

ADAPTIVE MONTE CARLO ALGORITHMS FOR STOPPED DIFFUSION

KYOUNG-SOOK MOON*

Abstract. We present adaptive algorithms for weak approximation of stopped diffusion using the Monte Carlo Euler method. The goal is to compute an expected value $E[g(X(\tau), \tau)]$ of a given function g depending on the solution X of an Itô stochastic differential equation and on the first exit time τ from a given domain. The adaptive algorithms are based on an extension of an a posteriori error expansion, for the approximation of $E[g(X(T))]$ with a fixed final time $T > 0$, introduced in [Szepessy, Tempone and Zouraris, Comm. Pure and Appl. Math., 54, 1169-1214, 2001] using stochastic flows and dual backward solutions. The main steps in the extension are to use a conditional probability to estimate the first exit time error and introduce difference quotients to approximate the initial data of the dual solutions. Numerical results show that the adaptive algorithms achieve the time discretization error of order N^{-1} with N adaptive time steps, while the error is of order $N^{-1/2}$ for a method with N uniform time steps.

Key words. adaptive mesh refinement algorithm, diffusion with boundary, barrier option, Monte Carlo method, weak approximation

AMS subject classifications. 65C30, 60H35, 60J60

1. Introduction. In this paper, we compute adaptive approximations of an expected value

$$E[g(X(\tau), \tau)] \tag{1.1}$$

of a given function, $g : D \times [0, T] \rightarrow \mathbb{R}$, where the stochastic process X solves an Itô stochastic differential equation (SDE)

$$dX_k(t) = a_k(X(t))dt + \sum_{l=1}^{l_0} b_k^l(X(t))dW^l(t), \quad k = 1, 2, \dots, d, \quad t > 0 \tag{1.2}$$

and τ is the first exit time

$$\tau := \inf\{0 < t \leq T : X(t) \notin D\} \tag{1.3}$$

from a given domain $D \subset \mathbb{R}^d$. The functions $a : \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $b^l : \mathbb{R}^d \rightarrow \mathbb{R}^d$ for $l = 1, 2, \dots, l_0$, are given drift and diffusion fluxes and $W^l(t; \omega)$ for $l = 1, 2, \dots, l_0$, are independent Wiener processes. Such problems arise in physics and finance, for instance when computing the value of barrier options.

In the case when the dimension of the problem is large or when the related partial differential equation is difficult to formulate or to solve, the Monte Carlo Euler method is used to compute the expected value. The main difficulty in the approximation of the stopped (or killed) diffusion on the boundary ∂D is that a continuous sample path may exit the given domain D even though a discrete approximate solution does not cross the boundary of D . This hitting of the boundary makes the time discretization error $N^{-1/2}$ for the Monte Carlo Euler method with N uniform time steps, see [6], while the discretization error is of order N^{-1} without boundary in $\mathbb{R}^d \times [0, T]$. The work [11] and [8] reduce the large $N^{-1/2}$ first exit error to N^{-1} . The idea is to

*Institutionen för Numerisk Analys och Datalogi, Kungl. Tekniska Högskolan, SE-100 44 Stockholm, Sweden (moon@nada.kth.se).

generate a uniformly distributed random variable in $(0, 1)$ for each time step and compare it with a known exit probability to decide if the continuous path exits the domain during this time interval. A similar method with N uniform time steps in a domain with smooth boundary is proven to converge with the rate N^{-1} under some appropriate assumptions in [7]. Different Monte Carlo methods for stopped diffusions are compared computationally in [4]. To use these methods, the exit probability needs to be computed accurately.

Inspired by Petersen and Buchmann [14], this work uses the alternative to reduce the computational error by choosing adaptively the size of the time steps near the boundary, which has the advantage that the exit probability does not need to be computed accurately. Section 2 derives an error estimate in a posteriori form. Section 3 presents an adaptive algorithm based on the error estimate where the time discretization error is of order N^{-1} with N adaptive time steps.

Using the Monte Carlo Euler method, the expected value (1.1) can be approximated by a sample average of $g(\bar{X}(\bar{\tau}), \bar{\tau})$, where $(\bar{X}, \bar{\tau})$ is an Euler approximation of (X, τ) . The global error can then be split into time discretization error and statistical error,

$$\begin{aligned} & E[g(X(\tau), \tau)] - \frac{1}{M} \sum_{j=1}^M g(\bar{X}(\bar{\tau}; \omega_j), \bar{\tau}) \\ &= (E[g(X(\tau), \tau) - g(\bar{X}(\bar{\tau}), \bar{\tau})]) + \left(E[g(\bar{X}(\bar{\tau}), \bar{\tau})] - \frac{1}{M} \sum_{j=1}^M g(\bar{X}(\bar{\tau}; \omega_j), \bar{\tau}) \right) \\ &=: \mathcal{E}_T + \mathcal{E}_S \end{aligned} \tag{1.4}$$

where M is the number of realizations. The statistical error, \mathcal{E}_S in (1.4), is asymptotically bounded by $c_0 \bar{\sigma} / \sqrt{M}$ using the Central Limit Theorem, where $\bar{\sigma}$ is the sample average of the standard deviation of $g(\bar{X}(\bar{\tau}), \bar{\tau})$ and c_0 is a positive constant for a confidence interval, see Section 3.1.

Talay and Tubaro [16] proves an a priori error expansion of $E[g(X(T)) - g(\bar{X}(T))]$ for the case without boundary, i.e. in $\mathbb{R}^d \times [0, T]$. Bally and Talay [3] extends this error expansion to non-smooth functions using the Malliavin calculus. The work [15] by Szepessy, Tempone and Zouraris proves an a posteriori error estimate using an error density which depends on computable solutions and discrete dual solutions. In Section 2, we approximate the time discretization error, \mathcal{E}_T in (1.4), in a posteriori form by extending the error estimate in [15] to weak approximation of stopped diffusion. As in [16] and [15], the first step to derive an error estimate is to introduce a continuous Euler path. Then the error between the exact and continuous Euler path is approximated using stochastic flows and dual backward solutions in Section 2.3. The main idea in this extension is to use difference quotients to replace the stochastic flows that do not exist at the boundary. The approximate error between the continuous and discrete Euler path is derived by a conditional probability using Brownian bridges in Section 2.2. Note that the exit probability is used here only to decide the time steps, not to approximate the expected values directly. Therefore the accuracy of the approximation of the exit probability is not crucial.

The Monte Carlo method is powerful for high dimensional problems. Nevertheless, since a one dimensional problem already contains the main difficulties, as a first step, we consider here a one dimensional SDE driven by a one dimensional Wiener process, i.e. $d = l_o = 1$ in (1.2). We study the high dimensional stopped diffusion problems in

[12].

The paper is organized as follows. The error estimate in a posteriori form for stopped diffusions is derived in the next section and based on this error estimate we develop adaptive algorithms in Section 3. Finally some numerical results are given in Section 4 and they provide experimental support for our analytical estimates.

2. A Posteriori Error Expansion. Let us consider an interval $D \subset \mathbb{R}$ and assume that the initial value $X(0) = X_0$ lies in D . The goal is to compute the expected value $E[g(X(\tau), \tau)]$ of a given function g which depends on the stochastic process X and the first exit time τ defined in (1.3).

Let us first discretize the time interval $[0, T]$ into N subintervals $0 = t_0 < t_1 < \dots < t_N = T$ and denote \bar{X} to be the Euler approximation of the process X , i.e. start with $\bar{X}(0) = X_0$ and compute \bar{X}_{n+1} by

$$\bar{X}(t_{n+1}) = \bar{X}(t_n) + a(\bar{X}(t_n))\Delta t_n + b(\bar{X}(t_n))\Delta W_n \quad (2.1)$$

where $\Delta t_n := t_{n+1} - t_n$ denote the time increments and $\Delta W_n := W(t_{n+1}) - W(t_n)$ denote the Wiener increments for $n = 0, 1, \dots, N-1$. Let the approximation of the first exit time τ be

$$\bar{\tau} := \inf_{1 \leq n \leq N} \{t_n : \bar{X}(t_n) \notin D\} \quad (2.2)$$

using the Euler approximation path \bar{X} instead of the exact path X .

Now let us introduce a continuous Euler scheme $\bar{X}(t)$ by

$$\bar{X}(t) = \bar{X}(t_n) + \int_{t_n}^t a(\bar{X}(t_n))dt + \int_{t_n}^t b(\bar{X}(t_n))dW_t \quad (2.3)$$

for $t \in [t_n, t_{n+1}]$ and denote by

$$\tilde{\tau} := \inf\{0 < t \leq T : \bar{X}(t) \notin D\} \quad (2.4)$$

the exit time of the continuous Euler path.

Then the time discretization error of Euler approximation can be split in two parts:

$$\begin{aligned} & E[g(X(\tau), \tau) - g(\bar{X}(\bar{\tau}), \bar{\tau})] \\ &= E[g(X(\tau), \tau) - g(\bar{X}(\tilde{\tau}), \tilde{\tau})] + E[g(\bar{X}(\tilde{\tau}), \tilde{\tau}) - g(\bar{X}(\bar{\tau}), \bar{\tau})] \\ &=: \mathcal{E}_C + \mathcal{E}_D. \end{aligned} \quad (2.5)$$

In [6], Gobet proves the following a priori error estimate for a measurable function \check{f} with N uniform time steps:

$$E[\check{f}(X(\tau), \tau) - \check{f}(\bar{X}(\bar{\tau}), \bar{\tau})] = \mathcal{O}(N^{-1/2}). \quad (2.6)$$

In order to improve the convergence rate in (2.6), we adaptively refine the mesh according to the a posteriori error estimates. The error estimates for \mathcal{E}_D and \mathcal{E}_C are derived in Theorem 2.1 and Theorem 2.2 respectively.

2.1. Notation. To improve the readability, we use $X_t := X(t)$ and $\bar{X}_t := \bar{X}(t)$ for the continuous cases and $\bar{X}_n := \bar{X}(t_n)$ for the discrete case. Let the piecewise constant mesh function Δt be defined by

$$\Delta t(s) := \Delta t_n \text{ for } s \in [t_n, t_{n+1}) \text{ and } n = 0, 1, \dots, N-1 \quad (2.7)$$

and

$$\Delta t_{\max} := \max_{n, \omega} \Delta t_n(\omega).$$

Let $\mathbf{1}_{y \in A}$ denote the indicator function, i.e. $\mathbf{1}_{y \in A} = 1$ if $y \in A$, otherwise $\mathbf{1}_{y \in A} = 0$.

2.2. Expansion of Exiting Error using Probability. Let us consider the time discretization error between the continuous and the discrete Euler path, i.e. \mathcal{E}_D in (2.5). In the case when the continuous Euler path ends at time $t = T$, i.e. $\tilde{\tau} = T = \bar{\tau}$, there is no time discretization error between two paths since $E[g(\bar{X}_{\tilde{\tau}}, \tilde{\tau})\mathbf{1}_{\tilde{\tau}=T}] = E[g(\bar{X}_{\bar{\tau}}, \bar{\tau})\mathbf{1}_{\bar{\tau}=T}]$. On the other hand if the continuous Euler path is stopped at $\tilde{\tau} < T$ then there is a possibility of $\tilde{\tau} < \bar{\tau}$, i.e. the continuous Euler path $\bar{X}(t)$ exits the domain D in an interval $[t_k, t_{k+1}]$, though the discrete values \bar{X}_k and \bar{X}_{k+1} lie inside the domain. Fig. 2.1 shows an illustrative Monte Carlo trajectory when the discrete Euler process \bar{X}_n for $n = 0, \dots, N$ does not exit the domain $D = (-\infty, \lambda)$ until $\bar{\tau} = T$, while the continuous Euler $\bar{X}(t)$ stops at $t = \tilde{\tau} < \bar{\tau}$.

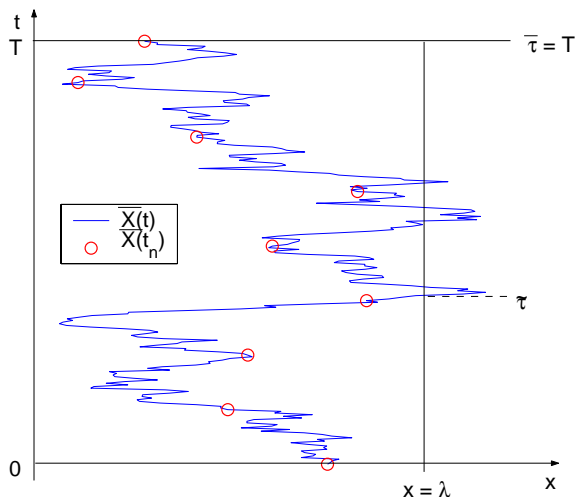


FIG. 2.1. An illustrative Euler Monte Carlo trajectory when $\tilde{\tau} < \bar{\tau}$.

Let $P_{Z,k}$ denote the probability that a Brownian motion $Z(t)$ exits a given domain D at $t \in (t_k, t_{k+1})$ when the values Z_k and Z_{k+1} are in D . In the one dimensional half interval case, $D = (-\infty, \lambda)$ for a constant λ , the probability $P_{Z,k}$ has a simple expression using the law of Brownian bridges conditioned to be in z_2 at time t_{k+1} and starting at z_1 , see [9];

$$P_{Z,k} = \mathbb{P} \left[\max_{t \in [t_k, t_{k+1}]} Z_t \geq \lambda \mid Z_k = z_1, Z_{k+1} = z_2 \right] = \exp \left(-2 \frac{(\lambda - z_1)(\lambda - z_2)}{\sigma^2 \Delta t_k} \right) \quad (2.8)$$

where $z_1 < \lambda$ and $z_2 < \lambda$ and $E[(\Delta Z_k - E[\Delta Z_k])^2] = \sigma^2 \Delta t_k$ with $\Delta Z_k = Z_{k+1} - Z_k$. The process $\bar{X}(t)$ for $t \in (t_n, t_{n+1})$ behaves like the Brownian motion $Z(t)$ with constant drift $a(\bar{X}_n)$ and constant diffusion $b(\bar{X}_n)$. The work [2] studies the estimates for the exit probability of the Brownian bridge in general cases of one dimension, e.g. with time dependent lower and upper boundaries. For a family of nondegenerate SDEs in high dimension, including the half space case, the exit probabilities are expressed as asymptotic series in [5], [1]. For the more general case, we can approximate D locally near the boundary to its tangent half space and use the approximation of the exit probability for the half space case, see [6], [7].

Provided that the approximations of the exit probabilities are known, we have the following error expansion for \mathcal{E}_D :

THEOREM 2.1. *Let $\bar{X}(t)$ and $\bar{X}(t_n)$ be the continuous and discrete Euler approximations defined in (2.3) and (2.1) respectively with bounded functions a, b in $C^6(D)$ and g in $C^6(D \times [0, T])$. Then the time discretization error has the expansion*

$$E[g(\bar{X}_{\tilde{\tau}}, \tilde{\tau}) - g(\bar{X}_{\bar{\tau}}, \bar{\tau})] = E \left[\sum_{n=0}^{N-1} \left(g(\lambda, \frac{1}{2}(t_n + t_{n+1})) - g(\bar{X}_{\bar{\tau}}, \bar{\tau}) + \mathcal{O}(\Delta t_{\max}) \right) \hat{P}_{\bar{X}, n} \right] \quad (2.9)$$

where the conditional exit probabilities $P_{\bar{X}, k}$, from e.g. (2.8), define

$$\hat{P}_{\bar{X}, n} = P_{\bar{X}, n} \prod_{k=0}^{n-1} (1 - P_{\bar{X}, k}), \quad n = 1, 2, \dots, N-1, \quad (2.10)$$

with $\hat{P}_{\bar{X}, 0} = P_{\bar{X}, 0}$ and $\lambda \in \partial D$ is the value of $\bar{X}_{\tilde{\tau}}$.

Proof. Note that the time discretization error satisfies $E[(g(\bar{X}_{\tilde{\tau}}, \tilde{\tau}) - g(\bar{X}_{\bar{\tau}}, \bar{\tau})) \mathbf{1}_{\tilde{\tau}=T}] = 0$ and that $\mathbf{1}_{\tilde{\tau} < T} = \sum_{n=0}^{N-1} \mathbf{1}_{\tilde{\tau} \in [t_n, t_{n+1})}$ to obtain

$$E[g(\bar{X}_{\tilde{\tau}}, \tilde{\tau}) - g(\bar{X}_{\bar{\tau}}, \bar{\tau})] = E \left[\sum_{n=0}^{N-1} (g(\bar{X}_{\tilde{\tau}}, \tilde{\tau}) - g(\bar{X}_{\bar{\tau}}, \bar{\tau})) \mathbf{1}_{\tilde{\tau} \in [t_n, t_{n+1})} \right]. \quad (2.11)$$

Substitute the midpoint approximation of $g(\bar{X}_{\tilde{\tau}}, \tilde{\tau})$, i.e.

$$g(\bar{X}_{\tilde{\tau}}, \tilde{\tau}) \mathbf{1}_{\tilde{\tau} \in [t_n, t_{n+1})} = \left(g(\lambda, \frac{1}{2}(t_n + t_{n+1})) + \mathcal{O}(\Delta t_n) \right) \mathbf{1}_{\tilde{\tau} \in [t_n, t_{n+1})}$$

in (2.11) to get

$$E[g(\bar{X}_{\tilde{\tau}}, \tilde{\tau}) - g(\bar{X}_{\bar{\tau}}, \bar{\tau})] = E \left[\sum_{n=0}^{N-1} \left(g(\lambda, \frac{1}{2}(t_n + t_{n+1})) - g(\bar{X}_{\bar{\tau}}, \bar{\tau}) + \mathcal{O}(\Delta t_{\max}) \right) \mathbf{1}_{\tilde{\tau} \in [t_n, t_{n+1})} \right]$$

Use the σ -algebra χ generated by $\{\bar{X}_n : n = 0, 1, \dots, N\}$ to obtain

$$\begin{aligned} & E[g(\bar{X}_{\tilde{\tau}}, \tilde{\tau}) - g(\bar{X}_{\bar{\tau}}, \bar{\tau})] \\ &= E \left[\sum_{n=0}^{N-1} E \left[\left(g(\lambda, \frac{1}{2}(t_n + t_{n+1})) - g(\bar{X}_{\bar{\tau}}, \bar{\tau}) + \mathcal{O}(\Delta t_{\max}) \right) \mathbf{1}_{\tilde{\tau} \in [t_n, t_{n+1})} \mid \chi \right] \right] \\ &= E \left[\sum_{n=0}^{N-1} \left(g(\lambda, \frac{1}{2}(t_n + t_{n+1})) - g(\bar{X}_{\bar{\tau}}, \bar{\tau}) + \mathcal{O}(\Delta t_{\max}) \right) \mathbb{P}[\tilde{\tau} \in [t_n, t_{n+1}) \mid \chi] \right] \quad (2.12) \end{aligned}$$

If $z_1 \notin D$ and $z_2 \notin D$ then $\mathbb{P}[\tilde{\tau} \in [t_n, t_{n+1}) | \bar{X}_n = z_1, \bar{X}_{n+1} = z_2] = 0$, and if $z_1 \in D$ and $z_2 \notin D$ then $\mathbb{P}[\tilde{\tau} \in [t_n, t_{n+1}) | \bar{X}_n = z_1, \bar{X}_{n+1} = z_2] = 1$. To compute the probability in (2.12), we observe also that for any n the event $\{\tilde{\tau} \in [t_n, t_{n+1})\}$ implies

$$\{\bar{X}_{t \in [t_0, t_1)} \in D, \dots, \bar{X}_{t \in [t_{n-1}, t_n)} \in D, \text{ and } \bar{X}_{t \in [t_n, t_{n+1})} \notin D\}.$$

Thus we obtain using the given values $P_{\bar{X},k}$ for $k = 0, 1, \dots, N-1$,

$$\hat{P}_{\bar{X},n} := \mathbb{P}[\tilde{\tau} \in [t_n, t_{n+1}) | \mathcal{X}] = P_{\bar{X},n} \prod_{k=0}^{n-1} (1 - P_{\bar{X},k})$$

and $\hat{P}_{\bar{X},0} := \mathbb{P}[\tilde{\tau} \in [t_0, t_1) | \bar{X}_0 = z_1, \bar{X}_1 = z_2] = P_{\bar{X},0}$, which together with (2.12) proves (2.9). \square

2.3. Error Expansion using Dual Solutions. In this subsection, we derive an a posteriori error estimate between the exact and the continuous Euler path, i.e. \mathcal{E}_C in (2.5). The main result is stated in Theorem 2.2 and we present the proof of this theorem afterwards.

Let us first define the discrete dual functions $\varphi(t_n)$, $\varphi'(t_n)$ and $\varphi''(t_n)$ as follows. Introduce

$$c(t_n, x) = x + \Delta t_n a(x) + \Delta W_n b(x), \quad \beta(x) = \frac{1}{2} b^2(x),$$

and an Euler approximation $\hat{X}(t_n)$ for $t_n \in [\bar{\tau}, \hat{\tau}]$ with initial value $\hat{X}(\bar{\tau}) = \bar{X}(\bar{\tau}) + \gamma \Delta x$, where $\gamma \in \{+1, -1\}$ denotes an inward unit normal vector and $\hat{\tau}$ denotes the first exit time of \hat{X} , i.e. $\hat{\tau} := \inf\{t_n : \bar{\tau} < t_n \text{ and } \hat{X}_n \notin D\}$. Then the function φ is defined by the dual backward problem

$$\varphi(t_n) = \partial_x c(t_n, \bar{X}_n) \varphi(t_{n+1}), \quad t_n < \bar{\tau}, \quad (2.13)$$

$$\varphi(\bar{\tau}) = \begin{cases} \partial_x g(\bar{X}_{\bar{\tau}}, \bar{\tau}), & \text{if } \bar{\tau} = T, \\ (g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - g(\bar{X}_{\bar{\tau}}, \bar{\tau})) / \Delta x, & \text{if } \bar{\tau} < T, \end{cases} \quad (2.14)$$

its first variation φ' satisfies

$$\varphi'(t_n) = (\partial_x c)^2(t_n, \bar{X}_n) \varphi'(t_{n+1}) + \partial_x^2 c(t_n, \bar{X}_n) \varphi(t_{n+1}), \quad t_n < \bar{\tau}, \quad (2.15)$$

$$\varphi'(\bar{\tau}) = \begin{cases} \partial_x^2 g(\bar{X}_{\bar{\tau}}, \bar{\tau}), & \text{if } \bar{\tau} = T, \\ -(\beta(\bar{X}_{\bar{\tau}}))^{-1} (\partial_t g(\bar{X}_{\bar{\tau}}, \bar{\tau}) + a(\bar{X}_{\bar{\tau}}) \varphi(\bar{\tau})), & \text{if } \bar{\tau} < T, \end{cases} \quad (2.16)$$

and its second variation φ'' satisfies

$$\begin{aligned} \varphi''(t_n) &= (\partial_x c)^3(t_n, \bar{X}_n) \varphi''(t_{n+1}) + \partial_x^3 c(t_n, \bar{X}_n) \varphi(t_{n+1}) \\ &\quad + 3 \partial_x^2 c \partial_x c(t_n, \bar{X}_n) \varphi'(t_{n+1}), \quad t_n < \bar{\tau}, \end{aligned} \quad (2.17)$$

$$\varphi''(\bar{\tau}) = \begin{cases} \partial_x^3 g(\bar{X}_{\bar{\tau}}, \bar{\tau}), & \text{if } \bar{\tau} = T, \\ -(\beta(\bar{X}_{\bar{\tau}}))^{-1} ((\partial_t g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - \partial_t g(\bar{X}_{\bar{\tau}}, \bar{\tau})) / \Delta x \\ + \partial_x a(\bar{X}_{\bar{\tau}}) \varphi(\bar{\tau}) + (a + \partial_x \beta)(\bar{X}_{\bar{\tau}}) \varphi'(\bar{\tau})), & \text{if } \bar{\tau} < T. \end{cases} \quad (2.18)$$

Then the time discretization error \mathcal{E}_C in (2.5) has the following error expansion:

THEOREM 2.2. Let $X(t)$, $\bar{X}(t)$ and $\bar{X}(t_n)$ be the exact, the continuous Euler and the discrete Euler path defined in (1.2), (2.3) and (2.1) respectively. Assume that the functions a, b and g are bounded in $C^6(D)$ and in $C^6(D \times [0, T])$ and that $\beta(x) \geq c > 0$. Then the time discretization error \mathcal{E}_C has the error expansion

$$E[g(X_\tau, \tau) - g(\bar{X}_{\tilde{\tau}}, \tilde{\tau})] = E \left[\sum_{n=0}^{N-1} \mathbf{1}_{t_{n+1} \leq \tilde{\tau}} \rho_n \Delta t_n^2 \right] + \mathcal{O} \left(\Delta x + \sqrt{\Delta t_{\max}} + \frac{\sqrt{\Delta t_{\max}}}{\Delta x} \right) E \left[\sum_{n=0}^{N-1} \Delta t_n^2 \right] \quad (2.19)$$

where Δx is a small positive constant to define ρ_n by (2.13)-(2.18), and

$$\begin{aligned} \rho_n &= \frac{1}{2} \left((\partial_t a + a \partial_x a + \beta \partial_x^2 a)(\bar{X}_n) \varphi(t_{n+1}) \right. \\ &\quad + (2\beta \partial_x a + \partial_t \beta + a \partial_x \beta + \beta \partial_x^2 \beta)(\bar{X}_n) \varphi'(t_{n+1}) \\ &\quad \left. + (2\beta \partial_x \beta)(\bar{X}_n) \varphi''(t_{n+1}) \right). \end{aligned} \quad (2.20)$$

The proof of Theorem 2.2 has several steps and we present them by following three lemmas. Let us first introduce a solution u of the Kolmogorov backward equation

$$\begin{aligned} \partial_t u + a \partial_x u + \beta \partial_x^2 u &= 0, \quad (x, t) \in D \times [0, T] \\ u(x, T) &= g(x, T), \quad x \in D \\ u(\lambda, t) &= g(\lambda, t), \quad (\lambda, t) \in \partial D \times [0, T]. \end{aligned} \quad (2.21)$$

Then by the Feynman-Kac formula u can be represented by the expectation

$$u(x, t) = E[g(X_\tau, \tau) | X(t) = x]. \quad (2.22)$$

Let \bar{a} and \bar{b} be the piecewise constant functions defined by $\bar{a}(t) = a(\bar{X}_n)$ and $\bar{b}(t) = b(\bar{X}_n)$ for $t \in [t_n, t_{n+1})$. Similarly let us define $\bar{\beta} = \frac{1}{2} \bar{b} \bar{b}$. Then the time discretization error \mathcal{E}_C has the following representation :

LEMMA 2.1. Let $X(t)$ and $\bar{X}(t)$ be the exact and the continuous Euler path defined by (1.2) and (2.3) respectively and let the function u be defined by (2.22). Suppose that the assumptions in Theorem 2.2 hold. Then the time discretization error between these two paths has the representation

$$E[g(X_\tau, \tau) - g(\bar{X}_{\tilde{\tau}}, \tilde{\tau})] = E \left[\int_0^{\tilde{\tau}} ((a - \bar{a}) \partial_x u + (\beta - \bar{\beta}) \partial_x^2 u)(\bar{X}_t, t) dt \right]. \quad (2.23)$$

Proof. Apply the Itô formula to the function u in (2.22) to get

$$du(\bar{X}_t, t) = (\partial_t u + \bar{a} \partial_x u + \bar{\beta} \partial_x^2 u)(\bar{X}_t, t) dt + \bar{b} \partial_x u(\bar{X}_t, t) dW_t.$$

Here the definition of the continuous Euler scheme in (2.3) is used, i.e. $d\bar{X}_t = \bar{a} dt + \bar{b} dW_t$ for $t \in [t_n, t_{n+1})$. Integrate both sides from 0 to $\tilde{\tau}$ and take the expectation to obtain

$$\begin{aligned} &E[u(\bar{X}_{\tilde{\tau}}, \tilde{\tau}) - u(\bar{X}_0, 0)] \\ &= E \left[\int_0^{\tilde{\tau}} (\partial_t u + \bar{a} \partial_x u + \bar{\beta} \partial_x^2 u)(\bar{X}_t, t) dt \right] + E \left[\int_0^{\tilde{\tau}} \bar{b} \partial_x u(\bar{X}_t, t) dW_t \right]. \end{aligned} \quad (2.24)$$

Note that the Itô integral in (2.24) is not adapted to the standard filtration generated by W alone. Instead consider the filtration \mathcal{G}_t , the σ -algebra generated by $\{W(s), \Delta t(s) : s \leq t\}$. Then from Lemma 4.2 in [13] the Itô integral in (2.24) is a martingale with respect to \mathcal{G}_t and since $\tilde{\tau}$ is a stopping time, we therefore have

$$E \left[\int_0^{\tilde{\tau}} \bar{b} \partial_x u(\bar{X}_t, t) dW_t \right] = 0.$$

Now apply the definition of u to the left hand side of (2.24), i.e.

$$\begin{aligned} u(\bar{X}_0, 0) &= E[g(X_\tau, \tau) | X_0 = \bar{X}_0] = E[g(X_\tau, \tau)], \\ u(\bar{X}_{\tilde{\tau}}, \tilde{\tau}) &= E[g(X_\tau, \tau) | X_{\tilde{\tau}} = \bar{X}_{\tilde{\tau}}] = E[g(\bar{X}_{\tilde{\tau}}, \tilde{\tau})], \end{aligned}$$

and the Kolmogorov backward equation in (2.21) to the first expectation of the right hand side in (2.24) to conclude (2.23). \square

Using the discrete time steps, the error representation (2.23) can be written by

$$\begin{aligned} &E[g(X_\tau, \tau) - g(\bar{X}_{\tilde{\tau}}, \tilde{\tau})] \\ &= E \left[\sum_{n=0}^{N-1} \int_{t_n}^{t_{n+1}} \mathbf{1}_{t \leq \tilde{\tau}} ((a - \bar{a}) \partial_x u + (\beta - \bar{\beta}) \partial_x^2 u)(\bar{X}_t, t) dt \right]. \end{aligned} \quad (2.25)$$

LEMMA 2.2. *Let $X(t)$ and $\bar{X}(t)$ be the exact path and the continuous Euler path defined in (1.2) and (2.3) respectively and assume that the assumptions in Theorem 2.2 hold. Then the time discretization error between these two paths has the following expansion*

$$E[g(X_\tau, \tau) - g(\bar{X}_{\tilde{\tau}}, \tilde{\tau})] = E \left[\sum_{n=0}^{N-1} \mathbf{1}_{t_{n+1} \leq \tilde{\tau}} \tilde{\rho}_n \Delta t_n^2 \right] + \mathcal{O}(\sqrt{\Delta t_{\max}}) E \left[\sum_{n=0}^{N-1} \mathcal{O}(\Delta t_n^2) \right] \quad (2.26)$$

where

$$\begin{aligned} \tilde{\rho}_n &= \frac{1}{2} ((\partial_t a + a \partial_x a + \beta \partial_x^2 a)(\bar{X}_n) \partial_x u(\bar{X}_{n+1}, t_{n+1}) \\ &\quad + (2\beta \partial_x a + \partial_t \beta + a \partial_x \beta + \beta \partial_x^2 \beta)(\bar{X}_n) \partial_x^2 u(\bar{X}_{n+1}, t_{n+1}) \\ &\quad + (2\beta \partial_x \beta)(\bar{X}_n) \partial_x^3 u(\bar{X}_{n+1}, t_{n+1})). \end{aligned} \quad (2.27)$$

Proof. Apply the Itô formula to each term in (2.25) to get

$$\begin{aligned} a(\bar{X}_t) - \bar{a}(\bar{X}_t) &= a(\bar{X}_t) - a(\bar{X}_n) \\ &= \int_{t_n}^t (\partial_s a + \bar{a} \partial_x a + \bar{\beta} \partial_x^2 a)(\bar{X}_s) ds + \int_{t_n}^t \bar{b} \partial_x a(\bar{X}_s) dW_s, \end{aligned}$$

and similarly

$$\begin{aligned} \beta(\bar{X}_t) - \bar{\beta}(\bar{X}_t) &= \int_{t_n}^t (\partial_s \beta + \bar{a} \partial_x \beta + \bar{\beta} \partial_x^2 \beta)(\bar{X}_s) ds + \int_{t_n}^t \bar{b} \partial_x \beta(\bar{X}_s) dW_s. \end{aligned}$$

Substitute the above integrals in (2.25) and use Malliavin derivatives, see [15], for example

$$\begin{aligned} & E \left[\sum_{n=0}^{N-1} \int_{t_n}^{t_{n+1}} \mathbf{1}_{t \leq \tilde{\tau}} \int_{t_n}^t \bar{b} \partial_x a(\bar{X}_s) \partial_x u(\bar{X}_t, t) dW_s dt \right] \\ &= E \left[\sum_{n=0}^{N-1} \int_{t_n}^{t_{n+1}} \mathbf{1}_{t \leq \tilde{\tau}} \int_{t_n}^t \bar{b} \partial_x a(\bar{X}_s) \partial_x^2 u(\bar{X}_t, t) ds dt \right] \end{aligned}$$

to get

$$\begin{aligned} & E[g(X_\tau, \tau) - g(\bar{X}_{\tilde{\tau}}, \tilde{\tau})] \tag{2.28} \\ &= E \left[\sum_{n=0}^{N-1} \int_{t_n}^{t_{n+1}} \mathbf{1}_{t \leq \tilde{\tau}} \left(\int_{t_n}^t (\partial_s a + \bar{a} \partial_x a + \bar{\beta} \partial_x^2 a)(\bar{X}_s) ds \partial_x u(\bar{X}_t, t) \right. \right. \\ &\quad \left. \left. + \int_{t_n}^t (2\bar{\beta} \partial_x a + \partial_s \beta + \bar{a} \partial_x \beta + \bar{\beta} \partial_x^2 \beta)(\bar{X}_s) ds \partial_x^2 u(\bar{X}_t, t) \right. \right. \\ &\quad \left. \left. + \int_{t_n}^t 2\bar{\beta} \partial_x \beta(\bar{X}_s) ds \partial_x^3 u(\bar{X}_t, t) \right) dt \right]. \end{aligned}$$

Each term in (2.28) has the form

$$E \left[\sum_{n=0}^{N-1} \int_{t_n}^{t_{n+1}} \int_{t_n}^t \mathbf{1}_{t \leq \tilde{\tau}} f(\bar{X}_s) h(\bar{X}_t, t) ds dt \right] \tag{2.29}$$

where f is a function of a, β and their derivatives representing the local error and h is a function of the derivatives of u . Finally apply the a priori error estimate (2.6) to the expected value (2.29) to conclude

$$\begin{aligned} & E \left[\sum_{n=0}^{N-1} \int_{t_n}^{t_{n+1}} \int_{t_n}^t \mathbf{1}_{t \leq \tilde{\tau}} f(\bar{X}_s) h(\bar{X}_t, t) ds dt \right] \\ &= E \left[\sum_{n=0}^{N-1} \frac{1}{2} \mathbf{1}_{t_{n+1} \leq \tilde{\tau}} f(\bar{X}_n) h(\bar{X}_{n+1}, t_{n+1}) \Delta t_n^2 \right] + \mathcal{O}(\sqrt{\Delta t_{\max}}) E \left[\sum_{n=0}^{N-1} \Delta t_n^2 \right] \end{aligned}$$

which proves (2.26). \square

Note that the quantities $\partial_x u, \partial_x^2 u$ and $\partial_x^3 u$ in (2.27) are not computable. The adaptive algorithms will use the computable approximations (2.13)-(2.18) for these functions. From the construction of u we have

$$\partial_x u(x, t) = E[\partial_x u(X_\tau, \tau) X'(\tau; t) | X'(t) = 1, X_t = x], \tag{2.30}$$

where $X'(s; t) := \partial X(s; X(t) = x) / \partial x$ is the first variation of $X(s)$ with respect to a perturbation in the initial location at time t , i.e. satisfies

$$\begin{aligned} dX'(s) &= \partial_x a(X(s)) X'(s) ds + \partial_x b(X(s)) X'(s) dW(s), \quad t < s < \tau, \\ X'(t) &= 1. \end{aligned} \tag{2.31}$$

The goal is to approximate $\partial_x u(\bar{X}_n, t_n)$ in (2.27) by a conditional expected value of the computable quantity φ defined in (2.13)-(2.14) and similarly to approximate $\partial_x^2 u$ and $\partial_x^3 u$ by expected values of φ' and φ'' in (2.15)-(2.16) and (2.17)-(2.18) respectively.

Note that if the continuous exact path finishes at $\tau = T$ then by the definition of u , we have $\partial_x u(X_T, T) = \partial_x g(X_T, T)$. However, for $\tau < T$ the first variation $\partial_{x_t} g(X_\tau, \tau)$ *does not exist*. Instead we approximate $\partial_x u(X_\tau, \tau)$ in (2.30) by the expected value of a difference quotient of g and remove this second expected value.

Let us first introduce a small positive constant Δx . Once the continuous exact path crosses the boundary, we start a new realization \hat{X} with the initial value

$$\hat{X}(\tau) = X(\tau) + \gamma \Delta x \in D$$

where γ denotes an inward unit normal vector. The new realization \hat{X}_t evolves by (1.2) for $\tau < t < \hat{\tau}$ until it stops with the first exit time of $\hat{\tau} \in (\tau, T]$. Then by the Taylor expansion we have

$$\partial_x u(X_\tau, \tau) = \frac{u(\hat{X}_\tau, \tau) - u(X_\tau, \tau)}{\Delta x} + \mathcal{O}(\Delta x)$$

and the Feynman-Kac formula (2.22) gives

$$\partial_x u(X_\tau, \tau) = \frac{E[g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - g(X_\tau, \tau) | \mathcal{G}_\tau]}{\Delta x} + \mathcal{O}(\Delta x)$$

where \mathcal{G}_t is the σ -algebra generated by $\{W(s), \Delta t(s) : s \leq t\}$. Substitute the above expansion in (2.30) and use the measurability of $X'(\tau; t) \in \mathcal{G}_\tau$ to get

$$\begin{aligned} & \partial_x u(x, t) \\ &= E \left[E \left[\frac{g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - g(X_\tau, \tau)}{\Delta x} \mid \mathcal{G}_\tau \right] X'(\tau; t) \mid \begin{array}{l} X'(t) = 1, X_t = x, \\ \hat{X}_\tau = X_\tau + \gamma \Delta x \end{array} \right] + \mathcal{O}(\Delta x) \\ &= E \left[E \left[\frac{g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - g(X_\tau, \tau)}{\Delta x} X'(\tau; t) \mid \mathcal{G}_\tau \right] \mid \begin{array}{l} X'(t) = 1, X_t = x, \\ \hat{X}_\tau = X_\tau + \gamma \Delta x \end{array} \right] + \mathcal{O}(\Delta x) \\ &= E \left[\frac{g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - g(X_\tau, \tau)}{\Delta x} X'(\tau; t) \mid \begin{array}{l} X'(t) = 1, X_t = x, \\ \hat{X}_\tau = X_\tau + \gamma \Delta x \end{array} \right] + \mathcal{O}(\Delta x). \end{aligned} \quad (2.32)$$

Let $(\hat{X}, \hat{\tau})$ be the Euler approximation of $(\hat{X}, \hat{\tau})$ and consider $Y = (X, X')^\top$, which satisfies the system of SDEs, (1.2) and (2.31) and write

$$dY(t) = A(Y(t))dt + B(Y(t))dW(t), \quad t > t_0, \quad Y(t_0) = (x, 1)^\top. \quad (2.33)$$

Similarly let us define the corresponding Euler approximation $\bar{Y} = (\bar{X}, \bar{X}')^\top$ of Y which solves

$$\bar{Y}(t_{n+1}) = \bar{Y}(t_n) + A(\bar{Y}(t_n))\Delta t_n + B(\bar{Y}(t_n))\Delta W_n. \quad (2.34)$$

Apply the a priori error estimate (2.6) to the function $\check{f}(Y_\tau, \tau) = \partial_x g(X_\tau, \tau)X'(\tau; t)\mathbf{1}_{\tau=T}$ to get

$$E[\check{f}(Y_\tau, \tau) - \check{f}(\bar{Y}_{\bar{\tau}}, \bar{\tau})] = \mathcal{O}(\sqrt{\Delta t_{\max}}).$$

Let $\mathcal{Y} = (Y, \hat{X})^\top$ and $\theta = (\tau, \hat{\tau})^\top$ with \hat{X}_t solves (1.2) for $\tau < t < \hat{\tau}$ and let the corresponding Euler approximation be $\bar{\mathcal{Y}} = (\bar{Y}, \bar{\hat{X}})^\top$ and $\bar{\theta} = (\bar{\tau}, \bar{\hat{\tau}})^\top$. Similarly apply the a priori error estimate (2.6) to the function $\check{f}(\mathcal{Y}_\theta, \theta) = (g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - g(X_\tau, \tau))/\Delta x X'(\tau; t)\mathbf{1}_{\tau < T}$ to obtain

$$E[\check{f}(\mathcal{Y}_\theta, \theta) - \check{f}(\bar{\mathcal{Y}}_{\bar{\theta}}, \bar{\theta})] = \mathcal{O}\left(\frac{\sqrt{\Delta t_{\max}}}{\Delta x}\right)$$

and consequently we have

$$\begin{aligned} \partial_x u(x, t) &= E[\partial_x g(\bar{X}_{\bar{\tau}}, \bar{\tau}) \bar{X}'(\bar{\tau}; t) \mathbf{1}_{\bar{\tau}=T} | \bar{X}'(t) = 1, \bar{X}_t = x] \\ &+ E \left[\frac{g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - g(\bar{X}_{\bar{\tau}}, \bar{\tau})}{\Delta x} \bar{X}'(\bar{\tau}; t) \mathbf{1}_{\bar{\tau} < T} \middle| \bar{X}'(t) = 1, \bar{X}_t = x, \hat{X}_{\hat{\tau}} = \bar{X}_{\bar{\tau}} + \gamma \Delta x \right] \\ &+ \mathcal{O} \left(\Delta x + \sqrt{\Delta t_{\max}} + \frac{\sqrt{\Delta t_{\max}}}{\Delta x} \right) \end{aligned} \quad (2.35)$$

which is an expansion of the expected value of φ defined in (2.13)-(2.14). The higher derivatives $\partial_x^2 u$ and $\partial_x^3 u$ can be computed in a similar way and we have the error expansion:

LEMMA 2.3. *Suppose the assumptions in Theorem 2.2 hold. Then the function u defined by (2.22) and the dual functions φ , φ' and φ'' defined by (2.13)-(2.18) satisfy for $\alpha = 1, 2, 3$*

$$\partial_x^\alpha u(\bar{X}(t_n), t_n) - E[\varphi_\alpha(t_n) | \mathcal{F}_n] = \mathcal{O} \left(\Delta x + \sqrt{\Delta t_{\max}} + \frac{\sqrt{\Delta t_{\max}}}{\Delta x} \right) \quad (2.36)$$

where \mathcal{F}_n denotes the σ -algebras generated by $\{W(s), \Delta t(s) : s \leq t_n\}$ and $\varphi_1 = \varphi$, $\varphi_2 = \varphi'$ and $\varphi_3 = \varphi''$.

Proof. For $\alpha = 1$, the approximation (2.35) and the definition (2.13)-(2.14) yield (2.36).

Let us as in [15] extend Y to be $(X, X', X'', X''')^T$ satisfying the SDE similar to (2.33) with $Y(t_0) = (x, 1, 0, 0)^T$. Here the first variation X' of X is defined in (2.31) and the other higher variations are defined similarly by taking the derivatives to the right hand side of (2.31). Let the corresponding Euler approximate $\bar{Y} = (\bar{X}, \bar{X}', \bar{X}'', \bar{X}''')^T$ satisfying the SDE similar to (2.34) and $(\bar{X}', \bar{X}'', \bar{X}''')$ denote the Euler approximations of (X', X'', X''') . For the case when $\tau = T$ and $\alpha = 2$ or 3 , we use the a priori error estimate (2.6) for the extended system Y and \bar{Y} with

$$\check{f}(Y_\tau, \tau) = \begin{cases} (\partial_x^2 g(X')^2 + \partial_x g X'') \mathbf{1}_{\tau=T}, & \text{if } \alpha = 2, \\ (\partial_x^3 g(X')^3 + 3\partial_x^2 g X' X'' + \partial_x g X''') \mathbf{1}_{\tau=T}, & \text{if } \alpha = 3. \end{cases}$$

For the case $\tau < T$, the initial condition of the second derivative is obtained by the Kolmogorov backward equation (2.21) and the approximation $\partial_x u \simeq E[\varphi | \mathcal{F}]$ in (2.36) gives

$$\begin{aligned} \partial_x^2 u(\bar{X}_{\bar{\tau}}, \bar{\tau}) &= -(\beta(\bar{X}_{\bar{\tau}}))^{-1} (\partial_t u(\bar{X}_{\bar{\tau}}, \bar{\tau}) + a(\bar{X}_{\bar{\tau}}) \partial_x u(\bar{X}_{\bar{\tau}}, \bar{\tau})) \\ &\simeq E [-(\beta(\bar{X}_{\bar{\tau}}))^{-1} (\partial_t g(\bar{X}_{\bar{\tau}}, \bar{\tau}) + a(\bar{X}_{\bar{\tau}}) \varphi(\bar{\tau})) | \mathcal{F}_{\bar{\tau}}] \\ &= E[\varphi'(\bar{\tau}) | \mathcal{F}_{\bar{\tau}}]. \end{aligned} \quad (2.37)$$

Apply one more derivative to the Kolmogorov backward equation (2.21) to obtain

$$\partial_x^3 u = -\beta^{-1} (\partial_t \partial_x u + \partial_x a \partial_x u + a \partial_x^2 u + \partial_x \beta \partial_x^2 u). \quad (2.38)$$

Here $\partial_x u$ and $\partial_x^2 u$ are approximated by $E[\varphi | \mathcal{F}]$ and $E[\varphi' | \mathcal{F}]$ using (2.36). The quantity $\partial_t \partial_x u$ can also be computed similarly. Let $v(x, t) = \partial_t u(x, t)$. Then from the Kolmogorov backward equation (2.21) v satisfies

$$\partial_t v + a \partial_x v + \beta \partial_x^2 v = 0, \quad v(\cdot, T) = \partial_t g(\cdot, T), \quad v(\lambda, \cdot) = \partial_t g(\lambda, \cdot),$$

which is the same partial differential operator as for u but with different boundary conditions. Therefore the derivation of (2.35) shows

$$\partial_t \partial_x u(X_\tau, \tau) = \partial_x v(X_\tau, \tau) = E \left[\frac{\partial_t g(\hat{X}_{\hat{\tau}}, \hat{\tau}) - \partial_t g(\bar{X}_{\bar{\tau}}, \bar{\tau}) \mathbf{1}_{\bar{\tau} < T}}{\Delta x} \right] + \mathcal{O} \left(\Delta x + \frac{\sqrt{\Delta t_{\max}}}{\Delta x} \right),$$

which together with (2.37) and (2.38) proves (2.36). \square

Proof of Theorem 2.2. The measurability of the function f_n depending on the derivatives of a and β , e.g. $f_n = \mathbf{1}_{t_{n+1} \leq \bar{\tau}} (\partial_t a + a \partial_x a + \beta \partial_x^2 a)(\bar{X}_n) \Delta t_n^2 \in \mathcal{F}_{n+1}$, proves

$$\begin{aligned} E \left[\sum_{n=0}^{N-1} f_n E[\varphi(t_{n+1}) | \mathcal{F}_{n+1}] \right] &= E \left[E \left[\sum_{n=0}^{N-1} f_n \varphi(t_{n+1}) | \mathcal{F}_{n+1} \right] \right] \\ &= E \left[\sum_{n=0}^{N-1} f_n \varphi(t_{n+1}) \right]. \end{aligned} \quad (2.39)$$

Similar representations hold for the other terms in (2.27). Consequently, the combination of Lemma 2.2-2.3 and the removal of the second expectation (2.39) prove (2.19). \square

REMARK 2.3. *The optimal size of the constant Δx for the difference quotient in (2.36) is $\mathcal{O}((\Delta t_{\max})^{1/4})$ and $\Delta x = \text{TOL}_T^{1/4}$ is used for the adaptive algorithm in Section 3 where TOL_T is a given time discretization error tolerance.*

3. Adaptive Algorithms for Stopped Diffusion. This section presents adaptive algorithms for the stopped diffusion problems. As described in Section 2, the computational error is separated into the following three terms : the time discretization error between the exact and the continuous Euler path \mathcal{E}_C , the time discretization error between the continuous and discrete Euler approximation \mathcal{E}_D , and the statistical error \mathcal{E}_S , i.e.

$$\begin{aligned} &E[g(X(\tau), \tau)] - \frac{1}{M} \sum_{j=1}^M g(\bar{X}(\bar{\tau}; \omega_j), \bar{\tau}) \\ &= E[g(X(\tau), \tau) - g(\bar{X}(\tilde{\tau}), \tilde{\tau})] + E[g(\bar{X}(\tilde{\tau}), \tilde{\tau}) - g(\bar{X}(\bar{\tau}), \bar{\tau})] \\ &+ \left(E[g(\bar{X}(\bar{\tau}), \bar{\tau})] - \frac{1}{M} \sum_{j=1}^M g(\bar{X}(\bar{\tau}; \omega_j), \bar{\tau}) \right) \\ &=: \mathcal{E}_C + \mathcal{E}_D + \mathcal{E}_S. \end{aligned} \quad (3.1)$$

For a given error tolerance TOL , the goal is to minimize the computational work, which is roughly $\mathcal{O}(M \cdot N) = \mathcal{O}(\text{TOL}_S^{-2} \text{TOL}_T^{-1})$ where TOL_S and TOL_T denote a statistical tolerance and a time discretization tolerance respectively. Thus we obtain

$$\text{TOL}_S = \frac{2}{3} \text{TOL} \quad \text{and} \quad \text{TOL}_T = \frac{1}{3} \text{TOL} \quad (3.2)$$

by solving

$$\min \text{TOL}_S^{-2} \text{TOL}_T^{-1} \quad \text{subject to} \quad \text{TOL}_S + \text{TOL}_T = \text{TOL}.$$

3.1. Control of the Statistical Error. Let us first introduce some notation. Define the sample average $\mathcal{A}(Y; M)$ and the sample standard deviation $\bar{\sigma}(Y; M)$ of Y by

$$\mathcal{A}(Y; M) := \frac{1}{M} \sum_{j=1}^M Y(\omega_j), \quad \bar{\sigma}(Y; M) := (\mathcal{A}(Y^2; M) - (\mathcal{A}(Y; M))^2)^{\frac{1}{2}}$$

and denote by Φ the distribution function of a normal random variable with mean zero and variance one, i.e.

$$\Phi(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}s^2\right) ds. \quad (3.3)$$

Then from the Central Limit Theorem, the statistical error \mathcal{E}_S in (3.1) satisfies

$$|\mathcal{E}_S| \leq \mathbf{E}_S(Y; M) := c_0 \frac{\bar{\sigma}(Y; M)}{\sqrt{M}} \quad (3.4)$$

with probability close to one asymptotically, where $Y = g(\bar{X}_{\bar{\tau}}, \bar{\tau})$ and c_0 is a constant for a confidence interval. For example, $c_0 \geq 1.65$ gives asymptotically the probability greater than 0.90.

For a given $\text{TOL}_S > 0$, the goal is to find M adaptively such that $\mathbf{E}_S(Y; M) \leq \text{TOL}_S$ as follows, see also in [13], [15] :

routine Monte-Carlo($\text{TOL}_S, M_0; EY$)

Set the batch counter $m = 1$, $M[1] = M_0$ and $\mathbf{E}_S[1] = +\infty$.

Do while ($\mathbf{E}_S[m] > \text{TOL}_S$)

Compute $M[m]$ new samples of Y and compute the sample average

$EY := \mathcal{A}(Y; M[m])$, the sample standard deviation $\bar{\sigma}[m] := \bar{\sigma}(Y; M[m])$

and the error estimate $\mathbf{E}_S[m+1] := \mathbf{E}_S(Y; M[m])$.

Compute $M[m+1]$ by **changeM** ($M[m], \bar{\sigma}[m], \text{TOL}_S; M[m+1]$).

Increase m by 1.

end-do

end of Monte-Carlo

routine changeM ($M_{in}, \bar{\sigma}_{in}, \text{TOL}_S; M_{out}$)

$$\begin{aligned} M^* &= \min \left\{ \text{integer part} \left(\frac{c_0 \bar{\sigma}_{in}}{\text{TOL}_S} \right)^2, \text{MCH} \times M_{in} \right\} \\ n &= \text{integer part} (\log_2 M^*) + 1 \\ M_{out} &= 2^n. \end{aligned} \quad (3.5)$$

end of changeM

Here, M_0 is a given initial value for M , and $\text{MCH} > 1$ is a positive integer parameter introduced to avoid a large new number of realizations in the next batch due to a possibly inaccurate sample standard deviation $\bar{\sigma}[m]$. Indeed, $M[m+1]$ cannot be greater than $\text{MCH} \times M[m]$.

3.2. Control of the Time Discretization Error. In this subsection, we present two refinement strategies to control the time discretization error. For a given partition $0 = t_0 < t_1 < \dots < t_{N-1} = T$, the piecewise constant mesh function Δt is defined by (2.7) and the corresponding number $N(\Delta t)$ of steps is

$$N(\Delta t) := \int_0^T \frac{1}{\Delta t(s)} ds.$$

Then the optimal choice of the time steps is formulated by minimizing the computational work $E[N(\Delta t)]$ such that $\Delta t \in \mathcal{K}$ subject to given accuracy constraints. The feasible set \mathcal{K} for the mesh function Δt is defined by

$$\mathcal{K} := \{ \Delta t : \Delta t \text{ is stochastic, positive and piecewise constant on } [0, T] \text{ for each realization } \}.$$

3.2.1. Total Time Discretization Error. The goal is to make the total time discretization error, $\mathcal{E}_T = \mathcal{E}_C + \mathcal{E}_D$ defined in (1.4), bounded by a given time discretization error tolerance TOL_T in (3.2). Therefore the accuracy constraint is

$$E \left[\sum_{n=0}^{N-1} r_n \right] \leq \text{TOL}_T \quad (3.6)$$

where the error indicators r_n are defined for $n = 0, 1, \dots, N-1$, by

$$r_n := \left| \mathbf{1}_{t_{n+1} \leq \bar{\tau}} \rho_n \Delta t_n^2 + (g(\lambda, \frac{1}{2}(t_n + t_{n+1})) - g(\bar{X}_{\bar{\tau}}, \bar{\tau})) \hat{P}_{\bar{X}, n} \right| \quad (3.7)$$

with ρ_n in (2.20) and $\hat{P}_{\bar{X}, n}$ in (2.10).

To have as few time steps as possible, we try to make

$$r_n(\omega) = \text{constant}, \quad \forall n \text{ and } \forall \omega$$

and by (3.6) the natural choice of the constant is then

$$r_n(\omega) = \frac{\text{TOL}_T}{E[N]}, \quad \forall n \text{ and } \forall \omega. \quad (3.8)$$

The choice (3.8) is optimal in the case of without boundary, see [13], [15], i.e. without the second term in (3.7). Numerical tests show that the error \mathcal{E}_D in (3.1), corresponding the second term in (3.7), converges exponentially fast as the number of adaptive steps is increased. Therefore an over-refinement in this part of the error does not seem to cost much. Note that in practice the quantity $E[N]$ is not known and we can only estimate it by the sample average $\bar{N}[j] := \mathcal{A}(N; M[j])$ of the final number of the time steps in the j th batch of $M[j]$ numbers of realizations. Then the statistical error, $|E[N] - \bar{N}[j]|$, is bounded by $E_S(N; M[j])$, with probability close to one, by the same argument as in (3.4).

To achieve (3.8), start with an initial mesh $\Delta t[1]$ and then specify iteratively a new partition $\Delta t[k+1]$ from $\Delta t[k]$, using the following refinement strategy: for each realization in the m th batch and for each time step $n = 0, 1, \dots, N[k]-1$,

$$\begin{aligned} & \mathbf{if} \left(r_n[k] \geq \frac{\text{TOL}_T}{\bar{N}[m-1]} \right) \mathbf{then} & (3.9) \\ & \quad \text{divide } \Delta t_n[k] \text{ into } H \text{ uniform substeps} \\ & \mathbf{else} \text{ let the new step be the same as the old} \\ & \mathbf{endif.} \end{aligned}$$

Here H is a given integer greater than 1, which bounds the increment of the number of time steps from one iteration to the next. The following analysis, for fixed H , can easily be extended to bounded and varying H .

The refinement strategy (3.9) motivates the following stopping criteria: for each realization of the m th batch

$$\mathbf{if} \left(\max_{1 \leq n \leq N[k]} r_n[k] < S \frac{\text{TOL}_T}{\overline{N}[m-1]} \right) \mathbf{then} \text{ stop.} \quad (3.10)$$

Here S is a given constant, determined more precisely as follows: we want the maximal error indicator to decay quickly to the stopping level $S\text{TOL}_T/\overline{N}$, but when almost all r_n satisfy $r_n \leq \text{TOL}_T/\overline{N}$, the reduction of the error may be slow. The constant S is motivated as a cure to this slow reduction.

3.2.2. Splitting of the Time Discretization Error. Let us compare the adaptive algorithm (3.9)-(3.10) with the following *ad hoc* refinement algorithm. First we split the time discretization tolerance $\text{TOL}_T = \text{TOL}_C + \text{TOL}_D$ by $\text{TOL}_C = \text{TOL}_D = \text{TOL}_T/2$ and define the error indicators r_n^C and r_n^D by

$$r_n^C := \mathbf{1}_{t_{n+1} \leq \bar{\tau}} |\rho_n| \Delta t_n^2 \quad \text{and} \quad r_n^D := \left| g\left(\lambda, \frac{1}{2}(t_n + t_{n+1})\right) - g(\overline{X}_{\bar{\tau}}, \bar{\tau}) \right| \hat{P}_{\overline{X},n} \quad (3.11)$$

with ρ_n in (2.20) and $\hat{P}_{\overline{X},n}$ in (2.10). This alternative refinement strategy is to take into account the computational observation that only a few time intervals for each realization have large error indicators r_n^D compared to the others, see Fig. 3.1, an illustrative Monte Carlo realization of r_n^D for Example 4.1 in Section 4.

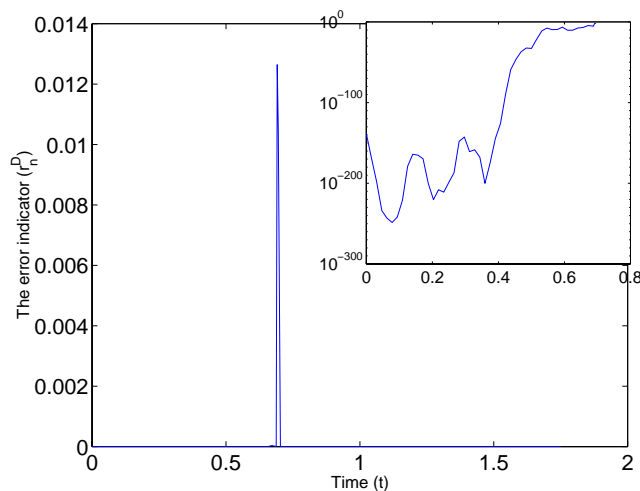


FIG. 3.1. Example 4.1: An illustrative Monte Carlo realization of r_n^D with $\text{TOL} = 0.1$.

Start the algorithm with an initial mesh $\Delta t[1]$ and then specify iteratively a new partition $\Delta t[k+1]$ from $\Delta t[k]$ using following refinement strategy: for each realization

in the m th batch and for each time step $n = 0, 1, \dots, N[k] - 1$,

$$\mathbf{if} \left(r_n^C[k] \geq \frac{\text{TOL}_C}{\overline{N}[m-1]} \text{ or } r_n^D[k] \geq \text{TOL}_D \right), \quad (3.12)$$

then divide $\Delta t_n[k]$ into H uniform substeps

else let the new step be the same as the old one

endif.

until the following stopping criteria is fulfilled: for each realization of the m th batch

$$\mathbf{if} \left(\max_{1 \leq n \leq N[k]} r_n^C[k] < S_C \frac{\text{TOL}_C}{\overline{N}[m-1]} \right) \text{ and } \left(\max_{1 \leq n \leq N[k]} r_n^D[k] < S_D \text{TOL}_D \right) \quad (3.13)$$

then stop.

Here S_C and S_D are given constants to cure the slow reduction when almost all r_n^C or r_n^D satisfy $r_n^C \leq \text{TOL}_C/\overline{N}$ or $r_n^D \leq \text{TOL}_D$.

3.3. The Adaptive Algorithms. The adaptive stochastic time stepping algorithms have structures similar to the `Monte-Carlo` routine. First we split the specified error tolerance by (3.2) and the outer loop computes the batches of realizations of \overline{X} , until an estimate for the statistical error (3.4) is below the tolerance, TOL_S . In the inner loop, for each realization, we apply our refinement strategy (3.9) or (3.12) to a given initial mesh iteratively until the error indicators satisfy the stopping criteria (3.10) or (3.13) with a given time discretization tolerance TOL_T . This procedure needs to sample the Wiener process W on finer partitions, given its values on coarser. Therefore, the use of Brownian bridges is natural to preserve the required independence between the Wiener increments.

The adaptive algorithm based on the refinement (3.9) and the stopping (3.10) is called `Algorithm A` and the algorithm based on the refinement (3.12) and the stopping (3.13) is called `Algorithm B`. We first describe `Algorithm A` in detail and define the additional changes for `Algorithm B` afterwards.

Algorithm A

Initialization Choose:

1. an error tolerance, $\text{TOL} = \text{TOL}_S + \text{TOL}_T$,
2. a number, $N[1]$, of initial uniform steps $\Delta t[1]$ for $[0, T]$ and set $\overline{N} = N[1]$,
3. a number, $M[1]$, of initial realizations,
4. an integer $H \geq 2$ for the number of subdivisions of a refined time step, and the stopping constant S in (3.10),
5. a positive constant c_0 for a confidence interval and an integer $\text{MCH} \geq 2$ to determine the number of realizations in (3.5), and
6. a constant Δx for the difference quotient in (2.35), see Remark 2.3.

Set the iteration counter for batches $m = 1$ and the stochastic error $\mathbf{E}_S[m] = +\infty$.

Do while ($\mathbf{E}_S[m] > \text{TOL}_S$)

For realizations $j = 1, \dots, M[m]$

Set $k = 1$ and $r[k] = +\infty$.

Start with the initial partition $\Delta t[k]$ and generate $\Delta W[k]$.

Do while ($r[k]$ violates the stopping (3.10))

Compute $\Delta t[k+1]$, $\Delta W[k+1]$, $r[k+1]$ and $N[k+1]$ by

```

    routine Control-Time-Error( $\Delta t[k], \Delta W[k], N[k]; \Delta t[k+1],$ 
       $\Delta W[k+1], r[k+1], N[k+1]$ ). Increase  $k$  by 1.
  end-do
  Let  $J$  be the number of the final refinement level and  $g_J := g(\overline{X}_\tau, \overline{\tau})[J]$ .
end-for
Compute the sample average  $Eg := \mathcal{A}(g_J; M[m])$ , the sample standard
deviation  $\mathcal{S}[m] := \mathcal{S}(g_J; M[m])$  and the a posteriori bound for the statistical
error  $E_S[m] := E_S(g_J, M[m])$  in (3.4).
Compute  $M[m+1]$  by changeM ( $M[m], \mathcal{S}[m], \text{TOL}_S; M[m+1]$ ) in (3.5),
and update  $\overline{N} = \mathcal{A}(N[J]; M[m])$ . Increase  $m$  by 1.
end-do
Accept  $Eg$  as an approximation of  $E[g(X_\tau, \tau)]$ , since the estimate of the
computational error is bounded by TOL.

routine Control-Time-Error( $\Delta t_{in}, \Delta W_{in}, N_{in}; \Delta t_{out}, \Delta W_{out}, r_{out}, N_{out}$ )
  Compute the new Euler approximation  $\overline{X}$  in (2.1) and the error indicators
   $r_{out}$  in (3.7) on  $\Delta t_{in}$  with the Wiener increments  $\Delta W_{in}$ .
  If (  $r_{out}$  violates the stopping (3.10) )
    For time steps  $i = 1, \dots, N_{in}$ 
      Do the refinement process (3.9) to compute  $\Delta t_{out}$  from  $\Delta t_{in}$  and
      compute  $\Delta W_{out}$  from  $\Delta W_{in}$  using Brownian bridges.
    end-for
  end-if
  Let  $N_{out}$  be the number of time steps for  $\Delta t_{out}$ .
end of Control-Time-Error

```

Algorithm B

In addition to the **Initialization** of Algorithm A, choose the error tolerances $\text{TOL}_T = \text{TOL}_C + \text{TOL}_D$ and the stopping constants S_C and S_D in (3.13). Inside the **Do while** loop of Algorithm A, use $(r^C[k], r^D[k])$ in (3.11) instead of $r[k]$ and the refinement (3.12) and stopping (3.13).

4. Numerical Experiments. This section presents numerical results from a C++ implementation of Algorithm A and Algorithm B described in Section 3. The numerical results are obtained by using the pseudo-random number generator, `drand48()`, in standard C library functions. The Box-Muller method is used to generate standard Gaussian random variable from the uniformly distributed pseudo-random numbers, see for example [10].

In all computations, the following constants are chosen for the initialization of both Algorithm A and Algorithm B: the number of time steps in the initial partition, $\Delta t[1]$, of $[0, T]$ is $N[1] = 4$; the initial number of realization is $M[1] = 128$; the number of subdivisions of a refined step is $H = 2$; the stopping constant $S = 4$ is used in (3.10) and $S_C = 4, S_D = 1$ in (3.13); the constants to determine the number of realizations in (3.5) are $c_0 = 1.65$ and $MCH = 16$, and the constant $\Delta x = \text{TOL}_T^{1/4}$ is used for the difference quotient in (2.35).

To describe the behavior of the adaptive algorithm, let us first define some notations. The index Q , which is the ratio between the exact error and the approximate

error, is defined by

$$Q := \frac{E_{approx}}{E_{exact}} := \frac{E_S + |E_T|}{|E[g(X_\tau, \tau)] - \mathcal{A}(g(\bar{X}_{\bar{\tau}}, \bar{\tau}); M)|}. \quad (4.1)$$

Here the statistical error E_S is defined by (3.4) and the time discretization error E_T is defined by

$$E_T := \mathcal{A} \left(\sum_{n=0}^{N-1} \mathbf{1}_{t_{n+1} \leq \bar{\tau}} \rho_n \Delta t_n^2 + (g(\lambda, \frac{1}{2}(t_n + t_{n+1})) - g(\bar{X}_{\bar{\tau}}, \bar{\tau})) \hat{P}_{\bar{X}, n}; M \right).$$

EXAMPLE 4.1. Consider (1.2) with

$$a(t, x) = \frac{11}{36}x, \quad b(t, x) = \frac{1}{6}x, \quad t \in [0, T], \quad x \in (-\infty, 2)$$

and the initial condition $X(0) = 1.6$ and $T = 2$. For $g(x, t) = x^3 e^{-t}$ with $x \in \mathbb{R}$, this problem has the exact solution $E[g(X_\tau, \tau)] = u(X(0), 0) = X(0)^3$, where the solution u of the Kolomogorov backward equation (2.21) is $u(x, t) = x^3 e^{-t}$.

To check the behavior of the error expansion described in Section 2, Example 4.1 is constructed such that most of the realizations exit at $\bar{\tau} < T$, for instance, with TOL = 0.01, 99% of the paths exit at $\bar{\tau} < T$ and $\mathcal{A}(\bar{\tau}; M) \simeq 0.77$.

TOL	M	Algorithm A			Algorithm B		
		$\mathcal{A}(N; M)$	$\bar{\sigma}(N; M)$	E_{exact}	$\mathcal{A}(N; M)$	$\bar{\sigma}(N; M)$	E_{exact}
0.5	2^7	27	11.7	0.028	24	6.9	0.02
0.1	2^{11}	81	30.6	0.024	84	25.8	0.06
0.05	2^{13}	126	44.0	0.015	158	54.2	0.02
0.01	2^{18}	453	170.7	0.003	700	287.7	0.005

TABLE 4.1

Example 4.1: Comparisons of the final number of the realizations, M , the sample average of the final number of steps, $\mathcal{A}(N; M)$, the sample standard deviation of the final number of steps, $\bar{\sigma}(N; M)$, and the exact error, E_{exact} for different error tolerances, TOL.

Table 4.1 shows the comparisons between Algorithm A and Algorithm B for the computational results of Example 4.1. As the error tolerance TOL decreases, E_{exact} decreases and is bounded by a given TOL. The sample standard deviation of the number of time steps is around 35% of the average of the number of time steps, which shows that highly varying step sizes are used for each realization (see also the histogram of the time step sizes in Fig. 4.3) and that the stochastic time stepping is adequate for these stopping problems rather than deterministic time stepping, see [15],[13].

To check the accuracy of the a posteriori error estimate in Section 2, choose the number of realizations M sufficiently large so that the total statistical error is small compared to the time discretization error. Here we use $M = 2^{22} = 4,194,304$, which makes the statistical error approximately 0.001. Then the comparison of the convergence between the uniform and the adaptive method is shown in Fig. 4.1. The x -axis denotes the number of time steps for the uniform method and the sample average of the final number of steps for the adaptive method. The y -axis is the exact error E_{exact} defined by (4.1). The number of steps $N = 2^k, k = 3, 4, \dots, 10$ are used

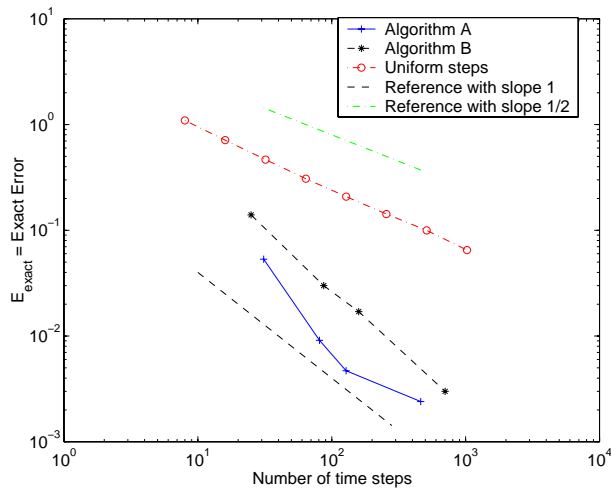


FIG. 4.1. *Example 4.1: Comparison of the convergence rates with uniform and adaptive meshes. The convergence rate of the adaptive method is of order N^{-1} with N adaptive time steps, while the rate for the uniform method is of order $N^{-1/2}$ with N uniform time steps.*

for the uniform method and for adaptive method the tolerances $TOL = 0.5, 0.1, 0.05$ and 0.01 are used. Fig. 4.1 shows that the convergence rate of the adaptive method is of order N^{-1} with N adaptive time steps, while the uniform method converges with the rate $N^{-1/2}$ with N uniform time steps.

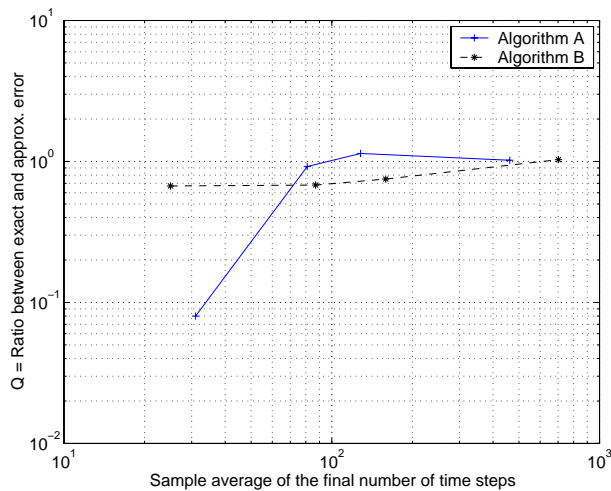


FIG. 4.2. *Example 4.1: The ratio of the approximate and exact error on adaptive mesh. The ratio tends to 1 as the number of time steps increases.*

Fig. 4.2 shows the convergence of the ratio Q between the exact and the approximate error in (4.1), still with $M = 2^{22}$ so that the statistical error is negligible. As predicted by Theorem 2.1 and 2.2, Fig. 4.2 shows that the ratio Q tends to 1 as N increases. From Fig. 4.1 and 4.2, Algorithm B seems more stable than Algorithm A for Example 4.1, on the other hand Algorithm A achieves smaller exact error for the

same number of time steps.

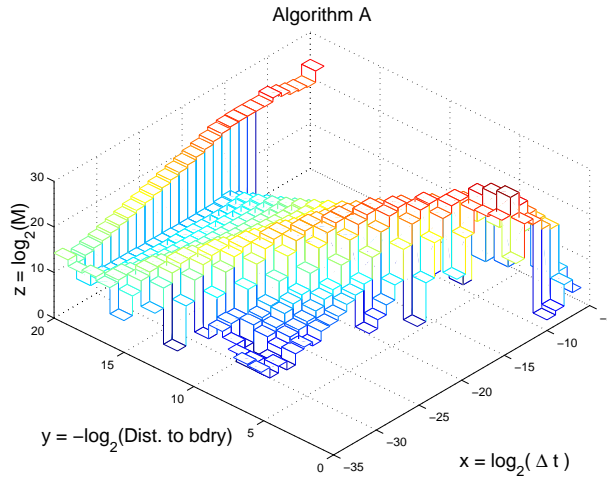


FIG. 4.3. *Example 4.1: The histogram of the step sizes depending on the distance from the boundary using Algorithm A. Relatively small step sizes are used close to the boundary to improve the accuracy.*

Fig. 4.3 shows the histogram of the step sizes depending on the distance from the boundary with $TOL = 0.05$ and the number of realizations $M = 2^{22}$ of **Algorithm A**. The histogram of **Algorithm B** also has a similar feature. The x -axis denotes base 2 log-scale of the step size, ranging from 2^{-35} to 2^{-5} , y -axis denotes base 2 log-scale of the distance from the boundary, ranging from 2^{-20} to 1, and z -axis denotes base 2 log-scale of the number of realizations M . To compensate the large error near the boundary, relatively small step sizes are used close to the boundary compared to further away from the boundary.

Acknowledgments. This work is supported by the Swedish Research Council, the Swedish Parallel and Scientific Computing Institute (PSCI), and the Swedish National Board for Industrial and Technical Development (NUTEK).

REFERENCES

- [1] P. Baldi, *Exact asymptotics for the probability of exit from a domain and applications to simulation*, Ann. Probab. 23 (1995), no. 4, 1644–1670.
- [2] P. Baldi, L. Caramellino and M.G. Iovino, *Pricing general barrier options: a numerical approach using sharp large deviations*, Math. Finance 9 (1999), no. 4, 293–322.
- [3] V. Bally and D. Talay, *The law of the Euler scheme for stochastic differential equations, I. Convergence rate of the distribution function*, Probab. Theory Related Fields 104 (1996), no. 1, 43–60.
- [4] F.M. Buchmann, *Computing exit times with the Euler scheme*, Research report no. 2003-02, ETH (2003).
- [5] W.H. Fleming and M.R. James, *Asymptotic series and exit time probabilities*, Ann. Probab. 20 (1992), no. 3, 1369–1384.
- [6] E. Gobet, *Weak approximation of killed diffusion using Euler schemes*, Stochastic Process. Appl. 87 (2000), no. 2, 167–197.
- [7] E. Gobet, *Euler schemes and half-space approximation for the simulation of diffusion in a domain*, ESAIM Probab. Statist. 5 (2001), 261–297.
- [8] K.M. Jansons and G.D. Lythe, *Efficient numerical solution of stochastic differential equations using exponential timestepping*, J. Stat. Phys. 100 (2000), no. 5/6, 1097–1109.

- [9] I. Karatzas and S.E. Shreve, *Brownian motion and stochastic calculus*, Graduate Texts in Mathematics, 113. Springer-Verlag, New York, 1991.
- [10] P.E. Kloeden and E. Platen, *Numerical solution of stochastic differential equations*, Applications of Mathematics, 23. Springer-Verlag, Berlin, 1992.
- [11] R. Mannella, *Absorbing boundaries and optimal stopping in a stochastic differential equation*, Phys. Lett. A 254 (1999), no. 5, 257–262.
- [12] K.-S. Moon, E. von Schwerin, A. Szepessy and R. Tempone, *An adaptive Monte Carlo algorithm for stopped diffusion*, in preparation, (2003).
- [13] K.-S. Moon, A. Szepessy, R. Tempone and G.E. Zouraris, *Convergence rates for adaptive weak approximation of stochastic differential equations*, preprint, (2001).
- [14] W. P. Petersen and F. M. Buchmann, *Solving Dirichlet problems numerically using the Feynman-Kac representation*, Research report no. 2002-01, ETH (2002).
- [15] A. Szepessy, R. Tempone and G.E. Zouraris, *Adaptive weak approximation of stochastic differential equations*, Comm. Pure Appl. Math. 54 (2001), no. 10, 1169–1214.
- [16] D. Talay and L. Tubaro, *Expansion of the global error for numerical schemes solving stochastic differential equations*, Stochastic Anal. Appl. 8 (1990), no. 4, 483–509.