Parallel solution of the discretized and linearized G-heat equation

Pierre Spitéri, Amar Ouaoua, Ming Chau, Hacène Boutabia

Parallel solution of the discretised and linearised G-heat equation

Pierre Spiteri*

ENSEEIHT-IRIT,
2 rue Charles Camichel,
31071 Toulouse CEDEX, France
Email: pierre.spiteri@enseeiht.fr
*Corresponding author

Amar Ouaoua

Laboratoire LaPS,
Badji Mokhtar University,
P.O. Box 12, 23 000 Annaba, Algeria
Email: ouaoua_amar@yahoo.fr

Ming Chau

Advanced Solutions Accelerator,
199 rue de l'Oppidum,
34170 Castelnau le Lez, France
Email: mingchau.1@gmail.com

Hacène Boutabia

Laboratoire LaPS,
Badji Mokhtar University,
P.O. Box 12, 23 000 Annaba, Algeria
Email: hboutabia@hotmail.com

- **Abstract:** The present study deals with the numerical solution of the G-heat equation. Since the G-heat equation is defined in an unbounded domain, we firstly state that the solution of the G-heat equation defined in a bounded domain converges to the solution of the G-heat equation when the measure of the domain tends to infinity. Moreover, after time discretisation by an implicit time marching scheme, we define a method of linearisation of each stationary problem, which leads to the solution of a large scale algebraic system. A unified approach analysis of the convergence of the sequential and parallel relaxation methods is given. Finally, we present the results of numerical experiments.

- **Keywords:** G-heat equation; relaxation methods; parallel computing; asynchronous iteration; financial application.

- **Biographical notes:** Pierre Spiteri graduated in Mathematics at the University of Franche-Comté, Besançon, France in 1968. He received his Doctor degree in 1974 and his Docteur d'Etat es Sciences Mathématiques degree from the same university in 1984. He is now a Full Professor at ENSEEIHT in the Department of Applied Mathematics and Computer Sciences, in the Institut National Polytechnique of Toulouse (University of Toulouse). He teaches numerical analysis, optimisation and numerical solution of boundary value problems. His fields of interest are in numerical analysis, large scale nonlinear systems of evolution equations, optimal control, parallel computing and more particularly, domain decomposition methods for the solution of nonlinear boundary values problems; he is interested to apply the obtained theoretical results to applications concerning finance, image processing, mechanics, etc. He is also a scientific expert, advisor and referee for several international scientific committees and journals.

Amar Ouaoua is a PhD student. He received his Magister degree in Mathematical Sciences from Badji Mokhtar University, Annaba, Algeria, in 2010. His main research interests include stochastic calculus, the theory of Markov's processes and dynamical systems.

Ming Chau is an Engineer and Researcher in Applied Mathematics and Computer Science. He graduated with an Engineer degree in Computer Sciences and Applied Mathematics from ENSEEIHT Toulouse, France in 2001. He currently works at Advanced Solutions Accelerator, a French IT firm specialised in scientific software and hardware. He received his PhD degree in 2005 from the Institut National Polytechnique of Toulouse (INP Toulouse – University of Toulouse), France. He also received his Habilitation à Diriger des Recherches degree in Computer Sciences from INP Toulouse in 2015. His research interests include asynchronous algorithms, image processing and numerical simulation, with applications to biomechanical engineering and medical imaging.

Hacène Boutabia received his Doctoral degree (3ième cycle) in Pure Mathematics from the Scientific and Medical University of Grenoble (France) and his PhD status from the University of Annaba (Algeria), where he was promoted to a Professor holder in 2006. He is the Director of the Laboratory of Probability and Statistics created in 2012 and supervises many students in both Master and PhD. His main fields of research include: stochastic calculus, particle systems and random matrices. He is also a scientific expert and advisor for several national scientific committees and agencies, and was Head of the Department of Mathematics during the 2003–2009 period.

1 Introduction

Many modelled phenomena by the Hamilton-Jacobi-Bellman equation are the results of the method of dynamic programming initiated by Richard Bellman to solve optimisation problems, i.e., problems where the best possible decision for each given date performance criterion is made.

The equation of dynamic programming generalises previous work in classical mechanics. Historically applied in engineering and other areas of applied mathematics, the Hamilton-Jacobi-Bellman equation has become an important tool in decision making problems involving economics and financial markets (see Dong et al., 2015; Wan et al., 2006).

Motivated by uncertainty problems about the stochastic volatility, risk measures in finance, Peng (2005, 2010) has introduced a new notion of nonlinear expectation space; the so-called *G*-expectation which it can take the uncertainty into consideration. The *G*-expectation has been developed very recently and opened the way to the introduction of *G*-normal random variables under the framework of *G*-expectation space (Peng, 2010). The main difficulty lies in the fact that the *G*-expectation is intrinsic in the sense that it is not based on a given linear probability space. The *G*-heat equation defined by Peng (2009) is a nonlinear equation related to the *G*-normal distribution and generalises the classical normal distribution. This equation, which is a special kind of Hamilton-Jacobi-Bellman equation (Crandall et al., 1992) has a unique viscosity solution (Peng, 2005). The existence and the uniqueness of the solution of the *G*-heat equation in the sense of viscosity solution can be found for example in Peng (1992) and Yong and Zhou (1999)

Mingshang (2009) obtained the explicit viscosity solution of the G-heat equation [see equation (1) below] with the initial condition $\phi(x) = x^k$ for each integer $k \geq 1$.

1.1 Background

The theory of *G*-normal distribution, *G*-expectation (a kind of nonlinear expectation), the one-dimensional *G*-Brownian motion and the associated stochastic calculus is introduced in Peng (2010). Unlike the classical normal distribution $\mathcal{N}(0, \sigma^2)$, characterised by the linear heat equation:

$$\frac{\partial u}{\partial t} = \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial x^2}, \quad (t, x) \in (0,T)\times\mathbb{R}, \; T > 0,$$

the *G*-heat equation satisfied by the *G*-normal distribution $\mathcal{N}(0, [\underline{\sigma}^2, \bar{\sigma}^2])$ with $G(\alpha) := \frac{1}{2}[\alpha^+\bar{\sigma}^2 - \alpha^-\underline{\sigma}^2]$, is associated to the nonlinear heat equation:

$$\frac{\partial u}{\partial t} = G\left(\frac{\partial^2 u}{\partial x^2}\right), \quad (t, x) \in (0,T)\times\mathbb{R}.$$

In the sequel, we will extend this kind of boundary value problem to the *d*-dimensional space.

1.2 Preliminaries definitions

The aim of this section is to recall some basic definitions and properties of *G*-expectations, *G*-heat equation and *G*-Brownian motions, that will be needed subsequently. For a more detailed description of these notions, the reader is referred to Peng (2005). Adapting the approach in Peng (2005, 2010), let Ω be a given non-empty fundamental space and $\mathcal{H}$ be a linear space of real functions defined on Ω such that:

- $1 \in \mathcal{H}$
- $\mathcal{H}$ is stable with respect to local Lipschitz functions, i.e., for all $d \geq 1$, and for all $X_1, ..., X_d \in \mathcal{H}$, $\phi \in C_{l,Lip}(\mathbb{R}^d)$, it holds also $\phi(X_1, ..., X_d) \in \mathcal{H}$.

Recall that $C_{l,Lip}(\mathbb{R}^d)$ denotes the space of all local Lipschitz functions ϕ: $\mathbb{R}^d \to \mathbb{R}$ satisfying

$$|\phi(x)-\phi(y)| \le C\left(1+|x|^m+|y|^m\right)|x-y|, \quad x, y \in \mathbb{R}^d,$$

for some $C > 0$, $m \in \mathbb{N}$ depending on ϕ. The set $\mathcal{H}$ is interpreted as the space of random variables defined on Ω.

Definition 1: A sublinear expectation $\mathbb{E}$ on $\mathcal{H}$ is a functional $\mathbb{E}$: $\mathcal{H} \to \mathbb{R}$ with the following properties: for all X, $Y \in \mathcal{H}$, we have

- *monotonicity*: if $X \ge Y$, then $\mathbb{E}[X] \ge \mathbb{E}[Y]$
- *preservation of constants*: $\mathbb{E}[c] = c$, for all $c \in \mathbb{R}$
- *sub-additivity*: $\mathbb{E}[X] - \mathbb{E}[Y] \le \mathbb{E}[X - Y]$
- *positive homogeneity*: $\mathbb{E}[\lambda X] = \lambda\mathbb{E}[X]$, for all $\lambda \ge 0$.

The triple $(\Omega, \mathcal{H}, \mathbb{E})$ is called a sublinear expectation space.

Definition 2: A random variable $Y \in \mathcal{H}^{d_1}$ is said to be independent of $X \in \mathcal{H}^{d_2}$ under $\mathbb{E}$ if for each test function $\phi \in C_{l,Lip}(\mathbb{R}^{d_1+d_2})$ *we have* $\mathbb{E}[\phi(X, Y)] = \mathbb{E}[\mathbb{E}[\phi(x, Y)]_{x=X}]$.

Definition 3: Given two sublinear expectation spaces $(\Omega, \mathcal{H}, \mathbb{E})$ and $(\Omega_1, \mathcal{H}_1, \mathbb{E}_1)$, two d-dimensional random vectors $X \in \mathcal{H}^d$ and $X_1 \in \mathcal{H}_1^d$ are said to be identically distributed ($X \sim X_1$) if for each test function $\phi \in C_{l,Lip}(\mathbb{R}^d)$ we have $\mathbb{E}[\phi(X)] = \mathbb{E}_1[\phi(X_1)]$.

This theory related to the G-heat equation has been motivated by the earlier studies of *G*-expectation and conditional *G*-expectation defined with the help of a backward stochastic differential equation and by describing coherent and dynamic risk measures. The *G*-expectation can be regarded as a coherent risk measure and the conditional *G*-expectation can be regarded as a dynamic risk measure. Peng (2008) extended his former studies to the multidimensional *G*-Brownian motion; after having introduced a multidimensional *G*-normal distributions via the nonlinear heat equation, called the *G*-heat equation:

$$\frac{\partial u}{\partial t} = G\left(D^2u\right), \quad (t, x) \in (0,T)\times\mathbb{R}^d,$$

where D^2u is the Hessian matrix of u, i.e., $D^2u = \left(\frac{\partial^2 u}{\partial x_i \partial x_j}\right)$ and $G(D) = \frac{1}{2}\sup_{B\in\Gamma} tr(DB)$, $D = (D_{ij})_{i,j=1}^d \in \mathbb{S}_d$, where $\mathbb{S}_d$ denotes the space of $d \times d$ symmetric matrices; note that $\mathbb{S}_d$ is a $\frac{d(d+1)}{2}$-dimensional Euclidean space of the inner product (D, B): $= tr[DB]$. Γ is a given non-empty convex, bounded and closed subset of the space of non-negative $d \times d$ matrices. Peng defines the nonlinear *G*-expectation $\hat{\mathbb{E}}$ on $C_0([0, T]; \mathbb{R}^d)$ under which the coordinate process B is a process which future increments are independent of the past ones and *G*-normally distributed. Such a process is called *G*-Brownian motion. Peng (2008) developed then the related stochastic calculus. In particular, he has introduced the stochastic integral with respect to a *G*-Brownian motion, establishes an Itô formula and study stochastic differential equations driven by a *G*-Brownian motion. The function G: $\mathbb{S}_d \mapsto \mathbb{R}$ defined by:

$$G(D) = \frac{1}{2}\sup_{B\in\Gamma} tr(DB)$$

is a monotonic and sublinear mapping, that is:

a $D \ge B \Rightarrow G(D) \ge G(B)$

b $G(\lambda D) = \lambda^+ G(D) + \lambda^- G(-D)$

c $G(D + B) \le G(D) + G(B)$.

Now we start introducing the definition of *G*-Brownian motion by considering the following parabolic partial differential equation, called the *G*-heat equation:

$$\begin{cases} \dfrac{\partial u}{\partial t} = G\left(D^2u\right), \quad (t, x) \in (0,T)\times\mathbb{R}^d \\ u|_{t=0} = \phi \end{cases}, \tag{1}$$

Definition 4: Let $(B_t)_{t\ge0}$ be a d-dimensional process. $(B_t)_{t\ge0}$ is called *G*-Brownian motion under a sublinear expectation $(\Omega, \mathcal{H}, \mathbb{E})$ if $B_0 = 0$ and

1 for each $t \ge s$, $t \ge 0$ and $\phi \in C_{l,Lip}(\mathbb{R}^d)$, $B_t \sim (B_{t+s} - B_s)$ and $u(t, x)$ defined by $\mathbb{E}(\phi(x + B_t))$ is the unique viscosity solution of (1)

2 for each $m = 1, 2, \ldots, 0 = t_0 < t_1 < \ldots < t_m < \infty$ the increment $B_{t_m} - B_{t_{m-1}}$ is independent to $(B_{t_1}, \cdots, B_{t_{m-1}})$.

Following Peng (2009), the property "$u(t, x)$ defined by $\mathbb{E}(\phi(x + B_t))$ is the unique viscosity solution of (1)" means that B_t is a d-dimensional *G*-normal distribution $\mathcal{N}(0, t\Gamma)$.

Remark 1: The 1-dimensional case which corresponds to $d = 1$ and $\Gamma = [\underline{\sigma}^2, \overline{\sigma}^2] \subset \mathbb{R}$, the nonlinear heat equation (1) becomes:

$$\begin{cases} \dfrac{\partial u}{\partial t} = G\left(\dfrac{\partial^2 u}{\partial x^2}\right), \quad (t, x) \in (0,T)\times\mathbb{R} \\ u|_{t=0} = \phi \in C_{l,Lip}(\mathbb{R}) \end{cases},$$

with $G(\alpha) := \frac{1}{2}\left[\alpha^+\overline{\sigma}^2 - \alpha^-\underline{\sigma}^2\right]$. In the multidimensional cases we also have the following typical nonlinear heat equation:

$$\begin{cases} \dfrac{\partial u}{\partial t} = \sum_{i=1}^{d} G_i\left(\dfrac{\partial^2 u}{\partial x_i^2}\right), \quad (t,\ x) \in (0,T)\times\mathbb{R}^d \\ u\,|_{t=0} = \phi \in C_{l,Lip}(\mathbb{R}^d) \end{cases}, \tag{2}$$

where $G_i(\alpha) := \frac{1}{2}\,[\alpha^+ \overline{\sigma_i}^2 - \alpha^- \underline{\sigma_i}^2]$ and $0 \le \underline{\sigma_i} \le \overline{\sigma_i}$ are given constants. This corresponds to:

$$\Gamma \subset \left\{ diag[\gamma_1, \gamma_2, ..., \gamma_d]:\ \gamma_i \in \left[\underline{\sigma_i}^2, \overline{\sigma_i}^2\right],\ i = 1,2,...,d \right\}.$$

Note that $x + B_t$ is $\mathcal{N}(x, t\Gamma)$-distributed. In the sequel we will consider the general case of equation (2).

1.3 The unbounded case

For the numerical solution of the typical nonlinear heat equation, the following original result is necessary. Note that this result generalises the result of Jaillet et al. (1990) concerning the Black-Scholes equation.

Let $T_k = \inf\{s \in [0,\ T]:\ |B_s| > k\}$ and $u_k(t,x) = \mathbb{E}(\phi(x + B_{t\wedge T_k}))$, where $\phi \in C_{l,Lip}(\mathbb{R}^d)$ satisfying

$$\left|\phi(x) - \phi(y)\right| \le C\left(1 + |\,x\,|^m + |\,y\,|^m\right)|\,x - y\,|, \quad x, y \in \mathbb{R}^d,$$

$m \in \mathbb{N}$ depending on ϕ. Let $\mathcal{B}_R = \{x \in \mathbb{R}^d:\ |x| \le R\}$. The following lemma holds

Lemma 1: u_k converges to u (uniformly on compact sets) as k tends to infinity; more precisely, for all $R > 0$

$$\lim_{k\to\infty} \sup_{(t,x)\in[0,T]\times\mathcal{B}_R} |\,u(t,x) - u_k(t,x)\,| = 0.$$

Proof: We have

$$|\,u(t,x) - u_k(t,x)\,| \le \mathbb{E}\left(\left|\phi\left(x + B_t\right) - \phi\left(x + B_{t\wedge T_k}\right)\right|\mathbf{1}_{\{T_k<T\}}\right),$$

where $a \wedge b = \min(a, b)$. By using the property of ϕ, we obtain that

$$\begin{aligned} &\left|\phi\left(x + B_{t\wedge T_k}\right) - \phi\left(x + B_t\right)\right| \\ &\le C\left|B_t - B_{t\wedge T_k}\right|\left(1 + \left|x + B_t\right|^m + \left|x + B_{t\wedge T_k}\right|^m\right) \\ &\le C\left(\left|B_t\right| + \left|B_{t\wedge T_k}\right|\right)\left(1 + \left|x + B_t\right|^m + \left|x + B_{t\wedge T_k}\right|^m\right), \end{aligned}$$

and so (with $C' = 6C$), by using the Holder inequality under $\mathbb{E}$ and the fact that $(a + b)^p \le 2^p\,(a^p + b^p)$ and $(a + b + c)^p \le 3^p\,(a^p + b^p + c^p)$ for $a, b, c \ge 0$ and $p \ge 1$ we have

$$\begin{aligned} &|\,u(t,x) - u_k(t,x)\,| \\ &\le C'\mathbb{E}^{\frac{1}{2}}\left(\left(\left|B_t\right|^2 + \left|B_{t\wedge T_k}\right|^2\right)\left(1 + \left(|\,x\,| + \left|B_t\right|\right)^{2m} + \left(|\,x\,| + \left|B_{t\wedge T_k}\right|\right)^{2m}\right)\right) \\ &\times \mathbb{E}^{\frac{1}{2}}\left(\mathbf{1}_{\{T_k<T\}}\right). \end{aligned}$$

The same arguments implies that:

$$\begin{aligned} &\mathbb{E}\left(\left(\left|B_t\right|^2 + \left|B_{t\wedge T_k}\right|^2\right)\left(1 + \left(|\,x\,| + \left|B_t\right|\right)^{2m} + \left(|\,x\,| + \left|B_{t\wedge T_k}\right|\right)^{2m}\right)\right) \\ &\le \mathbb{E}^{\frac{1}{2}}\left(4\left(\left|B_t\right|^4 + \left|B_{t\wedge T_k}\right|^4\right)\right) \\ &\times \mathbb{E}^{\frac{1}{2}}\left(9\left(1 + \left(|\,x\,| + \left|B_t\right|\right)^{4m} + \left(|\,x\,| + \left|B_{t\wedge T_k}\right|\right)^{4m}\right)\right) \\ &\le 6\left(2\sup_{s\in[0,T]}\mathbb{E}\left(\left|B_s\right|^4\right)\right)^{\frac{1}{2}} \\ &\times \mathbb{E}^{\frac{1}{2}}\left(1 + 2^{4m}\left(|\,x\,|^{4m} + \left|B_t\right|^{4m}\right) + 2^{4m}\left(|\,x\,|^{4m} + \left|B_{t\wedge T_k}\right|^{4m}\right)\right) \\ &\le 6\left(2\sup_{s\in[0,T]}\mathbb{E}\left(\left|B_s\right|^4\right)\right)^{\frac{1}{2}} \\ &\times \left(1 + 2^{4m+1}\left(R^{4m} + \sup_{s\in[0,T]}\mathbb{E}\left(\left|B_s\right|^{4m}\right)\right)\right)^{\frac{1}{2}} \end{aligned}$$

Let

$$\begin{aligned} &C(R,\ T,\ m, d) \\ &= 6\left(2\sup_{s\in[0,T]}\mathbb{E}\left(\left|B_s\right|^4\right)\right)^{\frac{1}{2}}\left(1 + 2^{4m+1}\left(R^{4m} + \sup_{s\in[0,T]}\mathbb{E}\left(\left|B_s\right|^{4m}\right)\right)\right)^{\frac{1}{2}} \end{aligned}$$

Note that $C\,(R,\ T,\ m,\ d)$ is a constant depending only on T, R, m, d. We have then

$$|\,u(t,x) - u_k(t,x)\,| \le C'C^{\frac{1}{2}}(R,\ T,\ m, d)\ \mathbb{E}^{\frac{1}{2}}\left(\mathbf{1}_{\{T_k<T\}}\right).$$

Now observe that since for all $i = 1, 2, ..., d$, B_s^i is $\mathcal{N}(0, [\underline{\sigma_i}^2 s, \overline{\sigma_i}^2 s])$ -distributed, then following Peng (2010) $\mathbb{E}(|\,B_s^i\,|^4) = \frac{12}{\sqrt{2\pi}}\overline{\sigma_i}^4 s^2 \le \frac{12}{\sqrt{2\pi}}\overline{\sigma_i}^4 T^2$ and $\mathbb{E}(|\,B_s^i\,|^{4m}) = \frac{2(4m-1)!}{\sqrt{2\pi}}\overline{\sigma_i}^{4m} s^{2m} \le \frac{2(4m-1)!}{\sqrt{2\pi}}\overline{\sigma_i}^{4m} T^{2m}$. We can easily see that $C\,(R,\ T,\ m,\ d) < +\infty$. In order to complete the proof, we need to prove that:

$$\lim_{k\to\infty}\sup_{(t,x)\in[0,T]\times\mathcal{B}_R}\mathbb{E}\left(\mathbf{1}_{\{T_k<T\}}\right) = 0. \tag{3}$$

Recall that there exists a family of probability measures $\mathcal{P}$ on $(\Omega, \mathcal{H})$ such that for each random variable X

$$\mathbb{E}(\mathbf{X}) = \sup_{P\in\mathcal{P}}\mathbb{E}_P(\mathbf{X}),$$

where $\mathbb{E}_P$ is the linear expectation under P (for more details, see Peng, 2008, 2009). Since

$$\left\{T_k < T\right\} = \left\{\sup_{s\in[0,T]}\left|B_s\right| > k\right\} \subset \bigcup_{i=1}^{d}\left\{\sup_{s\in[0,T]}\left|B_s^i\right| > \frac{k}{\sqrt{d}}\right\},$$

then

$$\mathbb{E}\left(\mathbf{1}_{\{T_k<T\}}\right) \le \sum_{i=1}^{d} \mathbb{E}\left(\mathbf{1}_{\left\{\sup_{s\in[0,T]}\left|B_s^i\right|>\frac{k}{\sqrt{d}}\right\}}\right).$$

By using Markov inequality under each probability measure $P \in \mathcal{P}$ we obtain

$$P\left(\sup_{s\in[0,T]}\left|B_s^i\right| > \frac{k}{\sqrt{d}}\right) \le \frac{\mathbb{E}_P\left(\sup_{s\in[0,T]}\left|B_s^i\right|\right)}{\frac{k}{\sqrt{d}}} \le \frac{\sqrt{d}\,\mathbb{E}\left(\sup_{s\in[0,T]}\left|B_s^i\right|\right)}{k},$$

and so

$$\mathbb{E}\left(\mathbf{1}_{\left\{\sup_{s\in[0,T]}\left|B_s^i\right|>\frac{k}{\sqrt{d}}\right\}}\right) \le \frac{\sqrt{d}\,\mathbb{E}\left(\sup_{s\in[0,T]}\left|B_s^i\right|\right)}{k}.$$

Let $\varepsilon > 0$, then there exists $s_\varepsilon \in [0, T]$ such that

$$\sup_{s\in[0,T]}\left|B_s^i\right| \le \left|B_{s_\varepsilon}^i\right| + \varepsilon,$$

and

$$\mathbb{E}\left(\sup_{s\in[0,T]}\left|B_s^i\right|\right) \le \mathbb{E}\left(\left|B_{s_\varepsilon}^i\right|\right) + \varepsilon \le \overline{\sigma}_i\sqrt{s_\varepsilon} + \varepsilon \le \overline{\sigma}_i\sqrt{T} + \varepsilon,$$

which implies that

$$\mathbb{E}\left(\sup_{s\in[0,T]}\left|B_s^i\right|\right) \le \overline{\sigma}_i\sqrt{T},$$

and

$$\mathbb{E}\left(\mathbf{1}_{\{T_k<T\}}\right) \le \frac{\sqrt{dT}}{k}\sum_{i=1}^{d}\overline{\sigma_i};$$

then (3) is true and the proof is complete. □

1.4 Objectives of the present study

The goal of the present study is to solve by various numerical methods equation (1). Nevertheless, the numerical solution of the previous problem brings up many questions. The first question is related to the fact that the previous boundary value problem is not defined in a bounded domain, but is defined in the unbounded domain $\mathbb{R}^d$, $d \ge 1$. This difficulty is solved by considering the problem defined on a bounded domain $\Omega \subset \mathbb{R}^d$, and, thanks to the result stated in Lemma 1, we know that the solution of the time dependent problem defined on the bounded domain Ω converges to the solution of the problem (1) when the measure of Ω tends to infinity. The second question concerns the effective solution of the algebraic systems derived from the discretisation process of the continuous problem to be solved. Indeed, for such discretisation we consider on one hand the temporal discretisation and on the other hand the spatial discretisation. Owing to the stability condition to be satisfied by the classical explicit time marching scheme, the temporal discretisation is usually achieved by considering implicit or semi-implicit time marching schemes; this previous schemes lead to solve at each time step, a stationary nonlinear boundary value problem. After spatial discretisation we have then to solve large-scale algebraic systems. The third question is related to the fact that the problem (1) is a nonlinear problem; so, by considering an analogous way to the one considered for the solution of the obstacle problem (Spiteri et al., 2001), we linearise the G-heat equation by a complementary formulation of this previous problem and so, at each time step, we have to solve such linearised algebraic system obtained by the Howard process. Taking into account on one hand the size of the algebraic systems of the linearised and discretised G-heat equation and on the other hand the sparsity of the discretisation matrices, sequential iterative methods, like the point Jacobi or the point Gauss-Seidel method, and more generally the parallel iterative relaxation algorithms are well adapted to the solution of the considered problem. Therefore, for the use of the parallel iterative relaxation methods, we concentrate in this paper to the subdomain methods defined by gathering several adjacent blocks of the discretisation matrices; then, in an unified presentation, we consider in the sequel sequential and more generally, parallel synchronous or asynchronous iterative subdomain methods well suited for solving these algebraic systems. Recall that the asynchronous relaxation method corresponds to a general scheme of computation where the computations are performed in parallel without order nor synchronisation among the processors (Baudet, 1978). Then, when a large number of processors are used, idle times due to the synchronisations among the processors are suppressed; consequently, in this case, since the synchronisations are time consuming, the elapsed time of computation decreases when parallel asynchronous relaxation algorithms are performed. So we consider two kinds of such parallel algorithms: on one hand, the parallel synchronous or asynchronous block relaxation algorithm, corresponding in fact to a subdomain method without overlapping between the subdomains and on the other hand the parallel synchronous or asynchronous Schwarz alternating method in which the subdomains overlap each other. For both previous mathematical formulations of the sequential and parallel methods, the convergence of the algorithms applied to the solution of the discretised and linearised problem, can be proved by contraction techniques (Giraud and Spiteri, 1991; Miellou and Spiteri, 1985) for every splitting of the problem to solve or by partial ordering techniques (Miellou et al., 1998; Spiteri et al., 2001). Indeed, a main property, ensuring the convergence of such sequential or parallel synchronous or asynchronous iterative method, is related to the fact that after appropriate discretisation, the spatial part of the operator leads to a discretisation matrix which is an M-matrix (Ortega and

Rheinboldt, 1970); moreover, note that the use of an implicit or semi-implicit time marching scheme preserves this last property. For more details, the reader is referred to Giraud and Spiteri (1991), Miellou et al. (1998), Miellou and Spiteri (1985), and Spiteri et al. (2001).

Implementation of the considered algorithms is carried out on HPC@LR, a supercomputing centre, and on Grid5000, a grid computing platform. These two architectures allow us to study the behaviour of parallel algorithms. The studied algorithms are parallelised with MPI facilities (see Chau et al., 2007). Asynchronous and synchronous efficiency of the studied parallel algorithms can be compared.

The present study is organised as follows. In Section 2 we give the formulations, on one hand of the appropriate discretisation schemes and on the other hand of the linearisation process. Then, we recall and present in an unified way the sequential and parallel synchronous and asynchronous algorithms for the solution of the G-heat equation and we show that the appropriate discretisation schemes considered ensure the convergence of the sequential and both parallel synchronous or asynchronous subdomain methods. Section 3 is devoted to the presentation of the sequential and parallel experiments. Lastly in Section 4 we give some conclusion and perspectives.

2 Numerical solution of the general G-heat equation

2.1 Preliminaries

Let $d \in \mathbb{N}$ and $\Omega \subset \mathbb{R}^d$, be a bounded domain. Let us also denote by $\partial\Omega$ the boundary of Ω. In a context adapted to a numerical solution of the G-heat equation, we can consider this previous problem equipped with boundary conditions

$$\begin{cases} \dfrac{\partial u}{\partial t} - \displaystyle\sum_{i=1}^{d} G_i\left(\frac{\partial^2 u}{\partial x_i^2}\right) = 0, \text{ e.w. in } [0, T] \times \Omega, \\ u\,|_{t=0} = \phi, \\ B.C. \text{ on } u \text{ defined on } \partial\Omega. \end{cases} \tag{4}$$

where ϕ is the initial condition, T is the final time, B.C. describes the boundary conditions on the boundary $\partial\Omega$ and e.w. means everywhere. Since $\alpha^+ = \max(\alpha, 0)$ and $\alpha^- = \max(-\alpha, 0)$, in what follows, we consider a more appropriate writing of the equation (4) specified below

$$\begin{cases} \dfrac{\partial u}{\partial t} - \dfrac{1}{2}\displaystyle\sum_{i=1}^{d}\left(\overline{\sigma}_i^2 . \max\left(\frac{\partial^2 u}{\partial x_i^2}, 0\right) - \underline{\sigma}_i^2 . \max\left(-\frac{\partial^2 u}{\partial x_i^2}, 0\right)\right) \\ = 0, \text{ e.w. in } [0, T] \times \Omega, \\ u\,|_{t=0} = \phi, \\ B.C. \text{ on } u \text{ defined on } \partial\Omega. \end{cases} \tag{5}$$

For example, if the Dirichlet homogeneous boundary conditions are considered on the boundary $\partial\Omega$, then (4) is written as follows

$$\begin{cases} \dfrac{\partial u}{\partial t} - \dfrac{1}{2}\displaystyle\sum_{i=1}^{d}\left(\overline{\sigma}_i^2 . \max\left(\frac{\partial^2 u}{\partial x_i^2}, 0\right) - \underline{\sigma}_i^2 . \max\left(-\frac{\partial^2 u}{\partial x_i^2}, 0\right)\right) \\ = 0, \text{ e.w. in } [0, T] \times \Omega, \\ u\,|_{t=0} = \phi, \\ u = 0 \text{ on } \partial\Omega, \end{cases} \tag{6}$$

while if the Neumann homogeneous boundary conditions are considered on the boundary $\partial\Omega$, then (4) is written as follows

$$\begin{cases} \dfrac{\partial u}{\partial t} - \dfrac{1}{2}\displaystyle\sum_{i=1}^{d}\left(\overline{\sigma}_i^2 . \max\left(\frac{\partial^2 u}{\partial x_i^2}, 0\right) - \underline{\sigma}_i^2 . \max\left(-\frac{\partial^2 u}{\partial x_i^2}, 0\right)\right) \\ = 0, \text{ e.w. in } [0, T] \times \Omega, \\ u\,|_{t=0} = \phi, \\ \dfrac{\partial u}{\partial n} = 0 \text{ on } \partial\Omega, \end{cases} \tag{7}$$

where classically $\frac{\partial u}{\partial n}$ denotes the normal derivative. Note also that other classical boundary conditions such as the Robin or the mixed boundary conditions can be considered. In the sequel, we consider the analysis of equation (6) equipped with Dirichlet boundary conditions; nevertheless this analysis is still valid for other boundary conditions.

Remark 2: In fact, for all every boundary conditions considered, the numerical solution can be performed without difficulty by the same way after appropriate discretisation of the continuous problem.

Actually, the initial problem (1) is defined in an unbounded domain; then by applying the result of Lemma 1 we have to consider an analogous problem defined now on a bounded domain Ω such that the measure of Ω tends to infinity; then we have to consider one of the previous equations (5) to (7) for the numerical solution of the G-heat equation. Classically the numerical solution of the G-heat equation needs on one hand the discretisation of the partial derivatives arising in the considered problem and on the other hand the linearisation and also, after linearisation, the solution of the obtained algebraic linear systems.

2.2 Discretisation.

In order to discretise the G-heat equation we consider:

1 for the temporal part of the evolution equation, an implicit or a semi-implicit time marching scheme such as, for example, the classical implicit time marching scheme or the Crank-Nicholson semi-implicit scheme; thus, at each time step, after appropriate spatial discretisation, we have to solve an algebraic large scale system. In the sequel we will denote by δ_t the discretisation time step,

2 for the spatial part of the equation, since we have to discretise the second derivative with respect to the spatial variable, we consider the following classical approximation for all $l \in \{1, \ldots, N_i\}$

$$\frac{\partial^2 u}{\partial x_i^2}(\ldots, x_l, \ldots, t) \approx \frac{\underline{u}_l - 2\hat{u}_l + \overline{u}_l}{h_i^2} + \mathcal{O}\left(h_i^2\right)$$

where

$$\underline{u}_l \approx u\left(\ldots, x_j, x_l - h_i, x_k, \ldots, t\right)$$
$$\hat{u}_l \approx u\left(\ldots, x_l, \ldots\right)$$
$$\overline{u}_l \approx u\left(\ldots, x_j, x_l + h_i, x_k, \ldots, t\right)$$

where N_i is the number of mesh points along the i^{th}-axis, h_i is the spatial discretisation step of the i^{th} coordinate. Let us denote now by $u^n_{\ldots,j,l,k,\ldots}$ the approximation by finite difference method of $u(\ldots, x_j, x_l, x_k, \ldots, t_n)$; so, the total discretisation of the G-heat equation leads to solve the following nonlinear algebraic system

$$\frac{u^{n+1}_{\ldots,j,l,k,\ldots} - u^n_{\ldots,j,l,k,\ldots}}{\delta_t} - \frac{1}{2}\sum_{i=1}^{d}\left(\overline{\sigma}_i^2.\max\left(\frac{u^{n+1}_{\ldots,j,l+1,k,\ldots} - 2.u^{n+1}_{\ldots,j,l,k,\ldots} + u^{n+1}_{\ldots,j,l-1,k,\ldots}}{h_i^2}, 0\right) - \underline{\sigma}_i^2.\max\left(-\frac{u^{n+1}_{\ldots,j,l+1,k,\ldots} - 2.u^{n+1}_{\ldots,j,l,k,\ldots} + u^{n+1}_{\ldots,j,l-1,k,\ldots}}{h_i^2}, 0\right)\right) = 0;$$

let

$$\alpha_{j,l,k} = \frac{u^{n+1}_{\ldots,j,l+1,k,\ldots} - 2.u^{n+1}_{\ldots,j,l,k,\ldots} + u^{n+1}_{\ldots,j,l-1,k,\ldots}}{h_i^2},$$

then, using this previous notation, any component of the previous algebraic system can be written as follows

$$u^{n+1}_{\ldots,j,l,k,\ldots} - \frac{\delta_t}{2}\sum_{i=1}^{d}\left(\overline{\sigma}_i^2.\max\left(\alpha_{j,l,k}, 0\right) - \underline{\sigma}_i^2.\max\left(-\alpha_{j,l,k}, 0\right)\right) = u^n_{\ldots,j,l,k,\ldots}, \ \forall n \in \mathbb{N}. \quad (8)$$

2.3 Linearisation by the Howard process

Gong and Yang (2013) have solved the G-heat equation by considering the linearisation of the problem by using the Newton method. Nevertheless, it seems that, due to the formulation of the problem to solve, the discrete operator associated to the G-heat equation can not be always derived, particularly at the origin; consequently, the Newton method seems difficult to apply rigorously. In the present section, we propose a linearisation process using the complementary formulation of the problem to solve (see Spiteri et al., 2001). So, we can use a linearisation process similar to the one used when we have to solve the obstacle problem arising in finance or in mechanics or more generally when we consider the solution of the Hamilton-Jacobi-Bellman equation arising for example in image processing (Giraud and Spiteri, 1991). This linearisation constitutes an extension of the Howard process usually used for the linearisation of complementary problems. In fact, this method of linearisation is a Newton-like method for the solution of the stationary problem associated with the G-heat equation. Owing to the properties of the problem to solve, the previous discretisation scheme leads at each time step to the solution of a large algebraic system.

Preliminarily, we can state the following simple results:

Lemma 2: Consider the mathematical expression $\beta = -\max(-\alpha, 0)$. Then $\beta = \min(\alpha, 0)$.

Proof: Indeed, if $-\alpha < 0$, then, $\max(-\alpha, 0) = 0$ and $-\max(-\alpha, 0) = 0$; but $\alpha > 0$ involves that $\min(\alpha, 0) = 0 = -\max(-\alpha, 0)$. On the contrary, if $-\alpha > 0$, then $\max(-\alpha, 0) = -\alpha$ and $-\max(-\alpha, 0) = \alpha$; but $\alpha < 0$ involves that $\min(\alpha, 0) = \alpha = -\max(-\alpha, 0)$, and the proof is achieved. □

Lemma 3: Consider the mathematical expression $\theta = \max(\alpha, 0) - \max(-\alpha, 0)$. *Then* $\theta = \max(\alpha, 0) + \min(\alpha, 0)$.

Proof: Indeed, if $\alpha > 0$, then $\max(\alpha, 0) = \alpha$ and $-\max(-\alpha, 0) = \min(\alpha, 0) = 0$. On the contrary, if $\alpha < 0$, then $\max(\alpha, 0) = 0$ and $-\max(-\alpha, 0) = \min(\alpha, 0) = \alpha$, which achieves the proof. □

Remark 3: In the previous valuation of θ note that if $\max(\alpha, 0) = \alpha$ then $\min(\alpha, 0) = 0$ and conservely if $\max(\alpha, 0) = 0$ then $\min(\alpha, 0) = \alpha$; so, according to the sign of α, the real number θ is given by

$$\theta = \begin{cases} \max(\alpha, 0) = \alpha, & \text{if } \alpha > 0, \\ \min(\alpha, 0) = \alpha, & \text{if } \alpha < 0. \end{cases}$$

Based on Lemmas 2 and 3 results and Remark 3, we can now define a linearisation process, analogous to the Howard process (Spiteri et al., 2001), allowing to solve numerically the G-heat equation. Indeed, any component of the system (8) can be written as follows

$$u^{n+1}_{\ldots,j,l,k,\ldots} - \frac{\delta_t}{2}\sum_{i=1}^{d}\left(\overline{\sigma}_i^2.\max\left(\alpha_{j,l,k}, 0\right) + \underline{\sigma}_i^2.\min\left(\alpha_{j,l,k}, 0\right)\right) = u^n_{\ldots,j,l,k,\ldots}, \ \forall n \in \mathbb{N}. \quad (9)$$

Then, we associate to this previous system the linearised system (see Giraud and Spiteri, 1991)

$$\left(I + \frac{\delta_t}{2}.A(\tilde{U}^{n+1})\right).U^{n+1} = U^n, \quad (10)$$

where $\tilde{U}^{n+1}$ denotes the current value of the Howard iterate and each line of the matrix $A(\tilde{U}^{n+1})$ is constructed block by block by the following way: for all line of the system a part of the line is constructed as follows:

1 if the real number $\alpha_{j,l,k}$ is positive then the off-diagonal entries of the corresponding line of the matrix $A(\tilde{U}^{n+1})$ is constituted with $-\frac{\overline{\sigma}_i^2}{h_i^2}$

2 else, if the real number $\alpha_{j,l,k}$ is negative then the off-diagonal entries of the corresponding line of the matrix $A(\tilde{U}^{n+1})$ is constituted with $-\frac{\underline{\sigma}_i^2}{h_i^2}$

3 and all the numbers $2\frac{\overline{\sigma}_i^2}{h_i^2}$ or $2\frac{\underline{\sigma}_i^2}{h_i^2}$ are added in order to compute the corresponding diagonal entry.

Remark 4: Obviously, for $d > 1$, the matrix $A(\tilde{U}^{n+1})$ is a block matrix; for example if d = 2, then the matrix A(U) is a tridiagonal block matrix. Moreover, if for all $i \in \{1, ..., d\}$, $\overline{\sigma}_i = \underline{\sigma}_i = 1$, then the matrix A(U) is similar to the discretisation matrix of the Laplacian operator. In the previous linearisation process, note that each line is built line by line and block by block.

Lemma 4: The matrix $(\mathrm{I}+\frac{\delta_t}{2}.A(U))$ is an M-matrix.

Proof: Indeed, since the linearisation process and the spatial discretisation process lead to construct a matrix A irreducibly diagonally dominant, then, the matrix $(\mathrm{I}+\frac{\delta_t}{2}.A(U))$ is strictly (irreducibly) diagonally dominant (Ortega and Rheinboldt, 1970); moreover, at each time step, the global matrix $A(U)$ has strictly positive diagonal entries and non-positive off-diagonal entries. Consequently, $(\mathrm{I}+\frac{\delta_t}{2}.A(U))$ is an M-matrix and the proof is achieved. □

Remark 5: Then, at each time step we have to solve a large linear algebraic system where the matrix to invert is an M-matrix. So, for solving such linear system, we can use the sequential or the parallel synchronous or asynchronous subdomain methods with or without overlapping between the subdomains. In the next sub-section we will recall the general formulation of such sequential and parallel methods, and, for each time step, due to the property of the matrix $(\mathrm{I}+\frac{\delta_t}{2}.A(U))$, these algorithms converge.

2.4 Sequential and parallel relaxation algorithms

In this subsection, we recall the formulation of the parallel synchronous or more generally asynchronous subdomain methods, and we will use these algorithms for the sequential and parallel solutions of the global model problem (10).

2.4.1 Parallel subdomain iterative methods without overlapping

Consider the following algebraic system of equations

$$\mathcal{A}V = \mathcal{B}, \tag{11}$$

where $\mathcal{A} \in \mathcal{L}(\mathbb{R}^N)$, $V \in \mathbb{R}^N$, $\mathcal{B} \in \mathbb{R}^N$ and N denotes the size of the linear algebraic system (11), i.e., the number of points of discretisation in each direction. Furthermore, assume that

$$\mathcal{A} \text{ is an M-matrix.} \tag{12}$$

We consider the numerical solution of (11) by a parallel subdomain iterative method without overlapping, which in fact corresponds to a parallel block relaxation method, in which, a subdomain is constituted by gathering several adjacent blocks. Let $\beta \in \mathbb{N}$, be a positive integer and consider now the following block decomposition of the problem (11) into β subproblems

$$\sum_{j=1}^{\beta} \mathcal{A}_{ij}\mathcal{V}_j = \mathcal{B}_i, \ \forall i \in \{1,...,\beta\}, \tag{13}$$

where $\mathcal{V}_i, \mathcal{B}_i \in \mathbb{R}^{n_i}$, with $\sum_{i=1}^{\beta} n_i = N$, and $\mathcal{A} = (\mathcal{A}_{ij})$, according to the associated block decomposition. Consider now the solution of the subproblems (13) by an asynchronous parallel iteration (see Baudet, 1978; Miellou, 1975) which can be written as follows

$$\begin{cases} \mathcal{A}_{ii}\mathcal{V}_i^{(r+1)} = \mathcal{B}_i - \sum_{j \neq i} \mathcal{A}_{ij}\mathcal{W}_j \ , \text{ if } i \in s(r), \\ \mathcal{V}_i^{(r+1)} = \mathcal{V}_i^{(r)}, \text{ if } i \notin s(r), \end{cases} \tag{14}$$

where $\{\mathcal{W}_1, .., \mathcal{W}_{i-1}, -, \mathcal{W}_{i+1}, .., \mathcal{W}_\beta\}$ are the available values of the components $\mathcal{V}_j$ for $j \neq i$, defined by $\mathcal{W}_j = \mathcal{V}_j^{(\rho_j(r))}$, where $\mathcal{S} = \{s(r)\}_{r \in N}$ is a sequence of non-empty subsets of $\{1, ..., \beta\}$ denoting the subsets of indices of the components updated at the $(r + 1)^{\text{th}}$ iteration and $\mathcal{R} = \{\rho_1(r), \ldots, \rho_\beta(r)\}_{r \in \mathbb{N}}$, is a sequence of element of $\mathbb{N}^\beta$; furthermore $\mathcal{S}$, $\mathcal{R}$ verify the following assumptions

$$\begin{aligned} &\forall i \in \{1,2, ..., \beta\}, \ s(r) \neq \emptyset \ \forall r \in \mathbb{N}, \\ &\text{the set } \{r \in \mathbb{N} \mid i \in s(r)\} \text{ is infinite,} \\ &\forall i \in \{1,2, ..., \beta\}, \ \forall r \in \mathbb{N}, \ \rho_i(r) \leq r, \\ &\forall i \in \{1,2, ..., \beta\}, \ \lim_{r \to \infty} \rho_i(r) = +\infty. \end{aligned}$$

The previous asynchronous iterative scheme models computations are carried out in parallel without order or synchronisation and describe in fact a subdomain method without overlapping. In particular, it permits one to consider distributed computations whereby processors go at their own pace according to their intrinsic characteristics and computational load. The parallelism between the processors is well described by $\mathcal{S}$ since $s(r)$ contains the number of components relaxed by each processor on a parallel way while the use of delayed components in (14) permits one to model non-deterministic behaviour and does not imply inefficiency of the considered distributed scheme of computation. Note that, theoretically, each component of the vector must be relaxed an infinite number of times. The choice of the relaxed components may be guided by any criterion, and, in particular, a natural criterion is to pick-up the most recently available values of the components computed by the other processors.

Remark 6: The algorithm (14) describes a computational method where the communications between the processors can be asynchronous. Among them parallel synchronous methods are modelled, when $\rho(r) = r$, $\forall r \in \mathbb{N}$; moreover if $s(r) = \{1, ..., \beta\}$ and $\rho(r) = r$, $\forall r \in \mathbb{N}$, i.e., $\mathcal{S} = \{\{1, ..., \beta\}, \ldots \{1, ..., \beta\}, \ldots\}$, then (14) describes the sequential block Jacobi method while if $s(r) = r.mod(\beta) + 1$ and $\rho(r) = r$, $\forall r \in \mathbb{N}$, i.e., $\mathcal{S} = \{\{1\}, \{2\}, ..., \{\beta\}, \{1\}, \ldots, \{\beta\}, \ldots\}$ then (14) models the sequential block Gauss-Seidel method (see

Mittal, 2014); in addition, if $\mathcal{S} = \{\{1\}, ..., \{\beta\}, \{\beta\}, \{\beta-1\}, \ldots, \{1\}, \{1\}, ..., \{\beta\}, \{\beta\}, \ldots, \{1\}, ...\}$ and $\rho(r) = r$, $\forall r \in \mathbb{N}$, then (14) models the alternating direction method (ADI, see Ge, 2006; Tang and Christov, 2007). So, the previous model of parallel asynchronous relaxation algorithm appears like a general model.

Subtracting (13) to (14), we obtain

$$\mathcal{A}_{ii}\left(\mathcal{V}_i^{(r+1)} - \mathcal{V}_i\right) = -\sum_{j\neq i} \mathcal{A}_{ij}\left(\mathcal{W}_j - \mathcal{V}_j\right), \text{ if } i \in s(r). \qquad (15)$$

Let $g_i \in G_i(\mathcal{V}_i^{(r+1)} - \mathcal{V}_i)$ be an element of the duality map, where $\forall i \in \{1, ..., \beta\}$, $\forall p \in [1, \infty]$, g_i satisfy

$$G_i(\mathcal{V}_i) = \left\{ g_i \in \mathbb{R}^{n_i} \mid < \mathcal{V}_i, g_i >_i = \|\mathcal{V}_i\|_p \text{ and } \|g_i\|_i^* = 1 \right\};$$

where $\|.\|_p$ is the classical l_p-norm defined in $\mathbb{R}^{n_i}$; then by multiplying (15) by g_i, we obtain for $i \in s(r)$

$$< \mathcal{A}_{ii}\left(\mathcal{V}_i^{(r+1)} - \mathcal{V}_i\right), g_i >_i = -\sum_{j\neq i} < \mathcal{A}_{ij}\left(\mathcal{W}_j - \mathcal{V}_j\right), g_i >_i . \quad (16)$$

$\mathcal{A}$ being an M-matrix, it follows that the diagonal submatrices $\mathcal{A}_{ii}$, $\forall i \in \{1, ..., \beta\}$, are also M-matrices; so applying a characterisation of M-matrices from Spiteri (2003), these submatrices are strongly accretive matrices and consequently $\forall i \in \{1, ..., \beta\}$ the following inequality holds

$$< \mathcal{A}_{ii}\left(\mathcal{V}_i^{(r+1)} - \mathcal{V}_i\right), g_i >_i \geq \mu_{ii} \left\|\mathcal{V}_i^{(r+1)} - \mathcal{V}_i\right\|_p, \ (\mu_{ii} > 0); \quad (17)$$

concerning the right hand side of the relation (16), the mapping $< ., . >_i$ being a bilinear form, $\forall j \in \{1, ..., \beta\}$, $j \neq i$, for all $p \in [1, \infty]$, we obtain the following upperbound

$$\sum_{j\neq i} < \mathcal{A}_{ij}\left(\mathcal{W}_j - \mathcal{V}_j\right), g_i >_i \leq \sum_{j\neq i} \mu_{ij} \left\|\mathcal{W}_j - \mathcal{V}_j\right\|_p, \ (\mu_{ij} > 0), \quad (18)$$

where μ_{ij} denotes the subordinate matricial norm associated with the scalar norm $\|.\|_p$. Now taking into account relations (17) and (18) we obtain finally the following inequality

$$\left\|\mathcal{V}_i^{(r+1)} - \mathcal{V}_i\right\|_p \leq \sum_{j\neq i} \frac{\mu_{ij}}{\mu_{ii}} \left\|\mathcal{W}_j - \mathcal{V}_j\right\|_p, \ \forall i \in \{1,...,\beta\}. \qquad (19)$$

Let us now denote by $\tilde{\mathcal{J}}$ the $(\beta \times \beta)$ matrix, with entries

$$\tilde{\mathcal{J}}_{ij} = \begin{cases} 0 & \text{if } i = j \\ \dfrac{\mu_{ij}}{\mu_{ii}} & \text{if } i \neq j \end{cases};$$

then, clearly $\tilde{\mathcal{J}}$ is the Jacobi matrix associated to the matrix $\tilde{\mathcal{M}}$, with diagonal entries μ_{ii} and off-diagonal entries $-\mu_{ij}$ and obviously $\tilde{\mathcal{J}}$ is a non-negative matrix. Moreover, let us define the vectorial norm of a vector $\mathcal{Y}$, by the positive vector of $\mathbb{R}^{\beta}$, the components of which are given by

$$\mathcal{Y} \to \overline{q}(\mathcal{Y}) = \{....,\| \mathcal{Y}_i \|_p, ...\};$$

thus, the inequalities (19) correspond to a vectorial Lipschitz condition and can be written as follows

$$\overline{q}\left(\mathcal{V}^{(r+1)} - \mathcal{V}\right) \leq \tilde{\mathcal{J}} \cdot \overline{q}(\mathcal{W} - \mathcal{V}), \ \forall \mathcal{W}.$$

If furthermore the spectral radius of the matrix $\tilde{\mathcal{J}}$ is less than one, then the parallel asynchronous iterative subdomain method without overlapping defined by (14) converges (see Baudet, 1978; Miellou, 1975). Such situation occurs when $\tilde{\mathcal{M}}$ is an M-matrix.

The matrix $\tilde{\mathcal{J}}$ being a non-negative matrix, then according to a result of Kaszkurewicz and Bhaya (1999), there exists a strictly positive vector Γ of $\mathbb{R}^{\beta}$ and a positive real number ν satisfying $0 \leq \nu < 1$, such that $\tilde{\mathcal{J}}.\ \Gamma \leq \nu.\Gamma$. Consider now the weighted maximum norm defined by

$$\| \mathcal{V} \|_{\Gamma,\infty} = \max_{1\leq i\leq \beta} \left(\frac{\| \mathcal{V}_i \|_p}{\Gamma_i} \right).$$

Then, according to a result of Miellou (1975), the fixed point mapping associated with the parallel asynchronous method (14) is contracting with respect to the above weighted uniform norm; so, according to a result of El Tarazi (1982), the asynchronous method converges to the solution of the problem (11). From a practical point of view, we can consider now some particular cases. Indeed, note that the previous mathematical analysis is particularly easy (see Giraud and Spiteri, 1991; Miellou and Spiteri, 1985) when $p = 1$ or $p = \infty$ due to the properties of diagonal dominance of the block diagonal matrices $\mathcal{A}_{ii}$, $\forall i \in \{1, ..., \beta\}$. Indeed in these cases, if, for example, a natural ordering of the grid points is chosen, then for the natural block decomposition of the matrix $\mathcal{A}$, the entries μ_{ij} of the matrix $\tilde{\mathcal{M}}$ correspond to the entries of the discretisation matrix obtained, for example in the two dimensional case when $d = 3$ and as previously said $\tilde{\mathcal{J}}$ is the Jacobi matrix of the matrix $\tilde{\mathcal{M}}$; note that the matrix $\tilde{\mathcal{M}}$ is an M-matrix. When $p = 2$, in order to obtain the value of μ_{ii}, the smallest eigenvalue of the i^{th} diagonal block of $\mathcal{A}^{(p)}$ has to be computed; the values of μ_{ij}, $i \neq j$ correspond then to the values of the norms of the off-diagonal blocks $\mathcal{A}_{ij}$, $j \neq i$; compared to the case where $p = 1$ or $p = \infty$, the case where $p = 2$, needs the necessary computation of the eigenvalues of the i^{th} diagonal block of $\mathcal{A}^{(p)}$ and consequently, due to the amount of computation, makes this last criterion less convenient since the diagonal blocks of the matrix change during the computation. Since obviously $\tilde{\mathcal{M}}$ is an M-matrix, then the parallel synchronous and asynchronous subdomain methods without overlapping converge. The reader is referred to Giraud and Spiteri (1991) and Miellou and Spiteri (1985) for more details.

So we can summarise the previous study as follows:

Proposition 1: Consider the solution of the algebraic system (11) by the parallel asynchronous relaxation methods (14); assume that assumption (12) holds. Furthermore, assume also that the matrix $\tilde{\mathcal{M}}$ with diagonal entries equal to μ_{ii} and off-diagonal entries equal to $-\mu_{ij}$, $j \neq i$, is an M-matrix. Then the parallel synchronous and asynchronous subdomain methods without overlapping (14) converges to the solution of the problem (11).

Remark 7: The assumptions of Proposition 1 correspond to the context developed in Giraud and Spiteri (1991) and Miellou and Spiteri (1985) where the convergence of the iterative algorithm is analysed by contraction techniques; nevertheless we can also analyse the convergence of the considered iterative method by partial ordering techniques, by applying the results of Miellou et al. (1998) and Spiteri et al. (2003)

Remark 8: Moreover assume that the algebraic system is split into $\bar{\beta}$ blocks, $\bar{\beta} \leq \beta$, corresponding to a coarser subdomain decomposition without overlapping; then using a result in Miellou and Spiteri (1985), it can be shown by using the same arguments, that the parallel asynchronous block relaxation methods converge for this coarser decomposition. Furthermore, if the subdomain decomposition associated with β blocks is a point decomposition, then the classical parallel asynchronous block relaxation methods converge for every subdomain coarser decomposition and for every numbering (lexicographical or red-black) of the blocks.

2.4.2 Parallel subdomain iterative methods with overlapping

If we consider now, the numerical solution of (11) by the Schwarz alternating method corresponding to the fact that the subdomains overlap each other, then we have to solve the following system

$$\tilde{\mathcal{A}}\tilde{\mathcal{V}} = \tilde{\mathcal{B}} \tag{20}$$

instead of (11), where $\tilde{\mathcal{A}}$, $\tilde{\mathcal{V}}$ and $\tilde{\mathcal{B}}$ are derived from the augmentation process of the Schwarz alternating method.

Let $\beta \in \mathbb{N}$ be a positive integer and consider now the following block decomposition of problem (20) into β subproblems

$$\sum_{j=1}^{\beta} \tilde{\mathcal{A}}_{ij}\tilde{\mathcal{V}}_j = \tilde{\mathcal{B}}_i, \ \forall i \in \{1,...,\beta\}, \tag{21}$$

where $\tilde{\mathcal{V}}_i, \tilde{\mathcal{B}}_i \in \mathbb{R}^{\tilde{n}_i}$, with $\sum_{i=1}^{\beta} \tilde{n}_i \geq N$, and $\tilde{\mathcal{A}} = (\tilde{\mathcal{A}}_{ij})$, according to the associated block decomposition.

Consider now the solution of the subproblems with overlapping (21) by an asynchronous parallel iteration (see Baudet, 1978; Miellou, 1975) which can be written as follows

$$\begin{cases} \tilde{\mathcal{A}}_{ii}\tilde{\mathcal{V}}_i^{(r+1)} = \tilde{\mathcal{B}}_i - \sum_{j \neq i} \tilde{\mathcal{A}}_{ij}\tilde{\mathcal{W}}_j \ , \text{ if } i \in s(r), \\ \tilde{\mathcal{V}}_i^{(r+1)} = \tilde{\mathcal{V}}_i^{(r)}, \text{ if } i \notin s(r), \end{cases} \tag{22}$$

where $\{\tilde{\mathcal{W}}_1, .., \tilde{\mathcal{W}}_{i-1}, -, \tilde{\mathcal{W}}_{i+1}, .., \tilde{\mathcal{W}}_\beta\}$ are the available values of the components $\tilde{\mathcal{V}}_j$ for $j \neq i$, and defined by $\tilde{\mathcal{W}}_j = \tilde{\mathcal{V}}_j^{(\rho_j(r))}$, where $\mathcal{S} = \{s(r)\}_{r \in N}$ is a sequence of non-empty subsets of $\{1, 2, .., \beta\}$, denoting the subsets of indices of the components updated at the $(r+1)^{\text{th}}$ iteration, $\mathcal{R}$ and $\mathcal{S}$ are defined by an analogous way than the one considered in Subsection 2.4.1.

According to a result of Evans and Deren (1991), the matrix $\tilde{\mathcal{A}}$ is also an M-matrix. So, the system (20) resulting from the augmentation process has the same properties as the initial algebraic system (11). In this context, we can apply a result of Miellou et al. (1998) and conclude to the convergence of the parallel asynchronous Schwarz alternating method (22) applied to the numerical solution of the problem (11). So we can summarise the previous study as follows.

Proposition 2: Consider the solution of the algebraic system (11) by the parallel asynchronous Schwarz alternating method (22); assume that assumption (12) holds. Then the parallel asynchronous Schwarz alternating method (22) converges to the solution of the problem (11), for every initial guess.

Remark 9: Note that, instead of the Schwarz method, we can also solve the large algebraic system (11) by an asynchronous multisplitting method (see Spiteri et al., 2003).

2.4.3 Application to the solution of the G-heat equation.

Now, thanks to the result of the Lemma 4, since at each time step of the time marching scheme the matrix $\mathcal{A} = (I + \frac{\delta_t}{2}.A(U))$ is an M-matrix, we can conclude immediately that the parallel synchronous or more generally asynchronous subdomain method with or without overlapping converge for every decomposition of the domain Ω. Moreover, according to the Remark 6 the classical sequential relaxation algorithms converge too.

3 Numerical experiments

For the numerical experiments we have implemented both sequential Howard-relaxation algorithms corresponding to a coupling of the Howard method with a relaxation method, like the Jacobi method or the Gauss-Seidel method, and also the comparison of the parallel synchronous and asynchronous subdomain methods without overlapping.

Table 1 Number of time steps and of discretisation points on each axis for $d = 1, 2, 3$

$d = 1$		$d = 2$		$d = 3$	
Time step	*Discretisation points*	*Time step*	*Discretisation points*	*Time step*	*Discretisation points*
20	1,000	20	50 * 50 = 2,500	20	50 * 50 * 50 = 125,000

In such experiments the finite difference discretisation is performed on a classical Cartesian grid defined in a square or cubic domain ($[0, 1]^d$, $d \geq 2$). Note that we can choose the spatial discretisation step $h_i = h = \frac{1.}{q+1}$, $q \in \mathbb{N}$, $\forall i \in \{1, ..., d\}$. Then the time marching algorithm for the solution of the G-heat equation can be written as follows:

$$U^{n+1} + \delta_t A(\tilde{U}^{n+1}).U^{n+1} = U^n. \tag{23}$$

For sake of simplification, we have considered that $\overline{\sigma}_i = \overline{\sigma}$ and also that $\underline{\sigma}_i = \underline{\sigma}$, for all i. The iterative scheme that computes U^{n+1} can be written as follows:

$$\begin{cases} U^{n+1,0} = U^n \\ U^{n+1,r+1} + \delta_t A(U^{n+1,r}).U^{n+1,r+1} = U^n. \end{cases} \tag{24}$$

In the 3D case, the matrix $A(U)$ is computed as follows:

$$\begin{cases} (\Delta u)_i = 0 \Rightarrow A_{i,j} = 0 \text{ for each } j \\ (\Delta u)_i > 0 \Rightarrow A_{i,j} = -\dfrac{\overline{\sigma}^2}{2h^2} \text{ for } j = i \pm 1,\ i \pm q,\ i \pm q^2 \\ \qquad\qquad A_{i,i} = \dfrac{3.\overline{\sigma}^2}{h^2} \\ (\Delta u)_i < 0 \Rightarrow A_{i,j} = -\dfrac{\underline{\sigma}^2}{2h^2} \text{ for } j = i \pm 1,\ i \pm q,\ i \pm q^2 \\ \qquad\qquad A_{i,i} = \dfrac{3.\underline{\sigma}^2}{h^2}. \end{cases} \tag{25}$$

The results of such numerical experiments are summurised below.

3.1 Sequential experiments.

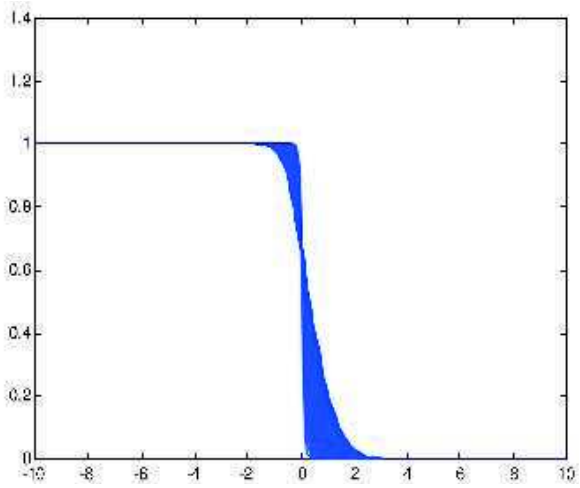
Sequential experiments have been conducted on a personal computer. For the sake of simplicity, the implementation is carried out by using MATLAB facilities. In these experiments we have considered the cases where $\Omega = [0, 1]^d$ is successively included in $\mathbb{R}^d$ for $d = 1, 2, 3$. Table 1 shows the maximal number of time steps and the number of discretisation points on each axis for each value of d. In the sequential experiments, note that the threshold of the stopping criteria for the Howard method and for the relaxation method is fixed to $1.0e-06$.

In the case where $d = 1$ we have compared the solution given by our method to the one obtained by Gong and Yang (2013) by using the Newton method for the solution of the following example arising from financial markets:

$$\begin{cases} \dfrac{\partial u(t,x)}{\partial t} - \dfrac{1}{2} \cdot \left(\overline{\sigma}^2 \cdot \max\left(\dfrac{\partial^2 u(t,x)}{\partial x^2}, 0 \right) \right. \\ \qquad \left. - \underline{\sigma}^2 \cdot \max\left(-\dfrac{\partial^2 u(t,x)}{\partial x^2}, 0 \right) \right) = 0, \\ \qquad -9.99 \leq x \leq 10, \\ u(0,\ x) = \mathcal{I}_{x \leq 0},\ -9.99 \leq x \leq 10, \\ u(t, -9.99) = 1,\ u(t,\ 10) = 0,\ 0 \leq t \leq 1, \end{cases}$$

where $\overline{\sigma} = 1, \underline{\sigma} = 0.5,\ T = 1,$ and we can conclude that the two solutions are the same (see Figure 1).

Figure 1 Numerical solution of the evolution G-heat equation (see online version for colours)



For the model problem, we have to solve very large algebraic systems; so, since the matrices are very sparse, iterative methods, like relaxation methods are generally well adapted, except when $d = 1$, where we can also consider the solution of each algebraic system by the TDMA method[1], corresponding to the classical Gauss method adapted to the case where the matrix A is a tridiagonal matrix. Note also that, for large values of d, the point-relaxation methods are well adapted to the solution of our model problem; indeed the block-relaxation methods need the Gaussian elimination of each tridiagonal block at each step of the Howard method, and this is very time consuming.

Finally, other iterative methods, like the conjugate gradient method or the preconditioned conjugate gradient method are not relevant, since in fact, we have to solve a kind of complementary problem related to applications where the solution is submitted to some constraints (see Spiteri et al., 2001); then when the constraints on the solution are saturated, it is difficult to find a conjugate direction.

Table 2 Elapsed time (sec.), number of linearisation and average number of Gauss-Seidel iteration for each linearisation phase, for $d = 2, 3$

Howard-relaxation 3D			*Howard-relaxation 2D*			*Howard-Gauss 1D*	
Time	*Linear.*	*G.S. iterat.*	*Time*	*Linear.*	*G.S. iterat.*	*Time*	*Linear.*
3,567.7	2	116 (1)	50.08	2	131 (1)	2.13	2
-	-	1 (2)	-	-	1 (2)	-	-

Owing to the faster rate of convergence both for the linearisation process and also for the relaxation method for the solution of the linear systems, we consider for the sequential experiments, the use of the Gauss-Seidel method; indeed, on one hand the Gauss-Seidel method is twice faster than the Jacobi method and on the other hand, we cannot consider the use of a relaxation parameter due to the fact that at each step of the Howard method, the matrix to invert is not the same. Note also that the matrix $\mathcal{A}$ is not symmetric and then the conjugate gradient method can not be used, except the generalised minimum residual method (GMRES).

For 20 time steps, Table 2 shows for $d = 1, 2, 3$ the elapsed time of computation in seconds, the number of linearisation at each time step and when $d = 2, 3$ the average number of iterations for the Gauss-Seidel method for each linearisation step. We note that the convergence is fast and that the elapsed time of computation is short. Then the Howard-relaxation method is well adapted to the solution of the nonlinear model problem.

3.2 Parallel experiments

In the 3D numerical experiments, the domain Ω is a cube ($[0, 1]^3$). In the linearisation process at each time step, the linear equations are solved by a point Gauss-Seidel relaxation method, which is easy to implement. The associated stopping criterion is the uniform norm of the difference between two successive iteration vectors.

The parallelisation is performed with MPI facilities. The reader is referred to Petcu (2005) for a study of the speedup when implementation of p.d.e. is performed on workstations. The domain decomposition consists in splitting the cube Ω into parallelepipeds (only the y and z axis are cut). The whole computation (23), (24), (25) is parallelised and implemented with MPI. The parallel numerical experiments have been carried out on the HPC@LR supercomputing centre located in Montpellier (France) and on Grid5000.

3.2.1 Parameters

The parameters that have been set up for the numerical experiments are:

- number of discretisation points for the axis x, y, and z: 300, and so $N = 300^3 = 27{,}000{,}000$
- $\underline{\sigma} = 0.353$
- $\overline{\sigma} = 0.707$
- $\delta_t = 0.02$
- number of time steps: 5
- stopping threshold for linearisation iterations: 10^{-8}
- stopping threshold for Gauss-Seidel relaxations: 10^{-10}.

Furthermore, the initial condition is:

$$u(t=0) = 10^7 \times P(x) \times P(y) \times P(z)$$

where

$$P(\chi) = \chi(1-\chi)\left(\frac{1}{2}-\chi\right)\left(\frac{1}{3}-\chi\right)\left(\frac{2}{3}-\chi\right).$$

This initial condition is suitable for the Dirichlet boundary conditions and moreover, the curvature of $u(t = 0)$ is such that the sign of Δu is not constant across the domain Ω. As a consequence, both $\overline{\sigma}$ and $\underline{\sigma}$ are considered in the computation of the linearisation matrix of the G-heat operator.

3.2.2 Results on HPC@LR

The parallel numerical experiments have been carried out on the HPC@LR[2] supercomputing centre located in Montpellier (France). The HPC@LR cluster is composed of bi-processor SMP nodes on 40 GB/s Infiniband network. Each node has two Intel Xeon Westmere 6 core processors sharing 24 Gbytes of memory.

The parallel efficiencies reported in Table 3 are poor owing to the domain splitting that does not cut the x axis, which is not optimal from the message passing standpoint. Since CPU technology tends to improve faster than interconnection bandwidth, the impact of communication overhead will be increasing. This situation is very interesting for the study of the asynchronous iterative algorithms.

Table 3 shows that the asynchronous relaxations are interesting above eight nodes. Below eight nodes, the synchronous relaxations have a tendency to perform better. Table 4 may suggest that the drop of efficiency observed in the synchronous algorithm is due to the increase of the total number of relaxation. Indeed, this well known behaviour is related to the parallelisation of the Gauss-Seidel relaxation scheme. However, Table 5 shows that the drop of efficiency is mainly due to message passing issues. Indeed, in the synchronous case, the mean efficiencies of one relaxation are close to the global efficiencies.

Table 3 Elapsed time (sec.), speed-up and efficiency for the solution of 3D G-heat equation (300 × 300 × 300) on five time steps, with HPC@LR

nb. node	*nb. core*	*Synchronous*			*Asynchronous*		
		Time	*Speed-up*	*Efficiency*	*Time*	*Speed-up*	*Efficiency*
1	1	50,413	-	-	-	-	-
1	6	14,549	3.46	0.57	15,695	3.21	0.53
1	12	7,433	6.78	0.56	8,564	5.88	0.49
2	24	4,778	10.55	0.43	4,153	12.13	0.50
4	48	1,911	26.38	0.54	2,073	24.31	0.50
6	72	1,291	39.04	0.54	1,421	35.47	0.49
8	96	1,321	38.16	0.39	1,053	47.87	0.49
10	120	1,130	44.61	0.37	840	60.01	0.50

Table 4 Total number of Gauss-Seidel relaxations and total number of linearisation iterations for the solution of 3D G-heat equation (300 × 300 × 300) on five time steps, with HPC@LR

nb. node	*nb. core*	*Synchronous*		*Asynchronous*	
		Relaxations	*Iter. lin.*	*Relaxations*	*Iter. lin.*
1	1	68,972	25	-	-
1	6	69,253	25	71,859	25
1	12	69,491	25	76,225	26
2	24	69,701	25	78,343	27
4	48	69,941	25	77,033	28
6	72	70,104	25	78,940	28
8	96	70,321	25	79,416	29
10	120	70,466	25	80,171	28

Table 5 Mean efficiency per relaxation for the solution of 3D G-heat equation (300 × 300 × 300) on five time steps, with HPC@LR

nb. node	*nb. core*	*Mean efficiency per relaxation*	
		Synchronous	*Asynchronous*
1	6	0.57	0.55
1	12	0.56	0.54
2	24	0.44	0.57
4	48	0.55	0.56
6	72	0.55	0.56
8	96	0.40	0.57
10	120	0.37	0.58

Moreover, Tables 4 and 5 show that using the asynchronous method leads to a slower convergence (higher number of relaxations), but this drop of convergence speed is compensated by more efficient parallel relaxations. The mean efficiency of a parallel relaxation is higher in the asynchronous case above two nodes. Furthermore, the parallel asynchronous relaxations do not suffer from message passing issues above eight nodes. The overall performance of the parallel computation has less variation in the asynchronous case.

Note that Table 4 also shows the total number of linearisation iterations. By using asynchronous relaxations inside the linearisation process, the number of iterations may be higher.

3.2.3 Grid5000 results

The parallel numerical experiments have been carried out on the Grid5000[3] national grid computing platform (France). The cluster that has been used for the numerical experiments is located at Sophia-Antipolis. The nodes are composed of bi-processor SMP Gigabit Ethernet network. Each node has two Intel Xeon Nehalem 4 core processors sharing 32 Gbytes of memory. Note that only two cores per node were used during the experiments.

Table 6 shows that the asynchronous relaxations are interesting above six nodes. Below six nodes, the synchronous relaxations have a tendency to perform better. Table 7 may suggest that the drop of efficiency observed in the synchronous algorithm is due to the increase of the total number of relaxation. Indeed, this well known behaviour is related to the parallelisation of the Gauss-Seidel relaxation scheme. However, Table 8 shows that the drop of efficiency is mainly due to message passing issues. Indeed, in the synchronous case, the mean efficiencies of one relaxation are close to the global efficiencies.

Compared to Table 3, Table 6 shows that using different CPUs allow both algorithms to perform better. Note that above six nodes in the Grid5000 cluster, the asynchronous algorithm becomes more efficient than the synchronous one, whereas in the supercomputing cluster with larger nodes and faster network, the asynchronous algorithms performs better from eight to ten nodes. This comparison allows us to state that asynchronous algorithms perform well with distributed nodes with few cores per node.

This corresponds in other words, to the context of high communication overhead, with respect to CPU speed. Note also that with fewer cores, each subdomain has more grid points. Thus, in Grid 5000 experiments, the workload per core is higher than in HPC@LR experiments. This also explains why parallel algorithms perform more efficiently here.

Table 6 Elapsed time (sec.), speed-up and efficiency for the solution of 3D G-heat equation (300 × 300 × 300) on five time steps, with Grid5000

nb. node	*nb. core*	*Synchronous*			*Asynchronous*		
		Time	*Speed-up*	*Efficiency*	*Time*	*Speed-up*	*Efficiency*
1	1	63,780	-	-	-	-	-
1	2	31,004	2.05	1.02	32,608	1.95	0.97
2	4	16,744	3.80	0.95	17,001	3.75	0.93
4	8	8,885	7.17	0.89	8,953	7.12	0.89
6	12	6,980	9.13	0.76	5,745	11.10	0.92
8	16	5,680	11.22	0.70	4,433	14.38	0.89
10	20	5,560	11.47	0.57	3,472	18.36	0.91
12	24	3,502	18.20	0.75	2,988	21.34	0.88
14	28	3,087	20.65	0.73	2,593	24.59	0.87
16	32	2,744	23.24	0.72	2,257	28.25	0.88
18	36	2,952	21.60	0.60	2,042	31.23	0.86

Table 7 Total number of Gauss-Seidel relaxations and total number of linearisation iterations for the solution of 3D G-heat equation (300 × 300 × 300) on five time steps, with Grid5000

nb. node	*nb. core*	*Synchronous*		*Asynchronous*	
		Relaxations	*Iter. lin.*	*Relaxations*	*Iter. lin.*
1	1	68,972	25	-	-
1	2	69,039	25	73,294	25
2	4	69,107	25	74,232	25
4	8	69,326	25	76,795	25
6	12	69,491	25	77,029	25
8	16	69,551	25	77,468	26
10	20	69,588	25	77,316	26
12	24	69,701	25	78,508	26
14	28	69,739	25	79,829	27
16	32	69,789	25	79,070	26
18	36	69,841	25	79,494	27

Table 8 Mean efficiency per relaxation for the solution of 3D G-heat equation (300 × 300 × 300) on five time steps, with Grid5000

nb. node	*nb. core*	*Mean efficiency per relaxation*	
		Synchronous	*Asynchronous*
1	2	1.02	1.03
2	4	0.95	1.00
4	8	0.90	0.99
6	12	0.76	1.03
8	16	0.70	1.01
10	20	0.57	1.02
12	24	0.76	1.01
14	28	0.74	1.01
16	32	0.73	1.01
18	36	0.60	0.99

The drop of efficiency of the synchronous algorithm, observed in Table 3, is also present in Table 6. In the same way, Table 8 shows that this drop of efficiency is also due to message passing issues, since the mean efficiencies per relaxation are close to the global efficiencies, in the synchronous case. In Table 7, we can observe the same increase of the number of relaxations as in Table 4. For 1, 12 and 24 cores, the number of synchronous relaxations are identical in both tables. Thus, both results are numerically comparable.

The results with Grid5000 confirm that asynchronous algorithms have slower convergence rate, but shorter elapsed time, when the number of nodes is sufficiently high. We can state that the extra computation times due to the extra relaxations are compensated by the lower communication overhead obtained by using the asynchronism. Furthermore, such compensation works only if the communication overhead is significant with respect to CPU workload. Besides, the performances of asynchronous iterations suffer from less variation than the synchronous one.

Table 7 shows that the number of linearisation iterations is also higher in Grid5000. This behaviour is the same as in Table 4.

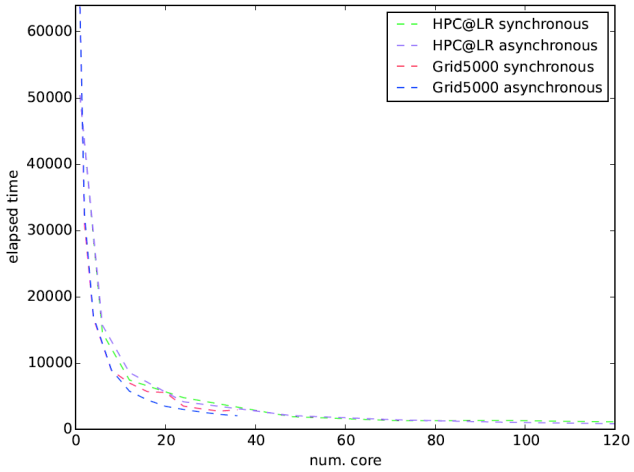
Finally note that with the same kind of parallelisation, we show in the present section that the efficiencies obtained were better than those obtained in Section 3.2.2. We will analyse in the next subsection the reasons of different performances obtained in HPC@LR and Grid5ooo environments.

3.2.4 Comparison of performance between the HPC@LR and Grid5ooo environments

In the present subsection we will discuss the difference of performances obtained between the two considered execution environments. Indeed, regarding the two architectures, HPC@LR is a classical HPC cluster (i.e., SMP nodes connected with Infiniband network),

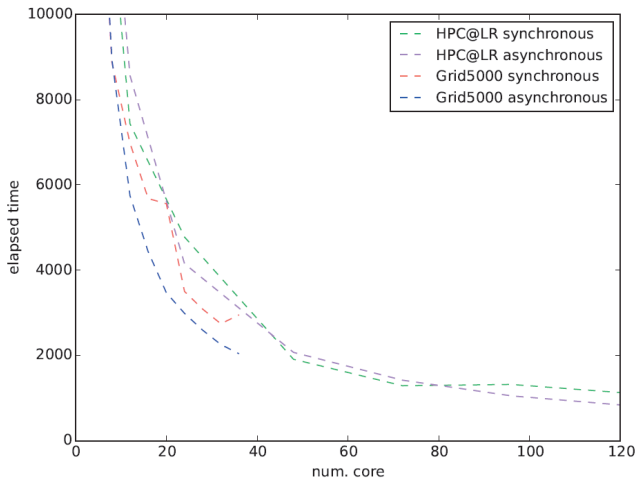
whereas Grid5000 is a distributed architecture deployed in geographically distant sites connected with internet network. For our experiments on Grid5000, we have performed the computations on a site where Infiniband network is not available and processors are different than the ones of HPC@LR. For the numerical aspects, the two testbeds make no difference, i.e., the used codes and the experiments are identical. What we want to show with the comparison of performance on these two testbeds, is the good behaviour of asynchronous iterations. Thus, it is classical that the performances of the parallel experiments change when the architecture of the multiprocessor changes. It is the case when we used HPC@LR and Grid5000 machines. A global comparison of elapsed times obtained with HPC@LR and Grid5000, for both synchronous and asynchronous algorithms, is given in Figure 2.

Figure 2 Comparison of elapsed times between HPC@LR and Grid5000 for both synchronous and asynchronous algorithms (see online version for colours)



For the sake of readability, Figure 3 focuses on the results for which elapsed times are lower than 10,000 seconds. Note that the same data is used in these two figures; the only difference is the range of the ordinate axis.

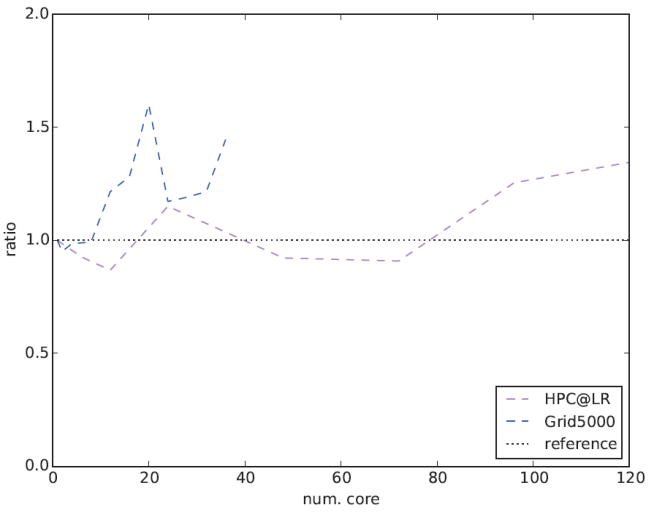
Figure 3 Comparison of elapsed times between HPC@LR and Grid5000 for both synchronous and asynchronous algorithms; focus on elapsed times lower than 10,000 seconds (see online version for colours)



The raw data plotted in Figure 2 shows that despite different CPU and network, regarding the number of cores the elapsed times of parallel runs in Grid5000 are slightly better than the ones obtained in HPC@LR.; for example for 24 cores the elapsed time obtained by using Grid5000, i.e., 3,502 seconds in synchronous mode and 2,988 seconds in asynchronous mode, respectively, is less than the one obtained with HPC@LR, i.e., 4,778 seconds and 4,153 seconds respectively. The explanation of this behaviour is the difference in the usage of the nodes. In HPC@LR, all the cores of each node are used in parallel experiments, whereas in Grid5000, only the half of the cores of each node are used. Consequently, the cache memory is better exploited in the latter context. Otherwise, without this tuning, there is absolutely no parallel efficiency on Grid5000. Nevertheless, we can notice that globally regarding the number of nodes, but not with the same number of cores, in the HPC@LR the elapsed times computation are short while in Grid5000 the efficiencies are better. Besides, the efficiencies presented in Table 8 are better than the one presented in Table 5 in HPC@LR even if the elapsed times presented in Table 6 are less good than the one presented in Table 3. In the asynchronous case the drop of convergence rate is compensated by more efficient parallel relaxations.

Furthermore, Figure 4 plots the ratio between synchronous and asynchronous elapsed times (t_s/t_a), in order to highlight the gain or the loss of performance provided by the asynchronous relaxations. This plot shows that a computing cluster with no Infiniband can benefit better from the asynchronous methods.

Figure 4 Comparison of the ratio between synchronous and asynchronous elapsed times on HPC@LR and Grid5000 (see online version for colours)



Besides, we can remark also that under ten nodes, the synchronous implementation gives better results, compared to the asynchronous one. Note that, when the number of subdomains varies, the input data that changes on an MPI process is the size of the subdomain, not the shape. Note that in our experiments we have not sought to minimise the number of boundary values to exchange with MPI.

Moreover, with our cutting procedure, the number of neighbours of each subdomain is either two or four, which is not a significant variation range. Moreover, the spatial discretisation steps along all directions remain the same as the ones we have in the sequential configuration. Besides, there is no specific discretisation technique for the boundaries of a subdomain that are adjacent to another subdomain. So putting more boundaries does not change the discretisation matrix. Furthermore, the parameters of the G-heat equation do not depend on the size of a subdomain, or its shape. So the numerical parameters that have an influence on the matrix entries remain unchanged in all possible cuttings of the domain. Our parallelisation is equivalent to a block decomposition of a large sparse matrix. In our case, the only influence of subdomain decomposition on numerical results is the number of relaxations, due to some classical ordering issues during parallel relaxations.

In our opinion and experience when the number of subdomains increases, the convergence slows down and when we use fewer than ten nodes, fewer relaxations are done. Thus the synchronisations are less penalising which perhaps is an explanation of the parallel performances in such case. It is also related to input data, particularly the size of the global system; finally, in this case, when fewer than ten nodes are used, the convergence is faster and the weight of synchronisations is low.

More precisely, for the reasons above, we can analyse the parallel efficiency with some simple HPC arguments. When the number of subdomains is small, on the one hand, we have more interior points per subdomain, which means more workload for the CPUs, and on the other hand, the total number of all boundary discretisation points of all processors is smaller, which leads to less traffic on the network. On the contrary, when the number of subdomains is large, there are more components to be exchanged, which leads to more traffic on the network and a larger weight of synchronisations, in addition to a slower convergence of the relaxation method. In the asynchronous mode idle times for the CPU are minimised.

One can remark sometimes a slower convergence, but not necessarily a longer elapsed time, as long as additional relaxations are cheap enough compared to the synchronous communications. Then, according to the former analysis of the parallel efficiency, we can state that the fewer subdomains, the higher efficiency for synchronous relaxations. So with few subdomains, it is difficult to ensure that the asynchronous relaxations can perform better than the synchronous ones.

4 Conclusions and perspectives

The study and the solution of the *G*-heat equation is a current challenge. In the presented study, we have solved this equation by a numerical way; such solution has been possible when Dirichlet conditions are considered. The proposed method uses an original linearisation process naturally based on an adaptation of the Howard process. This linearisation process leads to the solution of a large scale linear system; due to the fact that the resulting linear system is built with an M-matrix allows to study in a unified way the behaviour of the sequential and parallel relaxation methods used for the solution of the discretised and the linearised problem by various subdomains methods. Parallel experiments show the efficiency of the studied method.

As perspectives we can envisage easily in future works, by using a similar method, a situation where Dirichlet-Neumann boundary condition, or Robin (or Fourier) boundary condition and also Neumann boundary condition are considered, the situation is perhaps more complex from a theoretical point of view. We think that in this case, the considered study can be extended. Moreover, in the present study, the behaviour of the parallel algorithm has been studied by contraction techniques. We have good reason to believe that such analysis can be achieved also using perhaps different theoretical mathematical notions such that partial ordering techniques. Finally, since the *G*-heat equation is derived from the Hamilton Jacobi Bellman equation, which models applications in image processing, it will be also interesting to study other applications different to those arising in market finance.

References

Baudet, G. (1978) 'Asynchronous iterative methods for multiprocessor', *Journal of ACM*, Vol. 25, No. 2, pp.226–244.

Chau, M., El Baz, D., Guivarch, R. and Spiteri, P. (2007) 'MPI implementation of parallel subdomain methods for linear and nonlinear convection-diffusion problems', *Journal of Parallel and Distributed Computing*, Vol. 67, No. 5, pp.581–591.

Crandall, M.G., Ishii, H. and Lions, P.L. (1992) 'User's guide to viscosity solutions of second order partial differential equations', *Bulletin of the American Mathematical Society*, Vol. 27, No. 1, pp.1–67.

Dong, Y.H., Chen, H., Qian, W.N. and Ao-ying Zhou, A.Y. (2015) 'Micro-blog social moods and Chinese stock market: the influence of emotional valence and arousal on Shanghai Composite Index volume', *Int. J. of Embedded Systems*, Vol. 7, No. 2, pp.148–155.

El Tarazi, M.N. (1982) 'Some convergence results for asynchronous algorithms', *Num. Math*., Vol. 39, No. 3, pp.325–340.

Evans, D.J. and Deren, W. (1991) 'An asynchronous parallel algorithm for solving a class of nonlinear simultaneous equations', *Parallel Computing*, Vol. 17, Nos. 2–3, pp.165–180.

Ge, B. (2006) 'On the performance of parallel implementations of an ADI scheme for parabolic PDEs on shared and distributed memory systems', *Int. J. of High Performance Computing and Networking*, Vol. 4, Nos. 5/6, pp.222–231.

Giraud, L. and Spiteri, P. (1991) 'Parallel solution of nonlinear boundary values problems', *M.2.A.N*., Vol. 25, No. 5, pp.579–606.

Gong, X. and Yang, S. (2013) *The Numerical Properties of G-heat Equation and Related Application*, to appear, arXiv: 1304.1599V1 [math.NA].

Jaillet, P., Lamberton, D. and Lapeyre, B. (1990) 'Variational inequalities and the pricing of American options', *Acta Applicandae Mathematicae*, Vol. 21, pp.263–289.

Kaszkurewicz, E. and Bhaya, A. (1999) *Matrix Diagonal Stability in Systems and Computation*, Birkhauser, Boston.

Miellou, J.C. (1975) 'Algorithmes de relaxation chaotiques a retards', *Revue Française d'Automatique, Informatique, Recherche Opérationnelle. Analyse Numérique*, Vol. 9, No. R1, pp.55–82, RAIRO-R1, DUNOD 17 rue Remy Dumoncel 75 014 PARIS, Paris.

Miellou, J.C., El Baz, D. and Spiteri, P. (1998) 'A new class of iterative algorithms with order intervals', *Mathematics of Computation*, Vol. 67, No. 221, pp.237–255.

Miellou, J.C. and Spiteri, P. (1985) 'Un critère de convergence pour des méthodes générales de point fixe', *M.2.A.N.*, Vol. 19, No. 4, pp.645–669.

Mingshang, H. (2009) *Explicit Solutions of G-heat Equation with a Class of Initial Conditions by G-Brownian Motion*, pp.1–12, School of Mathematics, Shandong University, Jinan.

Mittal, S. (2014) 'A study of successive over-relaxation method parallelisation over modern HPC languages', *Int. J. of High Performance Computing and Networking*, Vol. 7, No. 4, pp.292–298.

Ortega, J.M. and Rheinboldt, W.C. (1970) *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York.

Peng, S. (1992) 'A generalized dynamic programming principle and Hamilton-Jacobi-Bellman equation', *Stochastics and Stochastic Reports*, Vol. 38, No. 2, pp.119–134.

Peng, S. (2005) 'G-expectation, G-Brownian motion and related stochastic calculus of Itô type', in Benth et al. (Eds.): *Stochastic Analysis and Applications, The Abel Symposium 2005, Abel Symposia 2*, pp.541–567, Springer-Verlag.

Peng, S. (2008) 'Multi-dimensional G-Brownian motion and related stochastic calculus under G-expectation', *Stochastic Processes and their Applications*, Vol. 118, No. 12, pp.2223–2253.

Peng, S. (2009) 'Survey on normal distributions, central limit theorem, Brownian motion and the related stochastic calculus under sublinear expectations', *Sci. China Ser. A*, Vol. 52, No. 7, pp.1391–1411.

Peng, S. (2010) *Nonlinear Expectations and Stochastic Calculus under Uncertainty with Robust Central Limit Theorem and G-Brownian Motion*, in book arxiv.math.PR/1002.4546v1.

Petcu, D. (2005) 'Speedup in solving differential equations on clusters of workstations', *Int. J. of Computational Science and Engineering*, Vol. 1, Nos. 2/3/4, pp.134–141.

Spiteri, P. (2003) 'A new characterization of M-matrices and H-matrices', *BIT Numerical Mathematics*, Vol. 43, No. 5, pp.1019–1032.

Spiteri, P., Miellou, J.C. and El Baz, D. (2003) 'Parallel asynchronous Schwarz and multisplitting methods for a non linear diffusion problem', *Numerical Algorithms*, Vol. 33, No. 1, pp.461–474.

Spiteri, P., Miellou, J.C. and El Baz, D. (2001) 'Asynchronous Schwarz alternating methods with flexible communication for the obstacle problem', *Calculateurs Parallèles, Réseaux et Systèmes Répartis*, Vol. 13, No. 1, pp.47–66.

Tang, X-H. and Christov, C.I. (2007) 'A multiunit ADI scheme for biharmonic equation with accelerated convergence', *Int. J. Computational Science and Engineering*, Vol. 3, No. 4, pp.295–304.

Wan, J.W.L., Lai, K., Kolkiewicz, A.W. and Tan, K.S. (2006) 'A parallel quasi-Monte Carlo approach to pricing multidimensional American options', *Int. J. of High Performance Computing and Networking*, Vol. 4, Nos. 5/6, pp.321–330.

Yong, J. and Zhou, X. (1999) *Stochastic Controls: Hamiltonian Systems and HJB Equations*, Springer-Verlag, New York.

Notes

1 TDMA means tridiagonal matrix algorithm, also known as the Thomas algorithm, is a simplified form of Gaussian elimination, that can be used to solve tridiagonal systems of equation.

2 https://www.hpc-lr.umontpellier.fr.

3 https://www.grid5000.fr.