



An Efficient Algorithm to Simulate a Brownian Motion Over Irregular Domains

S. Zein*, A. Lejay* and M. Deaconu*

January 4, 2010

Abstract

In this paper, we present an algorithm to simulate a Brownian motion by coupling two numerical schemes: the Euler scheme with the random walk on the hyper-rectangles. This coupling algorithm has the advantage to be able to compute the exit time and the exit position of a Brownian motion from an irregular bounded domain (with corners at the boundary), and being of order one with respect to the time step of the Euler scheme.

The efficiency of the algorithm is studied through some numerical examples by comparing the analytical solution with the Monte Carlo solution of some Poisson problems. The Monte Carlo solution of these PDEs requires simulating Brownian motions of different types (natural, reflected or drifted) over an irregular domain.

Keywords: Brownian Motion, Monte Carlo Methods, Partial Differential Equations, Euler Scheme, Random Walk On Rectangles.

1 Introduction

The term “Brownian motion” is used to describe a particle in a fluid that has a permanent and random agitation. The usage of this expression has been extended to all the mathematical models that describe random phenomena such that the fluctuations of the option markets in finance. Under appropriated hypothesis, a large class of stochastic processes can be modeled using the Brownian motion.

Mathematically, the Brownian motion is a Wiener process in which the conditional transition density of a particle at instant $t + \Delta t$ given its position x at instant t is a Gaussian law with mean x and variance Δt . Let X_t be this stochastic process in dimension d . When the Euler scheme

*INRIA Centre de Recherche Nancy-Grand Est — IECN, Campus scientifique, BP 70239, 56506 Vandoeuvre-lès-Nancy CEDEX, France, samih.zein@samtech.com

is used to simulate X_t over the time interval $[0, T]$ using N time steps, X_t is discretized by the sequence of random variables \tilde{X}_{t_k} as follows:

$$\begin{cases} \tilde{X}_0 = x \\ \tilde{X}_{t_k} = \tilde{X}_{t_{k-1}} + \sqrt{\Delta t} W_k, \end{cases} \quad (1)$$

where $\Delta t = T/N$, $\tilde{X}_{t_k} \stackrel{(law)}{=} X_{k\Delta t}$ is an approximation of the position of the particle at time t_k , W_k is a vector of d Gaussian variables with unit variance and $k = 1 \dots N$.

We are interested in simulating X_t over a bounded regular domain $D \subset \mathbb{R}^d$. In this case, two random variables are associated to X_t , the exit time τ defined by :

$$\tau = \inf\{t, X_t \notin D\}$$

and the exit position X_τ , the position the particle on the boundary of D at time τ . The exit time is approximated by $\tau_d = \inf\{t_k, \tilde{X}_{t_k} \notin D\}$. It is known that τ_d is not a good approximation of τ because it is of order $1/2$ with respect to Δt (see ([11])). A better approximation of τ would be to consider at each iteration k of the Euler scheme the exit time of the continuous stochastic process \tilde{X}_t defined between two iterations by:

$$\text{for } t \in [k\Delta t, (k+1)\Delta t], \tilde{X}_t = \tilde{X}_{t_k} + \sqrt{t - k\Delta t} W_k, \quad (2)$$

and $\tau_c = \inf\{t, \tilde{X}_t \notin D\}$. τ_c is the exit time of the continuous time stochastic process \tilde{X}_t starting at \tilde{X}_{t_k} . Its estimation is independent from the time step Δt . Some algorithms were presented to compute τ_c in ([14],[4]) for the one-dimensional case, in ([11],[14] and [5]) from a regular domain, ([15]) for a bi-dimensional cone and more generally in ([2]) in the case of a reflected Brownian motion. These algorithms are based on considering the Brownian bridge between \tilde{X}_{t_k} and $\tilde{X}_{t_{k+1}}$ and computing the probability that the Brownian bridge intersects the boundary. These algorithms have the drawback of assuming that the boundary is locally smooth when the Brownian particle is near the boundary. Hence these algorithms cannot consider a rectangular domain (for example) and simulate the Brownian motion near the corners.

In this paper, we propose to simulate the exit time τ and the exit position X_τ from a bounded domain, by coupling the Euler scheme with the random walk on the hyper-rectangles. The advantage of this algorithm over the ones described in ([11],[14] and [5]) is to handle domain boundaries with corners and for different types of Brownian motions (natural, reflected, drifted). To show the efficiency of our algorithm, numerical experiments are carried out by comparing the analytical with the Mont Carlo solution of some Poisson problems. The Monte Carlo solution is computed using the Monte Carlo method which requires the simulation of the exit time of the Brownian motion.

2 Brownian Motion and Diffusion Process

To describe this stochastic process, the transition density $p(x, y, t)$ of the particle's position at position y and time t starting at x must be specified. In the following, we show how it is derived

from the diffusion equation.

Einstein related the Brownian motion to the diffusion process (see [8],[9]). He considered a long thin tube of clear water. A drop of ink is injected at time $t = 0$ and at location $x = 0$. The ink will diffuse in the water due to the irregular motion of the ink particles. Let $p(0, y, t)$ be the density of ink at $y \in \mathbb{R}$ and time $t \geq 0$. This ink density is also the transition density of the position of an ink particle.

We have $p(0, y, 0) = \delta(y)$. The event that the particle moves from y to $y + h$ in time θ is also considered, and its transition density is denoted by $\rho(\theta, h)$. Then:

$$\begin{aligned} p(0, y, t + \theta) &= \int_{-\infty}^{\infty} p(0, y - h, t) \rho(\theta, h) dh \\ &= \int_{-\infty}^{\infty} (p - p_y h + \frac{1}{2} p_{yy} h^2 + \dots) \rho dh. \end{aligned}$$

Given that:

- ρ is a transition density, $\int_{-\infty}^{\infty} \rho dh = 1$
- $\rho(h, \theta) = \rho(\theta, -h)$ by symmetry so $\int_{-\infty}^{\infty} h \rho dh = 0$
- and the variance of ρ , $\int_{-\infty}^{\infty} h^2 \rho dh = \theta$.

These assumptions lead to:

$$\frac{p(0, y, t + \theta) - p(0, y, t)}{\theta} = \frac{p_{yy}(0, y, t)}{2} + \text{h.o.t.} \quad (3)$$

When θ tends to 0, we have

$$p_t = \frac{1}{2} p_{yy}. \quad (4)$$

This is the diffusion equation. It has the following analytical solution:

$$p(0, y, t) = \frac{1}{\sqrt{2\pi t}} e^{-\frac{y^2}{2t}}. \quad (5)$$

This says the transition density of the position of the Brownian motion at time t starting at $x = 0$ is a normal distribution $N(0, t)$. More specifically, this motion is called a **natural** Brownian motion. It is a stochastic process, X_t that verifies the following properties:

- $X(0) = x$ almost surely,
- $X(t) - X(s) \sim N(0, t - s), \forall 0 \leq s \leq t$,
- $\forall 0 \leq t_1 \leq t_2 \dots \leq t_n$ the random variables $X(t_1), X(t_2) - X(t_1) \dots$ are independent.

In our work, other types of Brownian motion are used. The transition densities of their positions are also deduced from the solutions of the corresponding diffusion equations. We give a short description of them in the following, without the exhaustive mathematical theory behind.

The Brownian motion over the segment $[-1, 1]$ and starting at x at $t = 0$ is a natural one that is killed when the particle reaches one of the two boundaries of the segment. Its position y at time t verifies the following PDEs (Kolmogorov's equations):

$$\begin{cases} \partial_t p(t, x, y) = \frac{1}{2} \Delta_x p(t, x, y) \text{ over } \mathbb{R}_+^* \times (-1, 1), \\ p(t, x, y) \longrightarrow \delta_y(x), t \rightarrow 0, \\ \partial_t p(t, x, y) = \frac{1}{2} \Delta_y p(t, x, y) \text{ over } \mathbb{R}_+^* \times (-1, 1), \\ p(t, x, y) \longrightarrow \delta_x(y), t \rightarrow 0, \\ p(t, x, y) = 0 \text{ if } |x| = 1 \text{ or } |y| = 1. \end{cases} \quad (6)$$

Two random variables can be defined based on this stochastic process, the exit time τ (the time X_t reaches the boundary) and the exit position X_τ :

$$\tau = \inf\{t, X_t \notin [-1, 1]\}, \quad (7)$$

$$X_\tau \in \{-1, 1\}. \quad (8)$$

Note that Dirichlet conditions are taken on the boundaries of the segment. One can deduce the analytical expression of the transition density p of the position of the particle (see [6]):

$$p(x, y, t) = \frac{1}{\sqrt{2\pi t}} \sum_{n=-\infty}^{\infty} \left(\exp\left(-\frac{(x-y-4n)^2}{2t}\right) - \exp\left(-\frac{(x-y-2-4n)^2}{2t}\right) \right). \quad (9)$$

The Brownian motion defined over $[-1, 1]$ and **reflected** at -1 is a natural Brownian motion where the particle is not killed at 1 but it rebounds into the interior of the segment and killed at -1 . This corresponds to a Neumann condition at 1 in the preceding system of PDEs (6). The exact mathematical definition of the reflection (using the local time notion) can be found in [2]. Finally, the third type of Brownian motion is the **drifted** one. In this case, the particle has a random displacement superposed to a deterministic one with a constant velocity μ . This corresponds to the transport term $\mu \nabla p$ that has to be added to the Laplacians in (6). The analytical expressions of $p(x, y, t)$ in these cases (reflected and drifted) can be found in [16].

The link between PDEs and Brownian motion is also valid in dimension d where the segment $[-1, 1]$ is replaced by a domain $D \subset \mathbb{R}^d$.

So far, the Brownian motion is characterized by the solution of some PDE. Conversely, the solution of a PDE can be computed using the Brownian motion. If the following Poisson problem has to be solved:

$$\begin{cases} \frac{1}{2} \Delta u + g(x) = 0, x \in D \\ u(x) = \psi(x), x \in \partial D, \end{cases} \quad (10)$$

then the solution $u(x)$ can be expressed in terms of Brownian motion through the Feymann-Kac relationship (see [10]):

$$u(x) = \mathbb{E}_x \left[\psi(X_\tau) + \int_0^\tau g(X_s) ds \right]. \quad (11)$$

In this relationship, X_t is a Brownian motion killed when it reaches the boundary ∂D , $\tau = \inf\{t, X_t \notin D\}$ is its exit time and X_τ is its exit position. If ∂D includes some Neumann conditions, X_t has to be reflected over this part of the boundary. If the Poisson equation contains an additional transport term $\mu \nabla u$ then X_t is the drifted Brownian motion. The usage the Feymann-Kac relationship to compute $u(x)$ is called Monte Carlo methods for solving PDEs.

3 Numerical Approximation of the Feymann-Kac Relationship

The Feymann-Kac relationship gives the solution of a PDE in terms of Brownian motion. Computing the solution using this relationship involves three ingredients: simulation of a Brownian motion X_t (as well as τ and X_τ), integration of g along the path of X_τ and approximation of the expectation operator. These three ingredients are detailed in the following.

Firstly, the simulation of a Brownian motion over a domain $D \subset \mathbb{R}^d$ is commonly done using the Euler scheme. A time step Δt is specified and X_t is replaced by $\tilde{X}_{t_k} \stackrel{(law)}{=} X_{k\Delta t}$, a sequence of random variables representing the position of the particle at time t_k . This sequence of random variables \tilde{X}_{t_k} is updated iteratively as follows:

$$\begin{cases} \tilde{X}_0 = x \\ \tilde{X}_{t_k} = \tilde{X}_{t_{k-1}} + \sqrt{\Delta t} W_k, \end{cases}$$

where W_k is a d-dimensional vector of Gaussian variables. The updating $(\tilde{X}_{t_k})_{k \in \mathbb{N}}$ is stopped when $\tilde{X}_{t_k} \notin D$ and τ is approximated by $\tilde{\tau} = t_k$. If X_t is a drifted Brownian motion, one has to add a drift term $\mu \Delta t$ to the update equation. If X_t is reflected on some part of ∂D and when \tilde{X}_{t_k} is near this reflecting boundary, the update equation has to be replaced by the one proposed in [2] (the symmetrized scheme) or use our coupling algorithm as it is explained in the numerical example.

Secondly, the integral over the path of X_t , $f = \int_0^\tau g(X_t) dt$, must be approximated. One must start by considering the following stochastic differential equations (SDE):

$$\begin{cases} df = g(X_t) dt \\ \text{with } f(0) = 0 \text{ and } X(0) = x \end{cases} \quad (12)$$

and integrate it until time τ . This SDE is discretized using the Euler scheme as follows:

$$\begin{cases} \tilde{X}_{k+1} = \tilde{X}_k + \sqrt{\Delta t} W_k, \\ \tilde{f}_{k+1} = \tilde{f}_k + g(\tilde{X}_k) \Delta t \end{cases} \quad (13)$$

where $\tilde{f}_{\tau^d} \approx \int_0^\tau g(X_t)dt$.

Finally the expectation operator in the Feymann-Kac relationship is estimated using a Monte Carlo method. N independent Brownian paths are simulated, namely $(\tilde{X}_{t_j}^i)_{j \in \{0, \dots, \tau^{d,i}\}}$, where $\tau^{d,i}$ stands for the discrete exit time associated to path i . The numerical solution of problem (10) is given by $u_{N,\Delta t}$:

$$u_{N,\Delta t} = \frac{1}{N} \sum_{i=1}^N [\psi(\tilde{X}_{\tau^{d,i}}^i) + f_{k^{d,i}}^i], \quad (14)$$

where $\tilde{X}_{\tau^{d,i}}^i$ is the exit position of path i . If the numerical solution of a PDE is computed using a Monte Carlo method, the numerical error is the sum of two errors:

- $E_{\Delta t}^D$, the error in the numerical approximation of the integral $\int_0^{\tau^d} g(X_s)ds$ and it is an $O(\Delta t^{1/2})$ in the case of the Euler scheme,
- $E_{N,\Delta t}^{MC}$, the Monte Carlo error from replacing the expectation operator with a finite sum, and we have $E_{N,\Delta t}^{MC} = O(N^{-1/2})$ with high probability.

We recall that $|u_{N,\Delta t} - u(x)| = (E_{\Delta t}^D + E_{N,\Delta t}^{MC}) = O(\Delta t^{1/2} + N^{-1/2})$.

3.1 Order of the Approximation

The difficulty of integrating numerically equation (12) until time τ is that τ is a random variable and we do not have an accurate estimation of it. As explained in [1] and [17], the naive approach to approximate τ with $\inf\{t_k, \tilde{X}_{t_k} \notin D\}$ makes the Euler scheme of order 1/2 with respect to Δt instead of one. This loss of accuracy is due to the fact that, when the particle is near the boundary and we have \tilde{X}_{t_k} and $\tilde{X}_{t_{k+1}}$ inside D , we do not know if the Brownian motion X_t has left D between \tilde{X}_{t_k} and $\tilde{X}_{t_{k+1}}$ (see Fig. 1).

To cope with this difficulty, we propose to couple the Euler scheme with the random walk on the hyper-rectangles. When \tilde{X}_{t_k} is near the boundary and the probability that $t_k \leq \tau \leq t_{k+1}$ is not neglectful, \tilde{X}_{t_k} is placed into a hyper-rectangle having one of its facets tangent to the boundary (see Fig. 2 and 4). The construction of such hyper-rectangle depends on the geometry of the domain. It is discussed in the numerical experiments in section 6.

Here, we are considering the continuous time Brownian motion \tilde{X}_t starting at \tilde{X}_{t_k} and killed at the boundary of the hyper-rectangle. Its exit time $\tau_c = \inf\{t, \tilde{X}_t \notin \text{hyperrectangle}\}$ is the exit time τ of the Brownian motion if it exits on the tangent facet to the boundary of the domain. This estimation of the exit time is independent of the time step of the Euler scheme. If \tilde{X}_{t_k} is near a reflecting boundary, then \tilde{X}_{t_k} is simulated using also the hyper-rectangle with a reflecting tangent facet (this case is discussed in the numerical examples). From [11] and [12], one can deduce that, when using the coupling algorithm, the numerical integration error along the Brownian path, $E_{\Delta t}^D$, is of order one with respect to Δt . The advantage of this algorithm over the other existing ones described in ([11], [14], [5]) is that it can handle the simulation of a Brownian

motion near an irregular boundary, a boundary with corners while the other existing algorithms impose some smoothness conditions on the boundary. It can also be extended to the case of the reflected and the drifted Brownian motion. These cases are discussed later and in the numerical experiments.

Some numerical experiments are carried out to verify the order 1 of $E_{\Delta t}^D$ with respect to Δt using our algorithm by comparing the analytical solution with the Monte Carlo solution of some PDEs.

4 Random Walk on The Hyper-rectangles

The algorithm for simulating the exit time and the exit position of a Brownian motion from a hyper-rectangle is presented . This algorithm is a generalization of the random walk on the rectangles algorithms proposed in [7]. The hyper-rectangle of dimension d is of the form $\prod_{i=1}^d [-L_i, L_i]$. We denote by X_t the Brownian motion starting from x at $t = 0$ inside the hyper-rectangle and (δ, y) the exit time and exit position of X_t from the hyper-rectangle. Algorithm 1 describes the simulation of the exit time and the exit position from the hyper-rectangle. It can be summarized in the following steps.

We recall that the multidimensional Brownian motion X_t has each of its coordinates X_t^i a one dimensional Brownian motion defined on the segment $[-L_i, L_i]$. let t^i be the exit time of each coordinate X_t^i and $\varepsilon_i = \pm L_i$ its exit position. The exit time of X_t , $\delta = \inf_i t^i$.

The algorithm is divided in two parts: finding the first coordinate (named i_{min}) that will exit first and its exit time (δ) and then simulating the position (y) of the Brownian motion at time δ .

The exit time and the exit position (t_1 and ε_1) of the first coordinate X_t^1 are simulated. δ and i_{min} are assigned to t_1 and 1 respectively and will be updated during the algorithm. Then each coordinate i from 2 to d is checked if it will exit before δ through the following steps. The exit position ε_i of X_t^i is simulated. The probability that its exit time t_i of X_t^i is less than δ and conditioned by ε_i is computed and a uniform random number is drawn in $[0, 1]$. If this random number is less than this probability, then current coordinate i will exit before all its preceding ones. The exit time t_i is simulated, i_{min} and δ are updated to i and t_i respectively and i is added to an initially empty set A . This set contains all the coordinates that their exit times have been simulated. Once all the coordinates are scanned, δ is the smallest exit time of all the coordinates thus it is the exit time of the multidimensional Brownian motion.

Finally, the exit position y of the Brownian motion has to be simulated. It is the position of each coordinate at time δ . For each coordinate i from 1 to d , if $i = i_{min}$ then its position is already simulated. If $i \in A$ then its position at δ is simulated conditioned only by its simulated exit time and position, else, the position is simulated conditioned by its simulated exit position and an exit time greater then δ .

Algorithm 1 Simulation of the exit of Brownian motion from a hyper-rectangle.

```
 $A = \emptyset$ 
{ Simulation of the exit position of  $X^1$  }
compute  $\alpha_1 = \mathbb{P}[X_{t_1}^1 = L_1]$ 
{ Choosing the exit position of  $X^1$  }
if  $U(0, 1) < \alpha_1$  then
     $\varepsilon_1 = L_1$ 
else
     $\varepsilon_1 = -L_1$ 
end if
{ Simulation the exit time of  $X^1$  }
draw  $t_1$  with  $\mathbb{P}[t_1 \in \cdot | X_{t_1}^1 = \varepsilon_1]$ 
 $\delta = t_1$  and  $i_{min} = 1$ 
{ Simulation of the exit time from the hyper-rectangle }
for  $i = 2$  to  $d$  do
    { Simulation of the exit position of  $X^i$  }
    compute  $\alpha_i = \mathbb{P}[X_{t_i}^i = L_i]$ 
    { Choosing the exit position of  $X^i$  }
    if  $U(0, 1) < \alpha_i$  then
         $\varepsilon_i = L_i$ 
    else
         $\varepsilon_i = -L_i$ 
    end if
    { Simulation of the exit time of  $X^i$  }
    draw  $t_i$  with  $\mathbb{P}[t_i \in \cdot | X_{t_i}^i = \varepsilon_i]$ 
    if  $t_i < \delta$  then
         $\delta = t_i$ ,  $i_{min} = i$  and  $A = A \cup \{i\}$ 
    end if
end for
{ Simulation of the exit position from the hyper-rectangle }
for  $i = 1$  to  $d$  do
    if  $i = i_{min}$  then
         $y_i = \varepsilon_i$ 
    else if  $i \in A$  then
        draw  $y_i$  with  $\mathbb{P}[y_i \in \cdot | t_i, X_{t_i}^i = \varepsilon_i]$ 
    else
        draw  $y_i$  with  $\mathbb{P}[y_i \in \cdot | \delta < t_i, X_{t_i}^i = \varepsilon_i]$ 
    end if
end for
return  $(\delta, y)$ 
```

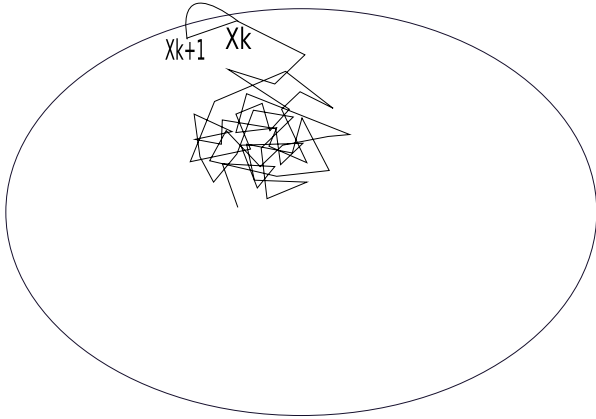


Figure 1: Simulation of the Brownian motion using the Euler scheme.

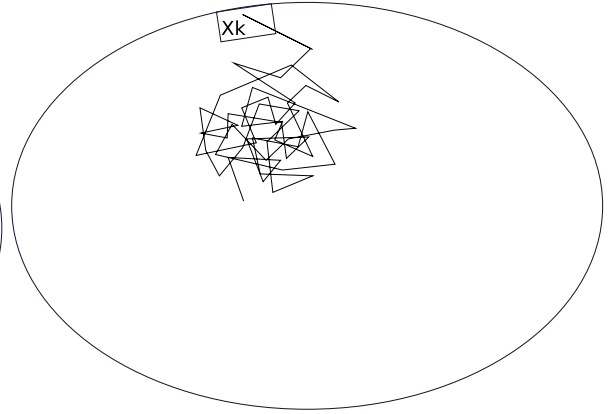


Figure 2: Simulation of the Brownian motion by coupling the Euler scheme with the random walk on the hyper-rectangles.

4.1 Some practical aspects for the algorithm implementation

In the preceding section, the transition density $p^{L_i}(x, y, t)$ of the Brownian motion over $[-L_i, L_i]$ is required. It can be deduced from the one over $[-1, 1]$ using the scaling principle:

$$p^{L_i}(x, y, t) = \frac{1}{L_i} p\left(\frac{x}{L_i}, \frac{y}{L_i}, \frac{t}{L_i^2}\right).$$

The distribution function $q(t, x)$ of the exit time τ :

$$q(t, x) = \Pr[\tau < t] = 1 - \int_{-1}^1 p(t, x, y) dy. \quad (15)$$

In the case where the Brownian motion is a natural one killed at the both boundaries of $[-1, 1]$, $p(x, y, t)$ is given by equation (9) and

$$q(t, x) = 2 - \sum_{n=-\infty}^{\infty} \left(F\left(\frac{(x-1-4n)^2}{\sqrt{2t}}\right) - F\left(\frac{(x+1-4n)^2}{\sqrt{2t}}\right) \right) \quad (16)$$

with $F(\cdot)$ the complementary error function. For the other Brownian motion types (reflected and drifted) see [16].

Computing the conditional transition density of the Brownian motion (see [3]):

- Conditioning by the exit position: let $S(x) = \mathbb{P}[X_\tau = 1 | X_0 = x]$ be the probability that X_t exits from 1. $S(x) = \frac{1-x}{2}$. The transition density of the position of X_t given that it will exit from 1 ($X_\tau = 1$) is given by:

$$p^1(t, x, y) = \frac{S(y)}{S(x)} p(t, x, y). \quad (17)$$

- Conditioning by the exit time: the transition density of the position of X_t given that it will exit at time $\tau = T$ is given by:

$$r(t, x, y) = \frac{p(t, x, y)}{q(T - t, y)} q(T, x). \quad (18)$$

The analytical expressions of these transition densities in the case of a reflected or a drifted Brownian motion can be found in [16].

The computation of $p(x, y, t)$ and $q(x, t)$ with these mathematical expressions (Equations 9 and 16) can be very expensive because it requires evaluation the function $erfc(\cdot)$ and the infinite series must be truncated at high order. An alternative method for computing $p(x, y, t)$ would be solving the diffusion equations in (6) with finite differences. Let δx and Δt be respectively the mesh size and the time step used to solve (6) using the finite differences method and $p_{ijk} = p(i\delta x, j\delta x, k\Delta t)$ be the numerical solution of (6) with this method computed at positions $x = i\delta x$, $y = j\delta x$ and time $t = k\Delta t$. One can compute all the possible values of p_{ijk} and place them into a three dimensional array, before simulating the Brownian motion.

Hence, $p(x, y, t)$ can be approximated by $p^*(x, y, t)$, a trilinear interpolation of the p_{ijk} 's. This alternative method is found to be very much faster than the one using the analytical expression of $p(x, y, t)$ (because it uses tabulated values) and is very accurate.

5 Coupling the Two Numerical Schemes

We denote by **euler_step** the updating procedure of the Euler scheme given in (13) and **rec_step**(δ, y) the updating procedure using the from the hyper-rectangle and defined as follows:

$$\begin{cases} \tilde{X}_{k+1} = y \\ f_{k+1} = f_k + g(\tilde{X}_k)(\delta - t_k). \end{cases} \quad (19)$$

Let $CONF$ be a confidence region of D such that the probability that the Brownian motion exits from the domain between two iterations ($t_k \leq \tau \leq t_{k+1}$) is neglectful. Given that $\tilde{X}_{t_{k+1}} \sim N(\tilde{X}_{t_k}, \Delta t)$, $CONF$ is the set of points such that the distance of \tilde{X}_{t_k} to the boundary of D is larger than $d_{CONF} = 4\sqrt{\Delta t}$. The coupling algorithm of the Euler scheme with the random walk on the hyper-rectangles consists in doing an Euler step if the X_t is very likely to stay in the domain and hyper-rectangles step if the exit probability is not neglectful. It is given in algorithm 2.

6 Numerical Experiments

Some numerical experiments are carried out to verify the efficiency of our proposed coupling algorithm. This is done by computing the relative error between the analytical solution and the Monte Carlo solution of some Poisson problems. The Monte Carlo solution requires the

Algorithm 2 Coupling the Euler scheme with the random walk on the hyper-rectangles.

```

 $\tilde{X}_0 = x, f = 0, k = 0$ 
while  $\tilde{X}_k \in D$  do
  if  $\tilde{X}_k \in CONF$  then
     $\tilde{X}_{k+1} = \tilde{X}_k + \sqrt{\Delta t} W_k$ 
     $\tilde{f}_{k+1} = \tilde{f}_k + g(\tilde{X}_k) \Delta t$ 
     $k = k + 1$ 
  else
    choose a hyper-rectangle containing  $\tilde{X}_{t_k}$  and having one facet tangent to  $D$ 
    simulate  $(\delta, y)$ 
     $\tilde{X}_{k+1} = y$ 
     $f_{k+1} = f_k + g(\tilde{X}_k)(\delta - t_k)$ 
     $k = k + 1$ 
  end if
end while
return  $(\delta, y)$ 

```

Euler time step	Relative Error
5×10^{-4}	0.0061
10×10^{-4}	0.0056
20×10^{-4}	0.0059
40×10^{-4}	0.0082

Table 1: Relative error between numerical and analytical solutions of problem (20) for different Euler time steps.

computation of the exit time and the exit position of the Brownian motion from the domain which are computed using the coupling algorithm.

We start with the Poisson problem defined in (10) with:

$$\begin{cases} D = [-1, 1] \times [-1, 1] \\ g(x) = 1 \\ \psi(x) = 0. \end{cases} \quad (20)$$

The analytical solution of this particular problem can be deduced from the eigenfunctions of the Laplacian over the square and we have $u(0,0) = 0.858\dots$. Concerning the construction of the confidence region $CONF$ and the tangent rectangles, four rectangles are taken such that each one has one side on the boundary of the domain (see Fig. 3). When \tilde{X}_{t_k} motion is inside one of these rectangles, the random walk on hyper-rectangles is used for the updating procedure otherwise the Euler scheme is used. The width of these four rectangles is $d_{CONF} = 4\sqrt{\Delta t}$.

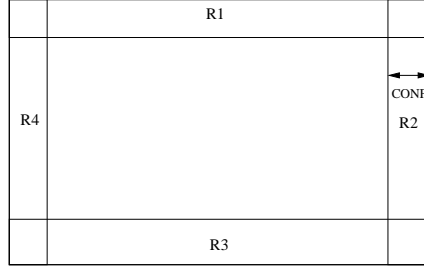


Figure 3: Construction of the confidence region and the tangent rectangle in case of problem 20

The choice of the value for the width comes from the property that $\tilde{X}_{t_{k+1}} - \tilde{X}_{t_k} \sim N(0, \Delta t)$ and $\Pr[|\tilde{X}_{t_{k+1}}^i - \tilde{X}_{t_k}^i| \geq 4\sqrt{\Delta t}]$ is neglectful. Hence the probability to have the Brownian motion exits the domain from the Euler updating is neglectful.

In this case, $E_{\Delta t}^D = 0$ so it is independent of the Euler time step. This is because we have $\int_0^\tau g(X_s) ds = \tau$ and the estimation of τ is drawn using the hyper-rectangle algorithm and the exact probabilities expressions. Thus, the numerical error only contains the Monte Carlo error out of the two cited before. Table 1 gives the relative error between numerical and analytical solutions of problem (20) for different Euler time steps. These results show that the relative error is independent from the Euler time step and it is relatively small (containing one source of error).

In the next numerical examples we compare the computational cost of our algorithm with the one proposed in [5] for the same Poisson problems. We recall that in [5] a numerical algorithm of order one is proposed to approximate the exit time and the exit position of the Brownian motion based on the Brownian bridge assumption. The following two cases of the Poisson problem are considered in dimensions $d = 2, 5$:

$$(21.1) \begin{cases} D = \{x, \|x\| \leq 1\} \\ g(x) = -\sum (x^i)^2 \\ \psi(x) = \frac{1}{6} \sum (x^i)^4 \\ u(x) = \psi(x), x \in D \end{cases} \quad (21.2) \begin{cases} D = \{x, \|x\| \leq 1\} \\ g(x) = \prod_{i=1}^d \cos(2\pi x^i) \\ \psi(x) = \frac{g(x)}{2\pi^2 \sqrt{d}} \\ u(x) = \psi(x), x \in D. \end{cases} \quad (21)$$

The construction of the hyper-rectangle is shown in Fig. 4. The outer circle represents the boundary of the domain and the inner circle is the boundary of the confidence region where the Euler scheme is used. The hyper-rectangle is taken as a hypercube of side length equals $d_{CONF} = 4\sqrt{\Delta t}$. One of the facets of the hypercube is tangent to the boundary of the domain at the orthogonal projection of \tilde{X}_{t_k} on the boundary. Note that the hypercube is not completely inside the domain but we neglect this geometrical error. The solution $u(x)$ is computed at point $x = (x_0, 0, \dots, 0)$ for different values of x_0 .

Figure 5 verifies the order one of $E_{\Delta t}^D$ with respect to Δt using the coupling algorithm in the case of problems (21). The numerical solution is computed using 5×10^6 independent paths.

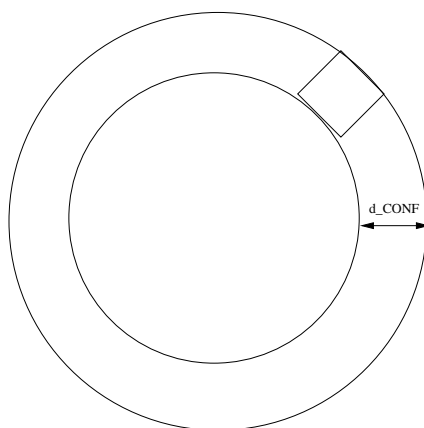


Figure 4: Construction of the confidence region and the tangent rectangle in case of problem 21

The 95% confidence interval for the numerical solution is about $u_{N,\Delta t} \pm 1.4\%$ in all the cases, and it took about 10 hours on a standard PC machine. However, the computational time of our algorithm is about 10 times larger than the one needed by the algorithm in [5] and it is not possible to handle the computational cost in the case of a very high dimensions ($d = 32$ or $d = 128$) as in [5].

Hence, one can conclude that it is only interesting to use our coupling algorithm near the domain's corners where the cited algorithms fail to simulate the Brownian motion and to use these cited algorithms where the domain boundary is locally smooth.

So far, the proposed coupling algorithm was studied in the case of the natural Brownian motion. The efficiency of the algorithm is now studied in the cases of the reflected and the drifted Brownian motion. For more details about the mathematical definitions of these two stochastic processes see for example [16]. In case (22), the Poisson problem in (22) is considered with a Neumann condition over Γ_N . In terms of Brownian motion, Γ_N is a reflecting boundary. The Construction of the confidence region in this case is the same as in the first example but the rectangle $R2$ has a reflecting side on the Neumann boundary. Inside the rectangle $R2$, the exit time and position of the Brownian motion are simulated using our preceding algorithm with a slight modification. For the first coordinate, we used the transition densities of the one dimensional Brownian motion reflected at one end of the segment and killed at the other.

Figure 6 gives the numerical results and the order one of the coupling algorithm.

$$\begin{cases} \frac{1}{2}\Delta u(x) + g(x) = 0 \\ \frac{\partial u}{\partial n} = 0 \quad x \in \Gamma_N \\ u(x) = \psi(x), \quad x \in \partial D - \Gamma_N. \end{cases} \quad (22)$$

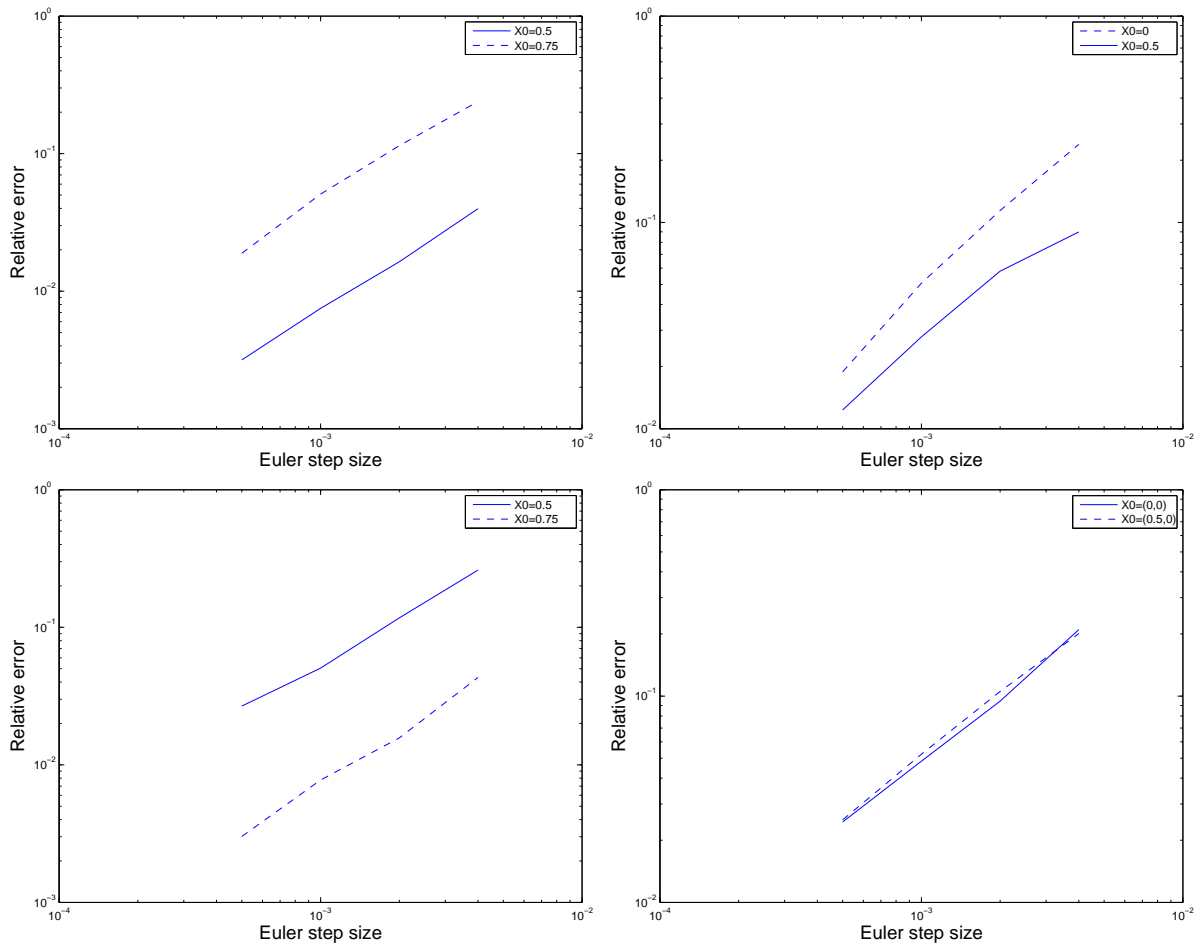


Figure 5: Verification of the order 1 of the coupled algorithm for problems (21.1) (first column) and (21.2) (second column) in dimensions 2 (first row) and dimensions 5 (second row).

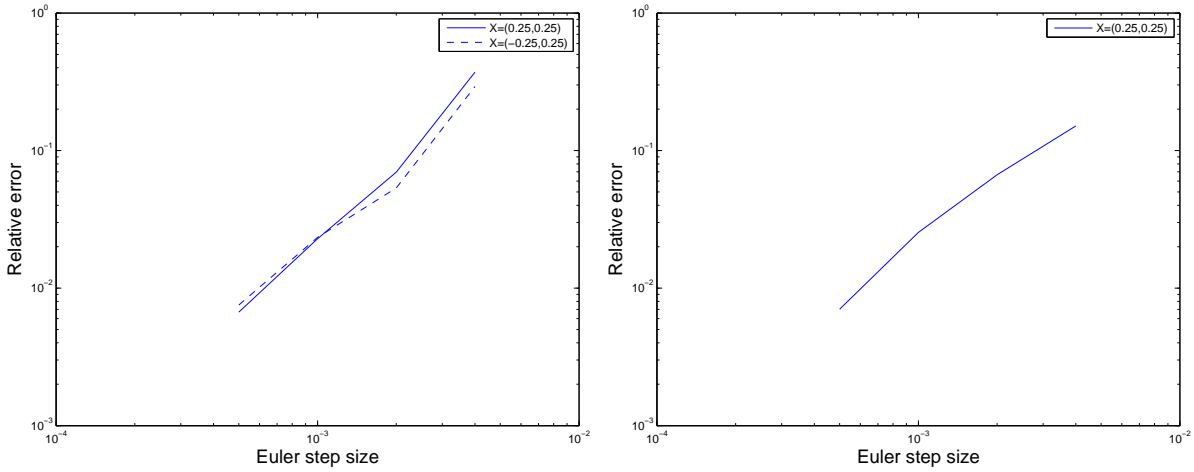


Figure 6: Verification of the order 1 of the coupling algorithm in the case of a reflected Brownian motion: problem 22.1 (left) and 22.2 (right).

with:

$$(22.1) \begin{cases} D = [-0.75, 0.75] \times [0, 1] \\ \Gamma_N = \{x \in D, x_2 = 0\} \\ g(x) = 2\pi \sin(2\pi x_1) (\pi \cos(2\pi x_2^2) (1 + 4x_2^2) + \sin(2\pi x_2^2)) \\ \psi(x) = \sin(2\pi x_1) \cos(2\pi x_2^2) \\ u(x) = \psi(x) \end{cases}$$

$$(22.2) \begin{cases} D = [-0.75, 0.75] \times [0, 1] \\ \Gamma_N = \{x \in D, x_2 = 0\} \\ g(x) = 2\pi \sin(2\pi x_1) (\pi \cos(2\pi x_2) (1 + 2x_2) + \sin(2\pi x_2)) \\ \psi(x) = \sin(2\pi x_1) \cos(2\pi x_2) \\ u(x) = \psi(x). \end{cases}$$

The efficiency of the simulation of the drifted Brownian motion is studied through the following Poisson problem:

$$\begin{cases} \frac{1}{2} \Delta u + \mu^t \cdot \nabla u + g = 0, x \in D \\ u(x) = \psi(x), x \in \partial D \end{cases} \quad (23)$$

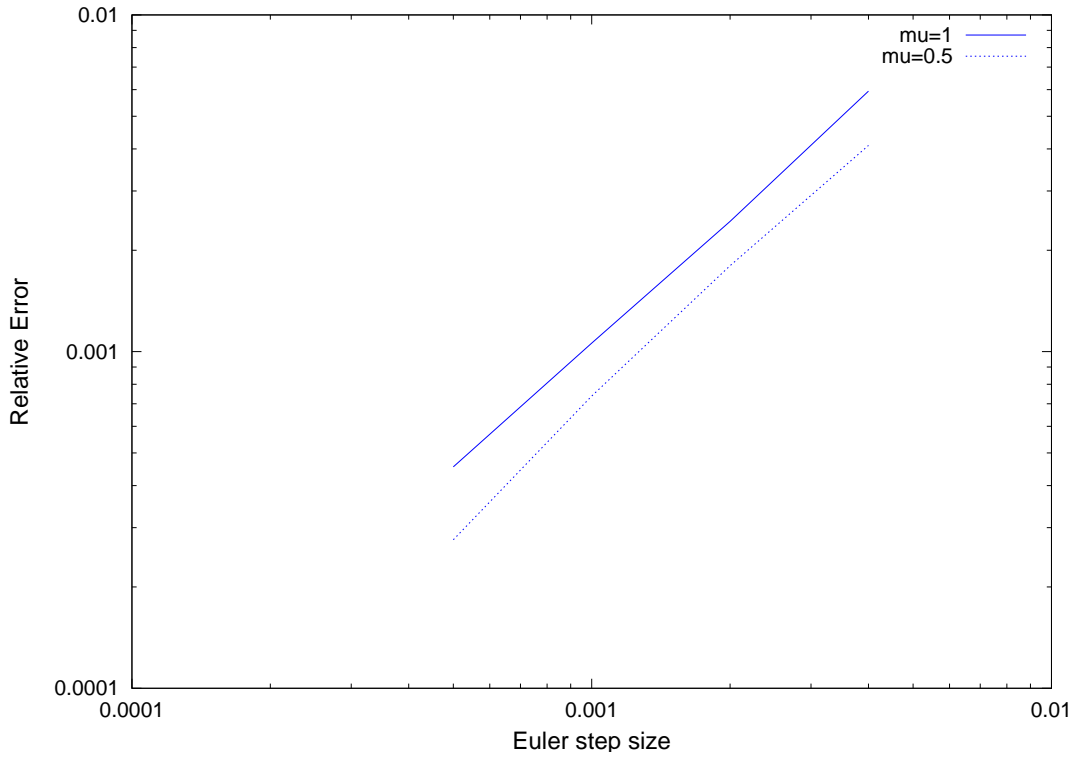


Figure 7: Verification of the order 1 of the coupling algorithm in the case of a drifted Brownian motion: problem (23), the solution is computed at the center of D .

with

$$\begin{cases} D = [-1, 1] \times [-1, 1] \\ g(x) = -\sum (x^i)^2 \left(1 + \frac{2}{3} \mu_i x_i\right) \\ \psi(x) = \frac{1}{6} \sum (x^i)^4 \\ u(x) = \psi(x). \end{cases}$$

The simulation of the exit time and the exit position from the rectangle for drifted Brownian motion is done with the same algorithm but using the expressions of the transition densities of the drifted one. Again, the order one the coupling algorithm is verified in the case of the drifted Brownian motion with a drift $\mu = 0.5$ and 1, see Figure 7.

7 Conclusion

An algorithm was presented in this paper to simulate a Brownian motion over an irregular domain. The algorithm couples the Euler scheme with the random walk on the hyper-rectangles.

It has the advantage over other existing algorithms of being of order one with respect to the time step of the Euler scheme to handle boundary domain with corners. The efficiency of the algorithm is studied by comparing the analytical solution with the Monte Carlo solution of some PDEs over irregular domains. The drawback of this algorithm is being computationally very expensive in high dimensional domains when compared with other existing algorithms. So in conclusion, we can say that our algorithm can replace the other existing algorithms in the low dimensional case (dimension less than 10) but in the high dimensional case one has to use it only near the corner of the boundary.

References

- [1] V. Bally and D Talay, The law of the Euler scheme for stochastic differential equations: I. Convergence rate of the distribution function, *Prob. Th. Rel. Fields*, 104:1, 43–60, 1996.
- [2] M. Bossy, E. Gobet and D. Talay, Symmetrized Euler scheme for an efficient approximation of reflected diffusions, *J. Appl. Probab.* 41:3, 877–889, 2004.
- [3] L. Breiman, *Probability*, Addison-Wesley, 1981.
- [4] M. Broadie, P. Glasserman and S. Kou, A continuity correction for discrete barrier options, *Math. Finance*, 7, 325–349, 1997.
- [5] F. M. Buchmann and W. P. Petersen, An Exit Probability Approach to Solve High Dimensional Dirichlet Problems, *SIAM J. Sci. Comput.*, 28:3, 1153–1166, 2006.
- [6] H. Carslaw and J. Jeager, *Conduction of heat in Solids*, Oxford Univ. Press, 1959.
- [7] M. Deaconu and A. Lejay, A random walk on rectangles algorithm, *Methodol. Comput. Appl. Probab.*, 8:1, 135–151, 2006.
- [8] A. Einstein, Uber die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen, *Annalen der Physik*, 17, 549-560 ,1905.
- [9] A. Einstein, *Investigations on the Theory of Brownian Movement*, Dover, 1956.
- [10] M. Freidlin, *Functional integration and partial differential equations*, Oxford Univ. Press, 1959.
- [11] E. Gobet, Weak approximation of killed diffusion using Euler schemes, *Stoch. Proc. Appl.*, 87, 167–197, 2000.
- [12] E. Gobet and S. Menozzi, *Stopped diffusion processes: boundary corrections and overshoot*, Preprint, 2009.

- [13] I. Karatzas and S. E. Shreve, *Brownian Motion and Stochastic Calculus*, Springer-Verlag, 1991.
- [14] R. Manella, Absorbing Boundaries and optimal stopping in a stochastic differential equation, *Phys. Lett. A*, 254:5, 257–262, 1999.
- [15] S. Menozzi, Improved simulation for the killed Brownian motion in a cone, *Siam J. Num. Anal.*, 44:6, 2610–2632, 2006.
- [16] G. N. Milstein and M. V. Tretyakov, *Stochastic Numerics for Mathematical Physics*, Springer-Verlag, 2004.
- [17] D. Talay and L. Tubaro, Expansion of the global error for numerical schemes solving stochastic differential equations, *Stoch. Anal. Appl.*, 8:4, 94–120, 1990.