Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2015-0016

Vol. **20**, No. 3, pp. 703-732
September 2016

A Comparison of Higher-Order Weak Numerical Schemes for Stopped Stochastic Differential Equations

Francisco Bernal[1,2] and Juan A. Acebrón[3,1,*]

[1] *INESC-ID\IST, TU Lisbon. Rua Alves Redol 9, 1000-029 Lisbon, Portugal.*
[2] *Center for Mathematics and its Applications, Department of Mathematics, Instituto Superior Técnico. Av. Rovisco Pais 1049-001 Lisbon, Portugal.*
[3] *ISCTE - Instituto Universitário de Lisboa Departamento de Ciências e Tecnologias de Informação. Av. das Forças Armadas 1649-026 Lisbon, Portugal.*

Received 1 December 2015; Accepted (in revised version) 8 March 2016

Abstract. We review, implement, and compare numerical integration schemes for spatially bounded diffusions stopped at the boundary which possess a convergence rate of the discretization error with respect to the timestep h higher than $\mathcal{O}(\sqrt{h})$. We address specific implementation issues of the most general-purpose of such schemes. They have been coded into a single Matlab program and compared, according to their accuracy and computational cost, on a wide range of problems in up to $\mathbb{R}^{48}$. The paper is self-contained and the code will be made freely downloadable.

AMS subject classifications: 60G50, 60-08, 65Cxx, 65-02

Key words: Weak convergence, Feynman-Kac, stochastic differential equation, bounded diffusion, first-exit problem.

1 Introduction

Producing numerical approximations, via Monte Carlo simulations, to the expected value of functionals involving stochastic differential equations (SDEs) stopped at a boundary is pervasive in scientific computing. Usually, the quantities of interest are the mean first passage time of a walker from that domain [33] or the pointwise solution of a boundary value problem (BVP) [9]. Some relevant fields where it encounters application are: biology [35], barrier problems in finance [13], mixing of fluids [18] and analysis of noisy dynamical systems [34]—where the interest often lies on first exit times—, as well as medical imaging [23], chemistry [26], design of integrated circuits [4], and high-performance supercomputing [1]—more focused on the connection to BVPs. It is well known that

*Corresponding author. *Email addresses:* francisco.bernal@ ist.utl.pt (F. Bernal), juan.acebron@ist.utl.pt (J. A. Acebrón)

 ©2016 Global-Science Press

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

the convergence rate with respect to the timestep h of the approximation to the expected value—the weak convergence rate of the numerical scheme—is only $\mathcal{O}(\sqrt{h})$, unless specific measures to handle the interaction between the diffusion random path and the boundary are undertaken [16]. This compares very unfavourably with the case of diffusions in free space, where even the most basic method—namely Euler-Maruyama's—has a weak order $\mathcal{O}(h)$ [20]. In the last two decades, a number of weak schemes for bounded diffusions have been put forward seeking to raise the weak convergence rate to $\mathcal{O}(h^\delta)$, with $1/2<\delta\leq 1$. Henceforth, we will refer to a scheme with $\delta>1/2$ as a "higher-order scheme". (We will use both terms "scheme" and "integrator" interchangeably.)

In applications, such a low exponent as $1/2$ means that in order to bound the total error by a tolerance a, a very small h is needed, which leads to a cost $\mathcal{O}(a^{-4})$ [10] and so to unpractically lengthy simulations if good accuracy is sought for. This exponent stems from the need to balance the statistical (Monte Carlo) error with the weak order of convergence of the integrator (i.e. the bias). In fact, the exponent can be split as $2+1/\delta$, where 2 and $1/\delta$ are the contribution of either factor. Consequently, raising δ to 1 means that the cost drops to $\mathcal{O}(a^{-3})$. In order to reduce it further, a suitable variance reduction method (tackling the $'2'$ in the exponent) and/or a $\delta>1$ are required.

Giles' Multilevel method [10] is the algorithm of choice for variance reduction, and has truly revolutionized the field of stochastic simulation. In it, several versions of the same stochastic paths—solved at ever smaller values of h—are combined into a telescopic estimator of the expected value in such a way that the rougher versions serve as control variates for the finer ones, thus reducing the overall variance. Importantly, within the Multilevel loop the trajectories are numerically integrated with a legacy scheme such as those considered in this article. The Multilevel method was first extended to stopped diffusions in [17] using the Euler-Maruyama scheme as integrator, achieving a simulation cost of $\mathcal{O}(a^{-3}|\log a|^{1/2})$. This has been recently reduced down to $\mathcal{O}(a^{-2}|\log a|^3)$ in [11], by improving the convergence rate of the variance to linear and replacing the Euler-Maruyama integrator by the Gobet-Menozzi one. The difficulty of combining Multilevel with a given integrator lies in the analysis of the variance, which depends on the *strong* convergence properties of the integrator. In any case, the *weak* order of convergence of the Multilevel estimator is that of the integrator inside it. Moreover, the Multilevel error estimate (see [10]) also relies on that weak order being known in advance—as it is the case whenever the SDE is to be solved to within a given tolerance a.

For unbounded diffusions, there exist superlinear weak integrators [20]. Moreover, Talay-Tubaro extrapolation can be used [36], also combined with Multilevel [10, 21]. On the other hand, while there are no superlinear integrators for bounded SDEs (to the authors' best knowledge), extrapolation could in principle still be used [29, chapter 5]. As a related idea, linear regression was used in [24] to substantially reduce the bias of bounded SDEs. But again, all those strategies critically depend on the prior knowledge of the integrator's order δ.

Given the importance of this topic, it may be surprising that no systematic comparison of such higher-order schemes has been carried out before (as far as we know). This

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

is the main purpose of this paper. A second aim is to identify, and hopefully clarify, the specific implementation issues with each specific integrator.

For the sake of conciseness, this paper is restricted to autonomous bounded diffusions stopped at the boundary. (We recall that, with killed diffusions, one is interested in finding out whether or not the process has hit the boundary—typically, a barrier—within some time interval. On the other hand, with stopped diffusions, the exact hitting point is relevant, too.) In Section 2, this problem will be explicitly connected with the pointwise solution of elliptic boundary value problems (BVPs) with Dirichlet boundary conditions (BCs). In general, the schemes that we review here can very easily be adapted to time-dependent, bounded SDEs (related, in turn, to parabolic BVPs with Dirichlet BCs). On the other hand, bounded SDEs reflected on the boundary require the approximation of the local time on the reflecting boundary, for which a different machinery is required [7,9,16]. (For the sake of completeness, we just point out that those SDEs are related to BVPs with Neumann and Robin BCs.)

Moreover, we have chosen to review only higher-order integrators which are of *general purpose*. By this, we mean that they are not restricted, by construction, to particular regular geometries or SDEs. For instance, the method of choice to solve Laplace's equation with Monte Carlo is the Walk on Spheres [32], which takes full advantage of the isotropy of the diffusion. Particularizing to convex polyhedric domains, the Walk on Rectangles algorithm [8] is preferable, for it deals efficiently with corners. These two methods can be extended to slightly more complex SDEs, such as those having gradient and potential terms with constant coefficients. Other methods for isotropic SDEs are: exponential timestepping [19], designed for parabolic boundaries; and the adaptive method in [3], which may be the best option if they have localized features such as boundary singularities. Finally, Mairé and his collaborators have developed an accurate technique for dealing with the boundaries based on "randomization" of the BCs [22], which has been successfully used in applications, like [23]. Randomization can also be applied to transmission BCs [4,26]. Most of the previous methods can be adapted to reflecting BCs.

Based on the general-purpose criterion, we have included the following integrators into the survey: Brownian bridge [25], Buchmann and Petersen's [5,6], Gobet and Menozzi's [14], and two methods by Milstein [28,30]. The first three are based on the Euler-Maruyama method, which is also included as a reference. As a matter of fact, the method of Buchmann and Petersen is restricted to diagonal diffusions, but given its excellent performance with high-dimensional Poisson's equations, we have included it nonetheless.

For stopped (as well as for killed) bounded diffusions, the reason for the lower weak rate of convergence compared to diffusions in free space is the ambiguity in determining the *first* intersection with a boundary of a random polygonal path—the numerical approximation to the diffusion under timestep h. See Fig. 1 (left). The most natural way of approximating that first intersection is taking some point between the last iteration inside and the first interaction outside. Unfortunately, not only does this result in a weak rate $\mathcal{O}(h^{1/2})$, but also leads to a systematic *overestimation* of the mean first exit time. In order

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

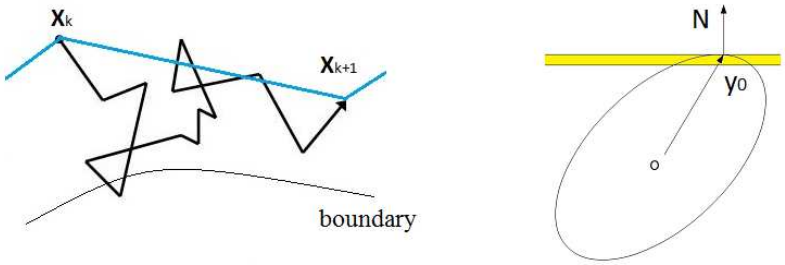


Figure 1: On the left, an excursion outside Ω missed by the coarser time discretization between $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$. Right: tangent ellipsoid with a given orientation. The absorption layer (of width $\sqrt{Dh}$) is highlighted (see Section 4.4).

to fix both issues, missed excursions such as in Fig. 1 (left) must be taken into account. In the literature, three approaches have been proposed:

- Compute the probability that an excursion took place during every discrete step inside and terminate accordingly (used in the Brownian bridge and in the related method by Buchmann and Petersen).
- Make up for the missed first exits by shrinking the domain (boundary shift method by Gobet and Menozzi).
- Control the step length at every step in order to make sure that the trajectory either stays within the domain, or touches its boundary exactly at one known point (methods of bounded increments by Milstein).

In order to actually enjoy a higher order of convergence, the integrators need that the SDE coefficients or the integration domain meet some regularity conditions, which applications often lack. On the other hand, it is interesting to check how the various higher-order methods fare precisely in that situation. Therefore, we have made a point of including both kind of domains (those very smooth and those with corners) in our comparison.

The remainder of the paper is organized as follows. In Section 2, the precise connection between autonomous SDEs stopped at the boundary and linear elliptic BVPs with Dirichlet BCs is made. We have put together the common aspects to all of the integrators in Section 3. Then, the five methods singled out in the introduction are reviewed in Section 4. The test problems are presented in Section 5, and Section 6 concludes the paper.

2 Probabilistic representation

In this paper we consider expectations of the form

$$u(\mathbf{x}_0) = \mathbb{E}\left[g(\mathbf{X}_\tau) e^{\int_0^\tau c(\mathbf{X}_s)ds} + \int_0^\tau f(\mathbf{X}_t) e^{\int_0^t c(\mathbf{X}_s)ds} dt \,\middle|\, \mathbf{X}_0 = \mathbf{x}_0 \right], \tag{2.1}$$

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

where $\mathbf{X}_\tau$ is the value at $t=\tau>0$ of the stochastic process $\mathbf{X}_t:[0,\tau]\to\mathbb{R}^D$, driven by the SDE

$$d\mathbf{X}_t=\mathbf{b}(\mathbf{X}_t)dt+\sigma(\mathbf{X}_t)d\mathbf{W}_t,\qquad \mathbf{X}_0=\mathbf{x}_0. \tag{2.2}$$

In (2.2), $\Omega\subset\mathbb{R}^D, D\geq 1$ is a bounded domain, σ is the diffusion matrix, and $\mathbf{W}_t$ is a standard D-dimensional Wiener process. The *first exit time* (or *first passage time*) τ is defined as

$$\tau=\inf_{t\geq 0}\mathbf{X}_t\in\partial\Omega, \tag{2.3}$$

i.e., the time when a solution of the SDE (2.2) first touches $\partial\Omega$ at the *first exit point* $\mathbf{X}_\tau$. The process $\mathbf{X}_t, 0\leq t\leq\tau$ can be thought of as a non-differentiable trajectory inside Ω.

If $c=g=0$ and $f=1$, (2.1) is the mean exit time from Ω of a diffusion starting at $\mathbf{x}_0\in\Omega$. The other most important case takes place when $\det(\sigma)\neq 0$, $c<0$ and the coefficients $a_{11},\cdots,a_{DD},b_1,\cdots,b_D,c,f,g:\mathbb{R}\mapsto\mathbb{R}$ are assumed regular enough that the solution to the elliptic BVP with Dirichlet BCs (2.4) exists and is unique [31]:

$$\sum_{i=1}^{D}\sum_{j=1}^{D}a_{ij}\frac{\partial^2 u}{\partial x_i\partial x_j}+\sum_{k=1}^{D}b_k\frac{\partial u}{\partial x_k}+cu+f=0,\ \text{ if }\ \mathbf{x}\in\Omega;\qquad u(\mathbf{x})=g,\ \text{ if }\ \mathbf{x}\in\partial\Omega, \tag{2.4}$$

where the matrix $A(\mathbf{x})=[a_{ij}]=\sigma\sigma^T/2$ is positive definite. Then, (2.1) is the pointwise solution of (2.4) at $\mathbf{x}_0$. Formula (2.1) is Dynkin's formula, a particular case of the more general Feynman-Kac formula for parabolic BVPs—see [9] for more details. In the remainder of the paper, we will assume that the identification between the SDE (2.2) and the BVP (2.4) holds and refer to (2.1) both as the solution of an SDE or a BVP.

An alternative representation of $u(\mathbf{x}_0)$, due to Milstein [29], is

$$u(\mathbf{x}_0)=\mathbb{E}[\phi(\mathbf{x}_0)]:=\mathbb{E}\Big[g(\mathbf{X}_\tau)Y_\tau+Z_\tau\Big], \tag{2.5}$$

where $\phi(\mathbf{x})$ are called the *scores* and $(\mathbf{X}_\tau,Y_\tau,Z_\tau)$ are the evaluation at time $t=\tau$ of the solution of the system of SDEs

$$\begin{cases} d\mathbf{X}_t=\mathbf{b}(\mathbf{X}_t)dt+\sigma(\mathbf{X}_t)d\mathbf{W}_t, & \mathbf{X}_0=\mathbf{x}_0,\\ dY=c(\mathbf{X}_t)Ydt, & Y(0)=1,\\ dZ=f(\mathbf{X}_t)Ydt, & Z(0)=0. \end{cases} \tag{2.6}$$

Numerically, the expected value (2.5) is approximated with a Monte Carlo (MC) method involving $N\gg 1$ independent realizations of (2.6), which are in turn integrated numerically with a suitable scheme (integrator) based on a small timestep $h>0$, which we will consider constant:

$$u(\mathbf{x}_0)\approx u_{h,N}(\mathbf{x}_0):=\frac{1}{N}\sum_{j=1}^{N}\phi_h^{(j)}(\mathbf{x}_0), \tag{2.7}$$

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

where $\phi_h^{(j)}(\mathbf{x}_0)$ is the j^{th} realization of (2.6) with a timestep h. In doing this, two kinds of numerical errors arise: the time-stepping error involved in the numerical integration (called bias), and the statistical error, due to replacing the mean with the expected value. The bias depends on the integrator. By virtue of the central limit theorem, the statistical error is $\sqrt{V[\phi(\mathbf{x}_0)]/N}$, where $V[\phi(\mathbf{x}_0)]$ is the variance of the scores [20]. Extending well-established results for the case of unbounded diffusions [10], the root mean square (RMS) error of the MC approximation to (2.4) as $h\to 0^+, N\to\infty$ obeys

$$\epsilon := RMS\left(|u(\mathbf{x}_0)-u_{h,N}(\mathbf{x}_0)|\right) \le |C|h^\delta + q\sqrt{\frac{V[\phi(\mathbf{x}_0)]}{N}} \quad \text{with probability } P_q, \tag{2.8}$$

where C is a constant, $\delta>0$; and $P_q \approx 68.3\%, 95.5\%$, and 99.7%, for $q=1,2,3$. The value of δ for different integrators is the object of study of this paper.

2.1 Alternative representation

Consider the related system to (2.6):

$$\begin{cases} d\tilde{\mathbf{X}}_t = [b(\tilde{\mathbf{X}}_t)-\sigma(\tilde{\mathbf{X}}_t)\boldsymbol{\mu}(\tilde{\mathbf{X}}_t)]dt+\sigma(\tilde{\mathbf{X}}_t)d\mathbf{W}_t, & \tilde{\mathbf{X}}_0=\mathbf{x}_0, \\ d\tilde{Y} = c(\tilde{\mathbf{X}}_t)\tilde{Y}dt+\boldsymbol{\mu}^T(\tilde{\mathbf{X}}_t)\tilde{Y}d\mathbf{W}_t, & \tilde{Y}(0)=1, \\ d\tilde{Z} = f(\tilde{\mathbf{X}}_t)\tilde{Y}dt+\mathbf{F}^T(\tilde{\mathbf{X}}_t)\tilde{Y}d\mathbf{W}_t, & \tilde{Z}(0)=0, \end{cases} \tag{2.9}$$

and let $\tilde{\phi}(\mathbf{x}_0)=g(\tilde{\mathbf{X}}_\tau)\tilde{Y}_\tau+\tilde{Z}_\tau$. It can be proved [29, chapter 2] that $\mathbb{E}[\tilde{\phi}(\mathbf{x}_0)]=\mathbb{E}[]=u(\mathbf{x}_0)$, but $V[\tilde{\phi}(\mathbf{x}_0)]\neq V[\phi(\mathbf{x}_0)]$.

We will exploit (2.9) in two ways. First, it allows us to remove the drift, which is crucial in the Walk on Ellipsoids method in Section 4.4. Secondly, if $\boldsymbol{\mu}(\mathbf{x})$ and $\mathbf{F}(\mathbf{x})$ are so chosen that

$$\sigma^T\nabla u + u\boldsymbol{\mu}+\mathbf{F}=0, \tag{2.10}$$

then $V[\tilde{\phi}(\mathbf{x}_0)]=0$. Unfortunately, this requires the solution of (2.4) in the first place. However, it will allow us to accelerate the simulations in Section 5 (because far fewer trajectories are required for a given statistical error), and so to explore deeper the weak error curves of the various integrators.

3 Preliminaries

3.1 Boundary data

In this paper, $||\cdot||$ is always the 2-norm. The signed distance from a point $\mathbf{x}$ (in- or outside Ω) to the boundary is denoted as $d(\mathbf{x})$, with the criterion that $d(\mathbf{x})$ is negative inside Ω, zero on $\partial\Omega$, and positive outside. The projection of $\mathbf{x}$ onto the boundary $\partial\Omega$ is the closest point to $\mathbf{x}$ on $\partial\Omega$ and is denoted by $\Pi_{\partial\Omega}(\mathbf{x})$. This projection is considered unique, which

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

may not be true unless $|d(\mathbf{x})| = ||\mathbf{x}-\Pi_{\partial\Omega}(\mathbf{x})||$ is small enough. $\partial\Omega$ is assumed smooth enough that the outward unit normal is uniquely defined on any point of it, so that there is a well-defined tangent hyperplane. The notation $\mathbf{N}(\mathbf{x})$ and $H_{\partial\Omega}(\mathbf{x})$ stand for the normal and the tangent hyperplane evaluated at $\Pi_{\partial\Omega}(\mathbf{x})$.

3.2 Signed distance map

For a given point $\mathbf{x}\in\mathbb{R}^D$, all the numerical integrators need to know the *boundary data* $d(\mathbf{x})$, $\Pi_{\partial\Omega}(\mathbf{x})$, $\mathbf{N}(\mathbf{x})$, and $H_{\partial\Omega}(\mathbf{x})$. Unless Ω has a very simple geometry (or is a combination thereof), the signed distance $d(\mathbf{x})$ cannot be computed analytically, but the eikonal equation must be solved in advance [4]:

$$||\nabla d(\mathbf{x})|| = 1 \quad \text{if } \mathbf{x}\notin\partial\Omega, \qquad d(\mathbf{x}) = 0 \quad \text{if } \mathbf{x}\in\partial\Omega. \tag{3.1}$$

Once the signed distance map is available, $\mathbf{N}(\mathbf{x}) = \nabla d(\mathbf{x})$ (hence the sign convention), $\Pi_{\partial\Omega}(\mathbf{x}) = \mathbf{x} - d(\mathbf{x})\nabla d(\mathbf{x})$, and $H_{\partial\Omega}(\mathbf{x}) = \{\mathbf{y}\,|\,(\mathbf{y}-\Pi_{\partial\Omega}(\mathbf{x}))^T\mathbf{N}(\mathbf{x}) = 0\}$. The numerical procedure of choice for solving (3.1) is the Fast Marching method. For more examples and details on Matlab implementations in the context of SDEs see [4]. We remark that, since the boundary data are needed only to handle the interaction trajectory-boundary, the signed distance is usually required in a shell of $\mathcal{O}(\sqrt{h})$ thickness around $\partial\Omega$, not in the whole of $\mathbb{R}^D$.

3.3 Timestepping notation

Let $k=0,1,2,\cdots$ be the *steps* and $h>0$ the timestep. The numerical approximation to $\mathbf{X}_t, t=kh$ is $\mathbf{X}_k$ (and analogously for Y_k and Z_k). The notation σ_k stands for $\sigma(\mathbf{X}_k)$; $\mathbf{b}_k, c_k$ and f_k, as well as d_k and $\mathbf{N}_k$ are defined in the same way. The numerical approximation to the exit time τ is represented by ν. Finally, $\phi_h := g(\mathbf{X}_\nu)Y_\nu + Z_\nu$ are the numerical approximations to $\phi, \mathbf{X}_\tau, Y_\tau$, and Z_τ, respectively.

3.4 Probability distributions and integrators

The notation $w\sim P$ means that the stochastic variable w has been drawn from the probability distribution P. Table 1 displays the probability distributions quoted in the descrip-

Table 1: Probability distributions used in this paper.

Symbol	Name	Integrator	Matlab command
$\mathcal{U}$	Bernoulli (uniform in $[0,1]$)	*BB*,*BP*,*RW*	*rand*
$\mathcal{N}$	standard normal	*EM*,*GM*,*BB*,*BP*	*randn*
$\mathcal{IG}(\gamma,\delta)$	inverse Gaussian	*BP*	*igpdf* (see text)
$\mathcal{B}$	binary (±1 with 50% chance)	*RW*	$2*randi-1$
$\mathcal{S}(D)$	uniform on the surface of a unit D-sphere	*WoE*	*dboluda* (see text)

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

tion of the integrators. The initials EM, GM, BB, BP, RW, and WoE stand for the Euler-Maruyama, Gobet-Menozzi, Brownian bridge, Buchmann-Petersen, Milstein's random walk and Walk on Ellipsoids integrators, respectively.

The distribution $\mathcal{S}(D)$ can be constructed according to [29, remark 5.1.3], but we provide a Matlab code based on a simpler idea below, as well as for the inverse Gaussian distribution (see also [27]).

```
function w= dboluda(N,D)
x= rand(N,D)-1/2; w= w./sqrt(sum(x.^2,2));
function [res]= igpdf(gamma,delta,sampleSize)
chisq1 = randn(sampleSize).^2;
res = delta + 0.5*delta./gamma.* ( delta.*chisq1
      - sqrt(4*delta.*gamma.*chisq1 + (delta.^2).*chisq1.^2) );
l = rand(sampleSize) >= delta./(delta+res);
res( l ) = (delta( l ).^2)./res( l );
```

3.5 SDE coefficients

The functions $\sigma, \mathbf{b}, c \leq 0, f$ and g are assumed to have at least the regularity required for the Feynman-Kac formula to hold, and the matrix A in (2.4) is assumed symmetric positive definite in Ω (strict ellipticity condition). Thus, for any $\mathbf{v} \in \mathbb{R}^D$, $||\sigma\mathbf{v}|| = \mathbf{v}^T\sigma^T\sigma\mathbf{v} > 0$. Then, A has a Cholesky decomposition so that σ can always be chosen lower triangular. Moreover, the eigenvalues of A are $\lambda_{\max}(A) \geq \lambda_2 \geq \cdots \geq \lambda_D > 0$ (note that they may depend on $\mathbf{x}$). Some integrators need $\lambda_{\max}(A)$. Computing it may be very costly (especially in high dimensions, unless A has an exploitable pattern), so that it may be worth bounding it above via Gershgorin's circle theorem: let M be a square matrix and λ any of its eigenvalues. Then,

$$|\lambda - M_{ii}| \leq \sum_{i \neq j} |M_{ij}|. \tag{3.2}$$

If $\lambda > 0$ as with A, $|\lambda - M_{ii}| \geq \lambda - |M_{ii}|$ so that

$$\lambda_{\max}(A) \leq \max_{j=1,\cdots,D} \sum_{i=1}^{D} |a_{ij}|. \tag{3.3}$$

4 Review of the integrators

4.1 Boundary shift method of Gobet and Menozzi (GM)

The Euler-Maruyama method (EM) for (2.6) is Algorithm 1. As already pointed out, neglecting intermediate excursions leads to $0 < \mathbb{E}\nu - \mathbb{E}\tau = \mathcal{O}(\sqrt{h})$. Gobet and Menozzi's idea is to compensate for them by *shrinking* the domain: if one lets Ω_h' and $\partial\Omega_h'$ be the shrunken domain and its boundary, the *discrete* trajectories will *on average* hit $\partial\Omega_h'$ before they would hit $\partial\Omega$. Gobet and Menozzi prove that if the shift inwards is locally given by

$$\text{shift} = -0.5826||\sigma(\mathbf{x})\mathbf{N}(\mathbf{x})||\sqrt{h}\mathbf{N}(\mathbf{x}), \qquad \mathbf{x} \in \partial\Omega, \tag{4.1}$$

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

then the weak convergence rate of the Euler-Maruyama scheme improves to $o(\sqrt{h})$—see [14] for the technical conditions. Importantly, in many instances the weak convergence rate turns out to be in fact linear, although precise conditions for this were not yet determined in [14].

Based on this result, the Gobet-Menozzi scheme (GM) boils down to replacing line 3 in Algorithm 1 with "**if** $d_k > -0.5826||\sigma_k \mathbf{N}_k||\sqrt{h}$", as well as line 1 with "$d_0 < -0.5826||\sigma_0 \mathbf{N}_0||\sqrt{h}$".

Algorithm 1 Euler-Maruyama (EM)

1: **Data:** $h>0$, $\mathbf{X}_0=\mathbf{x}_0\in\Omega, Y_0=1, Z_0=0$, $(d_0<0,\mathbf{N}_0)$
2: **for** $k=0,1,2,\cdots$ until EXIT **do**
3: **if** $d_k<0$ **then**
4: Let $\boldsymbol{\omega}_{k+1}\sim\mathcal{N}$, and take one unbounded step according to:

$$\begin{cases} \mathbf{X}_{k+1}=\mathbf{X}_k+h\mathbf{b}_k+\sqrt{h}\sigma_k\boldsymbol{\omega}_{k+1}, \\ Y_{k+1}=Y_k+hY_kc_k, \\ Z_{k+1}=Z_k+hY_kf_k. \end{cases} \tag{4.2}$$

5: Compute $(d_{k+1},\mathbf{N}_{k+1})$ according to the distance map of Ω
6: **else**
7: EXIT: **goto** 10
8: **end if**
9: **end for**
10: Take $\nu=kh, \mathbf{X}_\tau=\Pi_{\partial\Omega}(\mathbf{X}_k), Y_\tau=Y_k, Z_\tau=Z_k$, and $\phi_h=g(\mathbf{X}_\nu)Y_\nu+Z_\nu$.

4.2 Brownian bridge (BB)

The introduction of Brownian bridge techniques in the numerical solution of SDEs representing elliptic problems with Dirichlet BCs is inspired by the earlier application to barrier problems in finance. A simple barrier problem is as follows. Let $X(t)$ be a one-dimensional process driven by a constant drift $\hat{b}$ and a constant volatility $\hat{\sigma}$:

$$dX(t)=\hat{b}dt+\hat{\sigma}dW_t, \qquad X(0)=y. \tag{4.3}$$

Assume that after $t=h>0$, $z:=X(h)$ is known. Then, the process (4.3) in $0\le t\le h$ conditioned to $X(h)=z$ is called a Brownian bridge pinned at $(0,y)$ and (h,z). Let $y<B>z$ be a barrier. One is interested in the probability of the Brownian bridge hitting the barrier, *i.e.* taking the value $X(\tau')=B$ for some $0<\tau'<h$. This probability turns out to be independent of $\hat{b}$ and is [6, section 3.1]

$$P(\tau'<h)=\exp\Big(-\frac{2(B-y)(B-z)}{h\hat{\sigma}^2}\Big). \tag{4.4}$$

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

This suggests the incorporation into the Euler-Maruyama method of a *boundary test* for detecting excursions during a given time step [2]. Assume a one-dimensional autonomous SDE with non-constant coefficients $b(X)$ and $\sigma(X)$ in $\Omega := (-\infty, x_B]$, and for some $k \geq 1$ let $X_k \in \Omega$ and $X_{k+1} \in \Omega$ (*i.e.* $X_k < x_B > X_{k+1}$). Further, assume that the trajectory during the time interval $kh < t < (k+1)h$ can be approximated by the Brownian bridge pinned at X_k and X_{k+1} with frozen diffusion $\sigma(X_k)$. (Such an approximation may not always be consistent: see [12].) Now let w be a variable uniformly distributed in $[0,1]$. If $w < P$, where

$$P = \exp\Big(-\frac{2(x_B - X_k)(x_B - X_{k+1})}{h\sigma^2(X_k)}\Big), \tag{4.5}$$

then the trajectory is deemed to have undergone an excursion, so that it is finished at the iteration $k+1$ with $\nu = (k+1)h$. A finite interval $\Omega := [x_A, x_B]$ can also be tackled in this way provided that $|x_A - x_B| >> \sqrt{h}$, so that (4.5) approximately holds (*i.e.* if the probability of hitting the farther end can be neglected). These ideas were implemented by Mannella in [25], who also improved on the zero-order frozen-coefficient approximation taken here.

Higher dimensions can be tackled via the *half-space approximation*, which reduces the interaction trajectory-boundary to the one-dimensional case. Let us consider a diffusion with constant coefficients like (4.3) taking place in $\mathbb{R}^D$ $(D \geq 2)$; a hyperplane H defining two half-spaces; and two points $\mathbf{x}_1$ and $\mathbf{x}_2$ in the same half-space. The probability that a Brownian bridge pinned on $\mathbf{X}(0) = \mathbf{x}_1$ and $\mathbf{X}(h) = \mathbf{x}_2$ crosses H within h is still given by (4.4), provided that $(B-x)(B-y)$ is replaced by $d_H(\mathbf{x}_1) d_H(\mathbf{x}_2)$, where $d_H(\mathbf{x}) \geq 0$ is the distance between $\mathbf{x}$ and H. Therefore, the Brownian bridge test can be extended to any dimension if the boundary can be locally approximated by a hyperplane. Intuitively this occurs when $\partial\Omega$ is locally smooth and its radius of curvature is much larger than $\sqrt{h}$ and d_H. Importantly, this does not hold close to a corner or a cusp regardless of h.

The Brownian bridge with half-space approximation for the Euler-Maruyama integration of (2.2) then consists in substituting $\partial\Omega$ with $H_{\partial\Omega}(\mathbf{X}_k)$ and considering the frozen-coefficient Brownian bridge

$$d\mathbf{X} = \mathbf{b}_k dt + \sigma_k d\mathbf{W}_t, \quad \text{with } \mathbf{X}(kh) = \mathbf{X}_k \quad \text{and} \quad \mathbf{X}((k+1)h) = \mathbf{X}_{k+1} \tag{4.6}$$

after computing the Euler-Maruyama step $\mathbf{X}_{k+1}$. The probability of the particle exiting at the iteration $k+1$ is then [16]

$$p = \begin{cases} 1, & \text{if } \mathbf{X}_{k+1} \notin \Omega, \\ \exp\Big(-\frac{2 d_k d_{k+1}}{h \mathbf{N}_k^T \sigma_k \sigma_k^T \mathbf{N}_k}\Big), & \text{otherwise.} \end{cases} \tag{4.7}$$

Note that the distances in (4.7) are now signed, but $d_k d_{k+1} > 0$. The pseudocode is given as Algorithm 2. Assuming great smoothness (C^5) of the domain, the drift, and σ, Gobet proves that the half-space approximation has a linear weak order of convergence [16]. (So that taking the exit at the iteration k instead of $k+1$—or averaging between

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

them—has an effect of order h and does not alter the weak convergence rate.) However, note that the proof pertains to *killed*, rather than *stopped* diffusions. Therefore, it is valid in the estimation of mean exit times or barrier options. On the other hand, the Feynman-Kac functional (2.1) (or equivalently (2.5)), depends on the point where the BC is evaluated (unless $g\equiv 0$, or if it is one end of a one-dimensional interval, *i.e.* $D=1$).

Algorithm 2 Brownian bridge (BB)

Data: $h>0$, $\mathbf{X}_0=\mathbf{x}_0\in\Omega, Y_0=1, Z_0=0$, and $(d_0<0,\mathbf{N}_0)$
for $k=0,1,2,\cdots$ until EXIT **do**
  Get $\mathbf{X}_{k+1}, Y_{k+1}$ and Z_{k+1} according to the Euler-Maruyama scheme (4.2)
  Compute $(d_{k+1},\mathbf{N}_{k+1})$ according to the distance map of Ω
  if $d_{k+1}\geq 0$ **then**
    EXIT: **goto** 14
  else
    Let $w'\sim\mathcal{U}$
    if $w'<\exp\Big(-\dfrac{2d_k d_{k+1}}{h\mathbf{N}_k^T\sigma_k\sigma_k^T\mathbf{N}_k}\Big)$ (boundary test) **then**
      EXIT: **goto** 14
    end if
  end if
end for
Take $\nu=kh, \mathbf{X}_\tau=\Pi_{\partial\Omega}(\mathbf{X}_k), Y_\tau=Y_k, Z_\tau=Z_k$, and $\phi_h=g(\mathbf{X}_\nu)Y_\nu+Z_\nu$.

4.3 The method of Buchmann and Petersen (BP)

This method is an improvement for the case of a class of stopped diffusions of the previous Brownian bridge test. The idea is to determine—in a statistical sense—*when* within the last time step of length h the trajectory exited the domain, and *where* on the tangent hyperplane the crossing took place. Consider again the one-dimensional diffusion (4.3) and the same notation as in (4.4). There are two cases: 1) $y<B\leq z$, when the trajectory has exited for sure, and 2) $y<B>z$, when there may have been an excursion. Then, the associated exit times τ_1 and τ_2 are sampled from the distributions P_1 and P_2 below [6]:

$$P_1(\tau_1)=\mathbf{1}_{[\tau_1<h]}\frac{B-y}{\sqrt{2\pi\hat{\sigma}^2\tau_1^3}}\sqrt{\frac{h}{h-\tau_1}}\exp\Big[-\frac{1}{2\hat{\sigma}^2}\Big(\frac{(z-B)^2}{h-\tau_1}-\frac{(z-y)^2}{h}+\frac{(B-y)^2}{\tau_1}\Big)\Big], \tag{4.8}$$

$$P_2(\tau_2)=\exp\Big(-\frac{2(B-z)(B-y)}{\hat{\sigma}^2\tau_2}\Big). \tag{4.9}$$

In order to sample from those distributions, Buchmann shows that

$$w\sim\mathcal{IG}\Big(\frac{(B-y)^2}{h\hat{\sigma}^2},\frac{B-y}{z-B}\Big)\ \Rightarrow\ \tau_2=\frac{hw}{1+w} \tag{4.10}$$

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

(look up Table 1 for reference) and

$$w' \sim U \Rightarrow \tau_1 = -\frac{2(B-y)(B-z)}{\hat{\sigma}^2 \log w'}. \tag{4.11}$$

Contrary to the plain Brownian bridge method, now a more accurate estimate $kh < \nu < (k+1)h$ of τ is available, which can be used to improve on the exit point—where the Dirichlet BC is evaluated. In order to do so, Buchmann and Petersen use the half-space approximation to reason as follows [5]. Assume that the trajectory is stopped at a time $kh+\tau'$ $(0<\tau'<h)$ between $\mathbf{y}=\mathbf{X}_k$ and $\mathbf{z}=\mathbf{X}_{k+1}$ (either because $\mathbf{z}\notin\Omega$ or because an excursion was detected) at some exit point $\mathbf{y}+\boldsymbol{\eta}$ on the hyperplane $H_{\partial\Omega}(\mathbf{y})$ which is not necessarily the projection $\Pi_{\partial\Omega}(\mathbf{y})$. There is at least one orthogonal coordinate system centred on $\Pi_{\partial\Omega}(\mathbf{y})$ in which $\mathbf{y}$ is expressed as $\mathbf{y}'=(-||\Pi_{\partial\Omega}(\mathbf{y})-\mathbf{y}||,0,\cdots,0)$. Let $\boldsymbol{\eta}'$ be $\boldsymbol{\eta}$ in that frame. *If the particle is driven by a Brownian motion*, then the displacement along perpendicular directions to the first one is distributed according to [5, section 3.3]

$$\eta_j' \sim P_3 = (y_j'-z_j') + \frac{\tau'}{h}\sqrt{\tau(1-\frac{\tau'}{h})}\mathcal{N}, \qquad \text{for } j=2,\cdots,D, \tag{4.12}$$

which is the density of a Brownian bridge pinned at $(y_j',0)$ and (z_j',h) at some intermediate time τ'. Algorithm 3 is the pseudocode for the Buchmann-Petersen method (assuming Brownian motion). In order to implement the rotations in an efficient way (which is especially critical in high dimensions), the reader is referred to [5, section 3.4], where Givens rotations are used in order to construct and operate the orthogonal matrix Q.

The reasoning of Buchmann and Petersen can be extended to diagonal diffusions (*i.e.* in BVPs without crossed derivatives). Then, σ is also diagonal and it is possible to *first* derive τ' from the component parallel to $\mathbf{N}_k$ from the analogous formula to (4.10) or (4.11), and *secondly* insert τ' in (4.12) in order to sample the remaining $D-1$ components on the tangent hyperplane. On the other hand, if the diffusion along the normal direction is not independent from the rest of orthogonal components (which is the general case, even with the frozen component approximation) the reasoning above fails. In theory, formulas (4.10)-(4.12) might be extended to general diffusions based on high-dimensional Brownian bridges (with a full covariance matrix), which is a much more difficult problem. In any case, even in 1D, the method of Buchmann and Petersen with the frozen coefficient approximation may not improve on the Brownian bridge beyond the Brownian motion [6].

4.4 Milstein's walk on ellipsoids (WoE)

Milstein's methods take bounded steps in order to avoid overshooting outside Ω. Out of a variety of them (see [29, chapter 6]), we have picked two. While both of them have been proved to yield a linear bias (under mild technical assumptions), this is dependent on the availability of geometrical and spectral information that may not be immediate. Perhaps

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

Algorithm 3 Method of Buchmann and Petersen for $\nabla^2 u/2+f=0$ (BP)

Data: $h>0$, $\mathbf{X}_0=\mathbf{x}_0\in\Omega, Y_0=1, Z_0=0$, and $(d_0<0,\mathbf{N}_0)$
for $k=0,1,2,\cdots$ until EXIT **do**
  Get $\mathbf{X}_{k+1}, Y_{k+1}$ and Z_{k+1} according to the Euler-Maruyama scheme (4.2)
  Compute $(d_{k+1},\mathbf{N}_{k+1})$ according to the distance map of Ω
  if $d_{k+1}\geq 0$ **then**
    Let $\gamma=\dfrac{d_k^2}{h}, \delta=\dfrac{|d_k|}{d_{k+1}}, w\sim\mathcal{IG}(\gamma,\delta)$, and $\tau_1=\dfrac{hw}{1+w}$
    Let $\tau'=\tau_1$ and EXIT: **goto** 15
  else
    Let $w'\sim\mathcal{U}$ and $\tau_2=\dfrac{-2|d_k|d_{k+1}}{\log w'}$
    if $\tau_2<h$ (boundary test) **then**
      Let $\tau'=\tau_2$ and EXIT: **goto** 15
    end if
  end if
end for

Compute a $D\times D$ orthogonal matrix Q such that $Q\left(\Pi_{\partial\Omega}(\mathbf{X}_k)-\mathbf{X}_k\right)=(|d_k|,0,\cdots,0)^T$ using Givens rotations or Householder transformations [5]
Let $(\eta_1',\cdots,\eta_D')=\boldsymbol{\eta}'=Q(\mathbf{X}_{k+1}-\mathbf{X}_k)$
for $j=2,\cdots,D$ **do**
  Let $w''\sim\mathcal{N}$ and $\eta_j''=\eta_j'+w''\sqrt{\tau'(1-\tau'/h)}$
end for
Let $\mathbf{X}_k'=\mathbf{X}_k+Q^T\boldsymbol{\eta}''$ (such that $\mathbf{X}_k'\in H_{\partial\Omega}(\mathbf{X}_k)$)
Take $\nu=kh, \mathbf{X}_\tau=\Pi_{\partial\Omega}(\mathbf{X}_k'), Y_\tau=Y_k, Z_\tau=Z_k$, and $\phi_h=g(\mathbf{X}_\nu)Y_\nu+Z_\nu$.

for this reason, they are underreported—as far as we know, only [29] presents two numerical experiments. Here, we propose specific implementations of both, warning the reader that they rely on approximations which may spoil their theoretically linear weak convergence. In the WoE [29, section 6.3.2], the drift is first removed using Girsanov's theorem by taking $\boldsymbol{\mu}=\sigma^{-1}\boldsymbol{b}$ and $\boldsymbol{F}=0$ in (2.9)—which may also affect the variance. Then, the system of SDEs to be solved reads:

$$\begin{cases} d\mathbf{X}=\sigma(\mathbf{X})d\mathbf{W}_t, & \mathbf{X}(0)=\mathbf{x}_0, \\ dY=c(\mathbf{X})Ydt+Y\mathbf{b}^T(\mathbf{X})\sigma^{-T}(\mathbf{X})d\mathbf{W}_t, & Y(0)=1, \\ dZ=f(\mathbf{X})Ydt, & Z(0)=0, \end{cases} \tag{4.13}$$

where $\sigma^{-T}=(\sigma^{-1})^T$. Let us define

$$\theta(\mathbf{x},\mathbf{y},s)=\{\mathbf{y}\,|\,(\mathbf{y}-\mathbf{x})^T\sigma^{-T}(\mathbf{x})\sigma^{-1}(\mathbf{x})(\mathbf{y}-\mathbf{x})=s^2\}=\{\mathbf{y}\,|\,(\mathbf{y}-\mathbf{x})^T A^{-1}(\mathbf{x})(\mathbf{y}-\mathbf{x})=2s^2\}. \tag{4.14}$$

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

Since A^{-1} is a positive-definite matrix, $\theta(\mathbf{x},\mathbf{y},s)$ is a $D-$dimensional ellipsoid centred at $\mathbf{x}$ and with principal axes along the eigenvectors of A^{-1}. If $\lambda_{\max}(A)\geq\lambda_2\geq\cdots\geq\lambda_D>0$ are the eigenvalues of A, the lengths of the semiaxes of $\theta(\mathbf{x},\mathbf{y},s)$ are $s\sqrt{2\lambda_{\max}}\geq\cdots\geq s\sqrt{2\lambda_D}>0$. Let us define the function $\hat{r}:\Omega\mapsto\mathbb{R}$ which determines at any point in the domain the tangent ellipsoid to $\partial\Omega$:

$$\hat{r}(\mathbf{x})=\arg\min_s \theta(\mathbf{x},\mathbf{y},s)\cap\partial\Omega\neq\emptyset. \tag{4.15}$$

Milstein's WoE has the sequence $\{\mathbf{X}_k\}_{k=0,1,\cdots}$ hop over small ellipsoids of position-dependent size until coming close enough to the boundary, at which step $\partial\Omega$ is deemed hit—see Algorithm 4.

The discretization parameter in Algorithm 4 is $r=\sqrt{Dh}$ rather than h. At any point $\mathbf{X}_k$ of the trajectory, the next hop is over the ellipsoid $\theta(\mathbf{X}_k,\mathbf{y},r)$, except when $\mathbf{X}_k$ is so close to the boundary that $\theta(\mathbf{X}_k,\mathbf{y},r)$ intersects the boundary, and the tangent ellipsoid $\theta(\mathbf{X}_k,\mathbf{y},\hat{r}(\mathbf{X}_k))$ has to be taken instead, in order to make sure that the trajectory will not overshoot the boundary. This happens when $|d_k|<r\sqrt{2\lambda_{\max}(A)}$. (Note that λ_d in [29, chapter 6] is equivalent to $\sqrt{2\lambda_{\max}(A)}$ here.) Eventually, the trajectory is stopped at the projection on the boundary when the distance is less than r^2 —see Fig. 1 (right).

Let $\boldsymbol{\omega}\sim\mathcal{B}(D)$ (look up Table 1), and $\mathbf{y}=\sigma_k\boldsymbol{\omega}$. Since $\boldsymbol{\omega}^T\boldsymbol{\omega}=1=\mathbf{y}^T\sigma_k^{-T}\sigma_k^{-1}\mathbf{y}$, $\mathbf{y}$ lies on the surface of the ellipsoid $\mathbf{y}^TA^{-1}\mathbf{y}=2$. Then, the possible new positions $\mathbf{X}_{k+1}=\mathbf{X}_k+r_{k+1}\sigma(\mathbf{X}_k)\boldsymbol{\omega}_{k+1}$ in Algorithm 4 are distributed over the surface of the ellipsoid $\theta(\mathbf{X}_k,\mathbf{y},r_{k+1})$. Algorithm 4 has an asymptotic weak convergence rate $\mathcal{O}(r^2)=\mathcal{O}(h)$ for a C^4 solution of the BVP [29, theorem 3.8].

In order to implement Algorithm 4, the calculation of $\hat{r}$ is necessary, and this is not clarified either in [29] or in [28]. Since this geometric construction is difficult in the event of interaction with an irregularly shaped or nonsmooth boundary (and would likely be very time-consuming in that case), we propose here to take the half-space approximation (see the right side of Fig. 1, where for simplicity $\mathbf{x}=\mathbf{0}$.) Let

$$m(\mathbf{y})=\mathbf{y}^T\sigma^{-T}\sigma^{-1}\mathbf{y}-r^2=\mathbf{y}^TA^{-1}\mathbf{y}-2r^2. \tag{4.17}$$

The tangency point $\mathbf{y}_0$ belongs to the hyperplane which is at a distance $\mathbf{N}^T\mathbf{y}_0$ from the origin, and since $m(\mathbf{y}_0)=0$

$$\nabla m(\mathbf{y}_0)=2A^{-1}\mathbf{y}_0=2q\mathbf{N} \tag{4.18}$$

for some $q\neq0$. Since $\det\sigma^{-1}\neq0$,

$$\mathbf{y}_0=\frac{k}{2}A\mathbf{N}\Rightarrow\begin{cases} q^2\mathbf{N}^TA^TA^{-1}A\mathbf{N}=2\hat{r}^2, & \text{and}\\ \mathbf{N}^T\mathbf{y}_0=q\mathbf{N}^TA\mathbf{N}, \end{cases} \tag{4.19}$$

and recalling that $\mathbf{N}^T\mathbf{y}_0$, the distance to the tangent hyperplane, is the distance to the boundary in the half-space approximation,

$$\hat{r}(\mathbf{X}_k)=\frac{|d(\mathbf{X}_k)|}{\sqrt{2\mathbf{N}_k^TA^T\mathbf{N}_k}}=\frac{|d(\mathbf{X}_k)|}{||\sigma_k\mathbf{N}_k||}. \tag{4.20}$$

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

Algorithm 4 Milstein's walk on ellipsoids (WoE)

1: **Data:** $h>0$, $r=\sqrt{Dh}$, $\mathbf{X}_0=\mathbf{x}\in\Omega, Y_0=1, Z_0=0$, $(d_0<0,\mathbf{N}_0)$
2: **for** $k=0,1,2,\cdots$ until EXIT **do**
3: **if** $d_k\geq -r^2$ **then**
4: EXIT: **goto** 15
5: **else**
6: **if** $-r\sqrt{2\lambda_{\max}(A)}\leq d_k<-r^2$ **then**
7: Compute $\hat{r}(\mathbf{X}_k)$ and let $r_{k+1}=\hat{r}(\mathbf{X}_k)$
8: **else**
9: $r_{k+1}=r$
10: **end if**
11: Let $\boldsymbol{\omega}_{k+1}\sim\mathcal{S}(D)$, and take one step inside Ω according to:

$$\begin{cases} \mathbf{X}_{k+1}=\mathbf{X}_k+\sigma_k\boldsymbol{\omega}_{k+1}r_{k+1}, \\ Y_{k+1}=Y_k+Y_k c_k r_{k+1}^2/D+Y_k\mathbf{b}_k^T\sigma_k^{-T}\boldsymbol{\omega}_{k+1}r_{k+1}, \\ Z_{k+1}=Z_k+Y_k f_k r_{k+1}^2/D. \end{cases} \tag{4.16}$$

12: Compute $\{d_{k+1},\mathbf{N}_{k+1}\}$ according to the distance map of Ω and let $k=k+1$
13: **end if**
14: **end for**
15: Let $\nu=(k+1)h, \mathbf{X}_\nu=\Pi_{\partial\Omega}(\mathbf{X}_{k+1}), Y_\nu=Y_{k+1}, Z_\nu=Z_{k+1}$ and $\phi_h=g(\mathbf{X}_\nu)Y_\nu+Z_\nu$

Accurately determining $\lambda_{\max}(A)$ may be time-consuming if $A=A(\mathbf{x})$, specially in high dimensions. In those cases, a better approach may just be to bound it above using the Gershgorin's circles estimate (3.3). Even if $\lambda_{\max}(A)$ is sharp, introducing the half-space approximation for the computation of $\theta(\mathbf{X}_k,\mathbf{y},r_k)$ and thus $\hat{r}(\mathbf{X}_k)$ in Algorithm 4 may spoil the strict linear weak convergence rate of the WoE. The reason is that close to a corner or to a curved boundary, the tangent ellipsoid to the hyperplane tangent to boundary may stick slightly out of $\partial\Omega$, which might lead to the trajectory hopping outside Ω. In that event, $d_{k+1}>0$ and line 3 in Algorithm 4 terminates the trajectory.

As a last but important point, in (4.16), the inverse matrix $\sigma(\mathbf{X}_k)^{-T}$ must be calculated in order to compute Y_{k+1}. The practical approach is of course to solve the linear system $\mathbf{b}_k=\sigma_k\boldsymbol{\mu}_k$ with Gauss elimination, exploiting the fact that σ can always be chosen triangular (Section 3.5). Still, this involves a cost of up to $\mathcal{O}(D^2)$ per step and trajectory, which may slow down the simulation if a high-dimensional BVP with many crossed derivatives is being solved.

4.5 Milstein's random walk (RW)

Algorithm 5 summarizes algorithms 2.1 and 3.1 in [29, chapter 6]. This method takes bounded steps whose components along the coordinate axes are quantized, for the com-

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

Algorithm 5 Milstein's random walk (RW)

1: **Data:** $h>0$, $\lambda>0$, $\mathbf{X}_0=\mathbf{x}_0\in\Omega, Y_0=1, Z_0=0$, $(d_0<0,\mathbf{N}_0)$
2: **for** $k=0,1,2,\cdots$ until EXIT **do**
3: [SAFEGUARD: **if** $d_k\geq 0$ **then** EXIT: **goto** 18. **endif**]
4: **if** $d_k\geq -\lambda\sqrt{h}$ **then**
5: Let $w\sim\mathcal{U}$ and $p=\dfrac{\lambda\sqrt{h}}{|d_k|+\lambda\sqrt{h}}$
6: **if** $w>p$ **then**
7: Let $\mathbf{X}'_k=\mathbf{X}_k-\lambda\sqrt{h}\mathbf{N}_k$ (note that $\mathbf{X}'_k\in\Omega$ but $\mathbf{X}'_k\notin S_\lambda$)
8: **else**
9: The boundary is deemed hit at $\Pi_{\partial\Omega}(\mathbf{X}_k)$. EXIT and **goto** 18
10: **end if**
11: **else**
12: Let $\mathbf{X}'_k=\mathbf{X}_k$
13: **end if**
14: Let $\boldsymbol{\xi}=(\xi_1,\cdots,\xi_D)$ with $\xi_j\sim\mathcal{B}\,(j=1,\cdots,D)$
15: Take a step inside Ω according to:

$$\begin{cases}\mathbf{X}_{k+1}=\mathbf{X}'_k+h\mathbf{b}(\mathbf{X}'_k)+\sqrt{h}\sigma(\mathbf{X}'_k)\boldsymbol{\xi},\\ Y_{k+1}=Y_k+hc(\mathbf{X}'_k)Y_k,\\ Z_{k+1}=Z_k+hf(\mathbf{X}'_k)Y_k.\end{cases}\tag{4.21}$$

16: Compute $(d_{k+1},\mathbf{N}_{k+1})$ according to the distance map of Ω
17: **end for**
18: Let $\nu=kh, \mathbf{X}_\nu=\Pi_{\partial\Omega}(\mathbf{X}_k), Y_\nu=Y_k, Z_\nu=Z_k$ and $\phi_h=g(\mathbf{X}_\nu)Y_\nu+Z_\nu$.

ponents of the vector $\boldsymbol{\xi}$ in (4.21) are ±1 with equal probability $1/2$. In order to prevent the trajectory from taking a step outside Ω, a *boundary zone* is defined as the set of points from which at least one of the 2^D possible steps defined by (4.21) would take the trajectory outside Ω. Since this definition is non-constructive, one assumes that there is a positive scalar λ such that, if $\mathbf{x}\in\Omega$ and $|d(\mathbf{x})|\leq\lambda\sqrt{h}$, then $\mathbf{x}$ belongs to the boundary zone. Thus for practical purposes, the boundary zone is replaced by the interior shell S_λ of thickness $\lambda\sqrt{h}$ which contains it.

Whenever the trajectory enters S_λ, a special step is taken than either takes it on the closest boundary point, or throws it back inside Ω and out of S_λ. This binary event has an associated probability given by p in Algorithm 5, line 5. Eventually, the trajectory will hit the boundary after entering S_λ, and finish. This method has an asymptotic linear weak order of convergence [29, section 6.3.1] under C^2 $\partial\Omega$ and C^3 SDE coefficients [29, section 6.2.2].

This method's snag is to determine λ, for the optimal way of doing it turns out to be a problem-dependent issue, not addressed in [29]. In order to reduce the bias, it is

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

desirable that λ be as small as possible as long as S_λ contains the boundary zone. Note that

$$||\sigma\boldsymbol{\xi}|| \leq ||\boldsymbol{\xi}|| \sup_{\boldsymbol{\xi}\neq 0} \frac{||\sigma\boldsymbol{\xi}||}{||\boldsymbol{\xi}||} = ||\boldsymbol{\xi}|| \sqrt{\lambda_{\max}(\sigma^T\sigma)} = \sqrt{D\lambda_{\max}(A)}, \tag{4.22}$$

because $||\boldsymbol{\xi}|| = \sqrt{D}$ and $\lambda_{\max}(\sigma^T\sigma) = \lambda_{\max}(\sigma\sigma^T)$. (The latter comes from the spectra of $\sigma^T\sigma$ and $\sigma\sigma^T$ being the same: if $\mathbf{v}$ is an eigenvector of $\sigma^T\sigma$ with eigenvalue s, then $\mathbf{w} = \sigma\mathbf{v}$ is an eigenvector of $\sigma\sigma^T$ with the same eigenvalue, and viceversa.) Using (4.22), the steps can be bounded as

$$||\mathbf{X}_{k+1} - \mathbf{X}_k|| \leq h||\mathbf{b}_k|| + \sqrt{Dh\lambda_{\max}(A)}. \tag{4.23}$$

Then, a spatially-dependent $\lambda = \lambda(\mathbf{x})$ given by

$$\sqrt{h}\lambda(\mathbf{x}) = h||\mathbf{b}(\mathbf{x})|| + \sqrt{Dh\lambda_{\max}(A)} \tag{4.24}$$

is an obvious choice. But this implicitly assumes that the ball centred at $\mathbf{x}$ with radius $\sqrt{h}\lambda(\mathbf{x})$ is contained in Ω—which is similar to a local half-space approximation. Moreover, an upper bound to $\lambda_{\max}(A)$ has to be estimated. To sum up, this approach amounts to replacing line 4 in Algorithm 5 by

$$d_k \geq -h||\mathbf{b}(\mathbf{x})|| - \sqrt{Dh\lambda_{\max}(A)}. \tag{4.25}$$

On the other hand, if a constant λ is used, estimating it so that it is valid throughout Ω may be challenging, for it is an optimization problem involving the values that $\lambda_{\max}\big(A(\mathbf{x})\big)$ and the drift may take on inside a domain Ω with potentially irregular shape. Finally, as suggested in [29], a last option is setting λ to a number 'large enough', meaning that it will be valid on most steps, but allowing for the possibility that some trajectories get inadvertedly out of Ω. Since that any of these options will most likely spoil the linear weak order of convergence, the safeguard in line 3 of Algorithm 5 is required.

5 Numerical experiments

5.1 Setup

We consider three domains in $\mathbb{R}^D$, $D \geq 3$:

- $\Omega_B(R,\mathbf{C})$ is a hypersphere of radius R and center $\mathbf{C}$.
- $\Omega_G(R,\mathbf{C})$ is a hypersphere of radius R at the origin from which the positive $D-$dimensional orthant has been carved out, and afterwards it has been shifted a vector $\mathbf{C}$, as in Fig. 2 (in 2D). In 3D it looks like a piece of Gouda cheese.
- $\Omega_E(L,\mathbf{C})$ looks in 3D like an piece of Emmental cheese (see Fig. 2), and is constructed from a hypercube with vertices at the origin and $(L,\cdots,L)$, where $L>0$. Then, two balls of radius one-third of the diagonal and centred on those vertices are removed, and the resulting body is shifted a vector $\mathbf{C}$.

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

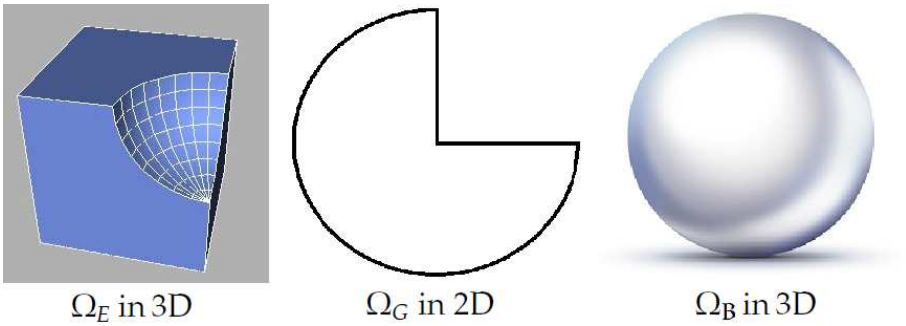


Figure 2: The three $D-$dimensional test domains. On the left is Ω_E in 3D. A ball with radius $\sqrt{D=3}/3$ and centred at $(1,1,1)$ has been removed, as well as another similar one at the origin (not visible).

Note that while Ω_G has one single corner regardless of the dimension, Ω_E has ever more as D increases. The code has been written in MATLAB in fully vectorized style, and run on single processors. It is made up of a main program, which manages the trajectories until exit and computes the Monte Carlo estimates; and a collection of files with the functions for the BVP to be solved, the domain, and the integrator. The main program input is a driver file with all the options for a given simulation. The whole suite will be made available from the first author's webpage.

The exact solution, u_{ex}, is always known and non-zero. The relative error is defined as

$$r_{h,N}=\frac{|u_{h,N}-u_{ex}|}{|u_{ex}|}. \tag{5.1}$$

For every BVP and integrator we are interested in:

1. The asymptotic weak order of convergence of $r_{h,N}$ as $h\to 0^+$ (*i.e.* δ). In order to do this for each of the problems, we run n independent simulations and construct an experimental plot of $r_{h,N}$ vs. the timestep $h=0.2,0.1,\cdots,0.2/2^{n-1}$. The simulations are run until the statistical error makes at most 20% of the relative error with 95.5% probability, *i.e.* $q\sqrt{Var[\phi_h(\mathbf{x}_0)]/N}\leq|u_{h,N}-u_{ex}|/5$ with $q=2$. Then, δ is extracted from a loglog fit of those simulations for which $r_{h,N}<0.15$. Whenever bias cancellation obviously occurs (explained later in Section 5.2), then the fit includes only the data points after the cancellation and with a relative error smaller than 15%. We always report the number of data points included in the fit along with the fitted value of δ.
2. The time it takes to produce a numerical solution $u_{h,N}$ with relative error $r_{h,N}$ less than au_{ex}. We set a depending on the BVP. Therefore, we choose h from the previous convergence curve such that the bias is $au_{ex}/2$, and N such that the statistical error is smaller than $au_{ex}/2$ with 95.5% probability.

Variance reduction. In order to speed up the simulations and being able to investigate the bias at small h, we apply variance reduction based on the exact formula (2.10).

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

Specifically, we let

$$\boldsymbol{\mu}=0, \qquad \mathbf{F}=-\sigma^T\nabla u_{ex}, \tag{5.2}$$

and modify the equation for Z_{k+1} in each method accordingly. In the WoE, $\boldsymbol{\mu}\neq 0$, so that the formula that must be used is

$$\mathbf{F}=-\sigma^T\nabla u_{ex}-u_{ex}\sigma^{-1}\mathbf{b}. \tag{5.3}$$

Note that in either case, the trajectories are the same as without variance reduction. While the SDE systems (2.6) and (2.9) (*i.e.* with and without VR) are not identical, the numerical difference is, to all practical purposes, negligible, other than reducing (by up to 99.99%, or even more) the statistical error. We stress that variance reduction is used to construct the bias convergence curves, but obviously not when comparing the computational cost of the integrators for a given error tolerance a.

5.2 Components of the discretization error (bias)

Let us compactly write the BVP (2.4) as

$$\mathcal{L}u+f=0 \quad \text{if } \mathbf{x}\in\Omega, \qquad u(\mathbf{x})=g \quad \text{if } \mathbf{x}\in\partial\Omega, \tag{5.4}$$

where $\mathcal{L}:=\sum_{i=1}^{D}\sum_{j=1}^{D}a_{ij}\frac{\partial^2 u}{\partial x_i\partial x_j}+\sum_{k=1}^{D}b_k\frac{\partial u}{\partial x_k}+cu$. The solution u can be split as

$$u=v+w, \quad \text{with} \begin{cases} \mathcal{L}v+f=0, & v(\mathbf{x}\in\partial\Omega)=0, \\ \mathcal{L}w=0, & w(\mathbf{x}\in\partial\Omega)=g. \end{cases} \tag{5.5}$$

From (2.5), it can be seen that $g(\mathbf{X}_\tau)Y_\tau$ vanishes for v, while $Z_\tau=0$ for w. Therefore, when solving for u, v and w with a numerical integrator, the *signed* biases are also related as

$$\lim_{N\to\infty}(u_{ex}-u_{h,N})=\lim_{N\to\infty}(v_{ex}-v_{h,N})+\lim_{N\to\infty}(w_{ex}-w_{h,N}). \tag{5.6}$$

In other words, the bias of $u_{h,N}$ is the sum of two components: one influenced by the sampling of the Dirichlet BC by the trajectories (first term in (2.1)), and a second one which is the numerical approximation of the line integral for f along the trajectory (second term in (2.1)). Both components, which we may call the BC sampling and quadrature errors, are respectively the signed biases of $w_{h,N}$ and $v_{h,N}$. Note that this is not to say that $v_{h,N}$ is unaffected by the presence of the boundary, for the length of individual trajectories is still determined by the first exit. However, since $\mathcal{L}$ is the same for u,v and w, any N trajectories involved in $u_{h,N}$,$v_{h,N}$ and $w_{h,N}$ will be the same provided that the stream of random numbers is replicated upon computation of each of them.

This implicit decomposition of the bias often brings about a numerical artifact (so far unreported, to the best of our knowledge) that can lead to a misinterpretation of the behaviour of weak error curve with respect to h when h is not small enough; and which we explain next with an example. Let us denote the signed biases in (5.6) with $\hat{B}$, and

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

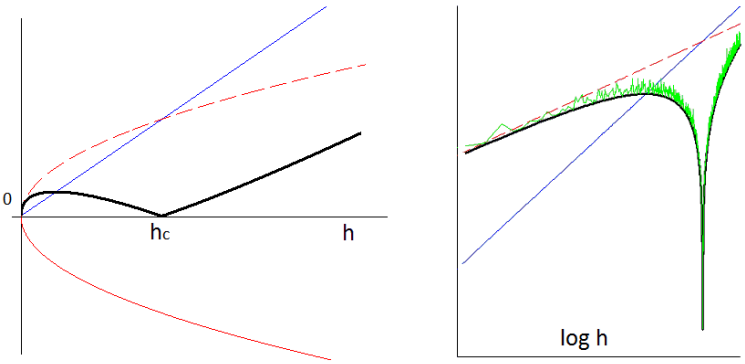


Figure 3: Bias decomposition and cancellation. Left: when the two components of the bias have opposed signs, a cancellation of the bias (thick line) may take place at finite $h=h_c$. Right: same plot in logarithmic scale. Note the 'plunge' and sharp 'rebound' of the bias. Gaussian noise proportional to one-fifth of the bias has been added (in green), to simulate the effect of the statistical error.

assume that $\hat{B}(w_{h,N})=C_w\sqrt{h}$ and $\hat{B}(v_{h,N})=C_v h$. Then, $|\hat{B}(u_{h,N})|\to\mathcal{O}(\sqrt{h})$, but if the signs of C_v and C_w are opposed, then the bias of $u_{h,N}$ 'vanishes' at a finite value of $h=h_c>0$ where $\hat{B}(v_{h,N})$ and $-\hat{B}(w_{h,N})$ intersect. See Fig. 3 for an illustration. In the loglog plot, the bias of $u_{h,N}$ (in absolute value) plunges, and then bounces back onto the slower rate of convergence (here, $\mathcal{O}(\sqrt{h})$). If data were only available for $h\gtrsim h_c$, the cancellation of the bias components might be mistaken for 'superlinear' convergence.

The above artifact, which we will refer to as 'bias cancellation', accounts for the 'plunges' and 'rebounds' which can be observed in many of our experiments. Consider for instance the plunge of the error curve of the GM integrator at around $h\approx 10^{-4}$ in Fig. 7. In Table 2 we indirectly demonstrate the artifact in that case. Let GM* be a variation of GM where, once the particle is deemed out at $\mathbf{X}_K$, the 'BC' g is evaluated there rather than on $\Pi_{\partial\Omega}(\mathbf{X}_K)$. (This is possible only because $g=u_{ex}|_{\partial\Omega}$, and u_{ex} is available). The MC approximation yielded by GM*—let it be $u_{h,N}(GM*)$—only carries the quadrature and statistical errors, so that its signed bias is the same as that of $v_{h,N}(GM)$. As can be seen, the relative error of $u_{h,N}(GM*)$ is nonetheless larger than that of $u_{h,N}(GM)$ (as well as linearly decreasing). The reason why that the latter is smaller is the opposite contribution of the BC sampling error. The rightmost column in Table 2 shows the difference (which is actually $u_{ex}-w_{h,N}(GM)$, plus some statistical noise), indicating that the BC sampling error starts dropping below $h\approx 10^{-4}$. Summing up, this worked out example illustrates that if various components of the bias converge at a different rate, cancellation, preceded by superlinearity, can take place.

5.3 Example I: General diffusion in 3D

This is the test problem of Gobet and Menozzi in [14].It is a $3D$ diffusion where all the coefficients are position dependent except for the potential c, which is zero. On it, Gobet

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

Table 2: Bias decomposition around the 'plunge' in the error convergence curve for the GM integrator on Fig. 7—see Section 5.2 for details.

h	$u_{ex}-u_{h,N}(GM)$	$u_{ex}-u_{h,N}(GM*)$	difference
.000781	$.00562\pm4.5\times10^{-5}$	$.00645\pm4.3\times10^{-5}$	$-.00083$
.000395	$.00219\pm3.2\times10^{-5}$	$.00318\pm3.0\times10^{-5}$	$-.00010$
.000195	$.00067\pm2.3\times10^{-5}$	$.00161\pm2.1\times10^{-5}$	$-.00095$
9.77×10^{-5}	$4\times10^{-6}\pm1.6\times10^{-5}$	$.00080\pm1.5\times10^{-5}$	$-.00080$
4.88×10^{-5}	$.00023\pm1.2\times10^{-5}$	$.00039\pm1.1\times10^{-5}$	$-.00016$
2.44×10^{-5}	$.00028\pm8.2\times10^{-6}$	$.00020\pm7.5\times10^{-6}$	.00008

and Menozzi observed linear convergence of their integrator. Like all examples, it will be solved on the three domains $\Omega_B,\Omega_E,\Omega_G$ (see Fig. 2). The diffusion matrix is given by

$$\sigma=\begin{pmatrix} \sqrt{1+|z|} & 0 & 0 \\ \frac{1}{2}\sqrt{1+|x|} & \frac{\sqrt{3}}{2}\sqrt{1+|x|} & 0 \\ 0 & \frac{1}{2}\sqrt{1+|y|} & \frac{\sqrt{3}}{2}\sqrt{1+|y|} \end{pmatrix}, \tag{5.7}$$

while the rest of functions are:

$$\begin{aligned} &\mathbf{b}=(y,z,x)^T, \qquad c=0, \\ &f=y^2z+z^2x+x^2y+\frac{1}{2}\sqrt{1+|z|}\sqrt{1+|x|}+\frac{\sqrt{3}}{2}x\sqrt{1+|x|}\sqrt{1+|y|}. \end{aligned} \tag{5.8}$$

The solution is $u_{ex}=g=xyz$. Regarding Milstein's methods, σ is already in triangular form as is convenient for computing $\boldsymbol{\mu}=\sigma^{-1}\mathbf{b}$. In any case, since $D=3$, σ^{-1} and $\lambda_{\max}(A)$ can be precalculated symbolically and stored, which will make the integration with WoE faster:

$$\sigma^{-1}=\begin{pmatrix} \frac{1}{\sigma_{11}} & 0 & 0 \\ \frac{-\sigma_{21}}{\sigma_{11}\sigma_{22}} & \frac{1}{\sigma_{22}} & 0 \\ \frac{\sigma_{21}\sigma_{32}}{\sigma_{11}\sigma_{22}\sigma_{33}} & \frac{-\sigma_{32}}{\sigma_{22}\sigma_{33}} & \frac{1}{\sigma_{33}} \end{pmatrix}, \qquad A=\frac{1}{2}\begin{pmatrix} \sigma_{11}^2 & \sigma_{11}\sigma_{21} & 0 \\ \sigma_{11}\sigma_{21} & \sigma_{22}^2+\sigma_{21}^2 & \sigma_{22}\sigma_{32} \\ 0 & \sigma_{22}\sigma_{32} & \sigma_{33}^2+\sigma_{32}^2 \end{pmatrix}. \tag{5.9}$$

We have solved this problem at $\mathbf{x}_0=(0.56,0.52,0.30)$ in $\Omega_B(1,[0,0,0])$; at $\mathbf{x}_0=(0.57,0.57,0.57)$ in $\Omega_G(1,[0.67,0.67,0.67])$; and at $\mathbf{x}_0=(0.57,0.57,0.57)$ in $\Omega_E(1,[0,0,0])$. Notice that the distances to the boundary are different in each case.

Results are shown in Figs. 4-6. Error bars for $r_{h,N}$ are not shown for the sake of clarity, but recall that they are always $\pm r_{h,N}/5$. In this example, we have overlaid the convergence curves with the estimates for δ (recall that not all the points are included into the δ fit), as well as with the computational times for an error tolerance $a=0.01u_{ex}$.

Here, RW uses the *ad-hoc* approximation $\lambda=2$. It works in Ω_B and Ω_E, but definitely not in Ω_G, due to the frequent overshoots of the inner corner.

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

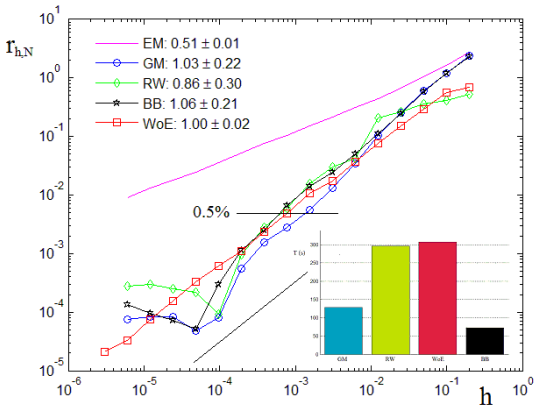


Figure 4: Relative error convergence, fitted δ, and comparison of times with $a=0.01u_{ex}$ for Example I in Ω_B. The segment of slope one is a guide for the eye.

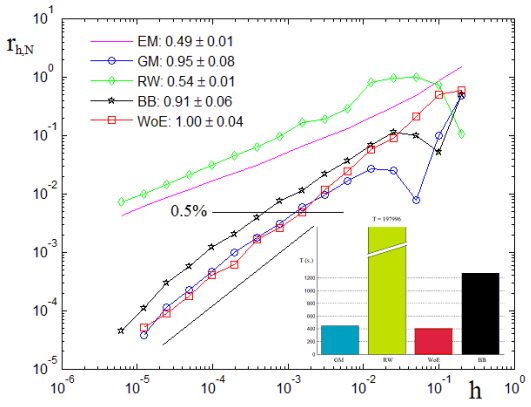


Figure 5: Relative error convergence, fitted δ, and comparison of times with $a=0.01u_{ex}$ for Example I in Ω_G.

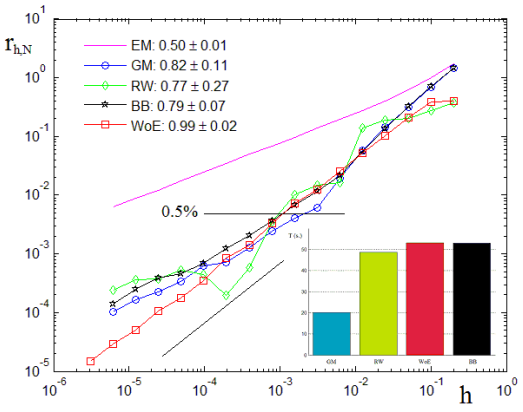


Figure 6: Relative error convergence, fitted δ, and comparison of times with $a=0.01u_{ex}$ for Example I in $\Omega_E(1,[0,\cdots,0])$.

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

Several instances of bias cancellation are evident on those plots. In Fig. 4, all the integrators except EM and WoE 'bounce back' slightly below $h=10^{-4}$.

5.4 Example II: General diffusion in arbitrary dimension

This is a synthetic problem with crossed second derivatives, oscillating coefficients and oscillating solution that can be easily extended to any arbitrary dimension. The exact solution is $u_{ex}=2+\cos\mathbf{k}^T(\mathbf{x}-\mathbf{y})$. The evaluation point is $\mathbf{x}_0=(.7,.7,\cdots,.7)^T$, and the domains are $\Omega_G(1,[0.67,\cdots,0.67])$ and $\Omega_E(1,[0,\cdots,0])$. In order for the solution to be qualitatively similar regardless of D, we choose $\mathbf{k}=(1,2,3,1,2,\cdots,mod(D/3))^T$, where $mod(D/3)$ is the modulus, and we set $\mathbf{y}=\mathbf{x}_0$, so that $u_{ex}(\mathbf{x}_0)=3$. The diffusion matrix is chosen as

$$\sigma=\begin{pmatrix}1&0&\cdots&0\\1&1&\cdots&0\\\vdots&\vdots&\ddots&\vdots\\1&1&\cdots&1\end{pmatrix}\Rightarrow\sigma\sigma^T/2=A=\frac{1}{2}\begin{pmatrix}1&1&1&\cdots&1\\1&2&2&\cdots&2\\1&2&3&\cdots&3\\\vdots&\vdots&\vdots&\ddots&\vdots\\1&2&3&\cdots&D\end{pmatrix}. \tag{5.10}$$

The rest of coefficients are

$$\mathbf{b}=\mathbf{x},\qquad c=-||\mathbf{x}||^2, \tag{5.11}$$

$$f=(\mathbf{k}^TA\mathbf{k}+||\mathbf{x}||^2)\cos\mathbf{k}^T(\mathbf{x}-\mathbf{y})+2||\mathbf{x}||^2+\mathbf{k}^T\mathbf{x}\sin\mathbf{k}^T(\mathbf{x}-\mathbf{y}). \tag{5.12}$$

The convergence of relative errors for Ω_E in 3D is shown in Fig. 7, and those for Ω_G in $3D$, $24D$, and $48D$ are shown in Fig. 8. As D grows, so does the bias (and the variance), and the fit of δ based on the acceptable available data becomes more unreliable. See Table 3.

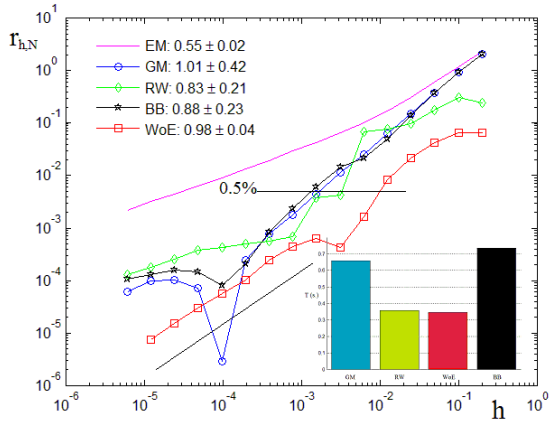


Figure 7: Relative error convergence, fitted δ, and comparison of times with $a=0.01u_{ex}$ for Example II in $\Omega_E(1,[0,\cdots,0])$ (see Section 5.4). The 'plunge' of the GM curve is worked out in Section 5.2.

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

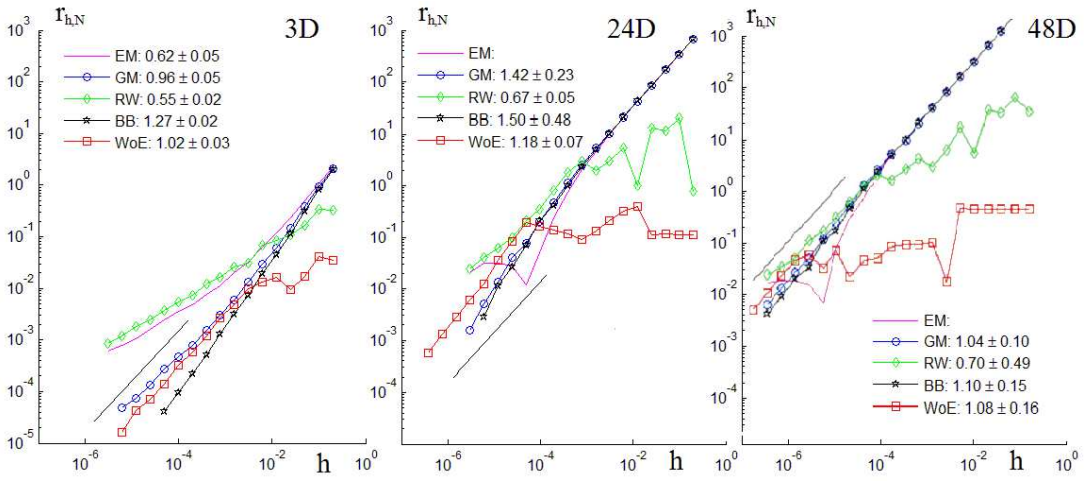


Figure 8: Convergence of the relative error and fitted δ for Example II in $D=\{3,24,48\}$. Notice that the scale is the same. There are not enough acceptable data to fit δ_{EM} in 24D and 48D.

Table 3: Fitted δ for Example II. The figure in parentheses is the number of data points included in the fit (see Section 5.1 for the criteria). A poor error bar on δ means that the relative error had not entered the asymptotic power-law regime.

domain	EM	GM	RW	BB	WoE
Ω_E in 3D	0.55±.02(11)	1.01±.42(12)	0.83±.21(13)	0.88±.23(12)	0.98±.04(7)
Ω_E in 6D	0.57±.02(9)	1.28±.11(10)	0.69±.06(10)	1.19±.28(10)	0.98±.03(10)
Ω_E in 12D	0.64±.08(7)	1.11±.08(7)	0.80±.06(7)	1.25±.19(7)	1.00±.04(9)
Ω_E in 24D	0.22±.29(10)	0.94±.12(6)	1.15±.86(6)	0.93±.09(6)	1.43±.10(6)
Ω_E in 48D	0.47±1.47(6)	1.02±.17(5)	1.09±.25(5)	0.77±.36(5)	0.88±.15(7)
Ω_G in 3D	0.62±.05(13)	0.96±.05(13)	0.55±.02(14)	1.27±.02(10)	1.02±.03(10)
Ω_G in 6D	0.55±.29(11)	1.21±.13(9)	0.90±.33(10)	0.97±.24(11)	1.00±.02(11)
Ω_G in 12D	0.44±.38(9)	1.12±.06(8)	0.71±.06(8)	1.18±.14(7)	1.11±.05(9)
Ω_G in 24D	not enough data	1.42±.23(5)	0.67±.05(4)	1.50±.48(4)	1.18±.07(7)
Ω_G in 48D	not enough data	1.04±.10(5)	0.70±.49(4)	1.10±.15(6)	1.08±.16(4)

Finally, we compare CPU times for 2% relative accuracy in Table 4. The fast growth of computational time (due to the larger bias and variance) along with increased dimension is also reported there.

Problem II suits itself well to assessing the importance of having analytical formulas for $\lambda_{\max}(A)$ and σ^{-1} when using the WoE in high dimensions. Here, WoE computes and stores both $\lambda_{\max}(A)$ and σ^{-1} before start. Since A in (5.10) is constant, $\lambda_{\max}(A)$ can be

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

Table 4: Comparable computing times (in seconds) taken to solve Example II down to a relative statistical error equal or less than 1%. Same domains and $\mathbf{x}_0$ as in Table 3. WoE uses analytic formulas for σ^{-1} and $\lambda_{\max}(A)$, while WoE* solves the linear system with σ and uses Gershgorin's circles.

Domain	*GM*	*RW*	*BB*	*WoE*	*WoE*∗
Ω_E in $3D$	0.31	0.18	0.41	0.34	1.11
Ω_E in $6D$	6.8	12.9	6.1	9.58	14.03
Ω_E in $12D$	153	296	121	216.50	327.85
Ω_E in $24D$	3045	835	4392	3320.27	4149.98
s in: cost $\sim D^s$	4.3	1.7	5.0	4.9	3.7
Ω_G in $3D$	3.7	30.7	2.7	5.88	12.29
Ω_G in $6D$	53	31	31	93.89	128.62
Ω_G in $12D$	297	1708	341	596.77	853.59
Ω_G in $24D$	1086	too long	1015	3199.99	4171.26
s in: cost $\sim D^s$	1.9	–	1.8	2.4	2.3

precalculated numerically in any dimension, and the inverse of σ in (5.10) is trivial:

$$\sigma^{-1}=\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & -1 & 1 \end{pmatrix}. \tag{5.13}$$

On the other hand, in a BVP where $A=A(\mathbf{x})$ instead of constant, there may be no analytical formulas available for $\lambda_{\max}[A(\mathbf{x})]$ and $\sigma^{-1}(\mathbf{x})$. In that case, as pointed out in Section 4.4, the best approach is to solve $\boldsymbol{b}(\mathbf{x})=\sigma(\mathbf{x})\boldsymbol{\mu}(\mathbf{x})$ (exploiting the row-echelon form of σ) and bound $\lambda_{\max}[A(\mathbf{x})]$ above using Gershgorin's theorem. In Table 4, we have appended a column for the 'WoE*' integrator, which does this. Computational times for WoE* in Table 4 are definitely larger than for WoE, but only moderately so. This is encouraging, for it suggests that the WoE is viable also when analytical formulas for $\lambda_{\max}[A(\mathbf{x})]$ are not available.

5.5 Examples III and IV: Poisson's equation in high dimension

Next, we tackle two problems solved by Buchmann and Petersen to test BP. First, we consider problem 4.2.1 in [5] (**Example III**):

$$\frac{1}{2}\nabla^2 u=-1, \qquad g=\sum_{i=1}^{D}x_i \Rightarrow u=(1-||\mathbf{x}||^2)/D+\sum_{i=1}^{D}x_i, \tag{5.14}$$

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

Table 5: Fitted δ in Example III, and number of points included in the fit.

D	EM	GM	RW	WoE	BB	BP
16	$0.52\pm.01(17)$	$1.06\pm.05(14)$	$0.53\pm.05(14)$	$1.00\pm.002(13)$	$0.99\pm.02(14)$	$1.11\pm.03(13)$
32	$0.54\pm.02(16)$	$1.05\pm.03(14)$	$0.65\pm.04(14)$	$1.00\pm.002(12)$	$1.01\pm.01(15)$	$1.21\pm.06(9)$
48	$0.56\pm.02(16)$	$1.05\pm.04(14)$	$0.65\pm.04(13)$	$1.00\pm.001(11)$	$0.99\pm.02(15)$	$1.34\pm.06(7)$
64	$0.57\pm.02(15)$	$1.04\pm.03(14)$	$0.71\pm.07(13)$	$1.00\pm9\times10^{-4}(11)$	$0.99\pm.01(15)$	$1.33\pm.08(7)$

Table 6: Comparable computing times in seconds for solving Example III with $a=0.005u_{ex}$.

D	EM	GM	RW	WoE	BB	BP
16	500	11.6	103.1	94.3	25.6	19.3
24	360	14.7	98.5	163.6	26.9	17.9
32	300	15.5	83.8	291.3	42.7	5.9
48	278	18.5	95.7	469.7	42.9	22.0

in a $D-$dimensional ball $\Omega_B(1,[0,\cdots,0])$. Notice that $\sigma=I_D$ (so that $\lambda_{\max}=1$), and that the exact solution is not an extension of the BC. Results for the weak order of convergence at $\mathbf{x}_0=(.9,0,\cdots,0)$ in $D=\{16,32,48,64\}$ are listed in Table 5, and comparable computing times for $a=0.5\%$ in Table 6. RW and WoE use the exact value of $\lambda_{\max}=\lambda(I)=1$. Here, computational times are only mildly sensitive to D except for the WoE, even though the problem has no drift to be removed in the first place.

Finally, **Example IV** is problem 4.2.2b in [5]. The BC is a quartic polynomial, and again $\sigma=I_D$:

$$\frac{1}{2}\nabla^2 u=\frac{1}{6}\sum_{i=1}^{32} i(x_i)^2, \qquad g=u=\frac{1}{6}\sum_{i=1}^{32} i(x_i)^4. \tag{5.15}$$

We take $\mathbf{x}_0=(5/100,\cdots,5/100)\in\mathbb{R}^{32}$ in $\Omega_B(1,[0,\cdots,0])$, $\Omega_G(1,[1/10,\cdots,1/10])$, and $\Omega_E(1,[-1/2,\cdots,-1/2])$. (Approximate distances to the boundary are 0.72,0.28,0.45, respectively.) The relative error behaviour is shown in Figs. 9 and 10. Here, we do not report δ because there are not enough data with $r_{h,N}\leq 0.15$ for a fit. For all the schemes but EM and RW, it looks like $\delta\to 1$. While BP does not improve on BB in terms of accuracy in Ω_B and Ω_G (Fig. 9), we remark the striking results in Ω_E (Fig. 10). There, BP is the only integrator, along with EM, which converges at all.

6 Conclusions

In the light of the numerical evidence reported in this paper, it is difficult to pick an obvious winner from the five schemes compared. Instead, let us briefly comment on

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

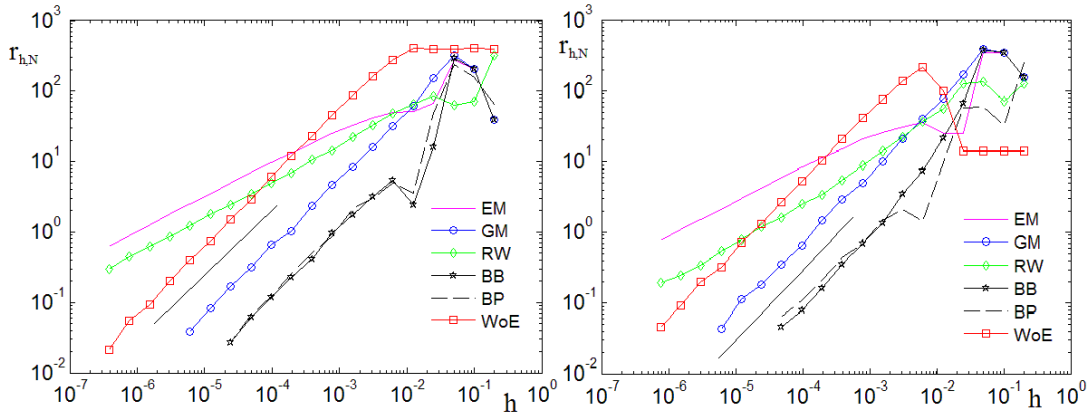


Figure 9: Convergence of relative errors for Problem IV in 32D Ω_B (left) and Ω_G(right).

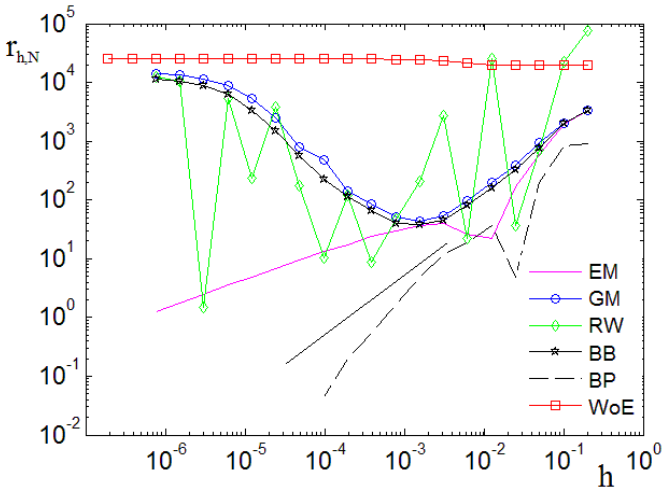


Figure 10: Convergence of relative errors for Problem IV in 32D Ω_E. The point with the smallest error with BP took around 1.2 million seconds, the longest simulation reported in this paper.

each of them:

- The boundary shift method by Gobet and Menozzi is reliable, fast and extremely easy to implement. Since it is basically Euler-Maruyama's method (on a slightly different domain), it inherits most of the available theory. For this reason, it was chosen for a Multilevel implementation [11].
- The Brownian bridge is also robust but somewhat less accurate (or slower) than the Gobet-Menozzi method, except if the diffusion is a Brownian motion. On the other hand, in the niche of Poisson equations, the method of Buchmann and Petersen is the scheme of choice, as clearly demonstrated by Example IV. Some numerical

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

tests (not reported here) also suggest that BP yields important improvements in non-autonomous bounded SDEs. This is reasonable, due to the better accuracy in determining the exit time.

- Despite its simplicity, the Random Walk method has been a disappointment, for it seems to be very sensitive to parameters not always easy to approximate and to the presence of corners in the domain.
- Finally, Milstein's Walk on Ellipsoids—here combined with our half-space approximation to determine the largest tangent ellipsoid—seems to us the most stable in terms of yielding $\delta = 1$ in nearly all the experiments, regardless of the dimension and the geometry. On the other hand, for an SDE involving a complicated diffusion and geometry (such as Examples I and II) where a really small tolerance a is required, WoE would be our choice. Moreover, it is the basis for a linear scheme for general SDEs reflected on the boundary [29, Section 6.6].

The point of this paper has been to show that weak orders of convergence better than $\mathcal{O}(\sqrt{h})$ for numerical approximations of bounded stopped diffusions are in general attainable, and quite often linear (or nearly so). This translates into large savings in computational times. The most significant challenge now is to raise the weak order of convergence in SDEs featuring non-smooth coefficients, reflecting BCs, or boundaries with sharp corners. Another important direction for improvement is the analysis of the strong convergence properties of higher-order integrators, in order to insert them into the Multilevel framework.

Acknowledgments

This work was supported by Portuguese national funds through FCT under grants UID/CEC/50021/2013 and PTDC/EIA-CCO/098910/2008. FB also acknowledges FCT funding under grant SFRH/BPD/79986/2011.

References

[1] J.A. Acebrón, M.P. Busico, P. Lanucara, and R. Spigler, *Domain decomposition solution of elliptic boundary-value problems via Monte Carlo and quasi-Monte Carlo methods*. SIAM J. Sci. Comp. **27**(2), 440-457 (2005).
[2] P. Baldi, *Exact asymptotics for the probability of exit from a domain and applications to simulation*. Ann. Probab. **23**(4), 1644-1670 (1995).
[3] C. Bayer, A. Szepessy, and R. Tempone, *Adaptive weak approximation of reflected and stopped diffusions*. Monte Carlo Methods and Applications **16**, 1-67 (2010).
[4] F. Bernal, J.A. Acebrón, and I. Anjam, *A stochastic algorithm based on fast marching for automatic capacitance extraction in non-Manhattan geometries*. SIAM J. Imag. Sci. **7**(4), 2657-2674 (2014).
[5] F.M. Buchmann and W.P. Petersen, *An exit probability approach to solving high dimensional Dirichlet problems*. SIAM J. Sci. Comput. **28**, 1153-1166 (2006).
[6] F.M. Buchmann, *Simulation of stopped diffusions*. J. Comput. Phys. **202**(2), 446-462 (2005).

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

[7] C. Constantini, B. Pacchiarotti, and F. Sartoretto, *Numerical approximation for functionals of reflecting diffusion processes*. SIAM J. Appl. Math. **58**, 73-102 (1998).
[8] M. Deaconu and A. Lejay, *A Random Walk on Rectangles algorithm*. Methodol. Comput. Appl. Probab. **8**, 135-151 (2006).
[9] M. Freidlin, Functional Integration and Partial Differential Equations. Princeton University Press (1985).
[10] M.B. Giles, *Multi-level Monte Carlo path simulation*. Operations Research, **56**(3), 607-617 (2008).
[11] M.B. Giles and F. Bernal, *Multilevel simulations of expected exit times and other functionals of stopped diffusions*. In preparation. (Unpublished)
[12] M.T. Giraudo and L. Sacerdote, *An improved technique for the simulation of first passage times for diffusion processes*. Comm. Statist. Simulation Comput. **28**(4), 1135-1163 (1999).
[13] P. Glasserman, Monte Carlo Methods in Financial Engineering. Springer (2003).
[14] E. Gobet and S. Menozzi, *Stopped diffusion processes: Overshoots and boundary correction*. Stoch. Proc. Appl., **120**, 130-162, (2010).
[15] E. Gobet, *Euler schemes for the weak approximation of killed diffusion*. Stoch. Process. Appl. **87**, 167-197 (2000).
[16] E. Gobet, *Euler schemes and half-space approximation for the simulation of diffusions in a domain*. ESAIM: Probab. Statist. **5**, 261-297 (2001).
[17] D.J. Higham, X. Mao, M. Roj, Q. Song, and G. Yin. *Mean exit times and the multilevel Monte Carlo method*. SIAM J. Uncertainty Quantification, **1**(1), 2-18 (2013).
[18] G. Iyer, A. Novikov, L. Ryzhik, and A. Zlatos, *Exit times for diffusions with incompressible drift*. SIAM J. Math. Anal. **42**, 2484-2498 (2010).
[19] K.M. Jansons and G.D. Lythe, *Efficient numerical solution of stochastic differential equations using exponential timestepping*. J. Statist. Phys. **100**(5-6), 1097-1109 (2000).
[20] P.E. Kloeden and E. Platten, Numerical Solution of Stochastic Differential Equations. Springer (1999).
[21] P. Lemaire and G. Pagés, *Multilevel Richardson-Romberg extrapolation*. To appear in Bernoulli, (2015).
[22] S. Mairé and E. Tanré, *Some new simulation schemes for the evaluation of Feynman-Kac representations*. Monte Carlo Methods and Applications, **14**(1), 29-51 (2008).
[23] S. Mairé and M. Simon, *A partially reflecting random walk on spheres algorithm for electrical impedance tomography*. Preprint in arXiv:1502.04318 (2015). (Unpublished)
[24] S. Mancini, F. Bernal, and J.A. Acebrón, *An efficient algorithm for accelerating Monte Carlo approximations of the solution to boundary value problems*. J. Sci. Comput. **66**(2), 577-597 (2016).
[25] R. Mannella, *Absorbing boundaries and optimal stopping in a stochastic differential equation*. Phys. Lett. A **254**(5), 257-262 (1999).
[26] M. Mascagni and N.A. Simonov, *Monte Carlo methods for calculating some physical properties of large molecules*, SIAM J. Sci. Comput. **26**, 339-357 (2004).
[27] J.R. Michael, W.R. Schucany, and R.W. Haas, *Generating random variates using transformations with multiple roots*. Am. Statist. **30**, 88-90 (1976).
[28] G.N. Milstein, *Weak approximation of a diffusion process in a bounded domain*. Stoch. Stoch. Rep. **62**, 147-200 (1997).
[29] G.N. Milstein and M.V. Tretyakov, Stochastic Numerics for Mathematical Physics. Springer, Berlin (2004).
[30] G.N. Milstein and M.V. Tretyakov, *The simplest random walks for the Dirichlet problem*. Teor. Veroyatnost. I Primenen. **47**(1), 39-58 (2002).
[31] C. Miranda, Partial Differential Equations of Elliptic Type. Springer (1970).

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press

[32] M.E. Muller, *Some continuous Monte Carlo methods for the Dirichlet problem*. Ann. Math. Statist. **27**, 569-589 (1956).
[33] S. Redner, A Guide to First-Passage Processes. Cambridge University Press (2001).
[34] J.T.C. Schwabedal and A. Pikovsky, *Phase description of stochastic oscillations*, Phys. Rev. Lett. **110**(20), 204102 (2013).
[35] M. Tamborrino, L. Sacerdote, and M. Jacobsen, *Weak convergence of marked point processes generated by crossings of multivariate jump processes. Application to neural network modeling*. Physica D, **288**, 45-52 (2014).
[36] D. Talay and L. Tubaro, *Expansion of the global error for numerical schemes solving stochastic differential equations*. Stoch. Anal. App., **8**(4), 94-120 (1990).

https://doi.org/10.4208/cicp.OA-2015-0016 Published online by Cambridge University Press