An exit probability approach to solve high dimensional Dirichlet problems

F.M. Buchmann and W.P. Petersen

Research Report No. 2004-09
October 2004

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

An exit probability approach to solve high dimensional Dirichlet problems

F.M. Buchmann and W.P. Petersen

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Research Report No. 2004-09 October 2004

Abstract

We present an approach to solve high dimensional Dirichlet problems in bounded domains which is based on a variant of the Feynman-Kac formula that connects solutions to elliptic partial differential equations and functional integration. We integrate the resulting system of stochastic differential equations numerically with the Euler scheme. To correct for possible intermediate excursions of the simulated paths and to find good approximations for first exit times, we extend to higher dimensions a strategy which we have introduced in [1] for stopping problems in $1d$. In addition, for Dirichlet's problem, good approximations of the first exit points are needed to evaluate the boundary condition. To this end, we sample a random point on the tangent hyperplane of the boundary of the domain once a first exit was estimated. To detect excursions and adequately compute first exit times, we apply the same local half space approximation. We therefore assume that the domain is smooth enough such that these approximations are possible.

Numerical experiments in dimensions up to 128 show that resulting approximations are of very high quality. In particular, we observe that first order convergence behavior of the Euler scheme can be maintained.

1 Introduction and notation

In this work we present an approach to solve numerically Poisson's equation in high dimensional domains. Our approach is based on the stochastic representation of the solution which is given by some variant of the Feynman-Kac formula. Numerically, we have to integrate a system of stochastic differential equations and will approximate mathematical expectations by finite sum arithmetic means (Monte-Carlo approach). In the stochastic formulation, a Brownian path starting from a point (x) in the domain is simulated until it reaches the boundary for the first time and the exit point must be estimated to compute the boundary contribution to an average of a functional. It is this special feature (stopping at first exit) that makes the problem especially interesting from a numerical point. Namely, the numerical approximation of the first exit time and the first exit point is essential for good convergence properties of the resulting algorithm.

In more detail, we consider the following boundary value problem for $u : \mathbf{R}^n \to \mathbf{R}$,

$$\frac{1}{2}\triangle u(x) + g(x) = 0 \quad \text{for} \quad x \in D, \quad \text{and} \quad u(x) \to \psi(x) \quad \text{as} \quad x \to \partial D. \tag{1}$$

Here, D denotes a bounded domain in n-space. We will assume throughout this work that D, ∂D, the inhomogeneity g and boundary condition ψ (both functions are $\mathbf{R}^n \to \mathbf{R}$) are such that a unique solution exists which is sufficiently smooth, see for example [2]. Then $u(x)$ is given by the Feynman-Kac formula

$$u(x) = \mathbb{E}\left[\psi\left(X\left(\tau^D\right)\right) + \int_0^{\tau^D} g\left(X(s)\right)\,\mathrm{d}s\right]. \tag{2}$$

In (2) X is an n-dimensional Brownian motion starting at $t = 0$ at $x \in D$ and τ^D is the first exit time of the process $X(t)$ from D (which by continuity is also the first passage time to ∂D),

$$\tau^D = \inf_{t>0}\left\{X(t) \notin D\right\} = \inf_{t>0}\left\{X(t) \in \partial D\right\}, \quad X(0) = x \in D. \tag{3}$$

Consequently, $X(\tau^D)$ is the first exit point of the process from D (the point where the boundary ∂D is hit for the first time).

Throughout this paper we will write $\xi \sim \mathcal{D}$ if a random number ξ (or a random variable) has the distribution (or density) $\mathcal{D}$. In particular we will make use of the distributions $\mathcal{N}(m, v)$ (normal with mean m and variance $v > 0$) and $\mathcal{IG}(\gamma, \delta)$ (inverse Gaussian with parameters $\gamma > 0$, $\delta > 0$, see [3]).

2 Weak approximation of stopped diffusions in dimension one

We have presented an algorithm to simulate stopped diffusions (in a weak sense) in [1, 4, 5]. In order to make this article self-contained we summarize it here briefly.

Let $D = (a, b) \subset \mathbf{R}$ be a bounded interval and suppose we wish to approximate $\mathbb{E}[f(\tau)]$ where

$$f(\tau) = \int_0^{\tau} g(X(s))\,\mathrm{d}s \quad \text{for some} \quad g : \mathbf{R} \to \mathbf{R}. \tag{4}$$

We again assume that the function g is such that the integral in (4) is well defined. In (4), $X(t)$, $t \geq 0$, is a one dimensional Brownian motion starting at $x \in D$ and $\tau = \inf\{t > 0 : X(t) \notin D\}$ (see (3) in §1).

The following formulation in terms of stochastic differential equations is equivalent:

$$\mathrm{d}X(t) = \mathrm{d}W(t) \quad \text{and} \quad \mathrm{d}f(t) = g(X(t))\,\mathrm{d}t, \quad 0 \leq t < \tau, \tag{5}$$

with initial conditions $X(0) = x$ and $f(0) = 0$. A weak approximate solution of system (5) is nowadays standard (see for example [6, 7]).

Our algorithm is based on the simplest possible scheme to numerically integrate the system (5), the Euler-Maruyama scheme. For a fixed final time $T < \infty$ Euler's method takes the following form: We partition T into K intervals, $T = Kh$, each of size h. Then we set $X_0 = 0$ and $f_0 = 0$, and compute for $k = 0, \dots, K-1$

$$X_{k+1} = X_k + \sqrt{h}\xi, \quad \xi \sim \mathcal{N}(0,1), \tag{6a}$$

$$f_{k+1} = f_k + hg(X_k). \tag{6b}$$

If we apply this strategy to approximate $\mathbb{E}[f(\tau)]$ (see (4)), the main difficulty presents itself immediately. In (4), stopping time τ is not known but must be determined by equation (3). On the other hand, we are only interested in finding a good approximation of the expecation of $f(\tau)$ (as in equation (2)). Suppose that we can approximate $f(\tau)$ and denote corresponding realizations $f^{(i)} := f^{(i)}_{K^{(i)}}$ (the number of steps changes from realization to realization and will be made precise in what follows: see Algorithm 2.1) . We then approximate the expectation by a finite mean over the N realizations as

$$\mathbb{E}[f(T)] \approx \frac{1}{N}\sum_{i=1}^{N} f^{(i)}. \tag{7}$$

Our algorithm takes advantage of the fact that we only want the mean of the approximation of f to be close to $\mathbb{E}[f(\tau)]$, but do not need pathwise convergence. In other words, we need good approximations in a *weak* and not *strong* (or *mean-square*) sense. The better the approximate distributions are, the better the approximation of $\mathbb{E}f$ will be.

To describe our method, consider the one dimensional case $D = (-\infty, b)$ with $x < b$. Then τ is the first hitting time of level b of the process $X(t)$. Our algorithm extends a method to approximate killed diffusions [8, 9] in a natural way: For the weak approximation of killed diffusions if suffices to know *if* $\tau < T$ where $T > 0$ is a fixed (deterministic) value. In that case a path is killed if $\tau < T$, i.e. it does not contribute to an expectation at T if it reached level b before T. For stopped diffusions, however, an approximation of τ is needed because we need to know *when* the first hitting of b actually occured. Algorithm 2.1 gives this approximation and we can add a last Euler step of appropriate length (either $\mathcal{T}_1$ or $\mathcal{T}_2$) to the approximation of the integral (4).

Clearly, we start for every path with $X_0 = x$ and $f_0 = 0$. For $k = 0, 1, \dots$ we proceed as described in Algorithm 2.1.

Remark 1 In $n-$dimensional cases, locally we apply a half-space approximation of D close to the boundary, which enales us to reduce our problem to this one dimensional situtation [9].

Algorithm 2.1

```
 1: Set $y \leftarrow X_k$, generate $\xi \sim \mathcal{N}(0,h)$ and set $z \leftarrow y + \xi$. {preliminary Euler step}
 2: if $z < b$ then {depending on the value of $z$ there are two possibilities}
 3:    Generate $u \sim \mathcal{U}(0,1)$ and set $\mathcal{T}_1 \leftarrow -2(b-y)(b-z)/\log u$
 4:    if $\mathcal{T}_1 \leq h$ then {compare with the step size $h$: test for excursion}
 5:       Set $f(\tau) \leftarrow f_k + \mathcal{T}_1 g(y)$ {last Euler step of length $\mathcal{T}_1$.}
 6:       stop
 7:    else
 8:       Set $f_{k+1} \leftarrow f_k + hg(y)$, $X_{k+1} \leftarrow z$ and goto item 1 {continue integrating}
 9:    end if
10: else {$z \geq b$ implying $\tau \leq t_{k+1}$}
11:    Generate $s \sim \mathcal{IG}((b-y)^2/h, (b-y)/(z-b))$ and set $\mathcal{T}_2 \leftarrow hs/(1+s)$
12:    Set $f(\tau) \leftarrow f_k + \mathcal{T}_2 f(y)$ {(last Euler step of length $\mathcal{T}_1$).}
13:    stop
14: end if
```

Algorithm 2.1 is executed for N independent sample paths (the sample size) and then the mathematical expectation is approximated by the finite mean, see (7) (Monte-Carlo method, see [10] and references therein).

After having stated our method, let us summarize briefly the formulae on which it is based. This will motivate our approach in higher dimensional settings (§3), see [1] for the derivation. Let $\mathbb{P}_{y,h,z}[\cdot]$ be the law of a Brownian bridge pinned at time-space coordinates (t_k, y) and (t_{k+1}, z) (where $t_{k+1} = t_k + h$). Recall that we set $y = X_k$ and $z = X_{k+1}$ and suppose $D = (-\infty, b)$ with $x < b$, i.e.

$$\tau = H_b(x) := \inf_{t>0} \{X(t) = b,\ X(0) = x\} \tag{8}$$

is the first hitting time of level b. We then have [11]

- For $y, z < b$:

$$\mathbb{P}_{y,h,z}\left[H_b(y) \leq t\right] = \exp\left(-\frac{2}{t}(b-y)(b-z)\right), \quad t > 0. \tag{9}$$

- For $y < z < b$:

$$\mathbb{P}_{y,h,z}[\tau \in \mathrm{d}t]$$
$$= \frac{b-y}{\sqrt{2\pi}}\sqrt{\frac{h}{t^3(h-t)}}\exp\left(-\frac{(h(b-y)-t(z-y))^2}{2ht(h-t)}\right)\mathbf{1}_{\{0<t<h\}}\,\mathrm{d}t \tag{10}$$

and after the substitution $x = t/(1-t) \geq 0$ (hence $t = x/(1+x)$)

$$= \sqrt{\frac{\gamma}{2\pi x^3}}\exp\left(-\frac{\gamma(x-\delta)^2}{2\delta^2 x}\right)\mathbf{1}_{\{x>0\}}\,\mathrm{d}x \tag{11}$$

where $\gamma = (b-y)^2/h$ and $\delta = (b-y)/(z-b)$. We recall that a random variable ξ *follows the inverse Gaussian distribution with parameters* $\gamma, \delta > 0$ if it has the density (11). In this case, we write $\xi \sim \mathcal{IG}(\gamma, \delta)$. We refer to [3] for a review on this distribution and to [12] for an algorithm to generate random variables $\xi \sim \mathcal{IG}(\gamma, \delta)$. This allows us (using (11)) to generate a random variable with density (10).

Remark 2 In the case that $z = b$ (which has probability zero in theory but might occur in finite-precision arithmetic on a computer) we set $\tau \approx t_{k+1}$ and hence stop the integration as $f(\tau) \approx f_k + hg(X_k)$.

3 Higher dimensions

We now turn our attention to the approximation of the solution $u(x)$ of (1) with $x \in D \subset \mathbf{R}^n$ with $n \geq 2$. From (1) we see that we need approximations of both τ and $X(\tau)$. We first describe how to extend the Algorithm 2.1 presented in §2 to higher dimensions and then show howto find $X(\tau)$.

3.1 Reduction to one dimension: half-space approximation

We assume that for any two points $y, z \in \mathbf{R}^n$ there is a unique point on the boundary ∂D which is "closest" to y and z in some sense (we make this statement precise later on). Denote this point by X_b and suppose further, that ∂D is sufficiently smooth in a sufficiently large neighbourhood of X_b, such that we can approximate ∂D locally by a tangent hyperplane. For a bounded domain $D \subset \mathbf{R}^n$, every point $X_b \in \partial D$ can be mapped by a simple scaling and rotation to $(b, 0, \ldots, 0)^T$ with $b > 0$ (see §3.4 for an efficient implementation of these rotations using Givens tranformations). To derive our algorithm, we therefore assume such a transformation has been done and that near ∂D, D is given by the half-space $\{v \in \mathbf{R}^n : v^1 < b\}$ with boundary (hyper plane) $\{v \in \mathbf{R}^n : v^1 = b\}$. Suppose $x^1 < b$. We then have (compare with (8))

$$\tau = H_b(x) := \inf_{t>0} \left\{X^1(t) = b,\ X(0) = x\right\}.$$

3.2 Approximation of the first exit time

For $y, z \in \mathbf{R}^n$ with $y^1, z^1 < b$ we have (compare with (9))

$$\mathbb{P}_{y,h,z}\left[H_b(y) \leq t\right] = \exp\left(-\frac{2}{t}(b-y^1)(b-z^1)\right), \quad t > 0,$$

whilst for $y^1 < b$ but $z^1 > b$ it holds that (compare with (10))

$$\begin{aligned}&\mathbb{P}_{y,h,z}[H_b(y) \in \mathrm{d}t] \\ &\quad = \frac{b-y^1}{\sqrt{2\pi}}\sqrt{\frac{h}{t^3(h-t)}}\exp\left(-\frac{\left(h(b-y^1)-t(z^1-y^1)\right)^2}{2ht(h-t)}\right)\mathbf{1}_{\{0<t<h\}}\,\mathrm{d}t.\end{aligned}$$

The following observation is immediate:

Remark 3 In the case of a boundary connected to only one of the components of an n-dimensional Brownian motion (and assuming that the boundary is locally flat, i.e. (locally) a hyperplane), the fact that these individual components are *n independent* one dimensional Brownian motions leads to formulae as simple as in the one dimensional setting. Furthermore, only the components connected to the boundary show up in the equations.

Therefore, extension of Algorithm 2.1 to higher dimensions is evident and we shall skip the details. Only the exit point estimation requires further elaboration.

3.3 Approximation of the exit point

In the previous section we described how to approximate the first exit time $\tau = \tau_D$ of Brownian motion from a domain D in n-space. As noted earlier, in order to approximate the solution $u(x)$ of Poisson's equation (1), by its stochastic representation (2) $u(x) = \mathbb{E}[\psi(X(\tau)) + f(\tau)]$, an approximation of the exit point $X(\tau)$ is required as soon as the boundary condition ψ is no longer a constant.

Using the notation introduced earlier, we cite the result which is the starting point for the approximation of $X(\tau)$, see [11, Lemma 7]. We recall that $\tau = \inf_{t>0}\{X^1(t) \geq b\}$. Assume $z^1 > b$. Given $X^1(s)$, $s \in [0, \tau]$, the joint distribution of $X^i(\tau)$ conditioned on $X(t) = z$ is given by [11, Lemma 6] (with $X(0) = 0$)

$$z^i\frac{\tau}{t} + \sqrt{\tau\left(1-\frac{\tau}{t}\right)}\cdot\xi^i \quad \text{where} \quad \xi^i \sim \mathcal{N}(0,1), \quad i = 2,\ldots,n. \tag{12}$$

The assumption that $z^1 > b$ (or equivalently that $z^1 \geq b$) can be replaced by the assumption that $\tau < t$ (or $\tau \leq t$ respectively). We can therefore apply (12) as soon as the integration process is stopped (as soon as an approximation of τ is found by the extension of Algorithm 2.1 to higher dimensions using localy half space apprximations of D). The extension to the general case with $y = X_k$ and $z = X_{k+1}$ with $\tau \leq t_{k+1}$ is then straightforward and yields for the resulting joint distribution of $X^i(\tau)$

$$y^i + (z^i - y^i)\frac{\tau}{h} + \sqrt{\tau\left(1-\frac{\tau}{h}\right)}\cdot\xi^i \quad \text{where} \quad \xi^i \sim \mathcal{N}(0,1), \quad i = 2,\ldots,n. \tag{13}$$

It only remains to incorporate the rotation of the map $X_b \mapsto (||X_b||, 0, \ldots, 0)^T$. This rotation can be described by multiplication with an orthogonal matrix R, but we used a more efficient implementation using Givens rotations, described below. We then have $||\tilde{X}_b|| = ||X_b||$ because R is orthogonal. The exit point of the transformed vector $\tilde{X}_b$ can be found from equation 13 by replacing $x^i \to \tilde{x}^i$ and $y^i \to \tilde{y}^i$. With this formula for the $\tilde{x}, \tilde{y}$ coordinates, we can thus *sample* an approximation for $\tilde{X}(\tau)$ on the tangent hyper-plane through $(||X_b||, 0, \ldots, 0)^T$. Once $\tilde{X}_b$ is generated, we project back to ∂D via equation 17.

Rotating back, $(||X_b||, 0, \ldots, 0)^T \mapsto X_b$, is easily affected by the inverse $R^{-1} = R^T$, gives us the approximation for the exit point as $X(\tau) \approx R^{-1}\tilde{X}_b$, where this inversion is applied by $n-1$ pair-wise rotations of equation (17).

Algorithm 3.1 Simulate one path (with index (i)) to get an approximation $\overline{u}^{(i)}$ for $u^{(i)} = \psi(X^{(i)}(\tau^{(i)})) + f^{(i)}(\tau)$ with $u(x) = \mathbb{E}[u^{(i)}]$. We assume that D is the unit-n-sphere and further omit the sample index (i) for better readibility.

```
$y \leftarrow x$, $f \leftarrow 0$ {initialize with initial conditions of SDEs}
for $k = 0, 1, 2, \ldots$ do {integrate until $\tau$, see Algorithm 2.1}
    $z^i = y^i + \sqrt{h}\xi$, $\xi \sim \mathcal{N}(0,1)$ for $i = 1, \ldots, n$
    $X_b \leftarrow$ a point on $\partial D$ closest to $y$ and $z$
    $\tilde{y} \leftarrow Ry$ and $\tilde{z} \leftarrow Rz$, see §3.4
    if $\tilde{z}^1 \geq 1$ then {$D$ was clearly left}
        Generate $\mathcal{T}_2$ according to (10,11) with input variables $\tilde{y}^1$, $\tilde{z}^1$ and $b = 1$.
        $f(\tau) \leftarrow f + \mathcal{T}_2 g(y)$
        stop
    else {check for an intermediate excursion}
        Generate $\mathcal{T}_1$ (see (9) with input variables $\tilde{y}^1$, $\tilde{z}^1$ and $b = 1$)
        if $\mathcal{T}_1 \leq h$ then {excursion detected}
            $f(\tau) \leftarrow f + \mathcal{T}_1 g(y)$
            stop
        else {continue integration}
            $f \leftarrow f + hg(y)$ and $y \leftarrow z$
        end if
    end if
end for
$\tilde{X}^1 \leftarrow 1$ and for $i = 2, \ldots, n$ generate $\tilde{X}^i$ according to (13)
$\tilde{X} \leftarrow \tilde{X}/|\tilde{X}|$ {project back on $\partial D$}
$X(\tau) \leftarrow R^{-1}\tilde{X}$, see §3.4
$\overline{u} \leftarrow \psi(X(\tau)) + f(\tau)$
```

3.4 On the implementation of the rotations

Once a suitable $X_b \in \partial D$ is found, we have to map the situation to the case where

$$\tilde{X}_b = (||X_b||, 0, \ldots, 0)^T.$$

Then, above operation is a rotation in n-space which can be written as $RX_b = \tilde{X}_b$ with an orthogonal $(n \times n)$-matrix R. In particular, we need $\tilde{y} = Ry$ and $\tilde{z} = Rz$ to generate $\mathcal{T}_1$ and $\mathcal{T}_2$ respectively.

Of course, one could simply calculate R and perform two matrix-vector-products, but this would result in an algorithm scaling like $\mathcal{O}(n^2)$. Because these rotations have to be done in every step, this is clearly too expensive computationally, especially if the dimension n increases. Also, suppose that we have some point $\tilde{u}$ in the transformed system and we additionally need the "original" point u, $u = R^T\tilde{u}$. This would be another $\mathcal{O}(n^2)$-operation.

We therefore used a procedure which has $\mathcal{O}(n)$ operations. It uses Givens rotations [13, Section 3.4] and is described below. Because each of a sequence of Givens rotations is unitary, the inverse rotation is simply the sequence of transposes applied in reverse order. Both directions are norm preserving.

To illustrate, let $X_b = (X_b^1, \ldots, X_b^n) =: b_{(1)} \in \mathbf{R}^n$ and define updated versions of X_b by $b_{(k)}$, for $k \geq 2$. Subscripts (k) indicate that the $k-$th element of the updated $b_{(k)}$ vector is zeroed. The

$k-$th rotation to compute the update $b_{(k)} = R_{(k)} b_{(k-1)}$, for $k = 2, \dots n$ having $\cos, \sin$ parameters in the $1-$st and $k-$th rows, uses

$$R_{(k)} = \begin{pmatrix} \cos(\varphi_{(k)}) & & & & -\sin(\varphi_{(k)}) & & & \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & 1 & & & & \\ \sin(\varphi_{(k)}) & & & & \cos(\varphi_{(k)}) & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{pmatrix} \tag{14}$$

with

$$\cos(\varphi_{(k)}) = \frac{b^1_{(k-1)}}{w_{(k-1)}} \quad \text{and} \quad \sin(\varphi_{(k)}) = \frac{-b^k_{(k-1)}}{w_{(k-1)}}$$

and

$$w_{(k-1)} = ((b^1_{(k-1)})^2 + (b^k_{(k-1)})^2)^{\frac{1}{2}}. \tag{15}$$

Proceeding in the same way, after $n-1$ such rotations, we have

$$b_{(n)} = R_{(n)} R_{(n-1)} \cdots R_{(2)} X_b. \tag{16}$$

It is easy to see that each $R_{(k)}$ (14) is unitary, $R^T_{(k)} R_{(k)} = \mathbb{I} = R_{(k)} R^T_{(k)}$, so the inverse transformation is performed in the reverse order:

$$R^{-1} = R^T = R^T_{(2)} R^T_{(3)} \cdots R^T_{(n)}. \tag{17}$$

The rotations $\tilde{y} = Ry$ and $\tilde{z} = Rz$ can be computed simultaneously, and likewise their inverses.

The calculated trigonometric tables $\cos(\varphi_{(i)}), \sin(\varphi_{(i)})$ from equation (16) are stored for the connected inverse transformation ($R^{-1} = R^T$) stored in two arrays c and s. These arrays thus have length $n-1$.

The procedure used in our experiments used Givens rotations, but Householder transformations [14] would do as well. Namely, X_b may be projected into direction $e_1 = (1, 0, \dots, 0)^T$ by a rank–one update H:

$$HX_b = \begin{pmatrix} ||X_b|| \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

where

$$H = \mathbb{I} - 2\frac{vv^T}{||v||^2}, \tag{18}$$

and vector $v = X_b - ||X_b|| e_1$. The Householder transformation is also unitary, in fact $H = H^T$. This transformation, HX_b, again requires $\mathcal{O}(n)$ operations. Givens rotations have a slight advantage on distributed memory machines (like our Beowulf) because inner products, which require global communications to get $||v||$ and $||X_b||$, are not required.

4 Numerical experiments

We show results from extensive tests performed with the method developped in the previous sections. We first show results from experiments performed when the boundary condition ψ is constant (w.l.o.g. we took $\psi \equiv 0$). Then, sampling of the exit point is unnecessary and we can concentrate of the approximation of $\mathbb{E}[\tau]$. We next show results from experiments with non-constant boundary conditions, hence sampling $X(\tau)$ on the boundary.

4.1 Constant boundary condition

We concentrate on the *exit problem*, i.e. Poisson's equation (1) with $g \equiv 1$ and $\psi \equiv 0$. Then $u(x) = \mathbb{E}[\tau^D]$ is the mean first exit time. Note that for this problem numerical integration using the Euler scheme is exact and any errors in numerical simulation (beside of the statistical error) are due to the stopping at first exit.

4.1.1 Two dimensional square

Our first test is such that no error arising due to the (local) approximation of ∂D by its tangent hyper-plane at the point X_b is present. The flatter ∂D, the smaller this error will be. We therefore start with a simple two dimensional domain without local curvature, i.e. a square.

We set $D = (-1, 1) \times (-1, 1)$ and consider $\triangle u = -2$ with a zero valued boundary condition. Expanding $u(x, y)$ in eigenfunctions, $\cos(\pi x(2k + 1)/2) \cos(\pi y(2l + 1)/2)$, we find

$$u(0) = \frac{128}{\pi^4} \sum_{k,l=0}^{\infty} \frac{(-1)^{k+l}}{(2k+1)(2l+1)\left((2k+1)^2 + (2l+1)^2\right)} \approx 0.5893708\ldots.$$

to compare with our numerical approximations.

When computing the closest boundary point, we did not take into account more than one boundary segment for an excursion test in each step, even if two boundaries were equally far away:

If $z \in D$, we simply took (one of) the boundary segment(s), that minimize(s) the sum of the eucledean distance of y and z to ∂D.

If $z \notin D \cup \partial D$, we assumed that $(X(t))_t$ crossed the boundary ∂D at the same point as a linear interpolated version of the discrete path $(X_k)_k$ would.

In doing so, we introduce an additional error. We neglect this error and do not try to correct for it, see [8] for a discussion and possible solutions to this problem. From our results it will be aparent that for sufficiently small h, these errors can indeed be neglected: asymptotically we get (at least) the desired $\mathcal{O}(h)$-behaviour of the error (see Figure 1 and the conclusions thereafter).

In this first test, we compare the following methods:

(T): This method stops only if $z \notin D$. In this case, it approximates $\tau \approx t_k$.

(K): This method applies a *killing* test and stops always (if $z \notin D$ or if an intermediate excursion is detected) at t_k. In other words, this method is similar to ours, but it ignores the last integration step of length $\mathcal{T}_1$ or $\mathcal{T}_2$ respectively.

(SK): This method is a mixture between methods (K) and (S): In the case of an excursion, integration is stopped at $t_k + \mathcal{T}_1$ and at t_k if $z \notin D$ (the random variable $\mathcal{T}_2$ is not generated).

(S): This is the advocated method which samples both $\mathcal{T}_1$ and $\mathcal{T}_2$.

From the results shown in Figure 1 we draw the following conclusions:

- The trivial stopping procedure as applied by method (T) is clearly not satisfying. Integration stops much too late on average (as the true solution is approached from above, see the left-hand plot) and resulting convergence order is only one half.
- Applying some sort of a posteriori test to detect intermediate excursions remedies the main difficulties. Convergence of all three methods (K), (SK) and (S) is much better.
- Method (K) stops integration too early on average (it always stops at the beginning of a time step). The same holds for method (SK), although applying a stopping test instead of a killing test to detect intermediate excursions reduces corresponding errors by approximately a factor of two.

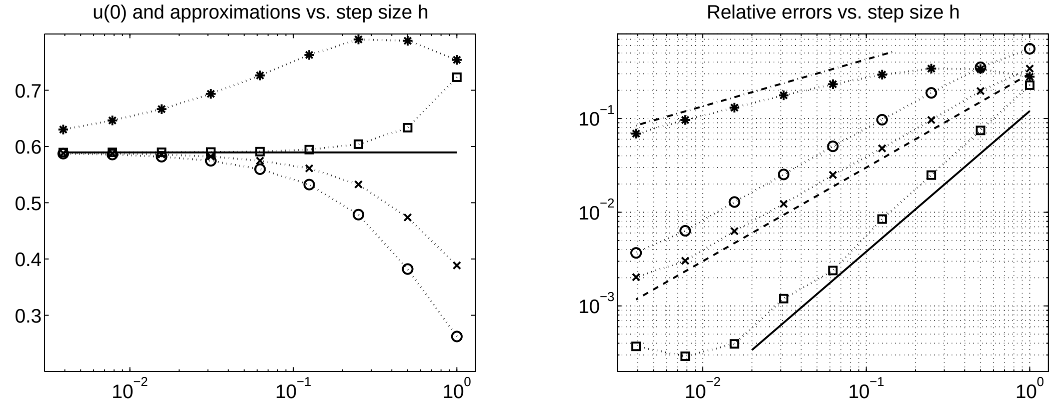

Figure 1: Results for two dimensional square with $g \equiv 1$ and $\psi \equiv 0$. On the left, approximate solutions are shown versus step size h together with the exact solution (which is indicated by a solid line). On the right corresponding relative errors are shown versus step size h. Three lines are added: a dash-dotted with slope one half, a dashed one with slope one and a solid one with slope three halfs. In both plots $\cdots * \cdots$ is method (T), $\cdots \circ \cdots$ is method (K), $\cdots \times \cdots$ is method (SK) and $\cdots \square \cdots$ is method (S).

- Method (S) is clearly superior to all the other methods tested: The approximation of $u(0)$ is very good and connected errors converge with an order higher than one (approximately three halfs).

- Method (S) stops integration for large step sizes (as did method (T)) too late on average. We speculate, that this is due to the fact that we always test only for an excursion across the *closest* boundary, even in situations when two boundary segments are close. This is for example the case when y and z are close to a corner.

4.1.2 Unit n-spheres

We next show simulations for unit n-spheres (unit $n-$ball), i.e. we set $D = B_1^n = \{x \in \mathbf{R}^n : |x| < 1\}$. The *closest* point on ∂D was chosen as follows (we denote it by X_b):

$z \in \partial D$: We set $X_b = z$.

$z \in D$: We set X_b to be the minimizer of the sum of the square of the distances to the boundary. For the unit n-sphere this yields $X_b \leftarrow (y+z)/|y+z|$ – which is computationally very efficient and produces for h small enough adequate choices.

$z \notin D \cup \partial D$: We set X_b to the point where the line connecting y and z crosses ∂D.

We show results obtained with method (S) when applied to the exit problem in various dimensions. Results for dimensions $n = 2, 4, 8, 16$ are in Figure 2, those for $n = 64, 128$ in Figure 3. Sample size was always $N = 4\text{e}6$ and confidence intervals were always smaller than the plot symbols.

From the results in Figures 2 and 2 we see, that for the exit problem in unit n-spheres, aymptotically a convergence order of one can always be observed. This is encouraging, as it shows that the local approximation of ∂D by a tangent hyper plane seems to work, independent of the dimension. Note, however, that for increasing dimension n, a sufficiently small step size h is needed to get clear evidence of first order convergence behavior.

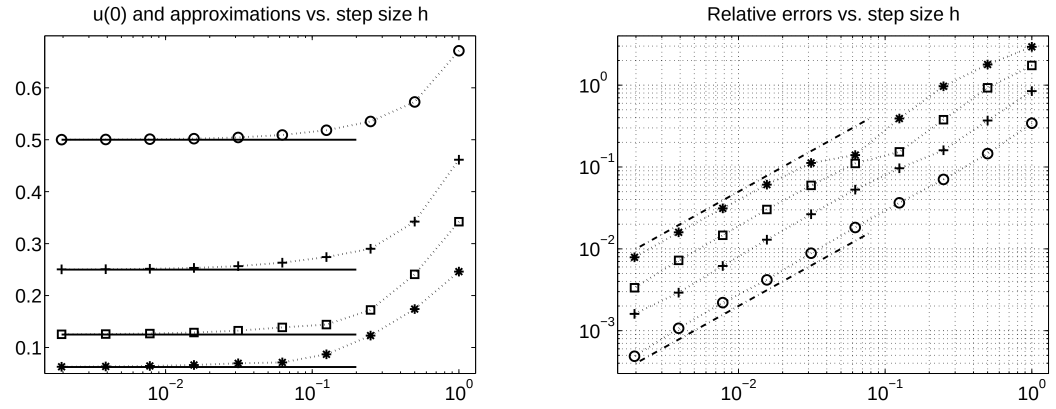

Figure 2: Results for exit problem in unit n-spheres. On the left, approximate solutions are shown versus step size h together with corresponding exact solutions (which are indicated by a solid line). On the right corresponding relative errors are shown versus step size h. Two dashed lines with slope one are added. In both plots $\cdots \circ \cdots$ is $n = 2$, $\cdots + \cdots$ is $n = 4$, $\cdots \square \cdots$ is $n = 8$ and $\cdots * \cdots$ is $n = 16$.

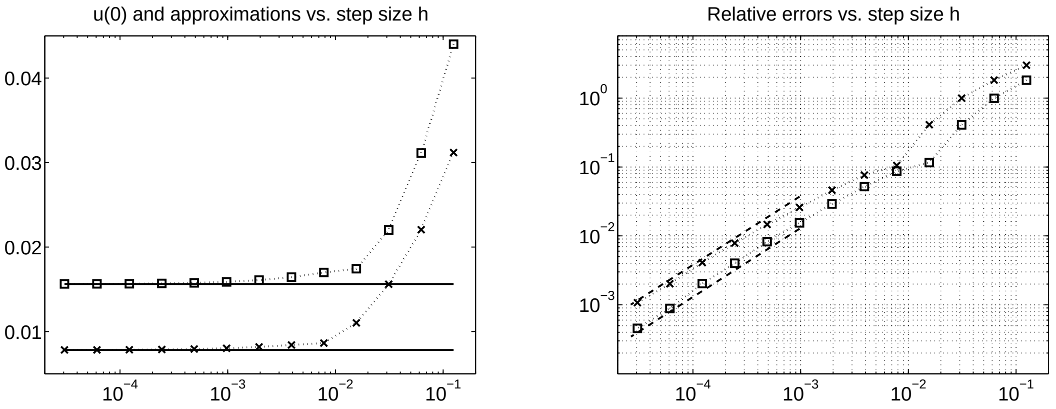

Figure 3: Results for exit problem in unit n-spheres.On the left, approximate solutions are shown versus step size h together with corresponding exact solutions (which are indicated by a solid line). On the right corresponding relative errors are shown versus step size h. Two dashed lines with slope one are. In both plots $\cdots \square \cdots$ is $n = 64$ and $\cdots \times \cdots$ is $n = 128$.

4.2 Non-constant boundary condition: Experiments in $n = 32$ dimensions

We show results from simulations in unit 32-spheres, see §4.1.2 for definitions and details concerning n-spheres. In particular, we look at three different problems which we describe next. The discussion of the individual results is postponed to §4.2.4.

4.2.1 Constant inhomogenety, linear boundary condition

We consider the problem

$$g(x) \equiv 1 \quad \text{and} \quad \psi(x) = \sum_{i=1}^{32} x^i \implies u(x) = \frac{1}{32}\left(1 - |x|^2\right) + \sum_{i=1}^{32} x^i. \tag{19}$$

For this problem, integration is still exact (as again $g(x) \equiv 1$), but in contrast to the *exit problem*, an approximation for $X(\tau)$ is required in order to evaluate the boundary condition for every path as soon as $x \neq (0, \ldots, 0)^T$. The boundary condition does not vary significantly and is linear.

For this problem we show results for $u(x)$ with $x = (\alpha/32)_{i=1,\ldots,32}$ with $\alpha = 0, 1, 2, 4$. Note that for higher α, x is closer to the boundary ∂D and therefore the probability that a point $X_b \in \partial D$ is the first exit point of a path starting at x is no longer uniformly distributed on ∂D (as it is the case for $x = (0, \ldots, 0)^T$). To better distinguish the individual results for different values of α we show plots of the absolute error, see Figure 4.

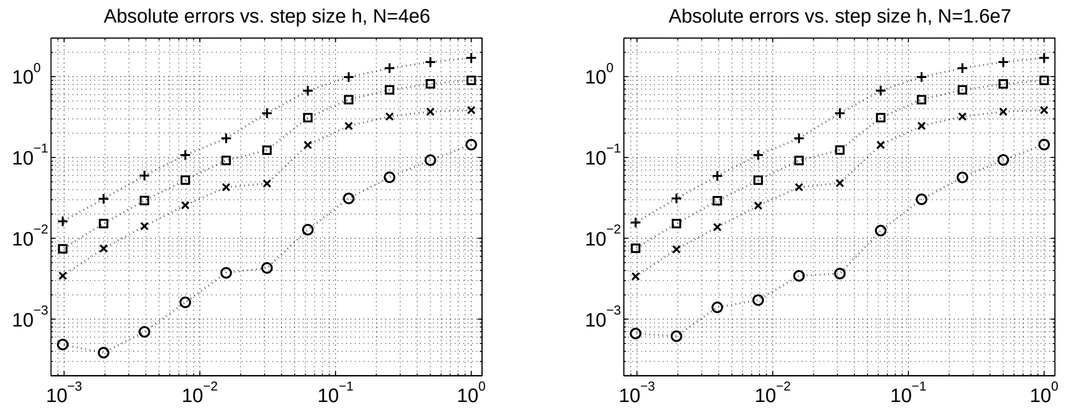

Figure 4: Results for test problem (19). Absolute errors vs. step size h for two different sample sizes N = 4e6 on the left and N = 1.6e7 on the right. Solution is evaluated at $x = (x^i)$ with $x^i = \alpha/32$ for $i = 1, \ldots, 32$. $\cdots \circ \cdots$ is $\alpha = 0$, $\cdots \times \cdots$ is $\alpha = 1$, $\cdots \square \cdots$ is $\alpha = 2$ and $\cdots + \cdots$ is $\alpha = 4$.

4.2.2 Quadratic inhomogenety, quartic boundary condition

We make $g(x)$ harder to integrate and choose a non-linear boundary condition. We show results of two variants of this problem:

$$g(x) = -\sum_{i=1}^{32} \left(x^i\right)^2 \quad \text{and} \quad \psi(x) = \frac{1}{6} \sum_{i=1}^{32} \left(x^i\right)^4 \implies u(x) = \psi(x) \tag{20a}$$

This problem has still an intrinsic symmetry: $g(\cdot)$ is constant on the spheres $|x| \equiv \|x\|_2 = r = \mathit{const}$ and $\psi(\cdot)$ is constant when $\|x\|_4^4 \equiv \sum_i (x^i)^4 = R = \mathit{const}$ respectively. We therefore consider the following extension:

$$g(x) = -\sum_{i=1}^{32} i \left(x^i\right)^2 \quad \text{and} \quad \psi(x) = \frac{1}{6} \sum_{i=1}^{32} i \left(x^i\right)^4 \implies u(x) = \psi(x) \tag{20b}$$

We show relative errors obtained versus step size h in Figure 5.

4.2.3 General (transcendental) inhomogenety and boundary condition

The last example we use to test Algorithm 3.1 is defined as follows: We set for $k^i = 1, 2, \ldots,$ with $k = (k^i)_{i=1,\ldots,32}$

$$g(x) = \prod_{i=1}^{32} \cos(2\pi k^i x^i) \quad \text{and} \quad \psi(x) = \frac{g(x)}{2\pi^2 |k|^2}. \tag{21}$$

Then $u(x) = \psi(x)$, $x \in D \cup \overline{D}$.

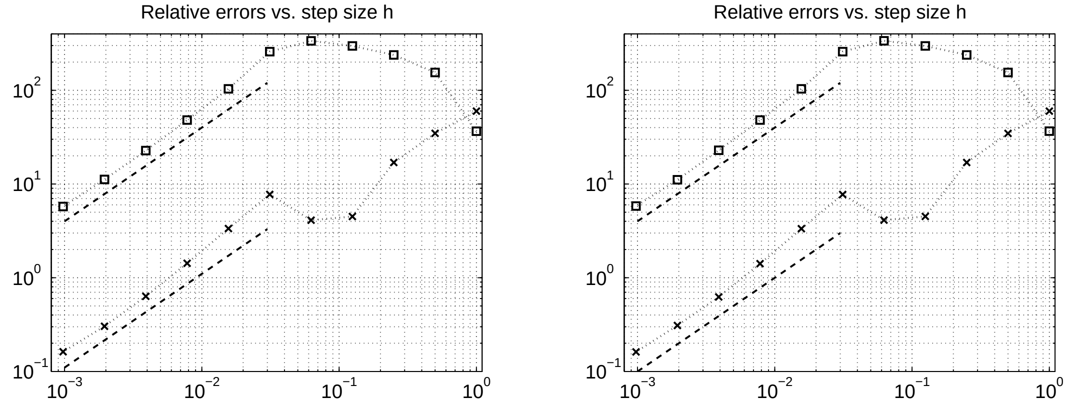

Figure 5: Results for test problems (20): The plot on the left shows results for test problem (20a) whilst on the right results for problem (20b) are shown. Relative errors vs. step size h for the sample size $N = 4e6$. Solution is evaluated at $x = (x^i)$ with $x^i = 5/100$ and $x^i = 1/10$ for $i = 1, \ldots, 32$. $\cdots\square\cdots$ is $x^i = 5/100$ and $\cdots\times\cdots$ is $x^i = 1/10$. In both plots, two dashed lines with slope one are added.

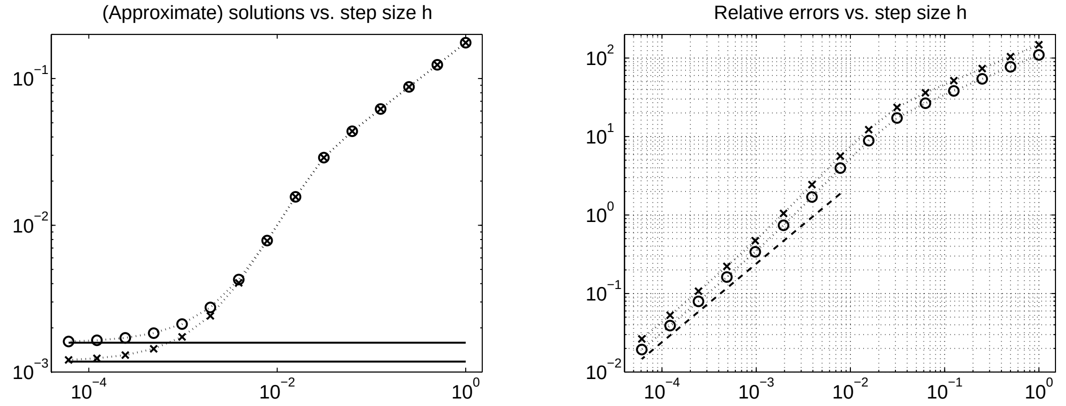

Figure 6: Results for test problem (21) when solution is evaluated at $x = (x^i)$ with $x^i = 0$ for $i = 1, \ldots, 32$ for $N = 1e6$ paths. On the left hand we show the approximate solutions versus step size h and indicate the exact solution with a solid line. On the right hand side, we plot corresponding errors versus step size h and add a dashed line with slope one. In both plots, $\cdots\circ\cdots$ are the results obtained for $k = (1, \ldots, 1)$ whilst $\cdots\times\cdots$ are those for $k = (3, 2, 1, 1, \ldots, 1)$.

4.2.4 Discussion of the numerical results for 32-dimensional spheres

Problem (19) results show that there is some absolute error increase as the initial point moves out from the center. This is not surprising since the solution (19) increases approximately linearly as the $|x| = r =$ initial radius increases. Thus the relative error is approximately constant in this range. Plots in Fig. 5 of the relative error for problems (20a) and (20b) confirm this assessment. Additionally, the expected $\mathcal{O}(h)$ accuracy of our Euler method is evident.

5 Conclusions and final remarks

We have outlined a general procedure for estimating exit times and exit points for Monte-Carlo solutions of elliptic partial differential equations in $n-$dimensional spaces. Since the Feynman-Kac

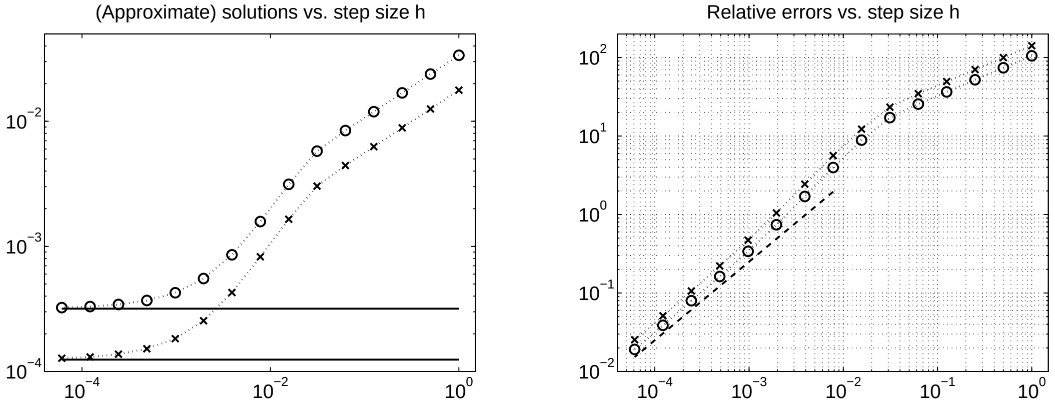

Figure 7: Results for test problem (21) when solution is evaluated at $x = (x^i)$ with $x^i = 5/100$ for $i = 1, \ldots, 32$ for $N = 1e6$ paths. See the caption of Figure 6 for further information.

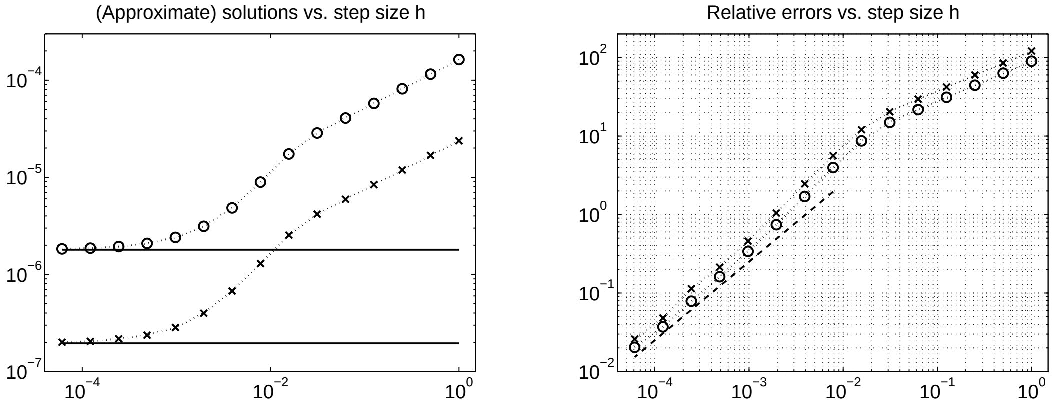

Figure 8: Results for test problem (21) when solution is evaluated at $x = (x^i)$ with $x^i = 1/10$ for $i = 1, \ldots, 32$ for $N = 1e6$ paths. See the caption of Figure 6 for further information.

representation requires good estimates for these statistics, an accurate sampling method seems essential. In our procedure, two situations are evident:

First, both $X_k \in D$ and $X_{k+1} \in D$, but possible excursions are estimated as in (9); and

second, $X_k \in D$ but $X_{k+1} \notin D$ the sample path has clearly exited domain D so an estimate of τ^D, $t_k < \tau^D < t_{k+1}$, is required.

In both cases, $X(\tau^D)$ must be sampled if an excursion has been determined in the first case, and always in the second case (definite exit). For this exit point, a bridge process (13) is used, via a transformation to a local tangent hyperplane coordinate system then pulled back to yield an exit point $X(\tau^D)$ in the original coordinates. These local hyperplane $\leftrightarrow$ original coordinate transformations can be done with Givens rotations (16) or Householder reflections (18) in $\mathcal{O}(n)$ arithmetic operations.

What remains to do, particularly for problems similar to our test in section 4.1.1, is a better analysis of corners or cusps. A subject of future work, but data in Figure 1 show encouraging results by a crude procedure which simply chooses the nearest side, at least for near-center values of x.

References

[1] F. M. Buchmann, Simulation of stopped diffusions, Tech. Rep. 2004-03, Seminar for Applied Mathematics, Swiss Federal Institute of Technology, CH-8092 Zürich, Switzerland, in press in Journal of Computational Physics (2004).

[2] M. Freidlin, Functional integration and partial differential equations, Vol. 109 of Annals of Mathematics Studies, Princeton University Press, Princeton, NJ, 1985.

[3] J. L. Folks, R. S. Chhikara, The inverse Gaussian distribution and its statistical application—a review, J. Roy. Statist. Soc. Ser. B 40 (3) (1978) 263–289, with discussion.

[4] F. M. Buchmann, Computing exit times with the Euler scheme, Tech. Rep. 2003-02, Seminar for Applied Mathematics, Swiss Federal Institute of Technology, CH-8092 Zürich, Switzerland, also available in ETH e-collection: http://e-collection.ethbib.ethz.ch/show?type=incoll&nr=877 (2003).

[5] F. M. Buchmann, Solving high dimensional Dirichlet problems numerically using the Feynman-Kac representation, Ph.D. thesis, Swiss Federal Institute of Technology (2004).

[6] P. E. Kloeden, E. Platen, Numerical solution of stochastic differential equations, Vol. 23 of Applications of Mathematics (New York), Springer-Verlag, Berlin, 1992.

[7] G. N. Milstein, Numerical integration of stochastic differential equations, Vol. 313 of Mathematics and its Applications, Kluwer Academic Publishers Group, Dordrecht, 1995, translated and revised from the 1988 Russian original.

[8] P. Baldi, Exact asymptotics for the probability of exit from a domain and applications to simulation, Ann. Probab. 23 (4) (1995) 1644–1670.

[9] E. Gobet, Weak approximation of killed diffusion using Euler schemes, Stochastic Process. Appl. 87 (2) (2000) 167–197.

[10] N. Madras, Lectures on Monte Carlo methods, Vol. 16 of Fields Institute Monographs, American Mathematical Society, Providence, RI, 2002.

[11] H. R. Lerche, D. Siegmund, Approximate exit probabilities for a Brownian bridge on a short time interval, and applications, Adv. in Appl. Probab. 21 (1) (1989) 1–19.

[12] J. R. Michael, W. R. Schucany, R. W. Haas, Generating random variates using transformations with multiple roots., Am. Stat. 30 (1976) 88–90.

[13] G. H. Golub, C. F. Van Loan, Matrix computations, Vol. 3 of Johns Hopkins Series in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 1983.

[14] J. W. Demmel, Applied numerical linear algebra, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.