Solving high dimensional Dirichlet problems numerically using the Feynman-Kac representation

Diss. ETH No. 15547

Solving High Dimensional Dirichlet Problems Numerically Using the Feynman-Kac Representation

FABIAN M. BUCHMANN

2004

Diss. ETH No. 15547

Solving High Dimensional Dirichlet Problems Numerically Using the Feynman-Kac Representation ‡

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZURICH

for the degree of
Doktor der Mathematik

presented by
FABIAN M. BUCHMANN
Dipl. Rech. Wiss. ETH
born May 25, 1974
citizen of Zürich ZH, Switzerland

accepted on the recommendation of
Prof. Dr. Rolf Jeltsch, examiner
Dr. Wesley P. Petersen, co-examiner
Prof. Dr. Peter Kloeden, co-examiner

2004

‡: This work was supported in part under the TH-Gesuch funds 0-20981-2.

Abstract

We propose two different algorithms to solve high dimensional Dirichlet problems numerically. Both approaches are based on the stochastic representation of the solution in the form of the famous Feynman-Kac formula. Here, the solution to an elliptic second order partial differential equation (PDE) is given by the expectation over some functionals of connected stochastic differential equations (SDEs). For the Dirichlet problem, the boundary condition makes the solution of the PDE unique.

Monte-Carlo simulations involve the numerical approximation of the solution of the SDEs and the approximation of the mathematical expectation by a finite mean. For the simulation in a bounded domain, however, the first exit time of the process solving the SDE from the domain also plays a crucial role. On the one hand, integration is stopped as soon as the process reaches the boundary and on the other hand, the boundary condition is to be evaluated at the first exit point of the process from the domain.

The proposed algorithms not only succeed in the approximation of the first exit time but also give an accurate approximation for the first exit point. Both algorithms make use of the fact that approximations of the solution of the SDEs have to be accurate only in a weak sense. Nevertheless, the two methods have essential differences.

The first method uses bounded approximations for the increments of the simulated Brownian motion. This allows us to prevent the resulting discrete random walk from leaving the closure of the domain and the walk to approach the boundary in a controlled way. We prove both, the expected number of steps and the accuracy of the resulting one step approximation in the weak sense. A series of numerical tests is included.

The second method uses Euler's method to approximately integrate the SDEs. Using normally distributed random variables to model the Brownian increments, the resulting discrete random walk is no longer restricted to the closure of the domain, but, might leave it within every step. An *a posteriori* test accounts for additional errors arising by the presence of boundaries. Corresponding exit times and exit points are sampled from analytical densities. Results from numerical experiments show that weak order one convergence of the Euler scheme is restored. Additionally, our results show the superiority of the proposed

algorithm compared to other methods.

Both methods are ideally suited for parallel simulations. Numerical tests in dimensions as high as 128 were run on the Beowulf cluster at ETH.

Kurzfassung

Wir stellen in dieser Arbeit zwei verschiedene Methoden zur numerischen Lösung hochdimensionaler Dirichlet-Probleme vor. Beide Lösungsvorschläge basieren auf der stochastischen Darstellung der Lösung in Form der berühmten Feynman-Kac-Formel. Mit dieser Formel ist die Lösung einer elliptischen partiellen Differentialgleichung (englisch: PDE) zweiter Ordnung als Erwartungswert bestimmter Funktionale stochastischer Differentialgleichungen (englisch: SDEs) darstellbar. Diese SDEs sind ihrerseits wieder mit der PDE über die Koeffizienten verbunden. Beim Dirichlet-Problem wird die Lösung der PDE durch Vorgabe einer Randbedingung eindeutig.

Monte-Carlo-Simulationen beinhalten im Prinzip nur die numerische Näherungslösung der SDEs und die Approximation des Erwartungswertes durch einen endlichen Mittelwert. Eine zentrale Rolle bei der Simulation in einem beschränkten Gebiet spielt jedoch zusätzlich der Zeitpunkt, in dem der Prozess, welcher die SDE löst, zum ersten Mal den Rand des Gebietes erreicht: Einerseits muss die Integration beendet werden, sobald der Prozess zum ersten Mal das Gebiet verlässt, und andererseits muss die Randbedingung an diesem Punkt ausgewertet werden.

Die in dieser Arbeit vorgeschlagenen Algorithmen approximieren mit hoher Genauigkeit neben dem Zeitpunkt, in welchem der Prozess zum ersten Mal das Gebiet verlässt, auch den zugehörigen Austrittspunktes. Beide Algorithmen nutzen aus, dass die Näherungslösungen der SDEs nur in einem schwachen Sinn gegen die exakten Lösungen konvergieren müssen. Trotz dieser Gemeinsamkeit unterscheiden sich die beiden Methoden in grundlegenden Punkten.

Die erste Methode verwendet beschränkte Approximationen für die Inkremente der simulierten Brown'schen Bewegung. Dadurch können wir verhindern, dass der resultierende diskrete Zufallslauf den Abschluss des Gebietes verlässt und können so dessen Annäherung an den Rand des Gebietes kontrollieren. Wir beweisen sowohl die Anzahl der im Mittel benötigten Schritte des Zufallslaufes, als auch die Genauigkeit der resultierenden Näherung für einen Zeitschritt. Wir zeigen Resultate mehrerer numerischer Experimente.

Die zweite Methode benutzt das Eulerverfahren zur näherungsweisen Integration der

SDEs. Da dabei normal verteilte Zufallszahlen zur Modellierung der Brown'schen Bewegung benutzt werden, kann der resultierende diskrete Zufallslauf den Abschluss des Gebietes in jedem Zeitschritt verlassen. Ein *a posteriori* Test berücksichtigt die Anwesenheit von Rändern und reduziert die dadurch bedingten zusätzlichen Fehler. Zugehörige Austrittszeiten und -punkte des Zufallslaufes werden durch Zufallszahlen mit den entsprechenden analytischen Verteilungen modelliert. Resultate numerischer Experimente zeigen, dass die schwache Konvergenzordnung des Eulerverfahrens wieder erreicht wird. Zusätzlich zeigen unsere Resultate die Überlegenheit des vorgeschlagenen Algorithmus im Vergleich zu anderen Methoden.

Beide Methoden sind ideal geeignet für parallele Simulationen. Numerische Tests in bis zu 128 Dimensionen wurden auf dem Beowulf-Cluster der ETH gerechnet.

Contents

List of Figures

Figures in Part I (Walk on cubes)

Figures in Part II (An exit probability approach)

List of Tables

Tables in Part I (Walk on cubes)

Tables in Part II (An exit probability approach)

Chapter 1

Introduction

In many applications, the dimension of the underlying space can be much larger than three. For instance, in financial applications or many body simulations, the underlying space may be of dimension 50 or more. When the dimension n of the configuration space is high, Monte-Carlo simulations [Mad02] become most attractive. They can be used not only for simulations in molecular dynamics but also for the solution of partial differential equations (PDEs) where they rely on the simulation of some stochastic differential equations (SDEs). This connection between PDEs and SDEs goes back to Feynman, who derived the path integral formalism in quantum mechanics. There, special attention is given to the solution of Schrödinger's equation [FH65]. His ideas were put on a solid mathematical basis by Kac, resulting in the famous *Feynman-Kac* formula. Many extensions to different problems in the theory of PDEs where presented over the years, see e.g. [Fre85].

In this thesis, we explore the Dirichlet problem in high dimensions using variants of the Feynman-Kac representation. What is involved is to simulate some SDEs, functionals of which form the Feynman-Kac formula. In simulation, most of the functionals turn out to be simple averages of multiple realizations of functions and their integrals of independent continuous sample paths. This independence makes the Feynman-Kac formula an ideal candidate for parallel simulations. Parallelizing the code is trivial and the resulting communication overhead is neglegible. Another big advantage is its simplicity which remains when the dimension increases. Also, the complexity of resulting algorithms usually scales linearly when the dimension of the problem increases. Furthermore, implementation is essentially no more complicated in higher dimensions than it is in two dimensions.

For the Dirichlet problem, the boundary condition makes the solution unique. The numerical solution of SDEs driven by a Brownian motion is nowadays standard and corresponding techniques are well documented [Mil95a, KP92]. A main problem remains,

though, when dealing with Dirichlet problems: Finding the boundary and determining the exit time and point of the diffusion process when it crosses the boundary. In particular, the simulated path should not follow an excursion across the boundary. Back in the fifties, Muller proposed the first solution to this problem, a random walk on spheres to solve Laplace's equation [Mul56]. Also promising for numerical approximation is another approach using random walks exclusively on the boundary, see [Sab91]. A similar approach uses corresponding Green functions, see e.g. [HMG03].

Depending on the problem, using the Feynman-Kac formula directly, can have its advantages too. Milstein et al. followed this approach in a series of papers, resulting in methods for both elliptic and parabolic PDEs in a bounded domain [MR93, Mil95b, MT02]. The basis was the approximation of a diffusion in a bounded (bounded in space or in space-time respectively) domain [Mil96, Mil97, Mil98, MT99]. These methods extend the basic walk on spheres of Muller to general PDEs and lead to higher order algorithms. One drawback of these methods is that due to the necessity of ending on the surface of a sphere at the end of a step, intermediate steps are hard to construct. These methods are thus not amenable for Runge-Kutta integration schemes and the higher order comes at the cost of inevitable derivatives in the algorithm. Our first method does not show this problem of a conditioning at the end of a step. Starting from that method, the construction of Runge-Kutta type methods, although not included in this thesis, should be straight forward. We finally note that in one dimension, the walk on spheres reduces to a simple two-point formulation.

An important ingredient in the Feynman-Kac formula is the involved expectation. Therefore, in numerical simulation approximations have to be accurate only in a weak sense. This means, loosely speaking, that only the expectations of above mentioned functionals have to be close for good convergence. Individual paths resulting from simulation can, on the other hand, be very different from the paths solving the initial SDE. We take advantage of this pecularity throughout this thesis.

Often, random numbers with a Gaussian distribution are used for the increments of Brownian motion. In the presence of boundaries, this leads (without some further thoughts) to very poor approximations. Corresponding algorithms converge only with the square root of the step size. To overcome this problem, we use crude discrete approximations for the Brownian motion instead of using increments distributed uniformly on a sphere (as is the case in the walk on spheres). Taking a simple three-point random number, this leads to a random walk on cubes.

Weak approximation means also that only the distributions have to be correct. We take advantage of this fact and *sample* appearing random variables from their analytical densities, leading to our second approach. There, we use again above mentioned Gaussian random numbers, but here additional work is needed to restore good convergence properties which were lost due to the presence of boundaries.

Outline

We give a brief outline of this thesis: In Chapter 2 we review basic results which are essential for our work. This includes the Feynman-Kac formula for Dirichlet's problem (in Section 2.1) and the stochastic analogy to ordinary Taylor expansion (in Section 2.2). After these preliminaries, we will present two algorithms for solving high dimensional Dirichlet problems. They are presented in two separate parts:

We start in Part I with a method which uses a bounded approximation for the increments of the driving Brownian motion. The construction of the algorithm follows in Chapter 3, and then we discuss some of its properties in Chapter 4. Numerical experiments in hyperspherical domains in Chapter 5 show its efficiency independent of the dimension of the underlying problem. In this chapter we begin with a numerical error analysis for a simple Poisson equation with unit inhomogeneity (in Section 5.2). Then we proceed to a study where we compare our *walk on cubes* with Milstein's *walk on spheres* (in Section 5.3). Chapter 6 concludes this part. Portions of this first part have already been published in [BP03].

In Part II we show how to apply the simplest numerical scheme to simulate SDEs (the Euler scheme) when approximating solutions to Dirichlet's problem. We start with the derivation of the method in one dimension in Chapter 7. There, we review the known approaches for the simulation of killed diffusions (in Section 7.1), and extend these to an approach for stopped diffusions (in Section 7.2). Results of numerical experiments show the efficiency of the proposed method for Brownian motion (in Section 7.3) and for a general one dimensional diffusion (in Section 7.4). In Chapter 8 these ideas presented for the one dimensional situation are extended to those for higher dimensional settings. Initially, approximation of the first exit *time* is shown (in Section 8.1) and then the question of approximating the first exit *point* is adressed (in Section 8.2). Results of numerical experiments are given in Sections 8.1.2 and 8.2.2. We conclude this part in Chapter 9.

A summary of our notation can be found beginning on page 127.

Acknowledgements

Firstly, I would like to thank Dr. Wesley P. Petersen for the immense support I received while working on this thesis. Not only did I learn a lot about mathematics and could I profit from his enormous experience in scientific computing, but I also found an excellent mentor for my first steps in the scientific community. He introduced me to numerous people working in our area of interest. I express him my deepest gratitude for the great time we have had working together. Our discussions were much more far-reaching than mathematics, involved many other important and interesting topics and took place not only in the office but also during our joint trips to conferences around the world and when regaining strength in nice restaurants.

I also thank Prof. Rolf Jeltsch. Without his willingness to support and supervise this project it would never have been possible.

Prof. Peter Kloeden from Frankfurt agreed to co-advise this thesis. For this and for many encouraging discussions in the past few years I would like to thank him very much.

I had the opportunity to discuss many problems with Prof. G. Milstein from WIAS in Berlin. He always found time to help me and I could always profit from his huge experience and extensive knowledge in the subject. I thank him for his valuable time that he shared with me.

I have come to know the encouragement of many other people who I had the possibility to enjoy mathematical discussions with. I list some of them alphabetically. Dr. Evelyn Buckwar (thanks for helping me with nasty details and for organizing my stays in Berlin!), Dr. Grant Lythe (special thanks for the nice presentation in Geneva which led to Part II!), Prof. Jonathan Mattingly (I always enjoyed our discussions while exploring down-town at night!).

I spent a great deal of my working time in the office. A very special note of thanks goes to Radu Todor for making this time so enjoyable. His help when I got stucked with a mathematical problem is equally unforgettable as the numerous laughters during lunch, the occasions having a beer or the workouts in the gym. At a later stage, Dr. Paul Ledger joined us in the E-floor. Many thanks also to him not only for the great friendship we developed but also and especially for his offer to proofread an early draft of this work.

Many thanks also go to Anna Schneebeli who helped to improve not only the quality of this work but of my life in general significantly. Clearly, all errors still present are mine.

Finally, I would like to thank all the other members of the Seminar of Applied Mathematics for the good time I had while working as an assistant.

This work was supported in part under the TH-Gesuch funds 0-20981-2.

Chapter 2

Prerequisites

We first recall some results on the connection between elliptic partial differential equations and stochastic differential equations. The representation of the solution to such PDEs by a stochastic expectation of certain functionals of the solution to a connected SDE is the indispensable basis of this work.

Moreover, to numerically approximate the above mentioned representation of the solution, another important tool is essential: the extension of classical Taylor expansions of the solution of an ordinary differential equation to the SDE case.

2.1 Probabilistic representation of the solution of Dirichlet's problem

Let L be the linear second order operator acting on functions $u \in C^2(\mathbb{R}^n, \mathbb{R})$ defined as (we frequently use summation convention where repeated indices are summed over, see p.127)

$$Lu(x) = \left[\frac{1}{2}a^{ij}(x)\partial_i\partial_j + b^i(x)\partial_i\right]u(x). \tag{2.1}$$

Let D be a domain in n-space and denote its boundary by ∂D. We will assume that the domain D is simply connected, bounded and that every $x_0 \in \partial D$ is a regular boundary point for the operator L (see [Fre85, Section 2.3]).

Additionally, we assume throughout this section that for $i, j = 1, \ldots, n$ and for any $x \in D$, the coefficients $a^{ij}(x)$ and $b^i(x)$ of the operator L are bounded and Lipschitz continuous. Furthermore, the characteristic form $v^i a^{ij}(x) v^j$ is assumed positive uniformly in x, and we do not allow the operator L to degenerate.

Remark 2.1. The major part of this work is devoted to the simpler case where the matrix $a(x)$ is the identity matrix and therefore L is the elliptic operator

$$L = \frac{1}{2} \triangle + b^i(x)\partial_i.$$

Let $a(x) = (a^{ij}(x))_{ij=1,\dots,n}$ denote the diffusion matrix of L and let $\sigma = \sigma(x)$ be a matrix satisfying $\sigma\sigma^T = a(x)$ (i.e. σ is some variant of $a^{\frac{1}{2}}$). If the matrix $a(x)$ does not degenerate and has Lipschitz continuous elements (see the assumptions above), then, such a factorization exists, and furthermore the $\sigma^{ij}(x)$ can be chosen to be Lipschitz [Fre85, Section 3.2]. Further let $b(x) = (b^i(x))_{i=1,\dots,n}$ denote the corresponding drift vector and consider the autonomous SDE

$$\mathrm{d}X(t) = b(X(t))\,\mathrm{d}t + \sigma(X(t))\,\mathrm{d}W(t). \tag{2.2a}$$

We denote by $X_x(t)$ solutions to (2.2a) with initial condition

$$X(0) = x \in \mathbb{R}^n. \tag{2.2b}$$

In (2.2a), $W(t)$ is an n-dimensional Wiener process, i.e. an n-vector whose components are n independent one-dimensional Wiener processes. (2.2) is to be understood in integral form as

$$X_x(t) = x + \int_0^t b(X_x(s))\,\mathrm{d}s + \int_0^t \sigma(X_x(s))\,\mathrm{d}W(s); \tag{2.3}$$

the last summand in (2.3) being a stochastic integral in the sense of Itô [IW89, RY91, RW00a, RW00b].

Consider the following boundary value problem (BVP)

$$Lu(x) + c(x)u(x) + g(x) = 0, \qquad x \in D \tag{2.4a}$$

$$\lim_{x \to x_0} u(x) = \psi(x_0), \qquad x_0 \in \partial D. \tag{2.4b}$$

As we will need even stronger assumptions later on, there is no harm now in assuming the following: The functions $g(x)$, $c(x)$ are continuous and bounded for all $x \in D$. In addition, $c(x) \leq 0$ and $\psi(x)$ is continuous and bounded for all $x \in \partial D$.

The solution of the BVP (2.4) can be represented as a generalized solution in a probabilistic sense. To this end, we introduce for $x \in \overline{D}$ the first exit time of the process $X_x(t)$ solving the SDE (2.2) from the domain D,

$$\tau_D^x = \inf\{t \geq 0 \,:\, X_x(t) \notin D\}. \tag{2.5}$$

If the trajectory $X_x(t)$ never leaves D, we set $\tau_D^x = \infty$. Due to the continuity of the trajectories of $X_x(t)$, it holds that τ_D^x is also the first passage time to the boundary ∂D,

$$\tau_D^x = \inf\{t \geq 0 \,:\, X_x(t) \in \partial D\}\,.$$

If D is bounded we have $\mathbb{P}[\tau_D^x < \infty] = 1$. If additionally ∂D is smooth and L is non-degenerate with sufficiently smooth coefficients, we have the even stronger assertion that $\mathbb{E}[\tau_D^x] < K_D < \infty$. Under these assumptions, $u(x)$ has the following representation:

Theorem 2.2 (Freidlin). *If $u(x)$, $x \in D$, is a bounded solution of problem (2.4) with bounded, continuous first and second order partial derivatives in any interior sub-domain $\widetilde{D} \subset D$, then $u(x)$ has the following representation:*

$$\begin{aligned} u(x) =& \mathbb{E}\left[\psi(X_x(\tau_D^x)) \exp\left(\int_0^{\tau_D^x} c(X_x(s))\,\mathrm{d}s \right) \right. \\ &\left. + \int_0^{\tau_D^x} g(X_x(t)) \exp\left(\int_0^t c(X_x(s))\,\mathrm{d}s \right) \mathrm{d}t \right]. \end{aligned} \tag{2.6}$$

The proof of this Theorem can be found in the monograph of Freidlin [Fre85]. There, more general situations such as a degenerating operator L or non-negative characteristic forms $v^i a^{ij}(x) v^j$ are also treated. Furthermore, this book considers situations of more general (possibly unbounded) domains and the peculiarities of boundaries where not all boundary points are regular. In addition, the existence of a (possibly symmetric and Lipschitz) factorization of $a(x)$ is proven for different classes of matrices $a(x)$.

Such details, although very important, are beyond the scope and not the goal of this work: Instead, we will concentrate on the numerical approximation of $u(x)$ in cases where a unique solution which can be represented in the form (2.6) exists.

For the numerical approximation of $u(x)$ via (2.6) and (2.2), the following probabilistic representation has been shown to be the most convenient (to simplify notation, we drop the subscript x indicating the initial condition (2.2b) and further write $\tau = \tau_D^x$),

$$u(x) = \mathbb{E}\left[\psi(X(\tau))Y(\tau) + Z(\tau)\right] \tag{2.7}$$

where $X(t)$, $Y(t)$ and $Z(t)$ are the solutions to the system

$$\mathrm{d}X(t) = \sigma(X(t))\,\mathrm{d}W(t) \tag{2.8a}$$

$$\mathrm{d}Y(t) = c(X(t))Y(t)\,\mathrm{d}t + \left(\sigma^{-1}(X(t))b(X(t))\right)^T Y(t)\,\mathrm{d}W(t) \tag{2.8b}$$

$$\mathrm{d}Z(t) = g(X(t))Y(t)\,\mathrm{d}t \tag{2.8c}$$

for $0 \leq t \leq \tau$. The initial conditions are

$$X(0) = x \in D, \quad Y(0) = 1 \quad \text{and} \quad Z(0) = 0. \tag{2.9}$$

In both (2.8) and (2.9), X is an n-vector and Y and Z are scalars. The above transformations are based on Girsanov's Theorem [Gir60] and the details can be found in [Mil97].

We note that the necessary inversion of the diffusion matrix $\sigma(x)$ might be problematic in practice, especially if the dimension n of the underlying problem is high. However, in the case of the Laplace operator, system (2.8) takes the simpler form

$$\mathrm{d}X(t) = \mathrm{d}W(t) \tag{2.10a}$$

$$\mathrm{d}Y(t) = c(X(t))Y(t)\,\mathrm{d}t + Y(t)b(X(t))^T\,\mathrm{d}W(t) \tag{2.10b}$$

$$\mathrm{d}Z(t) = g(X(t))Y(t)\,\mathrm{d}t \tag{2.10c}$$

and in particular no matrix inversion is required.

For the sake of simplicity, and to keep the calculations in what follows to a reasonable length, we will concentrate on this situation. Some remarks on extensions to more general cases will be given.

Remark 2.3. Both systems (2.8) and (2.10) are examples of autonomous systems of SDEs. The fact that all the coefficients (of the BVP (2.4) and of the corresponding SDEs (2.8) or (2.10)) do not depend on t explicitly will simplify the calculations in what follows.

2.2 Stochastic Taylor expansion

Any numerical algorithm based on the probabilistic representation (2.7) (which is equivalent to (2.6)) to approximate $u(x)$, relies on a numerical approximation of the solution to the system of SDEs (2.8) or (2.10) respectively. The aim of this section is to work out a one step approximation for this system. To accomplish this we first review the concept of a stochastic Taylor expansion (Wagner-Platen expansion [WP78, PW82]) for the general autonomous d-dimensional SDE driven by m independent Wiener processes,

$$d\mathcal{X}^i(t) = B^i(\mathcal{X}(t))\,\mathrm{d}t + \sum_{r=1}^{m} S^{ir}(\mathcal{X}(t))\,\mathrm{d}W^r(t), \quad i = 1, \ldots, d \tag{2.11}$$

with initial condition $\mathcal{X}(t_0) = \mathcal{X}_0$. Here, $B^i : \mathbb{R}^d \to \mathbb{R}$ and $S^{ir} : \mathbb{R}^d \to \mathbb{R}$ for $i = 1, \ldots, d$ and $r = 1, \ldots, m$. Conditions on the smoothness etc. of these coefficients will be given later. Introducing the vectors $B(\mathcal{X})$, $W(t)$ and the matrix $S(\mathcal{X})$ as

$$B(\mathcal{X}) = (B^i(\mathcal{X}))_{i=1,\ldots,d}, \quad W(t) = (W^r(t))_{r=1,\ldots,m} \quad \text{and} \quad S(\mathcal{X}) = (S^{ir}(\mathcal{X}))_{\substack{i=1,\ldots,d \\ r=1,\ldots,m}}$$

we can rewrite the SDE (2.11) in matrix-vector notation as

$$d\mathcal{X}(t) = B(\mathcal{X}(t))\,\mathrm{d}t + S(\mathcal{X}(t))\,\mathrm{d}W(t).$$

After having seen the general case we will, at a later stage, explicitly calculate the specific expansion for the solution of the system (2.10). Based on that expansion, we will then find several one step approximations.

2.2.1 Preliminaries

Following the notation used in the monograph of Kloeden and Platen, we introduce some vocabulary and notation [KP92, Sections 5.2-5.4], see also [Mil95a, Section 2.2]. To our knowledge, it was first derived by Wagner and Platen in [WP78], see also [PW82].

2.2.1-a) Multi-indices

For $j_i \in \{0, 1, \ldots, m\}$, we call for $i \in \{1, 2, \ldots, l\}$ the l-vector $\alpha = (j_1, \ldots, j_l)$ a *multi-index*. We denote by $l = l(\alpha)$ its length and by $n(\alpha)$ the number of its components j_i which are equal to zero. For the sake of completeness, also let v denote the (empty) multi-index of length zero ($l(v) = 0$, $v = ()$). The set $\mathcal{M}$ contains all multi-indices (including v). For $\alpha = (j_1, \ldots, j_l) \neq v$ we define $-\alpha$ (removing j_1) and $\alpha-$ (removing j_l) by

$$-\alpha = (j_2, \ldots, j_l) \quad \text{and} \quad \alpha- = (j_1, \ldots, j_{l-1}).$$

2.2.1-b) Hierarchical and remainder set

A non-empty subset $\mathcal{A} \subset \mathcal{M}$ is called a *hierarchical set* if the conditions

$$\sup_{\alpha \in \mathcal{A}} l(\alpha) < \infty \quad \text{and} \quad -\alpha \in \mathcal{A} \quad \forall \alpha \in \mathcal{A} \setminus \{v\}$$

hold. The *remainder set* $\mathcal{B}$ of the hierarchical set $\mathcal{A}$ is defined as

$$\mathcal{B}(\mathcal{A}) = \{\alpha \in \mathcal{M} \setminus \mathcal{A} \ : \ -\alpha \in \mathcal{A}\}.$$

2.2.1-c) Multiple Itô integrals

Using the notation of multi-indices, we are able to define multiple Itô integrals in a recursive way:

$$I_\alpha[f]_{t,t+h} = \begin{cases} f(t+h), & l(\alpha) = 0, \\ \int_t^{t+h} I_{\alpha-}[f(\cdot)]_{t,s}\,\mathrm{d}s, & l(\alpha) \geq 1 \quad \text{and} \quad j_l = 0, \\ \int_t^{t+h} I_{\alpha-}[f(\cdot)]_{t,s}\,\mathrm{d}W^{j_l}(s), & l(\alpha) \geq 1 \quad \text{and} \quad j_l \geq 1. \end{cases} \tag{2.12}$$

Of course, we assume in (2.12) that $f(\cdot)$ is such that these integrals exist. The exact conditions on f can be found in many textbooks, for example in [KP92, Section 5.2]. As we will need only the integrals with integrand $f \equiv 1$ in a later stage, we omit these details here. For $f \equiv 1$, we will use the abbreviated notation $I_{\alpha,h} = I_\alpha[1]_{t,t+h}$.

2.2.1-d) Coefficient functions

The notation becomes very convenient not only for multiple Itô integrals, but also for the coefficient functions appearing in a stochastic Taylor expansion.

Remembering the SDE (2.11) with coefficients B^i and S^{ij}, $i = 1, \ldots, d$, $j = 1, \ldots, m$, we define the operators

$$L = B^i(\mathcal{X})\partial_i + \frac{1}{2}S^{jr}(\mathcal{X})S^{kr}(\mathcal{X})\partial_j\partial_k = B^i(\mathcal{X})\partial_i + \frac{1}{2}\left(S(\mathcal{X})S^T(\mathcal{X})\right)^{jk}\partial_j\partial_k \tag{2.13a}$$

$$\Lambda_r = S^{ir}(\mathcal{X})\partial_i = \left\langle S^{\cdot r}, \frac{\partial}{\partial \mathcal{X}} \right\rangle, \quad r = 1, \ldots, m. \tag{2.13b}$$

For a generic n-vector $x = (x^1, \ldots, x^n)^T$, we write Lx for the vector $(Lx^1, \ldots, Lx^n)^T$. $\Lambda_r x$ is defined analogously.

For $\alpha \in \mathcal{M}$ and $f \in C^{l(\alpha)+n(\alpha)}(\mathbb{R}^d, \mathbb{R})$ we define the *coefficient functions* f_α recursively as

$$f_\alpha = \begin{cases} f, & l(\alpha) = 0 \\ Lf_{-\alpha}, & l(\alpha) \geq 1 \quad \text{and} \quad j_1 = 0 \\ \Lambda_{j_1} f_{-\alpha}, & l(\alpha) \geq 1 \quad \text{and} \quad j_1 \geq 1. \end{cases} \tag{2.14}$$

With these notations, we can now write down the stochastic Taylor expansion of a sufficiently smooth function f of the solution of the SDE (2.11) in a compact form.

2.2.2 Itô-Taylor expansion for the solution of the SDE (2.11)

Let $f : \mathbb{R}^d \to \mathbb{R}$ be sufficiently smooth. Then, the following expansion of $f(\mathcal{X}(t))$ for the solution of the SDE (2.11), the d-dimensional Itô process,

$$\mathcal{X}(t+h) = \mathcal{X}(t) + \int_t^{t+h} B(\mathcal{X}(s))\,\mathrm{d}s + \int_t^{t+h} S(\mathcal{X}(s))\,\mathrm{d}W(s), \quad t \le t+h \le T < \infty$$

exists.

Theorem 2.4 ([KP92, Thm 5.5.1]). *Let $\mathcal{A}$ be a hierarchical set, then*

$$f(\mathcal{X}(t+h)) = \sum_{\alpha\in\mathcal{A}} I_\alpha[f_\alpha(\mathcal{X}(t))]_{t,t+h} + \sum_{\alpha\in\mathcal{B}(\mathcal{A})} I_\alpha[f_\alpha(\mathcal{X}(\cdot))]_{t,t+h}, \tag{2.15}$$

provided all the needed derivatives of f, B and S exist and all the needed $I_\alpha[\cdot]_{t,t+h}$ exist.

In what follows, we will call the second sum in (2.15) the remainder and write $R = R[f]_h$,

$$R[f]_h = \sum_{\alpha\in\mathcal{B}(\mathcal{A})} I_\alpha[f_\alpha(\mathcal{X}(\cdot))]_{t,t+h}.$$

Choosing f to be the identity map, $f(x) = x$, an expansion for the solution $\mathcal{X}(t)$ itself follows from (2.15). Truncation of the resulting series by neglecting R then leads to a specific one step approximation (corresponding to a selected hierarchical set $\mathcal{A}$). To monitor convergence of the approximation to the true solution $\mathcal{X}(t)$, many different criteria are widely used. With the representation (2.7) in mind the concept of *weak convergence* is clearly appropriate here.

Definition 2.5 (Weak convergence). Denote by $\overline{\mathcal{X}}(t)$ an approximation of $\mathcal{X}(t)$ over the time interval $t \in [t_0, t_0+T]$. We say that $\overline{\mathcal{X}}(t)$ approximates $\mathcal{X}(t)$ *in the weak sense* with (weak) order of accuracy p (or of $\mathcal{O}(h^p)$) if for all $f : \mathbb{R}^d \to \mathbb{R}$ from a sufficiently large class of functions $\exists C_f$ (depending on f) such that

$$|\mathbb{E}[f(\overline{\mathcal{X}}(t))] - \mathbb{E}[f(X(t))]| \le C_f h^p \quad \text{for} \quad t_0 \le t \le t_0 + T \tag{2.16}$$

A class of test functions which has been shown to be convenient, is the space of l times continuously differentiable functions which, together with their partial derivatives up to (including) order l, have polynomial growth. More precisely, we write $f \in C_P^l(\mathbb{R}^d, \mathbb{R})$, where

$$C_P^l(\mathbb{R}^d, \mathbb{R}) = \left\{ f \in C^l(\mathbb{R}^d, \mathbb{R}) \;:\; f,\ \partial^\alpha f \in F,\ l(\alpha) \le l \right\}.$$

Hereby, a function $f(x)$ belongs to the class F ($f \in F$), if $\exists K > 0$ and an integer $\kappa \geq 1$ such that

$$|f(x)| \leq K\left(1 + |x|^{\kappa}\right), \quad \forall x \in R^d.$$

Remark 2.6. If $\exists K > 0$ and f is such that $|f(x)|^2 \leq K^2(1 + |x|^2)$ for all $x \in \mathbb{R}^d$, we say that f satisfies a *linear growth bound*. On a finite domain D, polynomial growth implies linear growth. Indeed, assume $\exists C_D > 0$ with $|x| \leq C_D < \infty$ for all $x \in D$, and assume that $f \in F$, then

$$|f(x)|^2 \leq K^2(1 + |x|^{\kappa})^2 \leq K^2(1 + 2C_D^{\kappa} + C_D^{2\kappa}) = \tilde{K}^2 \leq \tilde{K}^2(1 + |x|^2)$$

with $\tilde{K} = K(1 + C_D^{\kappa}) < \infty$. Furthermore, from this estimate we see that if $f \in C_P^l(D, \mathbb{R})$ with D bounded, then f is bounded.

Consider for $\beta = 0, 1, \ldots$ the hierarchical set $\Gamma_\beta = \{\alpha \in \mathcal{M} \ : \ l(\alpha) \leq \beta\}$ and the truncated Itô-Taylor expansion

$$f(\mathcal{X}(t+h)) - R[f] = \sum_{\alpha \in \Gamma_\beta} I_\alpha[f_\alpha(\mathcal{X}(t))]_{t,t+h} = \sum_{\alpha \in \Gamma_\beta} f_\alpha(\mathcal{X}(t)) I_{\alpha,h}.$$

Let $l = 2(\beta + 1)$. Then, the one step approximation $\sum_{\alpha \in \Gamma_\beta} f_\alpha(\mathcal{X}(t)) I_{\alpha,h}$ with f_α corresponding to the identity map is of weak order $\beta + 1$. We even have the stronger result, summarized in the following theorem.

Theorem 2.7 ([KP92, Thm 5.11.1]). *Let $t_0 = 0$, $h > 0$ and $0 < T < \infty$ be given. Suppose that $B^i(\mathcal{X})$ and $S^{ij}(\mathcal{X}) \in C_P^l(\mathbb{R}^d, \mathbb{R})$ satisfy a Lipschitz condition*

$$|B^i(x) - B^i(y)| + \sum_{j=1,\ldots,m} |S^{ij}(x) - S^{ij}(y)| \leq C|x - y|, \quad \forall x, y \in \mathbb{R}^d, \quad i = 1, \ldots, d, \tag{2.17}$$

and linear growth bounds for $i = 1, \ldots, d$ and $j = 1, \ldots, m$. Then, $\forall f \in C_P^l(\mathbb{R}^d, \mathbb{R})$ $\exists K > 0, \kappa > 0$, such that

$$\sup_{0 \leq h \leq T} \left| \mathbb{E}[f(\mathcal{X}(h))] - \mathbb{E}\Big[f\Big(\sum_{\alpha \in \Gamma_\beta} f_\alpha(\mathcal{X}(0)) I_{\alpha,h} \Big) \Big] \right| \leq K(1 + |\mathbb{E}[\mathcal{X}(0)]|^{2\kappa}) T^{\beta+1}. \tag{2.18}$$

We now explicitly calculate the expansion of $\mathcal{X}$ for $\beta = 2$. We have

$$\begin{aligned} \Gamma_2 &= \{\alpha \in \mathcal{M} \ : \ l(\alpha) \leq 2\} \\ &= \{v, (0), (i), (0,i), (i,0), (i,j), (0,0)\} \end{aligned}$$

and

$$\begin{aligned}\mathcal{B}(\Gamma_2) &= \{\alpha \in \mathcal{M} \,:\, l(\alpha) = 3\} \\ &= \{(0,0,i),(i,0,j),(0,i,0),(i,j,0),(0,i,j),(i,j,k),(0,0,0),(i,0,0)\},\end{aligned}$$

where $i, j, k = 1, \ldots, m$.

Setting $f(x) = x$ in (2.15) and inserting the corresponding coefficient functions (see (2.14)), we get the following expansion for $\mathcal{X}(t+h)$:

$$\begin{aligned}\mathcal{X}(t+h) =& \mathcal{X}(t) + (L\mathcal{X})(t)\int_t^{t+h} \mathrm{d}s_1 + \sum_{i=1}^{m}(\Lambda_i\mathcal{X})(t)\int_t^{t+h} \mathrm{d}W^i(s_1) \\ &+ (LL\mathcal{X})(t)\int_t^{t+h}\int_t^{s_1} \mathrm{d}s_2\,\mathrm{d}s_1 + \sum_{i=1}^{m}(\Lambda_i L\mathcal{X})(t)\int_t^{t+h}\int_t^{s_1} \mathrm{d}W^i(s_2)\,\mathrm{d}s_1 \\ &+ \sum_{i=1}^{m}(L\Lambda_i\mathcal{X})(t)\int_t^{t+h}\int_t^{s_1} \mathrm{d}s_2\,\mathrm{d}W^i(s_1) \\ &+ \sum_{i=1}^{m}\sum_{j=1}^{m}(\Lambda_j\Lambda_i\mathcal{X})(t)\int_t^{t+h}\int_t^{s_1} \mathrm{d}W^j(s_2)\,\mathrm{d}W^i(s_1) \quad + \quad R\end{aligned} \tag{2.19}$$

where the remainder R in integral form is

$$
\begin{aligned}
R = {} & \int_t^{t+h}\int_t^{s_1}\int_t^{s_2} (LLL\mathcal{X})(s_3)\,\mathrm{d}s_3\,\mathrm{d}s_2\,\mathrm{d}s_1 + \sum_{i=1}^{m}\int_t^{t+h}\int_t^{s_1}\int_t^{s_2} (\Lambda_i LL\mathcal{X})(s_3)\,\mathrm{d}W^i(s_3)\,\mathrm{d}s_2\,\mathrm{d}s_1 \\
& + \sum_{i=1}^{m}\int_t^{t+h}\int_t^{s_1}\int_t^{s_2} (L\Lambda_i L\mathcal{X})(s_3)\,\mathrm{d}s_3\,\mathrm{d}W^i(s_2)\,\mathrm{d}s_1 \\
& + \sum_{i=1}^{m}\sum_{j=1}^{m}\int_t^{t+h}\int_t^{s_1}\int_t^{s_2} (\Lambda_j\Lambda_i L\mathcal{X})(s_3)\,\mathrm{d}W^j(s_3)\,\mathrm{d}W^i(s_2)\,\mathrm{d}s_1 \\
& + \sum_{i=1}^{m}\int_t^{t+h}\int_t^{s_1}\int_t^{s_2} (LL\Lambda_i\mathcal{X})(s_3)\,\mathrm{d}s_3\,\mathrm{d}s_2\,\mathrm{d}W^i(s_1) + \\
& + \sum_{i=1}^{m}\sum_{j=1}^{m}\int_t^{t+h}\int_t^{s_1}\int_t^{s_2} (\Lambda_j L\Lambda_i\mathcal{X})(s_3)\,\mathrm{d}W^j(s_3)\,\mathrm{d}s_2\,\mathrm{d}W^i(s_1) \\
& + \sum_{i=1}^{m}\sum_{j=1}^{m}\int_t^{t+h}\int_t^{s_1}\int_t^{s_2} (L\Lambda_j\Lambda_i\mathcal{X})(s_3)\,\mathrm{d}s_3\,\mathrm{d}W^j(s_2)\,\mathrm{d}W^i(s_1) \\
& + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m}\int_t^{t+h}\int_t^{s_1}\int_t^{s_2} (\Lambda_k\Lambda_j\Lambda_i\mathcal{X})(s_3)\,\mathrm{d}W^k(s_3)\,\mathrm{d}W^j(s_2)\,\mathrm{d}W^i(s_1).
\end{aligned}
\tag{2.20}
$$

Using $(L\mathcal{X})(\cdot) = B(\mathcal{X}(\cdot))$ and $(\Lambda_i\mathcal{X})(\cdot) = S^{\cdot i}(\mathcal{X}(\cdot))$, we get the truncated expansion

for $\mathcal{X}(t+h)$ as (using summation convention)

$$\begin{aligned}\overline{\mathcal{X}}(t+h) = \mathcal{X}(t) + B(\mathcal{X}(t))h + S^{\cdot i}(\mathcal{X}(t)) \int_t^{t+h} \mathrm{d}W^i(s_1) \\ + \Lambda_i B(\mathcal{X}(t)) \int_t^{t+h}\int_t^{s_1} \mathrm{d}W^i(s_2)\,\mathrm{d}s_1 + LS^{\cdot i} \int_t^{t+h}\int_t^{s_1} \mathrm{d}s_2\,\mathrm{d}W^i(s_1) \qquad (2.21) \\ + \Lambda_i S^{\cdot j}(\mathcal{X}(t)) \int_t^{t+h}\int_t^{s_1} \mathrm{d}W^i(s_2)\,\mathrm{d}W^j(s_1) + LB(\mathcal{X}(t))\frac{h^2}{2}.\end{aligned}$$

Equation (2.21) should be read as a definition of $\overline{\mathcal{X}}$ (the left hand side differs from the right hand side by the higher order terms R, see (2.20)).

We will later (see Section 3.1) apply this approximation to the system (2.10) to get a one step approximation suitable for the Dirichlet problem.

Part I

Walk on cubes

Chapter 3

Construction of the algorithm

We construct an algorithm which allows the approximation of τ and $X(\tau)$ using *bounded* models for the $I_{\alpha,h}$ in the one step approximation (3.4). The fact that the chosen approximation $\hat{I}_{(i),h}$ satisfies $|\hat{I}_{(i),h}| \leq C_h$ with known constant C_h (depending on h), allows us to prevent an excursion of the approximate process $\overline{X}$ from D. Furthermore, the resulting random walk can approach ∂D as close as desired, yet without ever leaving $\overline{D}$. This then allows us to find a good approximation to both the exit time τ and to the true exit point $X(\tau) \in \partial D$.

3.1 One step approximation for the system (2.10) connected to the Dirichlet problem

System (2.10) is of the form of the SDE (2.11) with $d = n + 2$ and $m = n$. In more detail, we are looking for a solution vector $\mathcal{X} = (X^1, \ldots, X^n, Y, Z)^T$ of the SDE with coefficients

$$B(X^1, \ldots, X^n, Y, Z) = B(X, Y) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ c(X)Y \\ g(X)Y \end{pmatrix} \tag{3.1a}$$

and

$$S(X^1,\dots,X^n,Y,Z)=S(X,Y)=\begin{pmatrix}1&&\\&\ddots&\\&&1\\Yb^1(X)&\cdots&Yb^n(X)\\0&\cdots&0\end{pmatrix}. \tag{3.1b}$$

Here, B is an $(n+2)$-vector whilst S is an $((n+2)\times n)$-matrix. Inserting this into the operators (2.13) yields

$$\begin{aligned}L&=\frac{1}{2}\triangle+Y\left(b^k(X)\partial_k\partial_Y+\frac{1}{2}Yb^k(X)b^k(X)\partial_Y^2+c(X)\partial_Y+g(X)\partial_Z\right),\\ \Lambda_i&=\partial_i+Yb^i(X)\partial_Y,\quad i=1,\dots,n.\end{aligned}$$

We calculate the derivatives needed in (2.21) and introduce some abbreviations to simplify notation. We obtain for Y

$$\begin{aligned}&\Lambda_iY=Yb^i(X),\quad LY=Yc(X),\quad \Lambda_i\Lambda_jY=Y(\partial_ib^j(X)+b^i(X)b^j(X))=:Y\gamma^{ij}(X),\\ &\Lambda_iLY=Y\Big(\partial_ic(X)+b^i(X)c(X)\Big)=:Y\gamma^{i0}(X),\\ &L\Lambda_iY=Y\left(\frac{1}{2}\triangle b^i(X)+\left\langle b(X),\frac{\partial}{\partial X}b^i(X)\right\rangle+c(X)b^i(X)\right)=:Y\gamma^{0i}(X),\\ &LLY=Y\left(\frac{1}{2}\triangle c(X)+c^2(X)+\left\langle b(X),\frac{\partial}{\partial X}c(X)\right\rangle\right)=:Y\gamma^{00}(X).\end{aligned} \tag{3.2a}$$

Whilst for Z we get

$$\begin{aligned}&LZ=Yg(X),\quad \Lambda_iLZ=Y(\partial_ig(X)+b^i(X)g(X))=:Y\zeta^{i0}(X),\\ &LLZ=Y\left(\frac{1}{2}\triangle g(X)+\left\langle b(X),\frac{\partial}{\partial X}g(X)\right\rangle+c(X)g(X)\right)=:Y\zeta^{00}(X).\end{aligned} \tag{3.2b}$$

As we will use the identity $I_{(0,i),h}=hI_{(i),h}-I_{(i,0),h}$ (see for example [Mil95a, p.105] or the formula for integration by parts [Øks03, Theorem 4.1.5]), the required integrals in (2.19) or (2.21) respectively are, beside of $I_{v,h}=1$, $I_{(0),h}=h$ and $I_{(0,0),h}=\frac{h^2}{2}$, of the following form:

$$I_{(i),h}=\int_0^h \mathrm{d}W^i(t),\quad I_{(i,0),h}=\int_0^h\int_0^t \mathrm{d}W^i(s)\,\mathrm{d}t\quad\text{and}\quad I_{(i,j),h}=\int_0^h\int_0^t \mathrm{d}W^i(s)\,\mathrm{d}W^j(t). \tag{3.3}$$

Inserting the derivatives (3.2) together with the shorthand for the integrals (3.3) into the expansion (2.21), together with the aforementioned identity, gives the one step approximations,

$$\overline{X}^i(t+h) = X^i(t) + I_{(i),h}, \qquad i = 1, \ldots, n, \tag{3.4a}$$

$$\begin{aligned}\overline{Y}(t+h) = Y(t)\Big(1 + \Big(b^i(X(t)) + \gamma^{0i}(X(t))h\Big) I_{(i),h} + c(X(t))h + \gamma^{ij}(X(t)) I_{(i,j),h} \\ + \Big(\gamma^{i0}(X(t)) - \gamma^{0i}(X(t))\Big) I_{(i,0),h} + \gamma^{00}(X(t))\frac{h^2}{2}\Big),\end{aligned} \tag{3.4b}$$

and

$$\overline{Z}(t+h) = Z(t) + Y(t)\left(g(X(t))h + \zeta^{i0}(X(t)) I_{(i,0),h} + \zeta^{00}(X(t))\frac{h^2}{2}\right). \tag{3.4c}$$

Note that $X^i(t+h) = \overline{X}^i(t+h)$, whereas $Y(t+h) \approx \overline{Y}(t+h)$ and $Z(t+h) \approx \overline{Z}(t+h)$ in general.

To implement an algorithm based on the one step approximation (3.4), models for the integrals $I_{\alpha,h}$ are needed for $\alpha = (i)$, $\alpha = (i,0)$ and $\alpha = (i,j)$ for $i, j = 1, \ldots, n$. Suppose for the moment that corresponding models $\hat{I}_{\alpha,h}$ exist and that they model the corresponding $I_{\alpha,h}$ exactly in distribution (having the same joint distributions). Setting $X_0^i = x^i$, $Y_0 = 1$ and $Z_0 = 0$ corresponding to the initial conditions (2.9), we find successive approximations for $t = kh$, $k = 0, 1, \ldots$, as (omitting " $\bar{}$ " for the approximations to X, Y and Z)

$$X_{k+1}^i = X_k^i + \hat{I}_{(i),h}, \qquad i = 1, \ldots, n, \tag{3.5a}$$

$$\begin{aligned}Y_{k+1} = Y_k\Big(1 + \Big(b^i(X_k) + \gamma^{0i}(X_k)h\Big)\hat{I}_{(i),h} + c(X_k)h + \gamma^{ij}(X_k)\hat{I}_{(i,j),h} \\ + \Big(\gamma^{i0}(X_k) - \gamma^{0i}(X_k)\Big)\hat{I}_{(i,0),h} + \gamma^{00}(X_k)\frac{h^2}{2}\Big),\end{aligned} \tag{3.5b}$$

and

$$Z_{k+1} = Z_k + Y_k\left(g(X_k)h + \zeta^{i0}(X_k)\hat{I}_{(i,0),h} + \zeta^{00}(X_k)\frac{h^2}{2}\right). \tag{3.5c}$$

Note that in [KP92], the scheme (3.5) is called the *order 2.0 weak Taylor scheme* for the system (2.10).

Our final goal is an approximation for $u(x)$ in the form (2.7). Therefore additionally, approximations of $X(\tau)$ and of τ itself are required. Namely, the integration interval for the SDE (2.10) is $[0, \tau]$, where τ is not known or fixed, but a stopping time [RW00a, II.56]. If $\overline{X}$ denotes (again) an approximation for $X(t)$, then the quality of this approximation (in the sense of Definition 2.5) does (without further thoughts) by no means guarantee that the approximate stopping time $\overline{\tau}$ (connected to $\overline{X}$) is a "good" approximation to τ – not even in the weak sense. Furthermore, $\overline{X}(\overline{\tau})$ must lie on the boundary ∂D, as the function ψ, which has to be evaluated at $X(\tau)$ (or at $\overline{X}(\overline{\tau})$ respectively), is defined only on ∂D, see (2.4b).

3.2 Models suitable for the Dirichlet problem

Let ξ^i, $i = 1, \dots, n$, be i.i.d. (wrt. i) random variables, distributed according to the law

$$\mathbb{P}[\xi^i = a] = \begin{cases} \frac{1}{6}, & a = \pm\sqrt{3}, \\ \frac{2}{3}, & a = 0. \end{cases} \tag{3.6}$$

Let $\Xi = (\Xi^{ij})_{ij}$ be a random $(n \times n)$-array whose elements are i.i.d. distributed according to the law

$$\mathbb{P}[\Xi^{ii} = -1] = 1, \quad \mathbb{P}[\Xi^{ij} = \pm 1] = \frac{1}{2} \quad \text{for} \quad j = 1, \dots, i-1, \tag{3.7}$$

and set $\Xi^{ij} = -\Xi^{ji}$ for $j = i+1, \dots, n$ (i.e. Ξ is skew symmetric wrt. the off-diagonals, but -1 on the diagonal). Consider the following models for the integrals $I_{\alpha,h}$:

$$\hat{I}_{(i),h} = \sqrt{h}\xi^i, \quad \hat{I}_{(i,0),h} = \frac{h^{\frac{3}{2}}}{2}\xi^i \quad \text{and} \quad \hat{I}_{(i,j),h} = \frac{h}{2}\left(\xi^i\xi^j + \Xi^{ij}\right) \tag{3.8}$$

for $i, j = 1, \dots, n$.

Remark 3.1. We note that $|\hat{I}_{(i),h}| \leq \sqrt{3h}$ and that the vector $\Delta X = \sqrt{h}(\hat{I}_{(i),h})_{i=1,\dots,n}$ satisfies $|\Delta X| \leq \sqrt{3nh}$.

We have the following result (see for example [Mil95a, Lemma 8.4],[KP92, Models in (5.12.9) and Corollary 5.12.1]).

Lemma 3.2. *Under the conditions of Theorem 2.7, inequality* (2.18) *is still valid for* $\beta = 2$ *if the* $I_{\alpha,h}$ *are replaced by the models* $\hat{I}_{\alpha,h}$ *from* (3.8).

Remark 3.3. We shall see later in Section 4.2 that for the PDE (2.4a) arising in the general Dirichlet problem (2.4), the assumptions of Theorem 2.7 are not fulfilled in general. In particular, the Lipschitz condition (2.17) does not hold for the coefficients of either systems (2.8) or (2.10).

3.3 A Markov chain $(X_k)_k$ in $\overline{D}$ and a one step approximation

From the models (3.8), we are now ready to construct an algorithm to solve the SDEs (2.10) inside the bounded domain D. Introduce for $r > 0$ the interior boundary layer

$$\Gamma_r = \{x \in D \ : \ \rho(x, \partial D) < r\},$$

where

$$\rho(x, \partial D) = \inf_{x_0 \in \partial D} \{|x_0 - x|\}$$

denotes the distance of the point x to the boundary ∂D. In particular, we call $\Gamma_T = \Gamma_{\sqrt{3nh}}$ and $\Gamma_S = \Gamma_{h^2}$ tangent and stopping layers respectively.

We next construct a discrete Markov chain $(X_k)_{k\geq 0}$ approximating $X(t)$, the solution to $\mathrm{d}X(t) = \mathrm{d}W(t)$, $X(0) = x$, which satisfies $X_k \in \overline{D}$ for every $k \geq 0$. Moreover, the construction of the chain will allow an approximation for τ and in particular for $X(\tau) \in \partial D$. To simplify notation, we omit the symbol " $\bar{}$ " for the approximation.

Algorithm 3.1 Walk on cubes

Choose $h > 0$ small enough and set $X_0 \leftarrow x$, $h_0 \leftarrow h$.
for $k = 0, 1, \ldots$ **do**
  if $X_k \in \Gamma_S$ **then** {close enough to the boundary}
    $K \leftarrow k$
    STOP
  else if $X_k \in D \setminus \Gamma_T$ **then** {far away from the boundary}
    $h_{k+1} \leftarrow h$
  else {in tangent layer}
    $h_{k+1} \leftarrow \rho(X_k, \partial D)^2/(3n)$
  end if
  Generate $\xi_{k+1} = (\xi^i)_{i=1,\ldots,n}$ with ξ^i i.i.d. according to (3.6)
  Update $X_{k+1} \leftarrow X_k + h_{k+1}^{\frac{1}{2}} \xi_{k+1}$
end for
Set X_ν to a point on ∂D closest to $X_k \in \Gamma_S$
$\tau \leftarrow \sum_{k=1}^{K} h_k$

The chain $(X_k)_{k=0,\ldots,K}$ is stopped as soon as $X_k \in \Gamma_S$, i.e. if $\rho(X_k, \partial D) < h^2$. If the notation $(X_k)_{k\geq 0}$ is preferred, set $h_{k+1} = 0$ for $k \geq K$.

Remark 3.4. Note that $h_{k+1} \leq h \ \forall k = 0, 1, \ldots,$ as for $X_k \in \overline{\Gamma}_T \setminus \Gamma_S$,

$$h_{k+1} = \frac{\rho(X_k, \partial D)^2}{3n} \leq \frac{3nh}{3n} = h,$$

and if $X_k \in D \setminus \Gamma_T$, trivially $h_{k+1} = h$.

We will show later, that indeed this chain stops with probability one by proving (in Section 4.1) that the expected number of steps, $\mathbb{E}[K]$, is bounded if D is bounded.

Remark 3.5. In each step $k \to k+1$ with $X_k \notin \Gamma_S$, $X_{k+1} = X_k$ with probability $(2/3)^n$, the other $3^n - 1$ possible points are on the surface of the n-dimensional hyper-cube (see Figure 4.1 on p. 30 for an illustration in 3-space)

$$\left(X_k - \sqrt{3h_{k+1}}\ ,\ X_k + \sqrt{3h_{k+1}}\right)^n.$$

This is why we call the resulting random walk $X_0 \to X_1 \to X_2 \to \cdots \to X_K$ a *walk on cubes*.

The construction of the one step approximation for system (2.10) is now obvious. Let

$$X_0 = X(0) = x \in D, \quad Y_0 = Y(0) = 1 \quad \text{and} \quad Z_0 = Z(0) = 0. \tag{3.9}$$

Suppose h_k and K (stopping of the integration) are constructed as in Algorithm 3.1. By inserting the models (3.8) and the definitions of γ^{ij} (see (3.2a)) and of ζ^{ij} (see (3.2b)) in (3.5) we get for $k = 0, 1, \ldots, K-1$

$$X^i_{k+1} = X^i_k + h^{\frac{1}{2}}_{k+1}\xi^i_{k+1}, \quad i = 1, \ldots, n, \tag{3.10a}$$

$$\begin{aligned} Y_{k+1} = Y_k\Big(1 &+ (b^i(X_k) + \gamma^{0i}(X_k)h_{k+1})h^{\frac{1}{2}}_{k+1}\xi^i_{k+1} \\ &+ c(X_k)h_{k+1} + \gamma^{ij}(X_k)\frac{h_{k+1}}{2}\left(\xi^i_{k+1}\xi^j_{k+1} + \Xi^{ij}_{k+1}\right)\Big), \end{aligned} \tag{3.10b}$$

and

$$Z_{k+1} = Z_k + Y_k\left(g(X_k)h_{k+1} + \zeta^{i0}(X_k)\frac{h^{\frac{3}{2}}_{k+1}}{2}\xi^i_{k+1} + \zeta^{00}(X)\frac{h^2_{k+1}}{2}\right). \tag{3.10c}$$

We emphasize the fact that the random variables ξ^i and Ξ^{ij} are i.i.d. not only wrt. i and $j > i$ but also independent for every step by adding a subscript $k+1$.

Remark 3.6. The one step approximation (3.10) coincides with the scheme (14.2.7) in the monograph [KP92] where it is called the *simplified order 2.0 weak Taylor scheme*. Also, (3.10) is derived in [Mil95a] yielding there the scheme (8.31). We note though, that there, the proposed model for $I_{(ij),h}$ differs from our model as proposed in (3.7) and (3.8).

3.4 A Monte-Carlo algorithm based on the walk on cubes

In this section we give an algorithm to compute a Monte-Carlo approximation for $u(x)$ solving the BVP (2.4) with $L = \frac{1}{2}\triangle + b^i(x)\partial_i$. System (2.10) is integrated numerically using the one step approximation (3.10) with random step size h_k obtained by Algorithm 3.1. The approximate exit time and the approximation of the exit point follows this algorithm as well. We recall the two boundary layers, tangent and stopping, respectively,

$$\Gamma_T = \left\{x \in D \ : \ \rho(x, \partial D) < \sqrt{3nh}\right\} \quad \text{and} \quad \Gamma_S = \left\{x \in D \ : \ \rho(x, \partial D) < h^2\right\}.$$

N independent trials are performed and then the expectation in the probabilistic representation (2.7) is approximated by the mean over these N samples. The algorithm is displayed in Algorithm 3.2.

Algorithm 3.2 Monte-Carlo algorithm for Dirichlet problems using bounded increments

Choose $h > 0$ small enough and $x \in D \setminus \Gamma_S$
Set $u \leftarrow 0$ and $h_0 \leftarrow h$.
if $x \in \Gamma_T$ **then**
  $h_0 \leftarrow \rho(X, \partial D)^2/(3n)$
end if
for $p = 1, \ldots, N$ **do** {loop over paths}
  $X \leftarrow x$, $Y \leftarrow 1$, $Z \leftarrow 0$, $h \leftarrow h_0$ {initialization for this path}
  for $k = 0, 1, \ldots$ **do**
    Generate ξ and Ξ according to (3.6) and (3.7)
    Update (see (3.10c))

$$Z \leftarrow Z + Y\left(g(X)h + \zeta^{i0}(X)\frac{h^{\frac{3}{2}}}{2}\xi^i + \zeta^{00}(X)\frac{h^2}{2}\right)$$

    Update (see (3.10b))

$$Y \leftarrow Y\left(1 + \left(b^i(X) + \gamma^{0i}(X)h\right)h^{\frac{1}{2}}\xi^i + c(X)h + \gamma^{ij}(X)\frac{h}{2}\left(\xi^i\xi^j + \Xi^{ij}\right)\right)$$

    if $X \in \Gamma_S$ **then**
      STOP
    else if $X \in \Gamma_T$ **then**
      $h \leftarrow \rho(X)^2/(3n)$
    else
      Set h to initial step size
    end if
    Update $X \leftarrow X + h^{\frac{1}{2}}\xi$
  end for
  Set X_ν to a point on ∂D closest to X
  $u \leftarrow u + (\psi(X_\nu) + Z)$ {general solution}
end for
$u_h^N \leftarrow u/N$ {compute expectation}

Chapter 4

Discussion of Algorithm 3.1 and of the one step approximation (3.10)

In this section we prove two theorems concerning the expected number of steps of the Markov chain $(X_k)_k$ resulting from Algorithm 3.1 and the accuracy of the one step approximation (3.10).

4.1 The expected number of steps

Theorem 4.1. *Let $D \subset \mathbb{R}^n$ be bounded. Consider the Markov chain $(X_k)_{k=0,\dots,K}$ obtained by Algorithm 3.1 with initial step size $h > 0$. Then, the expected number of steps until $X_K \in \Gamma_S$ is bounded, in particular $\mathbb{E}[K] = \mathcal{O}(h^{-1})$.*

Proof. To simplify notation, we write $X = X_k$ and $\Delta X = X_{k+1} - X_k$ throughout this proof. We further denote by $\nu_x = K(x)$ the number of steps for a chain starting at x, $X_0 = x$.

We follow the outline of a similar proof in [Mil97] and define

$$A\nu(X) = \mathbb{E}[\nu(X + \Delta X)] - \nu(X).$$

Note that $\nu(x) \neq \nu_x$ in general but our goal is to find $\nu(x)$ such that $\mathbb{E}[\nu_x] \leq \nu(x)$.

Consider the boundary value problem with $f \in C(D \setminus \Gamma_S)$,

$$A\nu(X) = -f(X), \quad X \in D \setminus \Gamma_S \quad \text{and} \quad \nu(X) = 0, \quad X \in \Gamma_S, \tag{4.1}$$

connected with the chain $(X_k)_k$. Let $\nu_x = K$ if $X_0 = x$. For any Markov chain $(X_k)_{k=0,1,\dots}$ with $X_0 = x$ we have, by the definition of A and (4.1)

$$\begin{aligned}
\mathbb{E}\left[\nu(X_1) - \nu(x); x\right] &= -f(x) \\
\mathbb{E}\left[\nu(X_2) - \nu(X_1); X_1\right] &= -f(X_1) \\
\vdots \qquad & \qquad \vdots \\
\mathbb{E}\left[\nu(X_{\nu_x-1}) - \nu(X_{\nu_x-2}); X_{\nu_x-2}\right] &= -f(X_{\nu_x-2}) \\
-\mathbb{E}\left[\nu(X_{\nu_x-1}); X_{\nu_x-1}\right] &= -f(X_{\nu_x-1}).
\end{aligned}$$

In the last equation we have used the fact that $X_{\nu_x} \in \Gamma_S$ by the definition of ν_x and, consequently, using the boundary condition of (4.1), $\nu(X_{\nu_x}) = 0$. Working backwards, we find the unique solution to (4.1) (see [Mil97, Wen79]),

$$\nu(x) = \mathbb{E}\left[f(x) + f(X_1) + \cdots + f(X_{\nu_x-1})\right] = \mathbb{E}\sum_{k=0}^{\nu_x-1} f(X_k).$$

For $f(x) \equiv 1$ we have therefore $\nu(x) = \mathbb{E}[\nu_x]$ and, in particular, $f \geq 1 \implies \nu(x) \geq \mathbb{E}[\nu_x]$. Applying Itô's formula we find

$$\nu(X + \Delta X) - \nu(X) = \int_0^h \left\langle \frac{\partial}{\partial X}\nu(X(t)),\ \mathrm{d}X(t) \right\rangle + \frac{1}{2}\int_0^h \Delta\nu(X(t))\,\mathrm{d}t,$$

and therefore

$$A\nu(X) = \frac{1}{2}\int_0^t \mathbb{E}[\Delta\nu(X(t))]\,\mathrm{d}t.$$

Consequently, if $\Delta\nu(X(t)) < 0$ all along the path $X \to X + \Delta X$, we have $A\nu(X) < 0$.

Define for $L \in \mathbb{R}$ the following function:

$$\nu_0(X) = \begin{cases} L^2 - |X|^2, & \text{for} \quad X \in D \setminus \Gamma_S \\ 0, & \text{for} \quad X \in \Gamma_S. \end{cases}$$

If D is bounded, we may choose L large enough such that $\nu_0(X) \geq 0$. Additionally, let $\mathrm{d}S$ denote the outward oriented area increment of the inner surface of the exit layer Γ_S, that is $X \in \Gamma_S \setminus \partial D$, and let

$$\nu_h(X) = -\mathcal{K}\nu_0 = -\int_{\partial\Gamma_S\setminus\partial D} k(X, y)\nu_0(y)\,\mathrm{d}S_y,$$

where $k(\cdot,\cdot)$ is the Green function for the Laplace operator. For a sphere, for example, k is the Poisson kernel, see [Doo84, chap. II, sec. 1].

We need the following two Lemmata.

Lemma 4.2. *Let $X \in D \setminus \Gamma_S$. Then $\nu_0(X) + \nu_h(X) \geq 0$.*

Proof. With L large enough, $\nu_0 \geq 0$ and ν_0 is sub-harmonic. By construction, $\nu_h(X)$ is harmonic in $D \setminus \Gamma_S$ and $\nu_h(X) = -\nu_0(X)$ for $X \in \partial\Gamma_S \setminus \partial D$. Thus, from [Doo84, chap. II, sec. 7], $\nu_0 + \nu_h \geq 0$. □

Let

$$\nu_1(X) = \frac{1}{nh}\left(\nu_0(X) + \nu_h(X)\right).$$

Lemma 4.3. *The function ν_1 defined above satisfies $\mathbb{E}[\nu_1(X + \Delta X)] - \nu_1(X) = -1$.*

Proof. We have $\Delta\nu_0(X) = -2n$ and we recall that $\Delta\nu_h(X) = 0$, so

$$\Delta\nu_1(X) = -\frac{2n}{nh} = -\frac{2}{h}.$$

Noting that $\mathbb{E}[\Delta X^{i_1} \cdots \Delta X^{i_m}] = 0$ for $m \in \mathbb{N}$ odd and that $\Delta^{k+1}\nu_1(X) = 0$ for $k = 1, 2, \ldots$, we find, using Taylor expansion of the mathematical expectation (see (4.14), which is [Mil95a, Lemma 13.1]),

$$\mathbb{E}[\nu_1(X + \Delta X)] - \nu_1(X) = \frac{h}{2}\,\Delta\,\nu_1(X) = \frac{h}{2}\left(-\frac{2}{h}\right) = -1.$$

□

Following [Mil97], we next need the contribution appropriate for $X \in \Gamma_T$. When $X \in \Gamma_T \setminus \Gamma_S$, the time step h is computed from the distance to the boundary. Let $\rho = \rho(X, \partial D)$ and denote by $\delta_S = h^2$ the width of Γ_S. Define the function $\nu_2(X)$ as

$$\nu_2(X) = \begin{cases} c_2\left(\log\left(\frac{\sqrt{3h}}{\delta_S}\right) + 1\right), & X \in D \setminus \Gamma_T \\ c_2\left(\log\left(\frac{\rho}{\delta_S}\right) + 1\right), & X \in \Gamma_T \setminus \Gamma_S \\ 0, & X \in \Gamma_S \end{cases}$$

The constant c_2 depends only on n and h and is large enough such that $A\nu_2(X) \leq -1$ (see [Mil97]). ν_2 has the following properties:

Lemma 4.4. *For h small enough it holds that*

$$(i) \qquad \nu_2(X) \geq 0, \quad \forall X \quad \textit{with} \quad \rho(X, \partial D) \geq \delta_S,$$

$$(ii) \qquad \left\langle \Delta X, \frac{\partial}{\partial X} \nu_2(X) \right\rangle \leq 0,$$

$$(iii) \qquad A\nu_2 < 0.$$

Proof. (i): For h small enough, non-negativity is obvious.
(ii): When ΔX moves X away from ∂D, we have $\left\langle \Delta X, \frac{\partial}{\partial X} \nu_2(X) \right\rangle \leq 0$, because $\log(\rho/\delta_S)$ increases for increasing ρ.
(iii): For the walk on cubes we inscribe the polyhedra resulting from the possible points attained by $X + \Delta X$ in spheres with center at X. Figure 4.1 illustrates the situation for $n = 3$. In the general n-case, the possible points form a set of nested n-polyhedra whose

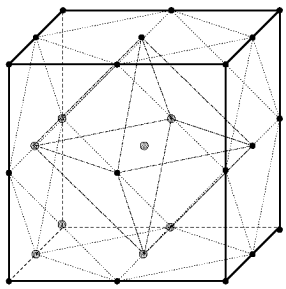

Figure 4.1: Nested n-polyhedra resulting from the possible points attained by $X + \Delta X$ for the walk on cubes (see Algorithm 3.1).

radii (distance from the center to a corner) are $r_k = \sqrt{3hk}$, $k = 0, \ldots, n$. The number of points N_k at the vertices of these polyhedra at radius r_k is given by

$$N_k = \binom{n}{k} 2^{n-k}.$$

We assign a weight P_k to each of these points on the surface of an n-ball of radius r_k by the

probability that these vertices have radius r_k, that is

$$P_k = \mathbb{P}[r_k = \sqrt{3kh}] = 3^{-n}\binom{n}{k}2^{n-k},$$

where $h = \rho^2/(3n)$. Note that

$$\sum_{k=0}^{n} P_k = \left(\frac{1}{6} + \frac{1}{6} + \frac{2}{3}\right)^n = 1.$$

Now let $M \geq 0$ be the number of these radii satisfying $r_k < \rho - \delta_S$, $k = 0, \ldots, M$. The $k = 0$ contribution is point, therefore contributes nothing to $A\nu_2(X)$. Thus, averaging over the weighted points as though they were distributed uniformly on the surfaces S_k of the balls of radii r_k gives

$$\begin{aligned} A\nu_2(X) &= \mathbb{E}[\nu_2(X + \Delta X)] - \nu_2(X) \\ &= \sum_{k=1}^{n} \frac{P_k}{A_k} \int_{S_k} \mathrm{d}S_k \nu_2(X + \Delta X^k) - P_k \nu_2(X). \end{aligned} \tag{4.2}$$

Here,

$$A_k = \frac{2\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}\right)} r_k^{n-1}, \quad \text{with} \quad \Gamma(k) = \int_0^\infty t^{k-1} e^{-t}\, \mathrm{d}t,$$

is the surface area of the n-ball of radius r_k (see [AS64, Chapter 6] for the properties of $\Gamma(k)$). We now split the sum in (4.2) into a sum over $k = 1, \ldots, M$ and another over $k = M + 1, \ldots, n$. For the first one, there is no overlap of the corresponding ball with Γ_S, so since $\Delta\nu_2(X) < 0$, the $k = 1, \ldots, M$ contributions to $A\nu_2(X)$ are negative. By Milstein's Theorem [Mil97, Lemmata 4.1,4.2 and Thm 4.1] the remaining contributions for $M < k \leq n$ are also negative and bounded away from zero. Indeed, by the choice of the step size h, at least one $r_k = \rho$ overlaps Γ_S. Therefore, the scaling c_2 is bounded and $A\nu_2 < 0$. □

We collect the results on $\nu : D \to \mathbb{R}$,

$$\nu(X) = \nu_1(X) + \nu_2(X),$$

to get the following intermediate

Proposition 4.5. *The function $v(X)$ defined above satisfies*

$$\begin{aligned} v(X) &\geq 0, \quad X \in D \setminus \Gamma_S \\ v(X) &= 0, \quad X \in \Gamma_S \\ Av(X) &\leq 1, \quad X \in D \setminus \Gamma_S \end{aligned}$$

Proof. This follows directly by Lemmata 4.2, 4.3 and 4.4. □

Since $c_2 > 0$ is bounded and therefore v_2 is bounded on a bounded domain, we have, combining everything together, that

$$\begin{aligned} \mathbb{E}[v_x] &\leq v(X) \\ &\leq \sup_{X \in D \setminus \Gamma_S} v_1(X) + \sup_{X \in D \setminus \Gamma_S} v_2(X) \\ &\leq \frac{1}{nh} \sup_{X \in D \setminus \Gamma_S} \{v_0(X) + v_h(X)\} + v_2\left(\sqrt{3h}\right) \\ &= \mathcal{O}\left(h^{-1}\right). \end{aligned}$$

Theorem 4.1 is proved. □

4.2 The weak order of accuracy of the one step approximation (3.10)

4.2.1 Preliminary remarks

The goal of this section is to prove a sequence of estimates that concern the weak order of accuracy of the one step approximation (3.10) arising in the walk on cubes, see Algorithms 3.1 and 3.2. In particular, we show that the one step error of the approximation is of (weak) order three. Instead of proving these estimates for (3.10) where the models $\hat{I}_{\alpha,h}$ from (3.8) appear, we assume that the one step approximation is given by (3.4). Using Lemma 3.2 and the fact that $h_k \leq h$ in the walk on cubes (see Remark 3.4), the assertion carries over to the approximation (3.10) with the bounded approximations for the $I_{\alpha,h}$ from (3.8).

Of course, one would like to apply Theorem 2.7 to the system (2.10). To verify the assumptions imposed on the coefficients, one has to impose certain conditions on the coefficients of system (2.10). It would be desirable to formulate these conditions based *only* on the coefficients of the PDE (2.4a). Namely, we wish to impose conditions on $b^i(\cdot)$, $c(\cdot)$ and $g(\cdot)$ alone and *not* on the products $yb^i(x)$, $yc(x)$ and $yg(x)$ which appear in (2.10) ($i = 1, \ldots, n$).

For a simple formulation of our assumptions we need the following

Definition 4.6. We say that a function $f : E \to \mathbb{R}$ *belongs to the class* A_E (and write $f \in A_E$) if f is such that

$$\text{(continuity)} \quad f \in C^6(E), \tag{4.3a}$$

$$\text{(Lipschitz condition)} \quad \exists L < \infty \quad s.t. \quad |f(x) - f(y)| \leq L|x - y| \quad \forall x, y \in E \tag{4.3b}$$

$$\text{(boundedness)} \quad \exists K < \infty \quad s.t. \quad |\partial^\alpha f(x)| \leq K, \quad \forall x, y \in E, \quad 2 \leq |\alpha| \leq 6. \tag{4.3c}$$

Note that in aforementioned definition E is not assumed bounded.
Our assumptions are as follows:

Assumption 4.7. The functions $c(x)$, $g(x)$ and $b^i(x)$, $i = 1, \ldots, n$, belong to the class A_D (with D bounded).

Remark 4.8. The fact that, for example, $c(x) \in C^1(D)$ is Lipschitz for $x \in D$ with D a bounded domain, implies that both $c(x)$ and $\partial_i c(x)$ are bounded $\forall x \in D$. The boundedness of the higher partial derivatives (of order up to including six) has to be imposed separately. The same holds, of course, for $g(x)$ and all the $b^i(x)$.

Remark 4.9. Note that for $x \in D$ with D a bounded domain, the boundedness of f and of the partial derivatives up to (and including) a certain order, is equivalent to imposing a (polynomial or linear) growth condition on f and on the corresponding partial derivatives. Hence, if $f \in A_D$, then $f \in C^6_P(D, \mathbb{R})$. On the other hand, we can define the class A_D as the class of Lipschitz continuous functions which are in $C^6_P(D, \mathbb{R})$.

As noted above, a function $\tilde{f} : D \times \mathbb{R}$, $\tilde{f}(\tilde{x}) = \tilde{f}(x, x^{n+1}) = x^{n+1} f(x)$ (where $x = (x^1, \ldots, x^n)$, $\tilde{x} = (x^1, \ldots, x^{n+1})$), cannot be Lipschitz for all $x^{n+1} \in \mathbb{R}$ in general, even if f is Lipschitz for x in a bounded D. In fact, if $\tilde{f}(\tilde{x})$ is Lipschitz on $D \times \mathbb{R}$, then $f(x)$ necessarily has to be constant on entire D. To see this, assume $\exists \tilde{L} < \infty$ such that $|\tilde{f}(\tilde{x}) - \tilde{f}(\tilde{y})| \leq \tilde{L}|\tilde{x} - \tilde{y}|$ for all $\tilde{x}, \tilde{y} \in D \times \mathbb{R}$ (where $\tilde{y}$ is defined analogously to $\tilde{x}$). Then,

$$|x^{n+1} f(x) - y^{n+1} f(y)| \leq \tilde{L}\sqrt{|x - y|^2 + (x^{n+1} - y^{n+1})^2}, \quad \forall x, y \in D \text{ and } \forall x^{n+1}, y^{n+1} \in \mathbb{R}. \tag{4.4}$$

In particular, (4.4) holds for all $x^{n+1} = y^{n+1} = z \in \mathbb{R}$, namely

$$|z||f(x) - f(y)| \overset{(4.4)}{\leq} \tilde{L}|x - y|, \quad \forall x, y \in D \quad \text{and} \quad \forall z \in \mathbb{R}. \tag{4.5}$$

Now choose an $\varepsilon > 0$ arbitrary and a $z \in \mathbb{R}$ such that $\tilde{L} < |z|\varepsilon$. Then, by (4.5),

$$|f(x) - f(y)| < \varepsilon|x - y|, \quad \forall x, y \in D. \tag{4.6}$$

(4.6) holds $\forall \varepsilon > 0$, as one can find for any such ε a $z \in \mathbb{R}$ such that $\tilde{L} < |z|\varepsilon$. Therefore, letting $\varepsilon \to 0$ in (4.6), we see that the assumption (4.4) on $\tilde{f}$ implies that $f \equiv c$, c a constant, on entire D.

However, we can show that if $f \in A_D$, then $\tilde{f}$ is *locally* Lipschitz, in $C^6(D \times \mathbb{R}, \mathbb{R})$ and satisfies a polynomial growth condition.

Lemma 4.10. *Assume $D \subset \mathbb{R}^n$ bounded and assume that $f \in C^6_P(D, \mathbb{R})$ satisfies the Lipschitz condition* (4.3b) *for $x \in D$ (i.e. we have $f \in A_D$, see Definition 4.6). Then, for any $M < \infty$ and $|y| \leq M$*

(i) $yf \in C^6_P(\tilde{D}, \mathbb{R})$ with $\tilde{D} = D \times (-M, M) \subset \mathbb{R}^{n+1}$.

(ii) yf satisfies the Lipschitz condition (4.3b) *on $\tilde{D} \subset \mathbb{R}^{n+1}$.*

In other words, $yf \in A_{\tilde{D}}$.

Proof. We only show (ii), as (i) is obvious (as noted in Remark 4.9, polynomial growth on a bounded domain is equivalent to the boundedness of the corresponding functions or derivatives).

We write $x = (x^1, \ldots, x^n)$, $\tilde{x} = (x^1, \ldots, x^{n+1})$ and define $\tilde{f}(\tilde{x}) = x^{n+1} f(x)$ as above. Let K denote the Lipschitz constant of f in (4.3b). We wish to show a Lipschitz condition for $\tilde{f}(\tilde{x})$ with $|x^{n+1}| \leq M$. With $\tilde{y} = (y^1, \ldots, y^{n+1}) \in D \times (-M, M)$ we have

$$\begin{aligned} |\tilde{f}(\tilde{y}) - \tilde{f}(\tilde{x})| &= |y^{n+1} f(y) - x^{n+1} f(x)| \\ &= |y^{n+1} f(y) - y^{n+1} f(x) + y^{n+1} f(x) - x^{n+1} f(x)| \\ &\leq M|f(y) - f(x)| + \sup_{x \in D} \{|f(x)|\} \, |y^{n+1} - x^{n+1}|. \end{aligned}$$

Now f is Lipschitz on the bounded domain D, then $\exists M_f < \infty$ with $\sup_{x \in D} |f(x)| \leq M_f$. Hence,

$$\begin{aligned} |\tilde{f}(\tilde{y}) - \tilde{f}(\tilde{x})| &\leq MK|y - x| + M_f|y^{n+1} - x^{n+1}| \\ &\leq \max\{MK, M_f\}(|y - x| + |y^{n+1} - x^{n+1}|). \end{aligned}$$

Next note that for $a, b \in \mathbb{R}$ it holds that $(a + b)^2 \leq 2(a^2 + b^2)$. Indeed,

$$\begin{aligned} & 0 \leq (a - b)^2 = a^2 - 2ab + b^2 \\ \implies \quad & 2ab \leq a^2 + b^2 \quad \implies \quad (a + b)^2 = a^2 + 2ab + b^2 \leq 2(a^2 + b^2). \end{aligned}$$

We get

$$|y - x| + |y^{n+1} - x^{n+1}| \leq \sqrt{2\left(|y - x|^2 + |y^{n+1} - x^{n+1}|^2\right)} = \sqrt{2}\,|\tilde{y} - \tilde{x}|$$

and finally

$$|\tilde{f}(\tilde{y}) - \tilde{f}(\tilde{x})| \leq \tilde{K}|\tilde{y} - \tilde{x}|$$

with $\tilde{K} \leq \sqrt{2}\max\{KM, M_f\} < \infty$ independent of x, y, x^{n+1} and y^{n+1}, which proves the statement (ii). □

4.2.2 Without first order terms

In the case that $b^i \equiv 0$, Theorem 2.7 can still be applied. For Poisson's equation (i.e. also $c \equiv 0$) the fact that $g(x)$ satisfies Assumption 4.7, immediately implies the conclusion (2.18) for $l = 3$. In the case that $0 \not\equiv c \leq 0$, we need the fact that the process variable Y, the solution to (2.10b), is bounded for all $t > s$ if $Y(s) < \infty$. Indeed, a (formal) solution to (2.10b) with $b^i \equiv 0$ for $i = 1, \ldots, n$ is

$$Y(t) = Y(s)\exp\left(\int_s^t c(X(s_1))\,\mathrm{d}s_1\right). \tag{4.7}$$

Therefore, $c(\cdot) \leq 0$ implies that $0 \leq Y(t) \leq Y(s)$. By the imposed initial condition, namely the deterministic $Y(0) = 1$, we see that $|Y(t)| < \infty$ for all $t \geq 0$. Hence, $c, g \in A_D$ imply again (2.18), by Lemma 4.10.

We summarize this in

Theorem 4.11. *In the case without first order terms, suppose $c(x)$, $g(x)$ satisfy Assumption 4.7. Then, the weak order of accuracy of the one step approximation* (3.5) *(and therefore also the accuracy of the one step approximation* (3.10)*) is three.*

More precisely, let $X_k = X(t_k)$, $Y_k = Y(t_k)$ and $Z_k = Z(t_k)$ where X, Y and Z are the solutions to system (2.10)*. Then, $\exists C_f$ such that $\forall f \in C_P^6(\mathbb{R}^{n+2}, \mathbb{R})$, $h > 0$*

$$\left|\mathbb{E}\left[f(X(t_k+h), Y(t_k+h), Z(t_k+h)) - f(X_{k+1}, Y_{k+1}, Z_{k+1})\right]\right| \leq C_f h^3 \tag{4.8}$$

where X_{k+1}, Y_{k+1} and Z_{k+1} are obtained by (3.5) *(or by* (3.10) *respectively).*

Proof. For the approximation (3.5) this follows immediately by (4.7), Lemma 4.10 and Theorem 2.7. For the approximation (3.10), use additionally Lemma 3.2 and Remark 3.4. □

4.2.3 General case: with first order terms

We now turn our attention to equation (2.10b).

4.2.3-a) Introductory remarks

The equation of interest is recalled (see (2.10b)),

$$\mathrm{d}Y(s) = c(X(s))Y(s)\,\mathrm{d}t + Y(s)\sum_{i=1}^{n} b^i(X(s))\,\mathrm{d}W^i(s), \quad s \geq t. \tag{4.9}$$

where for $s \geq t$ $\mathrm{d}X(s) = \mathrm{d}W(s)$. Solutions to the initial conditions $X(t) = x$, $Y(t) = y$ are denoted by $X_{t,x}(s)$ and $Y_{t,x,y}(s)$ respectively. A formal solution to (4.9) for $s = t + h$ is then

$$Y_{t,x,y}(t+h) = y \exp\left(\int_t^{t+h} c(X_{t,x}(s))\,\mathrm{d}s \right.$$
$$\left. + \int_t^{t+h} b^i(X_{t,x}(s))\,\mathrm{d}W^i(s) - \frac{1}{2}\int_t^{t+h} b^i(X_{t,x}(s))b^i(X_{t,x}(s))\,\mathrm{d}s \right). \tag{4.10}$$

Due to the possible unboundedness of the second integral in the exponent, $Y_{t,x,y}(s)$ itself might become arbitrarily large for any $0 < y < \infty$, $0 < h < \infty$ and the argument of the previous section does no longer hold true. Therefore, different proofs are needed.

Our proofs follow similar proofs introduced by Milstein in the papers [Mil74, Mil78], see also the monograph [Mil95a, Chapters 13 and 14]. We prove the estimates for the one step approximation (3.5). There, we assume that $I_{(i),h}$ is modelled exactly (using an i.i.d. normal random variable with mean zero and variance h in each step). We assume that X is modelled exactly, whereas we need an approximation $\overline{Y}$ for Y.

Now we introduce (with (4.10) in mind)

$$\varphi(t,x) = \exp\left(\int_t^{t+h} c(X_{t,x}(s))\,\mathrm{d}s + \int_t^{t+h} b^i(X_{t,x}(s))\,\mathrm{d}W^i(s) \right.$$
$$\left. - \frac{1}{2}\int_t^{t+h} b^i(X_{t,x}(s))b^i(X_{t,x}(s))\,\mathrm{d}s \right).$$

Note that $Y_{t,x,y}(t+h) = y\varphi(t,x)$ in (4.10). For a specific one step approximation $\overline{Y}_{t,x,y}(t+h)$ to $Y_{t,x,y}(t+h)$ with (by definition) $\overline{Y}_{t,x,y}(t) = y$ we will prove in view of (4.10) for fixed $h > 0$ (the step size)

$$\mathbb{E}\left[Y_{t,x,y}(t+h)\varphi(t+h, X_{t,x}(t+h)) - \overline{Y}_{t,x,y}(t+h)\varphi(t+h, X_{t,x}(t+h)) \right] = \mathcal{O}(h^3).$$

4.2.3-b) Some known results

We recall some results and introduce some abbreviation for our notation.

Weak Taylor expansion of order $\beta = 2.0$ for general $\varphi(t, \mathcal{X})$ (with local error of $\mathcal{O}(h^3)$): For $i = 1, \dots, d$ let $\mathcal{X}^i$ denote the solution to $d\mathcal{X}^i = B^i(\mathcal{X})\,\mathrm{d}t + S^{ir}(\mathcal{X})\,\mathrm{d}W^r$ (see (2.11)). Similar to the expansion introduced in Section 2.2 for $\mathcal{X}$, we have an expansion for $\varphi(s, \mathcal{X}(s))$ over the hierarchical set Γ_2 for $s \geq t$ (see (2.15) and (2.19)),

$$\varphi(s, \mathcal{X}_{t,x}(s)) = \varphi(t,x) + \Lambda_i\varphi(t,x))I_{(i)} + L_t\varphi(t,x)(s-t) + \Lambda_i\Lambda_j\varphi(t,x)I_{(i,j)} + \Lambda_i L_t\varphi(t,x)I_{(i,0)} + L_t\Lambda_i\varphi(t,x)I_{(0,i)} + L_t^2\varphi(t,x)\frac{(s-t)^2}{2} + R \tag{4.11}$$

where (compare with (2.13))

$$L_t = \partial_t + B^m\partial_m + \frac{1}{2}S^{mi}S^{ni}\partial_m\partial_n, \quad \Lambda_i = S^{mi}\partial_m. \tag{4.12}$$

We denote the second order operator in (4.12) by L_t, even if the variable of the first argument of the function it is applied on is denoted by s, s_1 etc. instead of t.

We finally recall (see (2.12))

$$I_{(i)} = I_{(i),s-t} = \int_t^s \mathrm{d}W^i(s_1), \quad I_{(i,0)} = I_{(i,0),s-t} = \int_t^s\int_t^{s_1} \mathrm{d}W^i(s_2)\,\mathrm{d}s_1$$

$$I_{(0,i)} = I_{(0,i),s-t} = \int_t^s\int_t^{s_1} \mathrm{d}s_2\,\mathrm{d}W^i(s_1), \quad I_{(i,j)} = I_{(i,j),s-t} = \int_t^s\int_t^{s_1} \mathrm{d}W^i(s_2)\,\mathrm{d}W^j(s_1).$$

Remark 4.12. Note that in this entire section we will *not* use the identity $I_{(0,i)} = hI_{(i)} - I_{(i,0)}$ for the truncated expansions of the form (4.11). This is in contrast to (3.4) and (3.5). There, this identity was only used to reduce the number of Itô integrals for which models where needed.

In (4.11), $R = \sum_{\alpha\in\mathcal{B}(\Gamma_2)} I_\alpha[\varphi_\alpha]$ (over the remainder set $\mathcal{B}(\Gamma_2)$), similar to (2.20) with L_t replacing L and a general $\varphi = \varphi(\cdot, \mathcal{X}(\cdot))$ replacing the identity map $\varphi = \mathcal{X}$.

Multiplication table: Assume A and B are two truncated expansions over Γ_2 of the form (4.11) (without corresponding remainder terms R),

$$A = \sum_{\alpha_a\in\Gamma_2} a_{\alpha_a}I_{\alpha_a}, \quad \text{and} \quad B = \sum_{\alpha_b\in\Gamma_2} b_{\alpha_b}I_{\alpha_b}.$$

In order to compute $\mathbb{E}[AB]$, we see that each term inside the expectation has the form $a_{\alpha_a} b_{\alpha_b} I_{\alpha_a} I_{\alpha_b}$. Assume the interval of integration has length h ($s = t + h$). Using the rules (see for example [KP92, pp.222-223])

$$\mathbb{E}[I_\alpha] = 0 \text{ if } l(\alpha) \neq n(\alpha), \quad \mathbb{E}[I_{\alpha_a} I_{\alpha_b}] = 0 \text{ if } l(\alpha_a) + l(\alpha_b) - (n(\alpha_a) + n(\alpha_b)) \text{ is odd},$$

$$\mathbb{E}[I_{(i)} I_{(j)}] = \delta_{ij} h, \quad \mathbb{E}[I_{(i)} I_{(0,j)}] = \mathbb{E}[I_{(i)} I_{(j,0)}] = \delta_{ij} \frac{h^2}{2},$$

$$\mathbb{E}[I_{(i,j)} I_{(k,l)}] = \delta_{ik}\delta_{jl} \frac{h^2}{2}, \quad \mathbb{E}[I_{(i,0)} I_{(j,0)}] = \mathbb{E}[I_{(i,0)} I_{(0,j)}] = \mathbb{E}[I_{(0,i)} I_{(0,j)}] = \delta_{ij} \frac{h^3}{3}$$

we get the values of $\mathbb{E}[I_{\alpha_a} I_{\alpha_b}]$ for $\alpha_a, \alpha_b \in \Gamma_2$. These are summarized in Table 4.1 for an interval of integration of length h.

Table 4.1: $\mathbb{E}[I_{\alpha_a} I_{\alpha_b}]$ for $\alpha_a, \alpha_b \in \Gamma_2$. Each entry of the table corresponds to the expectation of the product of the Itô integrals with multi-indices of the corresponding row and column. In each case the outermost integration is from t to $t + h$.

	v	(k)	(0)	(k,l)	$(0,k)$	$(k,0)$	$(0,0)$
v	1	0	h	0	0	0	$\frac{h^2}{2}$
(i)	0	$\delta_{ik} h$	0	0	$\delta_{ik}\frac{h^2}{2}$	$\delta_{ik}\frac{h^2}{2}$	0
(0)	h	0	h^2	0	0	0	$\frac{h^3}{2}$
(i,j)	0	0	0	$\delta_{ik}\delta_{jl}\frac{h^2}{2}$	0	0	0
$(0,i)$	0	$\delta_{ik}\frac{h^2}{2}$	0	0	$\delta_{ik}\frac{h^3}{3}$	$\delta_{ik}\frac{h^3}{3}$	0
$(i,0)$	0	$\delta_{ik}\frac{h^2}{2}$	0	0	$\delta_{ik}\frac{h^3}{3}$	$\delta_{ik}\frac{h^3}{3}$	0
$(0,0)$	$\frac{h^2}{2}$	0	$\frac{h^3}{2}$	0	0	0	$\frac{h^4}{4}$

For the expectation of the product of A and B we get (using summation convention)

$$\begin{aligned}\mathbb{E}[AB] =& a_v b_v + \left(a_v b_{(0)} + a_{(i)} b_{(i)} + a_{(0)} b_v\right) h \\ &+ \Big(a_v b_{(0,0)} + a_{(i)} b_{(0,i)} + a_{(i)} b_{(i,0)} + 2a_{(0)} b_{(0)} \\ &\quad + a_{(i,j)} b_{(i,j)} + a_{(0,i)} b_{(i)} + a_{(i,0)} b_{(i)} + a_{(0,0)} b_v\Big) \frac{h^2}{2} + \mathcal{O}(h^3)\end{aligned} \tag{4.13}$$

Taylor expansion of mathematical expectation: Let $f = f(\cdot, \mathcal{X}(\cdot))$, then for $s \geq t$ (see for example [Mil95a, Chapter 13.2])

$$\begin{aligned} f(s, \mathcal{X}_{t,x}(s)) = f(t,x) + (L_t f)(t,x)h + (L_t^2 f)(t,x)\frac{h^2}{2} \\ + \int_t^s \frac{(s-s_1)^2}{2} L_t^3 f(s_1, \mathcal{X}(s_1))\, \mathrm{d}s_1 \\ + \int_t^s \Big(\Lambda_i f(s_1, \mathcal{X}(s_1)) + (s-s_1)\Lambda_i L_t f(s_1, \mathcal{X}(s_1)) \\ + \frac{(s-s_1)^2}{2} \Lambda_i L_t^2 f(s_1, \mathcal{X}(s_1)) \Big)\, \mathrm{d}W^i(s_1) \end{aligned}$$

where $h = s - t$. Taking the expectation yields (assuming that f is such that all the needed derivatives of f exist and corresponding expectations exist and are continuous with respect to t, see [Mil95a, Lemma 13.1])

$$\mathbb{E}[f(s, \mathcal{X}_{t,x}(s)] = f(t,x) + (L_t f)(t,x)h + (L_t^2 f)(t,x)\frac{h^2}{2} + \mathcal{O}(h^3) \tag{4.14}$$

where $\mathcal{X}$ solves (2.11) and L_t and Λ_i are defined in (4.12).

4.2.3-c) Proofs

We first consider two special versions of (4.9) and prove corresponding estimates for the accuracy of our approximation. These two simplified versions will then allow us to prove the general case within a reasonable amount of work. To simplify notation, we often write functions without their arguments if no confusion arises.

Potential problem: We consider the PDE

$$\frac{1}{2} \triangle u(x) + c(x)u(x) = 0, \quad c(x) \leq 0.$$

We set $\mathcal{X} = (X^1, \ldots, X^n, Y)^T$ and consider the system of SDEs

$$\mathrm{d}X(s) = \mathrm{d}W(s) \quad \text{and} \quad \mathrm{d}Y(s) = Y(s)c(X(s))\, \mathrm{d}s.$$

Hence

$$B(\mathcal{X}) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ Yc(X) \end{pmatrix} \quad \text{and} \quad S(\mathcal{X}) = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ 0 & \dots & 0 \end{pmatrix}$$

yielding

$$L = \frac{1}{2} \triangle + yc(x)\partial_y \quad \text{and} \quad \Lambda_i = \partial_i$$

with $\Lambda_i y = 0$. The one step approximation for $s = t + h$ is now easily found to be

$$\overline{Y}_{t,x,y}(s) = y\left(1 + ch + \partial_i c \cdot I_{(i,0)} + \left(\frac{1}{2} \triangle c + c^2\right)\frac{h^2}{2}\right), \tag{4.15}$$

where $c = c(x)$.

Lemma 4.13. *Let the potential function c satisfy Assumption 4.7. Then, the following estimate holds:*

$$\mathbb{E}\left[\overline{Y}_{t,x,y}(s)\varphi(s, X_{t,x}(s)) - Y_{t,x,y}(s)\varphi(s, X_{t,x}(s))\right] = \mathcal{O}(h^3). \tag{4.16}$$

Proof. Note that $\varphi = \varphi(\cdot, X(\cdot))$ (independent of Y) and therefore $L_t = \partial_t + \frac{1}{2}\triangle$ (and $\Lambda_i = \partial_i$). Computing the expectation of the product of $\overline{Y}_{t,x,y}$ with the expansion of $\varphi(\cdot, X_{t,x}(\cdot))$ (see (4.11)) yields, using Table 4.1 and by (4.13)

$$\begin{aligned}\mathbb{E}\left[\overline{Y}_{t,x,y}(s)\varphi(s, X_{t,x}(s))\right] = y\Big(&\varphi + L_t\varphi h + L_t^2\varphi\frac{h^2}{2} + c\varphi h \\ &+ cL_t\varphi h^2 + (\partial_i c)\partial_i\varphi\frac{h^2}{2} + \gamma^{00}\varphi\frac{h^2}{2}\Big) + \mathcal{O}(h^3).\end{aligned}$$

Grouping together we find the expansion (inserting $\gamma^{00} = \frac{1}{2}\triangle c + c^2$, see (3.2a) with $b^i \equiv 0$)

$$\begin{aligned}\mathbb{E}\left[\overline{Y}_{t,x,y}(s)\varphi(s, X_{t,x}(s))\right] = y\Big(&\varphi + (L_t\varphi + c\varphi)\,h \\ &+ \left(L_t^2\varphi + 2cL_t\varphi + (\partial_i c)\partial_i\varphi + \left(\frac{1}{2}\triangle c + c^2\right)\varphi\right)\frac{h^2}{2}\Big) + \mathcal{O}(h^3).\end{aligned} \tag{4.17}$$

Computing (we apply the commonly used notations $\dot{\varphi} \equiv \partial_t\varphi$, $\ddot{\varphi} = \partial_t^2\varphi$)

$$L_t\varphi = \dot{\varphi} + \frac{1}{2}\triangle\varphi \quad \text{and} \quad L_t^2\varphi = \left[\partial_t + \frac{1}{2}\triangle\right]\left(\dot{\varphi} + \frac{1}{2}\triangle\varphi\right) = \ddot{\varphi} + \triangle\dot{\varphi} + \frac{1}{4}\triangle^2\varphi$$

and inserting into (4.17) gives

$$\begin{aligned}\mathbb{E}\left[\overline{Y}_{t,x,y}(s)\varphi(s, X_{t,x}(s))\right] = \\ y\Big(\varphi + \Big(\dot\varphi + \frac{1}{2}\triangle\varphi + c\varphi\Big)h \\ + \Big(\ddot\varphi + \triangle\dot\varphi + \frac{1}{4}\triangle^2\varphi + 2c\dot\varphi + c\triangle\varphi + (\partial_i c)\partial_i\varphi + \frac{1}{2}(\triangle c)\varphi + c^2\varphi\Big)\frac{h^2}{2}\Big) \\ + \mathcal{O}(h^3).\end{aligned} \tag{4.18}$$

On the other hand, for $u = u(t, X, Y)$ we have (see (4.14))

$$\mathbb{E}[u(s, X_{t,x}(s), Y_{t,x,y}(s))] = u(t, x, y) + (\hat{L}_t u)(t, x, y)h + (\hat{L}_t^2 u)(t, x, y)\frac{h^2}{2} + \mathcal{O}(h^3) \tag{4.19}$$

where $\hat{L}_t = L_t + yc\partial_y = \partial_t + \frac{1}{2}\triangle + yc\partial_y$. We set $u(t, x, y) = y\varphi(t, x)$ (always: $c = c(x)$, $\varphi = \varphi(t, x)$, $\tilde\varphi = \tilde\varphi(t, x)$) and compute

$$\hat{L}_t[y\varphi(t, x)] = y\left[\partial_t + \frac{1}{2}\triangle + c(x)\right]\varphi(t, x) = y\left(\dot\varphi + \frac{1}{2}\triangle\varphi + c\varphi\right) =: y\tilde\varphi(t, x),$$

and

$$\begin{aligned}\hat{L}_t{}^2[y\varphi(t, x)] = \hat{L}_t\left[\hat{L}_t[y\varphi]\right] = \hat{L}_t[y\tilde\varphi] = y\left(\partial_t\tilde\varphi + \frac{1}{2}\triangle\tilde\varphi + c\tilde\varphi\right) \\ = y\left(\partial_t\left(\dot\varphi + \frac{1}{2}\triangle\varphi + c\varphi\right) + \frac{1}{2}\triangle\left(\dot\varphi + \frac{1}{2}\triangle\varphi + c\varphi\right)\right. \\ \left. + c\left(\dot\varphi + \frac{1}{2}\triangle\varphi + c\varphi\right)\right) \\ = y\left(\ddot\varphi + \frac{1}{2}\triangle\dot\varphi + c\dot\varphi + \frac{1}{2}\triangle\dot\varphi + \frac{1}{4}\triangle^2\varphi\right. \\ \left. + \frac{1}{2}\triangle(c\varphi) + c\dot\varphi + \frac{1}{2}c\triangle\varphi + c^2\varphi\right).\end{aligned}$$

Now

$$\frac{1}{2}\triangle(c\varphi) = \frac{1}{2}c\triangle\varphi + \frac{1}{2}\varphi\triangle c + (\partial_i\varphi)(\partial_i c)$$

therefore

$$\hat{L}_t{}^2[y\varphi(t, x)] = y\left(\ddot\varphi + \triangle\dot\varphi + 2c\dot\varphi + \frac{1}{4}\triangle^2\varphi + c\triangle\varphi + \frac{1}{2}(\triangle c)\varphi + (\partial_i\varphi)(\partial_i c) + c^2\varphi\right)$$

and inserting into (4.19) gives

$$\mathbb{E}\left[Y_{t,x,y}(s)\varphi(s, X_{t,x}(s))\right] = y\Big(\varphi + \Big(\dot{\varphi} + \frac{1}{2}\triangle\varphi + c\varphi\Big)h + \Big(\ddot{\varphi} + \triangle\dot{\varphi} + 2c\dot{\varphi} + \frac{1}{4}\triangle^2\varphi + c\triangle\varphi + \frac{1}{2}(\triangle c)\varphi + (\partial_i\varphi)(\partial_i c) + c^2\varphi\Big)\frac{h^2}{2}\Big) + \mathcal{O}(h^3). \tag{4.20}$$

Comparing (4.18) with (4.20) we get (remember that $s = t + h$ with $h > 0$)

$$\mathbb{E}\left[\overline{Y}_{t,x,y}(s)\varphi(s, X_{t,x}(s)) - Y_{t,x,y}(s)\varphi(s, X_{t,x}(s))\right] = \mathcal{O}(h^3),$$

which was the claim in (4.16). □

With first order terms: We consider the PDE

$$\frac{1}{2}\triangle u(x) + b^i(x)\partial_i u(x) = 0.$$

We set again $\mathcal{X} = (X^1, \ldots, X^n, Y)^T$ and consider the system of SDEs

$$\mathrm{d}X(s) = \mathrm{d}W(s) \quad \text{and} \quad \mathrm{d}Y = Y(s)b^i(X(s))\,\mathrm{d}W^i(s).$$

Hence

$$B(\mathcal{X}) = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \quad \text{and} \quad S(\mathcal{X}) = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ Yb^1(X) & \ldots & Yb^n(X) \end{pmatrix}$$

yielding

$$L = \frac{1}{2}\triangle + yb^i(x)\partial_i\partial_y + \frac{1}{2}y^2|b(x)|^2\partial_y^2, \quad \Lambda_i = \partial_i + yb^i(x)\partial_y$$

with $Ly = 0$. Out of this we get for $s = t + h$ the one step approximation

$$\overline{Y}_{t,x,y}(s) = y\left(\gamma^v + \gamma^i I_{(i)} + \gamma^{ij} I_{(i,j)} + \gamma^{0i} I_{(0,i)}\right) \tag{4.21}$$

where

$$\gamma^0 = 1, \quad \gamma^i = b^i, \quad \gamma^{ij} = \partial_i b^j + b^i b^j, \quad \gamma^{0i} = \frac{1}{2}\triangle b^i + b^j\partial_j b^i$$

and $b^i = b^i(x)$.

Lemma 4.14. *Let for $i = 1, \dots, n$ the functions b^i satisfy Assumption 4.7. Then the following estimate holds:*

$$\mathbb{E}\left[\overline{Y}_{t,x,y}(s)\varphi(s, X_{t,x}(s)) - Y_{t,x,y}(s)\varphi(s, X_{t,x}(s))\right] = \mathcal{O}(h^3). \tag{4.22}$$

Proof. For $\varphi = \varphi(\cdot, X(\cdot))$ we have $L_t = \partial_t + \frac{1}{2}\Delta$ and $\Lambda_i = \partial_i$, as in the previous paragraph. Out of Table 4.1 we similarly find the required version of (4.13):

$$\begin{aligned}\mathbb{E}\left[\overline{Y}_{t,x,y}(s)\varphi(s, X_{t,x}(s))\right] = \gamma^v\varphi^v + \left(\gamma^v\varphi^0 + \gamma^i\varphi^i\right)h \\ + \Bigg(\gamma^v\varphi^{00} + \underbrace{\gamma^i\varphi^{0i} + \gamma^i\varphi^{i0}}_{=2\gamma^i\varphi^{0i}} + \gamma^{ij}\varphi^{ij} + \gamma^{0i}\varphi^i\Bigg)\frac{h^2}{2} \\ \text{(see below)} \\ + \mathcal{O}(h^3),\end{aligned}$$

where by definition

$$\varphi^v = \varphi,\ \varphi^i = \partial_i\varphi,\ \varphi^0 = \frac{1}{2}\Delta\varphi + \dot{\varphi},\quad \varphi^{ij} = \partial_i\partial_j\varphi,\ \varphi^{00} = \frac{1}{4}\Delta^2\varphi + \Delta\dot{\varphi} + \ddot{\varphi}$$

and $$\varphi^{0i} = \frac{1}{2}\Delta\partial_i\varphi + \partial_t\partial_i\varphi = \frac{1}{2}\partial_i\Delta\varphi + \partial_i\dot{\varphi} = \varphi^{i0}$$

with $\varphi = \varphi(t, x)$. Hence,

$$\begin{aligned}&\mathbb{E}\left[\overline{Y}_{t,x,y}(s)\varphi(s, X_{t,x}(s))\right] = \\ &\quad y\Bigg(\varphi + \left(\frac{1}{2}\Delta\varphi + \dot{\varphi} + b^i\partial_i\varphi\right)h \\ &\qquad + \Bigg(\frac{1}{4}\Delta^2\varphi + \Delta\dot{\varphi} + \ddot{\varphi} + b^i\Delta\partial_i\varphi + 2b^i\partial_i\dot{\varphi} \\ &\qquad\quad + (\partial_i b^j)\partial_{ij}\varphi + b^i b^j\partial_{ij}\varphi + \frac{1}{2}(\Delta b^i)\partial_i\varphi + (b^j\partial_j b^i)(\partial_i\varphi)\Bigg)\frac{h^2}{2}\Bigg) + \mathcal{O}(h^3).\end{aligned} \tag{4.23}$$

As before, let $\hat{L}_t = \partial_t + \frac{1}{2}\Delta + yb^i\partial_i\partial_y + \frac{1}{2}y^2|b|^2\partial_y^2$ and compute for $u(t, x, y) = y\varphi(t, x)$

$$\hat{L}_t[y\varphi(t, x)] = y\left(\dot{\varphi} + \frac{1}{2}\Delta\varphi + b^i\partial_i\varphi\right) \qquad \overset{(4.23)}{\Longrightarrow} \qquad \mathcal{O}(h)\text{-terms agree,}$$

and

$$\begin{aligned}\hat{L}_t^2[y\varphi(t,x)] &= y\left[\partial_t + \frac{1}{2}\triangle + b^i\partial_i\right]\left(\dot{\varphi} + \frac{1}{2}\triangle\varphi + b^j\partial_j\varphi\right)\\ &= y\Bigg(\ddot{\varphi} + \frac{1}{2}\triangle\dot{\varphi} + b^i\partial_i\dot{\varphi} + \frac{1}{2}\triangle\dot{\varphi} + \frac{1}{4}\triangle^2\varphi\\ &\qquad + \frac{1}{2}\triangle(b^i\partial_i\varphi) + b^i\partial_i\dot{\varphi} + \frac{1}{2}b^i\partial_i(\triangle\varphi) + b^i\partial_i b^j\partial_j\varphi\Bigg).\end{aligned}$$

Rearranging and using

$$\begin{aligned}\triangle\partial_i\varphi &= \partial_i\triangle\varphi,\\ \triangle(b^i\cdot\partial_i\varphi) &= (\partial_i\varphi)(\triangle b^i) + b^i\triangle\partial_i\varphi + 2(\partial_j b^i)(\partial_{ij}\varphi),\end{aligned}$$

and

$$\partial_i(b^j\cdot\partial_j\varphi) = (\partial_i b^j)(\partial_j\varphi) + b^j\partial_{ij}\varphi$$

we get

$$\begin{aligned}\hat{L}_t^2[y\varphi(t,x)] = y\Bigg(&\frac{1}{4}\triangle^2\varphi + \triangle\dot{\varphi} + \ddot{\varphi} + b^i\triangle\partial_i\varphi + 2b^i\partial_i\dot{\varphi} + (\partial_j b^i)\partial_{ij}\varphi\\ &+ b^i b^j\partial_{ij}\varphi + \frac{1}{2}(\partial_i\varphi)(\triangle b^i) + (b^i\partial_i b^j)(\partial_j\varphi)\Bigg)\end{aligned}$$

which in comparison with (4.23) shows that also the $\mathcal{O}(h^2)$-terms agree. This completes the proof of (4.22). □

General case: We finally consider the combined case, i.e. the PDE

$$\left[\frac{1}{2}\triangle + b^i(x)\partial_i + c(x)\right]u(x) = 0, \quad c(x)\le 0.$$

Setting as usual $\mathcal{X} = (X^1,\dots,X^n, Y)^T$ and considering now the SDEs

$$\mathrm{d}X(s) = \mathrm{d}W(s) \quad\text{and}\quad \mathrm{d}Y(s) = Y(s)c(X(s))\,\mathrm{d}s + Y(s)b^i(X(s))\,\mathrm{d}W^i(s)$$

we get

$$B(\mathcal{X}) = \begin{pmatrix} 0\\ \vdots\\ 0\\ Yc(X)\end{pmatrix} \quad\text{and}\quad S(\mathcal{X}) = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1\\ Yb^1(X) & \dots & Yb^n(X)\end{pmatrix}$$

yielding

$$L = yc(x)\partial_y + \frac{1}{2}\triangle + yb^i(x)\partial_i\partial_y + \frac{1}{2}y^2|b(x)|^2\partial_y^2 \quad \text{and} \quad \Lambda_i = \partial_i + yb^i(x)\partial_y.$$

Out of this we find again the corresponding one step approximation for $s = t + h$

$$\begin{aligned}\overline{Y}_{t,x,y}(s) = y\Big(1 + b^i I_{(i)} + ch + \left(\partial_i b^j + b^i b^j\right) I_{(i,j)} + \left(\frac{1}{2}\triangle b^i + cb^i + b^j\partial_j b^i\right) I_{(0,i)} \\ + \left(\partial_i c + b^i c\right) I_{(i,0)} + \left(\frac{1}{2}\triangle c + c^2 + b^i\partial_i c\right)\frac{h^2}{2}\Big)\end{aligned} \tag{4.24}$$

where $b^i = b^i(x)$ and $c = c(x)$.

Theorem 4.15. *Let the Assumption 4.7 hold. Then, the following estimate holds:*

$$\mathbb{E}\left[\overline{Y}_{t,x,y}(s)\varphi(s, X_{t,x}(s)) - Y_{t,x,y}(s)\varphi(s, X_{t,x}(s))\right] = \mathcal{O}(h^3). \tag{4.25}$$

Proof. We make use of Lemmata 4.13 and 4.14. Let for simplicity of notation $\overline{Y}(s) = \overline{Y}_{t,x,y}(s)$ and similarly define

- for $b = 0$ (see (4.15)):

$$\Delta_c(s) = \overline{Y}_c(s) - y = y\left(ch + \partial_i c I_{(i,0)} + \left(\frac{1}{2}\triangle c + c^2\right)\frac{h^2}{2}\right)$$

- for $c = 0$ (see (4.21)):

$$\Delta_b(s) = \overline{Y}_b(s) - y = y\left(b^i I_{(i)} + \left(\partial_i b^j + b^i b^j\right) I_{(i,j)} + \left(\frac{1}{2}\triangle b^i + b^j\partial_j b^i\right) I_{(0,i)}\right)$$

- general (see (4.24)):

$$\begin{aligned}\Delta_g(s) &= \overline{Y}_g(s) - y \\ &= y\Big(b^i I_{(i)} + ch + \left(\partial_i b^j + b^i b^j\right) I_{(i,j)} + \left(\frac{1}{2}\triangle b^i + cb^i + b^j\partial_j b^i\right) I_{(0,i)} \\ &\quad + \left(\partial_i c + b^i c\right) I_{(i,0)} + \left(\frac{1}{2}\triangle c + c^2 + b^i\partial_i c\right)\frac{h^2}{2}\Big).\end{aligned}$$

Hence

$$\Delta_g(s) = \Delta_c(s) + \Delta_b(s) + y\left(cb^i I_{(0,i)} + b^i c I_{(i,0)} + b^i \partial_i c \frac{h^2}{2}\right).$$

Therefore (see Table 4.1 and (4.13)),

$$\begin{aligned}
&\mathbb{E}\left[\overline{Y}_g(s)\varphi(s, X_{t,x}(s))\right] = \mathbb{E}\left[(y + \Delta_g(s))\varphi(s, X_{t,x}(s))\right] \\
&= \mathbb{E}\left[\left(y + \Delta_c(s) + \Delta_b(s) + y\left(cb^i I_{(0,i)} + b^i c I_{(i,0)} + b^i \partial_i c \frac{h^2}{2}\right)\right)\varphi(s, X_{t,x}(s))\right] \\
&= \mathbb{E}\left[\left(\overline{Y}_c(s) + \overline{Y}_b(s) - y + y\left(cb^i I_{(0,i)} + b^i c I_{(i,0)} + b^i \partial_i c \frac{h^2}{2}\right)\right)\varphi(s, X_{t,x}(s))\right] \\
&= \mathbb{E}\left[\overline{Y}_c(s)\varphi(s, X_{t,x}(s))\right] + \mathbb{E}\left[\overline{Y}_b(s)\varphi(s, X_{t,x}(s))\right] - y\varphi(t,x) \\
&\qquad + \underbrace{y\left(cb^i \partial_i \varphi \frac{h^2}{2} + b^i c \partial_i \varphi \frac{h^2}{2} + b^i (\partial_i c)\varphi \frac{h^2}{2}\right)}_{= \mathcal{O}(h^2)}.
\end{aligned} \tag{4.26}$$

This already shows (using Lemmata 4.13 and 4.14) that the order h-terms agree.

To compute the expansion of $u(t,x,y)$, we again need $\hat{L}_t = \partial_t + L$, where in the general situation $L = yc(x)\partial_y + \frac{1}{2}\triangle + yb^i(x)\partial_i\partial_y + \frac{1}{2}y^2|b(x)|^2\partial_y^2$. Hence

$$\hat{L}_t = \partial_t + yc(x)\partial_y + \frac{1}{2}\triangle + yb^i(x)\partial_i\partial_y + \frac{1}{2}y^2|b(x)|^2\partial_y^2$$

and we can compute

$$\hat{L}_t[y\varphi(t,x)] = y\left[\partial_t + c + \frac{1}{2}\triangle + b^i\partial_i\right]\varphi(t,x)$$

and

$$\hat{L}_t^2[y\varphi(t,x)] = y\left[\partial_t + c + \frac{1}{2}\triangle + b^i\partial_i\right]\left(\dot{\varphi} + c\varphi + \frac{1}{2}\triangle\varphi + b^j\partial_j\varphi\right).$$

The *new* terms in $\hat{L}_t^2[y\varphi(t,x)]$ are (new compared to $\hat{L}_t$, for the cases where either $b^i = 0$ or $c = 0$)

$$y\left(cb^i\partial_i\varphi + b^i\partial_i(c\varphi)\right).$$

Using the product rule $\partial_i(c\varphi) = \varphi\partial_i c + c\partial_i\varphi$, we see that these terms are in fact exactly the terms of order h^2 in (4.26). This completes the proof. □

Chapter 5

Numerical experiments

This chapter is concerned with numerical experiments using Algorithm 3.2 in hyper-spherical domains, i.e.

$$D = S_1^n(0) = \left\{ x \in \mathbb{R}^n \ : \ |x| < 1 \right\}. \tag{5.1}$$

For our experiments, we take as a variant of the BVP (2.4) Poisson's equation,

$$\frac{1}{2} \triangle u + g(x) = 0, \quad x \in D, \quad u(x) = \psi(x), \quad x \in \partial D. \tag{5.2}$$

We first study in detail the case $g(x) \equiv 1$ and $\psi \equiv 0$ (*exit time problem*) and present results of an experimental error analysis. We later show results from simulations with a varying inhomogeneity, $g_k(x)$, consisting of a product of cosines with different frequencies k^i, $i = 1, \ldots, n$. We compare the results from Algorithm 3.2 with other algorithms for the general Dirichlet problem that have appeared in the literature. A numerical experiment concerning a problem with first order terms concludes this chapter.

Before we present the results, we discuss the setup of our simulations used in this chapter.

5.1 Experimental setup

Algorithm 3.2 has two parameters which influence its accuracy: the initial step size h and the number of simulated paths N. The approximate solution $\overline{u} = u_h^N$ and the error $|u - u_h^N|$ are random variables themselves, varying both for different values of h and N. In order to get consistent experimental results, we chose the following strategy to improve the estimate for a specific outcome o. We computed the mean $\mathbb{E}[o]$ for N_S simulations having outcomes $o^{(1)}, \ldots, o^{(N_S)}$ by $\overline{o}_{N_S} = N_S^{-1} \sum_{i=1}^{N_S} o^{(i)}$. To check the consistency, the following procedure

was used (see Algorithm 5.1): an $f \in (0, 1)$ was chosen to compare the approximation $\overline{o}$ to a 95 % confidence interval computed from the standard deviation $S_{N_S}[o] = \sqrt{\text{var}[o]}$ (see (5.3)).

Algorithm 5.1 Getting reliable numerical results

1: Make at least $N_S > 1$ independent simulations
2: **while** $2S_{N_S}[o] \geq f\,\overline{o}_{N_S}$ **do**
3: Make another simulation
4: $N_S = N_S + 1$.
5: **end while**
6: Take $\overline{o}_{N_S}$ as an estimate for $\mathbb{E}[o]$.

In Algorithm 5.1, the mean is considered reliable if its (numerical) 95%-confidence interval is smaller than its value by the factor f. A smaller f implies a more reliable estimate for $\mathbb{E}[o]$. The standard deviation of o, $S_{N_S}[o]$, is the numerical estimate

$$S_{N_S}[o] = \left(\frac{D_{N_S}[o]}{N_S - 1}\right)^{\frac{1}{2}}, \quad D_{N_S}[o] = N_S^{-1} \sum_{i=1}^{N_S} \left(o^{(i)}\right)^2 - \left(N_S^{-1} \sum_{i=1}^{N_S} o^{(i)}\right)^2. \tag{5.3}$$

For the experiments below, we set $x = 0$ and show the approximations for $u(0)$. The maximal initial step size for Algorithm 3.2 was chosen such that $x = 0 \notin \Gamma_T$, i.e. we ensured that Γ_T is a true subset of D ($\Gamma_T \subsetneq D$). Therefore, we chose $h < 1/(3n)$ for $D = S_1^n(0)$.

5.2 Experimental error analysis of Algorithm 3.2 for the exit time problem

We show results for $g \equiv 1$ and $\psi \equiv 0$. The exact solution is $u(x) = (1 - |x|^2)/n$. Furthermore, $u(0) = \mathbb{E}[\tau]$, where τ is the first exit time of an n-dimensional Brownian motion X from $S_1^n(0)$.

We calculated an approximation for $u(0)$ and looked at the absolute error $e_a = |u(0) - u_h^N(0)|$. For the relative error $e_r = e_a/u(0)$ we have $e_r = ne_a$. When Algorithm 3.2 is applied to this problem, two errors arise: the systematic error e_{Sy} caused by replacing the first n equations of system (2.10) by the discrete Algorithm 3.2, and a statistical error, e_{St}, due to a finite sample of N paths which replaces the expectation $\mathbb{E}$ by a mean of the N sample. Both contribute to the absolute error $e_a = e_{Sy} + e_{St}$. This section hopes to illuminate the reader to the behavior of these errors. In all cases, we made at least 30 simulations and used a confidence factor of $f = 0.1$ or $f = 0.05$ (see Algorithm 5.1).

Random numbers where generated with `ran3` [PTVF02]. Results using the random number generator included in the blitz++ library [Vel98] did not differ.

Figure 5.1 shows the absolute error e_a from simulations with $N = 5 \cdot 10^3$ and $N = 10^4$ paths.

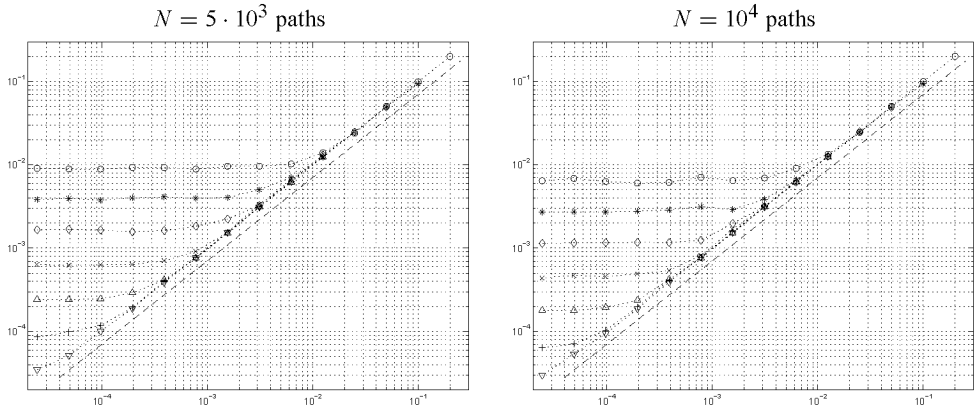

Figure 5.1: Absolute error $|u(0) - u_h^N(0)|$ versus initial step size h obtained when Algorithm 3.2 is applied to the problem (5.2) with $g = 1$, $\psi = 0$ in $D = S_1^n(0)$. Parameters are $f = 0.1$ and $N = 5 \cdot 10^3$ (left) and $N = 10^4$ (right). The following dimensions are shown: $n = 1(\circ), 2(*), 4(\Diamond), 8(\times), 16(\triangle), 32(+), 64(\triangledown)$. A dashed line with slope one $(- - -)$ was added.

We now examine the behavior of the numerical standard deviation $S_N[u_h^N]$ of solution u_h^N (see (5.3) with $o^{(i)} = \tau_h^{(i)}$ and $N_S = N$, where $\tau_h^{(i)}$ is the estimate of the exit time of path i, yielding)

$$S_N[\overline{u}] = \left(\frac{D_N[\overline{u}]}{N-1}\right)^{\frac{1}{2}}, \quad D_N[\overline{u}] = N^{-1}\sum_{i=1}^{N}\left(\overline{u}^{(i)}\right)^2 - \left(N^{-1}\sum_{i=1}^{N}\overline{u}^{(i)}\right)^2. \qquad (5.3')$$

where $\overline{u} = u_h^N = \overline{\tau}_h$ and $\overline{u}^{(i)}$ is the i-th approximation (for sample path i). Table 5.1 shows that the standard deviation S_N depends very weakly on the initial step size h when the dimension n and number of paths N are fixed.

In Table 5.2 we compare the absolute error e_a with S_N for sufficiently small initial step size h. Here, we see that the statistical error is proportional to S_N and is thus proportional to $1/\sqrt{N}$, as expected. It is important to note that the statistical error has the same dependence on N, regardless of the dimension n.

Table 5.1: Comparison of the numerical standard deviation $S_N[u_h^N]$ for different initial step sizes h and for different dimensions n and $N = 10^4$ paths.

	$S_N[u_h^N]$			
n	$h = 0.025$	$h = 0.0125$	$h = 0.00625$	$h = 0.003125$
1	8.153e-3	8.184e-3	8.183e-3	8.178e-3
2	3.538e-3	3.539e-3	3.533e-3	3.531e-3
4	1.444e-3	1.446e-3	1.448e-3	1.446e-3
8	5.593e-4	5.597e-4	5.623e-4	5.598e-4

Table 5.2: Comparison of $S_N[u_h^N]$ and absolute error e_a for different sized dimensions n and two different initial step sizes h (again $N = 10^4$ paths).

	$h = 0.1 \cdot 2^{-11}$			$h = 0.1 \cdot 2^{-12}$		
n	e_a	S_N	e_a/S_N	e_a	S_N	e_a/S_N
1	6.885e-3	8.152e-3	0.845	6.457e-3	8.172e-3	0.790
2	2.726e-3	3.531e-3	0.772	2.711e-3	3.540e-3	0.766
4	1.158e-3	1.443e-3	0.803	1.141e-3	1.443e-3	0.791
8	4.731e-4	5.584e-4	0.847	4.378e-4	5.587e-4	0.784

Table 5.3 shows the numerical standard deviation $S_N[u_h^N]$ for various dimensions n when considering two sample sizes, $N = 10^4$ and $N = 5 \cdot 10^3$, with fixed initial step size h. Alongside the numerical standard deviation we give the numerical slope of a graph of $S_N[u_h^N](n)$ in a double logarithmic plot, given by

$$\text{slope} = \frac{\log \frac{S_i}{S_{i+1}}}{\log \frac{n_i}{n_{i+1}}} .$$

To summarize, we find the following behavior for the absolute error when applying Algorithm 3.2 to the exit time problem:

$$e_a = C_{sy} h + C_{st} \frac{n^{-s}}{\sqrt{N}} \tag{5.4}$$

In equation (5.4), C_{sy} is independent of h and N (Figure 5.1), C_{st} is independent of h

Table 5.3: Comparison of $S_N[u_h^N]$ for different sized dimensions n shown for two sample sizes at fixed initial step sizes $h = 1/320$ and $h = 1/5120$.

	$h = 1/320$				$h = 1/5120$			
	$N = 10000$		$N = 5000$		$N = 10000$		$N = 5000$	
n	S_N	slope	S_N	slope	S_N	slope	S_N	slope
1	3.271e-2	-1.21	4.620e-2	-1.21	3.264e-2	-1.21	4.619e-2	-1.21
2	1.412e-2	-1.29	1.999e-2	-1.29	1.413e-2	-1.29	2.001e-2	-1.29
4	5.782e-3	-1.37	8.168e-3	-1.36	5.772e-3	-1.37	8.164e-3	-1.37
8	2.239e-3	-1.43	3.178e-3	-1.44	2.240e-3	-1.43	3.164e-3	-1.43
16	8.316e-4	-1.45	1.176e-3	-1.46	8.322e-4	-1.45	1.178e-3	-1.46
32	3.038e-4	-1.49	4.290e-4	-1.48	3.037e-4	-1.48	4.286e-4	-1.48
64	1.085e-4		1.538e-4		1.087e-4		1.536e-4	

(Table 5.1) and N (Table 5.2) and further $s = s(n) > 1$ (Table 5.3). This is an encouraging result because it implies that the statistical error retains the $\mathcal{O}(N^{-1/2})$ behavior expected of Monte-Carlo simulation, but the coefficient of this term actually decreases with n.

5.3 Comparison of different algorithms

We now show our results in which we compare the results of Algorithm 3.2 with results of other probabilistic methods for the solution of problem (5.2) that have appeared in the literature. We choose an $n = 3$ dimensional problem and (again) the domain $D = S_1^3(0)$.

5.3.1 Comparing algorithms

We compare our Algorithm 3.2 with other algorithms for elliptic Dirichlet problems, published previously by Milstein et al. [MR93, Mil96, Mil97, Mil98, MT99, Mil01].

For convenience, we summarize the basic features of all algorithms discussed in (5.5–5.8). To simplify notation, we define the increments as

$$\Delta X = X_{k+1} - X_k \quad \text{and} \quad \Delta Z = Z_{k+1} - Z_k.$$

We also neglect subscripts $k + 1$ on h, r and ξ. Using this notation with $X = X_k$, the algorithms we want to compare take the following form for Poisson's equation (5.2):

Simplest random walk (see [Mil01])

$$\begin{aligned} \Delta X &= \sqrt{h}\,\xi \\ \Delta Z &= g(X)h \end{aligned} \tag{5.5}$$

Walk on cubes (see Algorithm 3.1)

$$\begin{aligned} \Delta X &= \sqrt{h}\,\xi \\ \Delta Z &= g(X)h + \partial_i g(X)\frac{h^{\frac{3}{2}}}{2}\xi^i + \frac{1}{2}\,\Delta\, g(X)\frac{h^2}{2} \end{aligned} \tag{5.6}$$

Walk on spheres of global order $h \propto r^2$ (see [Mil97])

$$\begin{aligned} \Delta X &= r\xi \\ \Delta Z &= g(X)\frac{r^2}{n} \end{aligned} \tag{5.7}$$

Walk on spheres of global order $h^2 \propto r^4$ (see [Mil97])

$$\begin{aligned} \Delta X &= r\xi \\ \Delta Z &= g(X)\frac{r^2}{n} + \partial_i g(X)\frac{r^2}{2(n+2)}r\xi^i + \frac{1}{2}\,\Delta\, g(X)\frac{4+n}{n^2(2+n)}r^4 \end{aligned} \tag{5.8}$$

In Table 5.4 the random variables, width ρ_T (δ_S) of boundary layers Γ_T (Γ_S) and the global order of convergence as predicted by theory are summarized for all four algorithms.

Table 5.4: Random variables, boundary layers and global order of convergence for the compared algorithms.

Algorithm	random variables	ρ_T	δ_S	order
(5.5)	$\xi^i = \pm 1$ with probability $\frac{1}{2}$	$\sqrt{nh}$	h	h
3.2	ξ^i as in (3.6)	$\sqrt{3nh}$	h^2	h^2
(5.7)	ξ uniform on ∂S_1^n	r	r^2	r^2
(5.8)	ξ uniform on ∂S_1^n	r	r^4	r^4

For the Algorithm 3.1 and for the method (5.5) the `ran3` random number generator [PTVF02] was used to generate the ξ^i, whereas the random variables uniform on ∂S_1^n required for methods (5.7,5.8) were generated using normal random variables from the blitz++ library [Vel98] and scaling the resulting vectors to unit length.

In order to compare the convergence behavior, we set $h = r^2/n$.

5.3.2 Results

5.3.2-a) Exit time problem

In our numerical experiments in Section 5.2 we found a truncation error of order h for Algorithm 3.2, whereas theoretically a convergence behavior of $\mathcal{O}(h^2)$ should be achievable.

Figure 5.2 shows that the algorithm (5.8) also does not achieve the predicted rate of convergence (order $h^2 = r^4$).

Instead, its convergence rate is of the same order as that of walk on cubes, Algorithm 3.2. The simplest random walk (5.5) has the same order of convergence (it achieves the predicted order h) whereas the simpler walk on spheres (5.7) converges immediately. It should be noted, that for this simple problem, fewer steps are better (the stopping layer Γ_S is wider for the low order algorithms, see Table 5.4).

5.3.2-b) Varying source term $g(x)$

In this section we consider problem (5.2) with

$$g(x) = \prod_{i=1}^{n} \cos(2\pi k^i x^i), \quad k^i = 1, 2, \ldots, \quad k = (k^i)_{i=1,\ldots,n}. \tag{5.9}$$

A solution is easily found to be $u(x) = g(x)/(2\pi^2|k|^2)$ which we also assigned as boundary condition $\psi(x)$ in order to get uniqueness on the surface of the hyper-sphere.

From Figure 5.3 we see that for a harder problem (right plot) with higher frequencies k^i, higher order algorithms perform better than low order algorithms. Furthermore, Algorithm 3.2 shows slightly better convergence behavior than method (5.8).

5.4 A problem including first order terms

As an example with first order terms, we consider

$$\frac{1}{2} \Delta u + b^i(x)\partial_i u + g(x) = 0, \quad x \in D, \quad u(x) = \psi(x), \quad x \in \partial D, \tag{5.10}$$

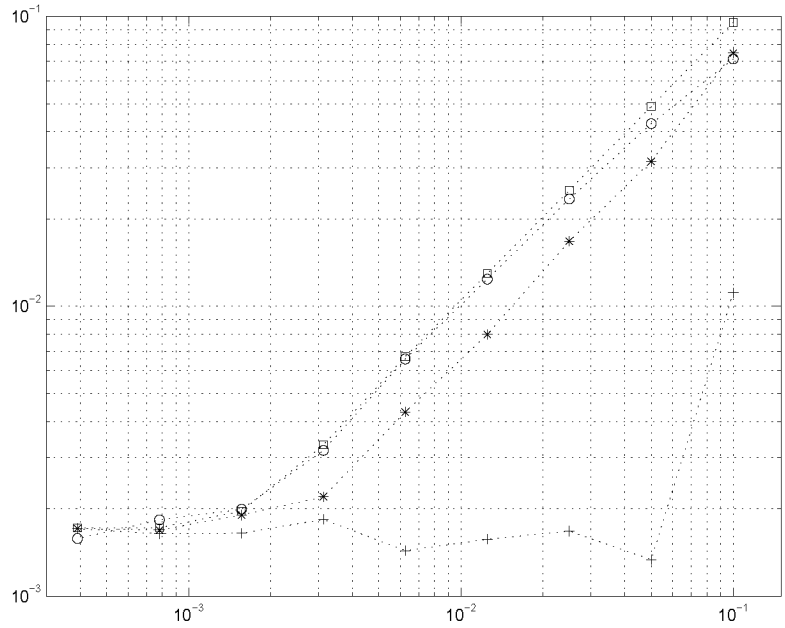

Figure 5.2: Absolute error $|u(0) - u_h^N(0)|$ versus initial step size $h = \frac{r^2}{n}$ obtained when Algorithm 3.2 and methods (5.5,5.7,5.8) are applied to the problem (5.2) with $g = 1$ and $\psi = 0$. The parameters used are $f = 0.05$, $N = 10^4$ and the dimension is $n = 3$. $\square$ is Algorithm 3.1, $\circ$ is (5.8), $*$ is (5.5) and $+$ is (5.7).

with $g(x)$ as in (5.9). For this example, the same boundary condition as given in the previous section, $\psi(x) = g(x)/(2\pi^2|k|^2)$, was applied. We define for even $n = 2, 4, \ldots$

$$b^i(x) = \begin{cases} +\partial_{i+1}u(x), & i = 1, 3, \ldots, n-1, \\ -\partial_{i-1}u(x), & i = 2, 4, \ldots, n. \end{cases} \tag{5.11}$$

This implies $b^i(x)\partial_i u(x) = 0$ and therefore as before $u(x) = g(x)/(2\pi^2|k|^2)$. Nevertheless, the *numerical* solution is quite different from that obtained by solving (5.2). Namely, in the

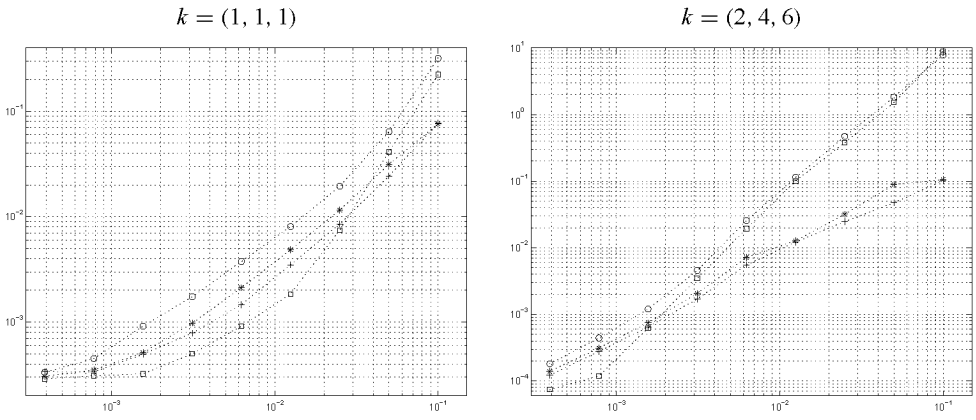

Figure 5.3: Absolute error $|u(0) - u_h^N(0)|$ versus initial step size $h = \frac{r^2}{n}$ obtained when Algorithm 3.2 and methods (5.5,5.7,5.8) are applied to the problem (5.2) with $g(x)$ from (5.9) with $k = (1, 1, 1)$ (left) and $k = (2, 4, 6)$ (right). The parameters used are $f = 0.05$, $N = 10^4$ and the dimension is $n = 3$. $\square$ is Algorithm 3.1, $\circ$ is (5.8), $*$ is (5.5) and $+$ is (5.7).

notation used in the previous section we have (compare with (5.6)),

$$\begin{aligned}
\Delta Z = Y\Big(& g(X)h + \Big(\partial_i g(X) + b^i(X)g(x)\Big)\frac{h^{\frac{3}{2}}}{2}\xi^i \\
& + \left(\frac{1}{2}\,\triangle\, g(X) + \left\langle b(X), \frac{\partial}{\partial X} g(X)\right\rangle\right)\frac{h^2}{2}\ \Big) \\
\Delta Y = Y\Big(& \left(b^i(X) + h\left(\frac{1}{2}\,\triangle\, b^i(X) + \left\langle b(X), \frac{\partial}{\partial X} b^i(X)\right\rangle\right)\right)\sqrt{h}\xi^i \\
& + \Big(\partial_i b^j(X) + b^i(X)b^j(X)\Big)\frac{h}{2}\Big(\xi^i\xi^j + \Xi^{ij}\Big) \\
& - \left(\frac{1}{2}\,\triangle\, b^i(X) + \left\langle b(X), \frac{\partial}{\partial X} b^i(X)\right\rangle\right)\frac{h^{\frac{3}{2}}}{2}\xi^i\ \Big)
\end{aligned}$$

where $Y = Y_k$ and $X = X_k$. In particular, an equation for Y has to be approximately solved in each step.

Figure 5.4 shows the results obtained for $k = (10, 9, \ldots, 10 - n + 1)$ in $n = 4, 6$ and 10 dimensions. To compare the approximate solutions for these different dimensions, we show

a plot of the relative error.

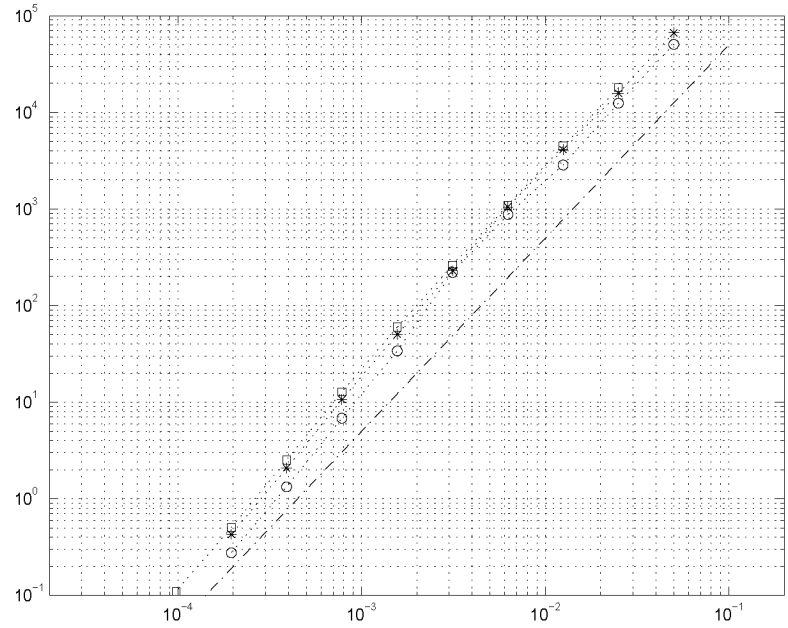

Figure 5.4: Relative error $|u(0) - u_h^N(0)|/u(0)$ versus initial step size h obtained when Algorithm 3.2 is applied to the problem (5.10) with $g(x)$ from (5.9) and $b(x)$ such that $b^i \partial_i u = 0$. Sample size is $N = 10^6$ and error bars are much smaller than plot symbols. $\circ$ is $n = 4$, $*$ is $n = 6$ and $\square$ is $n = 10$. A dash-dotted line with slope two ($- \cdot - \cdot -$) was added.

From Figure 5.4 we observe that the order of convergence does not change if the problem is modified such that it includes non-vanishing first order terms. In fact, convergence is still of $\mathcal{O}(h^2)$ as could be conjectured from the right plot in Figure 5.3 (where this order of convergence was observed for higher frequencies k^i).

Chapter 6

Conclusions and final remarks

Before concluding this part we make a few remarks about possible extensions.

6.1 Possible extensions

Several extensions of our Algorithm 3.2 to other problems relevant in the sciences are possible. First, an extension to parabolic PDEs in a bounded domain with Dirichlet boundary conditions based on the walk on cubes (Algorithm 3.1) is straightforward: Instead of considering D with boundary ∂D, one considers simply the diffusion in the cylinder $D \times [0, T)$. This extension can be done in an analogous fashion to [MT02], where the simplest random walk (5.5) is used to solve parabolic PDEs in a bounded domain $D \subset \mathbb{R}^n$ numerically. There, another procedure is introduced to find an approximation for the exit point $X(\tau)$. Nevertheless, the approach we previously adopted where h shrinks when approaching ∂D in order to prevent a possible excursion can also be used in $D \times [0, T)$. For instance, see [Mil95b] for a description with the walk on spheres in that situation. From this same paper [Mil95b, Section 7] we also see that an extension to PDEs with choices of a other than the identity matrix (see (2.1) and (2.4)) is possible. Then, instead of $\mathrm{d}X(t) = \mathrm{d}W(t)$ one has to solve the equation $\mathrm{d}X(t) = \sigma(X(t))\,\mathrm{d}W(t)$ with $\sigma(x)\sigma(x)^T = a(x)$, see (2.8). Close to the boundary (i.e. once the discrete random walk satisfies $X_k \in \Gamma_T$), the step size h has to be shrunk such that $X_k + \sigma(X_k)\Delta X \in \overline{D}$ for all ΔX. Of course, the operators (2.13) and resulting one step approximations are more complicated than those derived in Section 3.1. But still, in principle, we do not see any problem in deriving a corresponding algorithm similar to Algorithm 3.2. In that situation, RK-methods are especially desirable as the calculation of the derivatives required due to the complicated operators L and Λ_i might become too

tedious to compute in practice.

6.2 Conclusions

From the examples of three Poisson problems with Dirichlet boundary conditions in hyperspherical domains, we have seen that Monte-Carlo simulations of stochastic differential equations can indeed give very accurate results for the Dirichlet problem in n dimensions. The main problem to solve is the approximation of the path generating random walk inside the domain D. In particular, the resulting (discrete) Markov chain $(X_k)_k$ should not leave $\overline{D}$. We presented an algorithm (Algorithms 3.1 and 3.2) that uses bounded approximations for the Brownian increments to achieve this restriction.

From our examples, we saw that it is not difficult for the resulting random walk to find the boundary. Furthermore, as the dimension of the space is increased, not only does the canonical $\mathcal{O}(N^{-1/2})$ statistical error remain valid, but also the coefficient of $N^{-1/2}$ actually seems to decrease as the dimension n increases and termination of the walk at the boundary seems to become less sensitive to the initial step size. This is important good news and furthermore this conclusion seems to be valid for each of the methods tested: not only for a simple walk with $\Delta X^i = \pm\sqrt{h}$ for each coordinate i, but also for the higher order methods *walk on spheres* and *walk on cubes*. However, the global order of accuracy is sometimes not as expected from a theoretical point of view (see Table 5.4).

There is reason to believe that the three models previously discussed for the increments of Brownian motion have weaknesses. In one dimension, the walk on spheres procedure is the same as the simple 2-point method (simple random walk). The walk on cubes increment, which has the advantage that it lends itself readily to Runge-Kutta approaches (see [KP92, Mil95a, Pet98]), is not isotropically distributed in n-space. Thus, none of these models has all the properties we would wish. The walk on spheres increment is isotropically distributed, but the radial distribution does not do a good job of modelling the $p \sim \exp(-r^2/(2h))r^{n-1}$ radial density. Walk on cubes has a somewhat better radial distribution, but is not isotropic. A possible solution to this problem is as follows: After having generated $\Delta X = \sqrt{h}(\xi^i)_{i=1,\ldots,n}$, a random rotation could be applied. With a random orthogonal matrix R, a step then becomes $X_{k+1} = X_k + R\sqrt{h}(\xi^i_{k+1})_i$. See [Ste80] for an efficient technique to generate such random rotations.

In some cases the global order is the expected $\mathcal{O}(h^2)$, although in others it is only $\mathcal{O}(h)$. However, this is clearly better than the crude estimate $\mathcal{O}(h^{1/2})$ with an unsophisticated boundary finding algorithm.

It is our hope that with better bounded models for Brownian increments we will obtain improved levels of accuracy. The current results, however, have to be considered stimulating: higher dimensional problems do not seem any less tractable than lower dimensional ones.

On the other hand it is not completely satisfactory that the overall convergence is not always of order h^2. We are not able to say (or even prove) why convergence is sometimes only linear. Note, however, that also the higher order walk on spheres, (5.8), did not converge with the predicted order r^4 when applied to the exit problem but only with $\mathcal{O}(r^2)$ (Figure 5.2).

Nevertheless, these rates of convergence, $\mathcal{O}(h)$ ($\mathcal{O}(r^2)$ respectively) are still much better than the $\mathcal{O}(\sqrt{h})$ ($\mathcal{O}(r)$) convergence of a crude stopping procedure. On the other hand, an integration procedure as simple as the Euler scheme already converges with linear order when no boundaries are present. Furthermore, this scheme is completely derivative free, very simple, and therefore the resulting one step approximations can be derived and implemented in a straightforward fashion. – This is a big advantage, especially if the functions to be integrated are complicated and/or high dimensional. If linear convergence could be achieved for simulations in bounded domains, then this scheme would definitely have its advantages. Furthermore, if the need for boundary layers could be omitted, implementation in practice would become much simpler. It is these thoughts that give rise to the method that we will present in Part II.

Part II

An exit probability approach

This part discusses an approach to solve (2.4) by means of the representation (2.7,2.10) that is quite different from the attempt described in the previous part. Here, we will use a model for $I_{(i),h}$ which is exact in distribution, whereas in the walk on cubes approximate models for $I_{\alpha,h}$ were used. With these models we constructed a random walk with bounded increments which then led to an approximation that never left $\overline{D}$ (therefore did not allow excursions).

In this part, we will apply the Euler scheme to solve (2.10) approximately. This scheme uses an (unbounded) standard zero-mean, unit-variance normal random variable to model $I_{(i),1}$. After a step $X_k \to X_{k+1}$ an *a posteriori* test is performed to account for a possible excursion within the time interval (t_k, t_{k+1}), even if both $X_k, X_{k+1} \in D$. This test is based on a Brownian bridge. The bridge will be pinned at the discrete sample points before and after the step, namely at X_k (at t_k) and at X_{k+1} (at t_{k+1}).

The entire approach takes advantage of the fact, that an approximation of $u(x)$ via the representation (2.6) requires only *weak* approximations (see Definition 2.5)), i.e., loosely speaking, only the distributions have to be correct. We will construct random variables, which model the exit time τ within (t_k, t_{k+1}) – at least in one space dimension – "almost" exactly in distribution. In higher dimensions, we approximate ∂D locally by a half-space and apply the approximations found in $1d$ (see [Gob00] and in particular [Gob01]).

Clearly, this approximation is only meaningful if ∂D is locally flat, i.e. if $h \ll r_D$, where r_D denotes the radius of curvature of ∂D. Furthermore, we assume that D is such that for any $x \in D$ close to ∂D, (only) a unique segment of ∂D can be assigned "close" to x.

Once an exit of the path is detected (with "known" exit time τ), a corresponding exit point $X(\tau)$ is needed to evaluate the boundary condition ψ (see (2.4) and (2.6)). Again, we *sample* on the tangent hyper-plane to ∂D at a point "close" to X_k and X_{k+1}.

We will not be able to prove convergence of the resulting algorithm but we prove the exactness of the constructed random variables. We give numerical evidence for weak order one convergence and demonstrate numerically the superiority of our algorithm over other algorithms currently employed in the literature.

We focus mainly on Poisson's equation (5.2). Then, (2.8,2.10) reduce to

$$\mathrm{d}X(t) = \mathrm{d}W(t), \quad X(0) = x, \quad \mathrm{d}f(t) = g(X(t))\,\mathrm{d}t, \quad f(0) = 0 \tag{6.1}$$

and (2.6) becomes $u(x) = \mathbb{E}[\psi(X(\tau)) + f(\tau)]$. We have renamed $Z \to f$ (from equations (2.8c,2.10c) in Section 2.1 and corresponding equations in Part I) to prevent a possible confusion: Throughout this part we use the abbreviations $y = X_k$ and $z = X_{k+1}$. Note further, that a and b have two different meanings:

- When we talk about a PDE or a connected SDE, a and b mean the diffusion matrix and the drift vector respectively (or their corresponding coefficients in one dimension).

- In one space dimension, we denote by a and b the boundaries of an interval $(a, b) \subset \mathbb{R}$. Moreover, b denotes a general barrier.

In what follows, the actual meaning will always be clear from the context.

This part follows the structure of the proposed algorithm: We start with a one-dimensional situation and concentrate on the approximation of τ. We compare our algorithm numerically with other possible approaches. Statistical tests concerning the quality of the resulting distribution are shown. We then extend this algorithm to approximate τ to higher dimensions. Finally, we construct approximations of $X(\tau)$ and show results of simulations for domains with high dimensions.

Chapter 7

The one dimensional situation

The main ideas are derived and presented in the one dimensional setting. Even in this simple situation, the main difficulties one encounters when simulating SDEs in the presence of boundaries arise. For example, if a diffusion is stopped once it reaches (or exceeds) a certain level, usual convergence rates are lost. This well known fact does not only hold if one considers path wise convergence (i.e. strong approximations) but also holds in the case of weak approximations.

7.1 Killed diffusions in $1d$ and related formulae

We review the basic ideas and formulae needed to weakly approximate killed diffusions. These results are taken from [Man99, Gob00, Gob01], and see also the references therein. Convergence proofs for killed diffusions can be found in the two latter references and additional refinements in [Hau00].

We wish to simulate the diffusion with values in $\mathbb{R}$ defined by

$$X(t) = x + \int_0^t b(X(s))\,\mathrm{d}s + \int_0^t \sigma(X(s))\,\mathrm{d}W(s). \tag{7.1}$$

Let τ be defined as in (2.5), $\tau = \inf_t\{X(t) \notin D\}$, and let T with $0 < T < \infty$ be given. Suppose that we were interested in computing an approximation to

$$\mathbb{E}[f(X(T))\mathbf{1}_{\{T<\tau\}}] \quad \text{with} \quad X(0) = x \in D \tag{7.2}$$

with measurable $f : \mathbb{R} \to \mathbb{R}$. Neglecting in (3.4) higher order terms (corresponding to the hierarchical set $\Gamma_1 = \{v, (0), (1)\}$) we get the Euler scheme (or Euler-Maruyama

scheme [Mar53, Mar55]) for the above SDE. For a fixed step size $h = T/M$ (with $M \in \mathbb{N}$) this takes the form ($X_0 = x$)

$$X_{k+1} = X_k + b(X_k)h + \sigma(X_k)\sqrt{h}\xi_{k+1}, \quad k = 0, 1, \ldots, M-1, \tag{7.3}$$

where ξ_{k+1} is an i.i.d standard normal random variable (zero-mean and unit-variance),

$$\xi_{k+1} \sim \mathcal{N}(0, 1).$$

Remark 7.1. In the case of the Euler scheme, one could also use simpler random variables ξ_k, for example, the three-point variable as defined in (3.6) preserves weak order one. For this order of weak convergence, even a two-point random variable with law $\mathbb{P}[\xi_k = \pm 1] = \frac{1}{2}$ would be enough. Then, of course, one could apply again a boundary finding algorithm similar to Algorithm 3.1 to prevent the resulting path from leaving D. Milstein called the resulting random walk the *Simplest random walk for Dirichlet problems*, see [Mil01] and [MT02] for a further improvement. But this is not the goal here: In this part we present ideas which preserve weak order one convergence even for normals $\xi_k \sim \mathcal{N}(0, 1)$.

For $k < M$ let $y = X_k \in D$ and $z = X_{k+1}$ obtained by (7.3). Clearly, if $z \notin D$, then $\tau \leq T$ and the corresponding path is killed, i.e. does not contribute to the expectation in (7.2). If $z \in D$, the situation is more subtle, as the path might have taken an excursion, implying $\tau < t_{k+1}$.

Let

$$S_t(x) = \sup_{0 \leq s \leq t} X(s), \quad X(0) = x.$$

We further introduce the first hitting time of level $b \neq x$ as

$$H_b(x) = \inf_{t>0} \{X(t) = b\}, \quad X(0) = x. \tag{7.4}$$

We have for $y, z < b$ [Gob00] (formulae can be found in [BS02, p.198] and the derivation in [RW00a])

$$\begin{aligned} \mathbb{P}_y[H_b(y) \leq t; X(t) \in \mathrm{d}z] &= \mathbb{P}_y[S_t(y) \geq b; X(t) \in \mathrm{d}z] \\ &= \exp\left(-\frac{2}{\sigma^2(y)t}(b-y)(b-z)\right)\mathbf{1}_{\{t>0\}} \end{aligned} \tag{7.5}$$

where $\mathbb{P}_y$ denotes the law connected to the process $X(t)$ with $X(0) = y$.

For $D = (-\infty, b)$ with $x < b$ we have $\tau = H_b(x)$. If both $y, z < b$, the naive approach is to set $\tau \geq t_{k+1}$ and to continue integration. Following Mannella [Man99] and Gobet [Gob00] a path is nevertheless killed, if (from (7.5))

$$u \leq \exp\left(-\frac{2}{\sigma^2(y)h}(b-y)(b-z)\right) \quad \text{where} \quad h = t_{k+1} - t_k. \tag{7.6}$$

Here, $u \sim \mathcal{U}(0,1)$ is a random number uniformly distributed in $(0,1)$, independent from ξ_{k+1} in (7.3).

Mannella also cites in his paper a work by Strittmatter for this test – unfortunately, we were not able to obtain this work. See [BC02] for further refinement of formula (7.6).

For $D = (a,b)$ with $a < x < b$ we have $\tau = H_a \wedge H_b$. For $a < y, z < b$, it is known that [RY91, BCI99, Gob00]

$$\mathbb{P}_y[\tau \le h; X(h) \in \mathrm{d}z] = \sum_{k=-\infty}^{k=\infty} \left\{ \exp\left(-\frac{2k(b-a)(k(b-a)+z-y)}{h\sigma(y)^2} \right) - \exp\left(-\frac{2(k(b-a)+y-b)(k(b-a)+z-b)}{h\sigma(y)^2} \right) \right\} \mathbf{1}_{\{h>0\}}, \tag{7.7}$$

a formula rather complicated to evaluate in practice. Furthermore, for higher dimensions simple expressions are known only for half-spaces. And even in one dimension the local approximation of D by a half-space is already favorable.

If $y, z \in (a,b)$ are "close" to b, a test according to (7.6) is performed; if y, z are "close" to a, the path is killed if (by symmetry)

$$u \le \exp\left(-\frac{2}{\sigma^2(y)h}(a-y)(a-z) \right), \quad u \sim \mathcal{U}(0,1). \tag{7.6'}$$

Clearly, a test performed according to (7.7) would be the correct one for $D = (a,b)$. However, using the tests (7.6,7.6′) is much simpler and computationally more efficient. If h is sufficiently small, applying the combined test (7.6,7.6′) only to the "closest" boundary is hence an adequate strategy.

We define "close" as follows:

Definition 7.2 (Closest boundary for $D = (a,b)$). For $y, z \in D = (a,b)$ let $\rho_a = (y-a) \wedge (z-a)$ and $\rho_b = (b-y) \wedge (b-z)$. The *closest boundary* B is then

$$B = \begin{cases} a, & \rho_a < \rho_b \\ b, & \rho_a \ge \rho_b. \end{cases}$$

Remark 7.3. The event of both boundaries being equally far away has zero probability theoretically and an almost vanishing probability computationally. Additionally, in that case either $y = z$ (with probability zero) or y and z are symmetric wrt. the point $(a+b)/2$. In the latter case, both boundaries are far away, and the exit probability tends to zero exponentially fast. The error arising if we set $B = b$ even in these worst case situations is therefore negligible.

7.2 From killed to stopped diffusions

In our case, instead of the evaluation of $\mathbb{E}[f(X(T))\mathbf{1}_{\{T<\tau\}}]$ (see (7.2)) we have to evaluate functionals of the form $\mathbb{E}[f(X(\tau))]$ (see (2.6,2.7)). We are interested in the actual value of τ, rather than being satisfied by the assertion that $\tau < T$ for some predefined (deterministic) T. In other words, one wants to know (in a statistical sense) *when* the first exit time actually took place, rather than just asking *if* the exit actually occured.

To explain the main ideas we restrict ourselves to the case $\sigma = 1$ and $b = 0$ in (7.1), i.e. $\mathrm{d}X(t) = \mathrm{d}W(t)$ with $X(0) = x$. For simplicity, we show our ideas for $B = b$. The corresponding formulae for $B = a$ are obtained by symmetry arguments.

We need the concept of a Brownian bridge. Let $X^{y,h,z}(t)$ be a Brownian bridge starting at $t = 0$ at y and ending at $t = h$ at z (i.e. the bridge is pinned in time-space-coordinates at $(0, y)$ and at (h, z) and has $\mathbb{E}[X^{y,h,z}(t)] = y + (z - y)t/h$ and $\mathrm{Cov}(X^{y,h,z}(t), X^{y,h,z}(s)) = (s \wedge t) - st/h$ [IW89]). Denote its law by $\mathbb{P}_{y,h,z}[\cdot]$, see the appendix in [Szn98].

7.2.1 The case $y, z < b$

Formula (7.5) gives the probability, that b was hit before t, given $X(0) = y$ and $X(t) \in \mathrm{d}z$. Alternatively, this gives us the distribution F of the first hitting time of b wrt. Brownian bridge measure,

$$F^{y,h,z}_{H_b(y)}(t) \equiv \mathbb{P}_{y,h,z}[H_b(y) \le t] = e^{-\frac{2}{t}(b-y)(b-z)}\mathbf{1}_{\{t>0\}}, \quad y < b \quad \text{and} \quad z \le b. \tag{7.8}$$

The idea is now to generate a random variable with distribution (7.8).

Lemma 7.4. *Let $u \sim \mathcal{U}(0, 1)$ be uniformly $(0, 1)$ distributed. Then*

$$\mathcal{H}_1 = -\frac{2(b - y)(b - z)}{\log u} \tag{7.9}$$

has distribution (7.8).

Proof. This is a direct application of the inverse transform method: $\mathcal{H}_1(u) = (F^{y,h,z})^{-1}(u)$, see [KP92, p.12]. □

After a preliminary step resulting in $z < b$, we generate $\mathcal{H}_1$ according to (7.9). If $\mathcal{H}_1 \le h$, the path hit b (in a statistical sense) no later than at h, namely at $t_k + \mathcal{H}_1$.

Remark 7.5. Clearly, generating $\mathcal{H}_1$ according to formula (7.9) and comparing to h does not differ essentially in terms of computational cost from the evaluation of the right hand side of (7.6), generating $u \sim \mathcal{U}(0, 1)$ and comparing these two values.

The extension to general diffusions of the form (7.1) is now obvious from (7.5): In that case, the random time $\tilde{\mathcal{H}}_1$ is generated as

$$\tilde{\mathcal{H}}_1 = -\frac{2(b-y)(b-z)}{\sigma(y)^2 \log u} \tag{7.9'}$$

where $u \sim \mathcal{U}(0,1)$ is again uniformly $(0,1)$ distributed.

7.2.2 The case $y < b$ and $z \geq b$

In this case, clearly, b was hit within $(t_k, t_{k+1}]$. Nevertheless, we are interested in the actual value of the first hitting time.

Lemma 7.6. *For $y < b \leq z$ the density of $H_b(y)$ is given by the formula*

$$\mathbb{P}_{y,h,z}[H_b(y) \in \mathrm{d}t] = (b-y)\sqrt{\frac{h}{2\pi(h-t)t^3}} \cdot \exp\left(-\frac{((b-y)h - t(z-y))^2}{2ht(h-t)}\right) \mathbf{1}_{\{0<t<h\}}\,\mathrm{d}t \tag{7.10a}$$

$$= \frac{\tilde{b}}{\sqrt{2\pi h\tilde{t}^3(1-\tilde{t})}} \exp\left(-\frac{1}{2h}\frac{(\tilde{b}-\tilde{t}\tilde{z})^2}{\tilde{t}(1-\tilde{t})}\right) \mathbf{1}_{\{0<\tilde{t}<1\}}\,\mathrm{d}\tilde{t} \tag{7.10b}$$

where $\tilde{b} = b - y$, $\tilde{z} = z - y$ and $\tilde{t} = t/h$.

Proof. Let

$$p(t;x,y) = \frac{1}{\sqrt{2\pi t}} \exp\left(-\frac{(x-y)^2}{2t}\right)$$

be the Gaussian transition density. Using absolute continuity of the measures $\mathbb{P}_y$ and $\mathbb{P}_{y,h,z}$ we have [BS02, p.67]

$$\mathbb{P}_{y,h,z}[H_b(y) \leq h] = p(h;y,z)^{-1} \int_0^h p(h-t;b,z)\mathbb{P}_y[H_b(y) \in \mathrm{d}t]$$

with (see for example [RW00a, (13.5),p.26])

$$\mathbb{P}_y[H_b(y) \in \mathrm{d}t] = \frac{|b-y|}{\sqrt{2\pi t^3}} \exp\left(-\frac{(b-y)^2}{2t}\right) \mathbf{1}_{\{t>0\}}\,\mathrm{d}t. \tag{7.11}$$

Together,

$$\mathbb{P}_{y,h,z}[H_b(y) \le h] = \frac{|b-y|}{\sqrt{2\pi}} \int_0^h \sqrt{\frac{h}{t^3(h-t)}} \exp\left(-\frac{1}{2}\left(-\frac{(z-y)^2}{h} + \frac{(b-z)^2}{h-t} + \frac{(b-y)^2}{t}\right)\right) \mathrm{d}t. \quad (7.12)$$

Hence,

$$\mathbb{P}_{y,h,z}[H_b(y) \le h] = \int_0^h \mathbb{P}_{y,h,z}[H_b(y) \in \mathrm{d}t].$$

Recalling that $y < b \le z$ and rearranging the terms in the exponent, (7.10a) follows. Substituting the tilde variables introduced above and using $\mathrm{d}t = h\,\mathrm{d}\tilde{t}$, one gets (7.10b). □

Remark 7.7. For the standard bridge pinned at $(0, 0)$ and $(1, z)$ one gets the simpler formula

$$\mathbb{P}_{0,1,z}[H_b(0) \in \mathrm{d}t] = \frac{b}{\sqrt{2\pi t^3(1-t)}} \exp\left(-\frac{1}{2}\frac{(b-tz)^2}{t(1-t)}\right) \mathbf{1}_{\{0<t<1\}}\,\mathrm{d}t. \quad (7.13)$$

Some elementary manipulations in the exponent show that this formula is equivalent to the density given in [LS89, Lemma 3].

We next show how to sample from the density (7.10).

Definition 7.8 (Inverse Gaussian distribution). A random variable X follows the *inverse Gaussian distribution* with parameters $\lambda > 0$, $\mu > 0$ if it has the density function [FC78, Was68, Was69, RW68]

$$\mathbb{P}[X \in \mathrm{d}x] = \sqrt{\frac{\lambda}{2\pi x^3}} \exp\left(-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right) \mathbf{1}_{\{x>0\}}\,\mathrm{d}x.$$

We denote such a random variable by the symbol $X \sim \mathcal{IG}(\lambda, \mu)$.

Lemma 7.9. *For the bridge from* $(0, 0)$ *to* $(1, z)$ *in the case* $0 < b \le z$ *it holds that*

$$\mathbb{P}_{0,1,z}[H_b(0) \in \mathrm{d}t] = \frac{b}{\sqrt{2\pi x^3}} \exp\left(-\frac{(z-b)^2}{2}\frac{\left(x - \frac{b}{z-b}\right)^2}{x}\right) \mathbf{1}_{\{x>0\}}\,\mathrm{d}x$$

where $\mathrm{d}t = \mathrm{d}x/(1+x)^2$.

Proof. Write $\mathbb{P}_{0,1,z}[H_b(0) \in \mathrm{d}t] = p(t)\mathbf{1}_{\{0<t<1\}}\,\mathrm{d}t$ with

$$p(t) = \frac{b}{\sqrt{2\pi t^3(1-t)}} \exp\left(-\frac{1}{2}\frac{(b-tz)^2}{t(1-t)}\right)$$

where $0 \le t \le 1$. By the substitution $x = t/(1-t)$ (implying $t = x/(1+x)$) with $x \ge 0$ and $\mathrm{d}t = \mathrm{d}x/(1+x)^2$ we find

$$p(t)\,\mathrm{d}t = \frac{b}{\sqrt{2\pi x^3}}(1+x)^2 \exp\left(-\frac{z^2}{2x}\left(\frac{b}{z} - \left(1-\frac{b}{z}\right)x\right)^2\right)\frac{\mathrm{d}x}{(1+x)^2}.$$

With $\xi = b/z$ and $\eta = 1-\xi$, this yields

$$p(t)\,\mathrm{d}t = \frac{b}{\sqrt{2\pi x^3}} \exp\left(-\frac{z^2\eta^2}{2}\frac{(\xi/\eta - x)^2}{x}\right)\mathrm{d}x.$$

Now

$$\frac{\xi}{\eta} = \frac{b}{z-b} \quad \text{and} \quad z^2\eta^2 = (z-b)^2$$

so

$$p(t)\,\mathrm{d}t = \frac{b}{\sqrt{2\pi x^3}} \exp\left(-\frac{(z-b)^2}{2}\frac{\left(x - \frac{b}{z-b}\right)^2}{x}\right)\mathrm{d}x.$$

□

Introducing $\lambda = b^2$ and $\mu = b/(z-b)$ we have

$$p(t)\,\mathrm{d}t = \sqrt{\frac{\lambda}{2\pi x^3}} \exp\left(-\lambda\frac{(x-\mu)^2}{2\mu^2 x}\right)\mathrm{d}x$$

and we reformulate the result of Lemma 7.9.

Remark 7.10. If X is following an inverse Gaussian distribution with parameters $\lambda = b^2$ and $\mu = b/(z-b)$ ($X \sim \mathcal{IG}(\lambda, \mu)$), then $t = X/(1+X)$ is a random variable with density (7.13).

Corollary 7.11. *For the general bridge, we have that if X is following an inverse Gaussian distribution with parameters $\lambda = (b-y)^2/h$ and $\mu = (b-y)/(z-b)$, then the random variable $t = hX/(1+X)$ has density* (7.10).

Proof. By the translational invariance of Brownian motion

$$\mathbb{P}_{y,h,z}[H_b(y) \in \mathrm{d}t] = \mathbb{P}_{0,h,\tilde{z}}[H_{\tilde{b}}(0) \in \mathrm{d}t],$$

where (we recall that) $\tilde{z} = z - y$ and $\tilde{b} = b - y$. From (7.10a)

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

where $\tilde{t} = t/h$. Substituting $x = \tilde{t}/(1-\tilde{t})$ (or $\tilde{t} = x/(1+x)$, hence $t = hx/(1+x)$), one finds, with $\sqrt{h}\hat{b} = \tilde{b}$ and $\sqrt{h}\hat{z} = \tilde{z}$

$$\mathbb{P}_{0,h,\tilde{z}}[H_{\tilde{b}}(0) \in \mathrm{d}t] = \frac{\hat{b}}{\sqrt{2\pi x^3}} \exp\left(-\frac{\left(\hat{z}-\hat{b}\right)^2}{2} \frac{\left(x - \frac{\hat{b}}{\hat{z}-\hat{b}}\right)^2}{x}\right) \mathbf{1}_{\{x>0\}}\, \mathrm{d}x.$$

Introducing $\lambda = \hat{b}^2 = \tilde{b}^2/h = (b-y)^2/h$ and $\mu = \hat{b}/(\hat{z}-\hat{b} = \tilde{b}/(\tilde{z}-\tilde{b}) = (b-y)/(z-y)$ the result follows as in the proof of Lemma 7.9. □

Shuster [Shu68] noted that if $x \sim \mathcal{IG}(\lambda, \mu)$ then $\lambda(x-\mu)^2/(\mu^2 x)$ is $\chi^2(1)$ distributed. A chi-square distributed random variable with one degree of freedom can easily be generated by the square of a standard normal ($\sim \mathcal{N}(0,1)$) random variable [AS64, 26.4,p.940]. Based on this observation, Michael et al. presented an algorithm to generate random variables Xs following the inverse Gaussian distribution. There, a quadratic equation has to be solved and one root is picked with the correct probability from the two possibilities, see [MSH76, Dev86] and [Dag88, Sec. 3.10].

We show this procedure in the context of generating $\mathcal{H}_2$ with density (7.10a) in Algorithm 7.1. Recall that $y = X_k$, $z = X_{k+1}$ and that $b \in (y, z)$ denotes the boundary.

7.3 Numerical experiments

We show results of extensive tests performed with the algorithm derived in the previous section. For weak approximations, path wise convergence is not required, whereas a good approximation of the distribution is essential. In our case, special emphasis is placed on the computation of the first exit time of a process from a bounded domain. With this in mind, we start with a statistical test where we compare the numerically obtained density of a simple first hitting time (i.e. a histogram) with the known analytical density. We compare our algorithm with a variety of other approaches to the simulation of stopped diffusions. Some of these approaches appeared in the literature, whilst others are possible heuristic attempts. Then, we show the performance of our algorithm when applied to the numerical solution of some one dimensional Dirichlet problems via the representation (2.6).

Algorithm 7.1 Generate a random variable $\mathcal{H}_2$ with density (7.10a).

$b \leftarrow b - y$
$z \leftarrow z - y$
$\lambda \leftarrow b^2/h$
$\mu \leftarrow b/(z-b)$
Generate $n \sim \mathcal{N}(0, 1)$
$X \leftarrow n^2$
{generate a random variable $w \sim \mathcal{IG}(\lambda, \mu)$ [MSH76, p.89][Dev86, p.149]}

$$x \leftarrow \mu + \frac{\mu^2 X}{2\lambda} - \frac{\mu}{2\lambda}\sqrt{4\mu\lambda X + \mu^2 X^2}$$

if $\mathcal{U} \leq \mu/(\mu + x)$ **then** {pick this root with probability $\mu/(\mu + x)$}
    $w \leftarrow x$
else
    $w \leftarrow \mu^2/x$
end if{now $w \sim \mathcal{IG}(\lambda, \mu)$}

$$\mathcal{H}_2 \leftarrow h\frac{w}{1+w}$$

7.3.1 Testing the resulting hitting time distribution

Suppose our goal is to compute numerically the first hitting time of level $b = 1$ of a Brownian motion starting at $x = 0$ at $t = 0$. The corresponding density is (from (7.11))

$$\mathbb{P}_0 [H_1(0) \in \mathrm{d}t] = \frac{e^{-\frac{1}{2t}}}{\sqrt{2\pi t^3}} \mathbf{1}_{\{t>0\}} \, \mathrm{d}t. \tag{7.14}$$

It has its maximum at $t = \frac{1}{3}$ where it forms a non symmetrically shaped peak, and it has a very long tail, see Figure 7.1. It is therefore natural to perform tests trying to answer the following two questions: How good do the algorithms approximate (i) the peak and (ii) the tail of the density (7.14)?

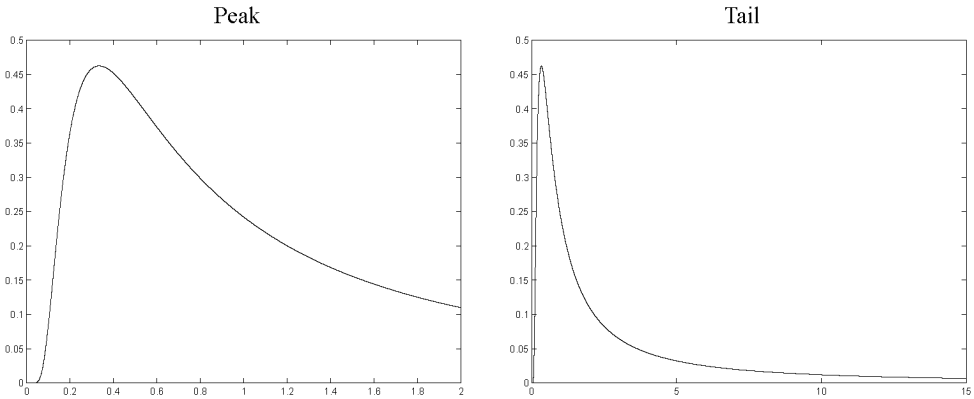

Figure 7.1: The density of the first hitting time of level one for a Brownian motion starting at $x = 0$ at time $t = 0$ (formula (7.14)). Note the different scalings used on the t-axis: on the left $0 \leq t \leq 2$, whereas $0 \leq t \leq 15$ on the right.

We summarize our algorithm to compute the first hitting time of $b > 0$ for one sample path with step size $h > 0$ in Algorithm 7.2.

Algorithm 7.2 Compute an approximation $\overline{H}_b$ to $H_b(0)$.

$y \leftarrow 0$ {initialization}
while true **do**
  $z \leftarrow y + \sqrt{h}\xi$ with $\xi \sim \mathcal{N}(0, 1)$ {preliminary Euler step}
  if $z < b$ **then**
    Generate $\mathcal{H}_1$ according to (7.9)
    if $\mathcal{H}_1 \leq h$ **then**
      $\overline{H}_b \leftarrow t + \mathcal{H}_1$ {excursion at $t + \mathcal{H}_1$ detected}
      STOP
    else
      $y \leftarrow z$ and $t \leftarrow t + h$ {no excursion – continue integration}
    end if
  else
    Generate $\mathcal{H}_2$ according to Algorithm 7.1
    $\overline{H}_b \leftarrow t + \mathcal{H}_2$
    STOP
  end if
end while

7.3.1-a) Experimental setup

We first note that $\mathbb{E}_0[H_1(0)] = \infty$. Furthermore, the density is almost zero for $t \approx 0$. We therefore compute a histogram only for $t \in [T_0, T_1]$ with $0 \leq T_0 < T_1 < \infty$ fixed. If the size of the bins is very small, we choose $T_0 > 0$ such that the first bin is hit with sufficiently high probability. If a simulated path ran longer than T_1 it was stopped and contributed only to the tail of the corresponding histogram (which was not included into the χ^2-test). To measure the approximation quality we performed a χ^2-test over two different sets of (equidistant) time intervals (the bins of the histogram). The number of bins (and hence the number of degree of freedoms in the χ^2-distribution) were chosen differently when testing the approximations of the peak and of the tail respectively. As we wished to take account of the rule of thumb, which states that the number of realizations N and the number of bins N_b should approximately follow the relation $N_b \approx N^{\frac{2}{5}}$, we had to take different numbers of realizations as well. Our choices are summarized in Table 7.1. Note that the number of DOFs equals the number of bins minus one.

Table 7.1: Setup for the two different χ^2-tests.

	Peak test	Tail test
number of bins	$N_b = 95$	$N_b = 250$
covered t-range	$T_0 = 0.05$, $T_1 = 1$	$T_0 = 0$, $T_1 = 250$
bin size	0.01	1
bins	$0.05, 0.06, \ldots, 1.0$	$0, 1, \ldots, 250$
number of realizations	$N = 1e5$	$N = 1e6$

As a measure of approximation we computed [KS73]

$$\overline{\chi}^2 = \sum_{i=0}^{N_b-1} \frac{(N_i - Np_i)^2}{Np_i} \tag{7.15}$$

where p_i denotes the relative expected frequency of bin i,

$$\frac{1}{N} p_i = \int_{\text{bin } i} p(t)\, \mathrm{d}t. \tag{7.16}$$

In (7.16) $p(t)$ is the density given in (7.14). In (7.15), N_i denotes the number of trials that fell in bin i (bin i's *hit count*). The integrals in (7.16) where computed using MAPLE. Asymptotically, $\overline{\chi}^2$ has a χ^2-distribution with $N_b - 1$ DOFs [Mos86, p.402].

7.3.1-b) Comparing algorithms

The algorithms we compare differ essentially in the following points:

Numerical integration method: To the best of our knowledge there are two methods which are currently employed to numerically integrate the simple SDE $\mathrm{d}X(t) = \mathrm{d}W(t)$ with $X(0) = 0$:

Gaussian increments: The most widely used method is also the most natural one: $X_{k+1} = X_k + \sqrt{h}\xi, \xi \sim \mathcal{N}(0, 1)$.

Exponential time stepping: This method is especially favorable for the approximation of diffusions in the presence of boundaries. The increment has a symmetric exponential distribution with density

$$\mathbb{P}[X_{k+1} - X_k \in \mathrm{d}x] = \frac{\nu}{2} e^{-\nu|x|}\,\mathrm{d}x, \quad \nu = \sqrt{2\lambda}. \tag{7.17}$$

The time step is a *random* time which is exponentially distributed with parameter $\lambda > 0$:

$$\mathbb{P}[t_{k+1} - t_k \in \mathrm{d}t] = \lambda e^{-\lambda t} \mathbf{1}_{\{t>0\}}\,\mathrm{d}t. \tag{7.18}$$

Note that the time step is never generated (although it would be easily accomplished following (7.18)), because as soon as it is known, $\Delta X = X_{k+1} - X_k$ is again $\mathcal{N}(0, t_{k+1} - t_k)$ distributed. The fact that $\mathbb{E}[t_{k+1} - t_k] = \frac{1}{\lambda}$ leads to the choice $\lambda = \frac{1}{h}$ for the comparison of results in experiments. Furthermore, as no time step is actually generated, the time passed until (and including) t_k is approximated by k/λ, i.e., it is smeared out. For further details on this method see [JL00, JL03] and for the formulae see [BS02, pp.153 ff].

Stopping criterion: The most trivial attempt is of course to stop as soon as $z = X_{k+1} \geq 1$. As motivated thoroughly, for the case when $z < 1$, we should check the path for an excursion above 1 within the time step. Either a test of the form as presented in (7.6) is possible or the test described in and after Lemma 7.4 can be evaluated. In the case of exponential time stepping, the analogon to (7.6) is given by (remember $y = X_k$ and $z = X_{k+1}$)

$$u \leq e^{-2\nu(1-y\vee z)}, \quad \nu = \sqrt{2\lambda}, \quad u \sim \mathcal{U}(0, 1). \tag{7.19}$$

Computation of exit time: When one decides to stop the integration for a path during (or at the beginning or end of) a time step $t_k \to t_{k+1}$ because either $z \geq 1$ or an excursion was detected, then one must choose $H_1(0) \in [t_k, t_{k+1}]$. In both cases, there are many possibilities for this choice:

Heuristic approximations: One can choose, for example, t_k, t_{k+1} or a certain deterministic value (depending only on y, z and $b = 1$) inbetween.

Statistical approximations: Here again, one has the choice of less or more justifiable approximations:

- a random value uniformly distributed within t_k and t_{k+1}.
- a value derived from the known law of $H_1(0)$ of the bridge pinned at (t_k, y) and (t_{k+1}, z), see (7.5) and (7.10) respectively.

A summary of all the tested algorithms is shown in Table 7.2. Details and necessary calculations are given below.

We discuss details of some of the methods summarized in Table 7.2 and use the labels introduced therein (**K1-K4**, **D1-D7** and **E1-E4**).

K1-K4: These are straight forward extensions from the method for simulating killed diffusions: In the case that $z < 1$, an excursion test as in formula (7.6) is performed. The approximation for the first hitting time is either at the beginning, at the end or in the middle of the time step in which the path was killed by the algorithm.

D1-D7: These generate $\mathcal{H}_1$ as in (7.9), Lemma 7.4 and then compare it to the step size h in order to test for an excursion when both $y, z < 1$. If an excursion was detected, i.e. $\mathcal{H}_1 \leq h$, the approximation for the first hitting time is $t_k + \mathcal{H}_1$. Note that $\mathcal{H}_1$ is exact in distribution. The difference arises in the approximation of $H_1(y)$ (with $y = X_k$ the position of the approximate path at t_k) in the case that the discrete path clearly hit $b = 1$ ($z \geq 1$). Our proposed algorithm (**D7**) is compared with the following methods:

D1 and D2: These are the simplest attempts: either stop at the beginning of the corresponding time step (**D1**), or at its midpoint. These approximations ignore the location of $b \in (y, z]$ completely.

D3 (linear interpolant): Here we choose the linear interpolant as an approximation for $H_1(y) \in (0, h]$, namely

$$H_1(0) \approx t_k + \hat{H}_{lip} \quad \text{with} \quad \hat{H}_{lip} = h\frac{1-y}{z-y}. \tag{7.20}$$

This approach is justified by a result which is interesting in its own right and is given in the following lemma.

Table 7.2: Summary of the algorithms for the numerical computation of an approximation to $H_1(0)$ for a Brownian motion starting at $x = 0$ at time $t = 0$ (see (7.14)). The algorithm that was derived in Section 7.2, Algorithm 7.2, is highlighted in bold font. In the first column, the symbol that is used in the plots (Figures 7.2-7.6) is given. Note that in later plots different symbols might be used.

Label		excursion test for $y, z < 1$	approximation of $H_1(0)$ for $y, z < 1$	relevant equation	approximation of $H_1(0)$ for $y < 1 \le z$	relevant equation or Algorithm
		Gaussian increments: $X_{k+1} = X_k + \mathcal{N}(0, h)$				
$\cdot\cdot\circ\cdot\cdot$	(T)	—	—		t_k	
$\cdot\cdot\square\cdot\cdot$	(K1)	(7.6)	t_k		t_k	
$\cdot\cdot\diamond\cdot\cdot$	(K2)	(7.6)	$t_k + h/2$		$t_k + h/2$	
$\cdot\cdot*\cdot\cdot$	(K3)	(7.6)	t_{k+1}		t_{k+1}	
$\cdot\cdot+\cdot\cdot$	(K4)	(7.6)	$t_k + h/2$		t_k	
$\cdot\cdot\square\cdot\cdot$	(D1)	Lemma 7.4	$t_k + \mathcal{H}_1$	(7.9)	t_k	
$\cdot\cdot\diamond\cdot\cdot$	(D2)	Lemma 7.4	$t_k + \mathcal{H}_1$	(7.9)	$t_k + h/2$	
$\cdot\cdot*\cdot\cdot$	(D3)	Lemma 7.4	$t_k + \mathcal{H}_1$	(7.9)	$t_k + \hat{H}_{lip}$	(7.20)
$\cdot\cdot\square\cdot\cdot$	(D4)	Lemma 7.4	$t_k + \mathcal{H}_1$	(7.9)	$t_k + \hat{H}_{uni}$	(7.23)
$\cdot\cdot\diamond\cdot\cdot$	(D5)	Lemma 7.4	$t_k + \mathcal{H}_1$	(7.9)	$t_k + \alpha h$	(7.24)
$\cdot\cdot*\cdot\cdot$	(D6)	Lemma 7.4	$t_k + \mathcal{H}_1$	(7.9)	$t_k + \hat{H}_E$	(7.25)
$-\circ-$	**(D7)**	**Lemma 7.4**	$t_k + \mathcal{H}_1$	**(7.9)**	$t_k + \mathcal{H}_2$	**7.1**
		Exponential time stepping: $X_{k+1} - X_k$ according to (7.17)				
$\cdot\cdot\square\cdot\cdot$	(E1)	(7.19)	k/λ		k/λ	
$\cdot\cdot\diamond\cdot\cdot$	(E2)	(7.19)	$k/\lambda + \hat{H}_e$	(7.26)	k/λ	
$\cdot\cdot*\cdot\cdot$	(E3)	(7.19)	$k/\lambda + \hat{H}_e$	(7.26)	$k/\lambda + \hat{H}_e$	(7.26)
$\cdot\cdot+\cdot\cdot$	(E4)	(7.19)	$(k+1)/\lambda$		$(k+1)/\lambda$	

Lemma 7.12. *Fix $y = X_k$ and $z = X_{k+1}$ and assume $y < b < z$. Then, the expectation $\mathbb{E}_{y,h,z}[H_b(y)]$ is a linear function of $b - y$, i.e.*

$$\mathbb{E}_{y,h,z}[H_b(y)] = C(b - y) \tag{7.21}$$

where C depends only on the distance $X_{k+1} - X_k$ and the time step h.

Proof. For simplicity of notation, we set $y = 0$ and $h = 1$ and show the calculations only for the bridge pinned at $(0, 0)$ and $(1, z)$ with $0 < b < z$. The general bridge follows from these results in obvious manner.

To show $\mathbb{E}_{0,1,z}[H_b(0)] = Cb$, we calculate the expectation explicitly. With (7.13) we have

$$\mathbb{E}_{0,1,z}[H_b(0)] = \frac{b}{\sqrt{2\pi}} \int_0^1 \exp\left(-\frac{1}{2}\frac{(b-tz)^2}{t(1-t)}\right) \frac{\mathrm{d}t}{\sqrt{t(1-t)}}$$
$$= \frac{b}{\sqrt{2\pi}} \int_0^1 \exp\left(-\frac{z^2}{2}\frac{(\xi-t)^2}{t(1-t)}\right) \frac{\mathrm{d}t}{\sqrt{t(1-t)}}$$

where $\xi = b/z$. Introduce the transformation

$$x = \frac{t}{1-t} \quad \text{with} \quad \frac{\mathrm{d}t}{\sqrt{t(1-t)}} = \frac{\mathrm{d}x}{\sqrt{x}(1+x)}.$$

Let $\eta = 1 - \xi$, then

$$\frac{(\xi-t)^2}{t(1-t)} = \frac{(\xi-\eta x)^2}{x}.$$

With

$$p = \frac{z^2\eta^2}{2} = \frac{(z-b)^2}{2} \quad \text{and} \quad q = \frac{z^2\xi^2}{2} = \frac{b^2}{2}$$

we have

$$\mathbb{E}_{0,1,z}[H_b(0)] = \frac{b}{\sqrt{2\pi}} e^{z^2\xi\eta} \int_0^\infty \frac{e^{-px-\frac{q}{x}}}{\sqrt{x}(1+x)}\,\mathrm{d}x.$$

We calculate the remaining integral

$$I = \int_0^\infty \frac{e^{-px-\frac{q}{x}}}{\sqrt{x}} \frac{1}{1+x}\,\mathrm{d}x = \int_0^\infty \frac{e^{-px-\frac{q}{x}}}{\sqrt{x}} \int_0^\infty e^{-s(1+x)}\,\mathrm{d}s\,\mathrm{d}x.$$

Changing the order of integration and using [PBM86, (2.3.16.3.),p.344]

$$\int_0^\infty e^{-(p+s)x-\frac{q}{x}} \frac{\mathrm{d}x}{\sqrt{x}} = \sqrt{\frac{\pi}{p+s}} e^{-2\sqrt{(p+s)q}}$$

we have with $t = p + s$

$$I = \sqrt{\pi} e^p \int_p^\infty e^{-t} e^{-2\sqrt{qt}} \frac{\mathrm{d}t}{\sqrt{t}}.$$

The obvious transformation $u^2 = t$ with $\mathrm{d}t/\sqrt{t} = 2\,\mathrm{d}u$ yields after completing the square in the exponent

$$I = 2\sqrt{\pi}e^{p+q}\int_{\sqrt{p}}^{\infty} e^{-(u+\sqrt{q})^2}\,\mathrm{d}u = 2\sqrt{\pi}e^{p+q}\int_{\sqrt{p}+\sqrt{q}}^{\infty} e^{-v^2}\,\mathrm{d}v = \pi e^{p+q}\mathrm{erfc}(\sqrt{p}+\sqrt{q})$$

where [AS64, (7.1.2),p.297]

$$\mathrm{erfc}(\zeta) = \frac{2}{\sqrt{\pi}}\int_{\zeta}^{\infty} e^{-v^2}\,\mathrm{d}v = 1 - \mathrm{erf}(\zeta) = 1 - \frac{2}{\sqrt{\pi}}\int_{0}^{\zeta} e^{-v^2}\,\mathrm{d}v.$$

Noting $z^2\xi\eta = b(z-b)$ and $\sqrt{p}+\sqrt{q} = z/\sqrt{2}$ we get finally

$$\mathbb{E}_{0,1,z}[H_b(0)] = \underbrace{\sqrt{\frac{\pi}{2}}e^{\frac{z^2}{2}}\mathrm{erfc}\left(\frac{z}{\sqrt{2}}\right)}_{C}\cdot b$$

which proves Lemma 7.12 for $y = 0$ and $h = 1$.

For the bridge pinned at $(0, y)$ and (h, z) we find analogously for $y < b < z$

$$\mathbb{E}_{y,h,z}[H_b(y)] = \underbrace{\sqrt{\frac{h\pi}{2}}e^{\frac{(z-y)^2}{2h}}\mathrm{erfc}\left(\frac{z-y}{\sqrt{2h}}\right)}_{C}\cdot(b-y). \tag{7.22}$$

□

D4 (uniform sample): Another heuristic approach is a uniformly distributed random variable, i.e.

$$H_1(0) \approx t_k + \hat{H}_{uni} \quad \text{with} \quad \hat{H}_{uni} = hu, \quad u \sim \mathcal{U}(0,1), \tag{7.23}$$

completely ignoring the location of the boundary $b = 1$ within y and z.

D5 (Mannella's approach [Man99]): We summarize Mannella's approach for finding an approximation for $H_1(y) \in (t_k, t_{k+1})$. For simplicity we set $y = 0$ and assume $0 < 1 \le z$. The Brownian bridge pinned at $(0, 0)$ and (z, h) can be written (see[BS02, p.64])

$$X^{0,h,z}(t) = \frac{h-t}{h}W\left(\frac{ht}{h-t}\right) + \frac{t}{h}z.$$

Scaling of the Brownian motion $W(\cdot)$ with $c = \sqrt{h/(h-t)}$ gives [RW00a, p.4]

$$X^{0,h,z}(t) = \sqrt{\frac{h-t}{h}} W(t) + \frac{t}{h} z.$$

Let $\alpha \in (0,1)$ (implying $\alpha h \in (0,h)$), set $t = \alpha h$ and scale again, this time with $c = \sqrt{\alpha h}$. This yields

$$X^{0,h,z}(\alpha h) = \sqrt{\alpha(1-\alpha)}\sqrt{h}\xi + \alpha z, \quad \xi \sim \mathcal{N}(0,1).$$

The idea is now to set $X^{0,h,z}(H_b(0)) = b$ and solve for α. With $\omega = \sqrt{h}\xi$ we get the equation

$$\sqrt{\alpha(1-\alpha)}\omega = b - \alpha z \tag{7.24}$$

which has to be solved for α after having generated $\xi \sim \mathcal{N}(0,1)$.

D6 (add the expected value): From the proof of Lemma 7.12, formula (7.22), we recall that

$$\mathbb{E}_{y,h,z}[H_b(y)] = \sqrt{\frac{h\pi}{2}} e^{\frac{(z-y)^2}{2h}} \operatorname{erfc}\left(\frac{z-y}{\sqrt{2h}}\right) \cdot (b-y)$$

which becomes our next approximation for $H_1(0) \in (t_k, t_{k+1}]$:

$$H_1(0) \approx t_k + \hat{H}_E \quad \text{with} \quad \hat{H}_E = \mathbb{E}_{y,h,z}[H_1(y)]. \tag{7.25}$$

D7 (exact in distribution): This is Algorithm 7.2, where in both cases a random number is *sampled with exact distribution*, see Lemma 7.4 and formula (7.9) for $\mathcal{H}_1$ and Lemma 7.9, Corollary 7.11 and Algorithm 7.1 for $\mathcal{H}_2$.

E1-E4: One feature of the exponential-time-stepping-method is the fact that the time step(s) are not known, but rather are a (unknown) random quantity. Only their distribution is known and hence $\mathbb{E}[t_{k+1} - t_k] = 1/\lambda$. A first approach is to set $H_1(0)$ to the expected value of either the beginning (method (**E1**)) or the end (method (**E4**)) of the time step in which a hitting of $b = 1$ occurred or an excursion was detected. Note that each time step comes to an end with an exponential-λ distribution, and further that the first hitting time of level $b = 1$ is independent of δt (the random time step) for the case of $y, z < b = 1$, namely (see also (7.19))

$$\mathbb{P}_y[H_1(y) \le \delta t;\, W(\delta t) \in \mathrm{d}z] = e^{-2\nu(b-(y\vee z))}, \quad \nu = \sqrt{2\lambda}.$$

It is therefore natural to assume that if a hitting of $b = 1$ was detected in the time step $t_k \to t_{k+1}$, that the hitting took place uniformly distributed in $t_k + (0, 1/\lambda)$, yielding

$$H_1(0) \approx t_k + \hat{H}_e \quad \text{with} \quad \hat{H}_e = \frac{u}{\lambda}, \quad u \sim \mathcal{U}(0,1). \tag{7.26}$$

These thoughts led to methods (**E2**) and (**E3**).

7.3.1-c) Results

We first propose a benchmark for our tests. Recall that we wish to numerically approximate the density (7.14). Let us assume that we know how to sample directly from this density. Generating N such samples and running the two χ^2-tests which were described at the beginning of this Section (see also Table 7.1) will certainly give an optimal bound of what one can expect with the specific random number generator used, the sample size (N) and the design of the test.

Inspired by the Box-Muller-method to generate two normally distributed i.i.d. random numbers [BM58][KP92, p.13] we propose and prove the following result:

Lemma 7.13. *Let u, v be uniformly in $(0,1)$ distributed ($u, v \sim \mathcal{U}(0,1)$). Define for $b \neq 0$ the random variables $s, t \in (0, \infty)$ as*

$$s = s(u,v) = \left(\frac{b}{\sqrt{-2\ln u}\sin(2\pi v)} \right)^2, \quad t = t(u,v) = \left(\frac{b}{\sqrt{-2\ln u}\cos(2\pi v)} \right)^2. \tag{7.27}$$

Then, s, t are i.i.d. with density (see (7.11) with $y = 0$)

$$\frac{|b|}{\sqrt{2\pi t^3}} e^{-\frac{b^2}{2t}}, \quad t > 0.$$

Proof. We first note that both s and t in (7.27) are four-to-one in the variable v: for u fixed there are, due to the periodicity and symmetry of $\sin(2\pi v)^{-2}$ (of $\cos(2\pi v)^{-2}$ respectively), four possible $v \in (0,1)$ for a given value of s (of t). On one of these four branches, the inverse functions to (7.27) are given by

$$u(s,t) = \exp\left(-\frac{b^2}{2}\left(\frac{1}{s} + \frac{1}{t} \right) \right) \quad \text{and} \quad v(s,t) = \frac{1}{2\pi}\arctan\left(\sqrt{\frac{t}{s}} \right), \tag{7.28}$$

with, by construction, $u(s(u,v), t(u,v)) = v(s(u,v), t(u,v)) = 1$. We have the partial derivatives

$$\text{and} \quad \begin{array}{ll} \dfrac{\partial u}{\partial s} = \dfrac{b^2}{2s^2} e^{-\frac{b^2}{2}\left(\frac{1}{s}+\frac{1}{t}\right)}, & \dfrac{\partial u}{\partial t} = \dfrac{b^2}{2t^2} e^{-\frac{b^2}{2}\left(\frac{1}{s}+\frac{1}{t}\right)} \\ \dfrac{\partial v}{\partial s} = \dfrac{-1}{4\pi}\dfrac{1}{s+t}\sqrt{\dfrac{t}{s}}, & \dfrac{\partial v}{\partial t} = \dfrac{1}{4\pi}\dfrac{1}{s+t}\sqrt{\dfrac{s}{t}} \end{array}$$

yielding the jacobian J of the map $(s,t) \mapsto (u(s,t), v(s,t))$ as

$$\begin{aligned} J &= \frac{b^2}{2 \cdot 4\pi} \left(\frac{1}{s^2} \frac{1}{(s+t)} \sqrt{\frac{s}{t}} + \frac{1}{t^2} \frac{1}{(s+t)} \sqrt{\frac{t}{s}} \right) e^{-\frac{b^2}{2}\left(\frac{1}{s}+\frac{1}{t}\right)} \\ &= \frac{1}{4} \cdot \frac{b^2}{2\pi} \left(\frac{1}{\sqrt{s^3 t^3}} \right) e^{-\frac{b^2}{2}\left(\frac{1}{s}+\frac{1}{t}\right)} \\ &= \frac{1}{4} \cdot \left(\frac{b}{\sqrt{2\pi s^3}} e^{-\frac{b^2}{2s}} \cdot \frac{b}{\sqrt{2\pi t^3}} e^{-\frac{b^2}{2t}} \right) \end{aligned}$$

To finish, let S and T be two random variables following (7.27) and assume that U and V follow (7.28). Summing over the four branches (i.e. multiplying by a factor four) we find for the joint densities denoted by $p_{S,T}$ and $p_{U,V}$ the relation

$$\begin{aligned} p_{S,T}(s,t) &= 4 p_{U,V}(u(s,t), v(s,t)) \cdot J = 4 \frac{1}{4} \frac{b}{\sqrt{2\pi s^3}} e^{-\frac{b^2}{2s}} \cdot \frac{b}{\sqrt{2\pi t^3}} e^{-\frac{b^2}{2t}} \\ &= \frac{b}{\sqrt{2\pi s^3}} e^{-\frac{b^2}{2s}} \cdot \frac{b}{\sqrt{2\pi t^3}} e^{-\frac{b^2}{2t}} = p_{H_b(0)}(s) \cdot p_{H_b(0)}(t) \end{aligned}$$

which concludes the proof. □

We obtained the results shown in Table 7.3 when the χ^2-tests for the approximation of the peak and the tail were performed with the setup presented in Table 7.1.

Table 7.3: Results obtained when generating various number of samples (N) of random numbers as proposed in Lemma 7.13 and running the two χ^2-tests for the approximation of the peak and of the tail of the density $\mathbb{P}_0[H_1(0) \in \mathrm{d}t]$ as given in (7.14). Shown are the values of both $\overline{\chi}^2$ and $\overline{\chi}^2$ per DOF with $\overline{\chi}^2$ from (7.15).

	peak test (94 DOFs)		tail test (249 DOFs)	
N_P	$\overline{\chi}^2$	$\overline{\chi}^2/94$	$\overline{\chi}^2$	$\overline{\chi}^2/249$
1e4	88.8241	0.9449	272.92	1.0960
1e5	95.2264	1.0130	279.71	1.1233
1e6	101.292	1.0776	223.30	0.8968
1e7	96.6975	1.0287	249.99	1.0040

From the results in Table 7.3 we see that a value of the order of unity can be expected for the variable $\overline{\chi}^2$ per DOF.

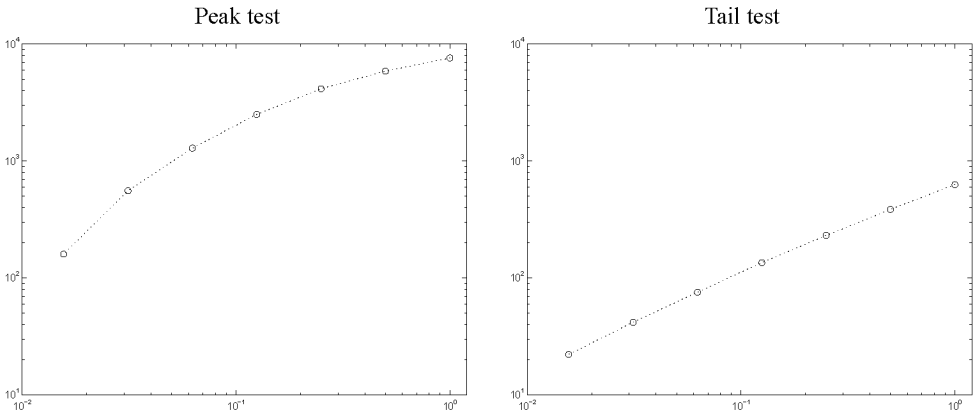

Figure 7.2: Results of χ^2-test for the trivial method to approximate $H_1(0)$: This method stops only if $X_{k+1} \geq 1$ and, in that case, approximates $H_1(0) \approx t_k$, see method (**T**) in Table 7.2. Shown are $\overline{\chi}^2/DOF$ vs. step size h. The left plot shows the quality of the approximation of the peak of the density, whilst the right plot shows the quality of the approximation of the corresponding tail, see Table 7.1 for the details of the problem setup.

In order to thoroughly motivate the need for an exit test we start with the results for the trivial method ((**T**) in Table 7.2), see Figure 7.2.

From Figure 7.2 it is evident that the trivial attempt (method (**T**)) fails completely in approximating both the peak and the tail of density (7.14) even for rather small step sizes h: For h as small as 2^{-6} we still observe a value of the order of 10^2 for the approximation of the peak and a value of the order of 10 for the corresponding approximation of the tail.

In the following Figures (7.3 to 7.6) we compare the remaining methods from Table 7.2 with Algorithm 7.2. The results of this algorithm, (**D7**), are added to all of these plots for purposes of comparison. In all figures, it has a thicker line connecting the plot symbols $\circ$.

- We start in Figure 7.3 with the plots for the methods (**K1**) to (**K4**). In these methods a *killing* test of the form (7.6) is used to check for a possible excursion. They differ in the (heuristic) approximation of $H_1(y) \in [t_k, t_{k+1}]$, see Table 7.2.

- We continue with methods (**D1**) to (**D6**). These methods use the *stopping* test derived in Lemma 7.4. In the case of an excursion, integration is stopped at $t_k + \mathcal{H}_1$, see formula (7.9). For clarity of the individual results, we show the results of methods (**D1**) to (**D6**) in two separate figures, in particular, see Figure 7.4 for (**D1**) to (**D3**) and Figure 7.5 for (**D4**) to (**D6**).

- Finally, we show the results obtained using variants of the exponential time stepping method. Results of the experiments with methods (**E1**) to (**E4**) are in Figure 7.6.

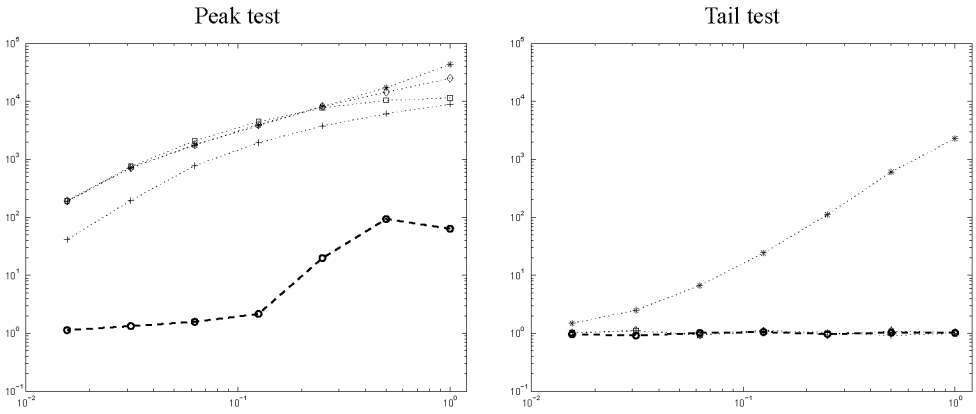

Figure 7.3: Results of χ^2-test for the methods which use a killing test, (7.6), to check for a possible excursion. See lines (**K1**) to (**K4**) in Table 7.2 for further details. The thicker curve represents method (**D7**) which is Algorithm 7.2. Shown are $\overline{\chi}^2/DOF$ vs. step size h. The left plot shows the quality of the approximation of the peak of the density, whilst the right plot shows the quality of the approximation of the corresponding tail, see Table 7.1 for details of the problem setup. $\cdot\cdot\,\square\,\cdot\cdot$ is (**K1**), $\cdot\cdot\,\Diamond\,\cdot\cdot$ is (**K2**), $\cdot\cdot\,*\,\cdot\cdot$ is (**K3**), $\cdot\cdot\,+\,\cdot\cdot$ is (**K4**) and $-\,\circ\,-$ is (**D7**) which is Algorithm 7.2.

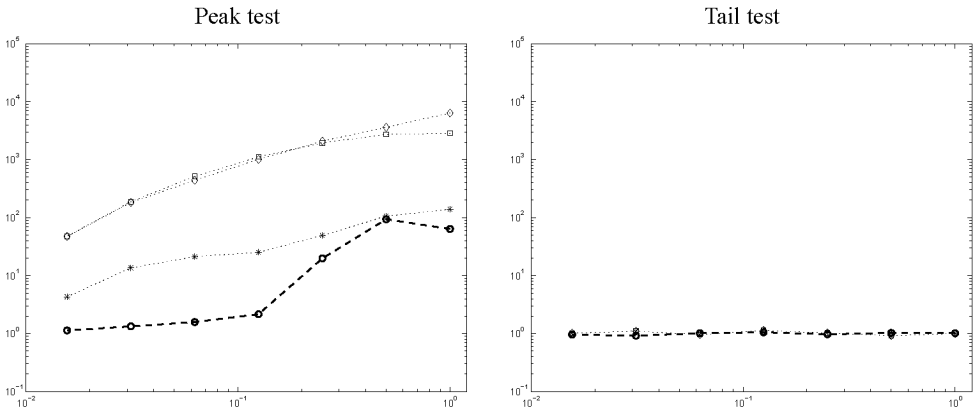

Figure 7.4: Results of χ^2-test for the methods which use the stopping test $h \leq \mathcal{H}\ell_1$, see Lemma 7.4 and in particular (7.9), to check for a possible excursion. See lines (**D1**) to (**D3**) in Table 7.2 for further details. The thicker curve represents method (**D7**) which is Algorithm 7.2. Shown are $\overline{\chi}^2/DOF$ vs. step size h. The left plot shows the quality of the approximation of the peak of the density, whilst the right plot shows the quality of the approximation of the corresponding tail, see Table 7.1 for details of the problem setup. $\cdot\cdot\Box\cdot\cdot$ is (**D1**), $\cdot\cdot\Diamond\cdot\cdot$ is (**D2**), $\cdot\cdot *\cdot\cdot$ is (**D3**) and $-\circ-$ is (**D7**) which is Algorithm 7.2.

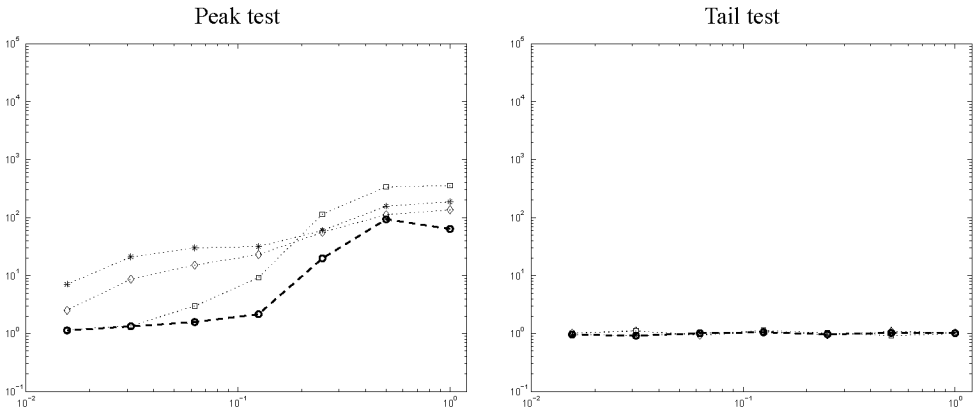

Figure 7.5: Results of χ^2-test for the methods which use the stopping test $h \leq \mathcal{H}\ell_1$, see Lemma 7.4 and in particular (7.9), to check for a possible excursion. See lines (**D4**) to (**D6**) in Table 7.2 for further details. The thicker curve represents method (**D7**) which is Algorithm 7.2. Shown are $\overline{\chi}^2/DOF$ vs. step size h. The left plot shows the quality of the approximation of the peak of the density, whilst the right plot shows the quality of the approximation of the corresponding tail, see Table 7.1 for details of the problem setup. $\cdot\cdot\Box\cdot\cdot$ is (**D4**), $\cdot\cdot\Diamond\cdot\cdot$ is (**D5**), $\cdot\cdot *\cdot\cdot$ is (**D6**) and $-\circ-$ is (**D7**) which is Algorithm 7.2.

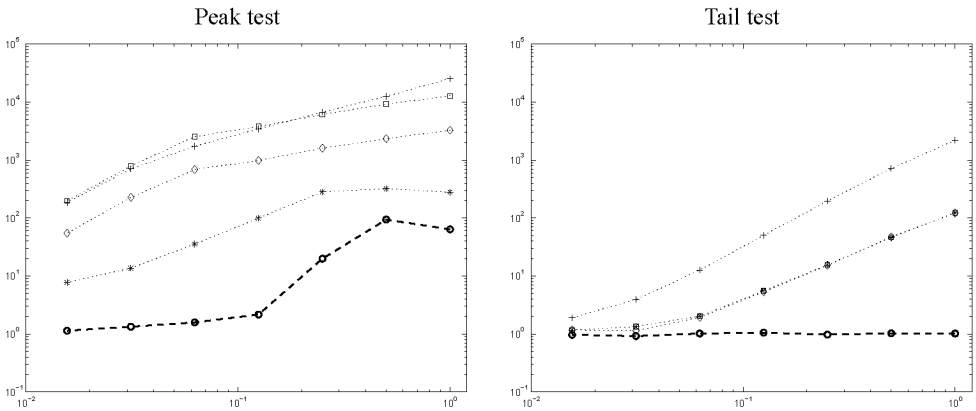

Figure 7.6: Results of χ^2-test for the methods which are based on exponential time stepping, see (7.17),(7.18) and (7.19) for the killing test. See lines (**E1**) to (**E4**) in Table 7.2 for further details. The thicker curve represents method (**D7**) which is Algorithm 7.2. Shown are $\overline{\chi}^2/DOF$ vs. inverse parameter λ for the methods (**E1**) to (**E4**) and vs. step size h for (**D7**). The left plot shows the quality of the approximation of the peak of the density, whilst the right plot shows the quality of the approximation of the corresponding tail, see Table 7.1 for details of the problem setup. $\cdots\square\cdots$ is (**E1**), $\cdots\Diamond\cdots$ is (**E2**), $\cdots*\cdots$ is (**E3**), $\cdots+\cdots$ is (**K4**) and $-\circ-$ is (**D7**) which is Algorithm 7.2.

After having shown the results, let us add a few comments concerning the individual plots:

Figure 7.3, methods (K1) to (K4): From Figure 7.3 we see that the methods (**K1**) to (**K4**) give a poor approximation of the peak of the density (7.14), especially compared to method (**D7**) (the Algorithm 7.2). On the other hand, most of them give a good approximation of the density's tail – only method (**K3**) (which approximates H_1 always by the end of the time step) requires fairly small step sizes to get a good approximation of the entire density.

Figures 7.4 and 7.5, methods (D1) to (D6): The methods in these figures generate the random number $\mathcal{H}_1$, which has the correct distribution to test for an excursion, and if an excursion is detected ($\mathcal{H}_1 \leq h$) integration is stopped at $t_k + \mathcal{H}_1$. It is evident that this approach is superior to the excursion test (7.6). First of all, the peak of density (7.14) is much better approximated, as one can see from the plots on the left hand graph. Only methods (**D1**) and (**D2**) (which stop at the beginning or at the mid point of the time step) show some weakness in the approximation of the peak. Furthermore, the approximation of the tail is now also very good, independent of the chosen strategy when $z = X_{k+1} \geq 1$. We note that (**D7**), Algorithm 7.2, performs best, although the method (**D4**) (adding a random number distributed uniformly in $(0, h)$) also performs very well and for small step sizes, the results of these two methods are virtually indistinguishable.

Figure 7.6, methods (E1) to (E4): The methods which use exponential time stepping show some weakness in the approximation of the density (7.14). First, the approximation of the peak (left plot) is much worse than the approximation obtained by method (**D7**), Algorithm 7.2. We think that this is due to the fact, that the time up to (and including) step k is *approximated* by its expected value, yielding a lower resolution. Second, we see from the right hand plot that these methods have problems in the approximation of the tail, especially for large time steps (corresponding to a small value of the parameter λ). Methods (**E1**) to (**E3**) yielded very similar results for this test, only method (**E4**) (which stops always at the expected value of the end of the time step) performed worse. Nevertheless they show (at least) good convergence properties in the tail test for smaller time steps (increasing λ). Clearly, method (**D7**), Algorithm 7.2 yields much better results.

To summarize, the method (**D7**) which was proposed in Algorithm 7.2 gives the most accurate results for the computation of an approximation to the density (7.14). Not only does it have no problems in the approximation of the tail, but it also shows the best results for the approximation of the peak. This good approximation is already evident for quite large time

steps h. Note that the density (7.14) has its peak at $t = \frac{1}{3}$ and the size of the bins for the peak test is 0.01 (see Table 7.1). It is therefore clear that for $h \gg 0$ a χ^2-value per DOF of the order of one cannot be expected. But method (**D7**) already gives very good results for step sizes as large as $h \leq \frac{1}{8}$.

7.3.2 Tests on the exit time problem

Motivated by the fact that that good results were obtained for an excursion test in the case that $D = \{x \in \mathbb{R} \,:\, x < b\}$ (in the previous section we had $b = 1$), we now try to extend these results to a bounded interval of the form $(a, b) \subset \mathbb{R}$ with $-\infty < a < b < \infty$. The extension of the methods summarized in Table 7.2 to this situation works in exactly the same way as for killed diffusions [Gob00, Gob01]: The domain is locally approximated as a half-space, i.e. an excursion test is applied only to the closest boundary: Instead of applying (7.7), the test (7.6) is applied only to the closest boundary, see Definition 7.2. When referring to the methods of Table 7.2, and in particular to Algorithm 7.2, we always imply these simple extensions. Of course, the same holds for Algorithm 7.1.

We start with results obtained when setting $g \equiv 1$ and $\psi \equiv 0$ in the problem (5.2) (leading to the system (6.1) in the new notation). We have $u(x) = \mathbb{E}[\tau_D^x]$ and $u(0) = b - a$ for $a < 0 < b$ (see for example [RW00a, Chapter I] for some basic properties) and we therefore call this the *exit problem*, as in Section 5.2.

Our first test addresses the question if it is sufficent to test only for an excursion across the *closest* boundary (see Definition 7.2).

7.3.2-a) Testing only for the closest boundary

We show results when method (**K1**) is applied to the exit problem. Recall that method (**K1**) stops integration at t_k if either $X_{k+1} \notin D$ or an excursion was detected during the step $t_k \to t_{k+1}$. Hence, this is the simplest possible extension from killed to stopped diffusions. – Initially we are only interested in the question if weak order one can be preserved by testing only for the closest boundary. We shall later see (in Section 7.3.2-b)) that more sophisticated methods (as for example extensions of (**D1**) and in particular (**D7**)) yield even better results.

To see whether the application of the test (7.6) instead of (7.7) is admissible, we set $b = 1$ and took three different values for the lower boundary, namely $a = -7, -3, -1$ (implying $|D| = b - a = 8, 4, 2$). In the plot of the absolute error $|u(0) - \overline{u}(0)|$ we add a 66%-confidence interval (the numerical standard deviation $S_N[\overline{u}]$, see (5.3')).

Results in Figure 7.7 are shown for a sample size of $N = 10^6$ in the left hand plot and for $N = 4 \cdot 10^6$ in the right hand plot. The maximal step size is chosen to be $h_{max} = (b - a)/4$ and we show results for h_{max} and seven smaller step sizes.

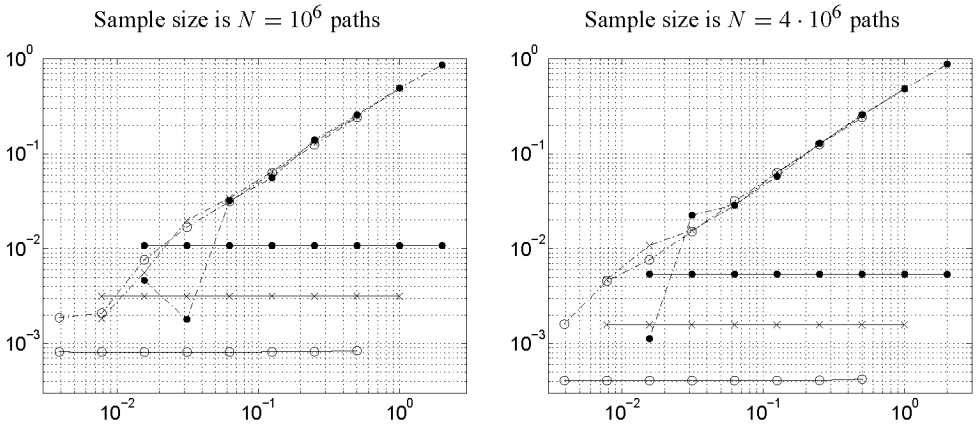

Figure 7.7: Testing only for the closest boundary preserves weak order one: Method (**K1**) from Table 7.2 was applied to the exit problem in $D = (a, b)$ and an excursion test was performed only for the *closest* boundary, see Definition 7.2. The plots show the error $|\overline{u}(0) - u(0)|$ and the numerical standard deviation vs. step size h for $a = -7, -3, -1$ and $b = 1$ using two different sample sizes. $\circ$ is for $a = -1$, $\times$ for $a = -3$ and $\bullet$ for $a = -7$.

From Figure 7.7 it can be seen, that the weak order of convergence is $\mathcal{O}(h)$ independent of the choice of a. Note that decreasing a (and hence increasing the size of the domain) increases the numerical standard deviation $S_N[\overline{u}]$. This occurs because on average more steps are needed until a path is stopped (the expected exit time increases).

7.3.2-b) Results with the extensions of methods (T), (K1), (D1), (D4), (D6) and (D7)

We show results of experiments performed with methods (**T**), (**K1**), (**D1**), (**D4**), (**D6**) and (**D7**) when applied to the exit problem in $D = (-1, 1)$. We recall the basic features of these methods presented earlier:

(T): trivial stopping, stop at t_k if $X_{k+1} \notin D$.

(K1): excursion test (7.6), stop always at t_k.

(D1): generate $\mathcal{H}_1$ according to (7.9) for the excursion test, stop at $t_k + \mathcal{H}_1$ if $\mathcal{H}_1 \leq h$ and at t_k if $X_{k+1} \notin D$.

(D4): like (**D1**) but stop at $t_k + \hat{H}_{uni}$ with $\hat{H}_{uni}$ distributed uniformly in $(0, h)$ if $X_{k+1} \notin D$.

(D6): like (**D1**) but stop at $t_k + \hat{H}_E$ (if $X_{k+1} \notin D$) where $\hat{H}_E$ is the expected value, see (7.22).

(D7): like (**D1**) but stop at $t_k + \mathcal{H}_2$ (if $X_{k+1} \notin D$) with $\mathcal{H}_2$ as in Algorithm 7.1 (inverse Gaussian distribution), see Algorithm 7.2.

For clarity of the individual results, we split them into three figures. In the first one, Figure 7.8, we compare methods (**T**) and (**K1**). In the second one, Figure 7.9, we compare methods (**K1**), (**D1**) and (**D7**). Finally, in Figure 7.10 we compare methods (**D4**) and (**D6**) with (**D7**).

We always show the approximate solution $\overline{u}(0)$ on the left and the resulting error $|\overline{u}(0) - u(0)|$ on the right. In the plot of the solution, we add in Figure 7.8 the numerical standard deviation as a confidence interval as defined in (5.3'). For each of the left hand plots the exact solution $u(0) = 1$ is indicated by a solid line.

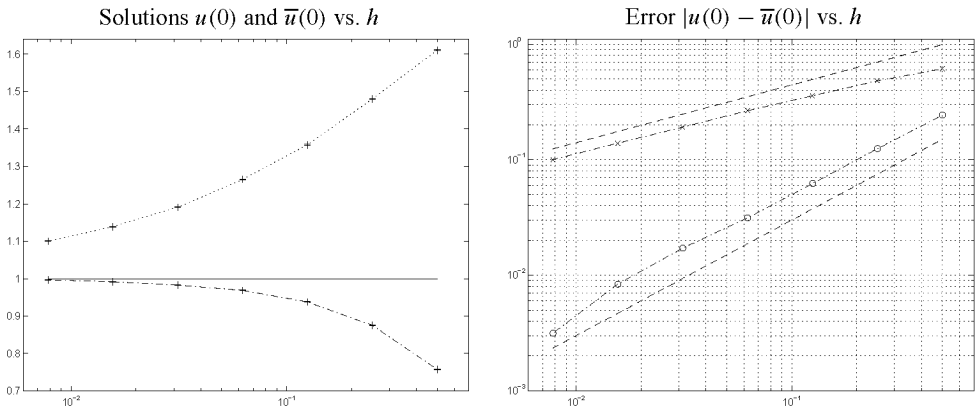

Figure 7.8: Comparison of methods (**T**) and (**K1**) when applied to the exit problem in $D = (-1, 1)$, see Table 7.2. On the left, resulting approximate solutions are shown together with a confidence interval versus step size h. The points connected with a dotted line ($\cdots\cdots$) correspond to method (**T**), whilst the dash-dotted ($-\cdot-\cdot-$) correspond to method (**K1**). The exact solution $u(0) = 1$ is drawn with a solid line (——). On the right the resulting errors are shown versus step size h. There, $-\cdot\times\cdot-$ is method (**T**) and $-\cdot\circ\cdot-$ is method (**K1**). Two dashed lines with slope one half and with slope one respectively are added ($--$).

From Figure 7.8 we see again that method (**T**) stops integration much too late on average, as the approximate solution for the exit problem is found to be much too large, $\overline{u} \gg u = 1$. Therefore, convergence is very slow and the $\mathcal{O}(\sqrt{h})$-behaviour of the error is evident. On the other hand we see that method (**K1**) stops integration too early on average, as the numerical

solution approaches the exact one, $u(0) = 1$, from below. From the plot on the right we see again that the $\mathcal{O}(h)$-convergence of the Euler scheme can also be recovered in the presence of two boundaries when the test (7.6) (and (7.6′) respectively) is only applied to the closest boundary.

We next show results from simulations with methods (**K1**), (**D1**) and (**D7**), see Figure 7.9. For clarity, no confidence intervals are added. In theses cases, the numerical standard deviation, see (5.3'), was of the order of 0.008 for all of the methods and step sizes considered. Note the different scaling of the abscissae compared to the corresponding plots in Figure 7.8.

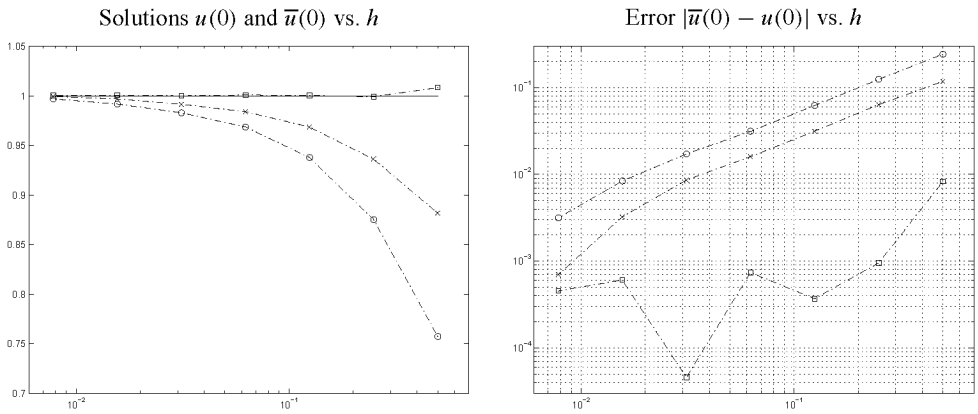

Figure 7.9: Comparison of methods (**K1**), (**D1**) and (**D7**) when applied to the exit problem in $D = (-1, 1)$, see Table 7.2. On the left, resulting approximate solutions are shown versus step size h. Confidence intervals are smaller than the plot symbols. The exact solution $u(0) = 1$ is drawn with a solid line (——). On the right, the resulting errors are shown versus step size h. In both plots $- \cdot \circ \cdot -$ is method (**K1**), $- \cdot \times \cdot -$ is method (**D1**) and $- \cdot \square \cdot -$ is method (**D7**).

From Figure 7.9 we see that results can be improved significantly when integration is not simply stopped at t_k. First, the test $h \leq \mathcal{H}_1$ with $\mathcal{H}_1$ from (7.9) (Lemma 7.4) and stopping at $t_k + \mathcal{H}_1$ in case of an excursion is clearly better compared to the tests (7.6) and (7.6′): Method (**D1**) leads to an error which is smaller by approximately a factor of two compared to method (**K1**). We note also, that despite its better performance compared to method (**K1**), this test still stops the integration too early on average, as the exact solution, $u(0) = 1$ is still an upper bound for the numerical approximation. This is due to the fact, that in the case $X_{k+1} \notin D$, method (**D1**) always stops at t_k (as does method (**K1**)). Method (**D7**) resolves this problem by adding the random variable $\mathcal{H}_2$ which follows an inverse Gaussian distribution (see Definition 7.8, Lemma 7.9, Corollary 7.11 and in particular Algorithm 7.1). We see that the numerical solution is now very accurate and fluctuates around the true solution $u(0) = 1$. From the plot of the error we additionally observe that the errors are of the order of one numerical standard deviation, as defined in equation (5.3'), which is of the order of 0.008 for the chosen sample size.

To complete the section concerned with numerical tests for the one dimensional exit problem, we show results that compare methods (**D4**), (**D6**) and (**D7**), see Figure 7.10. Note again the different scaling of the abscissae in these plots compared to the corresponding plots in Figures 7.8 and 7.9.

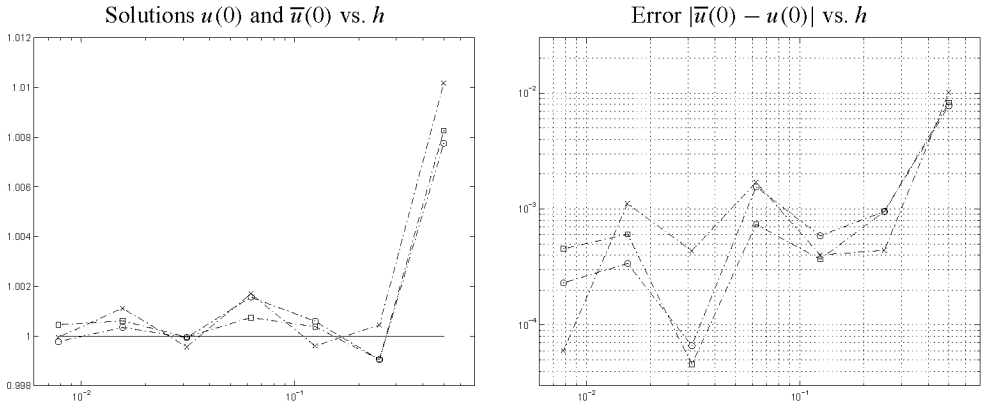

Figure 7.10: Comparison of methods (**D4**), (**D6**) and (**D7**) when applied to the exit problem in $D = (-1, 1)$, see Table 7.2. On the left, resulting approximate solutions are shown versus step size h. The exact solution $u(0) = 1$ is drawn with a solid line (——). On the right the resulting errors are shown versus step size h. In both plots $- \cdot \circ \cdot -$ is method (**D4**), $- \cdot \times \cdot -$ is method (**D6**) and $- \cdot \square \cdot -$ is method (**D7**).

From Figure 7.10 it is apparent that the results of all three methods ((**D4**), (**D6**) and (**D7**)), show very good approximations and the corresponding errors are very small. In fact, they are already reduced to the order of the statistical error when the second largest step size $h = 0.25$ is used.

We later show another comparison of these three methods in which a larger sample size is employed ($N = 1.6 \cdot 10^7$). See Section 8.1.2-a) for a two dimensional exit problem from $D = (-1, 1) \times (-1, 1)$.

7.4 Extension to general diffusions

In this section we extend the ideas presented so far to more general BVPs which lead to general diffusions. Again, we limit ourselves to a one dimensional situation.

7.4.1 The general one dimensional Dirichlet problem and connected diffusions: extension of the presented methods

Let $D = (\alpha, \beta)$ and consider the one dimensional version of (2.4),

$$\left[\frac{1}{2}a(x)\frac{\mathrm{d}^2}{\mathrm{d}x^2} + b(x)\frac{\mathrm{d}}{\mathrm{d}x}\right]u(x) + g(x) = 0, \quad x \in D, \quad u(x) = \psi(x), \quad x \in \partial D, \quad (7.29)$$

where $a(x) > 0$. The probabilistic solution is given by (2.6) (with $c(x) \equiv 0$) where X solves (7.1) with $\sigma^2(x) = a(x)$ (which recalls (2.3) in one dimension). Numerically, we get the Euler approximations (7.3) and $f_{k+1} = f_k + g(X_k)h$ with $f_0 = 0$.

For killed diffusions, we might apply the exit tests (7.6) and (7.6′), whereas for stopped diffusions we generate $\mathscr{H}_1$ according to (7.9') and stop if $\mathscr{H}_1 \leq h$. In that case, we add a last Euler update of length $\mathscr{H}_1$,

$$f(\tau) = \int_0^\tau g(X(s))\,\mathrm{d}s \overset{\text{law}}{=} \int_0^{t_k} g(X(s))\,\mathrm{d}s + \int_{t_k}^{\mathscr{H}_1} g(X(s))\,\mathrm{d}s \overset{(7.3)}{\approx} f_k + \mathscr{H}_1 g(X_k).$$

In this general case, the formula for $\mathscr{H}_1$ follows by inverting the corresponding distribution of the bridge (compare with (7.8))

$$\mathbb{P}_y\left[H_b(y) \leq t; X(t) \in \mathrm{d}z\right] = \mathbb{P}_{y,h,z}[H_b(y) \leq t] = \exp\left(-\frac{2(b-y)(b-z)}{t\sigma^2(y)}\right).$$

Out of these two formulae we find methods (**K1'**) and (**D1'**) which extend methods (**K1**) and (**D1**) respectively in a natural way (see Table 7.2 with obvious modifications).

7.4.2 Numerical experiment

We show results for the BVP (7.29) with $a(x) = \sigma^2(x)$ and

$$\sigma(x) = 2 + \sin(x), \quad b(x) = -\cos(x)\left(2 + \frac{\sin(x)}{2}\right) \quad \text{and} \quad g(x) = 2\cos(x).$$

We take $D = (-\pi/2, \pi/2)$ and set $\psi(x) = 0$. Then we have the solution $u(x) = \cos(x)$ which is evaluated numerically at $x = 0$ (so $u(0) = 1$).

In Figure 7.11 we show results for the naive approach (**T**) compared with methods (**K1'**) and (**D1'**). We have chosen sample sizes $N = 10^6$ and $N = 4 \cdot 10^6$. We show the approximate solutions and absolute errors. Confidence intervals S_N from (5.3') are much smaller than the symbols used in the plot.

From Figure 7.11 we see, that the procedure works also for BVPs with general elliptic operators, as in (7.29). Both methods (**K1'**) and (**D1'**) converge with a weak order of at least one, whereas the simple approach, method (**T**), converges only with order $\mathcal{O}(\sqrt{h})$. Further, method (**D1'**) improves the constant in the convergence order with no additional computational cost, compared to method (**K1'**).

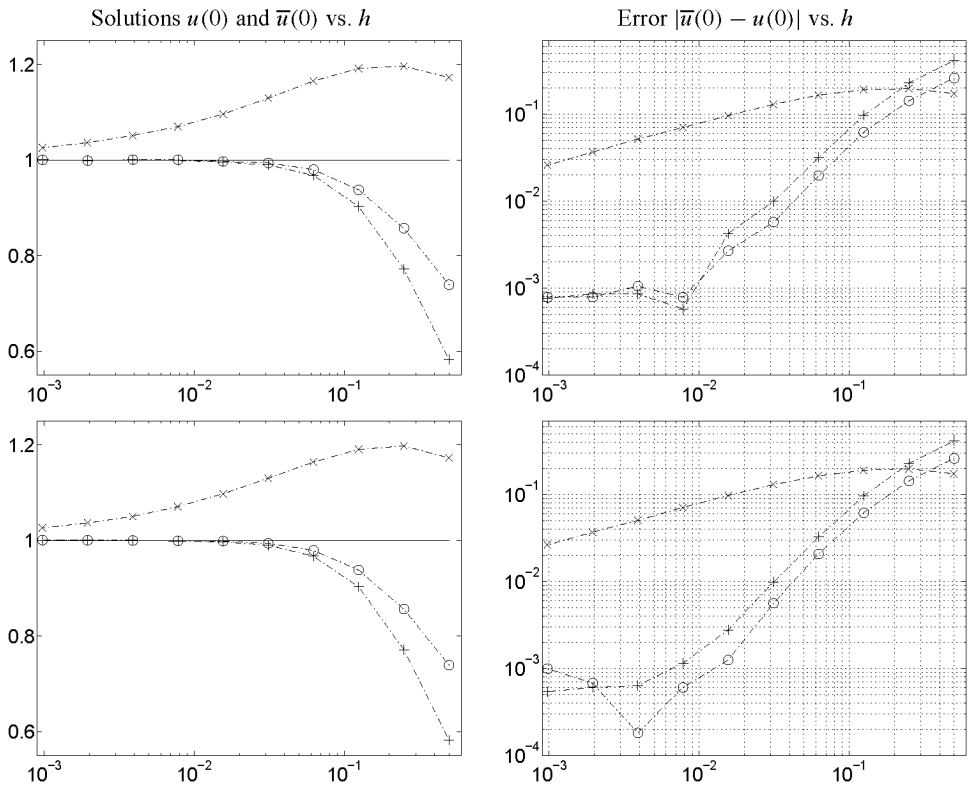

Figure 7.11: An exit test leads to a convergence rate of weak order $\mathcal{O}(h)$ also in the case of general elliptic operators with connected general diffusions $\mathrm{d}X = b(X)\,\mathrm{d}t + \sigma(X)\,\mathrm{d}W$. By generating $\mathcal{H}_1$ according to (7.9') and stopping integration at $t_k + \mathcal{H}_1$ in case of an excursion the result is improved further with no additional computational cost. Results are shown for two different sample sizes, $N = 10^6$ on top and $N = 4 \cdot 10^6$ at the bottom. In all four plots $- \cdot \times \cdot -$ is method (**T**), $- \cdot + \cdot -$ is method (**K1'**) and $- \cdot \circ \cdot -$ is (**D1'**), see Table 7.2.

Chapter 8

Higher dimensions

We now wish to expand the ideas presented so far to higher dimensions. Our approach is based on the heuristic argument that if the step size h is much smaller than the curvature of the boundary of the domain D (i.e. the boundary is locally flat), the multi dimensional situation resembles the one dimensional situation. We therefore project the multi dimensional case in such a manner that the formulae from the one dimensional situation can be applied.

We focus on the extension of Algorithm 7.2 (on p. 77) which gave most accurate results for our one dimensional computations. This leads to a similar approach for sampling the exit time from D as presented in method (**D7**) in Table 7.2.

When we finally try to apply these ideas to solve BVPs of the form (2.4) via (2.6) we notice, that additionally for dimensions $n \geq 2$, an approximation of the exit point $X(\tau)$ is needed to evaluate the boundary condition in the case of Dirichlet problems. This problem did not arise in one dimension, as in the case of a bounded interval $D = (a, b)$ we evaluated the boundary condition at the point through which we made an excursion test or at the natural boundary point when $z = X_{k+1} \notin D$. Our approach to this problem in general n-space is addressed at a later stage. We start by showing how to approximate the exit time, i.e. how to simulate stopped diffusions in bounded domains in n-space. Note that we will then be able to consider all BVPs of the form (2.4) with constant boundary condition ψ.

8.1 Exit time

The basis for simulating a stopped diffusion in a bounded domain in n-space using the Euler scheme and local approximation of D as a half space is formed from the following results.

8.1.1 Some known results for half spaces

Let $X(t)$ be a Brownian motion with values in $\mathbb{R}^n$ starting at t_k at $y = (y^1, \dots, y^n)^T$, i.e. $X(t)$ is solution to $\mathrm{d}X(t) = \mathrm{d}W(t)$ for $t \geq t_k$ with $X(t_k) = y$. For $h > 0$ we have the single step $t_{k+1} = t_k + h$. Define further for $b \in \mathbb{R}$ with $b > y^1$ the first hitting time of level b of the first component of process X as $H_b = \inf_{t>0}\{X^1(t) \geq b\}$. Let $X(t_{k+1}) = z = (z^1, \dots, z^n)^T$. Then it holds for $y^1, z^1 < b$ that

$$\mathbb{P}\left[H_b \leq h; X(t_k) = y, \;\; X(t_{k+1}) = z\right] = e^{-\frac{2}{h}(b-y^1)(b-z^1)}. \tag{8.1}$$

The proof of this formula for $y^i = 0$, $i = 1, \dots, n$ can be found in [LS89, Lemma 2]. The extension to $y^i \neq 0$ is straightforward by a shift of the origin and is therefore skipped here. In [Gob01, formula (11)] this formula is used for the simulation of killed diffusions in a n-dimensional half space.

For simplicity let $t_k = 0$ and $y^i = 0$ for $i = 1, \dots, n$. Then for $0 < t < 1$ the density of H_b is given by [LS89, Lemma 3]

$$\mathbb{P}\left[H_b \in \mathrm{d}t; X^i(0) = 0 \;\; (i = 1, \dots, n), \;\; X(1) = z\right] =$$
$$\frac{b}{\sqrt{t^3(1-t)}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(b\sqrt{\frac{1-t}{t}} - (z^1 - b)\sqrt{\frac{t}{1-t}}\right)^2\right) \mathrm{d}t. \tag{8.2}$$

With the notation of Section 7.2 (Brownian bridge and associated measure) we rewrite (8.1) as

$$\mathbb{P}_{y,h,z}[H_b \leq h] = e^{-\frac{2}{h}(b-y^1)(b-z^1)}, \quad y^1 < b \quad \text{and} \quad z^1 < b. \tag{8.1'}$$

Similarly, we find for (8.2) after some manipulations in the exponent

$$\mathbb{P}_{0,1,z}[H_b \in \mathrm{d}t] = \frac{b}{\sqrt{2\pi t^3(1-t)}} \exp\left(-\frac{1}{2}\frac{(b - z^1 t)^2}{t(1-t)}\right) \mathbf{1}_{\{0<t<1\}} \mathrm{d}t. \tag{8.2'}$$

Some scaling and shifting arguments lead to the formula for the general case (remember that b is a barrier in the first coordinate)

$$\mathbb{P}_{y,h,z}[H_b \in \mathrm{d}t] =$$
$$\frac{b - y^1}{\sqrt{2\pi h^3}} \frac{1}{\sqrt{\left(1 - \frac{t}{h}\right)\left(\frac{t}{h}\right)^3}} \exp\left(-\frac{1}{2h}\frac{\left((b - y^1) - \frac{t}{h}(z^1 - y^1)\right)^2}{\frac{t}{h}\left(1 - \frac{t}{h}\right)}\right) \mathbf{1}_{\{0<t<h\}} \mathrm{d}t. \tag{8.3}$$

We note the striking similarity to the formulae in the one dimensional situation: Compare (8.1),(8.1') with (7.8) and (8.2') with (7.13) or (8.3) with (7.10).

The following observation is immediate:

Remark 8.1. In the case of a boundary connected to only one of the components of an n-dimensional Brownian motion (and assuming that the boundary is locally flat, i.e. (locally) a hyperplane), the fact that these individual components are *n independent* one dimensional Brownian motions leads to formulae as simple as in the one dimensional setting. Furthermore, only the components connected to the boundary show up in the equations.

We have chosen the first component as an example. This situation could equally well be described by the following setup: Let $D = \{x \in \mathbb{R}^n : x^1 < b\}$ or let $b = (b, 0, \ldots, 0)$ and let $D = \{x \in \mathbb{R}^n : \langle e^1, b - x\rangle > 0\}$ where $e^1 = (1, 0, \ldots, 0)$. With $\tau = \inf_{t>0}\{X(t) \notin D\}$ we immediately have that $\tau = H_b$, where (we recall that) $H_b = \inf_{t>0}\{X^1(t) \geq b\}$.

By simple transformations like shifts and scalings we can map every $(n-1)$-dimensional hyper-plane in n-space to the plane given by $T_b = \{x \in \mathbb{R}^n : x^1 = b\}$. Assume that the boundary of a general domain $D \subset \mathbb{R}^n$ is locally flat, such that its (local) approximation by its tangent hyper-plane is sufficiently good, see [Gob01, Section 1]. With this assumption we have the main ingredients required for the extension of Algorithm 7.2 to compute the first exit time from D in the high dimensional setting. We collect them in Algorithm 8.1

Remark 8.2. After having stated the basic ideas to simulate a stopped diffusion in n-space in Algorithm 8.1 we add a few comments.

1. Clearly, instead of mapping $X_b \mapsto (b, 0, \ldots, 0)^T$, $y \mapsto \tilde{y}$ and $z \mapsto \tilde{z}$ one could also transform formulae (8.1') and (8.3) and the corresponding algorithms.

2. The map to the situation where the boundary is given by $(b, 0, \ldots, 0)^T$ includes a rotation of vectors y and z in n-space. Computationally, this can be done efficiently using $n - 1$ Givens rotations, details are given below in Section 8.1.2-b) and in Algorithm 8.2. This approach allows the inverse transformation to be computed efficiently too – something that we will need when additionally a point on ∂D has to be sampled as an approximation for the first exit point $X(\tau)$, see Algorithm 8.3 and Section 8.2.

3. The question of finding $X_b \in \partial D$ by minimizing the distances to y and z will be addressed later too. Depending on the specific domain D we will give a different interpretation of *closest*. Clearly, this is connected to the specific choice of the norm in which we measure the distance. When describing our different approaches, we will see, that efficiency considerations might lead to different choices for the two possible situations, $z \in D$ ($\tilde{z}^1 < b$) and $z \notin D$ ($\tilde{z}^1 \geq b$), for a given domain D.

We have shown the extension only of method (**D7**) to the higher dimensional situation (Algorithm 8.1). The other methods collected in Table 7.2 can be extended similarly and this procedure is straightforward. Therefore, we do not discuss these extensions in any detail, but note, that if we refer to any method from Table 7.2 in what follows we imply this extension.

Algorithm 8.1 One integration step with step size $h > 0$ to compute an approximation to $\tau(x) = \inf_{t>0}\{X(t) \notin D\}$ $(X(0) = x \in D \subset \mathbb{R}^n)$.

Assume $y = X_k \in D$ and that the diffusion has started at $X_0 = x \in D$. We describe the step $t_k \to t_{k+1}$.

1. Make a preliminary step: $z^i = y^i + \sqrt{h}\xi$, $\xi \sim \mathcal{N}(0, 1)$ for $i = 1, \ldots, n$.
2. Find a point $X_b \in \partial D$ closest (which will be subsequentially defined precisely for a particular problem) to y and z.
3. Map the situation s.t. $X_b \mapsto (b, 0, \ldots, 0)^T$. Denote the transformed variables by $\tilde{y}$ and $\tilde{z}$: $y \mapsto \tilde{y}$ and $z \mapsto \tilde{z}$.
4. **if** $\tilde{z}^1 < b$

   Generate $\mathcal{H}_1 = -2(b - \tilde{y}^1)(b - \tilde{z}^1)/\log u$ where $u \sim \mathcal{U}(0, 1)$ is uniformly in $(0, 1)$ distributed (compare with Lemma 7.4 in one dimension).

   if $\mathcal{H}_1 \leq h$

   STOP at $\tau(x) = t_k + \mathcal{H}_1$.

   else

   $X_{k+1} = z$ and continue integration.

else

Generate $\mathcal{H}_2$ following an inverse Gaussian distribution (see Definition 7.8) according to Algorithm 7.1 with the scalar input variables b, $\tilde{y}^1$ and $\tilde{z}^1$.
STOP at $\tau(x) = t_k + \mathcal{H}_2$.

As we are not going to prove anything in this section, we provide a series of numerical experiments which confirm the effectiveness of our approach, in particular the claimed (weak) $\mathcal{O}(h)$-convergence of the error *independent of the dimension n* (when neglecting the statistical error which is of the order of the reciprocal of the square root of the sample size).

8.1.2 Numerical experiments in different domains

A first additional error source that was not present in the one dimensional case is obviously due to the (local) approximation of ∂D by its tangent hyper-plane at the point X_b. The flatter ∂D, the smaller this error will be. We therefore start with a simple two dimensional domain without local curvature, i.e. a square.

8.1.2-a) Two dimensional square

We set $D = (-1, 1) \times (-1, 1)$ and consider again the exit problem, i.e. Poisson's equation, (5.2) with $g \equiv 1$ and $\psi \equiv 0$ or $\triangle u(x^1, x^2) = -2$.

Exact solution: We first compute the exact solution. The orthonormal eigenfunctions $u_{k,l}$ and corresponding eigenvalues $\lambda_{k,l}$ of the $\triangle$-operator in D are for $k, l = 1, 2, \ldots$ given by (here $x = (x^1, x^2)$)

$$u_{k,l}(x) = \sin\left(\frac{k\pi}{2}(1 + x^1)\right) \sin\left(\frac{l\pi}{2}(1 + x^2)\right) \tag{8.4a}$$

$$\text{and} \quad \lambda_{k,l} = -\frac{\pi^2}{4}\left(k^2 + l^2\right). \tag{8.4b}$$

We make the ansatz $-2 = \sum_{k,l=1}^{\infty} d_{k,l} u_{k,l}(x)$ and use the orthonormality of the eigenfunctions $u_{k,l}(x)$

$$-\int_D 2u_{m,n}(x)\,\mathrm{d}x = \sum_{m,n=1}^{\infty} \int_D d_{k,l} u_{k,l}(x) u_{m,n}(x)\,\mathrm{d}x = d_{m,n},$$

yielding for $m, n = 1, 2, \ldots$

$$d_{m,n} = \begin{cases} -\frac{32}{\pi^2}\frac{1}{mn}, & \text{if } m \text{ and } n \text{ are odd,} \\ 0, & \text{otherwise.} \end{cases} \tag{8.5}$$

The ansatz $u(x) = \sum_{k,l} c_{k,l} u_{k,l}(x)$ and the corresponding expansion

$$\triangle u(x) = \sum_{k,l} c_{k,l} \lambda_{k,l} u_{k,l} = \sum_{k,l} d_{k,l} u_{k,l}$$

leads to $c_{k,l} = d_{k,l}/\lambda_{k,l}$. Inserting (8.4) and (8.5) finally gives

$$u(x) = \frac{128}{\pi^4} \sum_{k,l \text{ odd}} \frac{\sin\left(\frac{k\pi}{2}(1+x^1)\right)\sin\left(\frac{l\pi}{2}(1+x^2)\right)}{kl(k^2+l^2)}$$

and in particular

$$u(0) = \frac{128}{\pi^4} \sum_{k,l=0}^{\infty} \frac{(-1)^{k+l}}{(2k+1)(2l+1)\left((2k+1)^2+(2l+1)^2\right)} \approx 0.5893708\ldots. \tag{8.6}$$

Technicalities: We define the closest boundary for this test problem (see item 2. in Algorithm 8.1). Recall that $y = X_k \in D$ and $z = X_{k+1}$.

Definition 8.3. We number for $D = (-1,1) \times (-1,1)$ with boundary $\partial D = \mathbb{R}^2 \setminus D$ the four boundary segments clockwise starting at the boundary $x^2 = 1$, in detail (see Figure 8.1): $B_1 = \{x^2 = 1\}$, $B_2 = \{x^1 = 1\}$, $B_3 = \{x^2 = -1\}$ and $B_4 = \{x^1 = -1\}$. We always have that $y \in D$ but have to separate the three possible cases:

a) $z \in D$: Let $s = y + z$ and let ρ_i denote the sum of the distances of y and z to B_i, $i = 1, \ldots, 4$: $\rho_1 = 2 - s^2$, $\rho_2 = 2 - s^1$, $\rho_3 = 2 + s^2$ and $\rho_4 = 2 + s^1$. Then, the index i of the *closest boundary* is the one with the minimal corresponding distance ρ_i. If this choice is not unique, we simply choose out of the possible candidates the one with minimal index.

b) $z \in \partial D$: We define the *closest boundary* to be the boundary segment which contains z. If this choice is not unique (because z coincides with one of the corners of D) we pick the boundary segment with the minimal index.

c) $z \notin \overline{D}$: Let $\vec{y}$ and $\vec{z}$ be the position vectors of y and z respectively. Define for $0 \le \alpha \le 1$ the vector $\vec{v}$ as $\vec{v} = \vec{y} + \alpha(\vec{z} - \vec{y})$ ($\vec{v}$ joins $\vec{y}$ and $\vec{z}$). We define the *closest boundary* to be the boundary segment B_i which is crossed by $\vec{v}$ ($i = 1, \ldots, 4$). Again, if this choice is not unique (if y and z lie on the same extended diagonal of D for example) we pick the boundary segment with the minimal index.

Remark 8.4. As can be seen by the definition of the closest boundary (Definition 8.3), we do not take into account more than one boundary segment for an excursion test in each step, even if two boundaries are equally far away. Furthermore, if D is clearly left ($z \notin D$), we assume that $(X(t))_t$ crossed the boundary ∂D at the same point as a linear interpolated version of the discrete path $(X_k)_k$ would. In doing so, we introduce an additional error. We neglect this error and do not try to correct for it, see [Bal95] for a discussion and possible

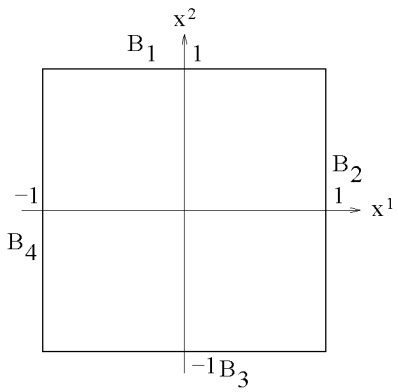

Figure 8.1: Numbering of boundary segments in Definition 8.3.

solutions to this problem. As we will see from our results which follow in the next paragraph, for sufficiently small h, these errors can indeed be neglected: asymptotically we get (at least) the desired $\mathcal{O}(h)$-behaviour of the error.

Results: We compare straightforward extensions of some of the methods in Table 7.2. In more detail, we first show results of simulations with methods (**T**) (trivial stopping), (**K1**) (test (7.6) for excursion), (**D1**) (generate $\mathcal{H}_1$ according to (7.9) and stop at $t_k + \mathcal{H}_1$ if $\mathcal{H}_1 \leq h$) and (**D7**) (same as (**D1**), but stop at $t_k + \mathcal{H}_2$ if $z \notin D$ where $\mathcal{H}_2$ is generated as in Algorithm 7.1). The results obtained when evaluating $u(0)$ (see (8.6)) for a sample size of $N = 4 \cdot 10^6$ are shown in Figure 8.2. Confidence intervals S_N are smaller than plot symbols (see (5.3')).

From Figure 8.2 we see that the main observations noted in the one dimensional situation ($D = (-1, 1)$) carry over to the two dimensional domain $D = (-1, 1) \times (-1, 1)$. As in Figures 7.8 and 7.9 we note in Figure 8.2 the following items:

(T): Method (**T**) is very inaccurate and therefore clearly not satisfying. First, this method stops integration much too late on average, as the exact solution is approached from above (see the left plot in Figure 8.2). Second, its convergence rate for sufficiently small h is again only of order $\sqrt{h}$.

(K1) and (D1): Both methods (**K1**) and (**D1**) restore the asymptotical $\mathcal{O}(h)$-convergence for sufficiently small step sizes h. Again, both methods stop integration too early on average as the exact solution $u(0)$ is approached from below. Furthermore, method (**D1**) improves the constant in the asymptotics of the error at the same computational

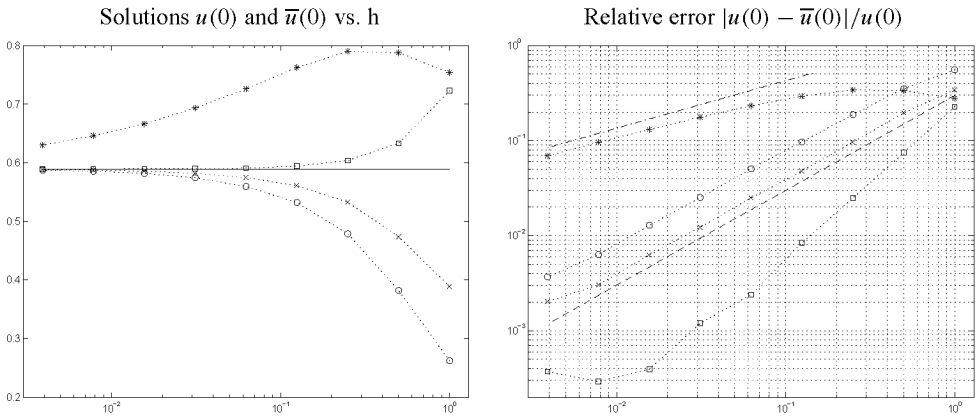

Figure 8.2: Comparison of methods (**T**), (**K1**), (**D1**) and (**D7**) when applied to the exit problem in $D = (-1, 1) \times (-1, 1)$, see Table 7.2. On the left, resulting approximate solutions are shown versus step size h. The exact solution $u(0)$ is drawn with a solid line (——). On the right the resulting relative errors are shown versus step size h. A dash-dotted line with slope one half (– · – · –) and a dashed one with slope one (– – –) are added to the plot of the errors. In both plots $\cdots * \cdots$ is method (**T**), $\cdots \circ \cdots$ is method (**K1**), $\cdots \times \cdots$ is method (**D1**) and and $\cdots \square \cdots$ is method (**D7**).

cost.

(D7): This method is clearly the best of the methods compared so far. In one dimension this method showed an error as small as the statistical fluctuations when quite large step sizes h were used. Furthermore, the resulting numerical solution $\overline{u}(0)$ only fluctuated around the exact solution $u(0)$. However, we note that for the two dimensional case method (**D7**) stops integration too late on average (as did the trivial method (**T**) in one dimension). We speculate, that this is due to the fact that we always test only for an excursion across the *closest* boundary, even in situations when two boundary segments are close (for example if y and z are close to a corner), see Remark 8.4. This implies that the methods will not detect all excursions. Nevertheless, the improvement (not only compared to method (**T**) but also to methods (**K1**) and (**D1**)) is evident when comparing the resulting errors (in the right plot of Figure 8.2). The error of method (**D7**) is clearly converging to zero at a rate faster than linear. We will see later (see Figure 8.3) that it converges in fact as fast as $\mathcal{O}(h^{\frac{3}{2}})$.

After having observed that method (**D7**) converges faster than linearly when applied to the exit problem from $D = (-1, 1) \times (-1, 1)$ the question naturally arises if extensions of

other methods from Table 7.2 show this behaviour as well. The answer is yes and is given in the form of Figure 8.3 where we show the results obtained with methods (**D4**) and (**D6**), in addition to those obtained with method (**D7**). Both of these two methods use the same test for an intermediate excursion (if $z \in D$) as (**D7**) and all of them stop at $t_k + \mathcal{H}_1$ if an excursion is detected within the step $t_k \to t_{k+1}$. Still, they differ in the method used for the computation of the exit time if the domain was clearly left, i.e. in the case that $z \notin D$: Method (**D4**) adds a uniform $(0, h)$ distributed random number to t_k whereas method (**D6**) adds the expected value, see Table 7.2.

As method (**D7**) already was found to give very small errors in the previous study (Figure 8.2), we show the results for a very large sample size, namely $N = 1.6 \cdot 10^7$.

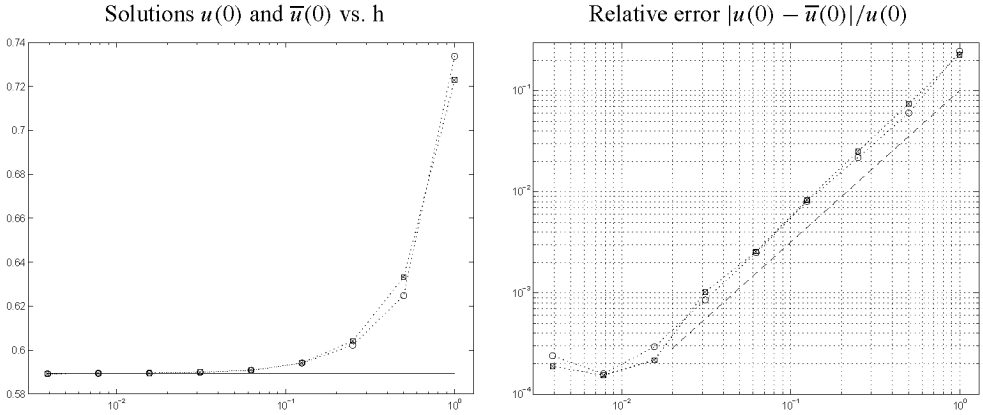

Figure 8.3: Comparison of methods (**D4**), (**D6**) and (**D7**) when applied to the exit problem in $D = (-1, 1) \times (-1, 1)$, see Table 7.2. On the left, resulting approximate solutions are shown versus step size h. The exact solution $u(0)$ is drawn with a solid line (——). On the right the resulting relative errors are shown versus step size h. A dashed line with slope three-halves (– – –) is added to the plot of the errors. In both plots $\cdots \circ \cdots$ is method (**D4**), $\cdots \times \cdots$ is method (**D6**) and and $\cdots \square \cdots$ is method (**D7**).

From Figure 8.3 we see that the methods (**D4**), (**D6**) and (**D7**) all show a very similar behaviour. They all stop integration too late on average but interestingly they all converge with $\mathcal{O}(h^{\frac{3}{2}})$, as can be seen from the right hand plot where a dashed line with slope $3/2$ has been added to confirm this conjecture. We are not able to explain this rate of convergence.

In what follows, we concentrate on method (**D7**) which is in our opinion the right approach for simulating a stopped diffusion in a domain in n-space, see Algorithm 8.1.

8.1.2-b) Hyper-spherical domains

We next consider the exit problem in hyper-spherical domains, i.e. $D = S_1^n(0)$, see (5.1). We concentrate on the extension of method (**D7**) from Table 7.2 and start with some details.

Technicalities: We first give a precise meaning to the word "closest" in Algorithm 8.1 (step 2.). We then show how to implement the map $X_b \mapsto (1, 0, \ldots, 0)^T$.

Choosing $X_b \in \partial D$: As for the square (see Definition 8.3) we separate the possible cases depending on the position of z after a preliminary step (we always have $y \in D$).

Definition 8.5. Let $\vec{y}$ and $\vec{z}$ be the position vectors of y and z respectively. Analogously, X_b is the point on $\partial D = \{x \in \mathbb{R}^n \ : \ |x| = 1\}$ with position vector $\vec{X}_b$ defined as follows:

a) $z \in D$: We set $\vec{X}_b = (\vec{y} + \vec{z})/|\vec{y} + \vec{z}|$.

b) $z \in \partial D$: We set $\vec{X}_b = \vec{z}$.

c) $z \notin \overline{D}$: We set

$$\vec{X}_b = \vec{y} + \frac{1}{2a}\left(-b + \sqrt{b^2 - 4ac}\right)(\vec{z} - \vec{y}) \tag{8.7a}$$

where

$$a = |\vec{z} - \vec{y}|^2, \quad b = \langle \vec{y}, \vec{z} - \vec{y} \rangle \quad \text{and} \quad c = |\vec{y}|^2 - 1. \tag{8.7b}$$

Now we explain the above definitions.

Remark 8.6. We discuss the choice of X_b in the cases that $z \notin \partial D$, cases a) and c).

a) Instead of setting $\vec{X}_b$ to be the minimizer of $\rho_1(x; y, z) = |\vec{x} - \vec{y}| + |\vec{x} - \vec{z}|$ (with $x \in \partial D$) which would computationally be quite expensive, especially if the dimension n grows, we minimize $\rho_2(x; y, z) = |\vec{x} - \vec{y}|^2 + |\vec{x} - \vec{z}|^2$. Note that

$$\rho_1(x; y, z) = \left(\sum_{i=1}^n \left(x^i - y^i\right)^2\right)^{\frac{1}{2}} + \left(\sum_{i=1}^n \left(x^i - z^i\right)^2\right)^{\frac{1}{2}}$$

whereas

$$\rho_2(x; y, z) = \sum_{i=1}^n \left(x^i - y^i\right)^2 + \sum_{i=1}^n \left(x^i - z^i\right)^2$$

with $\partial_i \rho_2(x; , y, z) = 2((x^i - y^i) + (x^i - z^i))$. Instead of minimizing the sum of the distances in the 1-norm we chose the 2-norm. Setting $\partial_i \rho_2(x; y, z) = 0$ and solving

for x^i one gets the components of $\vec{X}_b$. If the step size h is small enough and y and z are close, the resulting difference will be very small compared to the minimizer in the 1-norm but here we have the additional benefit of the savings in the computation time. These overall savings should not be neglected.

c) Define, as in the corresponding case for the square (Definition 8.3), for $0 \leq \alpha \leq 1$ the vector $\vec{v} = \vec{v}(\alpha)$ as $\vec{v}(\alpha) = \vec{y} + \alpha(\vec{z} - \vec{y})$. We want $X_b \in \partial D$ to be the point where $\vec{v}$ crosses ∂D. We are thus looking for $\alpha_0 \in (0, 1)$ with $|\vec{v}(\alpha_0)| = 1$. We have

$$|\vec{v}(\alpha)|^2 = |\vec{y}|^2 + 2\alpha \langle \vec{y}, \vec{z} - \vec{y} \rangle + \alpha^2 |\vec{z} - \vec{y}|^2 = 1$$

and it remains to convince ourselves that we should always pick the solution of above quadratic equation with positive discriminant. Recall that $y \in D$, $z \notin D$ and $\partial D = \{x \in \mathbb{R}^n \ : \ |x| = 1\}$. We therefore have two solutions, one with $\alpha < 0$ and (only!) one with $0 < \alpha < 1$. We have $a = |\vec{z} - \vec{y}|^2 > 0$ and it remains to show that $-b + \sqrt{b^2 - 4ac} > 0$. Indeed, we have (recall that $y \in D$)

$$\begin{array}{lll} & c = |\vec{y}|^2 - 1 < 0 & \\ \Longrightarrow & 4ac < 0 \quad (\text{as } a > 0) & \Longrightarrow \quad b^2 - 4ac > b^2 \\ \Longrightarrow & \sqrt{b^2 - 4ac} > b & \Longrightarrow \quad -b + \sqrt{b^2 - 4ac} > -b + b = 0. \end{array}$$

Hence $\alpha_0 = (-b + \sqrt{b^2 - 4ac})/(2a)$ and (8.7) follows.

About the implementation of the map $X_b \mapsto (1, 0, \ldots, 0)^T$: Once $X_b \in \partial D$ is found (see Definition 8.5), we have to map the situation to the case where

$$\tilde{X}_b = (1, 0, \ldots, 0)^T.$$

For $D = S_1^n(0)$ this is a rotation in n-space which can be written as $RX_b = \tilde{X}_b$ with an orthogonal $(n \times n)$-matrix R. In particular, we need $\tilde{y} = Ry$ and $\tilde{z} = Rz$ to apply formula (7.9) to generate $\mathcal{H}_1$ or Algorithm 7.1 to generate $\mathcal{H}_2$ respectively (see Algorithm 8.1). Of course, one could simply calculate R and perform two matrix-vector-products, but this would result in an algorithm scaling like $\mathcal{O}(n^2)$. Because these rotations have to be done in every step, this is clearly too expensive computationally, especially if the dimension n increases. Also, suppose that we have some point $\tilde{u}$ in the transformed system and we additionally need the "original" point u, $u = R^T\tilde{u}$. This would be another $\mathcal{O}(n^2)$-operation.

We therefore present a different algorithm which accomplishes this task and only scales linearly with n. It uses Givens rotations [GVL83, Section 3.4] and is described in detail in Algorithm 8.2. The inverse rotation is described in Algorithm 8.3.

To derive it, let $X_b = (X_b^1, \dots, X_b^n) =: b_{(1)} \in \mathbb{R}^n$ and define (note that here subscripts (i) are indicating the i-th rotation whereas superscripts are indices, and note further that entries which are equal to zero are omitted for better readability)

$$b_{(2)} = R_{(1)} b_{(1)} \quad \text{where} \quad R_{(1)} = \begin{pmatrix} \cos(\varphi_{(1)}) & -\sin(\varphi_{(1)}) & & & \\ \sin(\varphi_{(1)}) & \cos(\varphi_{(1)}) & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}$$

with

$$\cos(\varphi_{(1)}) = \frac{b^1_{(1)}}{w_{(1)}} \quad \text{and} \quad \sin(\varphi_{(1)}) = \frac{-b^2_{(1)}}{w_{(1)}} \quad \text{where} \quad w_{(1)} = \left(\left(b^1_{(1)} \right)^2 + \left(b^2_{(1)} \right)^2 \right)^{\frac{1}{2}}.$$

The next rotation is

$$b_{(3)} = R_{(2)} b_{(2)} \quad \text{where} \quad R_{(2)} = \begin{pmatrix} \cos(\varphi_{(2)}) & & -\sin(\varphi_{(2)}) & & \\ & 1 & & & \\ \sin(\varphi_{(2)}) & & \cos(\varphi_{(2)}) & & \\ & & & 1 & \\ & & & & \ddots \\ & & & & & 1 \end{pmatrix}$$

with

$$\cos(\varphi_{(2)}) = \frac{b^1_{(2)}}{w_{(2)}} \quad \text{and} \quad \sin(\varphi_{(2)}) = \frac{-b^3_{(2)}}{w_{(2)}} \quad \text{where} \quad w_{(2)} = \left(\left(b^1_{(2)} \right)^2 + \left(b^3_{(2)} \right)^2 \right)^{\frac{1}{2}}.$$

Clearly, $b_{(3)} = R_{(2)} R_{(1)} X_b$. Proceeding in the same way we have after $n-1$ such rotations

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

and we claim that $b_{(n)} = (1, 0, \dots, 0)^T = \tilde{X}_b$ if $X_b \in \partial D$, where $D = S_1^n(0)$. Indeed, note that

$$R_{(1)} = \frac{1}{w_{(1)}} \begin{pmatrix} b^1_{(1)} & b^2_{(1)} & & & \\ -b^2_{(1)} & b^1_{(1)} & & & \\ & & w_{(1)} & & \\ & & & \ddots & \\ & & & & w_{(1)} \end{pmatrix} \implies b_{(2)} = R_{(1)} b_{(1)} = \begin{pmatrix} w_{(1)} \\ 0 \\ b^3_{(1)} \\ b^4_{(1)} \\ \vdots \\ b^n_{(1)} \end{pmatrix}$$

and analogously $b_{(3)} = R_{(2)}b_{(2)} = (w_{(2)}, 0, 0, b^4_{(2)}, \ldots, b^n_{(2)})^T$ with $b^i_{(2)} = b^i_{(1)}$ for $i = 4, \ldots, n$. We have thus found the recurrence relation for $i = 2, \ldots, n-1$

$$b_{(i)} = (w_{(i-1)}, 0, \ldots, 0, b^{i+1}_{(i-1)}, \ldots, b^n_{(i-1)})^T = (w_{(i-1)}, 0, \ldots, 0, b^{i+1}_{(1)}, \ldots, b^n_{(1)})^T$$

and analogously we find $b_{(n)} = (w_{(n-1)}, 0, \ldots, 0)^T$. With the recurrence relation

$$w^2_{(i)} = w^2_{(i-1)} + \left(b^{i+1}_{(i)}\right)^2 \quad \text{for} \quad i = 2, \ldots, n-1 \tag{8.8}$$

and with the fact that $b^j_{(i)} = b^j_{(1)} = X^j_b$ for $j = i+1, \ldots, n$ it is easy to see that

$$w^2_{(n-1)} = \sum_{i=1}^{n} \left(b^i_{(1)}\right)^2 = |X_b|^2 = 1$$

as $X_b \in \partial D$, and the claim follows.

We summarize this in Algorithm 8.2 which performs the transformation $y \mapsto \tilde{y}$ corresponding to $X_b \mapsto (1, 0, \ldots, 0)^T$ (for this special domain we do not need any scaling). For simplicity, we write down only the rotation $\tilde{y} = Ry$, but of course, the rotation $\tilde{z} = Rz$ can be calculated simultaneously.

The calculated angles $\varphi_{(i)}$ are stored for the connected inverse transformation (using $R^{-1} = R^T$). More precisely, the corresponding trigonometric functions, $\cos(\varphi_{(i)})$ and $\sin(\varphi_{(i)})$ are stored in arrays c and s. These arrays thus have length $n-1$. For algorithmic purpose we write $w_i \equiv w_{(i)}$ etc.

The inverse transformation is performed similarly using

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

We show it for a generic vector $\tilde{u}$ (defined in the system corresponding to $\tilde{y}$ and $\tilde{z}$) in Algorithm 8.3, $\tilde{u} \mapsto u = R^{-1}\tilde{u}$. It is assumed that the angles $\varphi_{(i)}$ have been previously calculated (for example during execution of Algorithm 8.2). More precisely, the corresponding trigonometric functions, $\cos(\varphi_{(i)})$ and $\sin(\varphi_{(i)})$ are available in arrays c and s.

Results: We undertook simulations for the exit problem in dimensions $n = 2^k$ with $k = 1, 2, \ldots, 7$. For clarity we first show the lower dimensional problems (in Figure 8.4 we show results for $k = 1, \ldots, 4$) and then the higher dimensional ones (in Figures 8.5 and 8.6 we show results for $k = 5, 6, 7$). We always show the numerical solutions $\overline{u}(0)$ and the corresponding exact solutions $u(0) = n^{-1} = 2^{-k}$ in the left hand plot and show the resulting relative errors in the right hand plot. Sample size was always $N = 4 \cdot 10^6$ and resulting numerical standard deviations (see (5.3')) were always smaller than the plot symbols.

Algorithm 8.2 Rotation $y \mapsto \tilde{y} = Ry$ corresponding to $\partial S_1^n(0) \ni X_b \mapsto RX_b = (1, 0, \ldots, 0)^T$ using $n-1$ Givens rotations.

$\tilde{y} \leftarrow y$ {initialization}
$w_0 \leftarrow X_b^1$ {$i = 0$ is special}
$R_X \leftarrow (X_b^1)^2 + (X_b^2)^2$ {R_X stores the square of w}
for $i = 1$ to $n-1$ **do**
  $w_i \leftarrow \sqrt{R_X}$
  $c_i \leftarrow w_{i-1}/w_i$ and $s_i \leftarrow -X_b^{i+1}/w_i$ {note that $b_{(i)}^1 = w_{i-1}$ and $b_{(i)}^{i+1} = X_b^{i+1}$}
  $\underline{y} \leftarrow \tilde{y}^1$ and $\overline{y} \leftarrow \tilde{y}^{i+1}$ {store old values}
  {rotation: multiplication with $R_{(i)}$ affects only entries (k, l) with $k, l \in \{1, i+1\}$}
  $\tilde{y}^1 \leftarrow c_i \underline{y} - s_i \overline{y}$ and $\tilde{y}^{i+1} \leftarrow s_i \underline{y} + c_i \overline{y}$
  if $i < n-1$ **then**
    $R_X \leftarrow R_X + (X_b^{i+2})^2$ {use (8.8) to compute next $R_X = w_{(i+1)}^2$}
  end if
end for
Ensure: $\tilde{y}$ is in the system corresponding to $\tilde{X}_b = (1, 0, \ldots, 0)^T$
Ensure: $c_i = \cos(\varphi_i)$ and $s_i = \sin(\varphi_i)$, $i = 1, \ldots, n-1$

Algorithm 8.3 Rotation $\tilde{u} \mapsto u = R^{-1}\tilde{u} = R^T u$ corresponding to $(1, 0, \ldots, 0)^T \mapsto X_b \in \partial S_1^n(0)$ using $n-1$ Givens rotations.

Require: $c_i = \cos(\varphi_i)$ and $s_i = \sin(\varphi_i)$, $i = 1, \ldots, n-1$
$u \leftarrow \tilde{u}$ {initialization}
for $i = n-1$ to 1 by -1 **do**
  $\underline{u} \leftarrow u^1$ and $\overline{u} \leftarrow u^{i+1}$ {save old values}
  $u^1 \leftarrow c_i \underline{u} + s_i \overline{u}$ and $u^{i+1} \leftarrow -s_i \underline{u} + c_i \overline{u}$ {rotation: multiplication with $R_{(i)}^{-1} = R_{(i)}^T$}
end for
Ensure: u is in the system corresponding to X_b

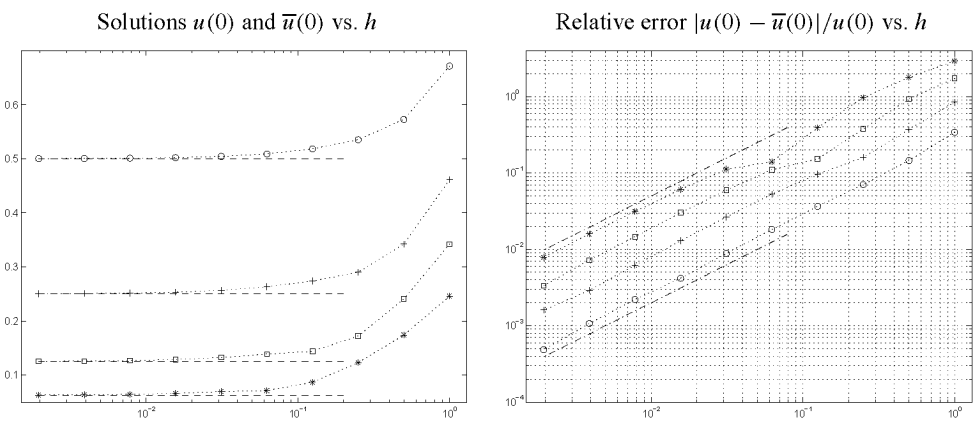

Figure 8.4: Exit problem from n-spheres in dimensions $n = 2$ ($\cdots\circ\cdots$), $n = 4$ ($\cdots+\cdots$), $n = 8$ ($\cdots\Box\cdots$) and $n = 16$ ($\cdots*\cdots$). On the left the approximate solutions $\overline{u}(0)$ are shown vs. step size h and corresponding exact solutions are indicated by dashed lines (– – –). In the plot of the relative errors vs. step size h (shown on the right) two dash-dotted lines with slope one (– · – · –) are added.

From the results in Figure 8.4 we see that the $\mathcal{O}(h)$-convergence of the error is achieved for sufficiently small step sizes h. Note that when $D = S_1^n(0)$ we really have to *approximate* ∂D locally by its tangent plane which introduces an additional error. We also note, that (independent of the dimension n) the approximate solution is always too large and approaches the exact solution $u(0)$ from above for decreasing h. This was already observed in the tests where $D = (-1, 1) \times (-1, 1)$, see Figures 8.2 and 8.3.

We next turn our attention to higher dimensions, see Figure 8.5.

From Figure 8.5 we see that the approximate solutions for $n = 32, 64$ and $n = 128$ show the same qualitative behavior as in dimensions $n = 2, 4, 8, 16$ (Figure 8.4). Although the convergence rate is less clear and for the step sizes shown, the asymptotical order-h-dependence of the error is not yet convincing. We think that this is due to the fact, that the higher the dimension, the cruder the one dimensional test for an excursion becomes.

However, we argue, that as h becomes smaller and smaller, this one dimensional approximation of the path's approach to the boundary should become better and better. We even conjecture that the linear convergence of the error is *independent of the dimension* and therefore show results from simulations in $n = 64$ and $n = 128$ dimensions with smaller step sizes h, see Figure 8.6. For clarity we scaled the approximation $\overline{u}(0)$ by a factor of two for the case of $n = 128$. The corresponding exact solution was scaled in the same way.

From Figure 8.6 we see that the conjectured asymptotic linear convergence of the error

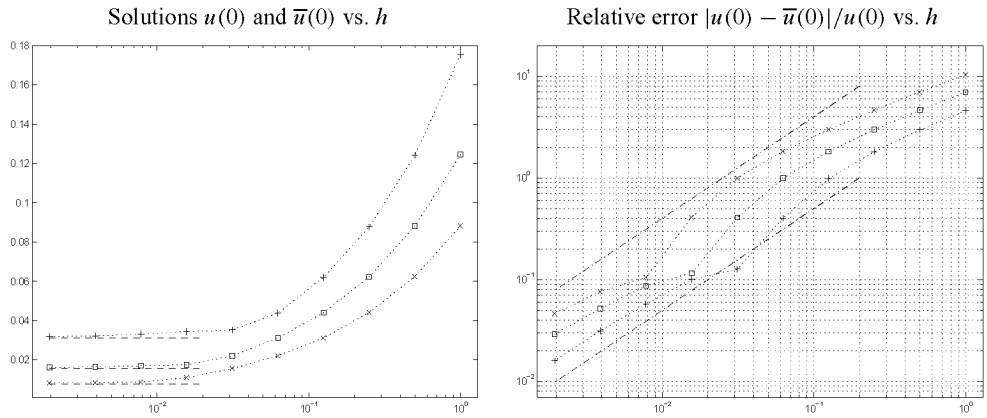

Figure 8.5: Exit problem from n-spheres in dimensions $n = 32$ ($\cdots + \cdots$), $n = 64$ ($\cdots \square \cdots$) and $n = 128$ ($\cdots \times \cdots$). See Figure 8.4 for more information.

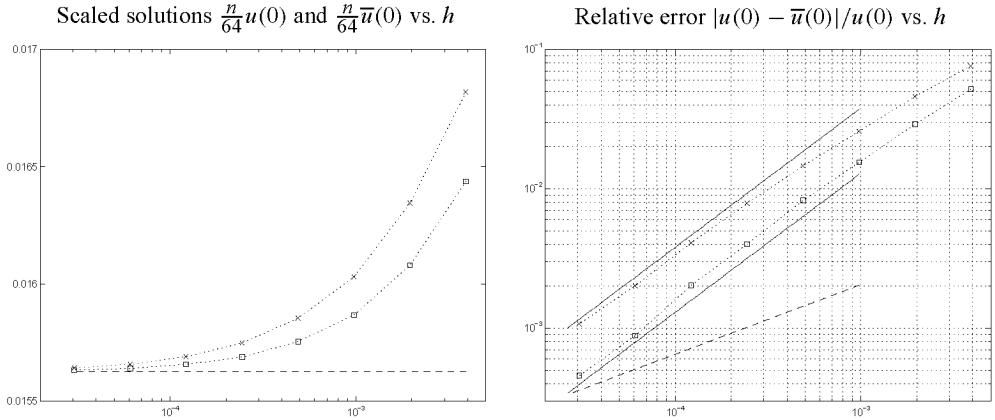

Figure 8.6: Exit problem from n-spheres in dimensions $n = 64$ ($\cdots \square \cdots$) and $n = 128$ ($\cdots \times \cdots$). On the left, corresponding exact solutions are indicated by a dashed line (– – –). Note that in the left plot both $\overline{u}$ and u where multiplied by a factor of two for $n = 128$ for clarity. Note also the different scalings compared to Figure 8.5: step sizes h are much smaller in this figure. In the plot of the relative errors shown on the right, two solid lines with slope one (——) and a dashed line with slope one half (– – –) are added .

is indeed achieved by method (**D7**) (see Algorithm 8.1) in dimensions as high as 64 or even 128. We added a dashed line with slope one half (– – –) to show that for sufficiently small h the error converges much better than in the case of the trivial stopping algorithm (method (**T**)).

8.2 Exit point: space sampling

In the previous section we described how to approximate the first exit time $\tau = \tau_D$ of Brownian motion from a domain D in n-space. As noted earlier, in order to approximate the solution $u(x)$ of Poisson's equation (5.2), by its stochastic representation $u(x) = \mathbb{E}[\psi(X(\tau)) + f(\tau)]$, an approximation of the exit point $X(\tau)$ is required as soon as the boundary condition ψ is no longer a constant. We first describe in Section 8.2.1 this approximation procedure and then show in Section 8.2.2 the results of a series of numerical experiments performed with the resulting algorithm (see Algorithm 8.4).

8.2.1 Derivation of the algorithm

Using the notation introduced at the beginning of Section 8.1.1, we cite the result which is the starting point for the approximation of $X(\tau)$, see [LS89, Lemma 7]. We recall that $\tau = \inf_t\{X^1(t) \geq b\}$. Assume $z^1 > b$. Given $X^1(s)$, $s \in [0, \tau]$, the joint distribution of $X^i(\tau)$ conditioned on $X(t) = z$ is given by (with $X(0) = 0$)

$$z^i \frac{\tau}{t} + \sqrt{\tau\left(1 - \frac{\tau}{t}\right)} \cdot \xi^i \quad \text{where} \quad \xi^i \sim \mathcal{N}(0, 1), \quad i = 2, \ldots, n. \tag{8.9}$$

The assumption that $z^1 > b$ (or equivalently that $z^1 \geq b$) can be replaced by the assumption that $\tau < t$ (or $\tau \leq t$ respectively). We can therefore apply (8.9) as soon as the integration process of the system of SDEs (6.1) is stopped (as soon as an approximation of τ is found by Algorithm 8.1). The extension to the general case with $y = X_k$ and $z = X_{k+1}$ with $\tau \leq t_{k+1}$ is then straightforward and we therefore skip the calculation. The resulting joint distribution of $X^i(\tau)$ is (recall that $t_{k+1} = t_k + h$)

$$y^i + (z^i - y^i)\frac{\tau}{h} + \sqrt{\tau\left(1 - \frac{\tau}{h}\right)} \cdot \xi^i \quad \text{where} \quad \xi^i \sim \mathcal{N}(0, 1), \quad i = 2, \ldots, n. \tag{8.10}$$

It only remains to incorporate the rotation of the map $X_b \mapsto (1, 0, \ldots, 0)^T$. We then have $\tilde{X}_b^1 = 1$ by definition and set in the transformed system $\tilde{X}_b^i$ to a random number following (8.10), in detail

$$\tilde{X}_b^i = \tilde{y}^i + (\tilde{z}^i - \tilde{y}^i)\frac{\tau}{h} + \sqrt{\tau\left(1 - \frac{\tau}{h}\right)} \cdot \xi^i \quad \text{where} \quad \xi^i \sim \mathcal{N}(0, 1), \quad i = 2, \ldots, n. \tag{8.11}$$

With formula (8.11) we can thus sample an approximation for $\tilde{X}(\tau)$ on the tangent hyperplane through $(1, 0, \ldots, 0)^T$. Once $\tilde{X}_b$ is generated, we project back to ∂D. For the unit n-sphere this can be done by simply scaling $\tilde{X}_b$ to unit length.

Rotating back (corresponding to the map $(1, 0, \ldots, 0)^T \mapsto X_b$ and described by the multiplication with R^{-1}) we find the approximation for the exit point as $X(\tau) \approx R^{-1}\tilde{X}_b$. Of course, we do not perform the matrix-vector-product but apply $n - 1$ Givens rotations, see Algorithm 8.3. Doing so, we get the final Algorithm 8.4 which incorporates the missing details of Algorithm 8.1.

Algorithm 8.4 Simulate one path (with index (i)) to get an approximation $\overline{u}^{(i)}$ for $u^{(i)} = \psi(X^{(i)}(\tau^{(i)})) + f^{(i)}(\tau)$ with $u(x) = \mathbb{E}[u^{(i)}]$, see (6.1) and the following paragraph. The approximation is calculated with fixed step size $h > 0$ small enough. For better readability the sample index (i) is omitted.

$y \leftarrow x$, $f \leftarrow 0$ {initialize with initial conditions of SDEs (6.1)}
for $k = 0, 1, 2, \ldots$ **do** {integrate till τ, see Algorithm 8.1}
  $z^i = y^i + \sqrt{h}\xi$, $\xi \sim \mathcal{N}(0, 1)$ for $i = 1, \ldots, n$
  $X_b \leftarrow$ a point on ∂D closest to y and z (see Definition 8.5)
  $\tilde{y} \leftarrow Ry$ and $\tilde{z} \leftarrow Rz$ with Algorithm 8.2
  if $\tilde{z}^1 \geq 1$ **then** {D was clearly left}
    Generate $\mathcal{H}_2$ according to Algorithm 7.1 with input variables $\tilde{y}^1$, $\tilde{z}^1$ and $b = 1$.
    $f(\tau) \leftarrow f + \mathcal{H}_2 g(y)$
    STOP
  else {check for an intermediate excursion}
    Generate $\mathcal{H}_1$ as in (7.9) with input variables $\tilde{y}^1$, $\tilde{z}^1$ and $b = 1$
    if $\mathcal{H}_1 \leq h$ **then** {excursion detected: stop integration at $t_k + \mathcal{H}_1$}
      $f(\tau) \leftarrow f + \mathcal{H}_1 g(y)$
      STOP
    else {continue integration: accept preliminary step}
      $f \leftarrow f + hg(y)$ and $y \leftarrow z$
    end if
  end if
end for
$\tilde{X}^1 \leftarrow 1$ and for $i = 2, \ldots, n$ generate $\tilde{X}^i$ according to (8.10) {$\tilde{X} \in \{x \in \mathbb{R}^n : x^1 = 1\}$}
$\tilde{X} \leftarrow \tilde{X}/|\tilde{X}|$ {project back on ∂D}
$X(\tau) \leftarrow R^{-1}\tilde{X}$ with Algorithm 8.3
$\overline{u} \leftarrow \psi(X(\tau)) + f(\tau)$

In the next section we show results obtained by applying Algorithm 8.4 to solve a general

Poisson's equation of type (5.2).

8.2.2 Numerical experiments

We show results in $D = S_1^{32}(0) = \{x \in \mathbb{R}^{32} \ : \ |x| < 1\}$. We solve Poisson's equation (5.2) by the stochastic representation of the solution (6.1) using Algorithm 8.4. As convergence of this algorithm is not proved, we select test examples such that they become more and more difficult. The test examples are:

(P0): Exit problem:

$$g(x) \equiv 1 \quad \text{and} \quad \psi(x) \equiv 0 \implies u(x) = \frac{1}{32}\left(1 - |x|^2\right) \tag{8.12}$$

This is the simplest non-trivial example as the numerical integration procedure is exact ($g \equiv 1$ implies that all the terms in the remainder of the corresponding Itô-Taylor expansion vanish, see Section 2.2.2) and for this example no space sampling is required.

(P1): We include a non-constant boundary condition:

$$g(x) \equiv 1 \quad \text{and} \quad \psi(x) = \sum_{i=1}^{32} x^i \implies u(x) = \frac{1}{32}\left(1 - |x|^2\right) + \sum_{i=1}^{32} x^i \tag{8.13}$$

For this problem, integration is still exact (as again $g(x) \equiv 1$), but in contrast to (P0), formulae (8.12), an approximation for $X(\tau)$ is required in order to evaluate the boundary condition for every path. The boundary condition does not vary significantly and is linear.

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

(P2) We make $g(x)$ harder to integrate and choose a non-linear boundary condition. We show results of two variants of this problem:

(P2a)

$$g(x) = -\langle x, x\rangle \quad \text{and} \quad \psi(x) = \frac{1}{6}\sum_{i=1}^{32}\left(x^i\right)^4 \implies u(x) = \psi(x) \tag{8.14a}$$

This problem has still (as (P0) and (P1), formulae (8.12) and (8.13)) an intrinsic symmetry: $g(\cdot)$ is constant on the spheres $|x| \equiv \|x\|_2 = r = const$ and $\psi(\cdot)$ is constant when $\|x\|_4^4 \equiv \sum_i (x^i)^4 = R = const$ respectively. We therefore consider the following extension:

(P2b)

$$g(x) = -\sum_{i=1}^{32} i \left(x^i\right)^2 \quad \text{and} \quad \psi(x) = \frac{1}{6}\sum_{i=1}^{32} i \left(x^i\right)^4 \implies u(x) = \psi(x) \tag{8.14b}$$

(P3) For our previous experiments we always chose $g(x)$ and $\psi(x)$ as polynomials – something we want to change now. The last example we use to test Algorithm 8.4 has already been used in Part I to test Algorithm 3.2, see Section 5.3.2-b). Let us recall that problem from (5.9): We set for $k^i = 1, 2, \ldots,$ with $k = (k^i)_{i=1,\ldots,32}$

$$g(x) = \prod_{i=1}^{32} \cos(2\pi k^i x^i) \quad \text{and} \quad \psi(x) = \frac{g(x)}{2\pi^2 |k|^2} \implies u(x) = \psi(x). \tag{5.9'}$$

Results of the first test problem (P0) have already been discussed in Section 8.1.2-b) (Figure 8.5) where a linear convergence was observed for sufficiently small step sizes h.

We next show results for problem (P1), (8.13), see Figure 8.7. Recall that this problem is the first one to incorporate a non-constant boundary condition $\psi(x)$.

From Figure 8.7 we see that for $\psi(x) = \sum_i x^i$ (problem (P1), see (8.13)), convergence is still of $\mathcal{O}(h)$ for sufficiently small step sizes h. In particular we see from this problem that the procedure of *sampling* an approximation for $X(\tau)$ on the tangent hyper-plane and projecting back on ∂D works remarkably well. However, we note that if $x^i = 0$ for $i = 1, \ldots, 32$ (i.e. we start the integration of the Brownian motion at the center of the spherical domain D) the errors obtained are very small and very quickly reach the regime where the statistical error (of $\mathcal{O}(1/\sqrt{N})$) dominates. This is clear, as in that case we have to sample the boundary condition *uniformly* on ∂D, which is achieved by Algorithm 8.4 in a perfect way independent of the chosen step size h. When moving away from the center ($x^i \neq 0$ for at least one i) we see that the convergence of Algorithm 8.4 is indeed of linear order for sufficiently small step sizes h (and sufficiently large sample size N). We have shown results where x lies on the diagonal, $x^i = \alpha/32$ for $i = 1, \ldots, 32$, and above result holds independent of α.

The next two plots, given in Figure 8.8, show results for problem (P2), (8.14). We evaluated the solution at $x = (x^i)_i$ with $x^i = 0.05$ and $x^i = 0.1$ (i.e. on the first (hyper-) diagonal away from the center of the n-sphere).

From Figure 8.8 we see that the resulting error when Algorithm 8.4 is applied to Poisson's equation with $g(x)$ and $\psi(x)$ as in (8.14) shows a linear decay for sufficiently small

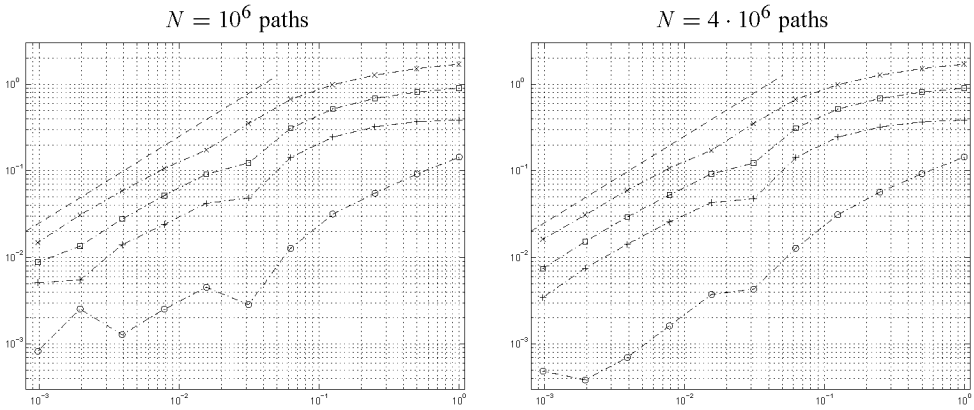

Figure 8.7: Results for problem (P1) in 32 dimensions, see (8.13). Shown are the (absolute) errors $|u(x) - \overline{u}(0)|$ vs. step size h. Results on the left hand side are for the case of a sample size of $N = 10^6$ whilst on the right hand side we show results for $N = 4 \cdot 10^6$ paths. Solution is evaluated at $x^i = \alpha/32$ with $\alpha = 0, 1, 2, 4$ for $i = 1, \ldots, 32$. $- \cdot \circ \cdot -$ is for $\alpha = 0$, $- \cdot + \cdot -$ is for $\alpha = 1$, $- \cdot \square \cdot -$ is for $\alpha = 2$ and $- \cdot \times \cdot -$ is for $\alpha = 4$. A dashed line with slope one $(- - -)$ is added to both plots.

step sizes h. This holds for both variants (8.14a) and (8.14b). This result is encouraging, as it shows that the two main parts of Algorithm 8.4, the approximation of a stopped integration and the sampling procedure of the first exit point, function correctly even if

- $g(x)$ can no longer be integrated exactly,
- the starting point x is such that the first exit point $X(\tau)$ is distributed non-uniformly on ∂D,
- $\psi(x)$ is non-linear,
- and finally $g(x)$ and $\psi(x)$ cannot be rewritten as functions of $\|x\|_p$ for some $p \geq 1$.

.

To conclude the series of numerical experiments we turn to problem (P3), see (5.9'). Results are shown in Figures 8.9 to 8.11. We chose two different k-vectors, $k = (1, \ldots, 1)$ and $k = (3, 2, 1, \ldots, 1)$ (both with length 32). Results are shown for studies in which the solution is evaluated at three different x values. In detail, the solution was evaluated at $x = (x^i)_i$ with $x^i = 0$ (in Figure 8.9), $x^i = 0.05$ (in Figure 8.10) and finally with $x^i = 0.1$ (in Figure 8.11) for $i = 1, \ldots, 32$.

$$g(x) = -\sum_{i=1}^{32} (x^i)^2, \qquad \psi(x) = \frac{1}{6}\sum_{i=1}^{32} (x^i)^4$$

$$g(x) = -\sum_{i=1}^{32} i(x^i)^2, \qquad \psi(x) = \frac{1}{6}\sum_{i=1}^{32} i(x^i)^4$$

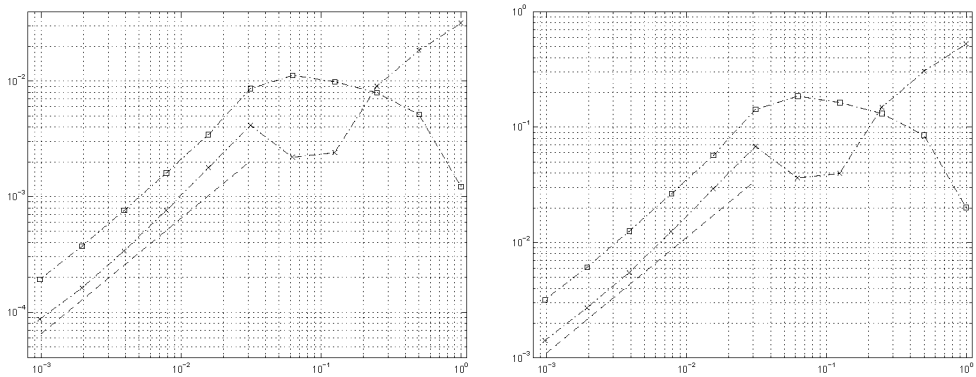

Figure 8.8: Results for problem (P2) in 32 dimensions, see (8.14). On the left, $g(x)$ and $\psi(x)$ are as in (8.14a) and on the right they are as defined in (8.14b). Solution is evaluated at $x^i = 0.05$ and at $x^i = 0.1$ for $i = 1, \ldots, 32$. Shown are the (absolute) errors $|u(x) - \overline{u}(x)|$ vs. step size h. Sample size is $N = 4 \cdot 10^6$. $- \cdot \square \cdot -$ is for $x^i = 0.05$ and $- \cdot \times \cdot -$ is for $x^i = 0.1$. A dashed line with slope one ($- - -$) is added to both plots.

From Figures 8.9, 8.10 and 8.11 we see that the procedure works very well for all variants of the problem (P3), see (5.9'). Independent of the chosen k-vectors and independent of the starting point of the simulated Brownian motion, the error scales asymptotically linear in h. From the plots on the left we see that the approximation of the corresponding exact solutions is very good.

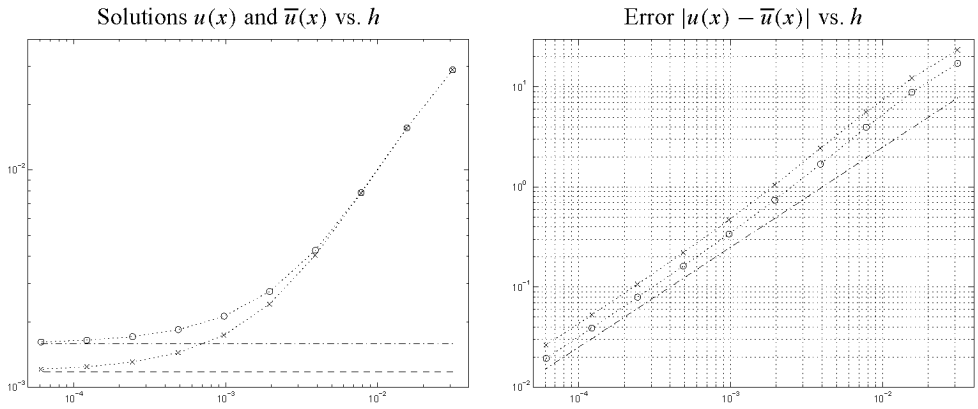

Figure 8.9: Results for problem (P3) in 32 dimensions, see (5.9'). On the left, solutions $u(x)$ and corresponding approximations $\overline{u}(x)$ are shown vs. step size h. On the right, resulting errors $|u(x) - \overline{u}(x)|$ are shown (vs. h). There, a dashed line with slope one (– – –) is added. Solution is evaluated at $x^i = 0$ ($i = 1, \ldots, 32$). Results for two different k-vectors are shown: $\circ$ corresponds to $k = (1, \ldots, 1)$ (the corresponding exact solution $u(x)$ is – · – · –) and $\times$ corresponds to $k = (3, 2, 1, 1, \ldots, 1)$ respectively (with corresponding exact solution $u(x)$ indicated by the dashed line (– – –)). Sample size is $N = 10^6$.

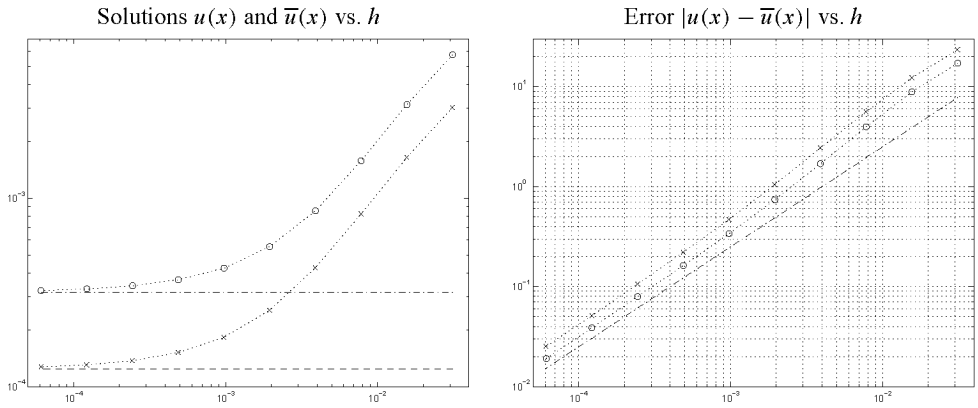

Figure 8.10: Test problem (P3) in 32 dimensions as in Figure 8.9 but solution is evaluated at $x^i = 0.05$ (as opposed to $x^i = 0$). See Figure 8.9 for further details.

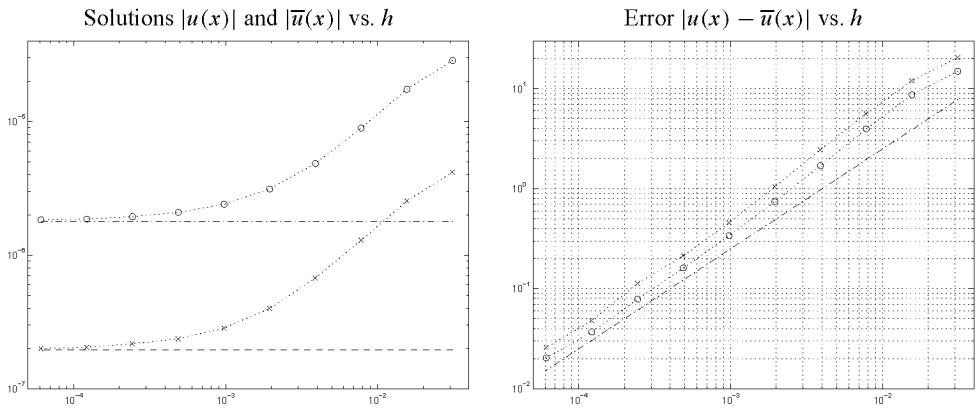

Figure 8.11: Test problem (P3) in 32 dimensions as in Figures 8.9 and 8.10 but solution is evaluated at $x^i = 0.1$ (as opposed to $x^i = 0$ and $x^i = 0.05$ respectively). For $k = (3, 2, 1, 1, \ldots, 1)$ we have $u(x) < 0$. For better comparison, we therefore plotted $-u(x)$ and $-\overline{u}(x)$ in this case (left plot, symbols $---$ and $\cdots \times \cdots$ respectively). See Figure 8.9 for further details.

Chapter 9

Conclusions and final remarks

In this part, we have presented a second approach to solve high dimensional Dirichlet problems. In this approach we try to sample the occurring random variables with distributions as exact as possible. Although the goal was the development of an algorithm suitable for the simulation in high dimensional settings, we started in one dimension. There, we extended ideas of an excursion test widely used for killed diffusions to stopped diffusions. We have shown how to sample the resulting hitting times exact in distribution. This is in contrast to another approach by Giraudo et al. [GSZ01]. Their work is based on the simulation of a Brownian bridge where a finer time scale is used once the path is close to the boundary and the possibility of an excursion has to be taken into account. Instead, we sample directly from the analytical density.

We were concerned mainly with processes which have constant diffusion coefficient σ, but extensions in the spirit of Algorithm 7.2 also seem to be possible to general $\sigma = \sigma(x)$. Of course, in this situation the distribution one samples from changes, but corresponding densities or suitable approximations are available, see [Bal95, GS99, BCI99, BC02]. In these works, additionally, possible applications in computational finance are discussed.

From our numerical experiments in one dimension, our approach seems superior to any other (heuristic) extensions one might think of. Not only is the order of convergence in the weak sense of order h with a constant lower than the simple extension of the algorithm for killed diffusions, but also a statistical test showed its ability to approximate first hitting time distributions extremely well.

Motivated by these results we showed how to extend the ideas presented so far to dimensions $n > 1$. There, we first showed that a local approximation of ∂D by a half-space enabled us to apply a straight forward extension of the proposed method. In fact, the formulae obtained where as simple as in the one dimensional setting. But in higher dimension,

beside of the approximation of the first exit time from D, an algorithm to approximate the corresponding exit point is needed. We have shown how this task can be accomplished. In the same spirit as the procedure performed for the approximation of the first exit time, we approximate the exit points through sampling as well.

In one dimension, all the results we obtained were very accurate and extremely satisfactory. The observations in one dimension carried over to the high dimensional problems. There too, very good convergence behavior was noticed in all the test problems. An exit problem in dimensions up to $n = 128$ showed weak order one convergence for sufficiently small step sizes.

The distinguishing feature of Dirichlet problems is the presence of a boundary condition $u(x) = \psi(x)$ for $x \in \partial D$. A series of test examples showed that convergence is asymptotically still of order one, and this held true for any boundary condition we tested. We showed results for this general situation in $n = 32$ dimensions to demonstrate the effectiveness of our method, Algorithm 8.4, in high dimensions.

It is worth noting that the implementation does not become more difficult as the dimension n increases. Algorithm 8.4 is completely independent of $n \geq 2$ (concerning the difficulty of implementation). Firstly, no triangulation of D (a grid) is needed and secondly no boundary layers or difficult transformations of any functions are used. This is clearly a big advantage over deterministic methods such as finite element or finite difference methods (FEM or FDM). On the other hand, we only get the solution of the PDE *pointwise*, i.e. we compute it only at a few points $x \in D$, whereas for example the FEM constructs an approximation $\forall x \in D$.

Furthermore, the method proposed is embarrassingly parallel and almost no additional work is required to run it on parallel machines: Each sample is independent and the only communication overhead is in the computation of the mean at the very end of a simulation. The methods presented therefore fit perfectly on heterogeneous cluster architectures where the machines are typically of different quality and the communication between individual nodes is usually slow. Note that here we did not emphasize or discuss these details. Furthermore, we used for example the same random number generator on every node with only a different seed in order to get "independent" streams of random numbers, see [Mas99] for possible improvements.

Notation

Dirichlet problem, PDE for $u(x)$

n	dimension of the problem
D	domain
∂D	boundary of D
$\overline{D}$	$D \cup \partial D$
a	$(n \times n)$-diffusion matrix $a = (a^{ij})_{ij}$
b	coefficient n-vector $b = (b^i)_i$ for first order derivatives
L	$L = \frac{1}{2} a^{ij} \partial_i \partial_j + b^i \partial_i$
ψ	Dirichlet boundary condition
g	inhomogeneity
c	potential, usually $c(x) \leq 0 \; \forall x \in D$

Stopping times

τ_D	first exit time from D
$\tau_D^x = \tau_D(x)$	first exit time from D for a process starting at x
τ	$= \tau_D$ or $= \tau_D^x$ if no confusion arises
H_b	first hitting time of level b
$H_b(x)$	first hitting time of level b for a process starting at x

Path generating SDE of dimension n

X	solution vector
W	driving Wiener process of dimension n
σ	diffusion matrix $\sigma = (\sigma^{ij})_{ij}$, $\sigma \sigma^T = a$
b	drift vector $b = (b^i)_i$
L	$L = \frac{1}{2} \sigma^{jr} \sigma^{kr} \partial_j \partial_k + b^i \partial_i$

Processes

X	process solving the SDE
X_x	solution to initial condition $X(0) = x$
$X_{t,x}$	solution to condition $X(t) = x$
$Y_{t,x,y}$	solution to conditions $X(t) = x$ and $Y(t) = y$
$X^{y,h,z}$	Brownian bridge pinned at (t, y) and at $(t + h, z)$

General SDE

d	dimension of solution vector
m	dimension of driving Wiener process
$\mathcal{X}$	solution vector of dimension d
S	diffusion matrix $S = (S^{ij})_{ij}$
B	drift vector $B = (B^i)_i$

Wagner-Platen expansion

α	multi-index
$l(a)$	length of α
$n(a)$	number of elements of α which are equal to zero
v	multi-index of length zero
$\mathcal{M}$	the set of all multi-indices
$\mathcal{A}$	a hierarchical set of multi-indices
$\mathcal{B}(\mathcal{A})$	the remainder set of $\mathcal{A}$
R	remainder of truncated Itô-Taylor expansion
Γ_β	hierarchical set for weak approximation
β	one step approximation is of weak order $\beta + 1$, $\beta = 0, 1, \ldots$

Vectors and matrices

For a vector $x \in \mathbb{R}^d$ and a matrix $A \in \mathbb{R}^{n \times m}$ we use the following notations

x^i	a component of x
$x = (x^i)_{i=1,\ldots,n}$	the vector built by its components
A^{ij}	a component of A
$A = (A^{ij})_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}}$	the matrix built by its components
$A^{\cdot j}$	the j-th column of A
$A^{i\cdot}$	the i-th row of A

p-norms of vectors and other vector operations

$\langle x, y\rangle$	$\sum_i x^i y^i$ (inner product)
$\|x\|_2^2$	$\langle x, x\rangle$
$\|x\|$	$\|x\|_2$
$\|x\|_4$	$\left(\sum_i (x^i)^4\right)^{\frac{1}{4}}$

Distributions

$\mathcal{U}(0, 1)$	uniform in $(0, 1)$
$\mathcal{N}(m, v)$	normal (Gaussian) with mean m and variance v
$\mathcal{IG}(\lambda, \mu)$	inverse Gaussian with parameters $\lambda, \mu > 0$

Probabilities, expectations, etc.

$\mathbb{P}[A, B]$	probability that both events A and B occur
$\mathbb{P}[A; B]$	conditional probability of an event A given B
$\mathbb{E}[A; B]$	to $\mathbb{P}[A; B]$ connected conditional expectation
$\mathbb{P}_x[A]$	conditional probability of an event A, given the process X starts at time $t = 0$ at x
$\mathbb{E}_x[A]$	to $\mathbb{P}_x[A]$ connected conditional expectation
$\mathbb{P}_{y,h,z}[A]$	conditional probability of an event A, given the process X is at t_k at y and at $t_k + h$ at z (Brownian bridge pinned at (t_k, y) and $(t_k + h, z)$)
$\mathbb{E}_{y,h,z}[A]$	to $\mathbb{P}_{y,h,z}[A]$ connected conditional expectation
$\mathrm{Cov}(\cdot, \cdot)$	covariance

Spaces

$C^k(D, \mathbb{R})$	k times continuously differentiable functions $f : D \to \mathbb{R}$
F	functions with polynomial growth
$C_P^k(D, \mathbb{R})$	functions in $C^k(D, \mathbb{R})$ which together with their partial derivatives up to order k belong to F
A_D	Lipschitz continuous functions in $C_P^6(D, \mathbb{R})$

Miscellaneous symbols

$a \wedge b$	$\min\{a, b\}$
$a \vee b$	$\max\{a, b\}$
$r \sim P$	r (usually a random variable) has the distribution or density P

Summation convention where repeated indices are summed over is applied frequently, hence, for example

$$v^i a^{ij} v^j \equiv \sum_i \sum_j v^i a^{ij} v^j \equiv \sum_{i,j} v^i a^{ij} v^j \quad \text{or} \quad \Lambda_r I_{ir} \equiv \sum_r \Lambda_r I_{ir}.$$

Bibliography

[AS64] Milton Abramowitz and Irene A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55 of *National Bureau of Standards Applied Mathematics Series*. For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C., 1964.

[Bal95] Paolo Baldi. Exact asymptotics for the probability of exit from a domain and applications to simulation. *Ann. Probab.*, 23(4):1644–1670, 1995.

[BC02] Paolo Baldi and Lucia Caramellino. Asymptotics of hitting probabilities for general one-dimensional pinned diffusions. *Ann. Appl. Probab.*, 12(3):1071–1095, 2002.

[BCI99] Paolo Baldi, Lucia Caramellino, and Maria Gabriella Iovino. Pricing general barrier options: a numerical approach using sharp large deviations. *Math. Finance*, 9(4):293–322, 1999.

[BM58] G. E. P. Box and Mervin E. Muller. A note on the generation of random normal deviates. *Ann. Math. Stat.*, 29:610–611, 1958.

[BP03] F. M. Buchmann and W. P. Petersen. Solving dirichlet problems numerically using the Feynman-Kac representation. *BIT*, 43(3):519–540, 2003.

[BS02] Andrei N. Borodin and Paavo Salminen. *Handbook of Brownian motion—facts and formulae*. Probability and its Applications. Birkhäuser Verlag, Basel, second edition, 2002.

[Dag88] John Dagpunar. *Principles of random variate generation*. Oxford Science Publications. The Clarendon Press Oxford University Press, New York, 1988.

[Dev86] Luc Devroye. *Nonuniform random variate generation*. Springer-Verlag, New York, 1986.

[Doo84] J. L. Doob. *Classical potential theory and its probabilistic counterpart*, volume 262 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, New York, 1984.

[FC78] J. L. Folks and R. S. Chhikara. The inverse Gaussian distribution and its statistical application—a review. *J. Roy. Statist. Soc. Ser. B*, 40(3):263–289, 1978. With discussion.

[FH65] R. P. Feynman and A. R. Hibbs. *Quantum mechanics and path integrals*. International series in pure and applied physics. McGraw-Hill, Boston, 1965.

[Fre85] Mark Freidlin. *Functional integration and partial differential equations*, volume 109 of *Annals of Mathematics Studies*. Princeton University Press, Princeton, NJ, 1985.

[Gir60] I. V. Girsanov. On transforming a class of stochastic processes by absolutely continuous substitution of measures. *Teor. Verojatnost. i Primenen.*, 5:314–330, 1960.

[Gob00] Emmanuel Gobet. Weak approximation of killed diffusion using Euler schemes. *Stochastic Process. Appl.*, 87(2):167–197, 2000.

[Gob01] Emmanuel Gobet. Euler schemes and half-space approximation for the simulation of diffusion in a domain. Technical report, Ecole polytechnique: Centre de mathématiques appliquées, 2001.

[GS99] Maria Teresa Giraudo and Laura Sacerdote. An improved technique for the simulation of first passage times for diffusion processes. *Comm. Statist. Simulation Comput.*, 28(4):1135–1163, 1999.

[GSZ01] Maria Teresa Giraudo, Laura Sacerdote, and Cristina Zucca. A Monte Carlo method for the simulation of first passage times of diffusion processes. *Methodol. Comput. Appl. Probab.*, 3(2):215–231, 2001.

[GVL83] Gene H. Golub and Charles F. Van Loan. *Matrix computations*, volume 3 of *Johns Hopkins Series in the Mathematical Sciences*. Johns Hopkins University Press, Baltimore, MD, 1983.

[Hau00] Erika Hausenblas. Monte Carlo simulation of killed diffusion. *Monte Carlo Methods Appl.*, 6(4):263–295, 2000.

[HMG03] Chi-Ok Hwang, Michael Mascagni, and James A. Given. A Feynman–Kac path-integral implementation for Poisson's equation using an h-conditioned Green's function. *Math. Comput. Simul.*, 62(3-6):347–355, 2003.

[IW89] Nobuyuki Ikeda and Shinzo Watanabe. *Stochastic differential equations and diffusion processes*, volume 24 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam, second edition, 1989.

[JL00] Kalvis M. Jansons and G. D. Lythe. Efficient numerical solution of stochastic differential equations using exponential timestepping. *J. Statist. Phys.*, 100(5-6):1097–1109, 2000.

[JL03] Kalvis M. Jansons and G. D. Lythe. Exponential timestepping with boundary test for stochastic differential equations. *SIAM J.Sci.Comput.*, 24(5):1809–1822, 2003.

[KP92] Peter E. Kloeden and Eckhard Platen. *Numerical solution of stochastic differential equations*, volume 23 of *Applications of Mathematics*. Springer-Verlag, Berlin, 1992.

[KS73] Maurice G. Kendall and Alan Stuart. *The advanced theory of statistics. Vol. 2.* Hafner Publishing Co., New York, third edition, 1973. Inference and relationship.

[LS89] H. R. Lerche and D. Siegmund. Approximate exit probabilities for a Brownian bridge on a short time interval, and applications. *Adv. in Appl. Probab.*, 21(1):1–19, 1989.

[Mad02] Neal Madras. *Lectures on Monte Carlo methods*, volume 16 of *Fields Institute Monographs*. American Mathematical Society, Providence, RI, 2002.

[Man99] R. Mannella. Absorbing boundaries and optimal stopping in a stochastic differential equation. *Phys. Lett. A*, 254(5):257–262, 1999.

[Mar53] Gisir̄o Maruyama. Markov processes and stochastic equations. *Nat. Sci. Rep. Ochanomizu Univ.*, 4:40–43, 1953.

[Mar55] Gisir̄o Maruyama. Continuous Markov processes and stochastic equations. *Rend. Circ. Mat. Palermo (2)*, 4:48–90, 1955.

[Mas99] Michael Mascagni. SPRNG: a scalable library for pseudorandom number generation. In *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing 1999 (San Antonio, TX)*, page 10, Philadelphia, PA, 1999. SIAM.

[Mil74] G. N. Milstein. Approximate integration of stochastic differential equations. *Teor. Verojatnost. i Primenen.*, 19:583–588, 1974.

[Mil78] G. N. Milstein. A method with second order accuracy for the integration of stochastic differential equations. *Teor. Verojatnost. i Primenen.*, 23(2):414–419, 1978.

[Mil95a] G. N. Milstein. *Numerical integration of stochastic differential equations*, volume 313 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1995. Translated and revised from the 1988 Russian original.

[Mil95b] G. N. Milstein. Solution of the first boundary value problem for equations of parabolic type by means of the integration of stochastic differential equations. *Teor. Veroyatnost. i Primenen.*, 40(3):657–665, 1995.

[Mil96] G. N. Milstein. The simulation of phase trajectories of a diffusion process in a bounded domain. *Stochastics Stochastics Rep.*, 56(1-2):103–125, 1996.

[Mil97] G. N. Milstein. Weak approximation of a diffusion process in a bounded domain. *Stochastics Stochastics Rep.*, 62(1-2):147–200, 1997.

[Mil98] G. N. Milstein. On the mean-square approximation of a diffusion process in a bounded domain. *Stochastics Stochastics Rep.*, 64(3-4):211–233, 1998.

[Mil01] G. N. Milstein. The simplest random walk for general dirichlet problems. Talk at Stochastic Numerics Conference, ETH Zürich, Switzerland, 2001.

[Mos86] Lincoln E. Moses. *Think and Explain with Statistics*. Addison-Wesley Publishing, Reading, Mass., 1986.

[MR93] G. N. Milstein and N. F. Rybkina. An algorithm for random walks over small ellipsoids for solving the general Dirichlet problem. *Comput. Math. Math. Phys.*, 33(5):631–647, 1993.

[MSH76] John R. Michael, William R. Schucany, and Roy W. Haas. Generating random variates using transformations with multiple roots. *Am. Stat.*, 30:88–90, 1976.

[MT99] G. N. Milstein and M. V. Tretyakov. Simulation of a space-time bounded diffusion. *Ann. Appl. Probab.*, 9(3):732–779, 1999.

[MT02] G. N. Milstein and M. V. Tretyakov. The simplest random walks for the Dirichlet problem. *Teor. Veroyatnost. i Primenen.*, 47(1):39–58, 2002.

[Mul56] Mervin E. Muller. Some continuous Monte Carlo methods for the Dirichlet problem. *Ann. Math. Statist.*, 27:569–589, 1956.

[Øks03] Bernt Øksendal. *Stochastic differential equations*. Universitext. Springer-Verlag, Berlin, sixth edition, 2003. An introduction with applications.

[PBM86] A. P. Prudnikov, Yu. A. Brychkov, and O. I. Marichev. *Integrals and series. Vol. 1*. Gordon & Breach Science Publishers, New York, 1986. Elementary functions, Translated from the Russian and with a preface by N. M. Queen.

[Pet98] W. P. Petersen. A general implicit splitting for stabilizing numerical simulations of Itô stochastic differential equations. *SIAM J. Numer. Anal.*, 35(4):1439–1451 (electronic), 1998.

[PTVF02] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C++. The art of scientific computing.2nd ed.* Cambridge: Cambridge University Press. xxviii, 1002 p. , 2002.

[PW82] Eckhard Platen and Wolfgang Wagner. On a Taylor formula for a class of Itô processes. *Probab. Math. Statist.*, 3(1):37–51 (1983), 1982.

[RW68] L. K. Roy and M. T. Wasan. The first passage time distribution of Brownian motion with positive drift. *Math. Biosci.*, 3:191–204, 1968.

[RW00a] L. C. G. Rogers and David Williams. *Diffusions, Markov processes, and martingales. Vol. 1*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, 2000. Foundations, Reprint of the second (1994) edition.

[RW00b] L. C. G. Rogers and David Williams. *Diffusions, Markov processes, and martingales. Vol. 2*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, 2000. Itô calculus, Reprint of the second (1994) edition.

[RY91] Daniel Revuz and Marc Yor. *Continuous martingales and Brownian motion*, volume 293 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1991.

[Sab91] Karl K. Sabelfeld. *Monte Carlo methods in boundary value problems*. Springer Series in Computational Physics. Springer-Verlag, Berlin, 1991. Translated from the Russian.

[Shu68] Jonathan Shuster. On the inverse Gaussian distribution function. *J. Amer. Statist. Assoc.*, 63:1514–1516, 1968.

[Ste80] G. W. Stewart. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM J. Numer. Anal.*, 17(3):403–409 (loose microfiche suppl.), 1980.

[Szn98] Alain-Sol Sznitman. *Brownian motion, obstacles and random media*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 1998.

[Vel98] Todd L. Veldhuizen. Arrays in blitz++. In *Proceedings of the 2nd International Scientific Computing in Object-Oriented Parallel Environments (ISCOPE'98)*, Lecture Notes in Computer Science. Springer-Verlag, 1998.

[Was68] M. T. Wasan. On an inverse Gaussian process. *Skand. Aktuarietidskr.*, 1968:69–96, 1968.

[Was69] M. T. Wasan. *First passage time distribution of Brownian motion with positive drift (inverse Gaussian distribution)*. Queen's Papers in Pure and Applied Mathematics, No. 19. Queen's University, Kingston, Ont., 1969.

[Wen79] Alexander D. Wentzell. *Theorie zufälliger Prozesse. In dtsch. Spr. hrsg. v. Hans Jürgen Engelbert u. Jürgen Groh. Dtsch. Übers. v. Jürgen Groh. Lizenzausg.* Lehrbücher und Monographien aus dem Gebiete der exakten Wissenschaften: Mathematische Reihe, Bd. 65. Basel - Boston - Stuttgart: Birkhäuser Verlag. X, 253 S. SFr. 69.00 , 1979.

[WP78] Wolfgang Wagner and Eckhard Platen. Approximation of Ito integral equations. (Preprint). Technical report, Akademie der Wissenschaften der DDR, Zentralinstitut für Mathematik und Mechanik. 27 S. , 1978.

Curriculum Vitae

25.May 1974	Born in Zürich, Switzerland
1981–1987	Primary School in Zürich
1987–1994	*Realgymnasium Rämibühl* in Zürich
Feb 1992–Feb 1993	Exchange year in Costa Rica, fifth year in *Colegio Santa Ana* in Liberia
1994	Matura typus B (latin)
1995–1997	Study of electrical engineering at ETH Zürich
1997–2000	Study of computational science at ETH Zürich
Nov 1999–April 2000	Diploma thesis at university Freiburg i.Br. (Germany) with Prof. Dr. Jan Korvink (chair for simulation at *Institut für Mikrosystemtechnik*)
April 2000	Diploma in computational science (*Rechnergestützte Wissenschaften*)
April 2000	Employed as research assistant at University of Freiburg i.Br.
since May 2000	Research and teaching assistant at Seminar for Applied Mathematics (ETH Zürich)