



HAL
open science

A random walk on rectangles algorithm

Madalina Deaconu, Antoine Lejay

► **To cite this version:**

Madalina Deaconu, Antoine Lejay. A random walk on rectangles algorithm. *Methodology and Computing in Applied Probability*, 2006, 8 (1), pp.135-151. 10.1007/s11009-006-7292-3 . inria-00092424

HAL Id: inria-00092424

<https://inria.hal.science/inria-00092424>

Submitted on 10 Sep 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Random Walk on Rectangles Algorithm

Madalina Deaconu¹ — Projet OMEGA (INRIA / Institut Élie Cartan, Nancy)

Antoine Lejay^{1,†} — Projet OMEGA (INRIA / Institut Élie Cartan, Nancy)

Abstract: In this article, we introduce an algorithm that simulates efficiently the first exit time and position from a rectangle (or a parallelepiped) for a Brownian motion that starts at any point inside. This method provides an exact way to simulate the first exit time and position from any polygonal domain and then to solve some Dirichlet problems, whatever the dimension. This method can be used as a replacement or complement of the method of the random walk on spheres and can be easily adapted to deal with Neumann boundary conditions or Brownian motion with a constant drift.

Keywords: Monte Carlo method, Laplace operator, random walk on spheres/squares, Green functions, Dirichlet/Neumann problem

AMS Classification: 60C05, 65N

Published in *Method. Comput. Appl. Probab.* **8**:1, 135–151, 2006

Archives, links & reviews:

- DOI: 10.1007/s11009-006-7292-3

¹Current address: Projet OMEGA

INRIA Lorraine and Institut Élie Cartan de Nancy (IECN)

Campus scientifique

BP 239

54506 Vandœuvre-lès-Nancy CEDEX, France

E-mail: {mdeaconu,lejay}@loria.fr

[†]Partially supported by the French “Groupe de Recherche MOMAS” funded by ANDRA, BRGM, CEA, CNRS and EDF.

1 Introduction

The method of random walk on spheres (WOS) was introduced by M. Muller in Muller (1956) and provides a fast and efficient Monte Carlo method to solve a Dirichlet problem of the form

$$\begin{cases} \frac{1}{2}\Delta u(x) = 0 \text{ on } D, \\ u(x) = \varphi(x) \text{ on } \partial D, \end{cases} \quad (1)$$

or of type

$$\begin{cases} \frac{\partial u}{\partial t}(t, x) + \frac{1}{2}\Delta u(t, x) = 0 \text{ on } \mathbb{R}_+ \times D, \\ u(t, x) = \psi(t, x) \text{ on } \mathbb{R}_+ \times \partial D, \end{cases} \quad (2)$$

where D is a piecewise smooth bounded domain, and φ (resp. ψ) is a continuous, bounded function on ∂D (resp. on $\mathbb{R}_+ \times \partial D$). It is well known that the solution of (1) is given by $u(x) = \mathbb{E}_x[\varphi(B_\tau)]$ (resp. the solution of (2) is given by $u(t, x) = \mathbb{E}_{t,x}[\psi(\tau, B_\tau)]$), where τ is the first exit time from D of the Brownian motion starting from x (resp. starting from x at time t). Since the quantities needed are the distribution of the first exit time τ and the corresponding position B_τ , the idea of the WOS is to generate iteratively the first exit time and position from a sphere as big as possible in the domain D and centered on the previous position of the Brownian motion. When the exit position becomes close enough to the boundary, the algorithm stops. The method relies on the use of analytical expressions of the distribution function of the first exit time and position from a sphere, which may be deduced from existing expressions for the Green functions of the Laplace operator on a sphere.

Since then, the WOS has been extended in many directions (see for example Sabelfeld and Talay (1995); Hwang *et al.* (2003); Golyandina (2004); Sabelfeld *et al.* (2004), ...) and has also been proposed as a method for dealing with non-homogeneous media in Mil'shtejn and Rybkina (1993); Milstein and Tretyakov (1999).

As the Green function of a square or (hyper-)cubic domain is known, it is also possible to use squares instead of spheres in the WOS method. This approach could be preferred when the domain has a polygonal boundary as in the study of fissured porous media: see Torquato and Kim (1999); Campillo and Lejay (2002).

With spheres, the WOS generates some bias due to the fact that the particle is generally stopped when it is within a small layer around the boundary. By using squares in a polygonal domain the error comes only from the replacement of an expectation by an empirical mean.

In this paper, we introduce an algorithm for computing the first exit time and position from a rectangle in \mathbb{R}^2 (higher dimension may be treated as well) which is not necessarily centered on the starting point. For these computations, we use one-dimensional laws related to the Brownian motion and proper conditioning. The great advantage is that the rectangles may be chosen before any Monte Carlo simulation and not dynamically. Thus, we do not need to use smaller and smaller squares (or spheres) as the particle becomes closer to the boundary.

Although it is more time consuming to simulate the first exit time and position from a rectangle than from a sphere or a square, this algorithm has the following advantages over the WOS algorithm in the case of a polygonal domain: less simulations are needed and the particle is really stopped the first time it reaches the boundary. Moreover, the rectangles may be chosen once the domain is given, and it takes less time to “locate” the position of the particle in the domain. Besides, this algorithm is easily extended to many other problems : to deal with a Brownian motion with a constant drift (the Girsanov theorem may be used as well), to deal with a zero Neumann boundary conditions on some part of the boundary of D , but also to compute the position of the Brownian motion at a given time t (when $t < \tau$).

Moreover, there is no reason that one restricts himself to one method. We can thus combine spheres, squares and rectangles in function of the geometry of the domain.

Having the distribution of the first exit time and position from a domain for the Brownian motion is very important for practical purpose: it may be used to compute some effective coefficients in porous media (Noetinger and Estébenet, 2000; Torquato and Kim, 1999; Campillo and Lejay, 2002; Simonov and Mascagni, 2004), to solve a bi-harmonic problem (Milstein and Tretyakov, 1999) to compute the first eigenvalue of the Laplace operator as in Maire (2001) with applications in neutronic and shape optimization, to solve barrier options in finance...

The paper is organized as follows. After this introduction, we recall in the second section some distributions connected to the one-dimensional Brownian motion killed when it exits from an interval and of this process conditioned to exit for one of the end-point of the interval. Some calculation on conditioning on time or space are also expressed. In the following part the two-dimensional situation is considered. The rectangular situation is detailed and our algorithm introduced. The general polygon situation is also discussed. Some numerical tests and discussions are presented in the fourth section. Possible extensions and applications of this procedure are given in the last section.

2 Some distributions related to the one-dimensional Brownian motion

2.1 Notations

We denote by $p(t, x, y)$ the probability density function of a Brownian motion $(B, (\mathcal{F}_t)_{t \geq 0}, (\mathbb{P}_x)_{x \in \mathbb{R}})$ at time t starting from x and killed when it exits from the interval $[-1, 1]$, that is

$$p(t, x, y) dy = \mathbb{P}_x [B_t \in dy; t < \tau], \quad t > 0, x \in (-1, 1),$$

with $\tau = \inf \{t \geq 0; B_t \notin [-1, 1]\}$. We denote by $p^1(t, x, y)$ the density transition function defined by

$$p^1(t, x, y) dy = \mathbb{P}_x [B_t \in dy; t < \tau | B_\tau = 1].$$

Note also by q the distribution function of τ under \mathbb{P}_x and by q^1 its analogue for $p^1(t, x, \cdot)$. That is

$$q(t, x) = \mathbb{P}_x [\tau < t] = 1 - \int_{-1}^1 p(t, x, y) dy,$$

$$q^1(t, x) = \mathbb{P}_x [\tau < t | B_\tau = 1] = 1 - \int_{-1}^1 p^1(t, x, y) dy.$$

2.2 The density of the killed Brownian motion

It is well known that $p(t, x, y)$ is the solution of the following PDEs

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

Indeed, $p(t, x, y)$ can be expressed explicitly (see for example Beck *et al.* (1992); Milstein and Tretyakov (1999)). Using the method of image,

$$p(t, x, y) = \frac{1}{\sqrt{2\pi t}} \sum_{n=-\infty}^{+\infty} \left(\exp\left(-\frac{(x-y-4n)^2}{2t}\right) - \exp\left(-\frac{(x+y-2-4n)^2}{2t}\right) \right), \quad (3a)$$

while with a spectral representation we obtain

$$p(t, x, y) = \sum_{n=1}^{+\infty} \exp\left(-\frac{n^2\pi^2}{8}t\right) \sin\left(\frac{n\pi}{2}(x+1)\right) \sin\left(\frac{n\pi}{2}(y+1)\right). \quad (3b)$$

Note that from a numerical point of view, these series converge quickly so that only few terms are needed to get a good accuracy. Besides, it is worth using (3a) for small times and (3b) for large times.

For any $t > 0$, the distribution function of the exit time τ and its density can be deduced from the above expressions. For small times, we have by using (3a)

$$\begin{aligned} \mathbb{P}_x[\tau < t] &= 1 - \int_{-1}^1 p(t, x, y) dy = q(t, x) \\ &= 2 - \sum_{n=-\infty}^{+\infty} \left[F\left(\frac{x-1-4n}{\sqrt{2t}}\right) - F\left(\frac{x+1-4n}{\sqrt{2t}}\right) \right], \end{aligned} \quad (4)$$

where F stands for the following function $F(y) = \frac{2}{\sqrt{\pi}} \int_y^{+\infty} \exp(-z^2) dz$, (commonly noted erfc). For large times, we obtain, from (3b)

$$\mathbb{P}_x[\tau < t] = 1 - \frac{4}{\pi} \sum_{n=0}^{+\infty} \frac{(-1)^n}{2n+1} \exp\left(-\frac{\pi^2(2n+1)^2t}{8}\right) \cos\left(x\pi\left(n+\frac{1}{2}\right)\right). \quad (5)$$

We denote by $q_t(x, y)$ the derivative of $q(t, x)$ with respect to time, which is the probability density function of τ under \mathbb{P}_x . For small times, we get from (4) :

$$\begin{aligned} \mathbb{P}_x[\tau \in dt] = q_t(t, x)dt &= \frac{1}{\sqrt{2\pi t^3}} \sum_{n=-\infty}^{\infty} \left[(x+4n+1) \exp\left(-\frac{(x+4n+1)^2}{2t}\right) \right. \\ &\quad \left. - (x+4n-1) \exp\left(-\frac{(x+4n-1)^2}{2t}\right) \right] dt. \end{aligned} \quad (6)$$

Similarly we can get the expression for large times by using (5)

$$\begin{aligned} \mathbb{P}_x[\tau \in dt] &= \frac{\pi}{2} \sum_{n=0}^{\infty} (-1)^n (2n+1) \exp\left(-\frac{(2n+1)^2\pi^2}{8}t\right) \\ &\quad \times \cos\left(x\pi\left(n+\frac{1}{2}\right)\right) dt. \end{aligned} \quad (7)$$

Remark on the simulation

As we get both distribution function and density of the random variables τ under \mathbb{P}_x and B_t under $\mathbb{P}_x[\cdot | t < \tau]$, it is possible to get some realizations of these random variables by inverting their distribution functions via the Newton method (see Press *et al.* (1992) for example).

2.3 Conditioning by the exit position

We remark first that the function $S(x) = (1+x)/2$ is the *scale function* for the Brownian motion (see Breiman (1981) for example) and satisfies $\mathbb{P}_x[B_\tau = 1] = S(x)$. As S is a harmonic function for $\frac{1}{2}\Delta$, one may define for any $x \in (-1, 1)$ a new probability \mathbb{P}_x^1 by

$$\frac{d\mathbb{P}_x^1}{d\mathbb{P}_x} \Big|_{\mathcal{F}_t} = \frac{S(B_t)}{S(x)}.$$

Furthermore, for any \mathcal{F}_t -measurable random variable Φ , we have

$$\mathbb{E}_x^1[\Phi; t < \tau] = \frac{1}{S(x)} \mathbb{E}_x[\Phi S(B_\tau); t < \tau] = \mathbb{E}_x[\Phi; t < \tau \mid B_\tau = 1] = \frac{1}{S(x)}$$

since, by its definition, $S(B_\tau) = 1$ if $B_\tau = 1$ and $S(B_\tau) = 0$ if $B_\tau = -1$. It follows that the density transition function $p^1(t, x, y)$ of B under \mathbb{P}_x^1 is

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

2.4 Conditioning by the exit time

We now want to compute the density transition function of the Brownian motion conditioned by $\{\tau = T\}$, which is an event of null probability.

In order to do this, fix $0 < t < T$ and a point $x \in (-1, 1)$. Let f be a measurable, bounded function on $(-1, 1)$. For $0 < \varepsilon < T - t$, the Markov property implies that

$$\begin{aligned} \mathbb{E}_x[f(B_t) \mid T - \varepsilon < \tau < T + \varepsilon] &= \\ &= \frac{\mathbb{E}_x[f(B_t); t < \tau \text{ and } |\tau - T| < \varepsilon]}{\mathbb{P}_x[T - \varepsilon < \tau < T + \varepsilon]} \\ &= \frac{\int_{-1}^1 p(t, x, y) \int_{T-\varepsilon}^{T+\varepsilon} q_t(s-t, y) f(y) ds dy}{\int_{T-\varepsilon}^{T+\varepsilon} q_t(s, x) ds} \\ &\xrightarrow{\varepsilon \rightarrow 0} \int_{-1}^1 r(t, x, y) f(y) dy \text{ with } r(t, x, y) = p(t, x, y) \frac{q_t(T-t, y)}{q_t(T, x)}. \end{aligned} \tag{8}$$

It follows that the distribution of B_t given $\{\tau = T\}$ has the density $r(t, x, y)$. In fact, one may pursue this analysis and show that $(B_t)_{t \in [0, T]}$ given $\{\tau = T\}$

is a non-homogeneous Markov process with density $r(t, x, y)$ and distribution $\mathbb{Q}_{x,T}$ defined by the density

$$\frac{d\mathbb{Q}_{x,T}}{d\mathbb{P}_x[\cdot | T < \tau]} \Big|_{\mathcal{F}_t} = (1 - q(T, x)) \frac{q_t(T - t, B_t)}{q_t(T, x)}$$

with respect to $\mathbb{P}_x[\cdot | T < \tau]$.

Similarly, one may define the distribution $\mathbb{Q}_{x,T}^1[\cdot] = \mathbb{P}_x[\cdot | \tau = T, B_\tau = 1]$ as being the distribution of a process with density transition function

$$r^1(t, x, y) = p^1(t, x, y) \frac{q_t^1(T - t, y)}{q_t^1(T, x)}.$$

Remarks on the simulation

It is quite costly to simulate the random variable B_t under the law $\mathbb{Q}_{x,T}^1$ (with $T > t$) since there is no “nice” analytical expression for its distribution function.

In order to simulate a random variable Y with density $r^1(t, x, \cdot)$, we can use the rejection method described in Devroye (1986, § 3.3, p. 47) by writing

$$r^1(t, x, y) = c \frac{p^1(t, x, y)}{1 - q^1(t, x)} \varphi(t, T, x, y)$$

with $\varphi(t, T, x, y) = (1 - q^1(t, x))q_t^1(T - t, y)/(cq_t^1(T, x))$ and c a real number such that $c \geq \max\{1, \sup_{y \in (-1,1)} \varphi(t, T, x, y)\}$. The idea of the algorithm is to simulate a realization \bar{x} of a random variable with density $p^1(t, x, \cdot)/(1 - q^1(t, x))$ until $\varphi(t, T, x, \bar{x})/c$ becomes greater than the realization of a uniform random variable on $[0, 1]$. As soon as this condition is satisfied, \bar{x} is a realization of a random variable with density $r^1(t, x, \cdot)$.

2.5 Dealing with a general interval

The scaling and invariance by translation properties of the Brownian motion allow to replace easily $[-1, 1]$ by any interval $[a, b]$. A simple way to do it is to use the following identity

$$\begin{aligned} & \mathbb{P}_x \left[(\tau_{a,b}, B_{\tau_{a,b}}) \in \cdot \right] \\ &= \mathbb{P}_{\frac{2(x-a)}{(b-a)} - 1} \left[\left(\left(\frac{(b-a)^2}{4} \tau_{-1,1}, (b-a) \frac{1 + B_{\tau_{-1,1}}}{2} + a \right)^{-1} \in \cdot \right) \right], \end{aligned}$$

where $\tau_{a,b} = \inf \{t \geq 0 ; B_t \notin [a, b]\}$.

3 Simulating the first exit time and position for a 2-dimensional Brownian motion

3.1 Analytical expressions

In this Section, we consider a 2-dimensional Brownian motion $(\beta, \mathbb{P}_x, (\mathcal{F}_t)_{t \geq 0})$ which is killed when it exits from a general domain D with a piecewise smooth boundary. We denote by τ' its first exit time.

The density $K_D(t, y)$ with respect to $dt d\sigma_y$ of $(\tau', B_{\tau'})$ under \mathbb{P}_x is given by

$$K_D(t, x, y) = \frac{\partial p_D}{\partial n}(t, x, y),$$

where n is the inward unit normal vector on the boundary ∂D , $\frac{\partial}{\partial n}$ denotes the derivative of p_D in this direction, σ_y is the Lebesgue measure on ∂D and $p_D(t, x, y)$ is the fundamental solution of $\frac{1}{2}\Delta$, that is the solution to

$$\begin{cases} \frac{\partial p_D(t, x, y)}{\partial t} = \frac{1}{2}\Delta_y p_D(t, x, y) \text{ on } \mathbb{R}_+^* \times D, \\ p_D(t, x, y) = 0 \text{ for } y \in \partial D, \\ p_D(t, x, y) \xrightarrow[t \rightarrow 0]{} \delta_x(y). \end{cases} \quad (9)$$

If $D = [-L, L] \times [-\ell, \ell]$ is a rectangle, then an analytical expression for $p_D(t, x, y)$ can be easily obtained through some spectral analysis (see for example Zauderer (1983, p. 389)).

For $n, m \in \mathbb{N}^*$, the eigenvalues of $\frac{1}{2}\Delta$ are given by

$$\lambda_{n,m} = \frac{1}{8} \left[\left(\frac{\pi n}{L} \right)^2 + \left(\frac{\pi m}{\ell} \right)^2 \right]$$

and the corresponding eigenfunctions are

$$\varphi_{n,m}(\xi, \eta) = \frac{1}{\sqrt{\ell L}} \sin\left(\frac{\pi n}{2L}(\xi + L)\right) \sin\left(\frac{\pi m}{2\ell}(\eta + \ell)\right) \text{ with } (\xi, \eta) \in D^2.$$

Here, the $\varphi_{n,m}$ are such that $\int_D \varphi_{n,m}(\xi, \eta) \varphi_{n',m'}(\xi, \eta) d\xi d\eta = \delta_{\lambda_{n,m}=\lambda_{n',m'}}$. Note that the eigenvalues may have a multiplicity greater than 1.

For (ξ, η) and (ξ', η') in D , the solution $p_D(t, (\xi, \eta), (\xi', \eta'))$ of (9) is then

$$\begin{aligned} p_D(t, (\xi, \eta), (\xi', \eta')) &= \frac{1}{\ell L} \sum_{n,m \geq 1} e^{-\lambda_{n,m}t} \sin\left(\frac{\pi n}{2L}(\xi' + L)\right) \sin\left(\frac{\pi n}{2L}(\xi + L)\right) \\ &\quad \times \sin\left(\frac{\pi m}{2\ell}(\eta' + \ell)\right) \sin\left(\frac{\pi m}{2\ell}(\eta + \ell)\right). \end{aligned} \quad (10)$$

So, one may compute easily $K_D(t, x, y)$ from (10). However, this formula is not practical when one wants to simulate directly (τ, B_τ) since this series uses two indices and the result is a realization of a random variable taking its values in $\mathbb{R} \times \partial D \subset \mathbb{R}^3$. This is why we introduce another approach that relies on conditioning and that leads to faster computations.

In some cases, one does not need the density $p_D(t, x, y)$ of the parabolic problem, but simply the Green function $G_{\alpha, D}$ of the elliptic problem

$$\left(\alpha - \frac{1}{2}\Delta\right) G_{\alpha, D}((\xi, \eta), (\xi', \eta')) = \delta_{(\xi, \eta)}((\xi', \eta')) \text{ and } G_{\alpha, D}(\xi', \eta') = 0 \text{ on } \partial D$$

for some $\alpha \geq 0$. This is the case for example when solving the Dirichlet problem $\frac{1}{2}\Delta u(x) = f(x)$ on D with $u(x) = \varphi(x)$ on ∂D , where

$$u(x) = \int_D G_{0, D}(x, y) f(y) dy + \int_{\partial D} \frac{\partial G_{0, D}}{\partial n}(x, y) \varphi(y) d\sigma_y.$$

The density of $B_{\tau'}$ with respect to $d\sigma_y$ is indeed given by $\frac{\partial G_{0, D}}{\partial n}(x, y)$.

The Green function $G_{\alpha, D}$ may be obtained through the Laplace transform of p_D :

$$G_{\alpha, D}(x, y) = \int_0^{+\infty} e^{-\alpha t} p_D(t, x, y) dt \text{ for } (x, y) \in D,$$

which is easy to compute when D is a rectangle: in (10), one has only to replace the term $\exp(-\lambda_{n, m} t)$ by $(\alpha + \lambda_{n, m})^{-1}$.

By combining an eigenvalue problem in dimension one together with a decomposition on a Fourier basis (See Zauderer (1983, Example 7.5, p. 389) for example in the case $\alpha = 0$), one can write down another formula for the Green function which is more suitable for numerical computations.

3.2 The case of a rectangular domain

In this section, we propose a new method to obtain a realization of the first exit time and position from a rectangle, that does not rely on (10).

Let ℓ, L be some positive numbers. Let $(\beta, \mathbb{P}_{(x, y)})$ be a two-dimensional Brownian motion with $\beta = (\beta^1, \beta^2)$. We are interested in simulating $(\tau', \beta_{\tau'})$ with

$$\tau' = \inf \{t > 0 ; \beta_t \notin [-L, L] \times [-\ell, \ell]\},$$

when the particle is initially at (x, y) .

In the following algorithm, (B, \mathbb{P}_x) denotes a one-dimensional Brownian motion and $\tau = \inf \{t > 0 ; B_t \notin [-1, 1]\}$.

The command `unif()` draws independently uniform random variable on $[0, 1]$. The command `real(Z)` draws independently a realization of the random

variable Z with an arbitrary distribution. The notation \leftarrow is used to allocate values to variables.

This algorithm returns (t, u, v) , where t is the exit time, and $(u, v) \in \mathbb{R}^2$ is the exit position.

Algorithm A

1. [Computing the probability that the exit point from $[-L, L]$ of β^1 is L]
 $\alpha_1 \leftarrow \mathbb{P}_{x/L} [B_\tau = 1]$.
 2. [Choosing the exit position of β^1]
 If $\text{unif}() < \alpha_1$ then $\varepsilon_x \leftarrow 1$ else $\varepsilon_x \leftarrow -1$.
 3. $u_x \leftarrow \varepsilon_x L$.
 4. [Simulating the exit time of β^1]
 $t_x \leftarrow L^2 \text{real}(Z)$ where $\mathbb{P}[Z \in \cdot] = \mathbb{P}_{\varepsilon_x x/L} [\tau \in \cdot \mid B_\tau = 1]$.
 5. [Computing the probability that β^2 reaches $\{-\ell, \ell\}$ before β^1 reaches $\{-L, L\}$]
 $\alpha_2 \leftarrow \mathbb{P}_{y/\ell} [\tau < t_x/\ell^2]$.
 6. [If β^2 reaches $\{-\ell, \ell\}$ before β^1 reaches $\{-L, L\}$]
 If $\text{unif}() < \alpha_2$ then
 - (a) [Computing the probability that β^2 reaches ℓ before $-\ell$]
 $\alpha_3 \leftarrow \mathbb{P}_{y/\ell} [B_\tau = 1 \mid \tau < t_x/\ell^2]$.
 - (b) [Choosing if β^2 reaches ℓ before $-\ell$]
 If $\text{unif}() < \alpha_3$ then $\varepsilon_y \leftarrow 1$ else $\varepsilon_y \leftarrow -1$.
 - (c) $u_y \leftarrow \varepsilon_y \ell$.
 - (d) [Simulating the exit time θ_y of $[-\ell, \ell]$ for β^2 given the exit position]
 $\theta_y \leftarrow \ell^2 \text{real}(Z)$ where $\mathbb{P}[Z \in \cdot] = \mathbb{P}_{\varepsilon_y y/\ell} [\tau \in \cdot \mid \tau < t_x/\ell^2, B_\tau = 1]$.
 - (e) [Simulating the position of β^1 at time θ_y]
 $z \leftarrow \varepsilon_x L \text{real}(Z)$ with $\mathbb{P}[Z \in \cdot] = \mathbb{P}_{\varepsilon_x x/L} [B_{\theta_y/L^2} \in \cdot \mid B_\tau = 1, \tau = t_x/L^2]$.
 - (f) Return (θ_y, z, u_y) .
- else [β^1 reaches $\{-L, L\}$ before β^2 reaches $\{-\ell, \ell\}$].
- (a) [Simulating the position of β^2 at time t_x]
 $z \leftarrow \ell \text{real}(Z)$ with $\mathbb{P}[Z \in \cdot] = \mathbb{P}_{y/\ell} [B_{t_x} \in \cdot \mid t_x/\ell^2 < \tau]$.
 - (b) Return (t_x, u_x, z) .
-

Remark 1. For $x \in (-1, 1)$ and $t > 0$, we have

$$\mathbb{P}_x [B_\tau = 1 \mid \tau < t] = \mathbb{P}_x [\tau < t \mid B_\tau = 1] \frac{\mathbb{P}_x [B_\tau = 1]}{\mathbb{P}_x [\tau < t]} = \frac{q^1(t, x)}{q(t, x)} \frac{1+x}{2}.$$

Remark 2. This algorithm takes more time if β^2 reaches $\{-\ell, \ell\}$ before β^1 reaches $\{-L, L\}$. Otherwise, it would have been equivalent to simulate the exit time of β^1 and β^2 , and then to simulate the position of β given these exit times. It is possible to speed up by inverting the x and y coordinates when $L/2 - |x| > \ell/2 - |y|$, *i.e.*, when the particle is closer to the horizontal side than to the vertical side. For $L = \ell$ and $(x, y) = (0, 0)$, we have to deal with a square. One may then use some symmetry property in order to avoid the computation of the position of the Brownian motion given the value of the exit time: see Milstein and Tretyakov (1999); Campillo and Lejay (2002).

3.3 General domains — construction of boxes

Consider now that the domain is the interior of a general polygon. It follows from the Markov property that the first exit time and position from this domain for the Brownian motion may be simulated by simulating iteratively the first exit times and positions from rectangles containing the particle, until it reaches the boundary.

3.3.1 Choosing the rectangles

As said in the introduction, the rectangles may be constructed at the beginning or dynamically. As the rectangles are not necessarily centered on a current position of the particle, there are many choices and yet no general rule to choose the “best configuration”. One should have in mind that the computational cost of the algorithm is connected to the number of steps needed by the particle to reach the boundary (this issue is discussed in Section 4 on a numerical test). Of course, constructing the rectangle before any simulation allows to speed up the computations.

A suitable set of rectangle should have the following properties:

- (1) Any point of the domain belongs to one or more rectangles;
- (2) Any point on the boundary of some rectangle (except those that are also on the boundary of the domain) belongs to the interior of at least another rectangle;
- (3) Any point on the boundary of the domain is also on the boundary of at least a rectangle;

- (4) The number of rectangles is as small as possible.

If our method is combined with a dynamical choice of a rectangle or a sphere, then there is no need for all these conditions to be satisfied. Indeed, it is impossible to construct a set of rectangles satisfying (1)–(3) for domain with “sharp” angles. An alternative method consists in changing the domain slightly (see below).

3.3.2 A method for constructing a suitable set of rectangles

There is a rather easy algorithm to construct a suitable set of rectangles.

The idea is to construct on each edge of the polygon the rectangle with maximal surface, based on this edge and which is included in the polygon. After this construction we have a number of rectangles which cover the domain. A particular treatment has to be done when an angle is less than $\pi/2$. For each situation of this type we shall “cut” the polygon by a small triangle based on this angle and distance ε from the vertex on which the angle is based (See Figure 1, where the domain is the shaded region).



Figure 1: Transforming the domain to suppress sharp edges

At each operation of this sort we introduce a new edge. After this we get a polygon, closely related to the initial one, and for which each maximal rectangle (as defined before) is based on the entire vertices.

The next step is: for each starting point x find the largest rectangle from the above construction which contains x . After that, do the calculation (the algorithm presented above) for this rectangle and the point x . Continue this procedure with the exit point found.

The algorithm ends when we leave first the initial domain.

4 A numerical test

We have performed a numerical test on a simple polygonal domain. In Figure 2, we represent the density of the exit position for each side. Five rectangles were used, each one being as big as possible and having at least (there is a right angle in the polygon) one side equal to one side of the polygon.

As any point of the polygon may be contained in more than one rectangle, we have to decide which one to use. We have thus tested our algorithm for different methods

- (a) using the rectangle with the biggest area;
- (b) using a rectangle having a side intersecting the boundary which is the closest to the current position;
- (c) using the rectangle having a side which is the closest to the current position;
- (d) using the rectangle whose boundary is at a maximal distance from the current position;
- (e) choosing the rectangle randomly;
- (f) using the rectangle whose side on the boundary is the largest.

Here are only a few criteria, but other may be used, or obtained by any combination of the previous ones.

We give in Table 1 the average number of steps together with the variance for the different ways of choosing the rectangles. As for the histograms of Figure 2, 10,000 particles are used. The first two lines refer to the case where the starting point is the one marked by the cross \times , while the last two lines refer to the case where the starting point is marked by the dot \bullet in Figure 2.

experiment	(a)	(b)	(c)	(d)	(e)	(f)
mean \times	4.30	3.30	5.40	3.25	3.93	4.02
variance \times	4.64	1.40	6.60	1.30	2.70	3.02
mean \bullet	4.22	2.45	4.42	2.45	3.34	3.70
variance \bullet	2.28	1.70	3.77	0.65	1.67	1.41

Table 1: Mean and variance for the number of rectangles used to reach the boundary.

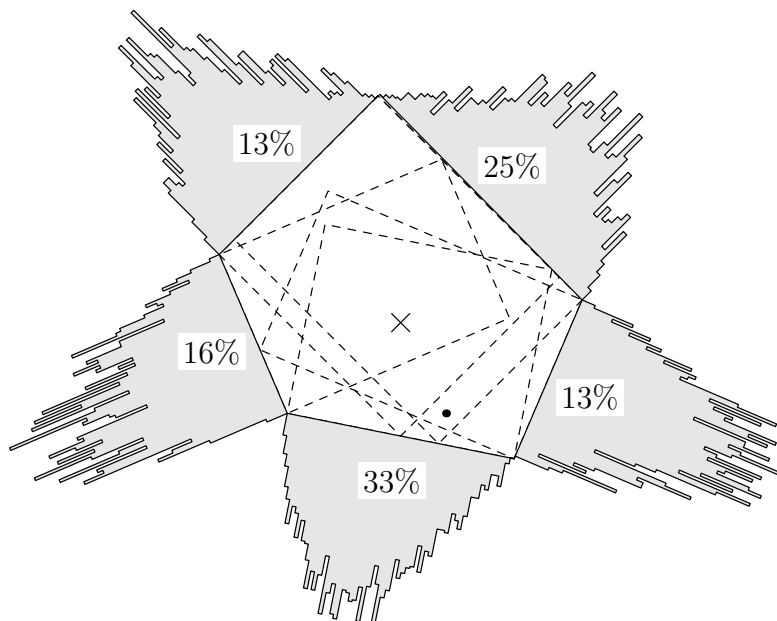


Figure 2: Histograms of density of the exit position conditioned by the exit side of the polygon (10,000 particles were used). The percentage represents the proportion of particles hitting this side. The dashed lines represent the boundary of the rectangles. The starting point is marked by the cross \times .

Obviously, the methods (b) and (d) provide the best choices, which can be explained intuitively: The closest is the side of a rectangle, the higher is the probability to exit by this side, so that (b) increases the probability that the algorithm stops quickly. While with (d), if the particle is far from the boundary of the rectangles, then it is far from the boundary of the domain. At the next step, it will be close from the boundary of the domain with a high probability.

5 Possible extensions

Of course, the previous algorithm may be adapted to deal with slightly different problems (that may be combined):

- The rectangle may be replaced by parallelepipeds when the domain has dimension d .

- One may solve the elliptic (resp. parabolic) equation

$$\begin{aligned} & \frac{1}{2}\Delta u(x) = f(x) \text{ on } D, \quad u(x) = \varphi(x) \text{ on } \partial D \\ \text{resp. } & \begin{cases} \frac{\partial u(t, x)}{\partial t} + \frac{1}{2}\Delta u(t, x) = f(t, x) \text{ on } [0, T] \times D, \\ u(t, x) = \psi(t, x) \text{ on } [0, T] \times \partial D, \quad u(T, x) = \varphi(x) \end{cases} \end{aligned}$$

by setting

$$\begin{aligned} u(x) &= \mathbb{E}_x \left[\sum_{k=0}^{n^*-1} \int_{R_k} G_{0, R_k}(\beta_{\theta_k}, y) f(y) \, dy + \varphi(\beta_{\theta_{n^*}}) \right] \\ \text{resp. } u(t, x) &= \mathbb{E}_{t, x} \left[\varphi(\beta_T) \mathbf{1}_{\{T < \theta_{n^*}\}} + \psi(\theta_{n^*}, \beta_{\theta_{n^*}}) \mathbf{1}_{\{T > \theta_{n^*}\}} \right. \\ & \quad \left. + \sum_{k=0}^{n^*-1} \int_{\theta_k \wedge T}^{\theta_{k+1} \wedge T} \int_{R_k} p_{R_k}(s, T, \beta_{\theta_k}, y) f(s, y) \, ds \, dy \right], \end{aligned}$$

where G_{0, R_k} (resp. p_{R_k}) is the Green function (resp. the fundamental solution or the parabolic Green function) of the Laplace operator for a rectangle R_k containing $\beta_{\theta_{k-1}}$ (see Section 3.1 for analytical expressions), θ_k is the exit time from this rectangle R_k when the Brownian motion is at $\beta_{\theta_{k-1}}$, $\theta_0 = 0$ (resp. $\theta_0 = t$), $\beta_{\theta_0} = x$ and n^* is the step at which the boundary is hit by the Brownian motion. When $T < \theta_{n^*}$, it is possible to simulate β_T using the algorithm B in Section 5.1.

Of course, the computations are simpler if $f = 0$ in the elliptic or the parabolic equations. If it is not the case, computing the integrals of type $\int_{R_k} G_{0, R_k}(\beta_{\theta_k}, y) f(y) \, dy$ and $\int_{\theta_k \wedge T}^{\theta_{k+1} \wedge T} \int_{R_k} p_{R_k}(s, T, \beta_{\theta_k}, y) f(s, y) \, ds \, dy$ may be costly. Yet, as shown in DeLaurentis and Romero (1990), G_{0, R_k} and p_{R_k} may also be considered as densities of random variables and a Monte Carlo method may then be used also to compute these integrals.

- One may solve $(\alpha - \frac{1}{2}\Delta)u(x) = 0$ on D with $u(x) = \varphi(x)$ on ∂D by setting $u(x) = \mathbb{E}[\exp(-\alpha\tau)\varphi(B_\tau)]$, where $\tau = \inf\{t > 0; B_t \in \partial D\}$.
- One may assume that a homogeneous Neumann boundary condition holds instead of a Dirichlet boundary condition on some part of ∂D : See Section 5.2.
- One may replace $\frac{1}{2}\Delta$ by $\frac{1}{2}\Delta + \mu\nabla$, where μ is a constant vector of \mathbb{R}^d : See Section 5.3.
- One may replace $\frac{1}{2}\Delta$ by $\sum_{i,j=1}^d \frac{1}{2} a_{i,j} \frac{\partial^2}{\partial x_i \partial x_j}$, where a is a symmetric matrix which is positive-definite. In this case, one has to consider a Brownian motion in the domain $a^{-1}D$ and then to apply a to its terminal position.

5.1 The position of the Brownian motion at a given time

Assume that we are interested in solving

$$\begin{cases} \frac{\partial u}{\partial t}(t, x) + \frac{1}{2}\Delta u(t, x) = 0 \text{ on } [0, T] \times D, \\ u(T, x) = f(x) \text{ on } D, \\ u(t, x) = \psi(t, x) \text{ on } [0, T] \times \partial D. \end{cases}$$

Then $u(t, x)$ is given by

$$u(t, x) = \mathbb{E}_{t,x} [f(B_T); T < \tau] + \mathbb{E}_{t,x} [\psi(\tau, B_\tau); \tau \leq T].$$

In this case, the algorithm is close to the previous one. Obviously, we are reduced to the case of simulating $(\tau \wedge T, B_{\tau \wedge T})$, where τ is the first exit time from $[-L, L] \times [-\ell, \ell]$.

Algorithm B

1. [Computing the probability that the exit point from $[-L, L]$ of β^1 is L]
 $\alpha_1 \leftarrow \mathbb{P}_{x/L} [B_\tau = 1]$.
2. [Choosing the exit position of β^1]
 If $\text{unif}() < \alpha_1$ then $\varepsilon_x \leftarrow 1$ else $\varepsilon_x \leftarrow -1$.
3. $u_x \leftarrow \varepsilon_x L$.
4. [Simulating the exit time of β^1]
 $t_x \leftarrow L^2 \text{real}(Z)$ where $\mathbb{P}[Z \in \cdot] = \mathbb{P}_{\varepsilon_x x/L} [\tau \in \cdot \mid B_\tau = 1]$.
5. [Comparing the exit time of β^1 with T]
 If $t_x < T$ then continue from Step 6 of Algorithm A. Otherwise, continue with the following algorithm.
6. [Computing the probability that β^2 reaches $\{-\ell, \ell\}$ before β^1 reaches $\{-L, L\}$ or before T]
 $\alpha_2 \leftarrow \mathbb{P}_{y/\ell} [\tau < t_x/\ell^2]$, $\alpha'_2 \leftarrow \mathbb{P}_{y/\ell} [\tau < T/\ell^2]$ and $u \leftarrow \text{unif}()$.
7. [If β^2 reaches $\{-\ell, \ell\}$ before T]
 If $u < \alpha'_2$ then continue as in Steps 6a–6f of Algorithm A by replacing t_x by T except at Step 6e.
8. [If β^2 reaches $\{-\ell, \ell\}$ after T but before β^1 reaches $\{-L, L\}$]
 else if $u \in (\alpha'_2, \alpha)$ then

- (a) [Computing the position of β^2 at time T]
 $u_y \leftarrow L\text{real}(Z)$ with $\mathbb{P}[Z \in \cdot] = \mathbb{P}_{y/\ell}[B_T \in \cdot \mid T/\ell^2 < \tau < t_x/\ell^2]$.
- (b) [Simulating the position of β^1 at time T]
 $z \leftarrow \varepsilon_x L\text{real}(Z)$ with $\mathbb{P}[Z \in \cdot] = \mathbb{P}_{\varepsilon_x x/L}[B_{T/L^2} \in \cdot \mid B_\tau = 1, \tau = t_x/L^2]$.
- (c) Return (T, z, u_y) .

else [β^1 reaches $\{-L, L\}$ before β^2 reaches $\{-\ell, \ell\}$ and thus $T > \tau$]

- (a) [Simulating the position of β^2 at time t_x]
 $z \leftarrow \ell\text{real}(Z)$ with $\mathbb{P}[Z \in \cdot] = \mathbb{P}_{y/\ell}[B_{t_x} \in \cdot \mid t_x/\ell^2 < \tau]$.
- (b) Return (t_x, u_x, z) .

5.2 The Neumann boundary condition

If some zero Neumann boundary conditions are imposed on some part ∂D^N of the boundary ∂D , then our algorithms can easily be adapted.

For that, for each rectangle, we have to assume that any side S contained in ∂D satisfies either $S \subset \partial D^N$ or $S \cap \partial D^N = \emptyset$.

All our construction relies on the fact that one knows explicit expressions for the fundamental solution $p(t, x, y)$ of $\frac{\partial}{\partial t} - \frac{1}{2}\Delta$ with the Dirichlet boundary conditions at -1 and 1 . Analytical expressions for $p(t, x, y)$ are available also when one of the Dirichlet boundary condition (or both) is replaced by a Neumann boundary condition: See Beck *et al.* (1992, Appendix X, p. 473) for some formulas of this kind. It is then possible to adapt Algorithms A and B when more than one side of the rectangle are subject to Neumann boundary condition.

Yet it is also possible to use the symmetry of the problem. Assume for example that the side $\{0\} \times [0, \ell]$ of $[0, L] \times [0, \ell]$ is contained in ∂D^N . Then one simulates (τ, B_τ) , where τ is the first exit time from the doubled rectangle $[-L, L] \times [0, \ell]$. The first exit time and position from $[0, L] \times [0, \ell]$ is simply given by $(\tau, \alpha(B_\tau))$ with $\alpha(x, y) = (|x|, y)$. If more than one side of the rectangle is contained in ∂D^N , then this kind of construction also works. But this fails for the case of the Brownian motion with drift (another method to solve this problem is given in Section 5.3.2).

5.3 Brownian motion with a constant drift

Our algorithm can be used also to simulate the first exit time and position from a polygonal domain for a Brownian motion with a constant drift.

There are two possibilities. It is possible to let a Brownian particle evolve in the media and then use the Girsanov theorem as in Section 5.3.1.

Another way consists in using densities related to the Brownian motion with drift in Algorithm A: When computing the first exit time and position from a rectangle for a 2-dimensional Brownian motions with a drift μ , one has to consider two one-dimensional Brownian motion with drifts : $B_t^1 + \mu^1 t$ and $B_t^2 + \mu^2 t$, where μ^1 and μ^2 are the projections of μ in the direction of the edges of the rectangle. It is then possible to get some explicit expressions for the density of the Brownian motion with drift killed when it exits from $[-1, 1]$. We give in Section 5.3.2 some ideas to compute the involved densities.

5.3.1 Using the Girsanov theorem

Consider the drifted Brownian motion $X_t = B_t + \mu t$, where B is a 1-dimensional Brownian motion and $\mu \in \mathbb{R}$. As the goal is to compute some quantities like $\mathbb{E}_{t,x}^\mu [\psi(\tau, X_\tau)]$, where τ is the first exit time from D for X (here, $\mathbb{P}_{t,x}^\mu$ denotes the distribution of a Brownian motion with drift μ starting from x at time t), another method consists in simulating (τ, B_τ) for a Brownian motion B starting at (t, x) and then to use the Girsanov formula:

$$\mathbb{E}_{t,x}^\mu [\psi(\tau, X_\tau)] = \mathbb{E}_{t,x} \left[\psi(\tau, B_\tau) \exp \left(\mu B_\tau - \frac{1}{2} \mu^2 \tau \right) \right].$$

5.3.2 Laws related to the Brownian motion with drift

We give briefly some methods to compute the density of the Brownian motion with drift $X_t = B_t + \mu t$, where B is a 1-dimensional Brownian motion and $\mu \in \mathbb{R}$.

Dirichlet boundary conditions. Let us remark that the density $p^\mu(t, x, y)$ of X killed when it exits from $[-1, 1]$ is given by the relation

$$p^\mu(t, x, y) = \exp \left(\mu \cdot (y - x) - \frac{\mu^2}{2} t \right) p(t, x, y). \quad (11)$$

Moreover, if $\tau = \inf \{t > 0 ; |X_t| = 1\}$, then

$$\mathbb{P}_x^\mu [X_\tau = 1] = S^\mu(x) \text{ with } S^\mu(x) = \frac{e^{2\mu} - e^{-2\mu x}}{e^{2\mu} - e^{-2\mu}}.$$

All the analytical expressions for the distribution function or the density of τ and X_τ (possibly conditioned to some event) can then be transformed to include the drift term μ .

Neumann boundary condition The transform (11) no longer works when a Neumann boundary condition is imposed at one or both endpoints of $[-1, 1]$. Nevertheless, it is possible to get some analytical expression for $p^\mu(t, x, y)$.

To do this we can use the spectral representation of $p^\mu(t, x, y)$

$$p^\mu(t, x, y) = \sum_{k \geq 0} e^{-\lambda_k t} \varphi_k(x) \varphi_k(y),$$

where the φ_k 's are the normalized eigenfunctions associated to the eigenvalues $0 \leq \lambda_0 \leq \lambda_1 \leq \dots$ of the Sturm-Liouville equation (see Eastham (1970) for example)

$$\frac{1}{2} \varphi''(x) + \mu \varphi'(x) + \lambda \varphi(x) = 0 \text{ on } [-1, 1],$$

with proper boundary conditions at -1 and 1 . The functions $\varphi_k(x)$ have the form $\varphi_k(x) = C_+ e^{\alpha_+ x} + C_- e^{\alpha_- x}$ where α_+ and α_- are the two solutions of $\alpha^2 + 2\mu\alpha + 2\lambda = 0$. The suitable values of λ are found when there exists a couple $(C_+, C_-) \neq (0, 0)$ for which φ_k satisfies the prescribed boundary conditions. It is then possible to compute numerically, if not analytically, the eigenvalues λ_k and the eigenfunctions φ_k .

Two different explicit formulae for $p^\mu(t, x, y)$ with Neumann boundary conditions at both -1 and 1 may be found in the articles Linetsky (2005) (for the spectral problem) and Veestraeten (2005) (where a formula similar to (3a) is given by inverting Laplace transforms).

References

- J.V. Beck, K.D. Cole, A. Haji-Sheikh and B. Litkouhi, *Heat conduction using Green's functions*, Series in Computational and Physical Processes in Mechanics and Thermal Sciences. Hemisphere Publishing Corp., London, 1992. 4, 17
- L. Breiman, *Probability*, Addison-Wesley, 1981. 6
- F. Campillo and A. Lejay, "A Monte Carlo method without grid for a fractured porous domain model," *Monte Carlo Methods Appl.* vol. 8(2) pp. 129–148, 2002. 2, 3, 11
- J.M. DeLaurentis and L. A. Romero, "A Monte Carlo method for Poisson's equation," *J. Comput. Phys.* vol. 90(1) pp. 123–140, 1990. 15
- L. Devroye, *Non-Uniform Random Variate Generation*, Springer-Verlag, 1986. 7

- M.S.P. Eastham, *Theory of ordinary differential equations*, Van Nostrand Reinhold Company, 1970. 19
- N. Golyandina, “Convergence rate for spherical processes with shifted centres,” *Monte Carlo Methods Appl.* vol. 10(3–4):287–296, 2004. Conference proceeding of *IV IMACS Seminar on Monte Carlo Methods*. 2
- C.-O. Hwang, M. Mascagni and J. A. Given, “A Feynman-Kac path-integral implementation for Poisson’s equation using an h -conditioned Green’s function,” *Math. Comput. Simulation* vol. 62(3-6) pp. 347–355, 2003. 3rd IMACS Seminar on Monte Carlo Methods—MCM 2001 (Salzburg). 2
- V. Linetsky, “On the transition densities for reflected diffusions,” *Adv. in Appl. Probab.* vol. 37(2) pp. 435–460, 2005. 19
- S. Maire, *Réduction de variance pour l’intégration numérique et pour le calcul critique en transport neutronique*, PhD thesis, Université de Toulon et de Var (France), 2001. 3
- G.N. Mil’shtejn and N.F. Rybkina, “An algorithm for random walks over small ellipsoids for solving the general Dirichlet problem,” *Comput. Math. Math. Phys.* vol. 33(5) pp. 631–647, 1993. 2
- G.N. Milstein and M.V. Tretyakov, “Simulation of a space-time bounded diffusion,” *Ann. Appl. Probab.* vol. 9(3) pp. 732–779, 1999. 2, 3, 4, 11
- M. E. Muller, “Some continuous Monte Carlo methods for the Dirichlet problem,” *Ann. Math. Statist.* vol. 27 pp. 569–589, 1956. 2
- B. Noëtinger and T. Estébenet, “Up-Scaling of Double Porosity Fractured Media Using Continuous-Time Random Walks Methods,” *Transport in Porous Media* vol. 39(3) pp. 315–337, 2000. 3
- W.H. Press, S.A Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C*, Cambridge University Press, 2nd ed., 1992. 5
- K. Sabelfeld, I. Shalimova and A.I. Levykin, “Discrete random walk on large spherical grids generated by spherical means for PDEs,” *Monte Carlo Methods Appl.* vol. 10(3–4):559–574, 2004. Conference proceeding of *IV IMACS Seminar on Monte Carlo Methods*. 2
- K.K. Sabelfeld and D. Talay, “Integral formulation of the boundary value problems and the method of random walk on spheres,” *Monte Carlo Methods Appl.* vol. 1(1) pp. 1–34, 1995. 2

- N.A. Simonov and M. Mascagni, “Random walk algorithm for estimating effective properties of digitized porous media,” *Monte Carlo Methods Appl.* vol. 10(3–4):599–608, 2004. Conference proceeding of *IV IMACS Seminar on Monte Carlo Methods*. 3
- S. Torquato and C. Kim, “Effective conductivity, dielectric constant, and diffusion coefficient of digitized composite media via first-passage-time equations,” *J. Appl. Phys.* vol. 85(3) pp. 1560–1571, 1999. 2, 3
- D. Veestraeten, “The conditional probability density function for a reflected Brownian motion,” *Comput. Econom.* vol. 24(2) pp. 185–207, 2005. 19
- E. Zauderer, *Partial differential equations of applied mathematics*. Pure and Applied Mathematics. John Wiley & Sons Inc., New York, 1983. 8, 9