Monte Carlo Methods Appl. Vol. 15 No. 1 (2009), pp. 11–47
DOI 10.1515 / MCMA.2009.002

On Monte Carlo algorithms applied to Dirichlet problems for parabolic operators in the setting of time-dependent domains

Kaj Nyström and Thomas Önskog

Abstract. Dirichlet problems for second order parabolic operators in space-time domains $\Omega \subset \mathbb{R}^{n+1}$ are of paramount importance in analysis, partial differential equations and applied mathematics. These problems can be approached in many different ways using techniques from partial differential equations, potential theory, stochastic differential equations, stopped diffusions and Monte Carlo methods. The performance of any technique depends on the structural assumptions on the operator, the geometry and smoothness properties of the space-time domain Ω, the smoothness of the Dirichlet data and the smoothness of the coefficients of the operator under consideration. In this paper, which mainly is of numerical nature, we attempt to further understand how Monte Carlo methods based on the numerical integration of stochastic differential equations perform when applied to Dirichlet problems for uniformly elliptic second order parabolic operators and how their performance vary as the smoothness of the boundary, Dirichlet data and coefficients change from smooth to non-smooth. Our analysis is set in the genuinely parabolic setting of time-dependent domains, which in itself adds interesting features previously only modestly discussed in the literature. The methods evaluated and discussed include elaborations on the non-adaptive method proposed by Gobet [4] based on approximation by half spaces and exit probabilities and the adaptive method proposed in [3] for weak approximation of stochastic differential equations.

Keywords. Time-dependent domain, non-smooth domain, heat equation, parabolic partial differential equations, Cauchy–Dirichlet problem, stochastic differential equations, stopped diffusion, Euler scheme, adaptive methods.

AMS classification. 65C05.

1. Introduction

This paper is devoted to the Dirichlet problem for operators of the form $\partial_t + L$, where

$$L = \sum_{i,j=1}^{n} a_{ij}(t,x)\partial^2_{x_i x_j} + \sum_{i=1}^{n} \mu_i(t,x)\partial_{x_i}, \tag{1.1}$$

$(t,x) \in \mathbb{R}^{n+1}$, $n \geq 1$, and the functions $\{a_{ij}(\cdot,\cdot)\}$ and $\{\mu_i(\cdot,\cdot)\}$ are bounded and continuous. Moreover, we assume that $a_{ij} = a_{ji}$, for $i,j = 1,\ldots,n$, and that there

exists a positive constant λ such that

$$\lambda^{-1}|\xi|^2 \leq \sum_{i,j=1}^{n} a_{ij}(t,x)\xi_i\xi_j \leq \lambda|\xi|^2, \qquad \xi \in \mathbb{R}^n,\ (t,x) \in \mathbb{R}^{n+1}.$$

Let $\Omega \subset \mathbb{R}^{n+1}$ be an open subset, let $\partial'_P\Omega$ denote the adjoint parabolic boundary of Ω and let $g : \overline{\Omega} \to \mathbb{R}$ be bounded and continuous on $\overline{\Omega}$. We consider the following problem for the operator $\partial_t + L$

$$\begin{cases} \partial_t u(t,x) + Lu(t,x) = 0, & \text{in } \Omega, \\ u(t,x) = g(t,x), & \text{on } \partial'_P\Omega. \end{cases} \tag{1.2}$$

Further smoothness assumptions on a_{ij}, μ_i, and g will be stated and discussed below. The problem in (1.2) represents the Dirichlet problem for the operator $\partial_t + L$ with boundary data g. To define the structure and smoothness imposed on Ω we consider graph domains

$$\Omega_\psi = \{(t,x) = (t,x',x_n) \in \mathbb{R} \times \mathbb{R}^{n-1} \times \mathbb{R} :\ x_n > \psi(t,x')\}, \tag{1.3}$$

where the defining function $\psi(t,x') : \mathbb{R}^n \to \mathbb{R}$ has compact support and $n \geq 1$. Given ψ, $0 \leq \epsilon \ll 1$ and $0 < T < \infty$ we let

$$\Omega_{\psi,\epsilon,T} = \{(t,x',x_n) : 0 < t < T,\ x' \in \mathbb{R}^{n-1},\ \epsilon + \psi(t,x') < x_n < \infty\}, \tag{1.4}$$

$$S_{\psi,\epsilon,T} = \{(t,x',x_n) : 0 < t < T,\ x' \in \mathbb{R}^{n-1},\ x_n = \epsilon + \psi(t,x')\}, \tag{1.5}$$

$$B_{\psi,\epsilon,T} = \{(t,x',x_n) : t = T,\ x' \in \mathbb{R}^{n-1},\ \epsilon + \psi(T,x') < x_n < \infty\}. \tag{1.6}$$

Then $\partial'_P\Omega_{\psi,\epsilon,T} = S_{\psi,\epsilon,T} \cup B_{\psi,\epsilon,T}$. In this paper we consider, for different operators L, defining functions ψ and pairs (ϵ, T), the following special case of the problem in (1.2):

$$\begin{cases} \partial_t u(t,x) + Lu(t,x) = 0, & \text{in } \Omega_{\psi,\epsilon,T}, \\ u(t,x) = g(t,x), & \text{on } S_{\psi,\epsilon,T}, \\ u(t,x) = 0, & \text{on } B_{\psi,\epsilon,T}. \end{cases} \tag{1.7}$$

The problem in (1.7) is the Dirichlet problem for L in $\Omega_{\psi,\epsilon,T}$ with terminal data zero and boundary data g on the lateral boundary. We will write $\mathcal{P}(L,g,\psi,\epsilon,T)$ as a short notation for the problem in (1.7). Using this notation we note that formally $\mathcal{P}(L,g,\psi,0,\infty)$ equals the Dirichlet problem for the operator L in the domain Ω_ψ introduced in (1.3).

The purpose of this paper, the paper being mainly of numerical and experimental character, is to understand the performance of several Monte Carlo algorithms for the problem $\mathcal{P}(L,g,\psi,\epsilon,T)$ for different choices of (L,g,ψ). We put a particular emphasis on the underlying geometry and note that the success of numerical methods

designed to resolve the problem in (1.7) depends on the dimension of the underlying space, $n+1$, and the smoothness of L, g and the defining function ψ. Concerning ψ, we consider the case of smooth ψ as well as the case of non-smooth ψ. In particular, we say that

$$\begin{aligned}&\psi \text{ is smooth if } \psi \text{ is five times continuously differentiable}\\&\text{with respect to the time- and space-variables}\end{aligned} \tag{1.8}$$

and that

$$\begin{aligned}&\psi \text{ is non-smooth if } \psi \text{ defines a time-dependent domain } \Omega_\psi\\&\text{in the sense of Definition 2.1 stated below.}\end{aligned} \tag{1.9}$$

In (1.8) we could assume, essentially for simplicity, that ψ is infinitely differentiable but we here choose to call ψ smooth if its smoothness is in line, as will be clarified further below, with the smoothness assumptions imposed by Gobet [4]. To briefly discuss the notion of non-smooth ψ defined in (1.9) we note that the study of boundary value problems for elliptic partial differential equations in Lipschitz domains is by now, from a theoretical as well as from a numerical perspective, a well established area of research, see [6]. In particular, unbounded and bounded Lipschitz domains appear naturally in many applications and by modern results in harmonic analysis, harmonic measures and singular integrals associated to linear second order elliptic operators can be controlled in this geometric setting, see [6]. The corresponding analysis for parabolic problems in the genuinely parabolic setting of time-dependent domains is less developed. However, recent years have seen several breakthroughs in the study of parabolic problems in non-smooth and time-dependent domains. In particular, through the works of Lewis and Murray [8] and Hofmann and Lewis [5], it has become clear that from the perspective of caloric measure and parabolic singular integrals, the parabolic analogue of the notion of unbounded Lipschitz domains is the notion of graph domains, see (1.3), whose defining function ψ is non-smooth in the sense of (1.9). In particular, in [5], [8] the Dirichlet problem $\mathcal{P}(\Delta, g, \psi, 0, \infty)$, where Δ denotes the Laplace operator, was resolved using parabolic layer potentials. Furthermore, (1.9) implies, as discussed in Section 2, that the smoothness of ψ is only slightly better than Lip(1,1/2), i.e. uniformly Lipschitz in the space variables and uniformly Hölder of order 1/2 in the time variable. We emphasize that the domains introduced in [5], [8] are from many perspective the correct replacement, for parabolic problems, of the notion of Lipschitz domains successfully used in the study of elliptic partial differential equations.

The essence of this paper is that we evaluate the performance of several Monte Carlo algorithms for the problem $\mathcal{P}(L, g, \psi, \epsilon, T)$ in the setting of smooth time-dependent domains as well as in the setting of the non-smooth and genuinely parabolic time-dependent Lip(1,1/2)-type domains briefly discussed above. In particular, we consider

the following specific problems

$$\begin{aligned}
&\text{(P1)} \quad \mathcal{P}(\Delta, g, \psi, 0, T),\ \psi \text{ as in (1.8)},\\
&\text{(P2)} \quad \mathcal{P}(\Delta, g, \psi, 0, T),\ \psi \text{ as in (1.9)},\\
&\text{(P3)} \quad \mathcal{P}(L_1, g, 0, 0, T),\ \psi \text{ as in (1.8) and } L_1 \text{ defined based on } \psi,\\
&\text{(P4)} \quad \mathcal{P}(L_2, g, 0, \epsilon, T),\ \psi \text{ as in (1.9) and } L_2 \text{ defined based on } \psi,
\end{aligned} \tag{1.10}$$

where $0 \leq \epsilon \ll 1$ and $0 < T < \infty$. The operators L_1 and L_2 appearing in problem (P3) and (P4) are defined below. We note that in (P1) and (P2) we consider the simplest operator and allow for a considerable degree of freedom in the specification of the underlying geometry. In contrast, in (P3) and (P4) we consider the simplest geometry, but allow for operators which may have non-smooth coefficients.

To specify the operators L_1 and L_2 we note that one way to handle a complicated domain which is locally given as a graph domain is to transform the problem to the upper half space by appropriate pull-backs. Moreover, the definition of such pull-backs depends on the smoothness of the defining function ψ and is less obvious in the non-smooth setting of (1.9) compared to the more classical case of smooth domains (1.8). In particular, assuming (1.8) we simply let

$$\rho(t, x', x_n) = (t, x', x_n + \psi(t, x')), \quad \rho(t, x', 0) = (t, x', \psi(t, x')). \tag{1.11}$$

Then $\rho(t, x) = \rho(t, x', x_n)$ is a natural 'parabolic' lifting from $\mathbb{R}^{n+1}_+ := \mathbb{R} \times \mathbb{R}^{n-1} \times \mathbb{R}_+$ onto Ω_ψ. Moreover, assuming (1.9) we, as outlined in the bulk of the paper, need to make use of a more involved 'parabolic' lifting $\rho(t, x', x_n) : \mathbb{R}^{n+1}_+ \to \Omega_\psi$. In particular, we let

$$\rho(t, x', x_n) = (t, x', x_n + P_{\gamma x_n}\psi(t, x')), \quad \rho(t, x', 0) = (t, x', \psi(t, x')). \tag{1.12}$$

Here $P(\cdot) \in C_0^\infty(\mathbb{R}^n)$ is a parabolic approximation of the identity defined in the bulk of the paper, γ is a small parameter and $P_{\gamma x_n}\psi(t, x')$ is defined as a convolution.

If u solves the problem $\mathcal{P}(\Delta, g, \psi, 0, \infty)$, with ψ as in (1.8), we define the pull-back of u from Ω_ψ to $\mathbb{R}^{n+1}_+$, denoted $v_1(t, x) = v_1(t, x', x_n)$, as

$$v_1(t, x', x_n) = u(t, x', x_n + \psi(t, x')). \tag{1.13}$$

Similarly, if u solves the problem $\mathcal{P}(\Delta, g, \psi, 0, \infty)$, with ψ as in (1.9), we define the pull-back of u from Ω_ψ to $\mathbb{R}^{n+1}_+$, denoted $v_2(t, x) = v_2(t, x', x_n)$, as

$$v_2(t, x', x_n) = u(t, x', x_n + P_{\gamma x_n}\psi(t, x')). \tag{1.14}$$

Then, formally v_1 and v_2 satisfy $(\partial_t + L_1)v_1 = 0$ and $(\partial_t + L_2)v_1 = 0$ respectively, for two operators L_1 and L_2 which can be written down explicitly in terms of derivatives of the pullback ρ. In fact, L_1 and L_2 are symmetric, uniformly elliptic operators and the

conditions on the time-dependent domains are transformed into appropriate conditions on the coefficients of the operators L_1 and L_2. In particular, in the case $n = 1$, then $\psi = \psi(t)$ as there is only one spatial direction, here denoted by x, and in this case the operator L_1, assuming (1.8), is given as

$$L_1 = \partial_{xx}^2 - (\partial_t \psi)\partial_x, \tag{1.15}$$

while the operator L_2, assuming (1.9), is given as

$$L_2 = D^{-2}\partial_{xx}^2 - (D^{-3}\partial_{xx}^2(P_{\gamma x}\psi) + D^{-1}\partial_t(P_{\gamma x}\psi))\partial_x, \tag{1.16}$$

where $D = 1 + \partial_x(P_{\gamma x}\psi)$ and $(t, x) \in \mathbb{R} \times \mathbb{R}_+$. In particular, if we assume (1.8), then the coefficient in front of the lower order term in the operator L_1 is bounded and four times continuously differentiable. Moreover, assuming (1.9), we are able to adjust γ so that $D(t, x) \in [1/2, 3/2]$ for all $(t, x) \in \mathbb{R} \times \mathbb{R}_+$, see (2.4). However, as outlined in (2.5), in this case the lower order coefficient may explode like x^{-2} as $x \to 0^+$ and the same is true for derivatives of the diffusion coefficient. Hence, the operator L_2 might have coefficients which are smooth but explode as we approach the boundary.

As we in this paper approach (P1)–(P4) using Monte Carlo algorithms, a stochastic formalism has to be developed. To do this we note that formally the unique solution to each of the problems in (P1)–(P4), at $(t, x) \in \Omega_{\psi,\epsilon,T}$, $\epsilon = 0$ in case of (P1)–(P3) and $\psi \equiv 0$ in case of (P3)–(P4), has the probabilistic representation

$$u(t, x) = E\Big[g(\tau, X(\tau))\Big|X(t) = x\Big], \tag{1.17}$$

where $X(t) = (X_1(t), \ldots, X_n(t))^*$,

$$X_i(t) = x_i + \int_0^t \mu_i(s, X(s))\, ds + \sum_{j=1}^n \int_0^t \sigma_{ij}(s, X(s))\, dW_j(s), \tag{1.18}$$

for $i \in \{1, \ldots, n\}$ and for appropriate functions $\{\mu_i\}$ and $\{\sigma_{ij}\}$. $X(t)^*$ is the transpose of $X(t)$. Moreover,

$$\tau = \tau^{t,x} = \inf\{s \geq t : (s, X(s)) \notin \Omega_{\psi,\epsilon,T}\},$$

is the first exit time of the process $(s, X(s))$ from the domain $\Omega_{\psi,\epsilon,T}$. In the case of (P1)–(P2) we have $\mu_i \equiv 0$ for $i \in \{1, \ldots, n\}$ while $\sigma = \{\sigma_{ij}\} = \sqrt{2}I_n$ where I_n is the $n \times n$-identity matrix. In the case of (P3)–(P4) we note that L_1 and L_2 can both be written in the form (1.1) for appropriate matrices $A(t, x) = \{a_{ij}(t, x)\}$ and vectors $\mu(t, x) = (\mu_1(t, x), \ldots, \mu_n(t, x))$. As $A(t, x) = \{a_{ij}(t, x)\}$ is symmetric and uniformly elliptic for all $(t, x) \in \mathbb{R}_+^{n+1}$, there exists a matrix $\sigma(t, x) = \{\sigma_{ij}(t, x)\}$ such that

$$A(t, x) = \frac{1}{2}\sigma(t, x)(\sigma(t, x))^*, \quad \text{for all } (t, x) \in \mathbb{R}_+^{n+1},$$

where $(\sigma(t,x))^*$ is the transpose of $\sigma(t,x)$. Note that if $n=1$, then, in the case of (1.8), $\sigma=\sigma(t,x)=\sqrt{2}$ and, in the case of (1.9), $\sigma=\sigma(t,x)=\sqrt{2}D^{-1}$, where D was defined in (1.16). Finally, note that $(W(t))_{0\leq t\leq T}$, $W(t)=(W_1(t),\ldots,W_n(t))^*$, is a standard Brownian motion in $\mathbb{R}^n$ defined on a filtered probability space $(\Omega,\mathcal{F},(\mathcal{F}_t)_{0\leq t\leq T},\mathbf{P})$ with the usual assumptions on $(\mathcal{F}_t)_{0\leq t\leq T}$. By a standard Brownian motion in $\mathbb{R}^n$ we mean a process whose components are independent one-dimensional Brownian motions.

In this paper we approach problems (P1)–(P4) using the representation in (1.17) and hence the rest of the paper is devoted to Monte Carlo algorithms for stopped diffusion in the geometric setting outlined above. We note that any rigorous proof of the order of convergence of non-adaptive or adaptive Monte Carlo algorithms for the problem in (1.7) requires a high level of smoothness on (L,g,ψ), a level of smoothness which cannot always be met in reality. Based on this, the objective of this paper is to investigate, numerically and experimentally, how different Monte Carlo algorithms for stopped diffusion perform when applied to problems (P1)–(P4).

Accepting (1.17) and (1.18), we note that the difficulty in any Monte Carlo algorithm for the approximation of the process $(t,X(t))$, with $X(t)$ being defined in (1.18), stopped on the boundary of the domain is that a continuous sample path may exit the domain even though the discrete solution does not cross the boundary. If we consider an Euler scheme with N time steps, the problem of resolving the hitting of the boundary results in a time discretization error of order $N^{-1/2}$ for the Monte Carlo Euler scheme with N uniform time steps, see [1], while the time discretization error is of order N^{-1} in the unbounded domain $\mathbb{R}^n\times[0,T]$, see [7]. Still, the large first exit error can be reduced from order $N^{-1/2}$ to order N^{-1} by using refined estimates of exit probabilities. In particular, on domains with smooth boundary, an estimate of the exit probability was used by Gobet [4] to derive an algorithm with time discretization error of order N^{-1} for a problem similar to (P1), but with prescribed terminal data and $g=0$. Moreover, the first set of algorithms evaluated in this paper is based on the ideas of Gobet [4].

An alternative approach is the algorithm proposed by Dzougoutov, Moon, von Schwerin, Szepessy and Tempone [3], where the computational error is reduced by adaptively choosing the size of the time steps near the boundary. This approach has the advantage that the exit probability does not have to be computed accurately and with care. In particular, in [3] an error estimate in a posteriori form is derived and an algorithm similar to the adaptive algorithm outlined in [11] is constructed. The article [3] contains numerical justification for the claim that the time discretization error, with N adaptive time steps, is of order N^{-1} also in the case of stopped diffusions, but there is no rigorous proof of this statement. Finally, we note that Section 4.2 in [3] is devoted to the performance of the adaptive algorithm in the simple setting of a domain in the plane with a corner on the boundary. While the situation at hand has too little smoothness compared to what is needed to rigorously derive the error expansions and

the algorithm, the order of convergence is better than N^{-1}. In fact, the numerical simulation indicate that the order of convergence may even be exponential. In particular, the paper at hand has been inspired by this example and the second set of algorithms evaluated in this paper are based on the algorithm proposed in [3].

The rest of the paper is organized as follows. In Section 2 we discuss the geometry of time-dependent domains and the associated pull-backs and introduce the Euler scheme. In Section 3 we describe the non-adaptive algorithm of [4] and we develop appropriate error expansions based on which we describe the adaptive algorithm for stopped diffusions proposed in [3]. We evaluate the performance of these algorithms in the context of problems (P1)–(P4) in Section 4 and end the paper with a section devoted to conclusions and discussion.

2. Preliminaries

In this section, which partially is of preliminary nature, we discuss the geometry of time-dependent domains and the associated pull-backs and introduce the Euler scheme.

2.1. Time-dependent domains and pull-backs

Recall that given a defining function $\psi : \mathbb{R}^n \to \mathbb{R}$ with compact support, $n \geq 1$, the graph domain Ω_ψ was introduced in (1.3). In the following we define the appropriate pull-backs discussed in the introduction and we discuss, in particular, the notion of non-smooth ψ left open in (1.9). In particular, let v_1 and v_2 be as in (1.13) and (1.14) respectively and let L_1 and L_2 be the associated operators. Then

$$L_k = \sum_{i,j=1}^{n} a_{ij}^k(t,x)\partial^2_{x_i x_j} + \sum_{i=1}^{n} \mu_i^k(t,x)\partial_{x_i}, \tag{2.1}$$

for $k \in \{1,2\}$ and in the following we specify a_{ij}^k and μ_i^k.

2.1.1. ψ is smooth

A straightforward calculation shows that $a_{ij}^1 = a_{ij}$ and $\mu_i^1 = \mu_i$ with

$$\begin{aligned} a_{ij}\left(t,x',x_n\right) &= \delta_{ij}, \quad \text{for } i,j \in \{1,2,\ldots,n-1\}, \\ a_{in}\left(t,x',x_n\right) &= a_{ni}\left(t,x',x_n\right) = \partial_{x_i}\psi, \quad \text{for } i \in \{1,2,\ldots,n-1\}, \\ a_{nn}\left(t,x',x_n\right) &= 1 + \left|\nabla_{x'}\psi\right|^2, \end{aligned} \tag{2.2}$$

where δ_{ij} is the Kronecker delta, and

$$\mu_i(t, x', x_n) = 0, \quad \text{for } i \in \{1, 2, \dots, n-1\},$$
$$\mu_n(t, x', x_n) = -\partial_t \psi - \sum_{k=1}^{n-1} \partial^2_{x_k x_k} \psi. \tag{2.3}$$

2.1.2. ψ is non-smooth

Let $z = (t, x') \in \mathbb{R} \times \mathbb{R}^{n-1}$ and let $\|z\|$ be the unique positive solution ρ to the equation

$$\frac{t^2}{\rho^4} + \sum_{i=1}^{n-1} \frac{x_i^2}{\rho^2} = 1.$$

Note that $\|(\delta^2 t, \delta x')\| = \delta \|(t, x')\|$ and we call $\|z\|$ the parabolic norm of z. We define parabolic *BMO* as the space of locally integrable functions modulo constants satisfying

$$\|b\|_* := \sup_B \frac{1}{|B|} \int_B |b(z) - m_B b| \, dz < \infty.$$

Here $z = (t, x')$, B denotes the parabolic ball $B = \{z \in \mathbb{R}^n : \|z - z_0\| < r\}$ and $m_B b$ denotes the average of the function b over the ball B. Let $\widehat{f}(\xi, \tau)$ be the Fourier transform of a function f defined on $\mathbb{R}^n$, and let ξ, τ denote the phase variables. For a function $g \in C_0^\infty(\mathbb{R})$ we introduce the fractional differentiation operator $D_{1/2}$ by

$$\widehat{(D_{1/2} g)}(\tau) := |\tau|^{1/2} \hat{g}(\tau).$$

If $h \in C_0^\infty(\mathbb{R}^n)$ then by $D^t_{1/2} h : \mathbb{R}^n \to \mathbb{R}$ we will mean $D_{1/2} h(\cdot, x')$ defined a.e. for each $x' \in \mathbb{R}^{n-1}$.

Definition 2.1. A time-dependent domain with defining function ψ and parameters a_1 and a_2, denoted TD(ψ, a_1, a_2), is a domain $\Omega \subset \mathbb{R}^{n+1}$ of the form introduced in (1.3) where the function $\psi : \mathbb{R}^n \to \mathbb{R}$ has compact support and satisfies

(i) $|\psi(t, x') - \psi(t, y')| \leq a_1 |x' - y'|, \quad \text{for } t \in \mathbb{R} \text{ and } x', y' \in \mathbb{R}^{n-1},$

(ii) $D^t_{1/2} \psi \in \text{BMO}(\mathbb{R}^n), \ \|D^t_{1/2} \psi\|_* \leq a_2.$

Remark 2.2. One can prove that if Ω is TD(ψ, a_1, a_2) for some a_1, a_2 then

$$|\psi(t, x') - \psi(s, y')| \leq \delta_0 (|t - s|^{1/2} + |x' - y'|), \quad \text{for } t, s \in \mathbb{R} \text{ and } x', y' \in \mathbb{R}^{n-1},$$

for some $\delta_0 = \delta_0(a_1, a_2)$, see [5].

Remark 2.3. The geometric conditions stated in Definition 2.1 can also be expressed in a different but equivalent form. Define a parabolic half-order time derivative by

$$\widehat{\mathbf{D}_n\psi}(\tau,\xi) := \frac{\tau}{\|(\tau,\xi)\|}\hat{\psi}(\tau,\xi),$$

let $\|\cdot\|_\infty$ be the supremum norm and define $\|\psi\|_{\text{comm}} = \|\nabla_x\psi\|_\infty + \|\mathbf{D}_n\psi\|_*$, where $\|\nabla_x\psi\|_\infty := \sup_t \|\nabla_{x'}\psi(t,\cdot)\|_\infty$. The condition $\|\psi\|_{\text{comm}} < \infty$ is equivalent to the statement that Ω is TD(ψ, a_1, a_2) for some a_1, a_2, see [5].

Let $P(z) \in C_0^\infty(\mathbb{R}^n)$, $z = (t, x')$, be a non-negative even function satisfying $\int_{\mathbb{R}^n} P(z)dz = 1$, so that $P(z)$ is a parabolic approximation of the identity. Let $d = n + 1$ and define

$$P_\lambda(z) = \lambda^{-d}P(\lambda^{-\alpha}z) = \lambda^{-d}P(\lambda^{-2}t, \lambda^{-1}x').$$

For a locally integrable function f, we denote the convolution operator as $P_\lambda f$. We now define a 'parabolic' lifting $\rho = \rho(t, x', x_n) : \mathbb{R}^{n+1}_+ \to \Omega_\psi$ as in (1.12). In (1.12) γ is a small parameter and, as $\|\nabla_{x'}\psi\|_\infty < \infty$, we can adjust γ so that

$$\frac{1}{2} \le 1 + \partial_{x_n}(P_{\gamma x_n}\psi)(t, x') \le \frac{3}{2}, \quad \text{whenever } (t, x') \in \mathbb{R} \times \mathbb{R}^{n-1}. \tag{2.4}$$

Let σ, θ be non-negative integers and let $\phi = (\phi_1, \dots, \phi_{n-1})$ be a multi-index. Define $\ell = \theta + |\phi| + \sigma$. Assume that $\|\psi\|_{\text{comm}} \le \beta < \infty$ and let, in the following, $b = \|\mathbf{D}_n\psi\|_*$, if $\theta \ge 1$, and $b = 1$, if $\theta = 0$. On pages 365–366 in [5], it is proved that if $\ell \ge 1$ then

$$\left|\frac{\partial^\ell P_{\gamma x_n}\psi(t, x')}{\partial t^\theta \partial x^\phi \partial x_n^\sigma}\right| \le c_1\gamma^{1-|\phi|-2\theta}x_n^{1-\ell-\theta}b(1+\beta). \tag{2.5}$$

Based on (2.5), it is clear that partial derivatives of $P_{\gamma x_n}\psi(t, x')$ may very well blow-up as $x_n \to 0^+$.

We are now ready to derive the explicit expressions for $a_{ij}^2 = a_{ij}$ and $\mu_i^2 = \mu_i$. Let v_2 be as in (1.14) and let

$$D = 1 + \partial_{x_n}(P_{\gamma x_n}\psi).$$

We then first note, by differentiating (1.13), that

$$\partial_{x_i} v = \partial_{x_i} u + \partial_{x_n} u \partial_{x_i}\left(P_{\gamma x_n}\psi\right),$$

for $i \in \{1, 2, \dots, n-1\}$ and that

$$\partial_{x_n} v = D\partial_{x_n} u, \quad \partial_t v = \partial_t u + \partial_{x_n} u \partial_t\left(P_{\gamma x_n}\psi\right).$$

Similarly we can derive expressions for $\partial^2_{x_i x_i} v$, $\partial^2_{x_i x_n} v$ and $\partial^2_{x_n x_n} v$ in terms of first and second order derivatives of u. From these relations it is easy to express $\partial_t u + \Delta u$ in a form containing derivatives of v only. In fact, a rigorous calculation shows that

$$\begin{aligned}
&\mu_i(t, x', x_n) = 0, \quad \text{for } i \in \{1, 2, \dots, n-1\},\\
&\mu_n(t, x', x_n) = -D^{-3}\partial^2_{x_n x_n}(P_{\gamma x_n}\psi)(1 + |\nabla_{x'}(P_{\gamma x_n}\psi)|^2) - D^{-1}\partial_t(P_{\gamma x_n}\psi)\\
&\qquad\qquad + \sum_{k=1}^{n-1}\Big(2D^{-2}\partial_{x_k}(\partial P_{\gamma x_n}\psi)\partial^2_{x_k x_n}(P_{\gamma x_n}\psi) - D^{-1}\partial^2_{x_k x_k}(P_{\gamma x_n}\psi)\Big),
\end{aligned}$$

and

$$\begin{aligned}
&a_{ij}\left(t, x', x_n\right) = \delta_{ij}, \quad \text{for } i, j \in \{1, 2, \dots, n-1\},\\
&a_{in}\left(t, x', x_n\right) = a_{ni}\left(t, x', x_n\right) = D^{-1}\partial_{x_i}(P_{\gamma x_n}\psi), \quad \text{for } i \in \{1, 2, \dots, n-1\},\\
&a_{nn}\left(t, x', x_n\right) = D^{-2}\left(1 + |\nabla_{x'}(P_{\gamma x_n}\psi)|^2\right).
\end{aligned}$$

2.2. The Euler scheme

In this section we introduce the Euler scheme associated to the stochastic representation in (1.17) and the system of stochastic differential equations in (1.18). Given a time horizon of T we let $\{t_k\}_{k=0}^N$ define a partition Δ of the interval $[0, T]$, i.e., $0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = T$, and we let $\Delta t_k = t_{k+1} - t_k$ for $k \in \{0, 1, \dots, N-1\}$. Let $\{X(t), t \in [0, T]\}$ be the solution to (1.18). In the following we let $\{X^\Delta(t), t \in [0, T]\}$ denote the continuous Euler approximation of $\{X(t), t \in [0, T]\}$ defined as follows. $X^\Delta(t)$ satisfies, for $k \in \{0, 1, \dots, N-1\}$, the difference equation

$$\begin{aligned}
X_i^\Delta(t_{k+1}) &= X_i^\Delta(t_k) + \mu_i(t_k, X^\Delta(t_k))\,\Delta t_k + \sum_{j=1}^n \sigma_{ij}(t_k, X^\Delta(t_k))\,\Delta W_j(t_k),\\
X_i^\Delta(t_0) &= x_i,
\end{aligned} \tag{2.6}$$

where $\Delta W_j(t_k) = W_j(t_{k+1}) - W_j(t_k)$ represents a Wiener increment during the time step $[t_k, t_{k+1}]$. The set $\{X^\Delta(t), t \in \{t_0, \dots, t_N\}\}$ is often referred to as the associated discrete Euler approximation. Let the function $\phi(t) = \sup\{t_k : t_k \le t\}$ be defined whenever $t \in [0, T]$. With this notation the continuous Euler approximation, $\{X^\Delta(t), t \in [0, T]\}$, is given by the relation

$$\begin{aligned}
X_i^\Delta(t) = X_i^\Delta(\phi(t)) &+ \int_{\phi(t)}^t \mu_i(\phi(s), X^\Delta(\phi(s)))\,ds\\
&+ \sum_{j=1}^n \int_{\phi(t)}^t \sigma_{ij}(\phi(s), X^\Delta(\phi(s)))\,dW_j(s).
\end{aligned} \tag{2.7}$$

Given $0 < \epsilon \ll 1$ we let $\Omega_{\psi,\epsilon,T}$ be defined as in (1.4). We introduce the stopping times

$$\tau_c^\Delta = \tau_c^{\Delta,t,x} = \inf\{s > t : \ (s, X^\Delta(s)) \notin \Omega_{\psi,\epsilon,T} |\ X^\Delta(t) = x\},$$

for $t \in [0,T]$ and

$$\tau_d^\Delta = \tau_d^{\Delta,t,x} = \inf\{t_l > t : \ (t_l, X^\Delta(t_l)) \notin \Omega_{\psi,\epsilon,T} |\ X^\Delta(t) = x\},$$

whenever $t = t_k$, $k \in \{0, 1, \ldots, N-1\}$. τ_c^Δ and τ_d^Δ are the first exit times from the region $\Omega_{\psi,\epsilon,T}$ for the continuous and discrete Euler approximation respectively.

Let M be an integer and let $\{\omega_r\}_{r=1}^M$ denote M realizations of the discrete Euler approximation. Then the Euler approximation of (1.17), at $(0,x)$, is given by

$$u(0,x) = E[g(\tau, X(\tau))|X(0) = x] \approx \frac{1}{M} \sum_{r=1}^{M} g\left(\tau_d^\Delta(\omega_r), X^\Delta\left(\tau_d^\Delta(\omega_r)\right)\right).$$

Note that, strictly speaking, the function g is undefined at $\left(\tau_d^\Delta(\omega_r), X^\Delta\left(\tau_d^\Delta(\omega_r)\right)\right) \notin \Omega_{\psi,\varepsilon,T}$ and we shall use the interpretation

$$X^\Delta\left(\tau_d^\Delta(\omega_r)\right) := \pi_{\tau_d^\Delta(\omega_r)}\left(X^\Delta\left(\tau_d^\Delta(\omega_r)\right)\right),$$

where π_t is either of the projection operators onto $S_{\psi,\epsilon,T}$ defined in (3.5) or (3.8). Define the error function $\Delta^{\Delta,M}(x)$ as

$$\Delta^{\Delta,M}(x) := E[g(\tau, X(\tau))|X(0) = x] - \frac{1}{M} \sum_{r=1}^{M} g\left(\tau_d^\Delta(\omega_r), X^\Delta\left(\tau_d^\Delta(\omega_r)\right)\right).$$

The error function can be decomposed as

$$\Delta^{\Delta,M}(x) = \Delta_c^N(x) + \Delta_d^N(x) + \Delta_s^{\Delta,M}(x),$$

where

$$\Delta_c^N(x) = E[g(\tau, X(\tau))|X(0) = x] - E[g(\tau_c^\Delta, X^\Delta(\tau_c^\Delta))|X^\Delta(0) = x], \tag{2.8}$$

$$\Delta_d^N(x) = E[g(\tau_c^\Delta, X^\Delta(\tau_c^\Delta))|X^\Delta(0) = x] - E[g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))|X^\Delta(0) = x], \tag{2.9}$$

$$\begin{aligned}\Delta_s^{\Delta,M}(x) &= E[g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))|X^\Delta(0) = x] \\ &\quad - \frac{1}{M} \sum_{r=1}^{M} g\left(\tau_d^\Delta(\omega_r), X^\Delta\left(\tau_d^\Delta(\omega_r)\right)\right).\end{aligned} \tag{2.10}$$

Here $\Delta_c^N(x)$ represents the time-discretization error due to the fact that X^Δ and τ_c^Δ are only approximations of X and τ and $\Delta_d^N(x)$ represents the exit error due to the fact that τ_c^Δ and τ_d^Δ do not coincide. $\Delta_s^{\Delta,M}(x)$ is the statistical error.

3. Monte Carlo algorithms for weak approximations of stochastic differential equations

In this section we present two Monte Carlo approaches to the problem in (1.7) based on the representation formula in (1.17). The first approach, which is non-adaptive, is based on [4] and the second approach, which is adaptive, is based on [3].

3.1. A non-adaptive approach

The algorithm in [4] was constructed to approximate solutions to the parabolic Cauchy–Dirichlet problem

$$\begin{cases} \partial_t w(t,x) + Lw(t,x) = 0, & \text{if } (t,x) \in [0,T] \times \Omega, \\ w(t,x) = 0, & \text{if } (t,x) \in [0,T] \times \partial\Omega, \\ w(T,x) = h(x), & \text{if } x \in \Omega, \end{cases} \tag{3.1}$$

where L is as in (1.1) and $\Omega \subset \mathbb{R}^n$ is a bounded domain. In particular, $[0,T] \times \Omega$ is a time-independent domain. In [4] Ω is assumed to be $\mathcal{C}^5$-smooth which means that the defining function for Ω is assumed to be locally five times continuously differentiable. Moreover, the coefficients of L are assumed to be $\mathcal{C}^5$-functions and the function h is assumed to be a compactly supported $\mathcal{C}^5$-function. The solution to the problem in (3.1) has the probabilistic representation

$$w(t,x) = E\left[\chi_{[0,\tau]}(T) h\left(X(T)\right) | X(t) = x\right], \tag{3.2}$$

where X solves a system of stochastic differential equations of the form (1.18) and τ is the first exit time from Ω of the process X. The problem in (3.1) is known as killed diffusion as the functional in (3.2) is zero if X reaches the lateral boundary before time T.

The algorithm in [4] is based on the observation that the exit probability of a stochastic differential equation with constant coefficients can be calculated exactly in the case of a half space and that any sufficiently smooth domain can be approximated, locally, by a half space. We here briefly describe the algorithm. Let $\{t_k\}_{k=0}^N$ denote the uniform partition Δ of the interval $[t,T]$ and let $\{\omega_r\}_{r=1}^M$ denote M realizations of the Euler approximation of the stochastic differential equation corresponding to L and with initial value at t given by x. For each realization we derive an approximation $F(\omega_r)$, $r \in \{1,\ldots,M\}$, of the functional in (3.2) and subsequently we get the estimate

$$w(t,x) \approx \frac{1}{M} \sum_{r=1}^{M} F(\omega_r), \tag{3.3}$$

where $F(\omega_r)$, $r \in \{1,\ldots,M\}$, is calculated as follows. If $X^\Delta(t_{k+1}) \notin \Omega$, for some $k \in \{0,\ldots,N-1\}$, then the trajectory has left the domain. Hence $\tau \leq T$ and we set

$F(\omega_r) = 0$. Otherwise, $X^\Delta(t_{k+1}) \in \Omega$, for all $k \in \{0, \ldots, N-1\}$, and we estimate the probability that the continuous Euler approximation has exited the domain during any of the time steps. Indeed, given $X^\Delta(t_k)$ we let z be a point on $\partial\Omega$ which minimizes the distance from $X^\Delta(t_k)$ to $\partial\Omega$ and we let $\tilde{n}(z)$ denote the unit inward normal to Ω at z. We then generate a Bernoulli random variable U_k with parameter

$$1 - \exp\left(-2\frac{\left\langle \tilde{n}(z), X^\Delta(t_k) - z\right\rangle \left\langle \tilde{n}(z), X^\Delta(t_{k+1}) - z\right\rangle}{\left(\sum_{i,j=1}^n \tilde{n}_i(z) a_{ij}(t_k)\, \tilde{n}_j(z)\right)\Delta t_k}\right). \tag{3.4}$$

If $U_k = 0$, for some $k \in \{0, \ldots, N-1\}$, then the trajectory of a continuous Euler approximation is considered as having left the domain on the interval $[t_k, t_{k+1}]$ and we set $F(\omega_r) = 0$. If $U_k = 1$, for all $k \in \{0, \ldots, N-1\}$, then we set $F(\omega_r) = h\left(X^\Delta(T)\right)$.

As mentioned above, the algorithm in [4] concerns killed diffusion on time-independent domains. As such, two issues have to be resolved before the algorithm in [4] can be implemented in the case of problems (P1)–(P4). Firstly, during each time step the domain has to be approximated by a half space and there are several ways, in the case of (P1)–(P2), of extending the spatial half space used in the original algorithm to the case of time-dependent domains. Note that in the case of (P3)–(P4) this issue is trivial as the domain itself is a half space. Secondly, the stochastic representation in (1.17) contains the functional $g(\tau, X(\tau))$. Hence we need to find an estimate of this functional given that $\tau \in [t_k, t_{k+1}]$.

In the case of problems (P1)–(P2) we consider three different approximating half spaces. These three choices of approximating half spaces will be referred to as STP, TP and CH in the simulations in Section 4 below. Let $[t_k, t_{k+1}]$ be an arbitrary time step and let $X^\Delta(t_k)$ and $X^\Delta(t_{k+1})$ be the discrete Euler approximations of X at the endpoints of the time step. Let

$$\Omega_{\psi,\epsilon,T}(t) = \{x : (t,x) \in \Omega_{\psi,\epsilon,T}\},$$

be the n-dimensional time section of $\Omega_{\psi,\epsilon,T}$ at $t \in [0,T]$. We begin by considering (P1) and let, for any $t \in [0,T]$,

$$\pi_t(x) = \left(\pi_t'(x), \pi_t^n(x)\right) \in \mathbb{R}^{n-1} \times \mathbb{R}, \tag{3.5}$$

be the normal projection of $x \in \mathbb{R}^n$ onto $S_{\psi,\epsilon,T}(t)$. Furthermore let $n(t_k)$ and $n(t_{k+1})$ be the (spatial) unit inward normals to $\Omega_{\psi,\varepsilon,T}(t_k)$ at $\pi_{t_k}\left(X^\Delta(t_k)\right)$ and to $\Omega_{\psi,\varepsilon,T}(t_{k+1})$ at $\pi_{t_k}\left(X^\Delta(t_{k+1})\right)$, respectively. Next, we specify the construction of the approximating half spaces.

STP In this case we approximate $S_{\psi,\varepsilon,T} \cap [t_k, t_{k+1})$ by the spatial tangent plane of $\Omega_{\psi,\varepsilon,T}(t_k)$ at $\pi_{t_k}\left(X^\Delta(t_k)\right)$, and the parameter of the Bernoulli random variable U_k is

$$1 - \exp\left(- \frac{\left\langle n(t_k), X^\Delta(t_k) - \pi_{t_k}\left(X^\Delta(t_k)\right)\right\rangle \left\langle n(t_k), X^\Delta(t_{k+1}) - \pi_{t_k}\left(X^\Delta(t_k)\right)\right\rangle}{\Delta t_k}\right).$$

TP In this case we approximate $S_{\psi,\varepsilon,T} \cap [t_k, t_{k+1})$ by the space-time tangent plane of $\Omega_{\psi,\varepsilon,T}$ at $\left(t_k, \pi_{t_k}\left(X^\Delta(t_k)\right)\right)$, and the parameter of the Bernoulli random variable U_k is

$$1 - \exp\left(- \frac{\left\langle n(t_k), X^\Delta(t_k) - \pi_{t_k}\left(X^\Delta(t_k)\right)\right\rangle \left\langle n(t_k), X^\Delta(t_{k+1}) - \widetilde{\pi}_{t_k}\left(X^\Delta(t_k)\right)\right\rangle}{\Delta t_k}\right),$$

where

$$\widetilde{\pi}_{t_k}(x) = \left(\pi'_{t_k}(x), \widetilde{\pi}^n_{t_k}(x)\right), \tag{3.6}$$

with

$$\widetilde{\pi}^n_{t_k}(x) = \pi^n_{t_k}(x) + (t_{k+1} - t_k)\,\partial_t\psi\left(t_k, \pi'_{t_k}(x)\right), \tag{3.7}$$

being the nth coordinate at $(t, x') = \left(t_{k+1}, \pi'_{t_k}(x)\right)$ of the space-time tangent plane of $\Omega_{\psi,\varepsilon,T}$ at $(t_k, \pi_{t_k}(x))$.

CH In this case we approximate $S_{\psi,\varepsilon,T} \cap [t_k, t_{k+1})$ by a linear combination in time of the spatial tangent planes of $\Omega_{\psi,\varepsilon,T}(t_k)$ at $\pi_{t_k}\left(X^\Delta(t_k)\right)$ and $\Omega_{\psi,\varepsilon,T}(t_{k+1})$ at $\pi_{t_k}\left(X^\Delta(t_{k+1})\right)$. The parameter of the Bernoulli random variable U_k is

$$1 - \exp\left(- \frac{\left\langle n(t_k), X^\Delta(t_k) - \pi_{t_k}\left(X^\Delta(t_k)\right)\right\rangle \left\langle n(t_{k+1}), X^\Delta(t_{k+1}) - \pi_{t_{k+1}}\left(X^\Delta(t_{k+1})\right)\right\rangle}{\langle n(t_k), n(t_{k+1})\rangle\, \Delta t_k}\right).$$

Note that, in most cases, this linear combination of half spaces in $\mathbb{R}^n$ will not give rise to a half space in $\mathbb{R}^{n+1}$. Hence, strictly speaking, this case does not represent a half space approximation of $\Omega_{\psi,\varepsilon,T}$. As the numerical results in Section 4 suggest it is, nevertheless, a good approximation to $\Omega_{\psi,\varepsilon,T}$ during $[t_k, t_{k+1})$.

In the case of (P2), the time derivative of ψ may not be well defined and, as a consequence, there is no counterpart of (3.6)–(3.7) in this case. This implies that the TP half space approximation cannot be used for problem (P2). Note also that since it may be troublesome to find an explicit expression for the normal projection onto $S_{\psi,\epsilon,T}(t)$, it is often more feasible to let π_t be the projection of $x = (x', x_n) \in \mathbb{R}^n$ onto $S_{\psi,\epsilon,T}(t)$ defined by

$$\pi_t(x) = \left(x', \psi\left(t, x'\right)\right) \in \mathbb{R}^{n-1} \times \mathbb{R}, \tag{3.8}$$

for all $t \in [0, T]$. In this case, let $n(t_k)$ be the unit vector in the positive x_n-direction.

To resolve the second issue discussed above, we need to find an estimate of $g(\tau, X(\tau))$, given that $\tau \in [t_k, t_{k+1}]$, and we do this by constructing suitable projections of $X^\Delta(t)$ onto $S_{\psi,\epsilon,T}$. We here only discuss the case of problems (P1)–(P2) as a similar argument can be applied to the case of problems (P3)–(P4). If $X^\Delta(t_{k+1}) \notin \Omega_{\psi,\epsilon,T}$, for some $k \in \{0, \ldots, N-1\}$, we determine the smallest such k and set

$$g(\tau, X(\tau)) = g\left(t_{k+1}, \pi_{t_{k+1}}\left(X^\Delta(t_{k+1})\right)\right), \tag{3.9}$$

where π_t is either of the projection operators onto $S_{\psi,\epsilon,T}$ defined in (3.5) or (3.8). Otherwise, if $X^\Delta(t_{k+1}) \in \Omega_{\psi,\epsilon,T}$, for all $k \in \{0, \ldots, N-1\}$, and the Bernoulli random variables $\{U_k\}$ satisfy $U_k = 0$ for some $k \in \{0, \ldots, N-1\}$, then we find the smallest such k and set

$$g(\tau, X(\tau)) = g\left(\frac{t_k + t_{k+1}}{2}, \pi_{\frac{t_k+t_{k+1}}{2}}\left(\frac{X^\Delta(t_k) + X^\Delta(t_{k+1})}{2}\right)\right). \tag{3.10}$$

If $g(\tau, X(\tau))$ has not been assigned a value at this point, we set $g(\tau, X(\tau)) = 0$.

3.1.1. A non-adaptive algorithm using the distribution of the first exit time

It is clear from (1.17) that it is important to specify the first exit time τ of the process X from the domain $\Omega_{\psi,\epsilon,T}$ as accurately as possible, especially for boundary data which varies significantly. Moreover, if X solves a stochastic differential equation with constant coefficients and Ω is a half space, then explicit expressions for the distribution of the first exit time can be derived (see [10] for example). Hence, as the algorithm in [4] approximates a given stochastic differential equation on a bounded domain with a constant coefficient stochastic differential equation on a half space, we can make use of the distribution of the first exit time of the approximating process as this distribution is explicitly known. We therefore suggest that the performance of the algorithm in [4] can be significantly improved by using the first exit time distribution in the estimation of the expectations in (1.17) and we here give a simple one-dimensional example to illustrate the idea. In particular, we consider $\Omega_{\psi,0,T} \subset \mathbb{R}^2$, i.e., $\psi = \psi(t)$. Consider the CH algorithm described in the previous section on the interval $[t_k, t_{k+1})$. Then, the density of the first exit time of the continuous Euler approximation during $[t_k, t_{k+1}]$ is given, see [10], by

$$p_{\tau - t_k}(u) = \left(X^\Delta(t_k) - \psi(t_k)\right)\sqrt{\frac{\Delta t_k}{(\Delta t_k - u)\, 2\pi u^3 \sigma^2(t_k)}}$$

$$\times \exp\left(-\frac{\left(\left(X^\Delta(t_{k+1}) - \psi(t_{k+1})\right)u + \left(X^\Delta(t_k) - \psi(t_k)\right)(\Delta t_k - u)\right)^2}{2\sigma^2(t_k)\Delta t_k u\,(\Delta t_k - u)}\right). \tag{3.11}$$

In this one-dimensional example $X(\tau) = \psi(\tau)$ for all $\tau \leq T$ and the expectations in (1.17) can be approximated as

$$E\left[g(\tau, X(\tau)) \,|X(0) = x\right] \approx E\left[g\left(\tau_c^\Delta, X^\Delta\left(\tau_c^\Delta\right)\right) |X(0) = x\right]$$
$$= \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} g(u, \psi(u))\, p_{\tau - t_k}(u)\, du. \tag{3.12}$$

For most boundary data the integral in (3.12) cannot be solved explicitly. Nevertheless, as we shall see in Section 4, the use of a numerical approximation of (3.12) in the algorithm in [4] improves the result of the algorithm in many cases. Algorithms using the density of the first exit time of the continuous Euler approximation will be referred to as FET algorithms in the numerical simulations in Section 4 below.

3.2. An adaptive approach

The idea of adaptive algorithms is to refine the time mesh at time steps where the absolute value of the error function is large. In order to derive an adaptive algorithm for the problem considered in this article, we must be able to control the three error terms given in (2.8)–(2.10) above. In the following we present this approach for the problems (P3)–(P4), the case of the problems (P1)–(P2) being, for obvious reasons, much simpler. In particular, in Section 3.2.1 we show that the time-discretization error $\Delta_c^N(x)$ can be controlled using an expansion from [3] and in Section 3.2.2 we develop an expansion for the first exit error $\Delta_d^N(x)$. Finally, as shown in Section 3.2.3, the statistical error $\Delta_s^{\Delta,M}(x)$ can be controlled using the central limit theorem and the Berry–Esséen theorem.

3.2.1. A computable expansion of $\Delta_c^N(x)$

We here describe the expansion, derived in [3], of the interior discretization error $\Delta_c^N(x)$ and we recall that $\Delta_c^N(x)$ was defined in (2.8). To present the initial steps in the deduction of the expansion in an accessible form, we first introduce some additional notation. In particular, we let $\{t_k\}_{k=0}^N$ define a partition Δ of the interval $[0, T]$ and we recall that $\phi(t) = \sup\{t_k : t_k \leq t\}$, whenever $t \in [0, T]$. Moreover, we let

$$\mu_i^\Delta(t,x) = \mu_i(\phi(t), x), \quad \sigma_{ij}^\Delta(t,x) = \sigma_{ij}(\phi(t), x), \quad a_{ij}^\Delta(t,x) = a_{ij}(\phi(t), x).$$

Furthermore, we define the operator

$$L^\Delta = \sum_{i,j=1}^n a_{ij}^\Delta(t,x)\partial_{x_i x_j}^2 + \sum_{i=1}^n \mu_i^\Delta(t,x)\partial_{x_i}.$$

Applying Itô's formula to the function u we see that

$$du(t, X^\Delta(t)) = \left(\partial_t u + L^\Delta u\right)(t, X^\Delta(t))\, dt + \sum_{i,j=1}^n \sigma_{ij}^\Delta \partial_{x_i} u(t, X^\Delta(t))\, dW_j(t),$$

whenever $t \leq \tau_c^\Delta$. Furthermore, as u is assumed to solve the problem $\mathcal{P}(L, g, 0, \epsilon, T)$, where either $L = L_1$ or $L = L_2$, we can use this to replace $\partial_t u(t, X^\Delta(t))$ and we obtain

$$du(t, X^\Delta(t)) = (L^\Delta - L)u(t, X^\Delta(t))\, dt + \sum_{i,j=1}^{n} \sigma_{ij}^\Delta(t) \partial_{x_i} u(t, X^\Delta(t))\, dW_j(t). \quad (3.13)$$

Integrating (3.13) from 0 to τ_c^Δ and taking expectations, we see that

$$E\left[u(\tau_c^\Delta, X^\Delta(\tau_c^\Delta))\right] - E\left[u(0, X^\Delta(0))\right] \\ = E\Big[\int_0^{\tau_c^\Delta} (L^\Delta - L)u(t, X^\Delta(t)))\, dt \Big| X^\Delta(0) = x\Big].$$

The vanishing of the expectation of the Itô integral in this deduction is non-trivial as τ_c^Δ is not adapted to the filtration generated by the Wiener process alone. Nevertheless, using Lemma 4.2 in [9], it follows that this term is zero. Hence we conclude that

$$\Delta_c^\Delta(x) = E\left[\int_0^{\tau_c^\Delta} (L - L^\Delta)u(t, X^\Delta(t)))\, dt \Big| X^\Delta(0) = x\right]. \quad (3.14)$$

The idea is now to proceed from the representation formula of $\Delta_c^\Delta(x)$ in (3.14). In particular, we let, for $k \in \{0, \ldots, N-1\}$,

$$\begin{aligned}\tilde{\rho}_k^\Delta = {} & \frac{1}{2}(\partial_t + L^\Delta)\mu_i(t_k, X^\Delta(t_k))\partial_{x_i} u(t_{k+1}, X^\Delta(t_{k+1})) \\ & + \frac{1}{2}(\partial_t + L^\Delta)a_{ij}(t_k, X^\Delta(t_k))\partial^2_{x_i x_j} u(t_{k+1}, X^\Delta(t_{k+1})) \\ & + a_{lm}(t_k, X^\Delta(t_k))\partial_{x_l}\mu_i(t_k, X^\Delta(t_k))\partial^2_{x_i x_m} u(t_{k+1}, X^\Delta(t_{k+1})) \\ & + a_{lm}(t_k, X^\Delta(t_k))\partial_{x_l} a_{ij}(t_k, X^\Delta(t_k))\partial^3_{x_i x_j x_m} u(t_{k+1}, X^\Delta(t_{k+1})), \end{aligned} \quad (3.15)$$

where we have used summation convention. Let $\Delta_{\max} = \max_{0 \leq k \leq N-1} (t_{k+1} - t_k)$. Using Lemma 2.2 in [3], we can conclude that the following holds in our situation.

Lemma 3.1. $\Delta_c^\Delta(x)$ *satisfies*

$$\Delta_c^\Delta(x) = E\left[\sum_{k=0}^{N-1} \chi_{[0,\tau_d^\Delta]}(t_{k+1})\tilde{\rho}_k^\Delta (\Delta_k)^2\right] + \mathcal{O}\left(\sqrt{\Delta_{\max}}\right) E\left[\sum_{k=0}^{N-1} \mathcal{O}((\Delta_k)^2)\right], \quad (3.16)$$

where $\tilde{\rho}_k^\Delta$ *was defined in* (3.15)*.*

Based on this lemma it is clear that the error expansion for $\Delta_c^\Delta(x)$ is computable if we are able to replace the derivatives

$$\partial_{x_i} u(t_{k+1}, X^\Delta(t_{k+1}))), \quad \partial^2_{x_i x_j} u(t_{k+1}, X^\Delta(t_{k+1}))), \quad \partial^3_{x_i x_j x_m} u(t_{k+1}, X^\Delta(t_{k+1}))),$$

by some computable quantities. To do this we follow [3] and introduce certain dual functions.

Recall that u solves either problem (P3) or (P4) and that $X(t) = (X_1(t), \ldots, X_n(t)) \in \mathbb{R}^n$, $t \in [0,T]$, solves (1.18). The definition of the appropriate discrete dual functions $\{\phi_i\}_{i=1}^N$, associated to g, is based on our choice of approximations of partial derivatives, in the spatial variables, of u on $S_{0,\epsilon,T}$. To proceed, we first define, whenever $i \in \{1, \ldots, n\}$ and $x \in \mathbb{R}^n$ with $x_n > 0$,

$$c_i(t_k, x) = x_i + \mu_i(t_k, x)\Delta t_k + \sum_{j=1}^n \sigma_{ij}(t_k, x)\Delta W_j\,(t_k)\,, \tag{3.17}$$

for $k \in \{0, \ldots, N-1\}$. Next, we define the discrete dual functions $\{\phi_i\}_{i=1}^n$, associated to g. In particular, we let Δx be a small positive constant and define

$$\begin{aligned}
&\phi_i(\tau_d^\Delta) = 0, && \text{if } \tau_d^\Delta = T \text{ and } i \in \{1, \ldots, n\},\\
&\phi_i(\tau_d^\Delta) = \partial_{x_i} g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta)), && \text{if } \tau_d^\Delta < T \text{ and } i \in \{1, \ldots, n-1\},\\
&\phi_n(\tau_d^\Delta) = \frac{g(\hat\tau_d^\Delta, \hat X^\Delta(\hat\tau_d^\Delta)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))}{\Delta x}, && \text{if } \tau_d^\Delta < T.
\end{aligned} \tag{3.18}$$

Note that if $\tau_d^\Delta < T$, then $(\tau_d^\Delta, X^\Delta(\tau_d^\Delta)) \in S_{0,\epsilon,R}$ and in this case any derivative of g at $(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))$, with respect to the variables defining x', i.e. $x_1, \ldots, x_{n-1}$, is well-defined as these derivatives are tangent to the boundary $x_n = 0$. This is used in the expression for $\phi_i(\tau_d^\Delta)$ in (3.18). However, to calculate derivatives of g at $(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))$ in the direction of the normal, i.e. with respect to x_n, one has to be more innovative and therefore we have introduced the difference quotient in the expression for $\phi_n(\tau_d^\Delta)$ in (3.18). As the discrete path of the solution $(t, X^\Delta(t))$ to (1.18) crosses the boundary, we start a new realization $(s, \hat X^\Delta(s))$ with the initial value

$$\hat X^\Delta(\tau_d^\Delta) = X^\Delta(\tau_d^\Delta) + (\Delta x)e_n, \quad s = \tau_d^\Delta,$$

so that $(\tau_d^\Delta, \hat X^\Delta(\tau_d^\Delta)) \in \Omega_{0,\epsilon,T}$. We then let $\hat X^\Delta(t)$ evolve according to the underlying system of stochastic differential equations during $(\tau_d^\Delta, \hat\tau^\Delta)$ until it stops at the first exit time $\hat\tau^\Delta \in (\tau^\Delta, T]$. Note that the exit time $\hat\tau^\Delta$ cannot be found exactly and, in our calculations, we use a discrete approximation $\hat\tau_d^\Delta$ of $\hat\tau^\Delta$. We then construct a difference quotient based on $g(\hat\tau_d^\Delta, \hat X^\Delta(\hat\tau_d^\Delta)$ and $g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))$ as stated in (3.18). Moreover, having defined $\{\phi_i(\tau_d^\Delta)\}_{i=1}^n$, we let $\{\phi_i(t_k)\}_{i=1}^n$, whenever $t_k < \tau_d^\Delta$, be defined recursively through the difference equation

$$\phi_i(t_k) = \sum_{l=1}^n \partial_{x_i} c_l(t_k, X^\Delta(t_k))\phi_l(t_{k+1}). \tag{3.19}$$

This completes the definition of $\{\phi_i\}_{i=1}^n$.

The next step is to introduce the first and second variations of the dual functions, here denoted ϕ'_{ij} and ϕ'_{ijm}, respectively. As in the case of ϕ_i, we let Δx be a small positive constant and define

$$\begin{aligned} &\phi'_{ij}(\tau_d^\Delta) = 0, && \text{if } \tau_d^\Delta = T \text{ and } i,j \in \{1,\dots,n\},\\ &\phi'_{ij}(\tau_d^\Delta) = \partial^2_{x_i x_j} g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta)), && \text{if } \tau_d^\Delta < T \text{ and } i,j \in \{1,\dots,n-1\}. \end{aligned} \tag{3.20}$$

Furthermore,

$$\phi'_{in}(\tau_d^\Delta) = \phi'_{ni}(\tau_d^\Delta) = \frac{\partial_{x_i} g(\hat\tau_d^\Delta, \hat X^\Delta(\hat\tau_d^\Delta)) - \partial_{x_i} g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))}{\Delta x}, \tag{3.21}$$

if $\tau_d^\Delta < T$ and $i \in \{1,\dots,n-1\}$. Finally we define the second order derivative in the x_n-direction as

$$\phi'_{nn}(\tau_d^\Delta) = \frac{g(\check\tau_d^\Delta, \check X^\Delta(\check\tau_d^\Delta)) - 2g(\hat\tau_d^\Delta, \hat X^\Delta(\hat\tau_d^\Delta)) + g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))}{\Delta x}, \tag{3.22}$$

if $\tau_d^\Delta < T$. Here $\check X^\Delta$ is a realization of (1.18) starting at $\check X^\Delta(\tau_d^\Delta) = X^\Delta(\tau_d^\Delta) + 2(\Delta x)e_n$ at time $s = \tau_d^\Delta$ and $\check\tau_d^\Delta$ is a discrete approximation of the first exit time of $\check X^\Delta$. Moreover, having defined $\{\phi'_{ij}(\tau_d^\Delta)\}_{i,j=1}^n$, we let $\{\phi'_{ij}(t_k)\}_{i,j=1}^n$, whenever $t_k < \tau_d^\Delta$, be recursively defined through the difference equation

$$\begin{aligned} \phi'_{ij}(t_k) = &\sum_{r,l=1}^n \partial_{x_i} c_r(t_k, X^\Delta(t_k)) \partial_{x_j} c_l(t_k, X^\Delta(t_k)) \phi'_{rl}(t_{k+1}) \\ &+ \sum_{r,l=1}^n \partial^2_{x_i x_j} c_r(t_k, X^\Delta(t_k)) \phi_r(t_{k+1}). \end{aligned} \tag{3.23}$$

The second variation $\{\phi''_{ijm}(t_k)\}_{i,j,m=1}^n$ of the dual functions is defined along the same lines, but we omit the details.

Using the notion introduced above we can now use Lemma 2.3 in [3] to conclude that the following holds in our situation.

Lemma 3.2. *The derivatives of u are related to the dual function and its first and second variation according to*

$$\partial_{x_i} u\left(t_{k+1}, X^\Delta(t_{k+1})\right) - E\left[\phi_i\left(t_{k+1}\right)|\mathcal{F}_{k+1}\right] = \mathcal{O}\Big(\Delta x + \sqrt{\Delta_{\max}} + \frac{\sqrt{\Delta_{\max}}}{\Delta x}\Big),$$

$$\partial^2_{x_i x_j} u\left(t_{k+1}, X^\Delta(t_{k+1})\right) - E\left[\phi'_{ij}\left(t_{k+1}\right)|\mathcal{F}_{k+1}\right] = \mathcal{O}\Big(\Delta x + \sqrt{\Delta_{\max}} + \frac{\sqrt{\Delta_{\max}}}{(\Delta x)^2}\Big),$$

$$\partial^3_{x_i x_j x_m} u\left(t_{k+1}, X^\Delta(t_{k+1})\right) - E\left[\phi''_{ijm}\left(t_{k+1}\right)|\mathcal{F}_{k+1}\right] = \mathcal{O}\Big(\Delta x + \sqrt{\Delta_{\max}} + \frac{\sqrt{\Delta_{\max}}}{(\Delta x)^3}\Big).$$

Moreover let, for $k \in \{0, \dots, N-1\}$,

$$\begin{aligned}\rho_k^\Delta = &\frac{1}{2}(\partial_t + L^\Delta)\mu_i(t_k, X^\Delta(t_k))\phi_i(t_{k+1}) + \frac{1}{2}(\partial_t + L^\Delta)a_{ij}(t_k, X^\Delta(t_k))\phi'_{ij}(t_{k+1}) \\ &+ a^\Delta_{lm}(t_k, X^\Delta(t_k))\partial_{x_l}\mu_i(t_k, X^\Delta(t_k))\phi'_{im}(t_{k+1}) \\ &+ a^\Delta_{lm}(t_k, X^\Delta(t_k))\partial_{x_l}a_{ij}(t_k, X^\Delta(t_k))\phi''_{ijm}(t_{k+1}),\end{aligned} \tag{3.24}$$

where we again have used summation convention. Then, combining Lemma 3.1 and 3.2, the following theorem follows, see also Theorem 2.4 in [3].

Theorem 3.3. $\Delta_c^\Delta(x)$ *satisfies*

$$\begin{aligned}\Delta_c^N(x) = E\left[\sum_{k=0}^{N-1} \chi_{[0,\tau_d^\Delta]}(t_{k+1})\rho_k^\Delta (\Delta_k)^2\right] \\ + \mathcal{O}\left(\Delta x + \sqrt{\Delta_{\max}} + \frac{\sqrt{\Delta_{\max}}}{(\Delta x)^3}\right) E\left[\sum_{k=0}^{N-1}(\Delta_k)^2\right],\end{aligned} \tag{3.25}$$

where ρ_k^Δ *was defined in* (3.24)*.*

Example 3.4. Here we calculate the dual functions in the case of problem (P3) for $n = 1$. In particular, we first see that if $\tau_d^\Delta = T$, then

$$\phi(\tau_d^\Delta) = \phi'(\tau_d^\Delta) = \phi''(\tau_d^\Delta) = 0,$$

and hence, as expected, only trajectories ending up at the lateral boundary contribute to the time-discretization error $\Delta_c^N(x)$. If $\tau_d^\Delta < T$ then we get, using $X^\Delta(\tau) = \psi(\tau)$ and the operators in Section 2.1.1,

$$\begin{aligned}\phi(\tau_d^\Delta) &= \frac{g(\hat\tau_d^\Delta, \psi(\hat\tau_d^\Delta)) - g(\tau_d^\Delta, \psi(\tau_d^\Delta))}{\Delta x}, \\ \phi'(\tau_d^\Delta) &= \partial_t\psi(\tau_d^\Delta)\phi(\tau_d^\Delta) - \partial_t g(\psi(\tau_d^\Delta), \tau_d^\Delta)), \\ \phi''(\tau_d^\Delta) &= \partial_t\psi(\tau_d^\Delta)\phi'(\tau_d^\Delta) - \frac{\partial_t g(\hat\tau_d^\Delta, \psi(\hat\tau_d^\Delta)) - \partial_t g(\tau_d^\Delta, \psi(\tau_d^\Delta))}{\Delta x}.\end{aligned}$$

Furthermore, we define

$$c(t_k, x) = x + \mu(t_k, x)\Delta t_k + \sigma(t_k, x)\Delta W(t_k), \tag{3.26}$$

whenever $x \in \mathbb{R}$, with $x > 0$. The dual functions are defined by the recursive relations

$$\begin{aligned}\phi(t_k) &= \partial_x c(t_k, X^\Delta(t_k))\phi(t_{k+1}), \\ \phi'(t_k) &= (\partial_x c(t_k, X^\Delta(t_k)))^2\phi'(t_{k+1}) + \partial^2_{xx}c(t_k, X^\Delta(t_k))\phi(t_{k+1}), \\ \phi''(t_k) &= (\partial_x c(t_k, X^\Delta(t_k)))^3\phi''(t_{k+1}) + 3\partial_x c(t_k, X^\Delta(t_k))\partial^2_{xx}c(t_k, X^\Delta(t_k))\phi'(t_{k+1}) \\ &\quad + \partial^3_{xxx}c(t_k, X^\Delta(t_k))\phi(t_{k+1}).\end{aligned} \tag{3.27}$$

For the smooth data in problem (P3), we can use (2.2)–(2.3) to conclude that $c(t_k, x) = x$, and hence (3.27) is further reduced into

$$\phi(t_k) = \phi(\tau_d^\Delta), \quad \phi'(t_k) = \phi'(\tau_d^\Delta), \quad \phi''(t_k) = \phi''(\tau_d^\Delta), \tag{3.28}$$

whenever $t_k \le \tau_d^\Delta$.

Example 3.5. Here we calculate the dual functions in the case of problem (P4) for $n = 1$. Again we first see that if $\tau_d^\Delta = T$, then

$$\phi(\tau_d^\Delta) = \phi'(\tau_d^\Delta) = \phi''(\tau_d^\Delta) = 0.$$

Moreover, if $\tau_d^\Delta < T$ we let $P_{\gamma x}\psi = P_{\gamma x}\psi(t)$ be evaluated at $(t, x) = (\tau_d^\Delta, \psi(\tau_d^\Delta))$ and we define $D = D(\tau_d^\Delta, \psi(\tau_d^\Delta))$. Using this notation and the operators in Section 2.1.2, we obtain

$$\begin{aligned}
\phi(\tau_d^\Delta) &= \frac{g(\hat\tau_d^\Delta, \psi(\hat\tau_d^\Delta)) - g(\tau_d^\Delta, \psi(\tau_d^\Delta))}{\Delta x},\\
\phi'(\tau_d^\Delta) &= \Big(D^{-1}\partial_{xx}^2(P_{\gamma x}\psi) + D\partial_t(P_{\gamma x}\psi)\Big)\phi(\tau_d^\Delta) - D^2\partial_t g(\psi(\tau_d^\Delta), \tau_d^\Delta)),\\
\phi''(\tau_d^\Delta) &= \Big(-3D^{-2}(\partial_{xx}^2(P_{\gamma x}\psi))^2 + D^{-1}\partial_{xxx}^3(P_{\gamma x}\psi) - \partial_{xx}^2(P_{\gamma x}\psi)\partial_t(P_{\gamma x}\psi)\\
&\quad + D\partial_{xt}^2(P_{\gamma x}\psi)\Big) \times \phi(\tau_d^\Delta) + \Big(3D^{-1}\partial_{xx}^2(P_{\gamma x}\psi) + D\partial_t(P_{\gamma x}\psi)\Big)\phi'(\tau_d^\Delta)\\
&\quad - D^2\frac{(\partial_t g(\hat\tau_d^\Delta, \psi(\hat\tau_d^\Delta)) - \partial_t g(\tau_d^\Delta, \psi(\tau_d^\Delta))}{\Delta x}.
\end{aligned} \tag{3.29}$$

From here on we can follow (3.26) and (3.27).

3.2.2. A computable expansion of $\Delta_d^N(x)$

We here derive an expansion of the exit error $\Delta_d^N(x)$ and we recall that $\Delta_d^N(x)$ was defined in (2.9). Let $\{X^\Delta(t), t \in \{t_0, \dots, t_N\}\}$ be the associated discrete Euler approximation and let

$$\begin{aligned}
&P^\Delta(t_k, t_{k+1})\\
&\quad = \mathbf{P}\big[\{t \in (t_k, t_{k+1}) : (t, X^\Delta(t)) \notin \Omega_{0,\epsilon,T}\} \neq \emptyset \big| X^\Delta(t_k), X^\Delta(t_{k+1}), \tau_d^\Delta > t_{k+1}\big]
\end{aligned}$$

denote the conditional probability that $(t, X^\Delta(t))$ exits the domain $\Omega_{0,\epsilon,T}$ for some $t \in (t_k, t_{k+1})$ conditioned on the values of $X^\Delta(t_k)$ and $X^\Delta(t_{k+1})$ and conditioned on the event that $\tau_d^\Delta > t_{k+1}$.

The event $\tau_d^\Delta > t_{k+1}$ is equivalent to the event $X^\Delta(t_k), X^\Delta(t_{k+1}) \in \Omega_{0,\epsilon,T}$. Note that we have the following explicit formula for $P^\Delta(t_k, t_{k+1})$,

$$\begin{aligned} P^\Delta(t_k, t_{k+1}) &= \mathbf{P}\left[\min_{t\in[t_k,t_{k+1}]} X_n^\Delta(t) \leq 0 \big| X^\Delta(t_k), X^\Delta(t_{k+1}), \tau_d^\Delta > t_{k+1}\right] \\ &= \exp\left(-2\frac{X_n^\Delta(t_k)X_n^\Delta(t_{k+1})}{\sum_{j=1}^n a_{nj}(t_k)\, a_{nj}(t_k)\, \Delta t_k}\right). \end{aligned} \tag{3.30}$$

Let $\mathbf{F}_k^\Delta$ denote the σ-algebra generated by $\{X^\Delta(t),\ t \in \{t_0, \ldots, t_k\}\}$, for $k \in \{0, \ldots, N\}$. We introduce the functions $\widetilde{P}^\Delta(t_k, t_{k+1})$

$$\begin{aligned} \widetilde{P}^\Delta(t_k, t_{k+1}) &= P^\Delta(t_k, t_{k+1}) \prod_{l=0}^{k-1} (1 - P^\Delta(t_l, t_{l+1})), \quad \text{for } k = 1, 2, \ldots, N-1, \\ \widetilde{P}^\Delta(t_0, t_1) &= P^\Delta(t_0, t_1), \end{aligned}$$

representing the probability, conditioned on $\mathbf{F}_N^\Delta$, that $\tau_c^\Delta \in (t_k, t_{k+1})$. To continue we note that, as $\tau_c^\Delta \leq \tau_d^\Delta$ with probability 1, we can use the decomposition

$$\chi_{[0,T)}(\tau_c^\Delta) = \sum_{k=0}^{N-1} \chi_{[t_k,t_{k+1})}(\tau_c^\Delta),$$

to obtain

$$\begin{aligned} \Delta_d^\Delta(x) &= E\left[\left(g(\tau_c^\Delta, X^\Delta(\tau_c^\Delta)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))\right)\chi_{[0,T)}(\tau_c^\Delta)\right] \\ &= E\left[\sum_{k=0}^{N-1}\left(g(\tau_c^\Delta, X^\Delta(\tau_c^\Delta)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))\right)\chi_{[t_k,t_{k+1})}(\tau_c^\Delta)\right] \\ &= E\left[\sum_{k=0}^{N-1} E\left[\left(g(\tau_c^\Delta, X^\Delta(\tau_c^\Delta)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))\right)\chi_{[t_k,t_{k+1})}(\tau_c^\Delta)\big|\mathbf{F}_N^\Delta\right]\right]. \end{aligned} \tag{3.31}$$

Moreover, using properties of conditional expectation, we see that

$$\begin{aligned} &E\Big[\left(g(\tau_c^\Delta, X^\Delta(\tau_c^\Delta)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))\right)\chi_{[t_k,t_{k+1})}(\tau_c^\Delta)|\mathbf{F}_N^\Delta\Big] \\ &= E\Big[E\big[\left(g(\tau_c^\Delta, X^\Delta(\tau_c^\Delta)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))\right)\chi_{[t_k,t_{k+1})}(\tau_c^\Delta)|\mathbf{F}_N^\Delta, \tau_c^\Delta \in [t_k, t_{k+1})\big]|\mathbf{F}_N^\Delta\Big] \\ &= E\Big[\chi_{[t_k,t_{k+1})}(\tau_c^\Delta)E\big[g(\tau_c^\Delta, X^\Delta(\tau_c^\Delta)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))|\mathbf{F}_N^\Delta, \tau_c^\Delta \in [t_k, t_{k+1})\big]|\mathbf{F}_N^\Delta\Big] \\ &= E\Big[g(\tau_c^\Delta, X^\Delta(\tau_c^\Delta)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))|\mathbf{F}_N^\Delta, \tau_c^\Delta \in [t_k, t_{k+1})\Big]E\Big[\chi_{[t_k,t_{k+1})}(\tau_c^\Delta)|\mathbf{F}_N^\Delta\Big] \\ &= E\Big[g(\tau_c^\Delta, X^\Delta(\tau_c^\Delta)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))|\mathbf{F}_N^\Delta, \tau_c^\Delta \in [t_k, t_{k+1})\Big]\widetilde{P}^\Delta(t_k, t_{k+1}). \end{aligned} \tag{3.32}$$

Hence, combining (3.31)–(3.32), we can conclude that

$$\Delta_d^\Delta(x)$$
$$= E\Bigg[\sum_{k=0}^{N-1} E\left[g(\tau_c^\Delta, X^\Delta(\tau_c^\Delta)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))\big|\mathbf{F}_N^\Delta, \tau_c^\Delta \in [t_k, t_{k+1})\right] \widetilde{P}^\Delta(t_k, t_{k+1})\Bigg].$$

Furthermore, using the mean value theorem, we see that there exists a point $(\tau_k, X^\Delta(\tau_k))$, satisfying $\tau_k \in (t_k, t_{k+1})$ and $X^\Delta(\tau_k) \in S_{0,\epsilon,T}$, such that

$$\Delta_d^\Delta(x) = E\Bigg[\sum_{k=0}^{N-1} \left(g(\tau_k, X^\Delta(\tau_k)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))\right) \widetilde{P}^\Delta(t_k, t_{k+1})\Bigg]. \tag{3.33}$$

The exact location of the point $(\tau_k, X^\Delta(\tau_k))$ is unknown, so we need an approximation of $g(\tau_k, X^\Delta(\tau_k)) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))$. In particular, let π_t be either of the projection operators onto $S_{0,\epsilon,T}$ defined in (3.5) or (3.8). We approximate $\Delta_d^\Delta(x)$ by:

$$E\Bigg[\sum_{k=0}^{N-1} \left\{g\Big(\frac{t_k + t_{k+1}}{2}, \pi_{\frac{t_k+t_{k+1}}{2}}\Big(\frac{X^\Delta(t_k) + X^\Delta(t_{k+1})}{2}\Big)\Big) - g(\tau_d^\Delta, X^\Delta(\tau_d^\Delta))\right\}$$
$$\times \widetilde{P}^\Delta(t_k, t_{k+1})\Bigg]. \tag{3.34}$$

A Taylor expansion shows that, under sufficient smoothness assumptions, the error arising due to the approximation of $\Delta_d^\Delta(x)$ in (3.34) is of the order

$$E\Bigg[\sum_{k=0}^{N-1} \widetilde{P}^\Delta(t_k, t_{k+1})\Bigg]\sqrt{\Delta_{\max}}.$$

3.2.3. Controlling the statistical error $\boldsymbol{\Delta_s^{\Delta,M}(x)}$

Let, in general, Y be a random variable defined on a probability space $(\Omega, \mathcal{F}, \mathbf{P})$ and let $\{Y(\omega_j)\}_{j=1}^M$, $\omega_j \in \Omega$, denote M independent samples of Y. Let $\mathcal{A}(M, Y)$ and $\mathcal{S}(M, Y)$ denote the sample average and sample standard deviation respectively, i.e.,

$$\mathcal{A}(M, Y) = \frac{1}{M}\sum_{j=1}^{M} Y(\omega_j), \quad \mathcal{S}(M, Y) = \Big(\mathcal{A}(M, Y^2) - (\mathcal{A}(M, Y))^2\Big)^{1/2}.$$

Moreover, let $\sigma = (E(|Y - E(Y)|^2))^{1/2}$ and assume that $\lambda = \frac{1}{\sigma}(E(|Y - E(Y)|^3))^{1/3} < \infty$. Define

$$Z_M = \frac{\mathcal{A}(M, Y) - E(Y)}{\sigma/\sqrt{M}},$$

and let $F_{Z_M}(z) = \mathbf{P}(Z_M \leq z)$ be the cumulative distribution function of Z_M. Similarly let $\Phi(z)$, $z \in \mathbb{R}$, be the cumulative distribution function of a standard normal random variable with zero mean and unit variance. By the Berry–Esséen theorem, see for example Theorem 2.4.10 in [2], we see that

$$\sup_{z\in\mathbb{R}} |F_{Z_M}(z) - \Phi(z)| \leq \frac{3\lambda^3}{\sqrt{M}}.$$

In particular, if we introduce the error $\mathcal{E}_S(M,Y) := E(Y) - \mathcal{A}(M,Y)$ then

$$\mathbf{P}\left(|\mathcal{E}_S(M,Y)| \leq c_0 \frac{\sigma}{\sqrt{M}} \right) \geq 2\Phi(c_0) - 1 - 2\sup_{z\in\mathbb{R}} |F_{Z_M}(z) - \Phi(z)|.$$

Let $M = \beta^2\lambda^6$ where $\beta \gg 1$ and let α be defined through the relation $\Phi(c_0) = \alpha$. Then, combining the estimates in the last two displays we see that

$$\mathbf{P}\left(|\mathcal{E}_S(M,Y)| \leq c_0 \frac{\sigma}{\sqrt{M}} \right) \geq 2\alpha - 1 - 6\beta^{-1}.$$

In particular, if we let $\beta^2 \gg 14400$ and $c_0 \geq 1.96$ then $\mathbf{P}(|\mathcal{E}_S(M,Y)| \leq c_0 \frac{\sigma}{\sqrt{M}}) \geq 0.90$. Finally, using $\mathcal{S}(M,Y)$ as an approximation of σ we can ensure that

$$|\mathcal{E}_S(M,Y)| \leq E_S(M,Y) := c_0 \frac{\mathcal{S}(M,Y)}{\sqrt{M}},$$

with probability close to one. Setting $Y(\omega_j) = g(\tau_d^\Delta(\omega_j), X^\Delta(\tau_d^\Delta(\omega_j))$, for $j = 1,\ldots,M$, where M is sufficiently large, it follows from the discussion above that the statistical error has the upper bound

$$\begin{aligned}
&\Delta_s^{\Delta,M}(x) \\
&\quad \leq \frac{c_0}{M}\Bigg[\sum_{r=1}^{M} \big(g(\tau_d^\Delta(\omega_r), X^\Delta(\tau_d^\Delta(\omega_r)) \big)^2 - \frac{1}{M}\Big(\sum_{r=1}^{M} g(\tau_d^\Delta(\omega_r), X^\Delta(\tau_d^\Delta(\omega_r)) \Big)^2 \Bigg]^{1/2},
\end{aligned} \tag{3.35}$$

with probability close to one.

3.2.4. An adaptive algorithm

Equipped with the error expansions for $\Delta_c^N(x)$, $\Delta_d^N(x)$ and $\Delta_s^{\Delta,M}(x)$ in (3.25), (3.34) and (3.35), respectively, we now outline the adaptive algorithm. As mentioned above, we use the algorithm proposed in [3] adapted to our setting. In short, we begin by calculating M trajectories of (1.18) using the standard Euler approximation on a uniform mesh of N_{init} time steps. For each trajectory, we calculate the errors $\Delta_c^N(x)$ and $\Delta_d^N(x)$

according to (3.25) and (3.34) and refine the time steps where the discretization error is larger than some tolerance. This process is then iterated until $\Delta_c^N(x)$ and $\Delta_d^N(x)$ are sufficiently small on all time steps. When these operations have been carried out for all M trajectories, (3.35) is used to estimate the statistical error. If the statistical error exceeds some given tolerance, the M trajectories are discarded and $M' \gg M$ new trajectories are generated as above. This process is then iterated until the statistical error is sufficiently low. For details we refer to the description of algorithm B in [3].

4. Numerical examples

In this section we use the following four classes of Monte Carlo algorithms, outlined in the previous section, to numerically solve problems (P1)–(P4) for $n = 1$.

(i) The Euler approximation.

(ii) Versions of the non-adaptive algorithm in [4] described in Section 3.1, referred to as Gobet type algorithms below.

(iii) The non-adaptive algorithm described in Section 3.1.1, referred to as the Gobet FET algorithm below.

(iv) The adaptive algorithm in [3] described in Section 3.2.

For (P1) there are three different Gobet type algorithms (STP, TP and CH), for (P2) two different Gobet type algorithms (STP and CH) whereas for (P3)–(P4) all three Gobet type algorithms coincide. As the basis for our study we consider three cases: smooth data on a smooth boundary, non-smooth data on a smooth boundary and smooth data on a non-smooth boundary. In the following let M be the number of trajectories and N be the number of time steps used.

Example 4.1 (Problem (P1) with smooth data). In this example we consider problem (P1) with $\psi(t) = \sin(mt)$, for some positive real number m, and $T = 25$. We let the boundary data g be given by $g(t,x) = \sin(mt)$ whenever $(t,x) \in S_{\psi,0,25}$. The sensitivity of the numerical algorithms to rapid variations of the defining function and boundary data is quantified by the parameter m. For $\Omega_{\psi,0,25}$ and g as above we estimate the solution $u(t,x)$ at $(t,x) = (0,5)$ for $m = 1$ and $m = 5$. To obtain 'correct' values of the solutions we produced $M = 10^8$ trajectories with the Gobet type algorithm (CH). We used $N = 10^4$ time steps in the case $m = 1$ and $N = 5 \cdot 10^4$ time steps in the case $m = 5$ and obtained the reference values 0.1522 and 0.2955 respectively.

Example 4.2 (Problem (P1) with non-smooth data). In this example we let ψ and T be as in Example 4.1 with the exception that we now consider only the case $m = 1$. Furthermore the boundary data g is non-smooth and given by the indicator function

of a given set $A \subset S_{\psi,0,25}$. In the stochastic setting this is equivalent to determining the probability that a trajectory of the stochastic differential equation (1.18) exits the domain at a point in A. In particular, we define g as $g(t,x) = \chi_{A_i}(t,x)$ where A_i is one of the two sets

$$A_1 = \left\{ (t,x) \in S_{\psi,0,25} : t \in \left[\frac{27\pi}{5}, \frac{28\pi}{5} \right] \right\}, \tag{4.1}$$

$$A_2 = \left\{ (t,x) \in S_{\psi,0,25} : t \in \left[\frac{9\pi}{2}, \frac{13\pi}{2} \right] \right\}. \tag{4.2}$$

Note that the center of these sets coincide, but that A_2 is considerably larger than A_1. For $\Omega_{\psi,0,25}$ and g as above we have estimated the solution $u(t,x)$ at $(t,x) = (0,5)$. To obtain 'correct' values of the solutions for the two choices of A_i, we conducted a simulation with the Gobet type algorithm (CH), using $M = 10^8$ and $N = 10^4$. The resulting reference values were 0.00335 and 0.08246 respectively.

Example 4.3 (Problem (P2) with smooth data). In this example we consider problem (P2) with $\psi(t) = \sqrt{|t - 12.5|}$ and $T = 25$. One can prove, by a straightforward calculation, that $\Omega_{\psi,0,25}$ is a time-dependent domain in the sense of Definition 2.1 for this choice of ψ. We let the boundary data g be given by $g(t,x) = \sin(t)$ whenever $(t,x) \in S_{\psi,0,25}$. For $\Omega_{\psi,0,25}$ and g as above we have estimated the solution $u(t,x)$ at $(t,x) = (0,8)$. To obtain 'correct' values of the solutions we calculated $M = 10^8$ trajectories with the Gobet type algorithm (CH) for $N = 20001$ and obtained the result -0.0038.

Example 4.4 (Problem (P3) with smooth data). In this example we consider problem (P3) based on $\psi(t) = \sin(mt)$, for some positive real number m. We let $T = 25$ and let the boundary data g be given by $g(t,x) = \sin(mt)$ whenever $(t,x) \in S_{0,0,25}$. We have estimated the solution $u(t,x)$ at $(t,x) = (0,5)$ in the cases $m = 1$ and $m = 5$. By construction we can use the same reference values as in Example 4.1.

Example 4.5 (Problem (P3) with non-smooth data). In this example we consider problem (P3) based on $\psi(t) = \sin(t)$. We let $T = 25$ and let the non-smooth boundary data g be given by $g(t,x) = \chi_{A_i}(t,x)$ where A_i is one of the two sets

$$A_1 = \left\{ (t,x) \in S_{0,0,25} : t \in \left[\frac{27\pi}{5}, \frac{28\pi}{5} \right] \right\},$$

$$A_2 = \left\{ (t,x) \in S_{0,0,25} : t \in \left[\frac{9\pi}{2}, \frac{13\pi}{2} \right] \right\}.$$

We have estimated the solution $u(t,x)$ at $(t,x) = (0,5)$. By construction we can use the same reference values as in Example 4.2.

Example 4.6 (Problem (P4) with smooth data). In this example we consider problem (P4) based on $\psi(t) = \sqrt{|t - 12.5|}$. We let $\epsilon = 0$, $T = 0$ and let the boundary data g be given by $g(t, x) = \sin(t)$ whenever $(t, x) \in S_{0,0,25}$. We have estimated the solution $u(t, x)$ at $(t, x) = (0, 8)$ and, by construction, we can use the same reference value as in Example 4.3. Note that in the adaptive algorithm for (P4) we need the time derivative of $P_{\gamma x}\psi(t)$ (see (3.29)), which for $\varepsilon = 0$ and $t = 12.5$ cannot be defined. Nevertheless, due to the symmetry of $P(t)$ and $\psi(t)$, this derivative is zero for all $\varepsilon > 0$ and hence it is reasonable to use $\partial_t (P_{\gamma x}\psi(12.5)) = 0$ in the adaptive algorithm.

4.1. Results for problem (P1)–(P2): Examples 4.1–4.3

Figure 1 shows convergence results for Example 4.1 in the cases $m = 1$ and $m = 5$. The results are based on $M = 2 \cdot 10^7$ realizations, which corresponds to a statistical error of approximately $2 \cdot 10^{-4}$. For the Euler algorithm, we find that the order of convergence on average is, as expected, close to 0.5 but that the result is unstable, especially in the case $m = 5$. This instability arises as the estimated value of $u(0, 5)$ oscillates around the correct value. As N is increased, this oscillation causes the illusion of high, and increasing, order of convergence during some interval $\left[N_0, N_{\text{peak}}\right]$ followed by an interval $\left[N_{\text{peak}}, N_1\right]$ of low, and even negative, order of convergence. Oscillations around the true value occur for both choices of m but, naturally, the value of N_{peak} is not the same in the two cases. Note that this is a possible explanation for the observed exponential order of convergence in Figure 4.7 in [3], that was also discussed in the introduction.

For N larger than some threshold (around 50 in the case $m = 1$ and around 250 in the case $m = 5$) all Gobet type algorithms are stable and have orders of convergence close to 1. The threshold value of N is five times higher in the case $m = 5$ and this is no surprise as the variation of the boundary and the boundary data is five times as high in this case as compared to the case $m = 1$. Although the orders of convergence are the same for all Gobet type algorithms, it should be noted that the TP and CH versions generally perform better than the STP version. The Gobet FET algorithm avoids the instability occurring for the Gobet type algorithms for small N and, at least for moderate values of N, its order of convergence exceeds 1. Also, the results indicate that, asymptotically, the performance of the Gobet FET algorithm is similar to that of the Gobet type algorithms. This is to be expected as the benefit of using information about the density of the first exit time should decrease as the size of the time step decreases. Note, however, that we have used the same tolerance in the numerical integration of (3.12) regardless of N, so it is indeed possible that a high order of convergence can be achieved also for large N by gradually decreasing the tolerance in the numerical integration.

The best performance is, indisputably, displayed by the adaptive algorithm, whose order of convergence by far exceeds the orders of all other algorithms. It remains an

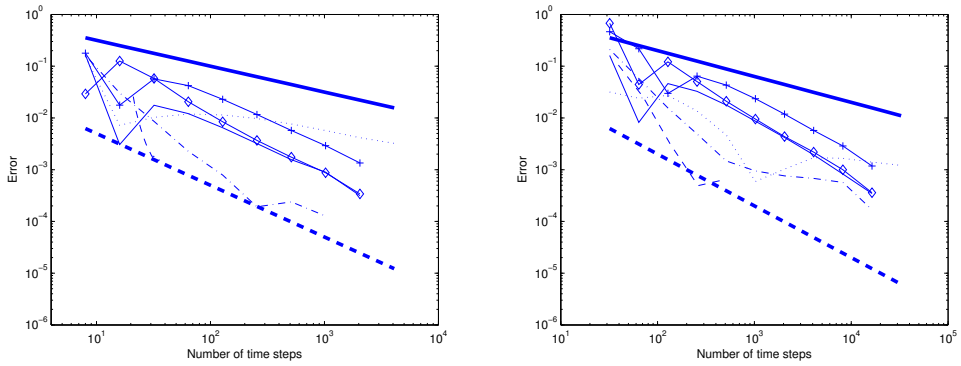

Figure 1. Loglog plots of the error in Example 4.1 as a function of the number of time steps. The left plot shows the results in the case $m = 1$ and the right plot the results in the case $m = 5$. Legend: dot, Euler algorithm; solid, Gobet type algorithms (plus, STP; diamond, TP; none CH); dash dot, Gobet FET algorithm; dash, Adaptive algorithm; solid thick, Reference line with slope 0.5; dash thick, Reference line with slope 1.

open question to what extent this high order of convergence remains for larger values of N as well and the fast convergence may be the result of oscillations around the true value as described above. Note also that the error expansions used in the adaptive algorithm only approximate error due to variation of the coefficients of the stochastic differential equation, whereas error due to large deviations in the data and boundary is unaccounted for. Hence one cannot be certain that the error produced by the adaptive algorithm is within the given tolerance and, in effect, the performance of the adaptive algorithm is determined by the choice of initial time discretization rather than by the choice of tolerance.

Figure 2 contains a more thorough analysis of the relation between the approximate value and the number of time steps. Here all integers values of N in the interval $[16, 256]$ have been used to generate estimates of $u(0, 5)$. We see that the estimated values do not vary much between subsequent choices of N, but also that the instabilities are larger in the case $m = 5$. Note that the seemingly good estimate of the Euler algorithm in the case $m = 5$ is caused by an oscillation around the correct value with $N_{\text{peak}} \approx 2^{10}$ (cf. Figure 1).

Figure 3 shows convergence results for Example 4.2 with A_1 and A_2 as in (4.1)–(4.2). The results are based on $M = 2 \cdot 10^7$ realizations, which corresponds to statistical errors of approximately $2.5 \cdot 10^{-5}$ and $1.2 \cdot 10^{-4}$ respectively. Although A_2 is considerably larger than A_1, the results are very much alike. For the Euler algorithm, the Gobet type algorithms and the adaptive algorithm, the convergence results are highly unstable. These instabilities arise as the estimates of the exit probability strongly depend on the number of time discretization nodes that are found within the

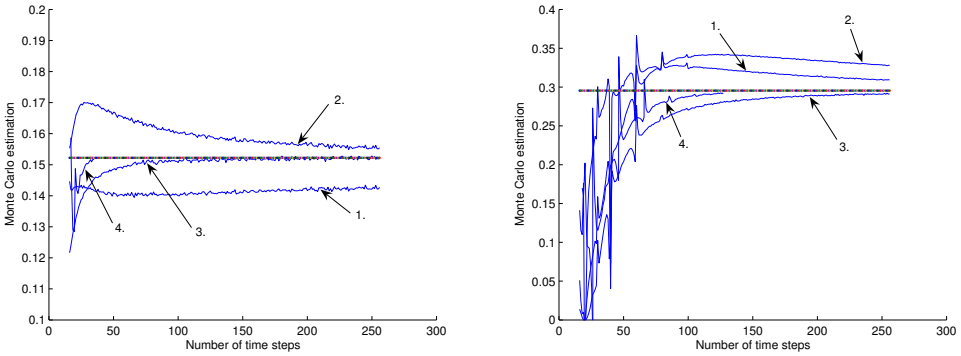

Figure 2. Estimated value of $u\,(0,5)$ in Example 4.1 as a function of the number of time steps. The left plot shows the results in the case $m = 1$ and the right plot the results in the case $m = 5$. Legend: 1. Euler algorithm; 2. Gobet type algorithm (CH); 3. Gobet FET algorithm; 4. Adaptive algorithm; solid, Correct value.

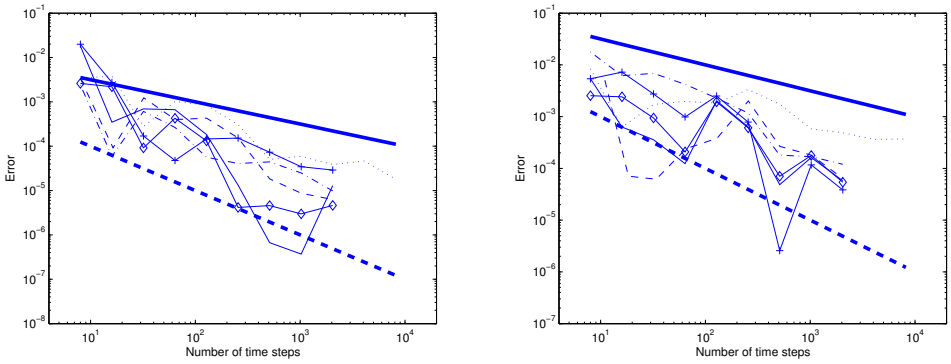

Figure 3. Loglog plots of the error in Example 4.2 as a function of the number of time steps. The left plot shows the results in the A_1 case and the right plot the results in the A_2 case. Legend: dot, Euler algorithm; solid, Gobet type algorithms (plus, STP; diamond, TP; none CH); dash dot, Gobet FET algorithm; dash, Adaptive algorithm; solid thick, Reference line with slope 0.5; dash thick, Reference line with slope 1.

time interval corresponding to the set A_i. For a uniform mesh in the A_1 case, the number of discretization nodes within A_1 is of the order $N/40$, resulting in a relative error of the order $40/N$. As an example, for $N = 1000$ the difference between estimates of $u(0,5)$ for subsequent choices of N is of the order of 4%, which is three times the statistical error in the plots in Figure 3. The instabilities make it almost impossible to estimate the order of convergence of these algorithms. The Gobet FET algorithm takes the variation of the boundary data during each time interval into account and, hence, it is much more stable than the other algorithms. Least square fits give orders of convergence around 0.9 in the A_1 and A_2 cases, so this algorithm retains the theoretical order of convergence for the Gobet type algorithm even for the non-smooth data considered in this example.

To investigate the stability further we have, as in Example 4.1, calculated the approximate value for all integer values of N in the interval $[16, 256]$. The results in the A_1 case can be found in Figure 4 (the results in the A_2 case are similar and have been left out). The variations between subsequent choices of N are quite large, especially for the Euler algorithm. Even when smoothing over as much as 50 subsequent choices of N, the Euler algorithm displays a high level of instability. Nevertheless, the smoothed values of the Euler algorithm converges faster than any other algorithm, so it appear as though there is a trade-off between fast convergence and stability in this case.

We conclude this section by considering Example 4.3. The left plot in Figure 5 shows convergence results based on $M = 2 \cdot 10^7$ realizations, which corresponds to a statistical error of approximately $2 \cdot 10^{-4}$. As could be expected, the orders of convergence of the Euler and Gobet type algorithms are close to 0.5 and 1, respectively. Note, however, that the Gobet type algorithms are fairly unstable up to approximately $N = 100$ with the consequence that for $N < 500$ the performance of the Euler algorithm is in fact better than that of the Gobet type algorithms. Note also that the results of the previous examples indicate that the TP and CH versions perform similarly and hence it should be a minor drawback that the TP version cannot be defined for problem (P2). The best performance is found for the Gobet FET algorithm, whose order of convergence is at least one. Furthermore, in contrast to the Gobet type algorithms, the Gobet FET algorithm is stable for small N as well. The order of convergence of the adaptive algorithm is near one, but convergence can only be obtained by choosing finer and finer initial time discretizations and not, as desired, by decreasing the tolerance.

4.2. Results for problem (P3)–(P4): Examples 4.4–4.6

Figure 6 shows convergence results for Example 4.4. As in Example 4.1, the order of convergence of the Euler algorithm is 0.5 on average, but the result is severely unstable due to oscillations around the correct value. The results of the Gobet type algorithm and the Gobet FET algorithm are similar and both algorithms have orders of convergence close to 1. For the adaptive algorithm the order of convergence, in

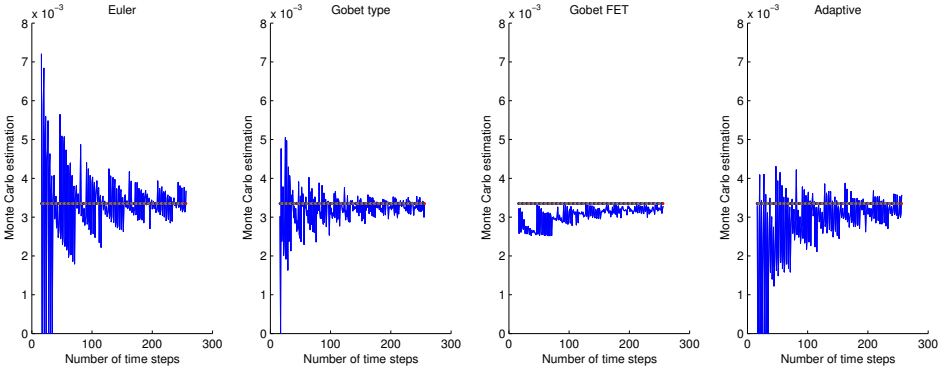

Figure 4. Estimated value of $u(0,5)$ in Example 4.2 as a function of the number of time steps. The plots show results in the A_1 case. From left to right the plots correspond to the Euler algorithm, the Gobet type algorithm (CH), the Gobet FET algorithm and the adaptive algorithm.

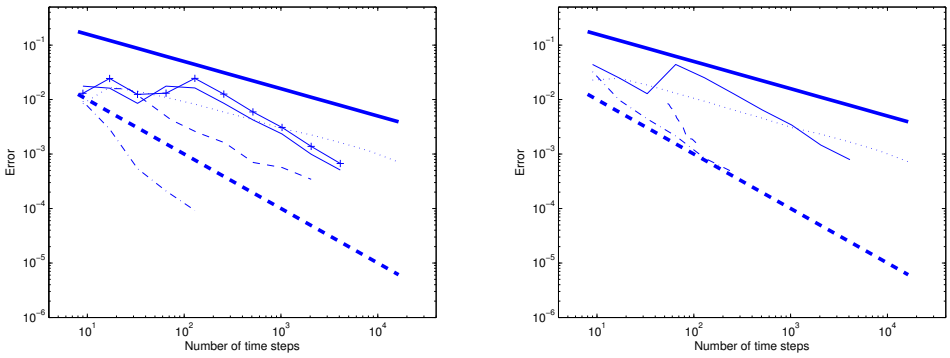

Figure 5. Loglog plots of the error in Examples 4.3 (left) and 4.6 (right) as a function of the number of time steps. Legend: dot, Euler algorithm; solid, Gobet type algorithms (plus, STP; none CH); dash dot, Gobet FET algorithm; dash, Adaptive algorithm; solid thick, Reference line with slope 0.5; dash thick, Reference line with slope 1.

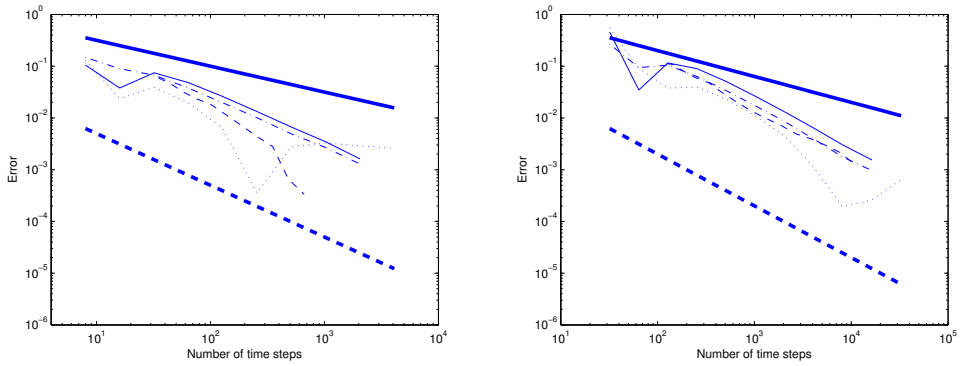

Figure 6. Loglog plots of the error in Example 4.4 as a function of the number of time steps. The left plot shows the results in the case $m = 1$ and the right plot the results in the case $m = 5$. Legend: dot, Euler algorithm; solid, Gobet type algorithm; dash dot, Gobet FET algorithm; dash, Adaptive algorithm; solid thick, Reference line with slope 0.5; dash thick, Reference line with slope 1.

the case $m = 1$, is approximately 1 for small N and then gradually increases as N is increased. This behaviour is not observed for $m = 5$, where the order of convergence remains close to 1 for all N. It is likely that the phenomenon of increasing order of convergence occurs for $m = 5$ as well but for values of N that are larger than those investigated here. Note that the adaptive algorithm is sensitive to the initial choice of time discretization. If the initial time discretization is too coarse the convergence may fail or be very slow. In Figure 6 we have used $N_{\text{init}} = 20$ and $N_{\text{init}} = 100$, respectively, in the two cases.

Comparing Figures 3 and 7, it is clear that the results in Examples 4.2 and 4.5 are similar as all algorithms suffer from severe instabilities. The most stable performance is found for the Gobet FET algorithm, whose order of convergence is approximately 1.

The right plot in Figure 5 shows convergence results in Example 4.5. The orders of convergence of the Euler and Gobet type algorithms are 0.5 and 1, respectively. The Gobet FET algorithm has order of convergence 1 and avoids the instability that occurs for the Gobet type algorithm for small N. For sufficiently fine initial time discretizations ($N_{\text{init}} = 51$ was used in Figure 5), the order of convergence of the adaptive algorithm exceeds 1, but, on the other hand, if N_{init} is too small then the convergence fails.

4.3. Computational cost

Up to this point we have compared the investigated algorithms with respect to stability and order of convergence. Another important factor that must be taken into account when choosing the appropriate numerical algorithm, for a given problem, is the com-

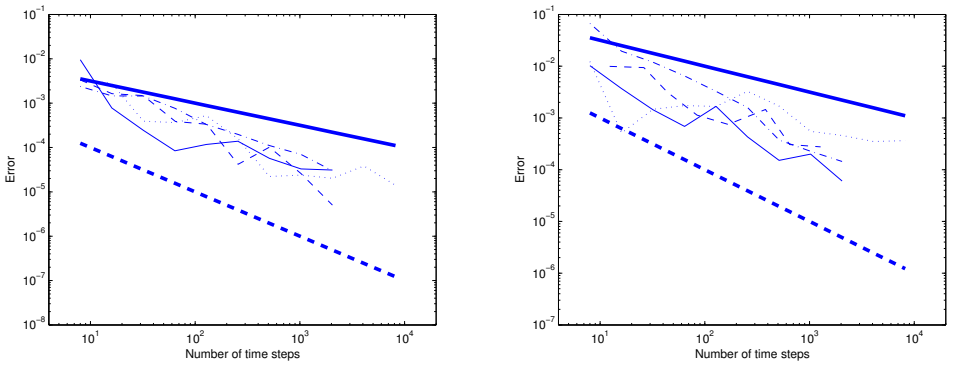

Figure 7. Loglog plots of the error in Example 4.5 as a function of the number of time steps. The left plot shows the results in the A_1 case and the right plot the results in the A_2 case. Legend: dot, Euler algorithm; solid, Gobet type algorithm; dash dot, Gobet FET algorithm; dash, Adaptive algorithm; solid thick, Reference line with slope 0.5; dash thick, Reference line with slope 1.

putational cost. For the non-adaptive algorithms, there is no significant difference in the computational cost of solving a given problem in the setting of (P1)–(P2) as compared to the same problem in the setting of (P3)–(P4). Hence, for the remainder of this section we only consider problems (P1)–(P2) for the non-adaptive algorithms.

Figure 8 shows the computational costs in Examples 4.1–4.6 and, as can be seen, the pattern does not vary much from problem to problem. For small N ($N < 100$) the computational costs of the Gobet FET algorithm and the adaptive algorithm are similar and considerably higher than for the Euler and Gobet type algorithms, for which the computational costs, in turn, are similar. For large N ($N > 1000$) the computational cost of the adaptive algorithm is significantly larger than for all other algorithms, which, in turn, have similar computational costs.

The computational cost naturally depends very strongly on the way in which a certain algorithm is implemented and, particularly for the Gobet FET algorithm and the adaptive algorithm, the implementation can be carried out in many different ways. It is, however, reasonable to expect that the overall pattern of Figure 8 remains the same regardless of how the algorithms are implemented. In this context one should also take into account that a drawback of the adaptive algorithm, and partly also of the Gobet FET algorithm, is that the high complexity of the algorithm makes the process of implementing the algorithm quite time-consuming as compared to implementing the Euler or Gobet type algorithm.

The computational cost of the Gobet FET algorithm in Examples 4.2 and 4.5 poses a minor exception to what holds for the same algorithm in the other examples. In general, the Gobet FET algorithm requires that a numerical integration is carried out

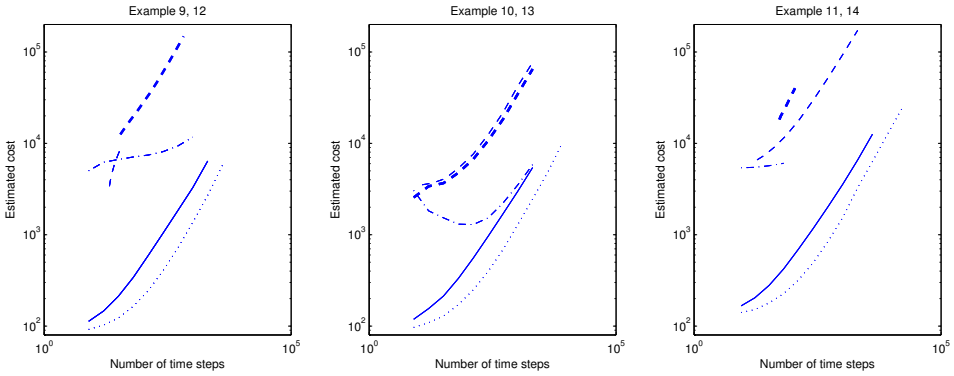

Figure 8. Loglog plots of the computational cost (corresponding to $M = 2 \cdot 10^7$) as a function of the number of time steps. From left to right the plots correspond to Examples 4.1, 4.4 for $m = 1$, Examples 4.2, 4.5 with A_1 and Examples 4.3, 4.6. Legend: dot, Euler algorithm; solid, Gobet type algorithm (CH); dash dot, Gobet FET algorithm; dash thin, Adaptive algorithm (P1)/(P2); dash thick, Adaptive algorithm (P3)/(P4).

every time a trajectory exits the domain during a time step with non-constant data. In Examples 4.2 and 4.5, the data is constant during all except, at most, two time steps and this reduces the number of numerical integrations, and consequently the computational cost, considerably. Hence, in this case, the computational cost of the Gobet FET algorithm is of the same order as for the Gobet type algorithms also for moderate N.

Figure 9 shows the relation between the errors and the computational costs of the algorithms and visualizes the efficiency of the algorithms. Disregarding Examples 4.2 and 4.5, where the results are inconclusive due to the instability of the algorithms, we see that if the required precision is low the Euler and Gobet type algorithms are the most efficient algorithms, whereas if a very small error is required the Gobet FET algorithm is the most efficient algorithm. Figure 9 clearly shows that there exists no shortcut to obtaining a low error with a low computational cost, as the algorithms that display a very high order of convergence also are quite time-consuming.

5. Conclusions

Based on the numerical results in Sections 4.1 and 4.2 a number of conclusions can be drawn. Asymptotically, the Euler and Gobet type algorithms retain their theoretical orders of convergence even in the non-smooth settings considered in this article. Note also that, in general, the CH and TP half space approximations provide better results than the STP half space approximation. However, the Euler and Gobet type algorithms

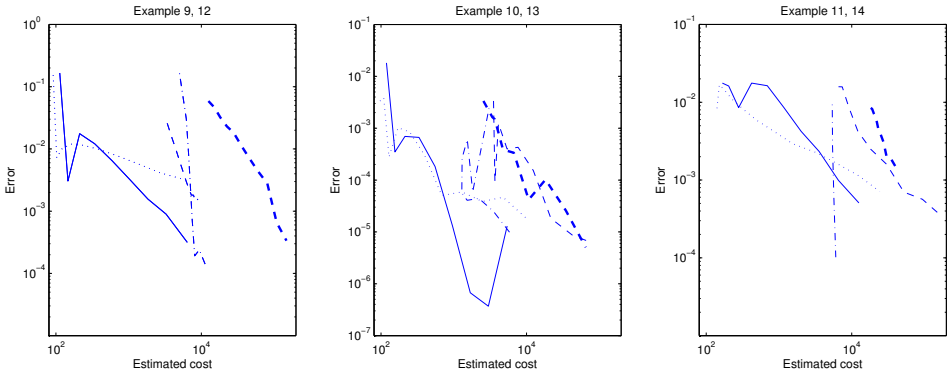

Figure 9. Loglog plots of the error as a function of the computational cost (corresponding to $M = 2 \cdot 10^7$). From left to right the plots correspond to Examples 4.1, 4.4 for $m = 1$, Examples 4.2, 4.5 with A_1 and Examples 4.3, 4.6. Legend: dot, Euler algorithm; solid, Gobet type algorithm (CH); dash dot, Gobet FET algorithm; dash thin, Adaptive algorithm (P1)/(P2); dash thick, Adaptive algorithm (P3)/(P4).

are quite sensitive to where the time discretization nodes are located and the asymptotic order of convergence may not be reached until N is very large. In addition the estimates of the Euler algorithm tend to oscillate around the correct value, giving rise to additional instability. On the other hand the computational costs of the Euler and Gobet type algorithms are low and, as a consequence, these algorithms are the most efficient ones for high tolerances. To conclude, the Euler and Gobet type algorithms are efficient but not very reliable.

In all cases studied here, the Gobet FET algorithm as defined in problems (P1)–(P2) is superior to the other non-adaptive algorithms in terms of both stability and order of convergence. The benefit of the Gobet FET algorithm is particularly large for the non-smooth boundary data considered in Example 4.2, as this algorithm is able to capture the rapid changes in the data better than any other algorithm. Note, however, that some of the results (in particular Figure 1) indicate that, asymptotically, the results of the Gobet type algorithms and the Gobet FET algorithms coincide and, in addition, that the Gobet FET algorithm, due to its high computational cost, is the most efficient algorithm for very small tolerances. Nevertheless the Gobet FET algorithm is the most stable algorithm for all problems considered in this article.

The adaptive algorithm works well when the data is smooth and the convergence is astonishingly fast in the case of smooth data on a smooth boundary (Figure 1). Furthermore, the adaptive algorithm is the only algorithm that obtains asymptotical orders of convergence exceeding one. On the other hand the adaptive algorithm is poor at capturing discontinuities in the boundary data. This is likely to be due to the fact that the adaptive algorithm never refines the interval, on which the trajectory exits

the domain. When the data of the boundary changes rapidly, it is vital to get a good estimate of exactly where the exit occurs.

Comparing the performance of a given algorithm as defined on problems (P1)–(P2) to the same algorithm as defined on problems (P3)–(P4), we see that the error for a given number of time steps is, almost without exception, smaller in the first setting. This is the main drawback of solving a certain problem in the setting of problems (P3)–(P4) as compared to the setting of problems (P1)–(P2). Nevertheless, as far as the adaptive algorithm is concerned, it may still be wise to work in the setting of problems (P3)–(P4) as we can then, with high probability, limit the error by a given tolerance, whereas, as described in Section 4.1, this may not be the case in the setting of problems (P1)–(P2). However, the problem of the adaptive algorithm being sensitive to the choice of initial time discretization still remains in the setting of problem (P3)–(P4).

References

1. C. Costantini, B. Pacchiarotti, and F. Sartoretto. Numerical approximation for functionals of reflecting diffusion processes. *SIAM Journal of Applied Mathematics* **58** (1998), 73–102.

2. R. Durrett. *Probability: Theory and Examples*. Wadsworth Pub. Co., Pacific Grove, CA, 2nd edition, 1995.

3. A. Dzougoutov, K.-S. Moon, E. von Schwerin, A. Szepessy, and R. Tempone. Adaptive Monte Carlo algorithms for stopped diffusion. *Multiscale methods in science and engineering*, Lecture Notes in Computational Science and Engineering 44, 59–88, 2005.

4. E. Gobet. Euler schemes and half-space approximation for the simulation of diffusion in a domain. *ESAIM: Probability and Statistics* **5** (2001), 261–297.

5. S. Hofmann and J. L. Lewis. L^2 solvability and representation by caloric layer potentials in time-varying domains. *Annals of Mathematics* **144**:2 (1996), 349–420.

6. C. E. Kenig. *Harmonic Analysis Techniques for Second Order Elliptic Boundary Value Problems*. CBMS series in Mathematics, no. 83, 1994.

7. P. E. Kloeden and E. Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 1992.

8. J. L. Lewis and M. A. M. Murray. *The Method of Layer Potentials for the Heat Equation in Time-Varying Domains*. Memoir of the A.M.S., no 545, 1993.

9. K.-S. Moon, A. Szepessy, R. Tempone, and G. E. Zouraris. Convergence rates for adaptive weak approximation of stochastic differential equations. *Stochastic Analysis and Applications* **23** (2005), 511–558.

10. P. Salminen. On the first hitting time and the last exit time for a Brownian motion to/from a moving boundary. *Advances in Applied Probability* **20**:2 (1998), 411–426.

11. A. Szepessy, R. Tempone, and G. E. Zouraris. Adaptive weak approximation of stochastic differential equations. *Communications of Pure and Applied Mathematics* **54** (2001), 1169–1214.

Received October 10, 2008; revised February 09, 2009

Author information

Kaj Nyström, Department of Mathematics and Mathematical Statistics, Umeå University, SE-901 87 Umeå, Sweden.
Email: kaj.nystrom@math.umu.se

Thomas Önskog, Department of Mathematics and Mathematical Statistics, Umeå University, SE-901 87 Umeå, Sweden.
Email: thomas.onskog@math.umu.se