

# Numerical Finance II

Prof. Dr. Karsten Urban

Universität Ulm  
Abteilung Numerik  
Wintersemester 2003/2004

# Contents

<b>Preface</b>	<b>2</b>
<b>1 Numerical Optimization</b>	<b>3</b>
1.1 Global and Local Optimization . . . . .	5
1.2 Direct Methods . . . . .	6
1.3 Descent Methods . . . . .	7
1.4 Optimization with Side Conditions (Constrained Optimization) . . . . .	10
<b>2 Numerical Methods for Integral Equations</b>	<b>15</b>
2.1 Classification of Integral Equations (IE) . . . . .	16
2.2 Volterra Integral Equations . . . . .	18
2.2.1 Numerical Solution using Quadrature . . . . .	20
2.2.2 Collocation Methods . . . . .	24
2.3 Fredholm Integral Equations of 2nd Kind . . . . .	25
2.3.1 Some Mathematical Theory . . . . .	26
2.3.2 Numerical Methods . . . . .	27
2.3.3 Discretization by Kernel-Approximation . . . . .	30
2.4 Projection Methods . . . . .	32
<b>3 American Option Pricing</b>	<b>36</b>
3.1 An Integral Formulation . . . . .	38
3.2 The Binomial Method . . . . .	42
3.3 Obstacle Problems . . . . .	43
3.4 Finite Difference Methods . . . . .	46
3.4.1 Classical Iterative Methods . . . . .	47
3.4.2 Projected SOR-method for Complementary Problems . . . . .	52

## Preface

This manuscript corresponds to a slightly extended version of my notes of the lecture *Numerical Finance II* that I taught at the University of Ulm in the winter term 2003/04. It was a lecture of 2 hours per week accompanied with exercises (also 2 hours per week) that were given by Michael Lehn. This lecture was given also within the Master programme *Finance* and hence all lectures and exercises were in English.

This manuscript is far from being complete and the lecture as well as this manuscript is not much more than a first iteration through various fields of interest in computational finance. For further details, I refer the reader to the quoted literature. Moreover, I am grateful for any kinds of comments, criticism or corrections. Additional material concerning the lecture can be found in the web under

<http://www.mathematik.uni-ulm.de/numerik/teaching/ws03/NumericalFinance2>

Finally, I wish to thank Michael Lehn for his excellent work as an assistant for this lecture and Timo Tonn who acted as tutor mainly for the foreign students in the Master programme. I am grateful to my colleague Prof. Dr. Hans-Joachim Zwiesler for providing me with nice examples and to the students Oliver Pauly, Johannes Ruf and others attending the class who gave several helpful remarks and Petra Hildebrand who did the sometimes hard job of typesetting this manuscript.

# Chapter 1

## Numerical Optimization

There are many examples of optimization problems in several areas of application

- Science (physics, chemistry):  
As an example think of a chemical process (e.g. a catalysis) where a maximal amount of a certain species (e.g. a high-quality oil) should be produced). Hence this is also a problem relevant in economy.
- Engineering (optimal flow, optimal control):  
In modern airports it is an important task to minimize the vortices that occur during the landing of an aircraft since the length of such a vortex street determines the minimal distance of two succeeding landing aircrafts. This, in turns, maximizes the profit of the airport company.
- Finance (portfolio, optimization):  
A portfolio or investment strategy should be designed in such a way to maximize the expected profit or to minimize a risk etc.

Many other examples can easily be derived. It is already clear from this short list that optimization of course is a relevant issue in numerical finance. Moreover, it is also clear that *optimization* and *minimization* are two sides of the same medal so that we will subsume both under the term *optimization*.

On the other hand, the above list also shows that optimization problems often are combined with other sometimes very difficult problems. E.g., the minimization of the vortex street requires to compute the flow of the air which is induced by a landing aircraft. This requires to solve (numerically) a non-linear, instationary 3D partial differential equation, which is far from being trivial.

Numerical optimization is a wide field also of very active research. There are big text books and monographs only concerned with special topics from numerical optimization. This shows that we can only highlight some aspects of numerical optimization within this lecture. We refer to the literature as well as to lectures on numerical optimization for further information.

**Example 1.0.1 (A Transport Problem)** *Let us start with an easy example from chemistry that also shows some of the features that are relevant for the numerical treatment of such problems. Let us assume that a chemical company has 2 factories  $F_1, F_2$  and two retail outlets  $R_1, \dots, R_{12}$  with the following configuration:*

- *Each factory  $F_i$  can produce  $a_i$  tons of a product per week,  $i = 1, \dots, 2$ . Here,  $a_i$  can be interpreted as a ‘capacity’ of the factory  $F_i$ .*
- *Each retail outlet  $R_j$  has a weekly demand of  $b_j$  tons.*
- *There are shipping costs for bringing one ton from the factory  $F_i$  to the retail outlet  $R_j$  of  $c_{ij}$  currency units.*

*The problem is now to determine an/the optimal strategy in order to minimize the overall cost. In order to model this problem mathematically, let  $x_{ij}$ ,  $i = 1, 2$ ,  $j = 1, \dots, 12$  denote the number of tons shipped from factory  $F_i$  to retail outlet  $R_j$ . Hence, we can view a strategy as the matrix  $X = (x_{ij}) \in \mathbb{R}^{2 \times 12}$ . Thus, we obtain the following minimization problem:*

$$\min \sum_{i,j} c_{ij} x_{ij} \tag{1.1}$$

*subject to the constraints*

$$\begin{aligned} \sum_{j=1}^{12} x_{ij} &\leq a_i, & i = 1, 2 \\ \sum_{i=1}^2 x_{ij} &\geq b_j, & j = 1, \dots, 12 \\ x_{i,j} &\geq 0 & \forall i, j. \end{aligned} \tag{1.2}$$

□

The problem (1.1), (1.2) is a linear minimization problem with the (inequality) constraints (1.2). It is called ‘linear’ because the mapping  $f$  relating the degrees of freedom  $X$  with the value subject to the optimization is linear. In fact, we have

$$f : \mathbb{R}^{2 \times 12} \rightarrow \mathbb{R}, \quad f(X) := \sum_{i,j} c_{ij} x_{ij}, \quad X = (x_{i,j})_{i=1,2; j=1, \dots, 12}.$$

This function  $f$  is called *objective function* and is in this case obviously linear.

In this section, we give a brief introduction to numerical methods in order to solve optimization problems.

## 1.1 Global and Local Optimization

A first obvious category of optimization problems is if one is interested in *the* maximum over all possible strategies or if one is satisfied with a strategy which is better as ‘many’ other strategies. The first problem is called *global*, the second one *local* optimization. Let us fix some notation first.

The problem of global optimization under consideration can be formulated as follows

**Problem 1.1.1** *Given a nonempty, closed set  $D \subset \mathbb{R}^n$  and  $f : A \rightarrow \mathbb{R}$ ,  $A \subset \mathbb{R}^n$ ,  $D \subseteq A$ . Find at least one point  $x^* \in D$  such that*

$$f(x^*) \leq f(x) \quad \forall x \in D ,$$

*or show that such a point, the global minimum does not exist.*

Often Problem 1.1.1 is abbreviated as

$$\min_{x \in D} f(x) \quad \text{or} \quad f(x) \rightarrow \min! \tag{1.3}$$

**Remark 1.1.2** *As already mentioned before, optimization and minimization problems are equivalent because of*

$$\max_{x \in D} f(x) = - \left( \min_{x \in D} (-f(x)) \right).$$

*Hence, all what is said for optimization problems also holds for minimizations problems and vice versa.*

In contrast, let us now turn to local optimization problems and their solutions.

**Definition 1.1.3** *A point  $x^* \in D$  is called local minimizer of  $f$  over  $D$  if there is an  $\varepsilon > 0$  such that*

$$f(x^*) \leq f(x) \quad \forall x \in \mathcal{N}_\varepsilon(x^*) \cap D,$$

*where  $\mathcal{N}_\varepsilon(x^*) := \{x \in \mathbb{R}^n : \|x - x^*\| < \varepsilon\}$  denotes a neighborhood of  $x$ .*

**Remark 1.1.4** (a) *All standard numerical techniques for solving optimization problems are able to determine a local extremum. Determining the global extremum usually requires some kind of an outer iteration.*

(b) *There is no local criterion for deciding if a local solution is global. This makes the design of an outer iteration a delicate task.*

(c) *If the objective function is smooth, i.e., if  $f \in C^1(A)$  and  $D \subset A$  is compact, then the existence of a global minimizer in  $D$  is assured by the Weierstraß theorem.*

(d) *However, in many complex applications,  $f$  is not known explicitly. This is the case e.g. if the objective includes the solution of another problem e.g. a partial differential equation. Thus, often there is no way to investigate the smoothness of  $f$  in practical applications.*

(e) If  $D = A$ , then (1.3) is an unconstrained optimization problem, if  $D \subsetneq A$  then we have a constrained optimization problem.

In Example 1.0.1 we have that

$$D := \{X = (x_{ij})_{ij} \in \mathbb{R}^{2 \times 12} : (1.2) \text{ holds} \}$$

$$\subsetneq A = \mathbb{R}^{2 \times 12},$$

so that (1.1,1.2) is a *constrained minimization problem*.

Often the constraints are modelled in terms of a function  $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $m \leq n$  (the so-called *cost functional*), where

$h(x) = 0$  model equality constraints, and

$h(x) \leq 0$  model inequality constraints (to be understood element-wise).

**Example 1.1.5** *The properties of the objective function  $f$  and of the cost functional  $h$  determine categories of optimization problems. The maybe best-known examples are*

- (a) *If  $f$  is convex and  $h$  has convex components, the optimization problem is known as convex optimization.*
- (b) *If both  $f$  and  $h$  are linear, one talks of linear optimization.*
- (c) *If the objective function  $f$  is quadratic and we have linear constraints  $h$ , this is known as quadratic optimization.*

## 1.2 Direct Methods

Direct methods require only the evaluation of the objective function  $f$ , i.e., no derivatives are needed. This means in particular, that direct methods do not pose regularity assumptions on the objective function. This is in fact a realistic framework in many applications of practical relevance. The counterpart of direct methods are *descent methods*, where a direction of descent is determined (often in terms of the gradient of  $f$ , thus derivatives are usually required). In general, direct methods converge slower than descent methods that will be described later.

We describe here one particular example of a direct method and refer to the literature for further methods, see e.g. [1, 6].

### The Method of Hooke and Jeeves

The method was introduced in 1961 and consists of two steps, namely

1. the *exploration step*, and
2. the *progressing step*,

which will be described in detail next.

## 1. Exploration step

Given  $x^{(0)} \in \mathbb{R}^n$  and a step size  $h_1 \in \mathbb{R}$ , compute

$$f(x^{(0)} + h_1 \mathbf{e}_1),$$

where  $\mathbf{e}_i = (\delta_{1,i}, \dots, \delta_{n,i}) \in \mathbb{R}^n$  denotes the  $i$ -th canonical basis vector. If the new point  $x^{(0)} + h_1 \mathbf{e}_1$  is an improvement, i.e.,  $f(x^{(0)} + h_1 \mathbf{e}_1) < f(x^{(0)})$ , the exploration step was successful and we use  $x^{(0)} + h_1 \mathbf{e}_1$  as new starting point.

Otherwise, check  $x^{(0)} - h_1 \mathbf{e}_1$  in the same way.

This is done for all  $n$  coordinate directions.

## 2. Progressing step

Given an iterate  $x^{(0)} \in \mathbb{R}^n$  and let  $y^{(0)}$  be the output of the exploration step 1. Then, we have to distinguish two cases.

Case 1:  $y^{(0)} = x^{(0)}$ , i.e., the exploration step did not give an improvement.

If the step size is too small, i.e.,  $\max_{i=1, \dots, n} h_i < \varepsilon$ , then STOP

and use  $x^{(0)}$  as output.

Else use the half step size  $h_i \leftarrow \frac{h_i}{2}$  and go to 1.

Case 2: If the exploration step yields  $y^{(0)} \neq x^{(0)}$ .

If the step size is too small, i.e.,  $\max_{i=1, \dots, n} h_i < \varepsilon$ , then STOP

and use  $y^{(0)}$  as output.

Else (Progress) use  $2y^{(0)} - x^{(0)}$  as new starting point for explore  $\rightarrow z$

If  $z = 2y^{(0)} - x^{(0)} \rightarrow x^{(1)} = y^{(0)}$

Else  $x^{(1)} = z$

Obviously, the method produces a non-increasing sequence

$$f(x^{(0)}) \geq f(x^{(1)}) \geq \dots \geq f(x^{(n)}) \geq \dots$$

but this does **not** guarantee convergence of the method. It could very well happen that the method reaches a stationary point  $x^{(k)}$  in which  $f(x^{(k)}) = f(x^{(k+1)}) = \dots$  holds. This is typical for direct methods. Usually, one needs more analytical information (such as derivatives) in a descent method in order to prove convergence.

An alternative direct method is the *simplex method* of Nelder and Mead (1965), [6, pp 318]. This should not be mixed up with the simplex method in linear programming.

## 1.3 Descent Methods

We now consider more sophisticated methods for numerical optimization. They usually require the evaluation of the derivative  $f'$  of the objective function. This, in turns, means that these methods can only be applied for smooth functions. On the other hand, we will see that this additional assumption will yield a proof of convergence. The typical form is:

$$\left\{ \begin{array}{l} \text{Given an initial guess } x^{(0)} \in \mathbb{R}^n \\ \text{For } k = 0, 1, 2, \dots \text{ until convergence} \\ x^{(k+1)} := x^{(k)} + \alpha_k d^{(k)}, \end{array} \right. \quad (1.4)$$

where  $\bullet \alpha_k \in \mathbb{R}^+$  is the step size,  
 $\bullet d^{(k)}$  is an appropriate direction (descent or ascent),

which have to be determined during the iteration. We first have to specify what an appropriate descent (or ascent) direction is. This definition obviously depends on the current iterate  $x^{(k)}$ .

**Definition 1.3.1** A vector  $\mathbf{d}^{(k)}$  is called descent direction for  $x^{(k)}$  if

$$\left\{ \begin{array}{ll} (\mathbf{d}^{(k)})^T \nabla f(x^{(k)}) < 0, & \text{if } \nabla f(x^{(k)}) \neq 0 \\ \mathbf{d}^{(k)} = 0, & \text{if } \nabla f(x^{(k)}) = 0. \end{array} \right. \quad (1.5)$$

A method of type (1.4) is called descent method if all  $\mathbf{d}^{(k)}$  are descent directions.

The computation of a suitable step size  $\alpha_k \in \mathbb{R}$  is the next ingredient of the method.

**Lemma 1.3.2** Let  $\mathbf{d}^{(k)}$  be descent directions, then there exist  $\alpha_k > 0$  such that

$$f(x^{(k)} + \alpha_k d^{(k)}) < f(x^{(k)})$$

if  $f \in C^1(D)$  and  $\nabla f(x^{(k)}) \neq 0$  (otherwise the exact solution is found).

**Proof:** Taylor's formula implies the existence of an intermediate point  $\xi^{(k)} \in (x^{(k)}, x^{(k)} + \alpha_k d^{(k)})$  such that

$$\begin{aligned} f(x^{(k)} + \alpha_k d^{(k)}) - f(x^{(k)}) &= \alpha_k \nabla f(\xi^{(k)})^T d^{(k)}, \\ &= \alpha_k \nabla f(x^{(k)})^T d^{(k)} + \alpha_k \varepsilon_k, \end{aligned} \quad (1.6)$$

where  $\varepsilon_k \alpha_k$  tends to zero for  $k \rightarrow \infty$  because of the continuity of  $\nabla f$ . Since the method stops if  $\nabla f(x^{(k)}) = 0$  we assume without loss of generality that  $\nabla f(x^{(k)}) \neq 0$ , thus  $\nabla f(x^{(k)})^T d^{(k)} < 0$ . Hence, the right-hand side of (1.6) is negative, if  $\alpha_k > 0$  is chosen small enough.  $\square$

**Remark 1.3.3** If  $f \in C^1(D)$ , the equation

$$\nabla f(x) = 0$$

is called Euler-Lagrange equation and a solution of it is called critical point. Any local extremum of  $f$  is also a critical point (if  $f$  is smooth). Thus, often numerical methods for  $\nabla f(x) = 0$  are used (Newton if  $f \in C^2(D)$ , bisection, Richardson, ...).

## Examples for the choice of $d^{(k)}$

So far, we have defined a suitable descent direction. It remains to actually compute such a vector. Secondly, it is obvious that  $d^{(k)}$  is not uniquely determined, there are several choices. Some of them are described in the sequel.

### a) **Newton's method**

This corresponds to applying standard Newton's method for the Euler-Lagrange equation  $\nabla f(x) = 0$ . To be precise, let

$$h_{ij}(x) = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \quad i, j = 1, \dots, n$$

and  $H(x) := (h_{ij}(x))_{i,j}$  be the *Hessian matrix* of  $f \in C^2(D)$ .

If  $H(x)$  is s.p.d. (symmetric and positive definite) in  $U_\varepsilon(x^*)$ , then set

$$d^{(k)} := -H^{-1}(x^{(k)})\nabla f(x^{(k)}).$$

This means that  $d^{(k)}$  is determined by numerically solving the linear system of equations  $H(x^{(k)})d^{(k)} = \nabla f(x^{(k)})$  e.g. by the cg method. Note that this implies that in *each iteration* there is the need of solving a linear system of equations. On the other hand, we obtain from the convergence of Newton's method locally second order convergence.

### b) **Inexact Newton's method**

As in the standard Newton's method, one can approximate the gradient by a cheaper approximation losing, however, the second order convergence. Thus, we set

$$d^{(k)} := -B_k^{-1}\nabla f(x^{(k)}),$$

where  $B_k$  is an appropriate approximation of  $H(x^{(k)})$ .

### c) **Gradient method** (also known as *steepest descent*)

Here, we set

$$d^{(k)} := -\nabla f(x^{(k)}).$$

This shows that the Gradient method is an inexact Newton method with  $B_k \equiv I$ . Note that because of

$$(d^{(k)})^T \nabla f(x^{(k)}) = -\|\nabla f(x^{(k)})\|_2^2$$

the vector  $d^{(k)}$  obviously is descent direction.

### d) **Conjugate gradient** (cg)

The cg method corresponds to the choice

$$d^{(k)} := -\nabla f(x^{(k)}) + \beta_k d^{(k-1)},$$

where  $\beta_k \in \mathbb{R}$  will be chosen in such a way that  $\{d^{(k)}\}_k$  are mutually orthogonal with respect to an appropriate inner product.

## The choice of $\alpha_k$

Lemma 1.3.2 guarantees the existence of a suitable step size  $\alpha_k$  such that there is a descent. For a numerical scheme, however, we need to determine this step size. Again, there are several possible choices.

**Line search strategy:** determine  $\alpha$  such that  $\phi(\alpha) = f(x^{(k)} + \alpha d^{(k)}) \rightarrow \text{Min}$ .

This means that  $\alpha_k$  arises from solving a 1d-minimization problem. This can be solved with any standard technique.

**Remark 1.3.4** *If  $\alpha_k$  is determined by an exact line search, one has*

$$0 = \nabla f(x^{(k+1)})d^{(k)} = \nabla f(x^{(k+1)})(x^{(k+1)} - x^{(k)}).$$

**Proof:** Exercise.

The latter condition that the gradient is orthogonal to the search direction, plays a crucial role in the analysis of the method.

Often such  $\alpha$  can **not** be determined exactly since this might be too costly or simply not possible. There are at least two ways to do so:

- Determine an approximation of  $f$  along  $x^{(k)} + \alpha d^{(k)}$  e.g. by a polynomial and minimize this. Typical examples are quadratic interpolation (Powel) and cubic interpolation (Davidon).
- Use an iterative method (bisection, secant, Richardson) for the 1d problem.

## 1.4 Optimization with Side Conditions (Constrained Optimization)

Often one has extra conditions that have to be taken into account, e.g.

- reaction between variables (chemistry, correlation of stochastical variables),
- side conditions (prices are positive e.g.).

The most simple formulation of such a problem is

$$\text{minimize } f(x) \text{ for } x \in \Omega \subseteq \mathbb{R}^n, \tag{1.7}$$

where  $\Omega$  describes the set of *suitable* strategies out of the set  $\mathbb{R}^n$  of all possible strategies. If  $f \in C(\Omega)$  and the set  $\Omega$  is closed and bounded, then (1.7) has a solution by Weierstraß theorem.

**Lemma 1.4.1** *Let  $\Omega \subset \mathbb{R}^n$  be convex,  $x^* \in \Omega$  and  $f \in C^1(\mathcal{N}(x^*; R))$  for some  $R > 0$ . Then we have.*

a) If  $x^*$  is a local minimum of  $f$ , then

$$\nabla f(x^*)^T(x - x^*) \geq 0 \quad \forall x \in \mathcal{N}(x^*; R). \quad (1.8)$$

b) If  $f$  is **convex** in  $\Omega$ , i.e.,

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) \quad \forall x, y \in \Omega$$

and (1.8) holds, then  $x^*$  is a global minimum of  $f$ .

c) If  $f$  is **strictly convex** in  $\Omega$ , i.e.

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y) - \alpha(1 - \alpha)\rho\|x - y\|_2^2$$

for all  $x, y \in \Omega$  and all  $\alpha \in [0, 1]$ , then there exists at most a unique minimum  $x^* \in \Omega$ .

The proof is standard and usually done in basic analysis courses.

## Equality constraints

A standard example of a problem of constraint optimization reads as follows: given an objective function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , then the problem is

$$\begin{cases} \text{minimize } f(x) \\ \text{subject to the constraint } h(x) = 0, \end{cases} \quad (1.9)$$

where  $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $m \leq n$ , is a given function with components  $h_1, \dots, h_m : \mathbb{R}^n \rightarrow \mathbb{R}$ . The goal is to reformulate the constrained optimization problem as an unconstrained one. This is done by means of the so-called *Lagrange function*  $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$

$$\mathcal{L}(x, \lambda) := f(x) + \lambda^T(h(x)), \quad (1.10)$$

where  $\lambda \in \mathbb{R}^m$  is also called *Lagrange Multiplier* (the side conditions are appended by Lagrange Multipliers).

The following result is (or should be) well-known from basic courses on analysis.

**Lemma 1.4.2** a) Let  $x^*$  be a local minimum of (1.9) and assume  $f, h_i \in C^1(\mathcal{N}(x^*, R))$  for some  $R > 0$  and all  $1 \leq i \leq m$ . Then there exists a unique  $\lambda^* \in \mathbb{R}^m$  such that

$$J_{\mathcal{L}}(x^*, \lambda^*) = 0,$$

where  $J_{\mathcal{L}}$  denotes the Jacobian of  $\mathcal{L}$ .

b) If  $x^* \in \mathbb{R}^n$  satisfies  $h(x^*) = 0$ ,  $f, h_i \in C^2(\mathcal{N}(x^*, R))$  for some  $R > 0$  and all  $1 \leq i \leq m$  and if there exists  $\lambda^* \in \mathbb{R}^m$  such that  $J_{\mathcal{L}}(x^*, \lambda^*) = 0$  and

$$z^T H_{\mathcal{L}}(x^*, \lambda^*) z > 0 \quad \forall z \neq 0 \text{ with } \nabla h(x^*)^T z = 0$$

(where  $H_{\mathcal{L}}$  is the Hessian matrix of  $\mathcal{L}$ ), then  $x^*$  is a (strong) local minimum of (1.9).

This lemma shows that it amounts to numerically solve the Euler-Lagrange equations for  $\mathcal{L}$  (which can be performed e.g. by Richardson, fixed point, Newton). Obviously, the dimension of the problem has been increased by appending the constraints. This is the price to be paid for involving constraints. On the other hand, we can use the same numerical methods for the solution. From this point of view, equality constraints are ‘straightforward’.

## Inequality constraints

An example also relevant in Finance (e.g. the valuation problem for American options) is the following.

$$\begin{cases} \text{minimize } f(x) \\ \text{subject to the constraints } h(x) = 0 \text{ and } g(x) \leq 0 \end{cases} \quad (1.11)$$

where the functions  $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $m \leq n$ , and  $g : \mathbb{R}^n \rightarrow \mathbb{R}^r$ ,  $r \leq n$ , are given. Here  $g(x) \leq 0$  is to be understood componentwise, i.e.,  $g_i(x) \leq 0$ ,  $i = 1, \dots, r$ .

Even though this is a more delicate problem than (1.9), we can still transform it by means of the Lagrange function

$$\mathcal{M}(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x), \quad (1.12)$$

where  $\lambda \in \mathbb{R}^m$ ,  $\mu \in \mathbb{R}^r$  are Lagrange multipliers.

The corresponding result reads.

**Lemma 1.4.3** *Let  $x^*$  be a ‘regular’ local minimum of (1.11) (i.e.,  $(J_h(x^*), \nabla g_j(x^*))$  is regular, where  $j \in \mathcal{J}(x^*)$  the set of indices  $j$  such that  $g_j(x^*) = 0$ ).*

*If  $f, h_i, g_j \in C^1(\mathcal{N}(x^*, R))$  for some  $R > 0$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, r$ , then there exist two vectors  $\lambda^* \in \mathbb{R}^m$ ,  $\mu^* \in \mathbb{R}^r$  such that*

$$J_{\mathcal{M}}(x^*, \lambda^*, \mu^*) = 0,$$

*where  $\mu_j^* \geq 0$  and  $\mu_j^* g_j(x^*) = 0 \forall j = 1, \dots, r$ .*

**Remark 1.4.4** (a) *The condition in Lemma 1.4.3 is only a necessary condition (the so-called ‘Kuhn-Tucker-condition’). In general, it is not sufficient, we need extra conditions on  $f$  and  $g$ .*

(b) *If  $f$  is concave and  $(x^*, \lambda^*, \mu^*)$  satisfies a certain Kuhn-Tucker-condition,  $g_i$  is convex for  $\lambda_i^* > 0$ , then  $f(x^*)$  is also the constrained minimizer.*

(c) *Numerically this amounts to solve the Euler-Lagrange equations for  $\mathcal{M}$ . Again, one can use the same numerical methods but for a problem in higher dimension.*

## Penalty methods

Since unconstrained problems are obviously easier to solve than constrained ones, one often tries to transform a constrained problem into an unconstrained problem. The idea of penalty methods is to allow the constraints to be satisfied only approximately and to introduce a measure that shows how well the constraints are satisfied.

To summarize, the main idea is as follows:

- Eliminate the constraints by transforming the constrained problem into an unconstrained one.
- Introduce a parameter which is a measure for the satisfaction of the constraints.

Let us consider again (1.9) and we look for  $x^* \in \Omega \subset \mathbb{R}^n$  (where existence is assumed). Consider the so-called *penalty-Lagrange function*

$$\mathcal{L}_\alpha(x) := f(x) + \frac{1}{2}\alpha\|h(x)\|_2^2, \quad (1.13)$$

where  $\alpha \in \mathbb{R}$  is called *penalty parameter*. Now, we consider the minimization problem

$$\text{minimize } \mathcal{L}_\alpha(x) \text{ for } x \in \Omega. \quad (1.14)$$

Obviously the minimization of  $\mathcal{L}_\alpha$  is equivalent to the minimization of  $\mathcal{L}$  in (1.10) if  $h(x) = 0$  (i.e., the constraints are satisfied exactly).

The penalty method is an iterative method for the solution of (1.13), i.e.

For  $k = 0, 1, 2, \dots$

$$\text{minimize } \mathcal{L}_{\alpha_k}(x) \text{ for } x \in \Omega \quad (1.15)$$

where  $\{\alpha_k\}$  is a monotonically increasing sequence of penalty parameters such that  $\alpha_k \rightarrow \infty$  for  $k \rightarrow \infty$ .

From this we obviously get  $\lim_{k \rightarrow \infty} \mathcal{L}_{\alpha_k}(x) \rightarrow 0$  if  $h(x) \neq 0$ . The general convergence results reads as follows.

**Lemma 1.4.5** *Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ ,  $m \leq n$ , be continuous on a bounded domain  $\Omega \subset \mathbb{R}^n$ .*

*Suppose that  $\alpha_k \nearrow \infty$  ( $k \rightarrow \infty$ ),  $\alpha_k > 0$ . If  $x_k^*$  denotes the solution of (1.15), then*

$$x_k^* \rightarrow x^* \text{ for } k \rightarrow \infty,$$

*where  $x^*$  denotes the solution of (1.11).*

**Proof:** Follows immediately from the Weierstraß theorem.  $\square$

- Remark 1.4.6** (a) *The problem (1.15) becomes ill-conditioned for large values of  $\alpha_k$ , i.e., one observes numerical problems near  $x^*$ . This is often seen as the most severe drawback of penalty methods.*
- (b) *On the other hand, if  $\alpha_k$  increases too slow (i.e., the penalty is weak), we obtain convergence problems in the sense that the speed of convergence may become very slow.*
- (c) *Note that Lemma 1.4.5 gives **no** rate of convergence.*
- (d) *A typical (still heuristic) strategy is to choose some  $\alpha_0 > 0$  and to set  $\alpha_k = \beta\alpha_{k-1}$ , for  $k > 0$ , where  $\beta \in \{4, \dots, 10\}$  is some a priory chosen parameter.*
- (e) *Further exercises can be found in [6, p. 338/339].*

# Chapter 2

## Numerical Methods for Integral Equations

Integral equations arise from several problems in various fields of applications. Let us just mention medicine, in particular computer tomography, shape optimization problems, computer graphics, structural mechanics. Again, the numerical solution of integral equations is a wide field so that we can only cover a small introduction within this lecture. For further information and details, we refer to the literature, e.g. [3].

### An Example from Actuarial Sciences

Also in finance and insurance, there are relevant problems where integral equations occur as the following example (which was kindly be given to me by my colleague Prof. Dr. Hans-Joachim Zwiesler).

Typically an insurance for active working people pays in case of a disability only after a certain waiting period  $f$  (typically 1/4 year). One is interested in the probability  $p_x^{aa}(t, f)$  that an  $x$ -year old active is also active at time  $t$ ,  $0 < f < t$  but was disabled in between for more than  $f$  years (which is the case in which the insurance company has to pay).

The following quantities are assumed to be given:

$p_x^{aa}(u)$	The probability that a person is active at time $u$ independent of any disability in between.
$p_{x+s}^{ia}(t-s)$	The probability that a person becomes active again.
$p_{x+u}^{ii}(f)$	The probability that a person is $f$ years without break disabled.
$\mu_{x+u}^{ai}$	Intensity of persons that become disabled at time $x+u$ .

These are the given parameter, the input. Then, one obtains the following integral equation for the unknown  $p_x^{aa}(t, f)$ :

$$p_x^{aa}(t, f) = \int_0^t (p_x^{aa}(u) - p_x^{aa}(u, f)) \mu_{x+u}^{ai} p_{x+u}^{ii}(f) p_{x+u+f}^{ia}(t-u-f) du.$$

**Remark 2.0.1** Note that any initial value problem (IVP) for a first order ordinary differential equation (ODE) of the form

$$y'(x) = f(x, y) , \quad x \geq x_0 , \quad y(x_0) = y_0$$

can be reformulated as an integral equation as follows

$$y(x) = y_0 + \int_{x_0}^x f(\xi, y(\xi)) \, d\xi \quad \text{for } x \geq x_0.$$

A similar remark also holds for boundary value problems for partial differential equations (PDE). It heavily depends on the particular application if the PDE or the integral equation (IE) formulation is more appropriate. A PDE on a domain in  $\mathbb{R}^n$  typically results in a linear system of equations with a sparse matrix. On the other hand, the IE leads to a problem only on the boundary  $\Gamma = \partial\Omega$  which is a  $n - 1$ -dimensional surface. Thus, the size of the problem is reduced. The price to be paid is usually that the system matrix for the IE is full (densely populated).

## 2.1 Classification of Integral Equations (IE)

One cannot hope to obtain a numerical method that is suitable for all integral equations. This is quite similar to numerical methods for PDEs (see Numerical Finance I). There are numerical schemes that rely on particular properties of certain integral equations. Thus, we first have to describe different classes and categories of integral equations.

From now on, we will frequently use the abbreviation ‘IE’ for integral equation in order to shorten notation.

### Classes of Integral Equations

We distinguish two main classes of integral equations, namely:

- *Fredholm* integral equations:  
The integral in the equations is taken over a *fixed* domain.
- *Volterra* integral equations:  
The integral in the equation is taken over a domain which depends on the variable  $x$  (as e.g. in our example above).

Thus, this is just a distinction based on the domain of integration.

### Types of Integral Equations

This is a classification based upon the appearance of the unknown function.

- IE of *1st kind*:  
The unknown function appears *only* in the integral.
- IE of *2nd kind*:  
The unknown function also appears outside the integral (and, of course, also inside).

## Characters of Integral Equations

Finally, we have a classification with respect to the analytical properties of the integral operator.

- *Regular* IE:  
The integral exists as a proper integral.
- *Weakly singular* IE:  
The integral exists as an improper integral.
- *Strongly singular* IE:  
The integral needs to be defined in a special manner (e.g. by the Cauchy mean value).

Of course, there is also the distinction whether a problem is linear or non-linear. Let us now illustrate the above categories with some examples.

**Example 2.1.1** *In the following examples, the function  $g$  is always assumed to be given and the function  $f$  is sought unknown.*

(a) *Linear Fredholm IE of 2nd kind:*

$$f(x) = g(x) + \int_a^b k(x, y)f(y) dy, \quad x \in [a, b]. \quad (2.1)$$

(b) *Linear Fredholm IE of 1st kind:*

$$g(x) = \int_a^b k(x, y)f(y) dy, \quad x \in [a, b]. \quad (2.2)$$

(c) *Linear Volterra IE of 2nd kind:*

$$f(x) = g(x) + \int_a^x k(x, y)f(y) dy, \quad x \geq a. \quad (2.3)$$

(d) *Linear Volterra IE of 1st kind:*

$$g(x) = \int_a^x k(x, y) f(y) dy, \quad x \geq a. \quad (2.4)$$

(e) *Nonlinear Fredholm IE of 2nd kind*

$$f(x) = g(x) + \int_a^b k(x, y, f(x), f(y)) dy, \quad x \in [a, b], \quad \text{or} \quad (2.5)$$

$$f(x) = F \left( x, \int_a^b k(x, y, f(x), f(y)) dy \right), \quad x \in [a, b]. \quad (2.6)$$

*The function  $k$  typically is called **kernel**.*

(f) *Our example from actuarial sciences is a linear Volterra IE of 2nd kind.*

Now we start with the description of numerical methods for the different types of IE.

## 2.2 Volterra Integral Equations

We consider the equation of 2nd kind in (2.3) for  $x \in I$  where either  $I = [a, b]$  (a finite interval) or  $I = [a, \infty)$  in the following form:

$$f(x) = g(x) + \int_a^x k(x, y, f(y)) dy, \quad x \in I. \quad (2.7)$$

In the sequel, we use the abbreviation

$$G := \{(x, y) : x \in I, y \in [a, x]\} \subset I \times I, \quad (2.8)$$

i.e., the domain of the kernel  $k(\cdot, \cdot, \cdot)$  is  $G \times \mathbb{R}$ , where  $f$  is a real-valued function defined on  $I$ ,  $f : I \rightarrow \mathbb{R}$ .

We first investigate under which conditions there is a unique solution for such an IE, i.e., when such a problem is well-posed.

**Theorem 2.2.1** *Let  $g \in C(I)$ ,  $k \in C(G \times \mathbb{R})$  and assume that  $k$  is Lipschitz continuous, i.e., there exists  $L \in C(G)$  such that*

$$|k(x, y, z) - k(x, y, z')| \leq L(x, y) |z - z'| \quad \forall (x, y) \in G, z, z' \in \mathbb{R}. \quad (2.9)$$

*Then (2.7) has a unique solution  $f \in C(I)$ .*

**Proof:** It suffices to prove the theorem for arbitrary intervals  $I = [a, b]$ . Thus, we may assume that  $G$  is compact.

Define

$$(T(\varphi))(x) := g(x) + \int_a^x k(x, y, \varphi(x)) dy ,$$

which is a mapping  $T : C(I) \rightarrow C(I)$ , because the functions  $g$  and  $k$  are continuous. The idea of the proof is to use the Banach fixpoint theorem on  $X := C(I)$  equipped with the norm

$$\|\varphi\| := \max_{x \in I} |e^{-\beta L_0 x} \varphi(x)|, \quad \beta \geq 1 \text{ fixed,}$$

where  $L_0 := \max_{(x,y) \in G} L(x, y)$  (which exists since  $L \in C(G)$  and  $G$  is compact). With this notation, we have by triangle inequality and Lipschitz continuity

$$\begin{aligned} |(T(\varphi) - T(\Psi))(x)| &= \left| \int_a^x [k(x, y, \varphi(y)) - k(x, y, \Psi(y))] dy \right| \\ &\leq \int_a^x L_0 |\varphi(y) - \Psi(y)| dy . \end{aligned}$$

Now, we estimate the difference of the integrands by using the particular definition of the norm as follows

$$|\varphi(y) - \Psi(y)| = e^{\beta L_0 y} |e^{-\beta L_0 y} (\varphi - \Psi)(y)| \leq e^{\beta L_0 y} \|\varphi - \Psi\|.$$

Combining this with the first estimate gives

$$\begin{aligned} |(T(\varphi) - T(\Psi))(x)| &\leq L_0 \|\varphi - \Psi\| \int_a^x e^{\beta L_0 y} dy \\ &= L_0 \|\varphi - \Psi\| \frac{1}{\beta L_0} [e^{\beta L_0 x} - e^{\beta L_0 a}] \\ &= \|\varphi - \Psi\| \frac{1}{\beta} e^{\beta L_0 x} \underbrace{[1 - e^{\beta L_0 (a-x)}]}_{\leq 1 - e^{\beta L_0 (a-b)}}, \end{aligned}$$

and thus we obtain

$$e^{-\beta L_0 x} |(T(\varphi) - T(\Psi))(x)| \leq q \|\varphi - \Psi\| \quad \forall x \in I ,$$

where the constant  $q$  is defined by

$$q := \frac{1}{\beta} (1 - e^{\beta L_0 (a-b)}) .$$

Hence, we finally obtain

$$\|T(\varphi) - T(\Psi)\| \leq q\|\varphi - \Psi\| ,$$

and since  $\beta \geq 1$ ,  $L_0 \geq 0$ ,  $b \geq a$  we have  $q < 1$  and  $T$  is a contraction.

Finally, the Banach fixpoint theorem implies the existence of a uniquely determined fixpoint  $f \in C(I)$ , i.e.,  $T(f) = f$ .  $\square$

### 2.2.1 Numerical Solution using Quadrature

The simple idea of this approach is to replace  $\int_a^x k(x, y, f(x)) dy$ ,  $\varphi(y) := k(x, y, f(x))$  ( $\varphi = k(x, \cdot, f(\cdot))$ ), by a standard quadrature formula.

Let  $Q_{[a,x]}$  denote an appropriate quadrature formula, then we obtain an approximation of the original equation by

$$\tilde{f}(x) = g(x) + Q_{[a,x]}(k(x, \cdot, \tilde{f}(\cdot))) , \quad x \in I . \quad (2.10)$$

However, we still have to define a discretization for the space variable  $x$ .

**Remark 2.2.2** (a) *The quadrature formula  $Q_{[a,x]}$  depends on the choice of quadrature points (knots), which in turns depend on the space variable  $x$ , i.e.,*

$$X_x := \{x_i = x_i(x) : i = 1, \dots, n = n(x)\}, \quad x_1 \leq x_2 \leq \dots \leq x_n.$$

(b) *Let  $\xi = x_i(x)$  be a knot of  $Q_{[a,x]}$ . For the computation, obviously  $\varphi_i(\xi)$  is needed, which in turns implies that  $\tilde{f}(\xi)$  is needed. Thus, the right-hand side of (2.10) has to be evaluated at the point  $\xi$ .*

*But this means, that all knots  $x_i(\xi)$  are needed for computation, which results in an enormous numerical effort.*

(c) *An idea to overcome this difficulty is as follows: Choose the knots  $x_0, x_1, x_2, \dots$  in such a way that all  $Q_{[a,x_i]}$  only use the first quadrature points, i.e.,  $x_0, \dots, x_i$ . Thus, one needs to evaluate (2.10) only at these knots.*

(d) *One has to start with some kind of initialization, e.g. choose  $x_0 := a \Rightarrow f(x_0) = f(a) = g(a)$ .*

With these preparation at hand, we obtain the following scheme which still is an abstract one since it contains degrees of freedom.

**Algorithm 2.2.3** (Discretization by quadrature)

1. Choose quadrature knots  $a = x_0 < x_1 < x_2 < \dots < x_{l-1} < x_l < \dots$

2. Choose a quadrature formula  $Q_i := Q_{[a, x_i]}$  with knots  $x_l$ ,  $l = 0, \dots, i$ , and the initialization

$$f_0 := g(a). \quad (2.11)$$

For  $i = 1, 2, \dots$  compute

$$f_i := g(x_i) + Q_{[a, x_i]}(k(x_i, \cdot, \tilde{f}(\cdot)))$$

using the approximation  $\tilde{f}(x_l) := f_l$ .

Typically, such a quadrature formula using only previous knots takes the form

$$Q_{[a, x_i]}(f) = \sum_{l=0}^i W_{l,i} f(x_l),$$

with weights  $W_{l,i} \in \mathbb{R}$ . Next, we insert this into the IE, i.e., (2.10) becomes

$$f_i := g(x_i) + \sum_{l=0}^i W_{l,i} k(x_i, x_l, f_l). \quad (2.12)$$

**Remark 2.2.4** (a) Obviously, (2.12) is a triangular system of equations, i.e., the unknown  $f_i$  is determined only by the values  $f_0, \dots, f_i$  of the unknown underlying function  $f$ .

(b) All equations in (2.12) are in general nonlinear since the kernel (and its approximation) is in general nonlinear. Thus, we cannot hope just to use a simple backward substitution like in the LU-decomposition of a matrix.

(c) If  $W_{i,i} = 0$ , the system is explicit. In this case,  $f_i$  can be determined directly by the already known approximations  $f_0, \dots, f_{i-1}$ . Otherwise, i.e., if  $W_{i,i} \neq 0$ , the system of equations is implicit, the current unknown  $f_i$  appears on both sides of the equation. This means, that one has to solve a (nonlinear) system of equations to go from  $f_0, \dots, f_{i-1}$  to  $f_i$ .

(d) In this form we can not use Gauß quadrature because the quadrature points for the Gauß quadrature are adapted to the interval of integration. In particular, one has that  $X_{x_i} \cap X_{x_{i-1}} = \emptyset$ , i.e., one cannot re-use any knot, so that (2.12) contains different unknowns on both sides of the equation.

(e) The naive approach of equidistant knots  $x_k = a + kh$ ,  $h > 0$ , leads to the Newton-Cotes method. For the interval  $[a, x_i]$ , the knots  $x_0, \dots, x_i$  are required, i.e., we obtain an implicit scheme.

**Example 2.2.5** Consider the so-called ‘starting point’-rule, where the left end point of a subinterval is used to define a quadrature rule by integrating the piecewise constant interpolation using the value of the function at the left end point. This leads to

$$Q_n^{sp}(f) := h \sum_{i=0}^{n-1} f(x_i) ,$$

i.e.,

$$f_i := g(x_i) + h \sum_{l=0}^{i-1} k(x_i, x_l, f_l) \quad (2.13)$$

**Remark 2.2.6** Note that the amount of work for the computation does **not** stay proportional to the number of computed values. In particular, **all** values of  $f_l$  have to be stored.

**Theorem 2.2.7** (Error estimate)

Let  $g \in C(I)$ ,  $k \in C(G \times \mathbb{R})$  be Lipschitz continuous on  $I = [a, b]$ . Assume that (2.7) has a solution  $f \in C(I)$ . Then, we have for  $\{f_i\}$  by (2.13)

$$|f_i - f(x_i)| \leq (b - a) C e^{L_k i h} , \quad x_i = a + i h , \quad (2.14)$$

where  $L_k$  is the Lipschitz constant of the kernel  $k$ .

**Proof:** The quadrature formula is of order 1, i.e.

$$|R(k, i)| = \left| \int_a^{x_i} k(x_i, y, f(y)) dy - h \sum_{l=0}^{i-1} k(x_i, x_l, f(x_l)) \right| \leq C(x_i - a)h .$$

Thus, by triangle inequality, we obtain

$$\begin{aligned} |d_i| &= |f_i - f(x_i)| \\ &= \left| g(x_i) + h \sum_{l=0}^{i-1} k(x_i, x_l, f_l) - g(x_i) - \int_a^{x_i} k(x_i, y, f(y)) dy \right| \\ &\leq h \sum_{l=0}^{i-1} |k(x_i, x_l, f_l) - k(x_i, x_l, f(x_l))| + |R(k, i)| \\ &\leq L_k h \sum_{l=0}^{i-1} \underbrace{|f_l - f(x_l)|}_{=|d_l|} + C \underbrace{(x_i - a)}_{\leq (b-a)} h . \end{aligned} \quad (2.15)$$

The next step is the use of the following standard result.

**Lemma 2.2.8** (*Gronwall Lemma*)

If  $\alpha, \beta_l \geq 0$  ( $0 \leq l < i$ ),  $0 \leq M_0 < 1$  such that

$$0 \leq E_i \leq \alpha + \sum_{l=0}^{i-1} \beta_l E_l + M_0 E_i, \quad (2.16)$$

then

$$E_i \leq \frac{\alpha}{1 - M_0} \exp \left\{ \sum_{l=0}^{i-1} \frac{\beta_l}{1 - M_0} \right\}. \quad (2.17)$$

**Proof:** By induction, see e.g. [3].  $\square$

To continue with the proof of Theorem 2.2.7, we apply the Gronwall Lemma for  $E_l = d_l$ ,  $\alpha = C(b - a)h$ ,  $M_0 = 0$ , and  $\beta_l = L_k h$ . Then, we obtain

$$|d_i| \leq C(b - a)h \exp \left\{ \sum_{l=0}^{i-1} L_k h \right\}$$

which gives (2.14).  $\square$

**Remark 2.2.9** (a) *The first order convergence of the quadrature rule is reflected by the first order estimate in (2.14).*

(b) *As an alternative, one can use the second order summed trapezoidal rule*

$$Q_n^{ST}(f) := h \left[ \frac{1}{2}(f(a) + f(b)) + \sum_{i=1}^{n-1} f(x_i) \right],$$

so that we obtain the following scheme

$$\begin{cases} f_0 & := g(0) \\ f_1 & := g(x_1) + \frac{h}{2}(k(x_1, a, f(a)) + k(x_1, b, f(b))) \\ & + h \sum_{j=1}^{n-1} k(x_1, x_j, f_j). \end{cases} \quad (2.18)$$

Obviously, this is an implicit scheme.

(c) *It can be shown by the Banach fixpoint theorem that (2.18) admits a unique solution which also yields a precise algorithm (using the fixpoint iteration)*

$$\left\{ \begin{array}{l} f_i^{(0)} := f_{i-1} \text{ or} \\ f_i^{(0)} = c_i + \frac{h}{2}k(x_i, x_{i-1}, f_{i-1}) \\ f_i^{(\nu+1)} := c_i + \frac{h}{2} \sum_{j=1}^{n-1} k(x_i, x_j, f_j^{(\nu)}) . \end{array} \right\} \quad (\text{initial values}) \quad (2.19)$$

(d) *Higher order methods may also be constructed. However, this is a non-trivial task.*

## 2.2.2 Collocation Methods

This is a class of numerical schemes that can be applied for several problems also including PDEs. The idea is to evaluate an infinite-dimensional equation (like a PDE or an IE) on a finite number of points (the so-called *collocation points*).

**Remark 2.2.10** *Using a quadrature rule like above enforced us to use equidistant knots*

$$x_j = a + jh.$$

A way-out is to use an interpolation of  $f$  with respect to the equidistant knots  $x_j$  for  $0 \leq j \leq i$  by means of a function  $\Phi$  in the sense

$$\Phi(x_j) = f_j = f(x_j), \quad 0 \leq j \leq i. \quad (2.20)$$

Thus, the integral is approximated by

$$\int_a^{x_i} k(x, y, f(y)) dy \approx \int_a^{x_i} k(x, y, \Phi(y)) dy \quad (2.21)$$

and  $\Phi(y)$  is known for all  $y \in [a, x_i]$ .

This means that any quadrature rule may be used for the approximation  $\int_a^{x_i} k(x, y, \Phi(y)) dy$ , in particular not necessarily the same rule of  $Q_{[a, x_i]}$ .

**Example 2.2.11** *Choose  $\Phi$  as the piecewise linear interpolant, i.e.,*

$$\Phi(x) = \frac{1}{h} [(x - x_j)f_{j+1} + (x_{j+1} - x)f_j], \quad x \in [x_j, x_{j+1}]. \quad (2.22)$$

*It is well-known that the error is of the order  $\mathcal{O}(h^2)$ .*

*Hence, there is no reason to use a quadrature of higher order for (2.21). One can use piecewise cubic Hermite polynomials, i.e.*

$$\Phi \in C^1([a, x_i]) \quad \text{and} \quad \Phi_{[x_j, x_{j+1}]} \in \mathcal{P}_3.$$

The most efficient way is to compute  $\Phi$  in a successive way on an interval of the form

$$[c, c + h] = [x_0, x_1], [x_1, x_2], [x_2, x_3], \dots$$

Assuming that  $\Phi(x_{i-1})$  and  $\Phi'(x_{i-1})$  is known, we obtain

$$\left\{ \begin{array}{l} \Phi(c) = g(c), \quad \Phi'(c) = f'(c) = g'(c) + k(c, c, g(c)) \\ \text{For } (k > 1) \\ \Phi(x_{i-1} + 0) = \Phi(x_{i-1} - 0) \\ \Phi'(x_{i-1} + 0) = \Phi'(x_{i-1} - 0) \end{array} \right. \quad (2.23)$$

Obviously, (2.23) fixes three of the four parameters of a cubic polynomial. The idea of this particular collocation method is to try to satisfy the original (2.7) in two points

$$x_{i,1} = x_{i-1} + \theta_1 h, \quad x_{i,2} = x_{i-1} + \theta_2 h, \quad \theta_1, \theta_2 \in (0, 1],$$

i.e.,

$$\Phi(x_{i,j}) = g(x_{i,j}) + \int_a^{x_{i,j}} k(x_{i,j}, y, \Phi(y)) dy, \quad j = 1, 2,$$

in theory and

$$\Phi(x_{i,j}) = g(x_{i,j}) + Q_{[a, x_{i,j}]}(k(x_{i,j}), \cdot, \Phi(\cdot)), \quad j = 1, 2,$$

in practice.

As an example, one use summed Gauß-Quadrature as an exercise.

**Remark 2.2.12** *It turns out that a stability condition is required in order to obtain the desired order of approximation. The above system can be shown to be stable for*

$$\theta_2 = 1 \quad \text{and} \quad \frac{1}{2} < \theta_1 < 1.$$

Moreover, it diverges for  $0 \leq \theta_1 < \frac{1}{2}$ . A practical choice (which is found heuristically) is  $\theta_1 = 0.6$ .

## 2.3 Fredholm Integral Equations of 2nd Kind

We now consider the model problem

$$f(x) = g(x) + \int_I k(x, y) f(y) dy, \quad x \in I := [a, b]. \quad (2.24)$$

If instead of  $I$  we also take 2D curves into account we consider a domain  $D \subset \mathbb{R}^d$ , ( $1 \leq d < \infty$ ) or a manifold  $D = \partial\Omega$ ,  $\Omega \subset \mathbb{R}^d$  and the more general formulation

$$f(x) = g(x) + \int_D k(x, y) f(y) dy, \quad x \in D. \quad (2.25)$$

With the kernel  $k(\cdot, \cdot)$ , one defines the *integral operator* (i.e., the operator takes a function as variable) in the following way

$$K : f \longmapsto \int_D k(\cdot, y) f(y) dy, \quad (2.26)$$

so that (2.25) (or (2.24)) reads

$$f = g + Kf . \quad (2.27)$$

It turns out to be useful to introduce some extra real parameter  $\lambda \neq 0$  and generalize (2.27) as

$$\lambda f = g + Kf. \quad (2.28)$$

Obviously the particular choice  $\lambda = 1$  gives (2.27), otherwise divide by  $\lambda \neq 0$  in order to obtain the desired  $f$ .

### 2.3.1 Some Mathematical Theory

Thus we consider

$$(\lambda I - K)f = g,$$

which is an operator equation on a suitable Banach space  $X$  (e.g.  $X = C(D)$  or  $X = L_2(D)$ ), i.e.  $K \in \mathcal{L}(X, X)$ , where  $\mathcal{L}(X, Y)$  denotes the space of bounded linear mappings from a normed space  $X$  to another normed space  $Y$ .

**Theorem 2.3.1 (Fredholm alternative)** *Consider  $(\lambda I - K)f = g$ , where  $\lambda \neq 0$  and  $K \in \mathcal{L}(X, X)$  is compact (i.e., the image  $(Kf_n)_n$  of a bounded sequence  $(f_n)_n \subset X$  contains a convergent subsequence).*

*Then, **either**  $\lambda I - K$  has a bounded inverse  $(\lambda I - K)^{-1} \in \mathcal{L}(X, X)$  so that  $(\lambda I - K)f$  has a unique solution*

$$f = (\lambda I - K)^{-1}g \text{ for any } g \in X$$

*or  $\lambda$  is one (of the countable many) eigenvalues of  $K$  that can only cluster in 0.*

**Proof:** Using Riesz-Schauder theory, see, e.g. [3].  $\square$

Now it remains to collect criteria to see if the assumptions of the latter theorem are satisfied for a given integral operator. In particular, these criteria need to be easy to check. For the proofs, we again refer to [3]. We first consider the boundedness of  $K$ .

**Lemma 2.3.2** *For any  $f \in C(D)$  we assume  $Kf \in C(D)$ . If*

$$\|K\|_{[C(D)]} := \sup_{x \in D} \int_D (k(x, y)) dy < \infty , \quad (2.29)$$

*then  $K \in \mathcal{L}(X, X)$ .  $\square$*

Finally, we consider the compactness which was also assume in the Fredholm alternative. The following result gives a characterization.

**Lemma 2.3.3** Let  $D \subset \mathbb{R}^d$  be a compact set and for the kernel  $k$  we assume

$$\int_D (k(x, y)) dy < \infty \text{ for all } x \in D, \quad (2.30)$$

as well as

$$\lim_{\xi \rightarrow x} \int_D (k(\xi, y) - k(x, y)) dy \text{ for all } x \in D. \quad (2.31)$$

Then  $K$  is compact.

On the other hand, if  $K \in \mathcal{L}(X, X)$  is compact with kernel  $k(x, \cdot) \in L_1(D)$ , then (2.30), (2.31) hold.

Finally, we mention a class of operators which often occur in several applications.

**Definition 2.3.4** A kernel  $k \in L_2(D \times D)$  is called Hilbert-Schmidt kernel.

Now, it is an easy exercise to see that such kernels satisfy the assumptions of the Fredholm alternative.

**Lemma 2.3.5** An operator  $K$  with a Hilbert-Schmidt kernel is compact for  $X = L_2(D)$ .  $\square$

## 2.3.2 Numerical Methods

It turns out that one can describe the design of appropriate numerical schemes in a fairly general setting. This general setting is not just a academic game, it turns out to be very useful. To this end, let  $X$  be a Banach space,  $K \in \mathcal{L}(X, X)$  and we use the standard operator norm

$$\|K\| = \|K\|_{[X]} := \sup_{f \in X} \|Kf\|_X .$$

We first consider a *semi-discretization*, where  $K$  is approximated by a sequence of operators  $K_n \in \mathcal{L}(X, X)$  and  $g \in X$  is approximated by  $g_n \in X$ . The space  $X$  is not jet discretized which is the reason for the name *semi-discretization*. Then, we have the following iteration in  $X$

$$\lambda f_n = g_n + K_n f_n , \quad n \in \mathbb{N} . \quad (2.32)$$

Let us first study the properties of this abstract iteration, of course keeping in mind that we cannot realize this in a computer since  $X$  is general infinite-dimensional.

**Definition 2.3.6** A discretization  $(K_n)_{n \in \mathbb{N}}$  is called consistent (in  $X$ ) if

$$\lim_{n \rightarrow \infty} \|K\varphi - K_n \varphi\|_X = 0 \text{ for all } \varphi \in X . \quad (2.33)$$

This property is also called pointwise consistency (describing pointwise convergence of  $K_n$  to  $K$ ).

**Definition 2.3.7** A discretization  $(K_n)_{n \in \mathbb{N}}$  is called **stable** (in  $X$ ), if there exist some  $n \in \mathbb{N}_0$ ,  $C > \infty$  such that  $\lambda I - K_n$  is invertible for all  $n \geq n_0$  with inverse  $(\lambda I - K_n)^{-1} \in \mathcal{L}(X, X)$  such that

$$\|(\lambda I - K_n)^{-1}\| \leq C \text{ for all } n \geq n_0 . \quad (2.34)$$

Consistency and stability are standard terms in Numerical Analysis. A stronger property as in Definition 2.3.6 is the *convergence in the operator norm* defined by

$$\|K - K_n\| \longrightarrow 0 \text{ for } n \rightarrow \infty . \quad (2.35)$$

Under this stronger assumption, one can proof (again, see [3]).

**Theorem 2.3.8** If  $(\lambda I - K_n)^{-1} \in \mathcal{L}(X, X)$  and (2.35), then the discretization is stable.  $\square$

**Remark 2.3.9** Note that also a certain inverse implication holds for the latter theorem, [3].

Finally, we come to the desired property of convergence.

**Definition 2.3.10** The discretization  $\{K_n\}$  is called *convergent* (in  $X$ ), if there exist some  $n_0 \in \mathbb{N}_0$  such that (2.32) for  $g_n = g$  is uniquely solvable for all  $n \geq n_0$  and  $\lim_{n \rightarrow \infty} f_n$  exists.

Then, we obtain the following results.

**Theorem 2.3.11 (Stability theorem)**

(a) *Convergence implies stability.*

(b) *Convergence plus consistency imply existence of the inverse  $(\lambda I - K)^{-1} \in \mathcal{L}(X, X)$  .*

**Proof:**

(a) Consider  $T_n := (\lambda I - K_n)^{-1}$ ,  $n \geq n_0$  (existence of the inverse of the approximation  $T_n$  by assumption) and  $\lim_{n \rightarrow \infty} T_n g$  exists for all  $g \in X$ . Then by the Uniform Boundedness Principle (UBP, see any lecture or textbook on Functional Analysis), we obtain that  $\|T_n\| \leq C$  (see, e.g., [4, § 40] or [9, §II.1]).

(b) We consider

$$K_n f_n - K f = K_n (f_n - f) + (K_n - K) f \quad (2.36)$$

and for the first term we have

$$\|K_n (f_n - f)\| \leq C \|f_n - f\| \rightarrow 0$$

for  $k \rightarrow \infty$  due to compactness and convergence. The second term tends to zero,  $(K_n - K) f \rightarrow 0$  for  $k \rightarrow \infty$  because of the consistency.

Thus, using (2.36), we take the limit of  $\lambda f_n = g + K_n f_n$  which leads to the equation  $\lambda f = g + K f$  for **any**  $g$  which shows that  $\lambda I - K$  is a surjective mapping (onto). By (a), we also have stability, i.e.,  $\lambda I - K$  is injective and hence  $\lambda I - K$  is one-to-one, which means that the inverse exists.  $\square$

**Theorem 2.3.12 (Convergence theorem)**

(a) Let us assume stability and consistency. Moreover, either

(i)  $\lambda I - K$  is surjective, or

(ii)  $\lambda \neq 0$  and  $K$  compact.

Then we have convergence of (2.32) for  $g_n = g$ .

(b) If in addition  $\lim_{n \rightarrow \infty} \|g - g_n\| = 0$ , then (2.32) converges.

**Proof:**

(a) By stability we have that  $\lambda I - K$  is injective. In case of (i), we thus obtain bijectivity. In case (ii) we know that a compact, injective operator is also surjective if  $\lambda \neq 0$ , which proves the assertion.

(b) Is ‘clear’ by using triangle inequality.  $\square$

The final idea is now to replace the infinite-dimensional equation  $\lambda f = g + Kf$  by a finite-dimensional approximation  $\lambda f_n = g_n + K_n f_n$  and show

- stability,
- consistency, and
- convergence.

We obtain:

**Theorem 2.3.13** *If  $\|K_n - K\| \rightarrow 0$  for  $n \rightarrow \infty$  and  $(\lambda I - K)^{-1} \in \mathcal{L}(X, X)$ , then we obtain consistency, stability and convergence of (2.32) for  $g_n = g$ .  $\square$*

For a numerical method, one is not only interested in its convergence, but also in its *robustness*, i.e., one aims at having methods that guarantee that a small variation in the input data only results in small changes in the computed approximate solution. For this sensitivity of the numerical solution with respect to small perturbations of  $g_n$  or  $K_n$ , we consider the *condition number* defined by

$$\text{cond}_X(\lambda I - K_n) := \|\lambda I - K_n\| \|(\lambda I - K_n)^{-1}\|. \quad (2.37)$$

Here, again, we use the operator norm.

**Theorem 2.3.14** *Let  $\{K_n\}$  be consistent and stable. Then*

$$\text{cond}_X(\lambda I - K_n) \leq C$$

for all  $n \geq n_0$

**Proof:** Functional Analysis (UBP + Banach-Steinhaus), [4, 9].  $\square$

Note that Theorem 2.3.14 gives an *optimal* result since the condition number of the discrete operator is independent of  $n$ . The condition number is a measure how hard a certain numerical problem is. In some numerical schemes (e.g. the cg-method for symmetric positive definite linear system of equations) it directly corresponds to the numerical amount of work. Now, think of  $n$  being the number of degrees of freedom in the sense that a larger value of  $n$  corresponds to a finer discretization and thus to a larger system to be solved numerically. Then, the statement of Theorem 2.3.14 says that e.g. an iterative scheme needs the same number of iterations no matter if  $n$  is small or big. This is the best one can expect.

### 2.3.3 Discretization by Kernel-Approximation

We still have to introduce a discretization for the space  $X$ . As a first step consider so-called *degenerated* kernels

$$k_n(x, y) = \sum_{j=1}^n a_j(x)b_j(y) \text{ with } a_j = a_{j,n}, b_j = b_{j,n}, \quad (2.38)$$

i.e., kernels where the variables  $x$  and  $y$  are separated. Thus, we obtain  $\text{Range}(K_n) = \text{span}\{a_1, \dots, a_n\}$ , i.e., if we choose  $a_j$  linearly independent, we obtain a basis for the range. This means for the approximation of the operator  $K$  that

$$K_n f_n = \sum_{j=1}^n \gamma_j a_j, \quad \gamma_j := \int_D b_j(y) f_n(y) dy.$$

Now, we insert this formula in (2.32) using  $g_n = g$  for the moment:

$$\lambda f_n = g + \sum_{j=1}^n \gamma_j a_j, \quad (\lambda \neq 0 \text{ i.e. second kind})$$

Thus, we can solve for  $f_n$  and obtain

$$f_n = \frac{1}{\lambda} g + \sum_{j=1}^n \alpha_j a_j, \quad \alpha_j := \frac{\gamma_j}{\lambda}, \quad (2.39)$$

i.e., the coefficients  $\alpha_j$  are actually the unknowns that need to be determined.

Plug (2.39) into (2.32) yields

$$\begin{aligned} g + \lambda \sum_{j=1}^n \alpha_j a_j &= \lambda f_n = g + K_n f_n \\ &= g + K_n \left( \frac{1}{\lambda} g + \sum_{k=1}^n \alpha_k a_k \right) \\ &= g + \frac{1}{\lambda} K_n g + \sum_{k=1}^n \alpha_k K_n a_k. \end{aligned}$$

For the approximation of the kernel, we have:

$$\begin{aligned}
\frac{1}{\lambda}(K_n g)(x) &= \frac{1}{\lambda} \int_D \sum_{j=1}^n a_j(x) b_j(y) g(y) dy \\
&= \sum_{j=1}^n \beta_j a_j(x), \text{ where } \beta_j := \frac{1}{\lambda} \int_D b_j(y) g(y) dy \\
\rightsquigarrow (K_n a_k)(x) &= \sum_{j=1}^n \beta_{j,k} a_j(x), \beta_{j,k} := \int_D b_j(y) a_k(y) dy.
\end{aligned}$$

Putting everything together yields:

$$\begin{aligned}
\lambda \sum_{j=1}^n \alpha_j a_j &= \sum_{j=1}^n \beta_j a_j + \sum_{k=1}^n a_k \sum_{j=1}^n \beta_{j,k} a_j \\
&= \sum_{j=1}^n \left\{ \beta_j + \sum_{k=1}^n \alpha_k \beta_{j,k} \right\} a_j.
\end{aligned}$$

Since the  $a_j$  are linearly independent, we can compare the coefficients and obtain

$$\lambda \alpha_j = \beta_j + \sum_{k=1}^n \alpha_k \beta_{j,k}, \tag{2.40}$$

i.e., in matrix form  $(\lambda I - B_n) a_n = b_n$  where

$$\begin{aligned}
a_n &:= (\alpha_1, \dots, \alpha_n)^T, \quad b_n = (\beta_1, \dots, \beta_n)^T \\
B_n &= (\beta_{j,k})_{j,k=1,\dots,n},
\end{aligned}$$

where  $a_n$  is the vector of unknowns and  $\beta_j, \beta_{j,k}$  are inner products of the above form

$$\beta_j = \frac{1}{\lambda} \int_D b_j(y) g(y) dy, \quad \beta_{j,k} = \int_D b_j(y) a_k(y) dy.$$

**Remark 2.3.15** (i) Obviously, we obtain a linear system for degenerated kernels. Still, the chosen basis  $\{a_1, \dots, a_n\}$  is a free parameter which of course influences the quality and behavior of the method.

The integrals in  $\beta_j$  and  $\beta_{j,k}$  are usually approximated by a suitable quadrature formula.

(ii) Other kernel approximations (in particular for non-degenerated kernels) lead to other systems that may also be non-linear.

(iii) In general, the arising matrix  $(\lambda I - B_n)$  is not symmetric and full. This means that one has to resort to special numerical schemes for their solution like multi-grid or compression techniques (like multi-pole or wavelets).

(iv) A simple example on  $D = [0, 1]$  are linear interpolants which will be considered as an exercise.

## 2.4 Projection Methods

Projection methods are again a quite general framework that is also similar to Galerkin schemes for elliptic partial differential equations. The idea is to replace the infinite-dimensional space  $X$  (think of  $L_2(D)$  or  $C(D)$ ) by a sequence of finite subspaces  $X_n$ ,  $n \in \mathbb{N}$ . In order to obtain a convergent scheme, we aim that finding subspaces  $X_n$  such that

$$\lim_{n \rightarrow \infty} \text{dist}(x, X_n) = 0, \text{ where}$$

$$\text{dist}(x, Y) := \inf_{\xi \in Y} \|x - \xi\|$$

denotes the distance from a point  $x$  to a set  $Y$ .

As usual, a linear mapping  $\Pi_n : X \rightarrow X$  is called *projector* if

$$\Pi_n^2 = \Pi_n$$

and each projector defines a subspace by  $X_n := \text{Range}(\Pi_n)$ .

One can define consistency, stability and convergence in a similar way as above.

Typically  $(\lambda I - K)f = g$  will not have a solution in  $X_n$ , but there will be a *residual* defined by

$$d_n := (\lambda I - K)f_n - g \quad (\text{note ,}$$

and note that in general  $d_n \notin X_n$ . If  $d_n = 0$  is not possible (which is the standard case), we aim that at least the projection vanishes:

$$\Pi_n d_n = 0, ,$$

i.e.

$$\begin{aligned} 0 &= \Pi_n d_n = \lambda \Pi_n f_n - \Pi_n(K f_n) - \Pi_n g \\ &= \lambda f_n - \Pi_n(K f_n) - \Pi_n g, \end{aligned}$$

since we aim that  $f_n \in X_n$  holds. This yields a semi-discrete projection method

$$\lambda f_n = \Pi_n g + \Pi_n(K f_n) =: g_n + K_n f_n, \tag{2.41}$$

where again a discretization for  $X$  needs to be introduced.

**Lemma 2.4.1** *Let  $\lambda \neq 0$ , then any solution  $f_n \in X$  of (2.41) belongs to  $X_n$ , i.e.  $f_n \in X_n$ .*

**Proof:** Clear, since  $\Pi_n g + \Pi_n(K f_n) = \Pi_n(g + K f_n) \in X_n$ , thus  $\lambda f_n \in X_n$ .  $\square$

Note that the statement of Lemma 2.4.1 becomes wrong if  $g_n \notin X_n$  is used instead of  $\Pi_n g$ . We obtain similar results as above.

**Theorem 2.4.2** *Assume that  $\{\Pi_n\}_{n \in \mathbb{N}_0}$  is convergent, i.e.,  $\Pi_n x \rightarrow x$  for  $n \rightarrow \infty$  for all  $x \in X$  and that  $K$  is compact.*

(a) Then

$$\|\Pi_n K - K\| \longrightarrow 0 \text{ for } n \rightarrow \infty$$

(b) Let  $\lambda$  be a regular value of  $K$ , i.e.,  $(\lambda I - K)^{-1} \in L(X, X)$ , then (2.41) is stable, consistent and convergent.

**Proof:**

(a) Again by the uniform boundedness principle (UBP).

(b) By Theorem 2.3.13 since  $\|K_n - K\| \rightarrow 0$ .  $\square$

A particular example of a projection method is the *collocation method* for  $X = C(D)$ , where  $\Pi_n$  is defined by an interpolation operator. Fix a set

$$\Xi := \{\xi_{1,n}, \xi_{2,n}, \dots, \xi_{n,n}\} \subset D \quad (2.42)$$

of mutually disjoint knots  $\xi_{i,n}$ . Then, let  $\Phi_{i,n}$  be a basis of interpolating functions with respect to the grid  $\Xi$ , i.e.,

$$(\Phi_{i,n}(\xi_{j,n}))_{i,j=1,\dots,n} \text{ is regular}$$

(e.g., the Lagrange basis functions, splines etc.). Then we obtain from (2.41) the *collocation equation*:

$$\lambda f_n(\xi_{j,n}) = g(\xi_{j,n}) + (K f_n)(\xi_{j,n}), \quad 1 \leq j \leq n, \quad (2.43)$$

i.e., the original IE is satisfied only at the collocation points on the grid  $\Xi$ . Using the basis functions  $\{\Phi_{i,n}\}_{1 \leq i \leq n}$ , we obtain

$$f_n = \sum_{k=1}^n \alpha_k \Phi_{k,n}$$

and hence  $\lambda \sum_{k=1}^n \alpha_k \Phi_{k,n}(\xi_{j,n}) = g(\xi_{j,n}) + \sum_{k=1}^n \alpha_k (K \Phi_{k,n})(\xi_{j,n})$ , i.e.,

$$\boxed{(\lambda A - B)a = b}, \quad (2.44)$$

where

$$\begin{aligned} a &:= a_n := (\alpha_1, \dots, \alpha_n)^T && \text{(vector of unknown)} \\ b &:= b_n := (g(\xi_{1,n}), \dots, g(\xi_{n,n}))^T && \text{(right-hand side)} \end{aligned}$$

and the system matrices

$$\begin{aligned} A := A_n &:= \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \vdots & \vdots & & \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nn} \end{bmatrix}, & \alpha_{j,k} := \Phi_{k,n}(\xi_{j,n}), \\ B := B_n &:= \begin{bmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1n} \\ \vdots & \vdots & & \\ \beta_{n1} & \beta_{n2} & \dots & \beta_{nn} \end{bmatrix}, & \beta_{j,k} := (K \Phi_{k,n})(\xi_{j,n}). \end{aligned}$$

**Remark 2.4.3** (a) The equation (2.44) is a linear  $(n \times n)$ -system. It can be solved by standard numerical schemes.

(b) If  $\Phi_{k,n}$  is a Lagrange basis, i.e.  $\Phi_{k,n}(\xi_{j,n}) = \delta_{j,n}$ , then  $A = I$ , i.e., the identity.

(c) To compute the matrix  $B$ , we need to compute  $n^2$  integrals of the form

$$\beta_{j,k} = \int_D k(\xi_j, y) \Phi_{k,n}(y) dy ,$$

which is often done by numerical quadrature.

(d) Typically the matrix  $B$  is densely populated. This has to be taken into account in the choice of a numerical scheme for the linear system of equations.

**Example 2.4.4** Consider piecewise linear interpolation using Lagrange functions, i.e.,  $A = I$ , and the matrix  $B$  consists of the terms

$$\beta_{j,k} = \frac{1}{\xi_k - \xi_{k-1}} \int_{\xi_{k-1}}^{\xi_k} k(\xi_j, y)(y - \xi_{k-1}) dy + \frac{1}{\xi_{k+1} - \xi_k} \int_{\xi_k}^{\xi_{k+1}} k(\xi_j, y)(\xi_{k+1} - y) dy.$$

For equidistant knots  $\xi_k = a + kh$ ,  $k = 0, \dots, n$ ,  $h = \frac{b-a}{n}$  we have the following straightforward error estimate

$$\|f - f_n\|_\infty \leq C \cdot h^2 .$$

**Example 2.4.5** Using piecewise constant interpolation with respect to the midpoints, i.e.,  $\xi_k := \frac{x_k + x_{k-1}}{2}$  and using

$$\Phi_{k,n}(x) = \begin{cases} 1, & x \in I_k = [x_{k-1}, x_k] \\ 0, & \text{else} \end{cases}$$

yields the following entries for  $B$

$$\beta_{j,k} = \int_{I_k} k(\xi_j, y) dy.$$

In general, we obtain first order convergence  $\|f - f_n\|_\infty \leq C \cdot h$ , however, if  $K \in L(C(D), C(D)) \cap L(H^1(D), L_\infty(D))$  and if  $f$  is smooth, i.e.,  $f \in H^2(D)$  (Sobolev space), then we observe so-called super-convergence, i.e.,

$$\|f - f_n\|_{\infty, \Xi} \leq Ch^2.$$

**Example 2.4.6 (for exercises)** Choose  $D = [0, 1]$  with the smooth kernel

$$k(x, y) = \cos(\pi xy)$$

and the solution  $f(x) \equiv 1$ . For  $\lambda = 1$ , compute the right-hand side

$$g(x) = 1 - \frac{\sin(\pi x)}{\pi x}.$$

Now, compare the two methods, namely

- Approximation of the kernel, and
- collocation.

**Remark 2.4.7** (a) As already mentioned, another example of projection methods are Galerkin methods, where  $X = L_2(D)$  and  $\Pi_n$  is the orthogonal projection onto finite-dimensional subspaces  $X_n \subset X$ .

The resulting method is similar to Galerkin methods for elliptic PDEs (see Numerical Finance I) but the stiffness matrices for IEs are full.

- (b) Numerical methods for the solution of the arising linear system of equations are needed (recall, they are full).
- Gauß :  $\mathcal{O}(n^3)$  is not useful, since it is too costly.
  - Fixpoint iterations (e.g. Picard, Richardson, ...) are possible, but in general slow.
  - Conjugate gradient (if  $\lambda I - Kn$  is s.p.d.)
  - Multi-grid methods, see, e.g. [3].

We will give a particular example for a Fredholm IE arising from finance in the next chapter.

# Chapter 3

## American Option Pricing

In contrast to European Options (see Numerical Finance I), an American Option can be exercised at **any** time  $t \leq T$ . This means that the payoff function is ( $S \hat{=}$  price of the underlying asset,  $K \hat{=}$  exercise price)

$$\begin{aligned} V_C^{\text{am}}(S, t) &= (S_t - K)^+ \quad \text{for a call and} \\ V_P^{\text{am}}(S, t) &= (K - S_t)^+ \quad \text{for a put.} \end{aligned} \tag{3.1}$$

This means, at expiration time  $T$ , the payoff coincides with that of European Options. Under standard no-arbitrage assumptions, one can easily show the following a-priori bounds

$$\begin{aligned} V_P^{\text{am}}(S, t) &\geq (K - S)^+ \\ V_C^{\text{am}}(S, t) &\geq (S - K)^+ \quad \forall S, t \end{aligned}$$

and trivially

$$V^{\text{am}} \geq V^{\text{eur}} ,$$

since a European Option is a special case of an American Option.

Again, we do not go into details of the modelling of American options but refer to any lecture on Financial Mathematics and the literature, e.g. [2, 7] which are also the basis for this chapter. Here we concentrate only on those facts that are relevant for the numerical simulation of these financial processes.

The *contact point*  $S_f$  is defined as follows

$$V_P^{\text{am}}(S_f, t) = K - S_f \quad (0 < S_f < K) . \tag{3.2}$$

Note that  $S_f$  depends on the time  $t$ , i.e.,  $S_f = S_f(t)$ . The contact point  $S_f(t)$  can be characterized by

$$\begin{cases} V_P^{\text{am}}(S, t) > (K - S)^+ & \text{for } S > S_f(t), \\ V_P^{\text{am}}(S, t) = K - S & \text{for } S \leq S_f(t). \end{cases} \tag{3.3}$$

The location of the manifold  $S_f(t)$ ,  $t \in (0, T)$  is *unknown* a priori. Since this graph is the interface between the ‘exercise’ and the ‘no-exercise’ region (see below), we are faced with

a so called *free boundary-value problem*. These kind of problems also occur in several other applications, e.g., the propagation of waves in a medium (water waves, acoustic waves), plastic deformation processes and so on. Most of what is said in this chapter can also be applied to this kind of ‘industrial’ problems.

The interpretation of the contact point for a put is as follows. The holder should exercise as soon as the price of the asset reaches  $S_f(t)$ . The corresponding time instant  $t_S$  is called *stopping time*.

**Boundary conditions:** The slope  $\frac{\partial V}{\partial S}$  with which  $V_P^{\text{am}}(S, t)$  touches the straight line  $K - S$  at  $S_f(t)$ .  $K - S$  has the slope  $-1 = \frac{\partial}{\partial S}(K - S)$

$$\frac{\partial}{\partial S} V_P^{\text{am}}(S_f(t), t) = -1, \quad (3.4)$$

which results in a tangential touching. This is the so-called *high contact condition*.

For the (somewhat hypothetical) case of a *perpetual option* (i.e., for maturity  $T = \infty$ ), we obtain an asymptotic condition (that can be calculated analytically). We will come back to this point later.

In general, we obtain two boundary conditions, namely (3.2), (3.4).

For the American call, we need to include also dividend yields  $\delta$  (otherwise they coincide with a European call option), i.e., the Black-Scholes equation takes the form

$$\frac{\partial}{\partial t} V(S, t) + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2}{\partial S^2} V(S, t) + (r - \delta) S \frac{\partial}{\partial S} V(S, t) - rV(S, t) = 0. \quad (3.5)$$

In fact, one can show that if  $\delta = 0$  an early exercise does not pay for a call. Since  $V_C^{\text{am}} \geq S - Ke^{-r(T-t)}$ , we have

$$V_C^{\text{am}} > S - K \quad \text{for } t < T \quad \text{and } r > 0,$$

i.e., for  $\delta = 0$  American and European calls are identical:  $V_C^{\text{am}} = V_C^{\text{eur}}$ .

In this case, the corresponding boundary conditions read as follows.

$$V_C^{\text{am}}(S_f(t), t) = S_f(t) - K \quad (3.6)$$

$$\frac{\partial}{\partial S} V_C^{\text{am}}(S_f(t), t) = 1 \quad (3.7)$$

## Black-Scholes Inequality

In the derivation of (3.5) early exercise was excluded (see Numerical Finance I). Following the lines of argumentation ‘= 0’ is now replaced by ‘ $\leq 0$ ’ and we obtain the *Black-Scholes Inequality*, which holds for all  $(S, t)$ .

The inequality can be reformulated taking into account that the contact boundary  $S_f$  divides the half strip into two disjoint regulars.

$$\begin{aligned} \textbf{Put: } V_P^{\text{am}} &= K - S && \text{for } S \leq S_f \quad (\text{stop}), \\ V_p^{\text{am}} &\text{ solves (3.5) } && \text{for } S > S_f \quad (\text{hold}). \end{aligned} \tag{3.8}$$

$$\begin{aligned} \textbf{Call: } V_C^{\text{am}} &= S - K && \text{for } S \geq S_f \quad (\text{stop}), \\ V_p^{\text{am}} &\text{ solves (3.5) } && \text{for } S < S_f \quad (\text{hold}). \end{aligned} \tag{3.9}$$

This shows that also here the Black-Scholes equations has to be solved but with the additional problem of the *free boundary*.

### 3.1 An Integral Formulation

This section is based upon the article [5]. It shows that an American Call can be modelled an integral equation so that the methods described in the previous chapter can be applied. The details of the derivation are somewhat technical and can also be found in [5]. Moreover, we only consider the *call* ( $\delta \neq 0$ ) here and refer to [5] for the Put.

Let us collect the information on the free boundary. Recall the boundary conditions (3.6, 3.7):

$$V_C^{\text{am}}(S_f(t), t) = S_f(t) - K, \tag{3.10}$$

$$\frac{\partial}{\partial S} V_C^{\text{am}}(S_f(t), t) = 1 \quad (\text{'high contact'}). \tag{3.11}$$

For the location of the free boundary at expiry time  $T$  we have

$$S_f(T) = \begin{cases} S_0 := K \cdot \frac{r}{\delta} > K, & \text{for } r > \delta > 0, \\ S_0 := K, & \text{for } r \geq \delta. \end{cases} \tag{3.12}$$

As already mentioned before, the asymptotic behavior for moving expiration dates can be obtained from the study of *perpetual Calls* ( $T = \infty$ ):

$$\begin{cases} S_f(t) \rightarrow S^* := \frac{K}{1-1/\alpha^+}, \text{ as } T - t \rightarrow \infty, \text{ where} \\ \alpha^+ := \frac{1}{2\sigma^2} \left[ \sigma^2 - 2(r - \delta) + \sqrt{4(\delta - r)^2 + \sigma^2(4\delta + 4r + \sigma^2)} \right]. \end{cases} \tag{3.13}$$

It will be convenient to revert the relation  $S = S_f(t)$  to

$$t = T_f(S),$$

i.e.,  $T_f(S)$  can be interpreted as ‘an’ optimal exercise time depending on the price of the underlying. Thus (3.12) reads

$$T_f(S_0) = T \tag{3.14}$$

and (3.13) becomes

$$T_f(S) \rightarrow \infty \text{ as } S \rightarrow S^* . \quad (3.15)$$

The optimal exercise boundary will be between these two limits

$$S_0 \leq S_f(t) \leq S^*$$

and early exercise is optimal if  $S > S_f(t)$  whereas retaining the option is optimal if  $0 \leq S < S_f(t)$ .

The idea is now to use a modified Laplace transform to transform the system (3.9) into a Fredholm integral equation. Therefore, we first recall some well-known facts for the Laplace transform, which is a standard tool to solve ordinary and partial differential equations. For more details, we refer to textbooks on calculus of differential equations, in particular on literature for engineers.

## Some Basics on the Laplace Transform

**Definition 3.1.1** For  $f : [0, \infty) \rightarrow \mathbb{R}$  (or  $\mathbb{C}$ ) we call

$$\mathcal{L}[f](s) = F(s) := \int_0^{\infty} f(t)e^{-st} dt , \quad s \in \mathbb{C}$$

the Laplace transform of  $f$  (in case of existence, of course).

**Lemma 3.1.2** The Laplace transform is linear,

$$\lim_{\text{Re } s \rightarrow \infty} \mathcal{L}[f](s) = 0 , \quad \text{and}$$

$$\mathcal{L}[f^{(n)}](s) = s^n \mathcal{L}[f](s) - \sum_{k=0}^{n-1} f^{(k)}(0) s^{n-1-k} , \quad f \in C^n[0, \infty)$$

with compact support.

**Proof** (only of the last claim, the second can be done by induction):

$$\begin{aligned} \int_0^{\infty} f^{(n)}(t)e^{-st} dt &= \underbrace{f^{(n-1)}(t)e^{-st} \Big|_{t=0}^{t=\infty}}_{=-f^{(n-1)}(0)} + s \int_0^{\infty} f^{(n-1)}(t)e^{-st} dt \\ &= \dots = - \sum_{k=0}^{n-1} f^{(k)}(0) s^{n-1-k} + s^n \int_0^{\infty} f(t)e^{-st} dt. \quad \square \end{aligned}$$

The Laplace transform can be used to transform initial value problems into algebraic equations by using the second equation. We just show this by one example.

**Example 3.1.3** *The vibration equation*

$$\begin{aligned} \ddot{x} + a_1\dot{x} + a_0x &= f, \quad t \geq 0 \\ x(0) &= x_0, \quad \dot{x}(0) = x_1, \end{aligned}$$

where  $a_1 = \frac{c}{m}$ ,  $a_0 = \frac{k}{m}$ , and  $k$  is a spring and  $c$  is a damper with corresponding spring and damper constants.

Then, we obtain for  $X(s) = \mathcal{L}[x](s)$  and  $F(s) = \mathcal{L}[f](s)$ :

$$\begin{aligned} F(s) &= s^2X(s) - x_0s - x_1 + a_1sX(s) - a_1x_0 + a_0X(s) \\ &= (s^2 + a_1s + a_0)X(s) - x_1 - (a_1 + s)x_0, \end{aligned}$$

so that we obtain the algebraic equation for the Laplace transform of the unknown solution  $x$  of the original differential equation

$$X(s) = G(s)\{F(s) + x_1 + (a_1 + s)x_0\}, \quad G(s) := \frac{1}{s^2 + a_1s + a_0}.$$

The function  $G$  is called transfer function. This algebraic equation can be solved for  $X(s)$  and in order to obtain  $x(t)$  we need an inverse Laplace transform.

**Theorem 3.1.4** *Let  $f : [0, \infty) \rightarrow \mathbb{C}$  be differentiable. Then, for the convergence abscissa of  $\mathcal{L}[f] = F$  defined by*

$$\sigma_f := \inf \left\{ \alpha \in \mathbb{R} : \int_0^{\infty} f(t)e^{-\alpha t} dt < \infty \right\}$$

we have

$$\frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} F(s)e^{xs} ds = \begin{cases} f(x), & \text{if } x > 0 \\ \frac{1}{2}f(0), & \text{if } x = 0 \\ 0, & \text{if } x < 0 \end{cases}$$

for  $\gamma > \sigma_f$ .  $\square$

Again, we refer to the literature for the proof.

The above integral is often computed with the aid of the residual theorem from complex analysis.

In order to adapt the integration domain  $[0, \infty)$  to the case of the unknown contact time  $T_f(S)$ , we also consider the *modified Laplace transform* defined as follows

$$\mathcal{C}[V](S, \sigma) := \int_{-\infty}^{T_f(S)} V(S, t)e^{\sigma t} dt. \quad (3.16)$$

This corresponds to a Fourier transform where  $V(S, t) := 0$  for  $t > T_f(S)$ , i.e., in the region where it is not optimal to hold. Hence, the inverse is only meaningful where it is optimal to retain the option.

For the modified Laplace transform we collect some properties.

**Lemma 3.1.5** *We have*

$$(i) \mathcal{C}[V](S, \sigma) \rightarrow as \ s \rightarrow s^*.$$

$$(ii) \lim_{\sigma \rightarrow \infty} \sigma \mathcal{C}[V](S, \sigma) = \lim_{\sigma \rightarrow \infty} V(S_f(s), s) e^{\sigma T_f(s)}.$$

$$(iii) V(S, t) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} \mathcal{C}[V](S, \sigma) e^{\sigma t} d\sigma.$$

**Proof:** Exercise.  $\square$

## Modified Laplace Transform of the Black-Scholes Equation

Let us now come to the transformation of the Black-Scholes equation. We have

$$\begin{aligned} \mathcal{C} \left[ \frac{\partial}{\partial t} V \right] (p) &= V(T_f(s), s) e^{\varphi T_f(s)} - pV, \\ \mathcal{C} \left[ \frac{\partial}{\partial s} V \right] (p) &= \frac{\partial}{\partial s} V(s, p) - T'_f(S) V(T_f(s), s), \\ \mathcal{C} \left[ \frac{\partial^2}{\partial s^2} V \right] (p) &= \frac{\partial}{\partial s} \mathcal{C} \left[ \frac{\partial}{\partial s} V(S, t) \right] (p) - T'_f(s) \frac{\partial}{\partial s} V(T_f(s), s), \end{aligned}$$

and then we obtain the following (nonhomogeneous Euler) ODE for the unknown  $\mathcal{C} = \mathcal{C}[V](S, \sigma)$ :

$$\left[ \frac{1}{2} \sigma^2 S^2 \frac{d^2}{ds^2} + (r - \delta) s \frac{d}{ds} - (p + r) \right] \mathcal{C} + F(S) = 0, \quad (3.17)$$

where

$$F(S, T)(\sigma) = \begin{cases} 0, & \text{for } 0 < S < E \\ & \text{(where } V(S_f(t), t) = 0, \frac{\partial}{\partial s} V(S_f(t), t) = 0, \\ & T_f = T) . \\ (S - E) e^{pT}, & \text{for } E < S < S_0 \\ & \text{(where } V(S_f(t), t) = S - E, \frac{\partial}{\partial s} V(S_f(t), t) = 1, \\ & T_f = T) . \\ (S - E) e^{pT_f(s)} - [(r - \delta)(S - E)S - \sigma^2 S^2] T'_f(S) \\ & - \frac{1}{2} \sigma^2 S^2 (S - E) T''_f(S), \\ & \text{for } S_0 < S < S^* \\ & \text{(where } V(S_f(t), t) = S - E, \frac{\partial}{\partial s} V(S_f(t), t) = 1, \\ & T_f < T) . \end{cases}$$

The different cases occur according to the distinction if one is left, on or right to the line built by the contact points.

We only consider the case  $0 < \delta < r$  (the case  $\delta \geq r$  can be treated analogously).

For the ODE (3.17), one can compute the general solution (which is quite technical) and also incorporate the boundary conditions. Then, applying the inverse transform in Theorem 3.1.4 yields

$$\begin{aligned}
& \int_{S_0}^{S^*} S^{-\left(\frac{\sigma^2}{2}\right)(2\delta-2r+3\sigma^2-\lambda(p))} F(S, T) dS \\
&= \frac{1}{4} e^{pT} E^{-\left(\frac{\sigma^2}{2}\right)(2\delta-2r-\sigma^2-\lambda(p))} \times \\
& \quad \times \left[ \frac{2\delta - 2r - \sigma^2 + \lambda(p)}{p + \delta} \left( 1 - \left(\frac{r}{\delta}\right)^{-\left(\frac{\sigma^2}{2}\right)(2\delta-2r-\sigma^2-\lambda(p))} \right) \right. \\
& \quad \left. - \left( \frac{2\delta - 2r - \sigma^2 + \lambda(p)}{r + \varphi} \right) \left( 1 - \left(\frac{r}{\delta}\right)^{-\left(\frac{\sigma^2}{2}\right)(2\delta-2r-\sigma^2-\lambda(p))} \right) \right], \tag{3.18}
\end{aligned}$$

where  $\lambda(p) = [4\delta^2 - 8\delta r + 4\delta\sigma^2 + 4r^2 + 4\sigma^2 r + \sigma^4 + 8\sigma^2 p]^{1/2}$ .

Note that (3.18) is a nonlinear Fredholm integral equation for the location of the free boundary  $T_f(S) = T$ . It can numerically be solved using the techniques of Chapter 2. Even though we did not give all details, it should be clear from this example, that modelling of financial products and their numerical treatment should be considered one task. The above equations can not be assumed to be solvable analytically. Hence, one has to resort to efficient numerical schemes.

The remaining cases can be treated analogously, see [5].

## 3.2 The Binomial Method

We have seen in Numerical Finance I the binomial method for European options. Now, we give an easy and straightforward modification to American Options. It just amounts to include one projection step.

In fact, the only difference is the computation of  $V_{ji}$ , the approximation of  $V(t_i, S_{ji})$  where  $S_{ji} = S_0 u^j d^{i-j}$  so that  $(t_i, S_{ji})$  can be seen as grid points. Then, we have.

**Theorem 3.2.1 Input:**  $r, \sigma, S_0, K, M$ , choice of put or call.

- $\Delta t = \frac{T}{M}$   $u, d, p$  like in Numerical Finance I.
- $S_{00} = S_0$
- $S_{jM} = S_{00} u^j d^{M-j}$  ,  $j = 0, 1, \dots, M$   
 $S_{ji} = S_{00} u^j d^{i-j}$  ,  $i = 1, \dots, M-1$   $j = 0, 1, \dots, y$

- $V_{j,M} = \begin{cases} (S_{jM} - K)^+ & \text{for Call} \\ (K - S_{jM})^+ & \text{for Put} \end{cases}$
- $V_{j,i} = \begin{cases} \max\{(S_{ji} - K)^+, e^{-r\Delta t}(pV_{j+1,i+1} + (1-p)V_{j,i+1})\}, & \text{Call} \\ \max\{(K - S_{ji})^+, e^{-r\Delta t}(pV_{j+1,i+1} + (1-p)V_{j,i+1})\}, & \text{Put} \end{cases}$   
for  $i < M$

**Output:**  $V_{00}$  is the approximation of  $V(S_0, 0)$ .

**Example:** See [7], Exercise 1.6.

### 3.3 Obstacle Problems

Another numerical method for solving the pricing problem for American Options uses the Black-Scholes inequality (see above). This can be seen as a particular instant of an *obstacle problem*, which is also common in several areas of application.

**Example 3.3.1** Given a membrane on which a force  $-f$  acts on some domain  $\Omega \subset \mathbb{R}^2$ . The membrane is fixed on  $\partial\Omega = \Gamma$  and the displacement  $u$  of the membrane is bounded in  $\Omega$  by a given function  $g$ .

Then,  $u$  is given as the solution of the following system

$$\left\{ \begin{array}{l} \Delta u \geq f \\ u \leq g \\ (\Delta u - f)(u - g) = 0 \end{array} \right\} \text{ in } \Omega \quad u|_{\Gamma} = 0 \quad (3.19)$$

Let us now subdivide  $\Omega$  into the (unknown) contact zone

$$D_2 := \{x \in \Omega : u(x) = g(x)\}$$

and  $D_1 := \Omega \setminus D_2$ . Then we have

$$\left\{ \begin{array}{l} \bullet \text{ in } D_2 : u = g \ (\Rightarrow \Delta u = \Delta g < f) \\ \bullet \text{ in } D_1 : u < g \ \Rightarrow \Delta u = f, \end{array} \right. \quad (3.20)$$

i.e., the same behavior as for the Black-Scholes inequality:

if  $V^{am} > \text{payoff} \Rightarrow \text{Black-Scholes-equation (3.1, \dots, 3.5) holds,}$

if  $V^{am} = \text{payoff} \Rightarrow \text{Black-Scholes-inequality.}$

As opposed to (3.20), the formulation (3.19) does **not** involve the unknown  $D_2$ , (3.19) is also called *linear complementary problem*. We use this for the design of a numerical method. If a solution  $u$  of (3.19) is determined, we can compute  $D_2$  from it.

## Variational Inequalities

We have seen in Numerical Finance I that the classical (strong) formulation of boundary value problems for partial differential equations is often not appropriate. The same also holds for inequalities.

**Example 3.3.2** Consider the 1d elliptic PDE:

$$-u''(x) = f(x), x \in (0, 1), \quad u(0) = u(1) = 0$$

which leads to the variational formulation of finding  $u \in H_0^1(0, 1)$  such that

$$(\nabla u, \nabla v)_0 =: a(u, v) = (f, v)_0 \quad \forall v \in H_0^1(0, 1),$$

which is equivalent to the minimization problem

$$J(v) := \frac{1}{2}a(v, v) - (f, v)_0 \rightarrow \min \quad \text{for } v \in H_0^1(0, 1).$$

If the above minimization has to be constrained, i.e.,  $v \in H_0^1(0, 1)$  is replaced by a (convex) subset  $K \subset H_0^1(0, 1)$ , then we obtain a variational inequality (its analysis leads to the field of convex analysis).

The general form (which is also appropriate for American option pricing problems) reads as follows: Let  $H$  be a real Hilbert space and  $K \subset H$  convex,  $K \neq \emptyset$ . Further, let  $L : K \subset H \rightarrow H'$  be given. Then, one has to determine  $u \in K$  such that

$$\langle L(u), v - u \rangle \geq 0 \quad \forall v \in K \tag{3.21}$$

where  $\langle \cdot, \cdot \rangle$  is the duality pairing of  $H$  and its dual  $H'$ , i.e., for  $w \in H'$ ,  $v \in H$ , the duality pairing is defined by  $\langle w, v \rangle := w(v) \in \mathbb{R}$ .

**Remark 3.3.3** If  $K$  is a linear subspace of  $H$ , i.e.,  $v := u \pm z \in K$  for all  $u, z \in K$ . Inserting this in (3.21) yields on the one hand  $\langle L(u), z \rangle \geq 0$  and on the other hand, due to the linearity of  $\langle L(u), \cdot \rangle$  that  $\langle L(u), -z \rangle = -\langle L(u), z \rangle \geq 0$ , i.e., we obtain  $\langle L(u), z \rangle = 0$  for all  $z \in K$ , i.e., a variational equality.

**Example 3.3.4** In Example 3.3.1, we obtain  $K = \{v \in H_0^1(\Omega) : v \geq g \text{ in } \Omega \text{ a.e.}\}$  and

$$-\int_{\Omega} \nabla u \nabla (v - u) dx = -a(u, v - u) \geq (f, v - u)_0, \quad \forall v \in K, \tag{3.22}$$

i.e.,  $L : K \rightarrow V'$  is defined on all  $V \supset K$ , i.e.,

$$\langle Lu, v \rangle = -a(u, v) - \langle f, v \rangle.$$

The relation to the complementarity problem (3.19) is as follows. If (in addition to the above assumptions)  $u \in H^2(\Omega)$ , we have by integration by parts

$$-\int_{\Omega} \nabla u \nabla (v - u) \, dx = \int_{\Omega} (\Delta u)(v - u) \, dx ,$$

i.e., (3.22) reads

$$(\Delta u, v - u)_0 \geq (f, v - u)_0 \quad \forall v \in K,$$

and since  $u \in K$  we have  $u \leq g$  in  $\Omega$  a.e. Now let  $\varphi \in H_0^1(\Omega)$ ,  $\varphi \geq 0$  in  $\Omega$  a.e. Then, setting  $v := \varphi + u \in K$ , we obtain

$$(\Delta u, \varphi)_0 \geq (f, \varphi)_0 \quad \forall \varphi \geq 0 , \tag{3.23}$$

i.e.,  $\Delta u \geq f$  in  $\Omega$ . Choosing  $v := g$ , we have

$$\begin{aligned} -\underbrace{(\Delta u - f)}_{\geq 0}, \underbrace{(g - u)}_{\leq 0} &\geq 0 \text{ as well as} \\ -(\Delta u - f, g - u) &= (f - \Delta u, g - u)_0 \leq 0 \text{ by (3.23), i.e.} \end{aligned}$$

$(\Delta u - f)(g - u) = 0$ . Hence  $u$  solves the complementary problem (3.19). On the other hand, let  $u \in H^2(\Omega)$  solve (3.19), then we have from the first equation that

$$(\Delta u - f, v - g)_0 \geq 0 \quad \forall v \in K$$

as well as

$$(\Delta u - f, u - g)_0 = 0,$$

from the the second equation. Next, by subtracting the latter two equations yields

$$\begin{aligned} 0 &\leq (\Delta u - f, v - g)_0 - (\Delta u - f, u - g)_0 \\ &= (\Delta u - f, v - u)_0 \\ &= -a(u, v - u) - (f, v - u)_0, \end{aligned}$$

i.e., (3.22).

**Remark 3.3.5** *The equivalent minimization problem reads*

$$J(v) := \frac{1}{2}a(v, v) + (f, v)_0 \rightarrow \min_{v \in K}!$$

Hence, an obstacle problem may equivalently be formulated as

- (a) free boundary value problem,
- (b) linear complementary problem,
- (c) variational inequality (if  $u \in H^2(\Omega)$ ),
- (d) minimization problem (if  $u \in H^2(\Omega)$ ).

### 3.4 Finite Difference Methods

For notational convenience, we restrict ourselves to the 1D case and consider the complementary problem analogous to (3.19)

$$\left\{ \begin{array}{l} -u''(x) \geq f(x) \\ u(x) \leq g(x) \\ (u''(x) - f(x))(u(x) - g(x)) = 0 \\ u(-1) = u(1) = 0, \quad u \in C^1[-1, 1] \end{array} \right\} \quad \forall x \in (-1, 1) \quad (3.24)$$

Again, we introduce a simple equidistant grid

$$\Delta := \{-1 = x_0 < x_1 < \dots < x_m = 1\}, \quad h := \frac{2}{m}, \quad m \in N, \\ x_i = -1 + ih, \quad 0 \leq i \leq m,$$

and use the central difference approximation

$$u''(x_i) \approx \frac{1}{h^2}(-u(x_{i-1}) + 2u(x_i) - u(x_{i+1})), \quad f_i := f(x_i), \\ g_i := g(x_i),$$

i.e., we compute an approximation  $u_i \approx u(x_i)$  by

$$\left\{ \begin{array}{l} u_0 = u_m = 0 \\ (-u_{i-1} + 2u_i - u_{i+1} - h^2 f_i)(u_i - g_i) = 0 \\ u_i \leq g_i \\ -u_{i-1} + 2u_i - u_{i+1} \geq h^2 f_i \end{array} \right\} \quad 1 \leq i \leq m-1, \quad (3.25)$$

or, in matrix-vector notation

$$\left\{ \begin{array}{l} (u - g)^T (Au - f) = 0, \\ u \leq g, \\ Au \geq f, \end{array} \right. \quad (3.26)$$

where  $g := (g_1, \dots, g_{m-1})^T$ ,  $f := h^2(f_1, \dots, f_{m-1})^T$ ,  $u := (u_1, \dots, u_{m-1})^T$  and the system matrix

$$A := \begin{bmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{(m-1) \times (m-1)}.$$

Note that

$$(u - g)^T (Au - f) = \sum_{i=1}^{m-1} (-u_{i-1} + 2u_i + u_{i+1} - h^2 f_i)(u_i - g_i) = 0 \\ \iff (-u_{i-1} + 2u_i + u_{i+1} - h^2 f_i)(u_i - g_i) = 0$$

in the case  $-u_{i-1} + 2u_i + u_{i+1} \geq h^2 f_i$   $u_i \leq g_i$  (i.e., if all signs are equal).

Let us now describe a numerical (iterative) method for the solution of (3.26). First we note that (3.26) is equivalent to

$$\min\{Au - f, g - u\} = 0 \quad (\text{componentwise}). \quad (3.27)$$

This means either  $u_i = g_u$  or  $(Au)_i = f_i$ ,  $1 \leq i \leq m-1$ . We now consider the decomposition

$$A = D - L - U$$

where  $L$  is the lower left and  $U$  the upper right part of  $A$ . From now on, we assume that  $A$  is s.p.d, then

$$D_{ii} = a_{ii} > 0 \quad \forall 1 \leq i \leq m-1 \quad (3.28)$$

and since

$$Au - f = D(u - D^{-1}(Lu + Uu + f))$$

the equations (3.27) is equivalent to

$$\min\{u - D^{-1}(Lu + Uu + f), g - u\} = 0, \quad (3.29)$$

or

$$u = \max\{D^{-1}(Lu + Uu + f), g\}. \quad (3.30)$$

The general idea is to modify appropriate iterative methods for  $Au = f$  in such a way that (3.30) is incorporated in the iteration.

### 3.4.1 Classical Iterative Methods

The idea is to construct a suitable fixpoint iteration that converges towards the solution  $x^*$  of a linear system  $Ax = b$ .

If  $A \in \mathbb{R}^{n \times n}$ , choose a regular matrix  $Q \in \mathbb{R}^{n \times n}$  and then

$$\begin{aligned} Ax = b &\iff Q^{-1}(Ax - b) = 0 \\ &\iff \phi(x) := \underbrace{(I - Q^{-1}A)}_{=:G} x + \underbrace{Q^{-1}b}_{=:c} = x, \end{aligned}$$

i.e., the solution of a linear system of equation is equivalent to the fixpoint problem  $\phi(x) = x$ . Then, Banach fixpoint theorem yields a corresponding iteration:

$$x_{k+1} = \phi(x_k) = Gx_k + c.$$

Then, we have the following standard result.

**Lemma 3.4.1** *The fixpoint iteration converges for any initial value  $x_0 \in \mathbb{R}^n$  if  $\rho(G) < 1$ , where*

$$\rho(G) := \max_{1 \leq j \leq n} |\lambda_j(G)|$$

*denotes the special radius of  $G$ .*

**Proof:** Consider the singular value decomposition of  $G$

$$G = U\Sigma V^T$$

with orthogonal matrices  $U, V$  and  $\Sigma = \text{diag}(\sigma_i)$ ,  $\sigma_i = \lambda_i(G)^2 < 1$ ,

$$\Rightarrow \lim_{k \rightarrow \infty} \Sigma^k = 0$$

thus

$$\lim_{k \rightarrow \infty} G^k = u \left( \lim_{k \rightarrow \infty} \Sigma^k \right) V^T = 0,$$

which proves the claim.  $\square$

We still have the choice of the matrix  $Q$ . We describe some standard and well-known examples.

**Richardson method:** This corresponds to the choice  $Q = \alpha I$ ,  $\alpha \in \mathbb{R}$ , i.e.,

$$x_{k+1} = x_k + \alpha(b - Ax_k).$$

The particular choice  $\alpha \neq 1$  is known as *damped Richardson iteration*.

**Jacobi method:** With the decomposition  $A = L + D + U$ , where  $D = \text{diag}(A)$  is the diagonal of  $A$ , one uses  $Q = D^{-1}$ . This results in the iteration:

$$x_{k+1} := (I - D^{-1}A)x_k + D^{-1}b = -D^{-1}(L + U)x_k + D^{-1}b.$$

For this method, we have the following convergence result.

**Theorem 3.4.2** *If  $A$  is strictly diagonal dominant, i.e.,*

$$|a_{ii}| > \sum_{i \neq j} |a_{ij}| \quad \forall 1 \leq i \leq n,$$

*then the Jacobi iteration converges towards  $x = A^{-1}b$  for all  $x_0 \in \mathbb{R}^n$ .*

**Proof:** Follows by Lemma 3.4.1 since  $G = I - D^{-1}A = -D^{-1}(L + R)$  and

$$\rho(D^{-1}(L + R)) \leq \|D^{-1}(L + R)\|_\infty = \max_i \sum_{i \neq j} \left| \frac{a_{ij}}{a_{ii}} \right| < 1,$$

which proves the theorem.  $\square$

Let us now consider the number of operations:

- $\mathcal{O}(N^2)$  for dense matrices  $A \in \mathbb{R}^{N \times N}$ ,
- $\mathcal{O}(N)$  for sparse matrices

per step.

**Gauß-Seidel method:** Again, we use the decomposition  $A = L+D+U$  and set  $Q = D+L$ , which yields the iteration:

$$\begin{aligned} x_{k+1} &:= (I - (D + L)^{-1}A)x_k + (D + L)^{-1}b \\ &= -(D + L)^{-1}Ux_k + (D + L)^{-1}b, \end{aligned}$$

since

$$\begin{aligned} I - (D + L)^{-1}A &= (D + L)^{-1}(D + L - A) \\ &= (D + L)^{-1}(D + L - D - L - U). \end{aligned}$$

For this method, the following result is known.

**Theorem 3.4.3** *The Gauß-Seidel method converges for all s.p.d. matrices  $A \in \mathbb{R}^{n \times n}$ .*

For the proof, we need some preparations. For a s.p.d. matrix  $A \in \mathbb{R}^{n \times n}$ , we consider the following scalar product

$$(x, y)_A := x^T A y = (x, Ay), \quad x, y \in \mathbb{R}^n.$$

Note that  $B^* = A^{-1}B^T A$  is the *A-adjoint matrix*, i.e.,

$$(Bx, y)_A = (x, B^*y)_A$$

for all  $x, y \in \mathbb{R}^n$ . In fact, we have

$$(Bx, y)_A = x^T B^T A y = x^T A \underbrace{A^{-1}B^T A}_{=B^*} y = (x, B^*y)_A.$$

Any *A*-self-adjoint matrix  $B$  (which means that  $B^* = B$ ) is called *A-positive* if

$$(Bx, x)_A > 0 \quad \forall x \neq 0.$$

**Lemma 3.4.4** *If  $B := I - G^*G$  with  $G \in \mathbb{R}^{n \times n}$ , is *A*-positive, then  $\rho(G) < 1$ .*

**Proof:** By assumption,  $B$  is *A*-positive, i.e.

$$0 < (Bx, x)_A = (x, x)_A - (G^*, Gx, x)_A = (x, x)_A - (Gx, Gx)_A$$

which means that  $(x, x)_A > (Gx, Gx)_A$ . Thus, we have for the *A*-norm

$$\|x\|_A := \sqrt{(x, x)_A}$$

that  $\|x\|_A > \|G_x\|_A$ . Finally,

$$\rho(G) \leq \|G\|_A := \sup_{\|x\|_A=1} \frac{\|Gx\|_A}{\|x\|_A} < 1$$

since the supremum is attained due to the compactness of the unit ball with respect to  $\|\cdot\|_A$ , i.e.,  $\partial B_{1,A}(0) := \{x \in \mathbb{R}^n \mid \|x\|_A = 1\}$ .  $\square$

**Proof of Theorem 3.4.3:** Show that  $B := I - G^*G$  is  $A$ -positive for  $G = I - (D + L)^{-1}A$ . Since  $U^T = L$ , we have

$$\begin{aligned} G^* &= I - A^{-1}A^T(D + L)^{-T}A \\ &= I - (D^T + L^T)^{-1}A = I - (D + U)^{-1}A, \end{aligned}$$

and thus by standard calculations

$$\begin{aligned} B &= I - G^*G = I - (I - (D + U)^{-1}A)(I - (D + L)^{-1}A) \\ &= I - I + (D + U)^{-1}A + (D + L)^{-1}A - \underbrace{(D + U)^{-1}A}_{=(D+U)^{-1}(D+U+L)} \\ &= (D + U)^{-1}A - (D + U)^{-1}L(D + L)^{-1}A \\ &= (D + U)^{-1}D \underbrace{(D^{-1}(D + L) - D^{-1}L)}_{=I+D^{-1}L-D^{-1}L} (D + L)^{-1}A \\ &= (D + U)^{-1}D(D + L)^{-1}A. \end{aligned}$$

Hence, we obtain

$$\begin{aligned} (Bx, x)_A &= ((D + U)^{-1}D(D + L)^{-1}Ax, Ax) \\ &= (D(D + L)^{-1}Ax, \underbrace{(D + U)^{-T}Ax}_{(D+L)^{-1}}) \\ &= (D^{1/2}(D + L)^{-1}Ax, (D^{1/2}(D + L)^{-1}A)x) \\ &= \underbrace{\|D^{1/2}(D + L)^{-1}Ax\|}_{\text{regular}} > 0 \end{aligned}$$

for  $x \neq 0$  which shows that  $B$  is s.p.d. Thus, the claim follows from Lemma 3.4.4.  $\square$

### Relaxation Methods

For Gauß-Seidel, we have  $G = I - (D + L)^{-1}A$ . In order to speed up the convergence, one can introduce an additional parameter  $\omega > 0$  and obtain the iteration matrix

$$G_\omega := I - \left( \frac{1}{\omega}D + L \right)^{-1} A,$$

which means in particular, that for  $\omega = 1$ , we obtain the above mentioned Gauß-Seidel method. Thus, we obtain the iteration

$$x^{(k+1)} := \left( I - \left( \frac{1}{\omega}D + L \right)^{-1} A \right) x_k + \left( \frac{1}{\omega}D + L \right)^{-1} b.$$

The method is applied in practice as follows for a given iterate  $x^{(k)}$

$$\left\{ \begin{array}{l} \text{For } i = 1, \dots, N \\ z_i^{(k+1)} = \frac{1}{a_{ii}} \left[ - \underbrace{\sum_{m < i} a_{im} x_m^{(k+1)} - \sum_{m > i} a_{im} x_m^{(k)} + b_i}_{= (-Lx^{(k+1)} - Ux^{(k)} + b)_i} \right] \\ x_i^{(k+1)} = x_i^{(k)} + \omega (z_i^{(k+1)} - x_i^{(k)}). \end{array} \right. \quad (3.31)$$

This can be seen as follows:

$$\begin{aligned} a_{ii} x_i^{(k+1)} &= a_{ii} x_i^{(k)} + \omega \left[ - \sum_{m < i} a_{im} x_m^{(k+1)} - \sum_{m > i} a_{im} x_m^{(k)} + b_i - a_{ii} x_i^{(k)} \right] \\ \iff Dx^{(k+1)} &= Dx^{(k)} + \omega (-Lx^{(k+1)} - Ux^{(k)} + b - Dx^{(k)}) \\ \iff (D + \omega L)x^{(k+1)} &= (D - \omega \overbrace{(U + D)}^{=A-L})x^{(k)} + \omega b \\ \iff \omega \left( \frac{1}{\omega} D + L \right) x^{(k+1)} &= \omega \left[ \left( \frac{1}{\omega} D + L \right) - A \right] x^{(k)} + \omega b \\ \iff x^{(k+1)} &= \left( I - \left( \frac{1}{\omega} D + L \right)^{-1} A \right) x^{(k)} + \left( \frac{1}{\omega} D + L \right)^{-1} b. \end{aligned}$$

For  $\omega < 1$ , this is called a *damped iteration*, for  $1 < \omega < 2$  it is called *over-relaxed*, the method is also known as *SOR* (*successive over relaxation*). Details can be found e.g. in [8, §8].

**Theorem 3.4.5 (Ostrowski, Reich)** *For any s.p.d. matrix  $A \in \mathbb{R}^{n \times n}$ , the SOR method converges for all  $0 < \omega < 2$ .  $\square$*

The proof can be found in any standard textbook on Numerical Analysis. The question naturally arises what might be an optimal choice for the parameter  $\omega$ .

**Definition 3.4.6** *A matrix  $A \in \mathbb{R}^{n \times n}$  is called consistently ordered if the eigenvalues of the matrices*

$$J(\alpha) := D^{-1}(\alpha L + \alpha^{-1}U) \quad (\alpha \neq 0) \quad (3.32)$$

*are independent of  $\alpha$ , if  $A = L + D + U$ .*

The following theorem can be found e.g. in [8].

**Theorem 3.4.7** *If  $A$  is consistently ordered, then*

$$\rho(G_1) = \rho(J)^2, \quad J = J(1),$$

where  $G_1 = I - (D + L)^{-1}A = -(D + L)^{-1}U$  is the iteration matrix of the Gauß-Seidel method.  $\square$

Note that  $-J(1) = -D^{-1}(L+U)$  is the iteration method of Jacobi, thus Theorem 3.4.7 says that the Jacobi method roughly needs the double number of iterations than Gauß-Seidel (if  $A$  is consistently ordered).

Now the optimal parameter is characterized by

$$\rho(G_{\omega_{\text{opt}}}) = \min_{\omega \in \mathbb{R}} \rho(G_\omega) = \min_{0 < \omega < 2} \rho(G_\omega)$$

and the following result is known (see again [8]).

**Theorem 3.4.8 (Young, Varga)** *Let  $A$  be consistently ordered and assume that  $J = J(1)$  has only real eigenvalues such that  $\rho(J) < 1$  (see Lemma 3.4.1). Then*

$$\omega_{\text{opt}} = \frac{1}{1 + \sqrt{1 - \rho(J)^2}}, \quad \rho(G_{\omega_{\text{opt}}}) = \omega_{\text{opt}} - 1. \quad \square$$

**Remark 3.4.9** *Note that tridiagonal- and also block-Tridiagonal-matrices are consistently ordered which can easily be verified.*

### 3.4.2 Projected SOR-method for Complementary Problems

Now, we modify the above described SOR-method for solving the complementary problem

$$(Au - f)^T(u - g) = 0, \quad u \geq g, \quad Au \geq f, \quad (3.33)$$

which we have seen to be equivalent to (3.30), i.e.,

$$u = \max\{D^{-1}(Lu + Uu + f), g\},$$

if the matrix  $A = D - L - U$  is s.p.d. We add a projection step in the SOR-method (3.31) (note: there we have used the decomposition  $A = D + L + U$ ):

$$\left\{ \begin{array}{l} \text{For } i = 1, \dots, N \text{ do} \\ z_i^{(k+1)} = \frac{1}{a_{ii}}(-Lu^{(k+1)} - Uu^{(k)} + f)_i \\ u_i^{(k+1)} = \max\{u_i^{(k)} + \omega(z_i^{(k+1)} - u_i^{(k)}), g_i\} \end{array} \right. \quad (3.34)$$

which is called *projected SOR-method*.

We aim to prove that (3.34) converges towards the solution  $u$  of (3.33). We need some preparations. The following proofs are taken from [2].

**Lemma 3.4.10** *The problem (3.33) is equivalent to*

$$u \geq g, \quad J(u) = \min_{v \geq g} J(v) \quad (3.35)$$

where  $J(v) := \frac{1}{2}v^T Av = f^T v$ , if  $A$  is s.p.d.

**Proof:** Let  $u$  be a solution of (3.33) and let  $v \geq g$ , then

$$\begin{aligned} J(v) - J(u) &= \frac{1}{2}v^T Av - \frac{1}{2}u^T Au + f^T(u - v) \\ &= \frac{1}{2}(v - u)^T A(v - u) + u^T Av - u^T Au + u^T f - v^T f \\ &= \underbrace{\frac{1}{2}(v - u)^T A(v - u)}_{\geq 0 \text{ since } A \text{ is s.p.d.}} + v^T(Au - f) - u^T(Au - f) \\ &\geq (v - u)^T(Au - f) \\ &= \underbrace{(v - g)^T}_{\geq 0} \underbrace{(Au - f)}_{\geq 0 \text{ by (3.33)}} - \underbrace{(u - g)^T(Au - f)}_{=0 \text{ by (3.33)}} \geq 0, \end{aligned}$$

i.e.,  $J(u) \leq J(v)$  for all  $v \geq g$ , i.e.,  $u$  solves (3.35).

On the other hand, let  $u \geq g$  solve (3.35). Let  $v^{(k)} := u + \varepsilon \delta_k$ ,  $\varepsilon > 0$  and denote by  $\delta_k = (\delta_{1,k}, \dots, \delta_{n,k})^T$  the  $k$ -th canonical vector. This means that  $v^{(k)} \geq u \geq g$  and

$$\begin{aligned} 0 \leq J(v^{(k)}) - J(u) &= \frac{\varepsilon^2}{2} \delta_k^T A \delta_k + \varepsilon \delta_k^T (Au - f) \\ &= \frac{\varepsilon^2}{2} a_{kk} + \varepsilon (Au - f)_k, \quad \forall \varepsilon > 0, \end{aligned}$$

which implies  $0 \leq (Au - f)_k + \frac{\varepsilon}{2} a_{kk} \rightarrow (Au - f)_k$  as  $\varepsilon \rightarrow 0$  for all  $k$ , i.e.,  $Au \geq f$

Now suppose  $(Au - f)_k > 0$  and  $u_k \geq g_k$  for some  $k$ . Choose  $\varepsilon > 0$  small enough such that  $w^{(k)} := u - \varepsilon \delta^k \geq g_k$ . This implies  $0 \leq J(w^{(k)}) - J(u) = \frac{\varepsilon^2}{2} a_{kk} - \varepsilon (Au - f)_k < 0$  for  $\varepsilon$  small enough, which finally gives  $(Au - f)^T(u - g) = 0$ .  $\square$

**Theorem 3.4.11 (Cryer)** *Let  $A \in \mathbb{R}^{n \times n}$  s.p.d.,  $b, f \in \mathbb{R}^n$ ,  $1 < \omega < 2$ , Then  $\{u^{(k)}\}_{k \in \mathbb{N}}$  defined by (3.34) converges towards the unique solution  $u$  of (3.33).*

**Remark 3.4.12** *The latter theorem also states that the complementary problem (3.33) has a unique solution.*

**Proof of Theorem 3.4.11:** We split the proof in two parts.

1.) **Uniqueness of solution:**

Let  $w_1, w_2$  be two solutions of (3.33) and (3.35), respectively. Thus, we have by

Lemma 3.4.10

$$\begin{aligned}
0 &= J(w_1) - J(w_2) \\
&= \underbrace{\frac{1}{2}(w_1 - w_2)^T A(w_1 - w_2)}_{\geq 0} + \underbrace{w_1^T(Aw_2 - f) - w_2^T(Aw_2 - f)}_{=0} \\
&= (w_1 - w_2)^T(Aw_2 - f) \\
&= \underbrace{(w_1 - g)^T}_{\geq 0} \underbrace{(Aw_2 - f)}_{\geq 0} - \underbrace{(w_2 - g)^T(Aw_2 - f)}_{=0} \\
&\geq \frac{1}{2}(w_1 - w_2)^T A(w_1 - w_2) \geq 0,
\end{aligned}$$

i.e.,  $0 = (w_1 - w_2)^T A(w_1 - w_2)$ , which means that  $w_1 - w_2 = 0$ .

## 2.) Existence of a solution:

The idea is to show the convergence of  $u^{(k)}$  in (3.34).

a) For all  $i, k$ , there exists some  $\omega_k \in [0, \omega]$  such that for  $u_i^{(k+1)} := u_i^{(k)} + \omega_{ik}(z_i^{(k+1)} - u_i^{(k)})$  we have

- If  $g_i \leq u_i^{(k)} + \omega(z_i^{(k+1)} - u_i^{(k)})$  and thus  $\omega_{ik} = \omega$ .
- If  $g_i > u_i^{(k)} + \omega(z_i^{(k+1)} - u_i^{(k)})$  and thus we have  $u_i^{(k+1)} = g_i$  since  $u_i^{(k)} \geq g_i$  (because  $u_i^{(k)} = \max\{\dots, g_i\}$ ), we have  $z_i^{(k+1)} - u_i^{(k)} < 0$  and hence

$$\frac{u_i^{(k)} - g_i}{u_i^{(k)} - z_i^{(k+1)}} \geq 0$$

and  $\omega_{i,k} < \omega$ . This implies that  $u_i^{(k)} + \omega_{i,k}(z_i^{(k+1)} - u_i^{(k)}) = u_i^{(k)} - u_i^{(k)} + g_i = g_i = u_i^{(k+1)}$ .

b) Set  $u^{(k,i)} = (u_1^{(k+1)}, \dots, u_i^{(k+1)}, u_{i+1}^{(k)}, \dots, u_N^{(k)})^T$  and

$$J_j := J(u^{(k,i)}), \quad j := N(k-1) + i.$$

Note that

$$u^{(k+1,0)} = (u_1^{(k+1)}, \dots, u_N^{(k+1)})^T := u^{(k,N+1)}$$

and

$$(u^{(k,i)} - u^{(k,i-1)}) = (u_i^{(k+1)} - u_i^{(k)})\delta_i, \quad \delta_i := (\delta_{1,i}, \dots, \delta_{N,i})^T, \quad (3.36)$$

since by (3.34)

$$\begin{aligned}
a_{ii}(z_i^{(k+1)} - u_i^{(k)}) &= (-Lu^{(k+1)} - Uu^{(k)} + f)_i - a_{ii}u_i^{(k)} \\
&= -(Au^{(k,i)})_i + f_i = -(Au^{(k,i)} - f)_i.
\end{aligned} \quad (3.37)$$

Thus, we obtain

$$\begin{aligned}
J_j - J_{j-1} &= J(u^{(k,i)}) - J(u^{(k,i-1)}) \\
&= \frac{1}{2}(u^{(k,i)} - u^{(k,i-1)})^T A(u^{(k,i)} - u^{(k,i-1)}) \\
&\quad + (u^{(k,i)} - u^{(k,i-1)})^T (Au^{(k,i)} - f) \\
&= \frac{1}{2}a_{ii}(u_i^{(k+1)} - u_i^{(k)})^2 - a_{ii}(u_i^{(k+1)} - u_i^{(k)}) \underbrace{(z_i^{(k+1)} - u_i^{(k)})}_{= \frac{1}{\omega_{ik}}(u_i^{(k+1)} - u_i^{(k)})} \\
&= -\frac{a_{ii}}{2} \left( \frac{2}{\omega_{ik}} - 1 \right) (u_i^{(k+1)} - u_i^{(k)})^2 \\
&\leq \underbrace{\frac{a_{ii}}{2}}_{>0} \underbrace{\left( \frac{2}{\omega} - 1 \right)}_{\geq 0} \underbrace{(u_i^{(k+1)} - u_i^{(k)})^2}_{\geq 0} \quad \text{if } \omega_{ik} > 0 \\
&\leq 0.
\end{aligned}$$

If  $\omega_{ik} = 0$ , we have  $u_i^{(k+1)} = u_i^{(k)}$ , hence  $J_j = J_{j-1}$ , i.e.,  $J_j \searrow (j \rightarrow \infty)$ .

On the other hand,  $J_j = \frac{1}{2} \underbrace{u^{(k,i)T} Au^{(k,i)}}_{\geq 0} - \underbrace{f^T u^{(k,i)}}_{\leq c}$  which implies the existence of

the limit  $J = \lim_{j \rightarrow \infty} J_j$

c) A standard estimate yields for any component index  $i$

$$\begin{aligned}
|u_i^{(k+1)} - u_i^{(k)}| &= \left( \frac{2}{a_{ii}(\frac{2}{\omega_{ik}} - 1)} (J_{j-1} - J_j) \right)^{1/2} \\
&\leq \left( \frac{2}{\min_i a_{ii}(\frac{2}{\omega_{ik}} - 1)} (J_{j-1} - J_j) \right)^{1/2} \rightarrow 0,
\end{aligned}$$

which means that  $\{u_i^{(k)}\}_k$  is a Cauchy sequence. Thus, there exists some  $u_i$  such that  $\lim_{k \rightarrow \infty} u_i^{(k)} = u_i$ .

d) If we define

$$z_i := \lim_{k \rightarrow \infty} z_i^{(k+1)} = \frac{1}{a_{ii}} (-Lu - Uu + f)_i = u_i - \frac{1}{a_{ii}} (Au - f)_i$$

we get

$$u_i = \max\{u_i + \omega(z_i - u_i), f_i\} = \max\{u_i - \omega \frac{1}{a_{ii}} (Au - f)_i, f_i\}.$$

Thus, we have  $\min\{\frac{\omega}{a_{ii}} (Au - f)_i, u_i - f_i\} = 0$ , which is equivalent to (3.33), which proves the theorem.  $\square$

# Bibliography

- [1] C. Geiger, C. Kanzow, Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben, Springer 1999.
- [2] M. Günther, A. Jüngel, Finanzderivate mit MATLAB, Vieweg 2003.
- [3] W. Hackbusch, Integralgleichungen, Theorie und Numerik, Teubner, 1989.
- [4] H. Heuser, Funktionalanalysis, 3. Aufl., Teubner 1992.
- [5] R. Mallier, G. Alobaidi, Laplace transforms and American options, Appl. Math. Finance 7 (2000), 241-256.
- [6] A. Quateroni, R. Sacco, F. Saleri, Numerische Mathematik 2, Springer 2002.
- [7] R. Seydel, Tools for Computational Finance, Springer 2002.
- [8] J. Stoer, R. Bulirsch, Numerische Mathematik 2, Springer, Berlin, Heidelberg 2000.
- [9] K. Yosida, Functional analysis, Reprint of the sixth (1980) edition, Springer, Berlin, 1995.