

Numerical Finance I

Prof. Dr. Karsten Urban

Universität Ulm
Abteilung Numerik
Sommersemester 2003

Contents

Preface	3
1 Introduction	4
2 Numerical Generation of Random Numbers	7
2.1 Congruence Methods	8
2.2 Frequency and Gap Tests	10
2.2.1 The χ^2 -Test	11
2.2.2 Gaps	12
2.3 Discrepancy	12
2.4 Transformed Random Variables	15
2.4.1 Inversion	16
2.4.2 Transformation of Random Variables	16
2.4.3 Normally Distributed Random Variables	18
3 Numerical Cubature and Monte-Carlo Methods	20
3.1 Product Formulas (are useless here)	21
3.2 Monte-Carlo Methods	23
3.3 Quasi-Monte-Carlo Methods	24
3.4 The Smolyak Method	28
4 Numerical Computation of European Options	37
4.1 Option Pricing: A Very Short Introduction	37
4.2 Binomial Methods	39
4.3 Finite Difference Methods	43
4.4 Discretization in Time	48
4.5 Stochastic Differential Equations (SDE)	51
4.6 Computation of Moments	56

4.6.1	Monte-Carlo Methods	56
4.6.2	A Deterministic Approach	58
5	Elliptic Partial Differential Equations	60
5.1	Finite Difference Methods	61
5.2	Categories of Second Order PDEs	63
5.3	Variational Formulation of Elliptic PDEs	64
5.4	Ritz-Galerkin methods	66
5.5	Some Simple Finite Elements	68
5.6	Approximation Results	73
5.7	Example: 1D Finite Element Discretization for the Black-Scholes Equation	76

Preface

This is a slightly extended version of my manuscript to the lecture *Numerical Finance I* that was given at the University Ulm in the summer term 2003. It was a lecture (in German) of 2 hours per week. The exercises (also 2 hours per week) were given by Dipl.Math. oec. Michael Lehn.

The aim of this manuscript is mainly to give those students a chance to attend *Numerical Finance II* in the winter term 2003/04, who did not attend the first part of the lecture. In particular, it should help the students of the new Master programme *Finance* for whom *Numerical Finance II* is an obligatory lecture. Since this Master programme is in English, this manuscript is in English too.

This manuscript is far from being complete and the lecture was not much more than a first iteration through the various fields of this topic. Hence, I am grateful for any kind of comments, criticism or corrections. Additional material can be found on the webpage of the lecture

www.mathematik.uni-ulm.de/numerik/urban/lehre/numfin1_03.html

Finally, I wish to thank Michael Lehn for his very good work as assistant to the lecture and for many very valuable comments. I am particularly grateful to my colleague Prof. Dr. Rüdiger Kiesel for providing several examples from mathematical finance. My students Timo Tonn and Johannes Ruf made several helpful remarks which I kindly acknowledge. Petra Hildebrand fought with my hand-writing and typed the first version of this manuscript in L^AT_EX.

Chapter 1

Introduction

At a first glance, one may ask why numerical methods in finance are needed at all since highly sophisticated software packages are available that do all computations needed. There are at least two main fields in finance in which numerical methods are needed, namely:

- Calculation of prices, values etc. with a given (often complicated) formula on a computer; as an example let us mention that ‘over-the-counter’ (OTC) derivatives are tailor-made for certain specific applications. Hence, there no standard software can be used in this case.
- Computation of an approximate solution to problems that do not have a closed formula such as
 - certain linear or non-linear systems of equations,
 - ordinary and/or partial differential equations,
 - problems of optimization,
 - differential-algebraic equations (DAEs),
 - variational inequations,
 - and so on.

In all these possible applications, the user (the person that makes use of the results of a numerical computation) in particular is interested in:

- Exactness and reliability of the results:
If a result of a numerical computation is the basis for further decisions,

the user has to know how ‘good’ this result is, namely how close the numerical approximation is to ‘the’ ‘exact’ solution. A precise error statement (e.g. an estimate of the relative error of the desired quantity) is a necessary information for the further use of the numerical results.

- **Stability:**
Often a numerical computation is based on input data. Not only in applications from finance those input data are not available at all or are at least subject to stochastic influences. This means, one cannot expect to have exact input data, they will in general contain errors. Consequently, the numerical computation must not be sensitive to small errors in the input data in the sense that small errors in the input cause large errors in the output. This topic is known in Numerical Analysis as *stability*.
- **Efficiency:**
In a large range of applications one needs a numerical computation not sometime but within a short period of time. There are even applications in which the computation has to be performed in *real time*, i.e., in the same time, the process to be simulated takes in reality. This demand is only achievable if the numerical method used is highly efficient.

From this different demands, particular numerical questions and problems arise, namely:

- **Reliability of computed approximations:**
In order to give a precise error estimate for the quantity under consideration, an error analysis of the corresponding numerical methods and algorithms is required. This in particular leads to the mathematical field of *Approximation Theory*.
- **Stability of the numerical methods:**
The study of the stability of numerical methods is an own field within *Numerical Analysis*, sometimes also called perturbation theory.
- **Efficiency:**
This topic is especially relevant for high-dimensional, highly complex, or time-critical problems (e.g. problems of control, real time problems). The study of these kind of questions is called *Complexity Theory* which is also a well-established field within Numerical Analysis.

From the above introduction, we see that a good knowledge and the correct use of numerical tools is very important for the user. In particular, also a practitioner should know which numerical tool is useful for which kind of problem. An incorrect use may not only yield to extremely large computing times (which might cause that the numerical results are worthless) but the numerical simulation may also have nothing to do with the underlying problem (which means that the numerical results are wrong).

Chapter 2

Numerical Generation of Random Numbers

The modelling of financial processes often requires also to take into account stochastic influences, e.g., the seemingly random development of a stock price in the future. In order to simulate a stochastic behavior within a numerical simulation, one has to realize randomness on a computer, i.e., the generation of random numbers.

Possible applications (among others) include:

- Numerical realization and simulation of stochastic processes. This field in fact has huge area of applications far beyond finance. Let us just mention traffic simulation, medicine, science and engineering.
- Monte–Carlo–Methods. We will come to these methods for a specific application later, they require in particular the availability of random numbers.

The main problem is that a computer is a *deterministic* calculating machine. I.e., any algorithm, any process on the computer is deterministic. Thus, the nature of a computer is in contrast to the generation of *random* numbers. Because of this, one usually considers *pseudo random numbers*, i.e., numbers that are generated in a deterministic way but that reflect a random behavior in a ‘good’ way. Moreover, often random numbers mimicking a given distribution are required. First we analyze the generators of pseudo random numbers for uniformly distributed numbers. Other distributions will then be realized with the aid of suitable transformations.

In this chapter, we mainly follow [8].

2.1 Congruence Methods

We start with the maybe most simple family of methods, the so-called congruence methods.

Definition 2.1.1 For $M \in \mathbb{N}$ set $Z_M := \{0, \dots, M - 1\}$. A congruence method of first order constructed by an initial value $y_0 \in Z_M$ with a function

$$f : Z_M \rightarrow Z \tag{2.1}$$

is a sequence $(y_n)_{n \in \mathbb{N}} \subset Z_M$ defined by the rule

$$y_{n+1} := f(y_n) \pmod{M} . \tag{2.2}$$

This method is called linear, if f is affine-linear, i.e., if there exist $a, b \in \mathbb{Z}$ such that $f(x) = ax + b$. \square

For the congruence method we can now easily prove the following properties.

Theorem 2.1.2 Let the sequence $(y_n)_{n \in \mathbb{N}}$ be generated by the congruence method. Then, the following statements hold:

- (a) The created sequence $(y_n)_{n \in \mathbb{N}}$ has a period with the maximal length M .
- (b) For the linear congruence method with $b = 0$ (the so called Prime-Modulo-Generator) $y_m = 0$ must be excluded.

Proof:

- (a) Because of $\#Z_M = M$ there exist at least two identical elements in $\{y_0, \dots, y_M\}$, i.e., there exist indices $i \in \{0, \dots, M - 1\}$, $p \in \{1, \dots, M\}$ such that $y_i, y_{i+p} \in \{y_0, \dots, y_M\}$ and therefore $y_i = y_{i+np}$ for all $n \in \mathbb{N}$.
- (b) For $y_n = 0$ and $b = 0$ we obtain

$$f(y_n) = ay_n + b = 0 = y_n ,$$

so that $y_m = y_n$ for all $m \geq n$, i.e., we obtain a constant sequence, which of course is non-random. \square

The periodicity of the generated sequence is of course a serious drawback of the congruence method. Thus, in practice M should be chosen as large as possible in order to obtain a maximal length of the period. However, Theorem 2.1.2 (a) gives only an *upper bound* for the length of the period, in practice it could even be much smaller as we have seen in (b). The next result gives a precise statement for the length of the period of the Prime-Modulo-Generator.

Theorem 2.1.3 *Let M be a prime number. Then the Prime-Modulo-Generator has the smallest period $M - 1$ if a is primitive root of M , i.e., if*

$$a^i - 1 \begin{cases} \not\equiv 0 \pmod{M}, & \text{if } 1 \leq i < M - 1, \\ \equiv 0 \pmod{M}, & \text{if } i = M - 1. \end{cases}$$

Proof: By Theorem 2.1.2 (b) we have $y_0 \neq 0$. For the sequence $(z_n)_{n \in \mathbb{N}}$ with $z_0 := y_0$, $z_n := f(z_{n-1}) = az_{n-1}$ we obviously have $z_n = a^n z_0$. Thus $y_n = z_n \pmod{M} = y_0$ holds if and only if $a^n \equiv 1 \pmod{M}$. Thus, by assumption we have $n = M - 1$ which is the smallest period. \square

Example 2.1.4 *We consider the case $M = 11$ with the choices of the parameters $a = y_0 = 5$. Note that in this case we have $a^5 = 3125 = 11 \times 284 + 1$, which implies $a^5 \pmod{11} \equiv 1$, i.e., we expect periodic length equals to 5. In fact:*

$$\begin{aligned} y_1 &= 25 \pmod{11} = 3 \\ y_2 &= 15 \pmod{11} = 4 \\ y_3 &= 20 \pmod{11} = 9 \\ y_4 &= 45 \pmod{11} = 1 \\ y_5 &= 5 \pmod{11} = 5 = y_0 \end{aligned}$$

Example 2.1.5 *The generator RANDU which is often used in mathematical software packages is a Prime-Modulo-Generator with $a = 2^{16} + 3$ and $M = 2^{31} - 1$.*

Example 2.1.6 *An example of a non-linear congruence method is the inverse congruence method, where*

$$f(x) = a\bar{x} + b, \quad a, b \in \mathbb{Z}_M, M \text{ prime}$$

is used and \bar{x} is defined for a given $x \in \mathbb{Z}_M$ as

$$\begin{cases} \bar{x} = 0, & \text{if } x = 0, \\ x\bar{x} \equiv 1 \pmod{M}, & \text{else.} \end{cases}$$

Obviously, the calculation of \bar{x} is the most expensive part from the numerical point of view. This can e.g. be done with the Euclidean Algorithm (see exercises).

If the length of the period is M , the calculated pseudo random numbers are obviously uniformly distributed. If they are normalized through $\frac{y_i}{M}$ on the unit interval $[0, 1]$, they can be subjected to statistical tests in order to check if the desired distribution is in fact matched. We will describe this in the next section.

2.2 Frequency and Gap Tests

Once a sequence of random numbers is generated, one of course wants to check if this sequence is of the desired distribution. For the uniform distribution one may look at a graphical visualization, where each random number within an interval is plot with a different vertical coordinate. To be precise, let us assume that we have generated N random numbers $\{x_1, \dots, x_N\}$ in $[0, 1]$. Then we display the following points $(x_i, i/N)$ for $i = 1, \dots, N$ in a 2d-graph. Such a visualization is shown in Figure 2.1. Even though

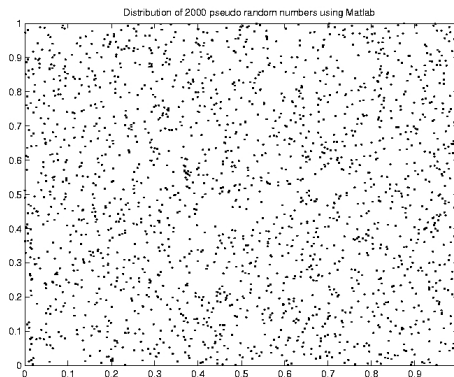


Figure 2.1: Visualization for the random number generator of MATLAB.

this graphical visualization gives a first idea, it clearly has a number of serious drawbacks. First of all, it is more or less restricted to the uniform distribution. Moreover, and more seriously, it does not give any quantitative information. Thus, we describe in this section how standard statistical

tests can be used in order to investigate the quality of generated sequences of pseudo random numbers.

2.2.1 The χ^2 -Test

Let us briefly recall the well-known χ^2 -test:

Divide $[0, 1]$ into $m + 1$ subintervals $J_i = [x_i, x_{i+1})$, $0 = x_0 < x_1 < \dots < x_{m+1} = 1$ and define the quantity

$$B_i := \#t_\nu \text{ in the interval } J_i$$

(t_ν are the pseudo random numbers). For the test to be meaningful every B_i should be at least of the size 5 to 10. Further let E_i be the expected quantity of t_ν in J_i and define

$$\chi^2 := \sum_{i=0}^m \frac{(B_i - E_i)^2}{E_i} .$$

Then one has

$$P(\chi^2 > x \mid \text{model is correct}) = \int_x^\infty f_m(x) dx =: F_m(x),$$

where

$$f_m(x) = \left(\frac{x}{2}\right)^{m/2} \frac{e^{-x/2}}{x \Gamma\left(\frac{m}{2}\right)}, \quad \Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt .$$

A significant test results, if this test is realized for a large number (say N) of realizations of a random number generator. Then the quantity p_i of the calculated χ^2 -values in

$$\left[i - \frac{1}{2}, i + \frac{1}{2}\right), \quad i = 1, 2, \dots$$

are counted. If the points $(i, p_i/N)$ are “close” to the probability density-function f_m , the random number generator has passed the χ^2 test.

Example 2.2.1 *Figure 2.2 shows the result of a χ^2 -test for the random number generator of MATLAB. The code for the χ^2 -test is also written in MATLAB and can be downloaded from the web-page of the lecture.*

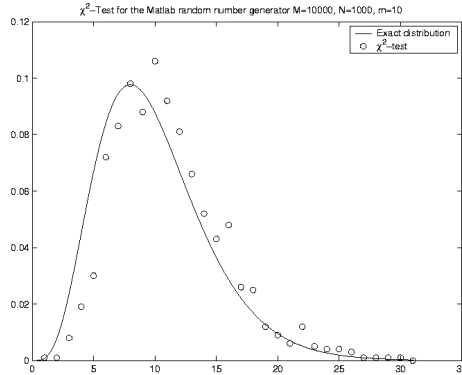


Figure 2.2: A χ^2 -test for the random number generator of MATLAB. Obviously, the test was successful.

2.2.2 Gaps

Definition 2.2.2 For a given interval $J \subset [0, 1]$ a sequence $(t_n)_{n \in \mathbb{N}_0}$ is said to have a gap of length k , if there exists some $n \in \mathbb{N}_0$ such that $t_n, \dots, t_{n+k-1} \notin J$, but $t_{n+k} \in J$. \square

For a corresponding test choose $h \in \mathbb{N}$, and count the number of gaps of length $0, 1, \dots, h-1, h$. On this sequence of pseudo random numbers, the χ^2 -test is applied.

For further information on random number generation and corresponding tests, we refer to [5].

2.3 Discrepancy

We have seen statistical tests to check the distribution of pseudo random numbers. We have concentrated on the uniform distribution. So far, we do not have a *measure* how good a uniform distribution is matched. We will now introduce such a measure.

Definition 2.3.1 Let $X := \{x_1, \dots, x_N\} \subset [0, 1]^m$ be a sequence of normalized pseudo random numbers.

(a) Let \mathcal{Q} be the set of all quads in $[0, 1]^m$. Then, we call

$$D(X) := \sup_{Q \in \mathcal{Q}} \left| \frac{\#\{x_i \in X : x_i \in Q\}}{\#X} - \text{vol}(Q) \right|$$

the discrepancy of X .

(b) For $X = \{x_1, \dots, x_M\}$, $M \geq N$ we also use the abbreviation

$$D_N := D(\{x_1, \dots, x_N\}).$$

If $\lim_{N \rightarrow \infty} D_N = 0$, then we say that X consists of uniformly distributed points. \square

The idea behind the latter definition is that for a set of uniformly distributed points the portion of those points lying in a quad Q should at least almost correspond to the volume of Q . Of course the quantity $D(X)$ is not so easy to compute since the determination of the supremum over all quads might be a delicate and in particular expensive task. Thus, one also considers the following measure.

Definition 2.3.2 Let $Q^* = \prod_{i=1}^m [0, y_i)$, $0 < y_i \leq 1$ be a quad with one corner in 0 and denote by \mathcal{Q}^* the set of all these quads. Then, the quantity

$$D^*(X) := \sup_{Q^* \in \mathcal{Q}^*} \left| \frac{\#\{x_i \in X : x_i \in Q^*\}}{\#X} - \text{vol}(Q^*) \right|$$

is called star discrepancy of X . \square

Obviously, the star discrepancy is easier to access. The next result shows that it is in fact an approximation of the discrepancy.

Proposition 2.3.3 The following estimates hold

- (a) $0 \leq D_N \leq 1$,
- (b) $D_N^* \leq D_N \leq 2^m D_N^*$ at least for $m \leq 2$,
- (c) $D_N^* \geq \frac{1}{2N}$ for $m = 1$.

Proof: We leave the proof as an easy exercise. \square

In Definition 2.3.1 (b) we have just asked that D_N tends to zero for $N \rightarrow \infty$. This is a statement of pure asymptotic character. In practice one is of course also interested that already a moderate number of pseudo random numbers is almost uniformly distributed. Hence, one is interested how fast D_N tends to zero, i.e., what is the rate of decay. This is reflected by the following definition.

Definition 2.3.4 A sequence $(x_k)_{k \in \mathbb{N}}$ is called of low discrepancy if

$$D_N \leq C_m \frac{(\log N)}{N} \quad (2.3)$$

with a constant $0 \leq C_m < \infty$ independent of N . A deterministic sequence of numbers is called a set of pseudo random numbers if (2.3) holds. \square

Some Examples

Example 2.3.5 For $m = 1$ and $x_i := \frac{2i-1}{2N}, i = 1, \dots, N$ we obtain $D_N^* = \frac{1}{2N}$. In fact, let $Q^* = [0, y)$, $0 < y \leq 1$ so that $\text{vol}(Q^*) = y$ and

$$\begin{aligned} x_i \in Q^* &\iff \frac{2i-1}{2N} < y &\iff 2i-1 \leq 2Ny \\ & &\iff i \leq \frac{2Ny+1}{2}. \end{aligned}$$

Hence, we have

$$D^*(X) \leq \sup_{0 < y \leq 1} \left\{ \underbrace{\left\lfloor \frac{2Ny+1}{2} \right\rfloor \frac{1}{N} - y}_{\leq \frac{2Ny+1}{2N} - \frac{2Ny}{2N} = \frac{1}{2N}} \right\} = \frac{1}{2N}.$$

Choosing the special case $y = x_i$ shows that $D^*(X) = \frac{1}{2N}$.

By Proposition 2.3.3 (c) this is optimal. On the other hand, the sequence (x_i) has to be computed for every N from scratch which of course is highly inefficient if N grows. Hence it would be better if the numbers could be set in a dynamical way. The next example shows one way to achieve this.

Definition 2.3.6 Let $b \geq 2$ be an integer and for $i \in \mathbb{N}$ consider the b -adic representation of i to the base b , namely

$$i = \sum_{k=0}^j d_k b^k, \quad d_k \in \{0, 1, \dots, b-1\}.$$

Then, the mapping ϕ_b defined by

$$\phi_b(i) := \sum_{k=0}^j d_k b^{-k-1}$$

is called radix-inverse function. \square

The radix-inverse function can be interpreted as a ‘reflection at the radix point’, i.e., $i \mapsto x \in \mathbb{Q}$, $0 < x < 1$. If the number of digits j in i is increased, the highest power of b is increased which in turns increases the fineness of the rational numbers i , i.e., new numbers are dynamically inserted. Combining different radix-inverse functions yields the following sequence.

Definition 2.3.7 Assume that p_1, \dots, p_m are co-prime integers. Then, the vectors

$$x_i := (\phi_{p_1}(i), \dots, \phi_{p_m}(i)) \in \mathbb{R}^m, \quad i = 1, 2, \dots$$

are called Halton-Folge.

As a particular example, the sequence $x_i := \phi_2(i)$ is called *Van der Corput sequence*.

2.4 Transformed Random Variables

So far we have considered “only” uniformly distributed pseudo random numbers. A (very) simple method to construct an approximately normally distributed sequence of random numbers from a uniformly distributed sequence $U_i \sim \mathcal{U}[0, 1]$ is the following

$$X := \sum_{i=1}^{12} U_i - 6.$$

One easily obtains that approximately $X \sim \mathcal{N}(0, 1)$. Obviously, this is not a very sophisticated method and as we shall see next, transformation methods are in fact much better.

2.4.1 Inversion

The quite simple idea of this approach is to invert the particular distribution function.

Theorem 2.4.1 *Let $U \sim \mathcal{U}[0, 1]$ and F be a uniformly continuous, strictly monotone distribution function. Then there exists the inverse $F^{-1} : [0, 1] \rightarrow \mathbb{R}$ and $F^{-1}(U)$ is distributed according to F .*

Proof: It is easily seen that

$$\begin{aligned} U \sim \mathcal{U}[0, 1] &\iff P(U \leq \xi) = \xi \text{ for } 0 \leq \xi \leq 1 \\ &\iff P(F^{-1}(U) \leq x) = P(U \leq F(x)) = F(x). \quad \square \end{aligned}$$

Remark 2.4.2 *The statement of Theorem 2.4.1 also applies for more general distribution functions.*

Even though this straightforward approach seems to yield the desired result, there is a serious drawback. E.g. for the normal distribution there is no Gaussian error-integral, in particular neither for $F(x)$ nor for $f = F'(x)$ a closed formula exists. Thus one has to solve the non-linear problem $f(x) = u$ numerically e.g. by an iterative method (bisection, secant method, Newton). Moreover, it can easily be seen that for $u \approx 1$ small modifications in u cause large modifications in x , i.e., instabilities occur. As an alternative, one may compute a numerical approximation $G(u) \approx F^{-1}(u)$ e.g. by *rational approximation* in order to reflect the poles correctly.

2.4.2 Transformation of Random Variables

As an alternative, we now consider transformation methods. The key result behind this approach is the following theorem. For the sake of simplicity, we will first state and prove it in the 1D case.

Theorem 2.4.3 *Let X be a random variable with density function $f(x)$ and probability distribution function $F(x)$. Further let $h : S \rightarrow B$, $S, B \subset \mathbb{R}$, where S denotes the support of f , i.e.,*

$$S := \text{supp } f := \overline{\{x \in \mathbb{R} : f(x) \neq 0\}}.$$

If h is strictly monotonously increasing, we have

(a) $Y := h(X)$ is a random variable with distribution function $F(h^{-1}(Y))$;

(b) If h^{-1} is absolutely continuous, then

$$f(h^{-1}(y)) \left| \frac{dh^{-1}(y)}{dy} \right| \quad (2.4)$$

is the density function of $h(X)$ for almost all y .

Proof:

(a) Because h is strictly monotonously increasing, this also holds for the inverse and we obtain for distribution of Y that $P(Y \leq y) = P(h(X) \leq y) = P(X \leq h^{-1}(y)) = F(h^{-1}(y))$.

(b) Because h^{-1} is absolutely continuous, the density of $Y = h(X)$ is the distribution function f . By the chain rule, we obtain

$$\frac{d}{dy} F(h^{-1}(y)) = \underbrace{F'(h^{-1}(y))}_{=f(h^{-1}(y))} \left(\frac{d}{dy} h^{-1}(y) \right)$$

and the absolute value in (2.4) is necessary for the correct reproduction of the sign of h' , see [8]. \square

Now we apply Theorem 2.4.3 to a given sequence of random numbers $X \sim \mathcal{U}[0, 1]$. Let f be the corresponding density function, i.e.,

$$f(x) := \begin{cases} 1, & \text{if } 0 \leq x \leq 1, \\ 0, & \text{else,} \end{cases}$$

i.e., $S = \text{supp } f = [0, 1]$. Assume, we are interested in a sequence of random numbers Y with density function $g(y)$. Thus, we need to define $h = g(y)$ in such a way that g coincides with the density in (2.4), i.e.,

$$\left| \frac{dh^{-1}(y)}{dy} \right| = g(y)$$

so that $Y := h(X)$ is the desired sequence.

Example 2.4.4 (*Exponential distribution*) It is well-known that the density function is

$$g(y) = \begin{cases} \lambda e^{-\lambda y} & \text{for } y \geq 0, \\ 0 & \text{for } y < 0, \end{cases}$$

where λ is the free parameter. Thus, $B := [0, \infty) = \mathbb{R}_0^*$ and $S := [0, 1]$. Hence $h : S \rightarrow B$ is defined by $y := h(x) := -\frac{1}{\lambda} \log x$ and thus $h^{-1}(y) = e^{-\lambda y}$ for $y \geq 0$. Hence,

$$\underbrace{f(h^{-1}(y))}_{=1} \left| \frac{d}{dy} h^{-1}(y) \right| = |(-\lambda)e^{-\lambda y}| = g(y),$$

and $h^{-1} : B \rightarrow S$ (note that both sides are zero for $y < 0$). According to Theorem 2.4.3, we see that $h(X)$ is exponentially distributed.

Example 2.4.5 (Standard normal distribution) It is well-known that the distribution function is

$$g(y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right) \stackrel{!}{=} \left| \frac{d}{dy} h^{-1}(y) \right|,$$

where the latter equation is the one to be satisfied by Theorem 2.4.3. This is a differential equation for h^{-1} that does not have an analytical solution. Thus one has to resort to numerical solution methods.

Without proof, we quote from [8] the generalization of Theorem 2.4.3 to the multivariate case.

Theorem 2.4.6 Let X be a sequence of random variables on \mathbb{R}^n with density function $f(x) > 0$ on $S = \text{supp } f$. Moreover, assume that $h : S \rightarrow B$, $S, B \subset \mathbb{R}^n$ is explicitly invertible and $Y := h(X)$ is the transformed sequence. If h^{-1} is continuously differentiable on B , then Y has the density function

$$f(h^{-1}(y)) \left| \det \left(\frac{\partial x_i}{\partial y_j} \right)_{i,j} \right|, \quad y \in B,$$

where $x = h^{-1}(y)$ and $\left(\frac{\partial x_i}{\partial y_j} \right)_{i,j}$ is the Jacobi-Matrix of h^{-1} . \square

2.4.3 Normally Distributed Random Variables

Since normally distributed pseudo random variables are highly relevant in many applications, we give a corresponding number generator for this case here. We apply the above described transformation method in order to generate normally distributed random numbers.

Method of Box-Muller (1952)

Define the function $h : [0, 1]^2 \rightarrow \mathbb{R}^2$ by

$$\begin{aligned} h_1(x_1, x_2) &:= \sqrt{-2 \log x_1} \cos(2\pi x_2) = y_1 \\ h_2(x_1, x_2) &:= \sqrt{-2 \log x_1} \sin(2\pi x_2) = y_2. \end{aligned}$$

It is readily seen that

$$h^{-1}(y_1, y_2) = \begin{bmatrix} \exp \left\{ -\frac{1}{2}(y_1^2 + y_2^2) \right\} \\ \frac{1}{2\pi} \arctan \frac{y_2}{y_1} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and the Jacobi-Matrix is given by

$$\mathcal{J} = \left(\frac{\partial x_i}{\partial y_j} \right)_{i,j} = \begin{bmatrix} (-y_1) \exp \left\{ -\frac{1}{2}(y_1^2 + y_2^2) \right\} & \frac{1}{2\pi} \frac{1}{1 + \frac{y_2^2}{y_1^2}} \cdot \left(-\frac{y_2}{y_1^2} \right) \\ (-y_2) \exp \left\{ -\frac{1}{2}(y_1^2 + y_2^2) \right\} & \frac{1}{2\pi} \frac{1}{1 + \frac{y_2^2}{y_1^2}} \cdot \frac{1}{y_1} \end{bmatrix}.$$

Hence, we obtain

$$\begin{aligned} \det \mathcal{J} &= \frac{1}{2\pi} \exp \left\{ -\frac{1}{2}(y_1^2 + y_2^2) \right\} \underbrace{\left[-y_1 \frac{1}{1 + \frac{y_2^2}{y_1^2}} \cdot \frac{1}{y_1} - y_2 \frac{1}{1 + \frac{y_2^2}{y_1^2}} \cdot \frac{y_2}{y_1^2} \right]}_{=-1} \\ &= - \left(\frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2}y_1^2 \right\} \right) \left(\frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2}y_2^2 \right\} \right), \end{aligned}$$

which is the density function of the standard normal distribution. Hence $h(X)$ is normally distributed. The corresponding algorithm then reads as follows.

Algorithm 2.4.7 (Box-Muller)

- (1) Generate two sequences $U_1 \sim \mathcal{U}[0, 1]$ and $U_2 \sim \mathcal{U}[0, 1]$;
- (2) Set $\Theta := 2\pi U_2$ and $\rho = \sqrt{-2 \log U_1}$;
- (3) Compute $Z_1 := \rho \cos \Theta$ and $Z_2 := \rho \sin \Theta$, which are random numbers according to the standard normal distribution. \square

A modification of this is the method of Masaglia, see [8, Chap. 2.3.2].

Chapter 3

Numerical Cubature and Monte-Carlo Methods

Many important applications of mathematical modelling in finance require the computation of integrals. These usually highly dimensional integrals are often so complex that this cannot be done analytically. Hence one has to resort to numerical methods. We start with an example that shows of which type these integrals might be. This also clearly shows the numerical challenges.

Example 3.0.1 (Mortgage-Backed Securities (MBS)) *MBS are a well-known and prominent example of asset-backed securities and widely used in the US. A MBS is a fixed-income security whose performance is related to a pool of customer mortgages. The bank, who is giving out a mortgage to a customer, faces its prepayment risk, e.g. the risk that the customer pays back his mortgage pre maturity (mostly to refinance in order to benefit from low interest rates). Thus there is only a supposed behavior that depends on external parameters. Hence a stochastic modelling is appropriate and required.*

The value of the bond coincides with the expected return, i.e., an expectation rate which mathematically is an integral. Let us illustrate the numerical problem for a concrete example. Let the duration of the bond be 30 years, i.e., 360 months. We denote by r_k the interest rate per month k , $1 \leq k \leq d = 360$. This is a random variable and we assume that it is log-normally distributed, i.e.,

$$r_k = r_{k-1} e^{\sigma Z_k - \frac{\sigma^2}{2}},$$

(Rendleman-Bartter interest-model). Here Z_k are independent and standard-normally distributed. Moreover, we denote the discounting coefficient by $d_k = \prod_{i=0}^{k-1} \frac{1}{1+r_i}$ so that the cash flow per month k is $M_k = C(1 - w_1) \dots (1 - w_{k-1})$, where C is the investment at the beginning of the contract. With the return rate w_k we obtain the current value of a MBS as

$$W = \sum_{k=1}^d d_k M_k.$$

A straightforward calculation for the expectation rate yields

$$\begin{aligned} \Rightarrow E(W) &= \underbrace{\int \dots \int}_d \sum_{k=1}^d M_k d_k(z_k) \varphi(z_1) \dots \varphi(z_d) dz_1 \dots dz_d \\ &= \int_{[0,1]^d} \tilde{W}(u) du, \quad u = (u_1, \dots, u_d), \quad u_k = \varphi(z_k), \end{aligned} \quad (3.1)$$

where $\tilde{W} : \mathbb{R}^d \rightarrow \mathbb{R}$ is a function determined by the first row by substitution and straightforward calculations. This shows that we have to compute (numerically) an integral in 360 space dimensions!

3.1 Product Formulas (are useless here)

In the latter example, we have to integrate a moderately complicated function over an easy domain (a cube), but in an extremely high-dimensional space. The simplest approach to do this is to use a 1D quadrature rule from any text book on numerical analysis and use a product formula built on this. We will now show that this would result in an extremely un-efficient method.

Let us again consider the above example, i.e., we want to calculate (3.1) numerically. To this end, we consider the following 1D quadrature formula for a function $f : [0, 1] \rightarrow \mathbb{R}$

$$Q_n[f] = \sum_{k=1}^n \gamma_k f(t_k),$$

i.e., a quadrature formula for the approximation of $\int_0^1 f(x) dx$ with weights $\gamma_i \in \mathbb{R}$ and quadrature points $t_i \in [0, 1]$. These formulas are now put together to obtain a cubature formula for a function $F : [0, 1]^d \rightarrow \mathbb{R}$

$$\begin{aligned}
I_d[F] &:= \int_{[0,1]^d} F(x_1, \dots, x_d) dx_1, \dots, dx_d \\
&= \underbrace{\int_0^1 \dots \int_0^1}_{d} F(x_1, \dots, x_d) dx_1, \dots, dx_d \\
&\approx \underbrace{\int_0^1 \dots \int_0^1}_{d-1} \sum_{i=1}^n \gamma_i F(t_i, x_2, \dots, x_d) dx_2, \dots, dx_d \\
&\approx \dots \approx \sum_{i_1=1}^n \dots \sum_{i_d=1}^n \gamma_{i_1} \dots \gamma_{i_d} F(t_{i_1}, \dots, t_{i_d}) =: Q_n^{[d]}[F]. \quad (3.2)
\end{aligned}$$

We now study the error of the latter product formula.

Definition 3.1.1 For a given 1D quadrature formula Q_n the term (3.2) is called product formula. The respective quadrature and cubature errors are

$$R_n[f] := I_1(f) - Q_n[f], \quad R_n^{[d]}[F] := I_d[F] - Q_n^{[d]}[F]. \quad (3.3)$$

Example 3.1.2 Before we study the error, let us just give a feeling for the complexity of the numerical problem. Let us assume that the parameter n reflects the exactness and also the complexity of $Q_n[f]$. For $n = 1$ (i.e., one quadrature point in $[0, 1]$, i.e., approximation by constants) one cannot expect a high order of exactness. Thus, let us consider the next higher case $n = 2$. For the above case of $d = 360$, this would amount to

$$2^{360} \approx 2.34 \cdot 10^{108}$$

evaluations of the function F to be integrated, which obviously is highly inefficient.

Moreover, there are also bad news concerning the behavior of the error. One expects of course that the error decreases for increasing n . In addition, the rate of convergence of the 1D method should be preserved in the multivariate case. This however is *not* true as the following result shows.

Theorem 3.1.3 *For every sequence of quadrature formulas Q_1, Q_2, \dots with*

$$|R_n[f]| \leq \frac{C_p}{n^p} \|f^{(p)}\|_\infty$$

and every sequence $(\delta_n)_{n \in \mathbb{N}}$, $\delta_n \searrow 0$ there exists a function $F : [0, 1]^d \rightarrow \mathbb{R}$ with $\|F\|_\infty \leq 1$, $\|F^{(p, \dots, p)}\|_\infty \leq \infty$ and

$$R_{n^d}^{(d)}[F] \geq \frac{\delta_n}{n^p}$$

for an infinite number of n . \square

We omit the proof of the latter theorem, refer e.g. to [6], and just remark that the counterexample is closely related to the famous example given by Runge, i.e., the function $\frac{1}{1+x^2} \in C^\infty$ for which the polynomial interpolation fails.

We conclude from the above discussion that for $N = n^d$ sampling points as for a product rule, the error is of the order $\mathcal{O}(N^{-\frac{p}{d}})$ which *cannot* be improved. Thus, product rules are useless for our kind of applications.

3.2 Monte-Carlo Methods

Monte-Carlo methods are nowadays widely used in stochastic modelling and simulation. We describe their use in numerical finance and also give precise error estimates in the sequel.

Example 3.2.1 *The midpoint rule reads*

$$Q_n^{\text{Mi}}[f] = \frac{1}{n} \sum_{i=1}^n f\left(\frac{2i-1}{2n}\right),$$

i.e., the quadrature points $t_i = \frac{2i-1}{2n}$ are uniformly distributed over the unit interval. For the corresponding product formula we would thus use the grid points

$$\left(\frac{2i_1-1}{2n}, \dots, \frac{2i_d-1}{2n}\right) \in [0, 1]^d, \quad i_j \in \{1, \dots, n\}.$$

In the above example, one would place n^d points uniformly over the unit cube $[0, 1]^d$. The idea is now to distribute these points in a random manner but in such a way that the random numbers are uniformly distributed. Hence, a method of the form

$$Q_n^{MC}[F] := \frac{1}{n} \sum_{i=1}^n F(x_i) , \quad n = 1, 2, \dots$$

with independent uniformly distributed random-numbers $x_i \in [0, 1]^d$ is called *Monte-Carlo-Method (MC)* for the approximation of $I_d[F]$.

Theorem 3.2.2 *If $I_d[F] < \infty$, $I_d[F^2] < \infty$, then we have*

$$P \left(\lim_{n \rightarrow \infty} R_n^{MC}[F] = 0 \right) = 1 .$$

Proof: The proof is a consequence of the *Strong Law of Large Numbers*. \square

Moreover, from of the *Central Limit Theorem* it follows with

$$\sigma_F^2 := -I_d[F]^2 + I_d[F^2]$$

that

$$\lim_{n \rightarrow \infty} P \left(\frac{\sigma_F}{\sqrt{n}} a < R_n^{MC}[F] < \frac{\sigma_F}{\sqrt{n}} b \right) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{t^2}{2}} dt,$$

i.e., we can expect an order of $\frac{1}{\sqrt{n}} = n^{-\frac{1}{2}}$. For high dimensional problems we have that $\frac{1}{2} \gg \frac{d}{2}$, i.e., the expected rate of convergence of a Monte-Carlo method is better compared with the above mentioned product formulas.

Remark 3.2.3 *With the use of so-called “antithetic variables” the variance σ_F can be reduced so that the method is (quantitatively) improved.*

3.3 Quasi-Monte-Carlo Methods

As in the chapter on random number generation we again focus the problem that a (pure) Monte-Carlo method cannot be used on a computer since we cannot realize randomness. Thus, random numbers are again replaced by pseudo random numbers which leads to a Quasi-Monte-Carlo method (QMC).

Definition 3.3.1 A cubature formula $Q_n[f] = \frac{1}{n} \sum_{i=1}^n f(x_i)$ for a sequence $X = (x_i)_{i=1, \dots, n}$ of pseudo random numbers is called Quasi-Monte-Carlo formula (QMC formula).

The ultimate goal for a QMC formula is of course to reach a best possible exactness with minimal amount of work. Moreover, the cubature points x_i should be placed asymptotically uniformly distributed with as few gaps as possible in order to represent the behavior of the function to be integrated in a best possible way.

We are now going to analyze this method rigorously in order to be able to compare the different methods. To this end, we have to introduce some notation and definitions.

Definition 3.3.2 For a function $f : [0, 1]^d \rightarrow \mathbb{R}$, $f \in C^1([0, 1]^d)$ we call

$$V^{(d)}(f) := \int_{[0,1]^d} |f^{(1, \dots, 1)}(u_1, \dots, u_d)| du, \quad u = (u_1, \dots, u_d)^T \quad (3.4)$$

the Vitali variance of f .

Definition 3.3.3 Set $J_k^{(d)} := \{(i_1, \dots, i_k) : 1 \leq i_1 < i_2 < \dots < i_k \leq d\}$ and for $I \in J_k^{(d)}$ let $f_I(u) := f(u)|_{u_{i_k}=1, k \notin I}$, i.e.,

$$f_I(u_1, \dots, u_d) := f(v_1, \dots, v_d), \quad \text{where } v_i := \begin{cases} u_i, & i \notin I, \\ 1, & \text{else.} \end{cases}$$

Then, the quantity

$$V(f) := \sum_{k=1}^d \sum_{I \in J_k^{(d)}} V^{(k)}(f_I) \quad (3.5)$$

is called Hardy-Krause variance of f .

Before we can give the error estimate, we introduce some notation: For $I = (i_1, \dots, i_k) \in J_k^{(d)}$ let

$$f^{(I)} := \frac{\partial}{\partial x_{i_1}} \dots \frac{\partial}{\partial x_{i_k}} f, \quad du_I := du_{i_k} \dots du_{i_1}, \quad \int_I f = \int_{t_{i_1}}^1 \dots \int_{t_{i_k}}^1 f.$$

Now we are ready to formulate and prove the main error estimate.

Theorem 3.3.4 (Koksma-Hlawka inequality) *Let R_n denote the error of a QMC formula with respect to $X = \{x_i\}_{i=1}^m \subset \mathbb{R}^d$. Then, we have*

$$|R_n[f]| \leq D^*(X) V(f) . \quad (3.6)$$

We need some preparations for the proof of the latter theorem.

Lemma 3.3.5 *For $t_1, \dots, t_d \in [0, 1]$ we have*

$$f(t_1, \dots, t_d) - f(1, \dots, 1) = \sum_{k=1}^d (-1)^k \sum_{I \in J_k^{(d)}} \int_I f_I^{(I)}(u) du_I. \quad (3.7)$$

Proof: By induction over d . For $d = 1$ the fundamental theorem of calculus gives

$$f(t) - f(1) = (-1) \int_t^1 f'(u) du ,$$

which coincides with (3.7), since here $J_1^{(1)} = \{1\}$ and $f_I \equiv f$, $f_I^{(I)} = f'$.

For $d > 1$, we split the sum on the right-hand side of (3.7) into 3 parts:

a) $k \geq 1$, $d \notin I$ $\Rightarrow v_d = 1$, hence this part reads

$$\sum_{k=1}^{d-1} (-1)^k \sum_{I \in J_k^{(d-1)}} \int_I f_I^{(I)}(u_1, \dots, u_{d-1}, 1) du_I$$

b) $k = 1$, $I = \{d\}$ $\Rightarrow v_d = u_d$, hence the sum becomes

$$(-1) \int_{t_d}^1 f^{(0, \dots, 0, 1)}(1, \dots, 1, u_d) du_d = f(1, \dots, 1, t_d) - f(1, \dots, 1)$$

c) $k > 1$, $d \in I$ $\Rightarrow v_d = u_d$, hence we have for this part

$$\begin{aligned}
& (-1) \sum_{k=1}^{d-1} (-1)^k \sum_{I \in J_k^{(d-1)}} \int_I \int_{t_d}^1 f_{I \cup \{d\}}^{(I \cup \{d\})}(u) du_d du_I \\
&= \sum_{k=1}^{d-1} (-1)^k \sum_{I \in J_k^{(d-1)}} \int_I f_I^{(I)}(u_1, \dots, u_{d-1}, t_d) du_I \\
&\quad - \sum_{k=1}^{d-1} (-1)^k \sum_{I \in J_k^{(d-1)}} \int_I f_I^{(I)}(u_1, \dots, u_{d-1}, 1) du_I
\end{aligned}$$

The right-hand side of (3.7) is the sum of these 3 parts:

$$\sum_{k=1}^{d-1} (-1)^k \sum_{I \in J_k^{(d-1)}} \int_I f_I^{(I)}(u_1, \dots, u_{d-1}, t_d) du_I + f(1, \dots, 1, t_d) - f(1, \dots, 1).$$

We now apply the induction hypothesis on the first term of the right-hand side of (3.7) and obtain

$$\begin{aligned}
& f(t_1, \dots, t_{d-1}, t_d) - f(1, \dots, 1, t_d) + f(1, \dots, 1, t_d) - f(1, \dots, 1) \\
&= f(t_1, \dots, t_d) - f(1, \dots, 1),
\end{aligned}$$

which proves the claim. \square

Proof of Theorem 3.3.4: Since $R_n[f(1, \dots, 1)] = 0$ for $n \geq 1$ we have by (3.7)

$$R_n[f] = \sum_{k=1}^d (-1)^k \sum_{I \in J_k^{(d)}} R_n[h_I(f^{(I)}; \cdot)], \tag{3.8}$$

where

$$h_I(f^{(I)}; t_1, \dots, t_d) = \int_I f_I^{(I)}(u) du.$$

We first consider a QMC formula for such kind of functions. Since

$$\begin{aligned}
h_I(f; t_1, \dots, t_d) &= \int_{[0,1]^d} C_I(t, u) f_I(u) du, \\
\text{where } C_I(t, u) &:= \begin{cases} 1, & \text{if } u_{i_\nu} \geq t_{i_\nu} \ \forall \nu, \\ 0, & \text{else,} \end{cases}
\end{aligned}$$

we have

$$\begin{aligned}
R_n[h_I(f; \cdot)] &= \int_{[0,1]^d} f_I(u) R_n[c_I(\cdot, u)] du \\
&\leq \left(\sup_{u \in [0,1]^d} R_n[c_I(\cdot, u)] \right) \int_{[0,1]^d} f_I(u) du \\
&\leq D^*(X) \int_{[0,1]^d} f_I(u) du ,
\end{aligned}$$

i.e., by (3.8)

$$\begin{aligned}
|R_n[f]| &\leq D^*(X) \left| \sum_{k=1}^d (-1)^k \sum_{I \in J_k^{(d)}[0,1]^d} \int f_I^{(I)}(u) du \right| \\
&\leq D^*(X) V(f) . \quad \square
\end{aligned}$$

Remark 3.3.6 *One can show that the Koksma–Hlawka estimate is in fact sharp, i.e., there exist functions f for which one has “=” in (3.7). Further details can e.g. be found in [6].*

3.4 The Smolyak Method

The Smolyak method is a deterministic method that has been used for various applications (under different names), not only for numerical cubature. The idea is starting from a 1D-method to construct an efficient n D-method using a clever setting of the grid points. This is also known as *sparse grids* which is a well-known method used for the numerical solution of partial differential equations.

Definition 3.4.1 *Let $L_i[f] := \sum_{\nu=1}^{n_1} c_{\nu,i} f(x_{\nu,i})$, $x_{\nu,i} \in \mathbb{R}$, $i = 1, \dots, d$, be a linear functional, then*

$$L_1 \otimes \dots \otimes L_d[f] := \sum_{\nu_1=1}^{n_1} \dots \sum_{\nu_d=1}^{n_d} c_{\nu_1,1} \dots c_{\nu_d,d} f(x_{\nu_1,1}, \dots, x_{\nu_d,d})$$

is called tensor product of the operators L_1, \dots, L_d .

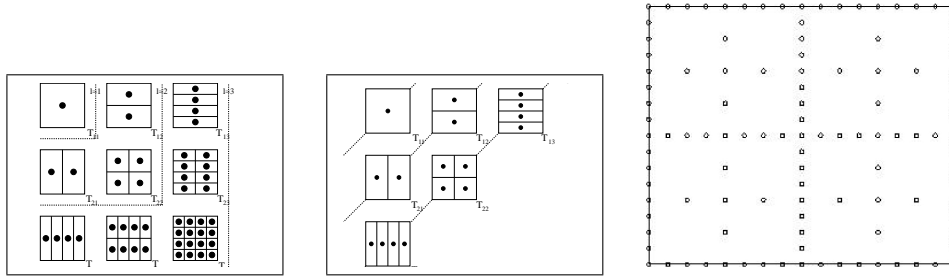


Figure 3.1: Idea for the building of a sparse grid (first two figures) and one particular example of a sparse grid.

Obviously, the product in (3.2) is of this form, i.e.,

$$Q_{n^d}^{[d]}[F] = \underbrace{Q_n \otimes \dots \otimes Q_n}_{d\text{-times}}[F] .$$

Remark 3.4.2 *As already mentioned above, the concept of Smolyak can be applied to many problems having a product-structure, here we look only at the particular application to quadrature, resp. cubature.*

Definition 3.4.3 *Let $Q^{(1)}, Q^{(2)}, \dots$, be a sequence of quadrature-formulas with n_i quadrature points and $Q^{(0)}[f] = 0$ (i.e. $n_0 = 0$) and set*

$$\Delta^{(i)} := Q^{(i+1)} - Q^{(i)} . \quad (3.9)$$

Then

$$Q(k, d) := \sum_{|i| \leq k} \Delta^{(i_1)} \otimes \dots \otimes \Delta^{(i_d)}, \quad i = (i_1, \dots, i_d), \quad (3.10)$$

is called k -th Smolyak-Quadrature formula, where, as usual, we set

$$|i| := \sum_{\nu=1}^d i_\nu .$$

Definition 3.4.4 *A sequence $Q^{(1)}, Q^{(2)}, \dots$ of quadrature formulas with respect to the quadrature points $X^{(i)}$ is called nested, if $X^{(i)} \subseteq X^{(i+1)}$ holds for the sampling points $X^{(i)}, i = 1, 2, \dots$*

To analyze the error of a Smolyak formula we define the following space of functions

$$\mathcal{F}_d^r := \left\{ g : \mathbb{R}^d \rightarrow \mathbb{R} : \|g\|_r := \left\| \frac{\partial^{|i|} g}{\partial x_1^{i_1} \dots \partial x_d^{i_d}} \right\| < \infty, \right. \\ \left. \text{for all } i = (i_1, \dots, i_d), i_\nu \leq r, 1 \leq \nu \leq d \right\}.$$

For a linear functional $L \in L(\mathcal{F}_d^r, \mathbb{R})$ set

$$\|L\|_r := \sup_{0 \neq f \in \mathcal{F}_d^r} \frac{|L[f]|}{\|f\|_r}.$$

Now we start with 1D quadrature formulas satisfying the following error estimate

$$|R_n^{(i)}, [f]| \leq \frac{C_r}{n_i^r} \|f\|_r, \quad f \in \mathcal{F}_1^r, \quad (3.11)$$

see Theorem 3.1.3.

Theorem 3.4.5 *Let $Q^{(1)}, Q^{(2)}, \dots$ be nested such that (3.11) holds as well as*

$$a2^i \leq n_i \leq A2^i. \quad (3.12)$$

Then

$$|R(k, d)[f]| \leq C_{d,r} \frac{(\log n(k, d))^{(d-1)(r+1)}}{n(k, d)^r} \|f\|_r, \quad (3.13)$$

holds for all $f \in \mathcal{F}_d^r$, where $n(k, d)$ is the number of quadrature points of $Q(k, d)$ and the constant $C_{d,r}$ does not depend on f .

For the proof we again need some preparations:

Lemma 3.4.6 *Let L_i be linear functionals on \mathcal{F}_d^r , then*

$$\|L_1 \otimes \dots \otimes L_d\|_r = \|L_1\|_r \dots \|L_d\|_r. \quad (3.14)$$

Proof: We proceed by induction over d . For $d = 1$ nothing has to be done. For $d \geq 2$ and $f \in \mathcal{F}_d^r$ we consider the following function

$$g := (L_2 \otimes \cdots \otimes L_d)[f] \in \mathcal{F}_1^r.$$

A straightforward calculation yields

$$\begin{aligned} \frac{\partial}{\partial x_1} g(x_1) &= \sum_{\nu_2=1}^{n_2} \cdots \sum_{\nu_d=1}^{n_d} c_{\nu_2,2} \cdots c_{\nu_d,d} \frac{\partial}{\partial x_1} f(x_1, \dots, x_d) \\ &= (L_2 \otimes \cdots \otimes L_d) \left[\underbrace{\frac{\partial}{\partial x_1} f(x_1, \cdot, \dots, \cdot)}_{\in \mathcal{F}_{d-1}^r \text{ with norm } \leq \|f\|_r} \right] (x_2, \dots, x_d), \end{aligned}$$

thus using the induction hypothesis

$$\begin{aligned} \|g\|_r &\leq \|(L_2 \otimes \cdots \otimes L_d)\|_r \|f\|_r \\ &= (\|L_2\|_r \cdots \|L_d\|_r) \|f\|_r \end{aligned}$$

and hence

$$\begin{aligned} |(L_1 \otimes \cdots \otimes L_d)[f]| &= |L_1[g]| \leq \|L_1\|_r \|g\|_r \\ &\leq (\|L_1\|_r \cdots \|L_d\|_r) \|f\|_r. \end{aligned}$$

This finally implies

$$\|L_1 \otimes \cdots \otimes L_d\|_r \leq \|L_1\|_r \cdots \|L_d\|_r. \quad (3.15)$$

Hence, for any $\varepsilon > 0$, there exists a function $f_i = f_i(\varepsilon)$ such that

$$L_i[f_i] \geq \|L_i\|_r \|f_i\|_r (1 - \varepsilon).$$

Using these functions for all i , we set

$$f(u_1, \dots, u_d) := f_1(u_1) \cdots f_d(u_d) = (f_1 \otimes \cdots \otimes f_d)(u_1, \dots, u_d),$$

thus

$$\|f\|_r = \|f_1\|_r \cdots \|f_d\|_r$$

and thus

$$\begin{aligned} |(L_1 \otimes \cdots \otimes L_d)[f]| &= |L_1[f_1] \cdots L_d[f_d]| \\ &\geq \|L_1\|_r \cdots \|L_d\|_r \|f_1\|_r \cdots \|f_d\|_r (1 - \varepsilon)^d, \end{aligned}$$

which yields the assertion with (3.12). \square

The next result will be needed in order to estimate the number of cubature points which in turns is required to analyze the rate of convergence.

Lemma 3.4.7 *Under the hypothesis of Theorem 3.4.5 we have*

$$n(k, d) \leq A^d 2^k \binom{d+k-1}{d-1}.$$

Proof: Let $X(k, d)$ be the cubature points of $Q(k, d)$ and X_i be the cubature points of $\Delta^{(i)}$ in (3.9). Because of the nestedness we have

$$\#\Delta^{(i)} = \#(Q^{(i+1)} - Q^{(i)}) = \#X^{(i+1)},$$

thus by (3.12)

$$\begin{aligned} \#X(k, d) &= \# \left(\sum_{|i|=k} X^{(i_1)} \otimes \dots \otimes X^{(i_d)} \right) \\ &= \sum_{|i|=k} (\#X^{(i_1)}) \dots (\#X^{(i_d)}) \\ &= \sum_{|i|=k} n_{i_1} \dots n_{i_d} \\ &\leq \sum_{|i|=k} A 2^{i_1} \dots 2^{i_d} = \sum_{|i|=k} A^d 2^{|i|}. \end{aligned}$$

Because of

$$\#\{(i_1, \dots, i_d) \in \mathbb{N}^d : |i| = k\} = \binom{d+k-1}{d-1}. \quad (3.16)$$

(for a proof see below) we conclude

$$\#X(k, d) \leq A^d 2^k \binom{d+k-1}{d-1}.$$

It remains to prove (3.16). We first show that

$$\sum_{n=0}^k \binom{n}{d} = \binom{k+1}{d+1} \quad (3.17)$$

for all $d \geq 1$ by induction. For $k = 0$ the claim holds because of

$$\binom{0}{d} = \binom{1}{d+1} = 0$$

for all $d \geq 1$. For $k \geq 1$ we conclude by the induction hypothesis

$$\sum_{n=0}^{k+1} \binom{n}{d} = \binom{k+1}{d} + \sum_{n=0}^k \binom{n}{d} = \binom{k+1}{d} + \binom{k+1}{d+1} = \binom{k+2}{d+1},$$

so that (3.17) is shown.

Now let $N_k^d := \#\{(i_1, \dots, i_d) \in \mathbb{N}^d : |i| = k\}$, then obviously we have $N_k^1 = \#\{(k)\} = 1 = \binom{1+k-1}{1-1} = \binom{k}{0} = 1$ and again by induction

$$N_k^d = \sum_{m=0}^k N_{k-m}^{d-1} = \sum_{m=0}^k N_m^{d-1} = \sum_{m=0}^k \binom{d-1+m-1}{d-2} = \binom{d+k-1}{d-1},$$

which proves (3.16) in view of (3.17). \square

Now we show one final auxiliary result in preparation for the proof of the main result.

Lemma 3.4.8 *Under the hypotheses of Theorem 3.4.5 we have*

$$\|R(k, d)\|_r \leq \tilde{C}_r 2^{-rk} (1 + 2^{-r})^{d-1} \binom{d+k}{d-1}.$$

Proof: Again by induction we obtain for a multi-index $i = (i_1, \dots, i_d)$ by using a telescoping sum

$$\begin{aligned} Q(k, d+1) &= \sum_{|i| \leq k} \left(\Delta^{(i_1)} \otimes \dots \otimes \Delta^{(i_d)} \otimes \sum_{\nu=0}^{k-|i|} \Delta^{(\nu)} \right) \\ &= \sum_{|i| \leq k} \left(\Delta^{(i_1)} \otimes \dots \otimes \Delta^{(i_d)} \otimes Q^{(k+1-|i|)} \right) \end{aligned}$$

thus by splitting the first d and the last variable

$$I_{d+1} - Q(k, d+1) = (I_d - Q(k, d)) \otimes I_1 + \sum_{|i| \leq k} \Delta^{(i_1)} \otimes \dots \otimes \Delta^{(i_d)} \otimes (I_1 - Q^{(k+1-|i|)})$$

Because of Lemma 3.4.6, (3.11) and (3.12) we have by the triangle inequality

$$\begin{aligned} \|\Delta^{(i_\nu)}\|_r &= \|Q^{(i_\nu+1)} - Q^{(i_\nu)}\|_r \leq \|Q^{(i_\nu+1)} - I_1\|_r + \|Q^{(i_\nu)} - I_1\|_r \\ &= \|R_{n_{i_\nu+1}}^{(i_\nu+1)}\|_r + \|R_{n_{i_\nu}}^{(i_\nu)}\|_r \leq \frac{c_r}{n_{i_\nu+1}^r} + \frac{c_r}{n_{i_\nu}^r} \\ &\leq \frac{c_r}{a^r} (2^{-r(i_\nu+1)} + 2^{-ri_\nu}) = \frac{c_r}{a^r} 2^{-ri_\nu} (1 + 2^{-r}). \end{aligned}$$

Next using $\|I_1\|_r = 1$ we have using the triangle inequality and Lemma 3.4.6

$$\begin{aligned}
\|R(k, d+1)\|_r &= \|I_{d+1} - Q(k, d+1)\|_r \\
&\leq \|R(k, d)\|_r + \sum_{|i|\leq k} \|\Delta^{(i_1)} \otimes \dots \otimes \Delta^{(i_d)} \otimes R(k+1-|i|, 1)\|_r \\
&= \|R(k, d)\|_r + \sum_{|i|\leq k} \|\Delta^{(i_1)}\|_r \dots \|\Delta^{(i_d)}\|_r \|R(k+1-|i|, 1)\|_r \\
&\lesssim \|R(k, d)\|_r + \underbrace{\sum_{|i|\leq k} (1+2^{-r})^d \underbrace{2^{-r|i} 2^{-r(k+2-|i|)}}_{=2^{-r(k+2)}}}_{=(1+2^{-r})^d 2^{-r(k+2)}} \binom{d+k}{d}.
\end{aligned}$$

Now, finally

$$\begin{aligned}
\|R(k, d)\|_r &\lesssim \sum_{m=0}^{d-1} 2^{-r(k+2)} \underbrace{(1+2^{-r})^m}_{\leq (1+2^{-r})^{d-1}} \binom{m+k}{m} \\
&\leq 2^{-r(k+2)} (1+2^{-r})^{d-1} \underbrace{\sum_{m=0}^{d-1} \binom{m+k}{m}}_{\binom{d+k}{d-1}},
\end{aligned}$$

which completes the proof. \square

Now we have all tools at hand and come to the proof of the main result in this section.

Proof of Theorem 3.4.5: Set $q := d+k$. According to Lemma 3.4.7 we have for the number of cubature points

$$n(k, d) \leq A^d 2^k \binom{q-1}{d-1} \leq A^d 2^k \frac{q^{d-1}}{(d-1)!}, \quad (3.18)$$

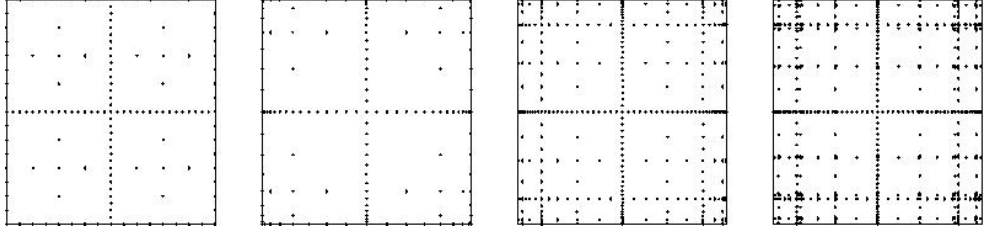


Figure 3.2: Sparse grids based on the trapezoidal, the Clenshaw-Curtis, Patterson and Gauss-Legendre rules, see also Table 3.1. This picture was taken from the web-page of T. Gerstner, Univ. Bonn.

thus in the worst case $q \sim \log n$, thus $n \sim 2^k \frac{q^{d-1}}{(d-1)!}$, or, equivalently $2^{-k} \sim n^{-1}(\log n)^{d-1}$. Now it results from Lemma 3.4.8:

$$\begin{aligned}
 \|R(k, d)\|_r &\leq \tilde{C}_r (1 + 2^{-r})^{d-1} 2^{-kr} \binom{q}{d-1} \\
 &= C_{r,d} \underbrace{2^{-k(r+1)}}_{\sim n} \underbrace{2^k \binom{q}{d-1}}_{\sim n} \\
 &\sim \left(\frac{(\log n)^{d-1}}{n} \right)^{r+1} \\
 &\lesssim C_{r,d} \frac{(\log n)^{(d-1)(r+1)}}{n^r},
 \end{aligned}$$

which proves the desired statement. \square

The Clenshaw-Curtis grids (1960)

These are widely used grids with the settings

$$n_1 = 1, \quad n_k = 2^{k-1} + 1 \quad \text{for } k \geq 2,$$

[3]. For the cubature points $X^{cc}(k, 2)$ we obtain for equidistant quadrature points the grid shown in Figure 3.2.

One further example is the Konrad-Patterson sequence, which uses the Gaussian quadrature points. The following table gives a short summary of different approaches.

$1d$	subdivision	$\#X$
Newton-Cotes	equidistant	$r_i - 1$
Chenshow-Curtis	Chebyshev	$n_i - 1$
Patterson (1968)	Stieltjes	$\frac{3}{2}n_i - 1$
Gauss	Legendre	$2n_i - 1$

Table 3.1: Kind of subdivision and degrees of freedom for different kinds of sparse grids.

Chapter 4

Numerical Computation of European Options

One key subject in mathematical finance is the modelling of stocks, derivatives and in particular option pricing. Nowadays there is a whole variety of different financial products all of which require a careful mathematical modelling. Here we describe the numerical simulation of the pricing problem of European options that will lead us to the numerical solution of (stochastic) partial differential equations.

4.1 Option Pricing: A Very Short Introduction

Since this is a lecture on *numerical* finance, we do not go into the details of the modelling of certain financial derivatives and refer to the lectures concerning mathematical finance. However, we give a very short introduction in order to describe which kind of mathematical problems occur in the option pricing problem. Here, we particularly focus on the description of the particular nature of problems that have to be treated numerically.

An option is a financial instrument depending on a *underlying* (e.g. a share, packets of shares, an index or a currency). Often options have a limited short term lifetime. The customer acquires the right to buy (Call) or sell (Put) the underlying for a previously agreed exercise price E (strike) at the date T (maturity). Let us collect some notation first.

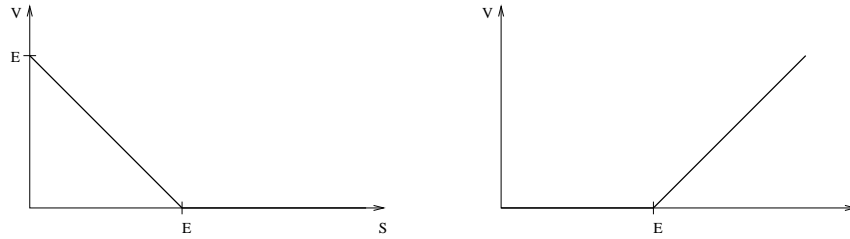


Figure 4.1: Payoff functions for a call (left) and a put (right) option.

- $S = S(t) = S_t$ denotes the stock price of the underlying;
- if an exercise of the option is only allowed at the date T , we talk of *European option*;
- for a *call* there are two scenarios, namely if
 - $E < S = S(T)$: the option is exercised and the benefit is $S - E$;
 - $E \geq S$: the option will not be exercised and is hence worthless.

This shows that the *value* of an option at maturity can be described as

$$V(S, T) = \left\{ \begin{array}{ll} 0, & \text{if } S_T \leq E, \\ S_T - E, & \text{if } S_T > E \end{array} \right\} = (S_T - E)^+, \quad (4.1)$$

where $f^+ = \max\{f, 0\}$ is the broken power function. Often $V(S, T)$ is also called *payoff function*. Correspondingly, the payoff function of a put is given by

$$V(S, T) = (E - S_T)^+ = (S_T - E)^-. \quad (4.2)$$

Before we proceed, let us collect some standard notation. Typically $r > 0$ denotes risk-less interest rate which also reflects the return that can be gained e.g. for fixed-interested bonds. The margin of fluctuation of S is typically denoted by σ , which is known as the *volatility* (always per year).

Under certain assumptions (for details, we refer e.g. to [8]), this leads to the famous *Black-Scholes-Equation* for the value function V , which is the following differential equation

$$\frac{\partial}{\partial t} V(S, t) + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2}{\partial S^2} V(S, t) + rS \frac{\partial}{\partial S} V(S, t) - rV(S, t) = 0 \quad (4.3)$$

equipped with the *end condition*

$$V(S, T) = \text{“payoff” like (4.1) or (4.2)} \quad (4.4)$$

and *boundary conditions*

$$V(0, t) = 0, \quad V(S, t) \rightarrow S \text{ for } S \rightarrow \infty \text{ (for Call)}. \quad (4.5)$$

This is a linear initial boundary value problem (PDE) for V . For (4.3, 4.4, 4.5) an analytical solution is known. However, when cost for deal (charges, taxes) k are also modelled, the additional non-linear term

$$-\sqrt{\frac{2}{\pi}} \frac{k\sigma S^2}{\sqrt{\sigma t}} \left| \frac{\partial^2}{\partial S^2} V \right|$$

is added on the left-hand side of (4.3). For this no analytical solution is known and one has to resort to numerical solution techniques.

The next sections are governed with different numerical methods for solving problems like the Black-Scholes equations. We start with the most simple ones.

4.2 Binomial Methods

Binomial methods are the first, very simple approach for the following special case of the above mentioned problem. In many applications, the user is only interested in $V(S_0, 0)$, the today's value of the option with respect to the actual rate S_0 . One uses a simple tree-like structure to develop a solution method.

The first step is to introduce a discretization in time, i.e., the continuous interval $[0, T]$ is now split by introducing knots $t_i = i\Delta t$, $i = 0, \dots, M$. Here M denotes the number of time steps and $\Delta t = \frac{T}{M}$ denotes the time step size. Then, $S(t)$ is approximated by (approximate) values $S_i := S(t_i)$ of S at the knots t_i .

The method relies on a number of assumptions which are now collected.

Assumption 4.2.1 (i) *Within a time period Δt of time, the value of S can only jump to uS ($u > 1$) or dS ($0 < d < 1$) (u means an increase of the rate -up-, d represents a decrease of the rate -down-);*

(ii) The probability for the increase of the stock is p (note that this is merely a notational assumption since p will drop out from the formulas);

(iii) The expected return corresponds to the risk-free rate of interest r , i.e.,

$$E(S_{i+1}) = S_i e^{r\Delta t}. \quad (4.6)$$

(iv) No dividends are paid.

Note that sometimes the notation $1+u$, $1+d$ is used since this is consistent with $1+r$ in the standard model for the return.

The idea is now to compare expectation rates and variances for the continuous and the discrete model. Using (i) and (ii) in Assumption 4.2.1, we obtain by (4.6)

$$E(S_{i+1}) = puS_i + (1-p)dS_i = S_i e^{r\Delta t},$$

so that

$$e^{r\Delta t} = pu + (1-p)d. \quad (4.7)$$

For variances in the continuous model it holds

$$E(S_{i+1}^2) = S_i^2 e^{(2r+\sigma^2)\Delta t},$$

thus

$$\begin{aligned} \text{Var}(S_{i+1}) &= E(S_{i+1}^2) - E(S_{i+1})^2 \\ &= S_i^2 e^{(2r+\sigma^2)\Delta t} - S_i^2 e^{2r\Delta t} \\ &= S_i^2 e^{2r\Delta t} (e^{\sigma^2\Delta t} - 1). \end{aligned}$$

In the discrete model we have

$$\text{Var}(S_{i+1}) = p(uS_i)^2 + (1-p)(dS_i)^2 - S_i^2 (pu + (1-p)d)^2,$$

so that we obtain by (4.7)

$$e^{2r\Delta t} (e^{\sigma^2\Delta t} - 1) = pu^2 + (1-p)d^2 - \underbrace{(pu + (1-p)d)^2}_{=e^{2r\Delta t}},$$

(where the assumption on the expectation is used) thus

$$e^{2r\Delta t + \sigma^2\Delta t} = pu^2 + (1-p)d^2. \quad (4.8)$$

Now (4.7, 4.8) are 2 equations for the 3 unknowns d, u, p and one additional equation is required to close the system. Often one uses (a little bit arbitrarily)

$$u \cdot d = 1 \quad (\text{or, alternatively: } p = \frac{1}{2}). \quad (4.9)$$

With $\alpha := e^{r\Delta t}$ we have by (4.7)

$$\begin{aligned} \alpha &= pu + (1-p)\frac{1}{u} \\ &= p\left(u - \frac{1}{u}\right) + \frac{1}{u} \\ &= p\left(\frac{u^2 - 1}{u}\right) + \frac{1}{u}, \end{aligned}$$

hence

$$\begin{aligned} p &= \left(\alpha - \frac{1}{u}\right) \frac{u}{u^2 - 1} = \frac{\alpha u - 1}{u^2 - 1} \\ 1 - p &= \frac{u^2 - 1 - \alpha u + 1}{u^2 - 1} = \frac{u^2 - \alpha u}{u^2 - 1}. \end{aligned}$$

Thus, we obtain

$$\begin{aligned} pu^2 + (1-p)d^2 &= \frac{\alpha u - 1}{u^2 - 1}u^2 + \frac{u^2 - \alpha u}{u^2 - 1} \frac{1}{u^2} \\ &= \frac{1}{u^2 - 1} \left[\alpha u^3 - u^2 + 1 - \frac{\alpha}{u} \right] \\ &= \alpha u - 1 + \frac{\alpha}{u} \end{aligned}$$

and with (4.8)

$$\begin{aligned} \alpha^2 e^{\sigma^2 \Delta t} &= \alpha u - 1 + \frac{\alpha}{u} \\ \iff u\alpha^2 e^{\sigma^2 \Delta t} &= \alpha u^2 - u + \alpha \\ \iff 0 &= u^2 - u \underbrace{(\alpha^{-1} + \alpha e^{\sigma^2 \Delta t})}_{=: 2\beta} + 1 \end{aligned} \quad (4.10)$$

which yields

$$u = \beta + \sqrt{\beta^2 - 1}, \quad d = \frac{1}{u} = \beta - \sqrt{\beta^2 - 1}, \quad (0 < d < 1 < u).$$

We may summarize our findings as follows

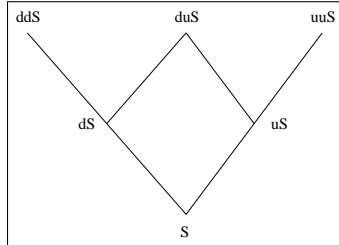
$$\begin{cases} \beta = \frac{1}{2}(e^{-r\Delta t} + e^{(r+\sigma^2)\Delta t}) \\ u = \beta + \sqrt{\beta^2 - 1} \\ d = \frac{1}{u} = \beta - \sqrt{\beta^2 - 1} \\ p = \frac{e^{r\Delta t} - d}{u - d} \quad (\text{model is valid if } 0 < p < 1). \end{cases} \quad (4.11)$$

The algorithm consists of three different phases, namely the *forward phase*, the *valuation of the tree* and the *backward phase* which we now describe.

Forward phase: Calculation of the grid, initialization of the tree

- u and d are known, hence $S(t_i), \dots, S(t_M)$ can be computed, by using S_0 as the root of the tree;
- for every time $t_i, i = 1, \dots, M$ there are $i + 1$ possibilities as shown in the following figure.

$$S_{ji} = S_0 u^j d^{i-j}, \quad j = 0, \dots, i \quad i = 1, 2, \dots, M. \quad (4.12)$$



On these grid points, we now compute approximate values for V , i.e., $V_{ji} = V(t_i, S_{ij})$, and we search for $v_{00} = V(t_0, S_0)$.

Evaluation of the tree: The value $V(S, t_M)$ of the option at the final time t_M is known through the final condition (the payoff function), i.e.

$$V_{jM} = (S_{jM} - E)^+ \text{ (Call)}, \quad V_{jM} = (E - S_{jM})^+ \text{ (Put)}. \quad (4.13)$$

Backward phase: Compute V_{ji} , $i = m - 1, m - 2, \dots, 0$ from V_{jM} . Because of (4.7) we obtain

$$\begin{aligned} S_{ji} e^{r\Delta t} &= puS_{ji} + (1 - p)dS_{ji} \\ &= pS_{j+1, i+1} + (1 - p)S_{j, i+1}, \end{aligned}$$

which we also transfer for V :

$$V_{ji} = e^{-r\Delta t}(pV_{j+1,i+1} + (1-p)V_{j,i+1}), \quad (4.14)$$

the *Martingale Property*.

Putting all pieces together, we obtain the following algorithm:

Algorithm 4.2.2 Input: r, σ, S_0, T, E, M , choice if *Call* or *Put*

- $\Delta t = \frac{T}{M}$, u, d, p like (4.11)
- $S_{00} := S_0$
- $S_{jM} = S_{00}u^j d^{M-j}$, $j = 0, 1, \dots, M$
 V_{jM} like (4.13)
- for $i = M - 1, \dots, 0$: V_{ji} like (4.14), $j = 0, \dots, i$

Output: V_{00} is an approximation for $V(S_0, 0)$.

4.3 Finite Difference Methods

Finite difference methods are maybe the simplest numerical methods for approximately solving ordinary and partial differential equations, see e.g. [4]. In this section, we give an introduction to these methods focussing on their application to the above mentioned examples from finance.

Before we do so, we reformulate the boundary conditions in (4.5). The problem in (4.5) for the numerical treatment is the asymptotic behavior of the boundary condition which cannot be handled directly. However, (4.5) can also be written as

$$V(0, t) = 0, \quad V(S_\infty, t) = S_\infty - E \quad (4.15)$$

for the call and

$$V(0, t) = E, \quad V(S_\infty, t) = 0 \quad (4.16)$$

for the put, where $E, S_\infty < \infty$. If we neglect the derivative with respect to time (i.e., we consider the *stationary process*), then (4.15,4.16) takes the following form

$$\begin{cases} Lu(x) := -u''(x) + b(x)u'(x) + c(x)u(x) = f(x), & x \in (0, 1), \\ u(0) = u(1) = 0, \end{cases} \quad (4.17)$$

where we assume $c(x) \geq 0$ for all x in order to ensure that a unique solution exists.

The most simple method for the numerical solution of (4.17) is the *classical finite difference method* in which the interval $(0, 1)$ is replaced by a set of *grid points* (or *nodes*) and the derivatives are approximated by differential quotients. For simplicity, we first consider an *equidistant grid*, i.e.,

$$x_i = i \cdot h, \quad h = \frac{1}{N}, \quad N \in \mathbb{N}, \quad N \geq 1, \quad (4.18)$$

where $h = \frac{1}{N}$ denotes the *step size*. Then,

$$\omega_h := \{x_i = ih : i = 1, \dots, N - 1\} \quad (4.19)$$

denotes the set of *interior grid points*, $\gamma_h := \{x_0, x_M\}$ the *boundary points* and $\bar{\omega}_h := \omega_h \cup \gamma_h$ the *full grid*. For the approximation of the derivatives one uses

$$\begin{aligned} \text{the forward difference} & \quad (D^+u)(x) := \frac{1}{h}(u(x+h) - u(x)) \\ \text{the backward difference} & \quad (D^-(u))(x) := \frac{1}{h}(u(x) - u(x-h)) \\ \text{the symmetric or central difference} & \quad (D^0u)(x) := \frac{1}{2h}(u(x+h) - u(x-h)) \\ \text{the second difference} & \quad (D^+D^-u)(x) := \frac{1}{h^2}(u(x+h) \\ & \quad \quad \quad - 2u(x) + u(x-h)) \end{aligned}$$

Our next aim is to study the convergence of finite difference methods. As a preparation, we have

Lemma 4.3.1 *The following error estimates hold*

- (i) $(D^0u)(x) = u'(x) + Rh^2$ with $|R| \leq \frac{1}{6}\|u'''\|_{C[0,1]}$, if $u \in C^3$
- (ii) $(D^+D^-u)(x) = u''(x) + Rh^2$ with $|R| \leq \frac{1}{12}\|u''''\|_{C[0,1]}$, if $u \in C^4[0, 1]$

Proof: Using Taylor's formula, we have

$$\begin{aligned} u(x \pm h) &= u(x) \pm hu'(x) + h^2 \frac{u''(x)}{2} \pm h^3 R_3 \\ u(x \pm h) &= u(x) \pm hu'(x) + h^2 \frac{u''(x)}{2} \pm h^3 \frac{u'''(x)}{6} + h^4 R_4 \end{aligned}$$

with the following remainder terms

$$R_3 = \frac{1}{6}h^{-3} \int_x^{x+h} [u''(\xi) - u''(x)](x \pm h - \xi) d\xi,$$

$$R_4 = h^{-4} \int_x^{x+h} \frac{1}{12}[u'''(\xi) - u'''(x)](x \pm h - \xi)^2 d\xi.$$

This already yields the desired claim. \square

Setting $g_i := g(x_i)$ and $Du_i := (Du)(x_i)$, we obtain the *classical finite difference method*

$$-D^+D^-u_i + b_iD^0u_i + c_iu_i = f, \quad i = 1, \dots, N-1,$$

$$u_0 = u_N = 0.$$

Due to

$$\begin{aligned} -D^+D^-u_i + b_iD^0u_i + c_iu_i &= \\ &= \frac{1}{h^2}(-u_{i-1} + 2u_i - u_{i+1}) + \frac{b_i}{2h}(u_{i+1} - u_{i-1}) + c_iu_i \\ &= \underbrace{\left(-\frac{1}{h^2} - \frac{b_i}{2h}\right)}_{=:r_i} u_{i-1} + \underbrace{\left(\frac{2}{h^2} + c_i\right)}_{=:c_i} u_i + \underbrace{\left(-\frac{1}{h^2} + \frac{b_i}{2h}\right)}_{=:t_i} u_{i+1} \end{aligned}$$

we obtain a *tridiagonal* system $L_h u_h = f_h$ to determine the unknown vector $u_h = (u_h(x_1), \dots, u_h(x_{N-1}))$, where the matrix L_h is given by

$$L_h = \begin{bmatrix} c_1 & t_1 & & & \\ r_2 & c_2 & t_2 & & \\ & \ddots & \ddots & \ddots & \\ & & r_{N-2} & c_{N-2} & t_{N-2} \\ & & & r_{N-1} & c_{N-1} \end{bmatrix} \in \mathbb{R}^{(N-1) \times (N-1)}$$

and the right-hand side reads $f_h = \left(f(x_i)\right)_{i=1, \dots, N-1}$.

In order to actually compute the approximation u_h , we obviously have to numerically solve a linear system of equations with a tridiagonal matrix.

where $R_h u = (u(x_1), \dots, u(x_{N-1}))$ is the restriction of the exact solution to the computational grid and $u_h = (u_1, \dots, u_{N-1})$ denotes the numerical approximation.

(ii) The finite difference method is called consistent of order k (with respect to $\|\cdot\|_\infty$) if

$$\|L_h R_h u - R_h L u\|_\infty \leq C h^k .$$

(iii) The method is called stable if $L_h u_h = f_h$ always implies the estimate $\|u_h\|_\infty \leq C \|R_h f\|_\infty$ (continuous dependence of the solution on the data).

Now we can give the first result which is essential for the convergence analysis.

Theorem 4.3.3 *If $u \in C^4[0, 1]$, then the classical finite difference method is consistent of order 2.*

Proof: An easy calculation shows

$$\begin{aligned} (L_h R_h u - R_h L u)(x_i) &= \frac{1}{h^2}(-u(x_i - h) + 2u(x_i) - u(x_i + h)) \\ &\quad + b(x_i) \frac{1}{2h}(u(x_i + h) - u(x_i - h)) + c(x_i)u(x_i) \\ &\quad + u''(x_i) - b(x_i)u'(x_i) - c(x_i)u(x_i) \\ &= u''(x_i) - (D^+ D^- u)(x_i) + b(x_i)(D^0 u(x_i) - u'(x_i)) , \end{aligned}$$

so that we obtain by Lemma 4.3.1 the estimate

$$|(L_h R_h u - R_h L u)(x_i)| \leq \frac{1}{12} h^2 \|u''''\|_{C[0,1]} + \frac{1}{6} h^2 \|b\|_{C[0,1]} \|u'''\|_{C[0,1]}$$

which proves the assertion. \square

Remark 4.3.4 *The proof of Lemma 4.3.1 shows that the statement of Theorem 4.3.3 also holds if u''' is only Lipschitz continuous, i.e., $u \in C^{3,1}[0, 1]$.*

Remark 4.3.5 *It is a central statement of the analysis of finite difference methods that consistency and stability imply convergence of the particular method. However, a rigorous proof of this goes far beyond the scope of the present lecture.*

4.4 Discretization in Time

Now we consider the time dependent problem (where here for simplicity we assume $b(x) = c(x) = 0$): determine $u(x, t)$, $x \in (0, 1)$, $t \in (0, T)$ such that

$$\begin{cases} \frac{\partial}{\partial t}u(x, t) - \frac{\partial^2}{\partial x^2}u(x, t) = f(x, t) & \text{in } (0, 1) \times (0, T), \\ u(x, 0) = u_0(x), \\ u(0, t) = u_1(t), \quad u(1, t) = u_2(t). \end{cases} \quad (4.20)$$

The simple idea is to use a finite difference method both with respect to space *and* time, [4]:

$$\begin{aligned} x_i &= ih, \quad i = 0, \dots, N, \quad h = \frac{1}{N}, \\ t^j &= j\Delta t, \quad j = 0, \dots, M, \quad \Delta t = \frac{T}{M}. \end{aligned}$$

Setting $f_i^k := f(x_i, t^k)$, we are looking for an approximation u_i^k of $u(x_i, t^k)$. Note that we always use a subscript to denote the discretization index in space and a superscript for the time discretization. For a fixed time t^k we again consider

$$D^+D^-u_i^k := \frac{1}{h^2}(u_{i-1}^k - 2u_i^k + u_{i+1}^k)$$

and define a *six point scheme* (with a free parameter $0 \leq \sigma \leq 1$) by

$$\begin{aligned} \frac{1}{\Delta t}(u_i^{k+1} - u_i^k) &= D^+D^-(\sigma u_i^{k+1} + (1 - \sigma)u_i^k) + \tilde{f}_i^k, & (4.21) \\ i &= 1, \dots, N - 1, \quad k = 1, \dots, M - 1 \\ u_i^0 &= u_0(x_i), \\ u_0^k &= u_1(t^k), \quad u_1^k = u_2(t^k), \end{aligned}$$

where \tilde{f}_i^k denotes an approximation of $f(x_i, t^k)$ (e.g. $\tilde{f}_i^k = f_i^k$).

For certain choices of σ , we obtain the following important special cases:

- (i) **Explicit method:** (with $\gamma := \frac{\Delta t}{h^2}$): $\sigma = 0$, $\tilde{f}_i^k := f_i^k$:

$$u_i^{k+1} = (1 - 2\gamma)u_i^k + \gamma(u_{i-1}^k + u_{i+1}^k) + \Delta t f_i^k$$

- (ii) Purely **implicit method:** $\sigma = 1$, $\tilde{f}_i^k := f_i^k$:

$$(1 + 2\gamma)u_i^{k+1} - \gamma(u_{i+1}^{k+1} + u_{i-1}^{k+1}) = u_i^k + \Delta t f_i^k$$

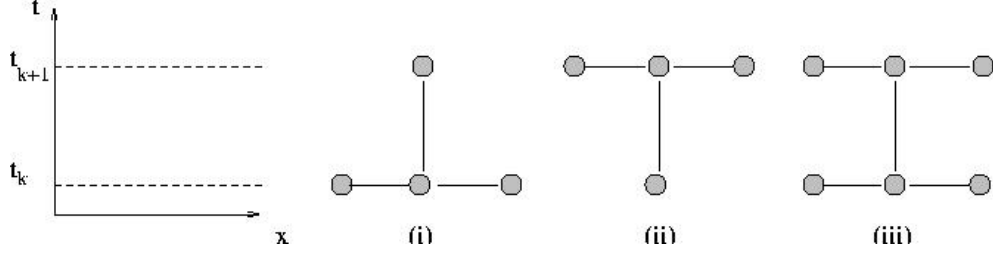


Figure 4.2: Finite difference methods in time for the special choices of σ .

(iii) **Crank-Nicolson method:** $\sigma = \frac{1}{2}$, $\tilde{f}_i^k := f(x_i, t^k + \frac{\Delta t}{2})$

$$2(\gamma + 1)u_i^{k+1} - \gamma(u_{i+1}^{k+1} + u_{i-1}^{k+1}) = 2(1 - \gamma)u_i^k + \gamma(u_{i+1}^k + u_{i-1}^k) + 2\Delta t f(x_i, t^k + \frac{\Delta t}{2}).$$

In (i), the approximation $(u_i^{k+1})_{i=1, \dots, N-1}$ on the new time level can be computed directly from the data $(u_i^k)_{i=1, \dots, N-1}$ on the previous time step. In the other two cases, one has to solve a linear system of equations to proceed in time. The three variants are shown in Figure 4.2.

Theorem 4.4.1 *For the consistency error on $Q := (0, 1) \times (0, T)$ we obtain the following orders*

- (i) $\mathcal{O}(h^2 + \Delta t)$ for arbitrary σ , $\tilde{f}_i^k = f(x_i, t^k)$ and for $u \in C^{4,2}(\bar{Q})$.
- (ii) $\mathcal{O}(h^2 + \Delta t^2)$ for the Crank-Nicolson method for $u \in C^{4,3}(\bar{Q})$.

Proof: We only prove (ii) here. Using Taylor's formula, we obtain

$$\frac{1}{\Delta t}(u(x, t + \Delta t) - u(x, t)) = u_t(x, t) + \frac{1}{2}u_{tt}(x, t)\Delta t + \mathcal{O}(\Delta t^2)$$

as well as

$$\frac{1}{2h^2}\{u(x - h, t + \Delta t) - 2u(x, t + \Delta t) + u(x + h, t + \Delta t) - u(x - h, t) + 2u(x, t) - u(x + h, t)\} =$$

$$\begin{aligned}
&= \frac{1}{2}\{2u_{xx} + \Delta t u_{xxt} + \mathcal{O}(\Delta t^2 + h^2)\} , \\
f(x, t + \frac{\Delta t}{2}) &= f(x, t) + \frac{\Delta t}{2} f_t + \mathcal{O}(\Delta t^2) .
\end{aligned}$$

Thus, we obtain for the consistency error

$$\begin{aligned}
C_{\text{cons}} &= \underbrace{u_t - u_{xx} - f + \frac{1}{2}\Delta t(u_{tt} - u_{xxt} - f_t)}_{=0} + \mathcal{O}(\Delta t^2 + h^2) \\
u_t &= u_{xx} + f \Rightarrow u_{tt} = u_{xxt} + f_t
\end{aligned}$$

which proves the theorem. \square

Now, in view of Remark 4.3.5 we come to the analysis of the stability.

Theorem 4.4.2 *We have*

$$\max_k \max_i |u_i^{k+1}| \leq \max_x |u_0(x)| + \Delta t \sum_{j=0}^k \max_i |\tilde{f}_i^j| ,$$

i.e., the method is stable with respect to the discrete supremum-norm, provided that $1 - 2(1 - \sigma)\gamma \geq 0$.

Proof: Rewrite (4.21) in the following form

$$-\gamma\sigma u_{i-1}^{k+1} + (2\sigma\gamma + 1)u_i^{k+1} - \sigma\gamma u_{i+1}^{k+1} = F_i^k$$

where

$$F_i^k := (1 - \sigma)\gamma u_{i-1}^k + (1 - 2(1 - \sigma)\gamma)u_i^k + (1 - \sigma)\gamma u_{i+1}^k + \Delta t \tilde{f}_i^k .$$

For simplicity we consider homogeneous boundary conditions, i.e., $u_1(x) = u_2(x) = 0$. Since the matrix

$$A = \begin{bmatrix} 2\sigma\gamma + 1 & -\gamma\sigma & & 0 \\ -\gamma\sigma & \ddots & \ddots & \\ & \ddots & \ddots & -\gamma\sigma \\ 0 & -\gamma\sigma & & 2\sigma\gamma + 1 \end{bmatrix}$$

is strictly diagonal dominant, i.e.,

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$$

we obtain

$$\|A^{-1}\| \leq \frac{1}{\min_k \left(a_{kk} - \sum_{j \neq k} |a_{j,k}| \right)} = 1.$$

The proof of the latter inequality we leave as an exercise. Then we have $\max_i |u_i^{k+1}| \leq \max_i |F_i^k|$ as well as

$$\max_i |F_i^k| \leq \max_i |u_i^k| + \Delta t \max_i |\tilde{f}_i^k|$$

which is now applied to $k, k-1, \dots, 0$. \square

Now we obtain the following error estimates.

Theorem 4.4.3 *Let $(1 - \sigma) \frac{\Delta t}{h^2} \leq \frac{1}{2}$ and $u \in C^{4,2}(\bar{Q})$ and $\tilde{f}_i^k = f(x_i, t^k)$, the we have*

$$\max_{i,k} |u(x_i, t^k) - u_i^k| \leq C(h^2 + \Delta t).$$

For the Crank-Nicholson method ($\sigma = \frac{1}{2}$) we have for $\frac{\Delta t}{h^2} \leq 1$ the estimate

$$\max_{i,k} |u(x_i, t^k) - u_i^k| \leq C(h^2 + \Delta t^2).$$

Proof: The statement is an immediate consequence of the Theorems 4.4.1 and 4.4.2, [4]. \square

Remark 4.4.4 *The stability condition $(1 - \sigma) \frac{\Delta t}{h^2} \leq \frac{1}{2}$ is always valid for the purely implicit method ($\sigma = 1$). For $\sigma \neq 1$ this condition yields a restriction for the relation of the step size with respect to time and space.*

4.5 Stochastic Differential Equations (SDE)

So far we have considered problems of the kind

$$\frac{\partial}{\partial t} u(x, t) = L(u, t)$$

where the differential operator L takes the form

$$L(u, t) = u''(x, t) - b(x, t)u'(x, t) - c(x, t)u(x, t) + f(x, t)$$

with deterministic functions b , c and f . As we have already seen, this may not be appropriate e.g. for modelling stocks since essential stochastic influences have to be taken into account, [8].

The most simple model is to consider additive stochastic perturbations, i.e.,

$$\frac{d}{dt}u(x, t) = L(u, t) + \tilde{b}(u, t)\xi_t \quad (4.22)$$

where L denotes the deterministic part, $\tilde{b}\xi_t$ is the stochastic part and ξ_t denotes an generalized stochastic process.

Definition 4.5.1 (i) A continuous stochastic process is a family of random variables $X(t)$, $t \in [0, T]$, resp. $t \in \mathbb{R}$. It is often denoted by X_t , $\{X_t : t \in [0, T]\}$.

(ii) A Wiener process W_t is a continuous stochastic process with

(a) $W_0 = 0$

(b) The increments $\Delta W_i := W_{t_{i+1}} - W_{t_i}$ are mutually independent for all $0 \leq t_1 < t_2 < t_3 < \dots$ with $\Delta W_i \sim \mathcal{N}(0, t_{i+1} - t_i)$.

Example 4.5.2 If a Wiener process W_t is interpreted as a generalized stochastic process, then the derivative (in the distributional sense)

$$\xi_t = \frac{d}{dt}W_t \quad \text{or} \quad W_t = \int_0^t \xi_s ds \quad (4.23)$$

is known as white noise. This is also a particular example within the above model.

Numerical Realization of a Wiener Process

Let $\Delta t > 0$ be the time step and $t_j := j \Delta t$, then we use a telescopic argument to obtain

$$\Rightarrow W_{j\Delta t} = \sum_{k=1}^j \underbrace{(W_{k\Delta t} - W_{(k-1)\Delta t})}_{=: \Delta W_k}, \quad W_0 := 0.$$

Since $E(W_t - W_s) = 0$ and

$$\text{Var}(W_t - W_s) = 0 = t - s \quad (\text{because } W_t - W_s \sim \mathcal{N}(0, t - s))$$

and the variables ΔW_k are independent, normally distributed, one can easily see that $\text{Var}(\Delta W_k) = \Delta t$. This means that $Z \sim \mathcal{N}(0, 1)$, thus $Z\sqrt{\Delta t} \sim \mathcal{N}(0, \Delta t)$ and finally

$$\Delta W_k := Z\sqrt{\Delta t} \text{ with } Z \sim \mathcal{N}(0, 1)$$

Hence we obtain:

Algorithm 4.5.3 (Approximation of a Wiener process)

- $t_0 = 0$, $W_0 = 0$; we select Δt
- for $j = 1, 2, \dots$
 - $t_j = t_{j-1} + \Delta t$
 - $Z \sim \mathcal{N}(0, 1)$
 - $W_j = W_{j-1} + Z\sqrt{\Delta t}$

In view of (4.23) one integrates (4.22) with respect to t in order to obtain a smoother version (a realization of W_t is in general continuous but nowhere differentiable)

$$u(x, t) = u_0(x) + \int_0^t L(u, s) ds + \int_0^t \tilde{b}(u, s) \xi_s ds \quad (4.24)$$

$$=: u_0(x) + \int_0^t L(u, s) ds + \int_0^t \tilde{b}(u, s) dW_s \quad (4.25)$$

which is known as the *Itô stochastic differential equation*. The first integral on the right-hand side of (4.24) is a Lebesgue integral whereas the second one is an *Itô integral* which requires a separate calculus. Often (4.25) is written in a symbolic form as

$$du = L(u, t) dt + b(u, t) dW_t \quad (4.26)$$

where the first term is called *drift* and the second *diffusion*. A solution u of (4.26) is also called an *Itô process*.

As already mentioned, the definition and computation of an Itô integral is a topic of its own and goes beyond the scope of this lecture. For constant $b(u, s) \equiv b_0$ one has

$$\int_0^t b(u, s) dW_s = b_0 W_t, \quad (W_0 = 0).$$

For more general functions b , we refer to the literature.

A simple idea for a numerical method

It is now straightforward to combine an explicit discretization in time (Euler method) with Algorithm 4.5.3. Then, we can write (4.26) in discrete form

$$\Delta u = L(u, t)\Delta t + b(u, t)\Delta W \quad (4.27)$$

in order to determine approximations u^j of $u(\cdot, t^j)$. Then, we obtain

Algorithm 4.5.4 (Euler-Maruyama method) Set $W_0 = 0$ and $u_0 = (u(x_k, 0))_k = (u_0(x_k))_k$ and select Δt . For $j = 0, 1, 2, \dots$

- $t^{j+1} = t^j + \Delta t$
- We define pseudo random numbers $Z \sim \mathcal{N}(0, 1)$, $Z = (Z_k)_k$
- $\Delta W = Z\sqrt{\Delta t}$
- $u^{j+1} = u^j + L(u^j, t^j)\Delta t + b(u^j, t^j)\Delta W$

where the derivatives in L are again replaced by finite differences.

Solutions of a SDE for one particular realization of W_t are called *trajectory* or *path*. A *simulation* of a SDE is the computation of several trajectories.

Example 4.5.5 (Geometrical Brownian motion of stocks)

This is one of the most important models for fluctuations in stocks S

$$\frac{dS}{S} = \mu dt + \sigma dW$$

where the left-hand side is called return (i.e., the relative change in a time interval dt) and the first term on the right-hand side is called drift. In our above general model we have $L(S,t) = \mu S$ is the expected drift rate, $\tilde{b}(S,t) = \sigma S$ where σ is the volatility and μ and σ are constant. This is a reference model and the assumptions yielding the Black-Scholes equations are based upon this. The deterministic part is

$$\frac{dt}{d}S = \mu S \Rightarrow S_t = S_0 e^{\mu(t-t_0)}$$

which is, due to $E(W_t) = 0$, the expectation of a stochastic process. The discrete form is

$$\begin{aligned} \frac{\Delta S}{S} &= \mu \Delta t + \sigma Z \sqrt{\Delta t} \\ \iff \Delta S &= \mu S \Delta t + \sigma S Z \sqrt{\Delta t} \end{aligned}$$

and on this we apply Algorithm 4.5.4.

We now come to the error analysis. Let $u_T = (u(x_i, T))_i$ be the vector of the exact solution at time T with respect to the computational grid $(x_i)_i$ and u_T^h denote a numerical approximation.

Definition 4.5.6 (i) The term $\varepsilon(h) := E(|u_T - u_T^h|)$ is called the error, where the expectation is computed over all Wiener processes.

(ii) The method converges strong with order $p > 0$ if

$$\varepsilon(h) = \mathcal{O}(h^p), \quad h \rightarrow 0+$$

(iii) The method converges strong if

$$\lim_{h \rightarrow 0} E(|u_T - u_T^h|) = 0$$

(iv) The method converges weakly w.r.t. a function g with order $p > 0$ if

$$|E(g(u_T)) - E(g(u_T^h))| = \mathcal{O}(h^p), \quad h \rightarrow 0+ .$$

Remark 4.5.7 The notion of strong convergence is appropriate if one is interested in a single trajectory ('pointwise' convergence). Often, one is interested in moments only (e.g. $E(u_T)$, $\text{Var}(u_T)$, $E(|u_T|^q)$, ...), or more general on a function g of the solution u .

We quote the following convergence result without giving its proof.

Theorem 4.5.8 *The Euler-Maruyama method converges strong with order $p + \frac{1}{2}$ and weakly with order $p + 1$ w.r.t. all polynomials. \square*

Remark 4.5.9 *One can construct higher order methods using stochastic Taylor expansions and the Itô calculus. Another approach is to consider Runge-Kutta type methods instead of a simple Euler discretization. For details, we refer to [8, Chap. 3]. One should however always take into account that the solution of a SDE might lack any regularity which limits the use of high order methods.*

4.6 Computation of Moments

In this section we introduce two different approaches for the solution of a SDE where one is only interested in certain moments of the solution and not in the solution itself.

4.6.1 Monte-Carlo Methods

A commonly used approach is a Monte-Carlo method, in which several (random) trajectories are computed and then a Monte-Carlo integration method is used to compute the desired moments (if they are integrals, of course).

Example 4.6.1 (Geometrical Brownian motion of stocks S)

This model is governed by the SDE

$$\frac{dS}{S} = \mu dt + \sigma dW$$

where μ is the expected growing rate. In the risk-free case, one has $\mu = r$. The idea is to compute an approximation for the expectation for the option at the final time T and discount this to obtain

$$V(S_0, 0) = \tilde{\mathbb{E}}(e^{-rT} V(S_T, T))$$

where $\tilde{\mathbb{E}}$ denotes the risk-free expectation. I.e., the discounted expectation of the option is computed.

Thus we obtain

Algorithm 4.6.2 (1) For $k = 1, \dots, N$ determine an approximation of the SDE

$$dS = rS dt + \sigma S dW, \quad S(0) = S_0, \quad 0 \leq t \leq T,$$

and call the result $(S_T)_k$.

(2) Evaluate the payoff function (4.1) resp. (4.2) and obtain

$$(V(S_T, T))_k := V((S_T)_k, T) \quad k = 1, \dots, N.$$

(3) The following is an estimator for the risk-neutral expectation

$$\hat{\mathbb{E}}(V(S_T, T)) := \frac{1}{N} \sum_{k=1}^N (V(S_T, T))_k.$$

(4) By discountation

$$\hat{V} := e^{-rT} \hat{\mathbb{E}}(V(S_T, T)) \quad (\approx V(S_0, 0)).$$

Some remarks on the latter algorithm are in order.

- In this simple form it is only applicable for European options.
- In order to obtain a reasonable approximation, the value of N has to be large in practice, e.g. in the range of 10000.
- Step (3) can also be replaced by a Smolyak method.
- Monte-Carlo methods are in particular a method of choice if the simplifying assumptions yielding the Black-Scholes equations do *not* hold. If these assumptions hold, a Monte-Carlo method is too costly.

Remark 4.6.3 (From the Brownian model to Black-Scholes equations) Let us briefly describe why the model of the Brownian motion is a key ingredient for the development of the Black-Scholes equations. One assumes continuous payments of dividends $\delta S dt$ so that we obtain

$$\frac{dS}{S} = (\mu - \delta) dt + \sigma dW$$

From this one obtains the Black-Scholes equations

$$\frac{\partial}{\partial t}V + \frac{\sigma^2}{2}S^2 \frac{\partial^2}{\partial S^2}V + (r - \delta)S \frac{\partial}{\partial S}V - rV = 0 \quad (4.28)$$

where $S = Ee^x$, $t = T - \frac{\tau}{\frac{1}{2}\sigma^2}$, $q := \frac{2r}{\sigma^2}$, $q_\delta := \frac{2(r-\delta)}{\sigma^2}$

$$V(S, t) = E \exp \left\{ -\frac{1}{2}(q_\delta - 1)x - \left(\frac{1}{4}(q_\delta - 1)^2 + q \right) \tau \right\} y(x, \tau)$$

Then, (4.28) is equivalent to

$$\frac{\partial}{\partial r}y(x, \tau) = \frac{\partial^2}{\partial x^2}y(x, \tau). \quad (4.29)$$

Note that the Itô formula is used in this example which we hide and refer the reader to the literature for details.

4.6.2 A Deterministic Approach

The following approach is taken from [7] and describes that in certain situations a stochastic PDE for certain moments can be transformed into a deterministic one, but often in higher dimension.

Let (Ω, Σ, P) be a σ -finite probability space, $D = [a, b] \subset \mathbb{R}$ and $u : D \times \Omega \rightarrow \mathbb{R}$ be a measurable function. Then, we consider the following SDE

$$\begin{cases} L(u) := -(a(x)u'(x, \omega))' = f(x, \omega) & \text{in } D \\ u(a, \omega) = u(b, \omega) = 0 \end{cases} \quad (4.30)$$

We are in particular interested in the first and second moment of the solution u , namely

$$\begin{aligned} E_u(x) &= \int_{\Omega} u(x, \omega) dP(\omega), \\ C_u(x, y) &= \int_{\Omega} u(x, \omega)u(y, \omega) dP(\omega). \end{aligned}$$

Then, we obtain the deterministic PDE for the first moment E_u

$$L(E_u) = E_f$$

For the second moment C_u we obtain the following equation

$$\frac{d}{dx} \frac{d}{dy} \left(a(x)a(y) \frac{d}{dx} \frac{d}{dy} C_u(x, y) \right) = F(f) \quad (4.31)$$

which is also a deterministic one but now in 2D. Of course, doubling the dimension heavily increases the numerical effort. But, as it was shown in [7], the solution C_u has a specific kind of regularity which allow the use of sparse grid methods. The amount of work of this, in turns, can be shown to be of the same range as a 1D problem.

Chapter 5

Elliptic Partial Differential Equations

In this chapter, we give a brief introduction to numerical methods for solving elliptic partial differential equations. This is a wide topic and could easily fill a lecture by its own. We can only focus on some aspects here. We will focus mainly on the 2D case here. We have already seen in (4.30) one example in mathematical finance yielding an elliptic PDE in 2D. Before we proceed, let us give another example.

Example 5.0.1 (Multi-Factor models) *A short rate model $r(t)$ gives the short interest rate $r(t)$ in dependence of t . A standard model for this is*

$$dr(t) = a(t, r(t)) dt + b(t, r(t)) dW_t$$

where the functions a, b are sufficiently smooth and W_t is a real-valued Brownian motion. In particular, this is the only source of risk in this model.

One however observes, that not all known short rates can be modelled with one single term W_t , thus one considers the model

$$r(t) = R(X_t, t),$$

where X_t is some Itô-process in \mathbb{R}^d . For example in the well-known Vasicek model one obtains then the following equation for the price function F of a bond in dependence of $r(t)$

$$\begin{cases} F_t(x, t) + \sigma(x, t)\Delta F(x, t) + \boldsymbol{\mu}(x, t)\nabla F(x, t) - R(x, t)F(x, t) = h(x, t) \\ \forall (x, t) \in \mathbb{R}^d \times [0, T] \end{cases} \quad (5.1)$$

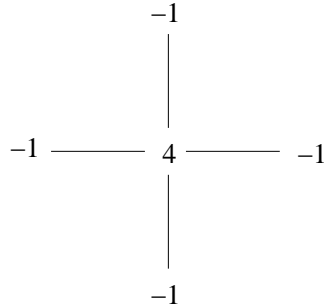


Figure 5.1: Five point stencil

5.1 Finite Difference Methods

In analogy to the 1D-case, we can approximate F_t by finite differences. Again, we do the same in space and obtain

$$(\Delta_2 u)(x, y) := \frac{1}{h^2} [u(x-h, y) + u(x+h, y) + u(x, y-h) + u(x, y+h) - 4u(x, y)]$$

i.e., the well-known *five point stencil* which is visualized in Figure 5.1. In the 1D case, we have seen that the resulting system matrix is tridiagonal and we could use a special version of a direct solver in order to obtain an efficient numerical method. In 2D, we obtain a block-tridiagonal matrix for which there is not such a nice recursion formula. Here, one has to use iterative methods such as Gauß-Seidel, Jacobi, cg or pcg-methods.

As we have already seen in the discussion of 1D problems, finite difference methods are quite simple to derive (and also to implement). But, as we also have seen in 1D, their use pose quite strong regularity assumptions on the solution which might not be satisfied in realistic applications. Moreover, since a finite difference method basically corresponds to a rectangular grid, the treatment of non-trivial geometric domains is a non-trivial task. Even more substantial is the following observation.

Example 5.1.1 *We consider the Poisson problem on the unit square*

$$\begin{aligned} -\Delta u &= 0 & \text{in } \Omega &= (0, 1)^2 \\ u &= x^2 & \text{on } \Gamma &= \partial\Omega \end{aligned}$$

This problem has a unique solution, but

$$u_{xx}(0, 0) = 2 \neq 0 = u_{yy}(0, 0)$$

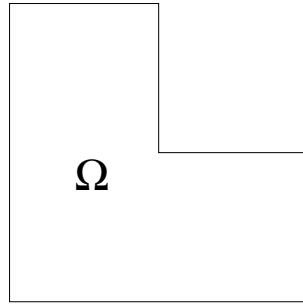


Figure 5.2: L-shaped domain.

due to the boundary conditions which contradicts the differential equation. Thus the solution u cannot be twice continuously differentiable: $u \notin C^2(\bar{\Omega})$.

Example 5.1.2 Again we consider the Poisson problem, but now on the L-shaped domain

$$\Omega := \left(-\frac{1}{2}, \frac{1}{2}\right) \times \left(-\frac{1}{2}, \frac{1}{2}\right) \setminus \left[0, \frac{1}{2}\right] \times \left[0, \frac{1}{2}\right]$$

which is shown in Figure 5.2. Also this problem has a unique solution which can be written in polar coordinates as

$$\begin{aligned} -\Delta u &= 0 && \text{in } \Omega \\ u &= r^{\frac{2}{3}} \sin\left(\frac{2\varphi - \pi}{3}\right) && \text{on } \Gamma \end{aligned}$$

It can easily be seen that the first derivatives of u are not bounded, i.e., we have $u \notin C^1(\bar{\Omega})$.

The above remarks and the two examples clearly show that in many situations the classical (strong) formulation of PDEs are not adequate. Obviously it is not always meaningful to pose a pointwise condition in the partial differential equation. In the sequel we will hence introduce certain weak formulations of elliptic PDEs. Before doing so, we should explain what is meant by *elliptic*.

This chapter is mainly based upon [1, 2].

5.2 Categories of Second Order PDEs

In the sequel, we consider the general linear second order differential equation in n variables

$$-\sum_{i,k=1}^n a_{i,k}(x) \frac{\partial}{\partial x_i} \frac{\partial}{\partial x_k} u(x) + \sum_{i=1}^n b_i(x) \frac{\partial}{\partial x_i} u(x) + c(x)u(x) = f(x). \quad (5.2)$$

In the case $a_{i,k}(x) \equiv a_{i,k}$, $b_i(x) \equiv b_i$, $c(x) \equiv c$, (5.2) is called a PDE with *constant coefficients*. Since $u_{x_i x_k} = u_{x_k x_i}$ if $u \in C^2$, we can assume without loss of generality that

$$A(x) := (a_{i,k}(x))_{i,k,\dots,n} \in \mathbb{R}^{n \times n}$$

is symmetric.

Definition 5.2.1 *The equation (5.2) is called*

- (i) *elliptic in x , if $A(x)$ is positive definite;*
- (ii) *hyperbolic in x , if $A(x)$ has one negative and $n-1$ positive eigenvalues;*
- (iii) *parabolic in x , if $A(x)$ is positive semi-definite but not definite and the rank of $(A(x), b(x))$ is n .*

If (5.2) is elliptic, one often abbreviates (5.2) as $Lu = f$.

Example 5.2.2 *Let $Lu = f$ be elliptic.*

- (i) *The equation $u_{tt} + Lu = f$ is hyperbolic.*
- (ii) *The equation $u_t + Lu = f$ is parabolic.*
- (iii) *Let $A(x) \equiv \text{Id}$, then $Lu = -\Delta u + b \cdot \nabla u + cu = f$ is elliptic.*
- (iv) *The wave equation $u_{tt} = u_{xx}$ is hyperbolic.*
- (v) *The heat equation $u_t = u_{xx}$ is parabolic.*

Remark 5.2.3 (i) *The PDEs from finance that we consider here are parabolic. If one uses a discretization in time by means of finite differences, the numerical solution of such problems is reduced to the solution of elliptic PDEs.*

- (ii) *The treatment of hyperbolic PDEs needs different techniques due to the presence of shocks, rarefaction waves etc.*

5.3 Variational Formulation of Elliptic PDEs

Introducing suitable discretizations in time, we are left with the treatment of elliptic PDEs. As we have seen before, the classical (i.e., pointwise) formulation of PDEs might not be the appropriate one. Note that the above mentioned examples are in fact elliptic. Hence, we introduce the weak (or variational) formulation of elliptic PDEs in this section.

For $u, v \in L_2(\Omega)$, $\Omega \subset \mathbb{R}^d$ open with piecewise smooth boundary, denote by

$$(u, v)_0 := \int_{\Omega} u(x) v(x) dx$$

the standard inner product in $L_2(\Omega)$ and denote by $\|u\|_0 := \sqrt{(u, u)_0}$ the induced norm.

We start by the definition of weak derivatives.

Definition 5.3.1 *A function $u \in L_2(\Omega)$ has the (weak) derivative $v = \partial^\alpha u$ of order $\alpha \in \mathbb{N}^d$, ($\alpha = (\alpha_1, \dots, \alpha_d)$, $|\alpha| := \alpha_1 + \dots + \alpha_d$) in $L_2(\Omega)$ if $v \in L_2(\Omega)$ and*

$$(\phi, v)_0 = (-1)^{|\alpha|} (\partial^\alpha \phi, u)_0 \quad \forall \phi \in C^\infty(\Omega) . \quad (5.3)$$

Next, similar to the classical smoothness spaces $C^m(\Omega)$ we define spaces of weakly differentiable functions as follows.

Definition 5.3.2 *For $m \geq 0$, $m \in \mathbb{N}$ we denote by $H^m(\Omega)$ the space of all functions $u \in L_2(\Omega)$ such that $\partial^\alpha u \in L_2(\Omega)$ for all $\alpha \in \mathbb{N}^d$ such that $|\alpha| \leq m$. This space is called Sobolev space of order m . The norm in $H^m(\Omega)$ is defined by*

$$\|u\|_m := \sqrt{(u, u)_m}, \quad (u, v)_m := \sum_{|\alpha| \leq m} (\partial^\alpha u, \partial^\alpha v)_0. \quad (5.4)$$

The following theorem that we quote from [1, 2] without proof shows that any $H^m(\Omega)$ -function can be approximated by smooth functions to any desired accuracy.

Theorem 5.3.3 *The space $H^m(\Omega) \cap C^\infty(\Omega)$ is dense in $H^m(\Omega)$.*

Definition 5.3.4 *The closure of $C_0^\infty(\Omega)$ (the space of arbitrarily smooth functions with compact support in Ω) with respect to the Sobolev norm $\|\cdot\|_m$ is called $H_0^m(\Omega)$.*

Roughly speaking, $H_0^m(\Omega)$ contains those functions in $H^m(\Omega)$ with generalized homogeneous boundary conditions on the boundary $\partial\Omega$. Note that $H_0^m(\Omega)$ is a subspace of $L_2(\Omega)$, thus point values are not well-defined. Thus one cannot talk about boundary conditions in the classical sense, namely pointwise. Generalized boundary conditions are defined by means of the so-called *trace operator*, for details see [1, 2].

Let us now consider the following **model problem**

$$-\Delta u = f \text{ in } \Omega, u = 0 \text{ on } \partial\Omega.$$

For a test function $\phi \in C_0^\infty(\Omega)$ we obtain by integration by parts

$$\begin{aligned} (f, \phi)_0 &= (-\Delta u, \phi)_0 = \sum_{i=1}^d \left(-\frac{\partial^2}{\partial x_i^2} u, \phi \right)_0 \\ &= \sum_{i=1}^d \left(\frac{\partial}{\partial x_i} u, \frac{\partial}{\partial x_i} \phi \right)_0 \\ &= (\nabla u, \nabla \phi)_0 =: a(u, \phi). \end{aligned}$$

Then, $u \in H_0^1(\Omega)$ is called a *weak solution* of the model problem, if

$$a(u, v) = (f, v)_0$$

holds for all $v \in H_0^1(\Omega)$.

Remark 5.3.5 (i) The bilinear form $a(\cdot, \cdot)$ is symmetric, i.e., $a(u, v) = a(v, u)$, positive, i.e., $a(u, v) \geq 0$ and $a(u, u) > 0$ if $u \neq 0$, bounded, i.e., $|a(u, v)| \leq C \|u\|_1 \|v\|_1$ for all $u, v \in H^1(\Omega)$ and coercive in $H_0^1(\Omega)$, i.e.,

$$a(u, u) \geq \alpha \|u\|_1^2.$$

The latter equation is a consequence of the Poincaré-Friedrichs inequality.

(ii) The existence of a weak solution follows from the following characterization theorem:

The linear functional $J(v) := \frac{1}{2}a(v, v) - (f, v)_0$ has its minimum in u if and only if

$$a(u, v) = (f, v)_0 \quad \forall v \in H_0^1(\Omega) .$$

(iii) A bilinear form $a : H \times H \rightarrow \mathbb{R}$ on a Hilbert space H is called continuous if it is bounded. A symmetric and continuous bilinear form is called elliptic if it is coercive.

(iv) **Lax-Milgram theorem:** Let $V \subset H$ be a closed and convex subset and let $a(\cdot, \cdot)$ be an elliptic bilinear form. Then, the variational problem

$$J(v) := \frac{1}{2}a(v, v) - \langle \ell, v \rangle \rightarrow \min!$$

has a unique solution in V for any $\ell \in H'$.

(v) For the special case $H = L_2(\Omega)$ and $V = H_0^1(\Omega)$ we obtain the existence of a weak solution for the general elliptic PDE.

(vi) If $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$ is a classical solution of $Lu = f$, $u|_{\partial\Omega} = 0^1$, then $u \in H_0^1(\Omega)$ is also a weak solution. On the other hand, if $u \in H_0^1(\Omega)$ is a weak solution and in addition $u \in C^2(\Omega) \cap C^0(\bar{\Omega})$, then it is also a classical solution.

(vii) **Reduction to homogeneous boundary conditions (homogenization):** Consider the boundary value problem (bvp) $Lu = f$, $u|_{\partial\Omega} = g$. Choose some $u_0 \in H^1(\Omega)$ such that $u_0|_{\partial\Omega} = g$ and consider the homogeneous problem

$$Lw = f_1 := f - Lu_0, \quad w|_{\partial\Omega} = 0.$$

Then $u := w + u_0$ satisfies $u|_{\partial\Omega} = u_0|_{\partial\Omega} = g$ as well as

$$Lu = Lw + Lu_0 = f - Lu_0 + Lu_0 = f,$$

i.e., u is a solution of the original bvp.

5.4 Ritz-Galerkin methods

The introduced variational formulation is a problem posed in a Hilbert space, in our particular case in a function space which is usually of *infinite dimension*. Thus we cannot treat this problem directly on a computer. The idea, which goes back to Ritz (1908) is to replace the infinite-dimensional minimization problem in the Hilbert space by a finite one using finite-dimensional subspaces $S_h \subset V$ as trial- and test spaces.

¹When we write $u|_{\partial\Omega}$ we always mean the boundary values in the sense of the trace operator, [1, 2].

Let $S_h \subset V$ be a finite-dimensional subspace. We consider the problem of determining $u_h \in S_h$ such that

$$a(u_h, v_h) = (f, v_h)_0 \quad \forall v_h \in S_h. \quad (5.5)$$

By the above remarks, this problem has a unique solution due to the properties of the bilinear form $a(\cdot, \cdot)$. If $\{\psi_1, \dots, \psi_N\}$ is a basis for S_h , $N = \dim S_h$, (5.5) is equivalent to the following problem: Determine $z_h = (z_1, \dots, z_N) \in \mathbb{R}^N$ such that

$$A_h z_h = b_h \quad (5.6)$$

where $A_h = \left(a(\psi_k, \psi_i) \right)_{i,k=1,\dots,N} \in \mathbb{R}^{N \times N}$ is called the *stiffness matrix* and $b_h = \left((f, \psi_k)_0 \right)_{k=1,\dots,N} \in \mathbb{R}^N$ is the right-hand side.

Lemma 5.4.1 *The stiffness matrix A_h is symmetric positive definite.*

Proof: Using bi-linearity and coercivity yields

$$\begin{aligned} d_h^T A d_h &= \sum_{i,k=1}^N d_i A_{i,k} d_k \\ &= a \left(\sum_{k=1}^N d_k \psi_k, \sum_{i=1}^N d_i \psi_i \right) \\ &= a(v_h, v_h) \geq \alpha \|v_h\|^2. \quad \square \end{aligned}$$

Remark 5.4.2 *If the basis $\{\psi_1, \dots, \psi_N\}$ is local in the sense that*

$$\#\{k = 1, \dots, N : |\text{supp } \psi_k \cap \text{supp } \psi_i| > 0\} \leq c \ll N$$

independent of i , then A_h is also sparse which means that the number of non-zero elements per row and column is $\mathcal{O}(1)$.

We now study the convergence of Ritz-Galerkin methods. The following central statements shows that the Galerkin solution is as good as the best approximation to the (unknown) solution out of the trial space S_h . In that sense, the method is optimal.

Theorem 5.4.3 (Ceá's lemma) *Let the bilinear form $a(\cdot, \cdot)$ be elliptic on V , where $H_0^1(\Omega) \subseteq V \subseteq H^1(\Omega)$ and denote the solutions of the variational*

problem in V and $S_h \subset V$ by u and u_h , respectively. Then, we have the following inequality

$$\|u - u_h\|_1 \leq \frac{C}{\alpha} \inf_{v_h \in S_h} \|u - v_h\|_1 .$$

Proof: We have

$$\begin{aligned} a(u, v) &= (f, v)_0 \quad \forall v \in V \\ a(u_h, v_h) &= (f, v_h)_0 \quad \forall v_h \in S_h , \end{aligned}$$

which implies the so-called *Galerkin orthogonality*

$$a(u - u_h, v_h) = 0 \quad \forall v_h \in S_h .$$

Let $v_h \in S_h$, so that with $w_h := v_h - u_h \in S_h$ we obtain

$$a(u - u_h, v_h - u_h) = 0$$

and by coercivity and boundedness

$$\begin{aligned} \alpha \|u - u_h\|_1^2 &\leq a(u - u_h, u - u_h) \\ &= a(u - u_h, u - v_h) + \underbrace{a(u - u_h, v_h - u_h)}_{=0} \\ &\leq C \|u - u_h\|_1 \|u - v_h\|_1 \end{aligned}$$

which proves the theorem. \square

Sometimes this method is also simply called *Galerkin method* and the solution $u_h \in S_h$ is called the *Galerkin solution* or *Galerkin approximation*. The finite dimensional space S_h is also called *trial space*.

5.5 Some Simple Finite Elements

Again, we consider the homogeneous model problem

$$-\Delta u = f \text{ in } \Omega \quad u = 0 \text{ on } \partial\Omega$$

where $\Omega = (0, 1)^2$. The idea is to subdivide Ω into regular triangles of meshsize h as indicated in Figure 5.3. Using this *triangulation* of Ω , we define the trial space

$$S_h := \{v \in C(\bar{\Omega}) : v \text{ is linear in each triangle and } v|_{\partial\Omega} = 0\}.$$

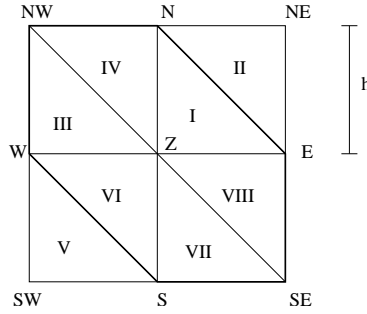


Figure 5.3: Support of the Courant Finite Element.

In each triangle (*element*) a $v_h \in S_h$ takes the form

$$v_h(x, y) = a + bx + cy, \quad a, b, c \in \mathbb{R},$$

i.e., v_h is uniquely determined by the value at three different nodes. From this we can easily see that

$$N = \dim S_h = \text{number of interior grid points}$$

and $v_h \in S_h$ is globally (i.e., on $\bar{\Omega}$) determined by its values on these N grid points. The canonical (so-called *nodal*) basis $\{\psi_i\}_{i=1, \dots, N}$ is defined by

$$\psi_i(x_j, y_j) = \delta_{ij}.$$

In Figure 5.3, the support of ψ_Z is shown. Let h denote the mesh size of the triangles, then we obtain the following values for the derivatives of ψ_Z (see Figure 5.3).

triangle	I	II	III	IV	V	VI	VII	VIII
$\partial_x \psi_Z$	$-h^{-1}$	0	h^{-1}	0	0	h^{-1}	0	$-h^{-1}$
$\partial_y \psi_Z$	$-h^{-1}$	0	0	$-h^{-1}$	0	h^{-1}	h^{-1}	0

Then, straightforward calculations show that by symmetry we have

$$\begin{aligned}
a(\psi_Z, \psi_Z) &= \int_{\text{I-VIII}} (\nabla\psi_Z)^2 dx dy \\
&= 2 \int_{\text{IUIIIUIV}} [(\partial_x\psi_Z)^2 + (\partial_y\psi_Z)^2] dx dy \\
&= 2h^{-2} \left\{ \underbrace{\int_{\text{IUIII}} dx dy}_{=h^2} + \underbrace{\int_{\text{IUIV}} dx dy}_{=h^2} \right\} \\
&= 4
\end{aligned}$$

for the diagonal entries of the stiffness matrix and

$$\begin{aligned}
a(\psi_Z, \psi_N) &= \int_{\text{IUIV}} \nabla\psi_Z \cdot \nabla\psi_N dx dy \\
&= \int_{\text{IUIV}} (\underbrace{\partial_x\psi_Z\partial_x\psi_N}_{=0} + \partial_y\psi_Z\partial_y\psi_N) dx dy \\
&= -h^{-2} \underbrace{\int_{\text{IUIV}} dx dy}_{=h^2} \\
&= -1 = a(\psi_Z, \psi_O) \\
&= a(\psi_Z, \psi_S) = a(\psi_Z, \psi_W).
\end{aligned}$$

Finally, we obtain

$$\begin{aligned}
a(\psi_Z, \psi_{NW}) &= \int_{\text{IIIUIV}} (\partial_x\psi_Z\partial_x\psi_{NW} + \partial_y\psi_Z\partial_y\psi_{NW}) dx dy \\
&= \int_{\text{III}} h^{-1} \cdot 0 + 0 \cdot h^{-1} + \int_{\text{IV}} 0 \cdot (h^{-1}) + (-h^{-1})0 = 0
\end{aligned}$$

and again by symmetry

$$a(\psi_Z, \psi_{SO}) = a(\psi_Z, \psi_{SW}) = a(\psi_Z, \psi_{NO}).$$

This means that in this particular case of a uniform triangulation and the nodal basis for piecewise linear finite elements the stiffness matrix coincides with the matrix arising from the 5-point-stencil in finite difference methods. This principle, however, is not true in general, i.e., for a finite element discretization there is in general *not* an equivalent finite difference discretization. Finite elements are much more flexible than finite differences and they allow the treatment of the weak formulation of the bvp.

Some properties of finite elements

- 1.) Subdivision (or *partition*) of Ω in triangular or quadrilateral elements. If all elements are congruent, this is called a *regular* subdivision.
- 2.) In 2D, we denote by

$$\mathcal{P}_t := \left\{ u(x, y) = \sum_{i+k \leq t; i, k \geq 0} c_{i,k} x^i y^k \right\}$$

the set of all algebraic polynomials of degree at most t . The restriction of the trial (or *shape*) functions to an element is a polynomial.

- 3.) Smoothness: A finite element is said to be of *order* k if it is in $C^k(\Omega)$.

For the example of the *Courant Finite Element* which is shown in Figure 5.3, we have $t = 1$ and $k = 0$.

Definition 5.5.1 (i) A partition $\mathcal{T} = \{T_1, \dots, T_M\}$ of Ω in triangular or quadrilateral elements is called *admissible*, if

- a) $\bar{\Omega} = \bigcup_{i=1}^M T_i$
- b) If $T_i \cap T_j$ consists of exactly one point, this point is a corner of both T_i and T_j .
- c) If $T_i \cap T_j$, $i \neq j$ consists of more than one point, then $T_i \cap T_j$ is a common edge of T_i and T_j .

(ii) We write \mathcal{T}_h if each element has a diameter of at most $2h$.

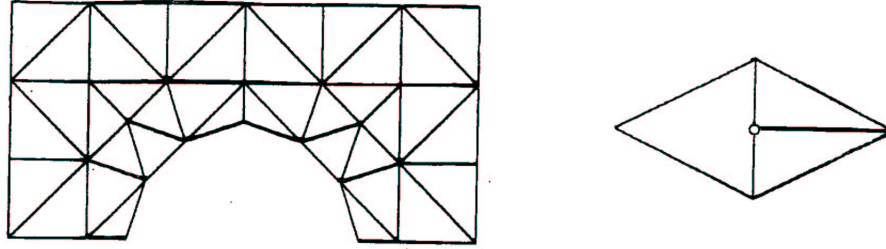


Figure 5.4: An admissible triangulation (left), non-admissible triangulation (right) with a hanging node. (Taken from [1].)

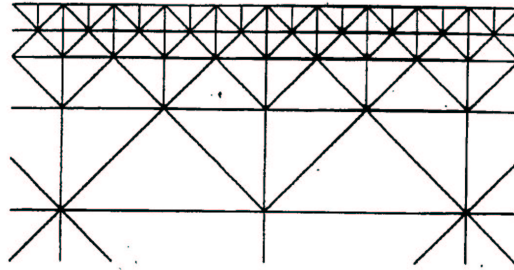


Figure 5.5: A quasi-uniform but non-uniform triangulation. (Taken from [1].)

(iii) \mathcal{T}_h is called quasi-uniform if there exists some $\kappa > 0$ such that each $T \in \mathcal{T}_h$ contains a circle with radius

$$\rho_T \geq \frac{h_T}{\kappa}.$$

Examples of triangulations are shown in Figures 5.4-5.6.

The following theorem shows how to choose the order of the elements in order to be contained in a certain Sobolev space. If the finite elements are contained in the Sobolev space corresponding to the variational formulation, the elements are called *conforming*, otherwise *non-conforming*. For the elliptic second order problem this would mean that the elements are conforming if $S_h \subset H_0^1(\Omega)$.

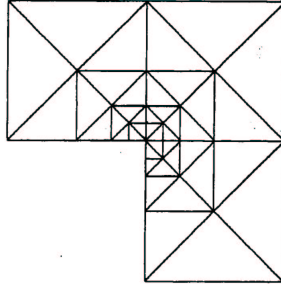


Figure 5.6: A non-uniform triangulation at a reentrant corner. (Taken from [1].)

Theorem 5.5.2 *Let $k \geq 1$ and Ω be bounded. A piecewise C^∞ -function $v : \bar{\Omega} \rightarrow \mathbb{R}$ is in $H^k(\Omega)$ if and only if $v \in C^{k-1}(\bar{\Omega})$. \square*

The latter theorem in particular implies that for the elliptic second order problem we would have $k = 1$ thus $v \in C^0(\bar{\Omega})$ which in particular shows that the Courant Finite Element is conforming.

Definition 5.5.3 *For any finite element space there is a set of points in the sense that the shape functions are uniquely defined by the values at these points. Those functions that take the value 1 at exactly one of these points and 0 on all the others are called nodal basis functions or Lagrange elements.*

Table 5.1 shows a number of standard finite elements. We also show some higher order elements in which the point values are not sufficient to define a shape function uniquely. Also certain derivatives are needed in that case.

Remark 5.5.4 *With the aid of affine mappings, one can usually reduce oneself to one single reference element T_{ref} , i.e., for any $T_j \in \mathcal{T}$ there exists an affine mapping $F_j : T_{\text{ref}} \rightarrow T_j$ such that*

$$v_h(x)|_{T_j} = p(F_j^{-1}x)|_{T_j}, \quad p \in \mathcal{P}_{\text{ref}}, \quad v_h \in S_h.$$

5.6 Approximation Results

In this section, we give some results concerning the approximation properties of the finite element method and also derive some error estimates.

- value of the function
- ⊙ value of the function, 1st and 2nd derivative
- ⊥ normal derivative

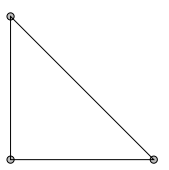
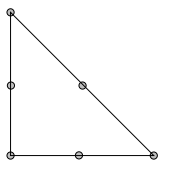
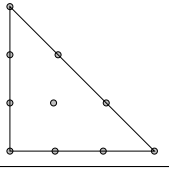
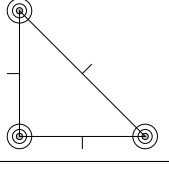
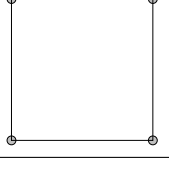
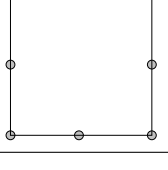
	<p>Linear triangular element</p> <p>$u \in C^0(\Omega)$</p> <p>$\Pi_{\text{ref}} = \mathcal{P}_1, \dim \Pi_{\text{ref}} = 3$</p>
	<p>Quadratic triangular element</p> <p>$u \in C^0(\Omega)$</p> <p>$\Pi_{\text{ref}} = \mathcal{P}_2, \dim \Pi_{\text{ref}} = 6$</p>
	<p>Cubic triangular element</p> <p>$u \in C^0(\Omega)$</p> <p>$\Pi_{\text{ref}} = \mathcal{P}_3, \dim \Pi_{\text{ref}} = 10$</p>
	<p>Agyris element</p> <p>$u \in C^1(\Omega)$</p> <p>$\Pi_{\text{ref}} = \mathcal{P}_5, \dim \Pi_{\text{ref}} = 21$</p>
	<p>Linear quadrilateral element</p> <p>$u \in C^0(\Omega)$</p> <p>$\Pi_{\text{ref}} = \mathcal{P}_2, u _{\partial T_i} \in \mathcal{P}_1, \dim \Pi_{\text{ref}} = 4$</p>
	<p>Quadrilateral serendipity element</p> <p>$u \in C^1(\Omega)$</p> <p>$\Pi_{\text{ref}} = \mathcal{P}_3, u _{\partial T_i} \in \mathcal{P}_2, \dim \Pi_{\text{ref}} = 8$</p>

Table 5.1: Standard Finite Elements. (Taken from [1].)

Definition 5.6.1 For any partition $\mathcal{T}_h = \{T_1, \dots, T_M\}$ of Ω and $m \geq 1$, we define the grid norm by

$$\|v\|_{m,h} := \left(\sum_{T_j \in \mathcal{T}_h} \|v\|_{m,T_j}^2 \right)^{1/2}.$$

Obviously, we have $\|v\|_{m,h} = \|v\|_{m,\Omega}$ for $v \in H^m(\Omega)$.

The following well-known theorem is a central statement in functional analysis.

Theorem 5.6.2 (Bramble-Hilbert theorem) Let $\Omega \subset \mathbb{R}^2$ be a domain with Lipschitz-continuous boundary, $t \geq 2$ and let $L : H^t(\Omega) \rightarrow Y$ be a linear, bounded operator on a normed space Y . If $(\Omega) \subset \text{Ker}(L)$, then we have

$$\|Lv\|_Y \leq c|v|_t$$

for all $v \in H^t(\Omega)$ with a constant $c = c(L, \Omega)$. \square

We now apply this theorem to the interpolation operator

$$I_h : H^t(\Omega) \rightarrow S_h,$$

which is defined by interpolation of the input function with respect to the nodal grid points of the underlying triangulation. Then, we immediately obtain the following error estimate.

Theorem 5.6.3 Let $t \geq 2$ and \mathcal{T}_h be a quasi-uniform triangulation of Ω . Then, we have for the interpolation operator I_h defined by interpolation with piecewise polynomials of degree $t - 1$ that

$$\|u - I_h u\|_{m,h} \leq ch^{t-m}|u|_{t,\Omega}$$

for $u \in H^t(\Omega)$, $0 \leq m \leq t$ and some constant $c = c(\Omega, \mathcal{T}_h, t)$. \square

The principle behind the latter theorem can be roughly described as ‘polynomial exactness implies approximation power’. This is also known as *Bramble-Hilbert type argument*.

Finally, we give an *a priori* error estimate which also shows the continuous dependence of the error on the right-hand side data.

Theorem 5.6.4 *Let \mathcal{T}_h be a family of quasi-uniform triangulations of a convex domain Ω . Then, the piecewise linear finite element approximation $u_h \in S_h$ satisfies the following estimate*

$$\|u - u_h\|_1 \leq ch\|u\|_2 \leq ch\|f\|_0 . \quad \square$$

Remark 5.6.5 (i) *The above regularity assumption $u \in H^2(\Omega)$ can also be weakened.*

(ii) *The computation of an appropriate triangulation is a problem on its own, in particular for complex geometries Ω .*

(iii) *The setup of the linear system (stiffness matrix and right-hand side) is usually done on the reference element.*

5.7 Example: 1D Finite Element Discretization for the Black-Scholes Equation

A variational formulation in 1D reads $a(u, v) = (f, v)_0$ for all $v \in H_0^1(\Omega)$, where $\Omega = (0, 1)$ and (e.g.)

$$a(u, v) = (u', v')_{0,(0,1)} + (u', v)_{0,(0,1)} + (u, v)_{0,(0,1)}.$$

We now subdivide $(0, 1)$ uniformly by equidistant grid points $x_h^k := kh$, where $h = \frac{1}{M}$, $k = 0, \dots, M$, i.e., we have $M - 1$ interior nodes. An ‘element’ in 1D is hence a subinterval (x_h^{k-1}, x_h^k) , $k = 1, \dots, M$. Next, we consider the nodal basis as shown in Figure 5.7. We now apply this for the Black-Scholes equation for European option pricing

$$\frac{\partial}{\partial t} u(t, x) + \frac{1}{2} \sigma^2 x^2 \frac{\partial^2}{\partial x^2} u(t, x) + rx \frac{\partial}{\partial x} u(t, x) - ru(t, x) = 0,$$

where u is the value of the option, x is the asset price, r is the interest rate and σ is the volatility. We consider the call, i.e., the termination condition

$$u(T, x) = \max(K - x, 0).$$

We rewrite this in coefficient form

$$\frac{\partial}{\partial t} u(t, x) + \frac{\partial}{\partial x} \left(c(x) \frac{\partial}{\partial x} u(t, x) \right) + b(x) \frac{\partial}{\partial x} u(t, x) - r u(t, x) + a d_a u(t, x) = 0,$$

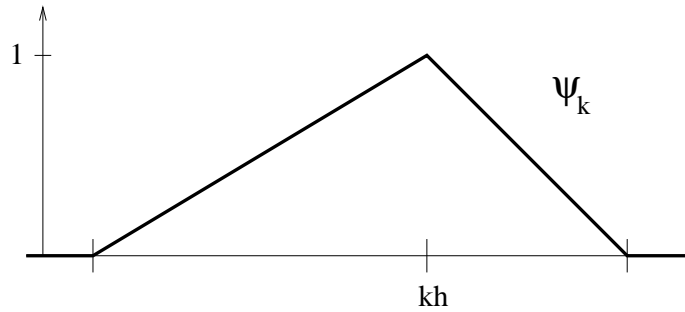


Figure 5.7: 1D nodal piecewise linear shape function.

which is required for FEMLAB

$$\frac{\partial}{\partial t}u(t, x) + \frac{\partial}{\partial x} \left(\frac{1}{2} \sigma^2 x^2 \frac{\partial}{\partial x} u(t, x) \right) + \left(rx - \frac{\partial}{\partial x} \left(\frac{1}{2} \sigma^2 x^2 \right) \right) \frac{\partial}{\partial x} u(t, x) - ru(t, x) = 0,$$

i.e.,

$$\begin{aligned} c &:= \frac{1}{2} \sigma^2 x^2 \\ b &:= (r - \sigma^2)x = rx - \frac{\partial}{\partial x} \left(\frac{1}{2} \sigma^2 x^2 \right) \\ a &:= r, \\ d_a &:= -1. \end{aligned}$$

In this form, we can immediately solve this equation in FEMLAB and we show the computed solution for $K = 40$ (initial price), $\sigma = 0.3$, $r = 0.12$, $x = 80$ and $T = 12$ in Figure 5.8

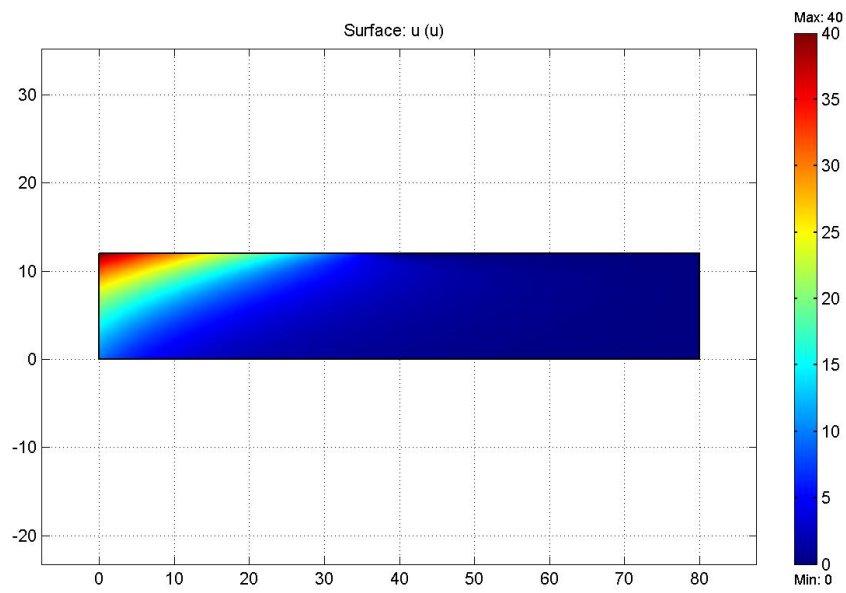


Figure 5.8: Result u of a numerical simulation using FEMLAB. The horizontal axis corresponds to x , the vertical to t .

Bibliography

- [1] D. Braess, *Finite Elemente*, Springer, 1997.
- [2] D. Braess, *Finite Elements*, Cambridge University Press, Cambridge, 2001.
- [3] C. W. Clenshaw and A. R. Curtis, *Numer. Math.* **2** (1960), 197–205.
- [4] C. Großmann and H.-G. Roos, *Numerik partieller Differentialgleichungen*, Second edition, Teubner, Stuttgart, 1994.
- [5] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM 1992.
- [6] K. Petras, *Numerische Methoden in der Finanzmathematik*, Manuskript, TU Braunschweig, <http://www-public.tu-bs.de:8080/~petras/lva/finanz/vorl.html>
- [7] Ch. Schwab., R.A. Todor, *Sparse Finite Elements for Elliptic Problems with Stochastic Loading*, erscheint in *Numer. Mathematik*, 2003.
- [8] R. Seydel, *Tools for Computational Finance*, Springer 2002.
- [9] S. Stojanovic, *Computational Financial Mathematics Using MATHEMATICA*, Birkhauser, 2003.