. Ph.D. thesis, Dept. of Mathematics, University of Kaiserslautern (1996)
25. Weickert, J., Haar Romeny, B.M., Viergever, M.A.: Efficient and reliable schemes for nonlinear diffusion filtering. IEEE Trans. Image Process. **7**(3), 398–410 (1998)
26. Yu, Y., Acton, S.T.: Speckle reducing anisotropic diffusion. IEEE Trans. Image Process. **11**(11), 1260–1270 (2002)