Numerical Solution of Discretised HJB Equations with Applications in Finance

Jan Hendrik Witte

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2011

Me siento profundamente en deuda con Klaus, por ser quien es, y sin el cual esta tesis no hubiese sido posible. Muchísimas gracias, amigo.

Simplicity is the keynote of all true elegance.

Coco Chanel, interview in Harper's Bazaar (1923)

Acknowledgements

It is my great pleasure to thank all the people who have supported me in writing this thesis.

I am most grateful to my supervisor, Christoph. I would like to thank him for his immeasurable enthusiasm and support. His door was always open for me, and he has made writing this thesis fun.

I would like to thank Guy Barles, Peter Forsyth, Mike Giles, and Raphael Hauser for helpful discussions and ideas.

I consider myself very fortunate to be able to acknowledge financial support from Balliol College, the UK Engineering and Physical Sciences Research Council (EPSRC), and the Oxford-Man Institute of Quantitative Finance.

I would like to thank all the good friends I have made over the course of the past few years, and who have made my time in Oxford such an unforgettable experience. In particular, this includes: Justin, who is a character and a half, and who has taught me most of the English I know, always distinguishing between upper class English, American English, and 'New York', which he speaks all fluently; Jérémie, who always put up with my mediocre French, and whom I have met for a weekly lunch for almost four years now; Philipp 'Roub' Reuss, who likes hummous, and who has been a dear friend since my first days in England; and, finally, my office mates Alok, Andrew, Daniel F, Greg, Nikolay, Till, and, most of all, Daniel S, who helped create a work environment that was always bursting with hilariously funny, juvenile, and non-maths related energy.

I would like to thank my girl friend, Anna, for her continuous love, support, and patience, for putting up with me being abroad, and not minding my eccentrism. I would also like to thank my siblings, Christina and Robert, for always being there for me.

Above all, I would like to thank my parents, for their unconditional support in everything, and for always believing in me; without them, none of this would have been possible.

Many thanks to all of you.

Abstract

We consider the numerical solution of discretised Hamilton-Jacobi-Bellman (HJB) equations with applications in finance.

For the discrete linear complementarity problem arising in American option pricing, we study a policy iteration method. We show, analytically and numerically, that, in standard situations, the computational cost of this approach is comparable to that of European option pricing. We also characterise the shortcomings of policy iteration, providing a lower bound for the number of steps required when having inaccurate initial data.

For discretised HJB equations with a finite control set, we propose a penalty approach. The accuracy of the penalty approximation is of first order in the penalty parameter, and we present a Newton-type iterative solver terminating after finitely many steps with a solution to the penalised equation.

For discretised HJB equations and discretised HJB obstacle problems with compact control sets, we also introduce penalty approximations. In both cases, the approximation accuracy is of first order in the penalty parameter. We again design Newton-type methods for the solution of the penalised equations. For the penalised HJB equation, the iterative solver has monotone global convergence. For the penalised HJB obstacle problem, the iterative solver has local quadratic convergence.

We carefully benchmark all our numerical schemes against current state-of-the-art techniques, demonstrating competitiveness.

Contents

List of Figures

Chapter 1

Introduction

In this first chapter, we give an outline of how this thesis is structured, and what it aims to achieve.

1.1 Motivation

An option is a financial instrument that gives its buyer the right, but not the obligation, to buy (or sell) an asset at an agreed price at a certain future date T. There are two main types of options, namely

- European options, which can only be exercised at expiry T,
- and American options, which can be exercised at any time $t \leq T$,

but many other, often non-standardised, options circulate as well. Even though of particular importance to both parties involved in the contract, i.e. option holder and writer (in many cases a big financial institution), the price of an option cannot trivially be obtained from the market price of the underlying, and much of modern mathematical finance – in research and industry – is centred around the modelling and valuation of financial options (cf. [67]).

In their monumental work [11], Fischer Black and Myron Scholes established that, under a number of simplifying assumptions, the value V of a (European-style) financial derivative on an asset S solves the partial differential equation (PDE)

$$-\frac{\partial V}{\partial t} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rS\frac{\partial V}{\partial S} + rV = 0, \tag{1.1.1}$$

where r denotes the market interest rate and σ denotes the volatiliy of S; equation (1.1.1) has been much celebrated and is commonly known as the *Black-Scholes PDE*. In today's financial markets, most models are, in one way or the other, based on or related to the Black-Scholes PDE (cf. [23, 67, 61]).

Even though conceptually groundbreaking, the standard Black-Scholes setup has severe limitations, and being able to numerically solve (1.1.1) usually does not suffice for computing prices of more complex contracts in modern, more-flexible, models. For example, pricing an American option with payoff P in the Black-Scholes model requires the solution (cf. [23]) of

$$\min\left\{-\frac{\partial V}{\partial t}-\frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2}-rS\frac{\partial V}{\partial S}+rV,\, V-P\right\}=0, \tag{1.1.2}$$

since, at every point in time, a decision between holding the option (in which case (1.1.1) is satisfied) and exercising it (in which case $V = P$) has to be made; if we have a model where the optimal policy switches between several possible actions (rather than merely holding and exercise), then we have to add even more operators to the equation. To put this more generally: many of today's pricing models, which includes uncertain volatility models [48], different borrowing and lending rates [10, 3], transaction costs problems [47], gas storage valuation [17, 22], indifference pricing [15], and early exercise in incomplete markets [54], result in equations of the form

$$\inf_{u\in\mathbf{U}}\left\{\mathcal{L}_u V\right\}=0 \tag{1.1.3}$$

$$\text{or}\quad \inf_{u\in\mathbf{U}}\left\{\sup_{v\in\mathbf{V}}\left\{\mathcal{L}_{u,v}V\right\}\right\}=0, \tag{1.1.4}$$

where $\{\mathcal{L}_u\,;\,u\in\mathbf{U}\}$ and $\{\mathcal{L}_{uv}\,;\,(u,v)\in\mathbf{U}\times\mathbf{V}\}$ are families of linear (or affine) differential operators. Equations (1.1.3) and (1.1.4) are called Hamilton-Jacobi-Bellman (HJB) and Isaacs equations, respectively.

1.1.1 Interpreting the Equations

For linear PDEs like (1.1.1), we can expect a smooth solution, and all differential terms can be interpreted as 'classical' derivatives. However, the solution to (1.1.2) is split into two halves – continuation and exercise region – by the free boundary (cf. [23]), and the second derivative of the solution is possibly not defined on the free boundary; hence, the solution to equation (1.1.2) cannot be thought of in terms of classical solutions, but we need to look at things differently; one standard approach is to rewrite the problem as a variational inequality (cf. [44, 24]) and interpret the solution in terms of 'weak' derivatives (in the Sobolev sense, e.g. see [1]). Unfortunately, both approaches, classical and weak solutions, are not well suited for HJB and Isaacs equations since, first, we are likely to have even less regularity than in (1.1.2), and, second, any variational formulation will be very specific to a particular control set $\mathbf{U}$ (or $\mathbf{V}$).

A convenient tool for dealing with non-linear equations like (1.1.3) and (1.1.4) is the notion of viscosity solution (cf. [40]), which combines two essential properties.

- The main requirement of the viscosity solution theory is for PDEs to satisfy an *ellipicity* condition, which essentially means monotonicity in the diffusion operator, a property that is not distorted by the 'inf' and 'sup' expressions above. Therefore, for a large class of HJB and Isaacs equations, existence and uniqueness of viscosity solutions can be shown relatively easily (cf. [8]).

- Many models from mathematical finance have a stochastic setting, and, frequently, the optimal policy in stochastic terms can be uniquely linked to the viscosity solution of the corresponding HJB or Isaacs equation; more precisely, by the dynamic programming principle, the value function of a stochastic control problem can often be shown to be the unique viscosity solution of the corresponding HJB or Isaacs equation (cf. [72, 56]).

Therefore, viscosity solutions present a meaningful way of looking at HJB and Isaacs equations, flexibly dealing with non-linearities while guaranteeing a connection to the underlying stochastic model.

1.1.2 Choosing a Numerical Approach

Now, since most financial models have descriptive purposes, knowing that a solution to an HJB or Isaacs equation exists is not quite enough, but we also have to 'find' the solution, which – in absence of an explicit representation – usually means solving the equations numerically.

For the numerical solution of linear PDEs, like (1.1.1), there is a large number of numerical schemes available (cf. [4, 64]), many of them – in one way or the other – related to finite difference or finite element methods; however, to perform a typical convergence analysis (cf. [4, 24]), we need a sufficiently smooth solution for the former, and a variational formulation for the latter.

For lack of alternatives, non-linear PDEs, like (1.1.3) and (1.1.4), are traditionally often solved by finite difference techniques, which feels natural but is mathematically not trivially justified (and not necessarily infallible, as shown in [27]). Fortunately, a rigorous framework (see [6, 8] and references therein) exists which guarantees pointwise convergence of finite difference schemes to the unique viscosity solution if the following conditions are met.

- The PDE satisfies a strong comparison principle, which allows the construction of upper and lower estimates of the unique viscosity solution. For many equations arising in finance, strong comparison principles can be found (cf. [8]).
- The numerical scheme is stable and consistent. Given their intuitive nature, finite difference approximations are usually consistent, whereas stability has to be treated more carefully (cf. [64, 60]).
- The discretisation is monotone. This is the discrete equivalent of the ellipticity condition for the continuous equations (cf. [6]).

Altogether, when constructing numerical schemes with a meaningful concept for convergence for HJB and Isaacs equations, our main attention has to be on their stability and monotonicity properties.

1.1.3 Discretising the Equations

To solve a linear PDE, like (1.1.1), by a finite difference scheme, we discretise the solution space in time and space, and, for every time step $m \to m-1$, we have to find V^{m-1} such that

$$V^{m-1} = BV^m \quad \text{(Fully Explicit)} \tag{1.1.5}$$

$$\text{or} \quad AV^{m-1} = V^m \quad \text{(Fully Implicit)}, \tag{1.1.6}$$

where A and B are matrices, and vector V^m is known from the previous time step (cf. [60]). Explicit schemes have the advantage of being very easy to implement, but they suffer from severe stability constraints in the form of time step constraints; for non-linear equations in particular, a priori bounds on the time step size are difficult to obtain. In contrast, implicit schemes are frequently unconditionally stable, but they require the solution of a linear system of equations in every step, as in (1.1.6).

When applying a standard fully implicit (or partially implicit, like for Crank-Nicolson) finite difference discretisation to the differential operators in (1.1.3) and (1.1.4), we obtain non-linear discrete systems of the form

$$\min_{u\in\mathbf{U}} \left\{ A_u\, x - b_u \right\} = 0 \tag{1.1.7}$$

$$\text{and} \quad \min_{u\in\mathbf{U}} \left\{ \max_{v\in\mathbf{V}} \left\{ A_{u,v}\, x - b_{u,v} \right\} \right\} = 0, \tag{1.1.8}$$

which, for every time step, we have to solve for a vector x; such discretisations preserve many of the advantages of (1.1.6), like stability, but, in return, usually require iterative methods for their solution (if they can be solved at all).

Definition 1.1.1. *(M-Matrix) Following [25], we introduce Z_N, $N \in \mathbb{N}$, to be the set of all real square matrices of dimension N whose off-diagonal entries are all non-positive, i.e.*

$$Z_N := \{A = (a_{r,s}) \in \mathbb{R}^{N\times N} : a_{r,s} \le 0,\, r \ne s\},$$

and define

$$K_N := \{A \in Z_N : \exists\, x \ge 0 \text{ s.t. } Ax > 0\}$$

to be the set of M-matrices in $\mathbb{R}^{N\times N}$; it can be found in [25] that $A^{-1} \ge 0$ for all matrices $A \in K_N$.

Remark 1.1.2. *There are two very helpful types of conditions sufficient for the M-matrix property to hold.*

- *(Tridiagonal Matrix) $A \in Z_N$ is an M-matrix if it is tridiagonal, has positive diagonal entries, has non-negative row sums, and has at least one positive row sum.*

- *(Strictly Diagonally Dominant Matrix) $A \in Z_N$ is an M-matrix if it is strictly diagonally dominant, i.e.*

$$K_N^o := \left\{ A = (a_{r,s}) \in Z_N : a_{s,s} > \sum_{r \neq s} |a_{s,r}| \right\} \subset K_N \,.$$

Proof. Both special cases can be shown by explicitly applying Definition 1.1.1. $\square$

This thesis is concerned with the question of how to efficiently solve discrete HJB and Isaacs equations like (1.1.7) and (1.1.8), respectively. Our main assumption is that all matrices A we deal with are M-matrices (see Definition 1.1.1), which means they satisfy, in particular, $A^{-1} \geq 0$. From an algorithmic point of view, we require the M-matrix property to prove monotonicity and, thereby, convergence of iterative solvers for the above non-linear discrete equations. From a methodological point of view, the M-matrix property is closely linked to the monotonicity condition for finite difference discretisations mentioned above (cf. [6, 65]), and, hence, it is usually naturally satisfied whenever a discretisation can be shown to converge.

In summary, we can say that, if we can solve the non-linear discrete systems (1.1.7) and (1.1.8), then, combined with a convergence analysis of the finite difference discretisations, we have a fully rigorous numerical approach for the solution the continuous HJB and Isaacs equations given in (1.1.3) and (1.1.4).

1.2 Existing Methods for the Numerical Solution of the Discretised Equations

We will use this section to briefly describe the most important numerical schemes that are currently used for the solution of the non-linear discrete problems introduced above, which includes PSOR, penalisation, and two forms of policy iteration.

1.2.1 Projected Successive Overrelaxation (PSOR)

Equation (1.1.2), which, in discretised form, becomes

$$\min\{Ax - b, \, x - c\} = 0, \tag{1.2.1}$$

is traditionally solved by PSOR, a method invented by Cryer (cf. [20, 2]). Equation (1.2.1) can be equivalently written as

$$\begin{aligned} Ax - b &\geq 0, \\ x &\geq c, \\ (Ax - b)^T (x - c) &= 0, \end{aligned}$$

which is termed a linear complimentarity problem (LCP), and PSOR is an adaption of SOR, a linear system solver, to discrete LCPs. SOR itself is an extension of the Gaus-Seidel method, which in turn is an improvement of the Jacobi method (see [25] for details on these relationships).

Algorithm 1.2.1. *Choose* ω_R *satisfying* $0 < \omega_R < 2$. *Let* $x^0 = (x_i) \in \mathbb{R}^N$ *be some starting value. Then, for known* x^n, $n \geq 0$, *compute* x^{n+1} *by*

$$\begin{aligned} r_i^{n+1} &:= b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{n+1} - \sum_{j=i}^{n} a_{ij} x_j^n \\ \text{and} \quad x_i^{n+1} &:= \max\{c_i\,, x_i^n - \omega_R\, r_i^{n+1}/a_{ii}\} \end{aligned}$$

for $1 \leq i \leq N$, *where we use the notation* $A = (a_{ij}) \in \mathbb{R}^{N \times N}$, $b = (b_i) \in \mathbb{R}^N$ *and* $c = (c_i) \in \mathbb{R}^N$.

Theorem 1.2.2. *Suppose* A *is symmetric positive definite. Let* $(x^n)_{n\geq 0}$ *be the sequence generated by Algorithm 1.2.1. As* $n \to \infty$, x^n *converges to a limit* x^* *which solves* (1.2.1).

The PSOR algorithm is essentially the method of SOR applied to $Ax = b$ while enforcing the additional requirement that $x \geq c$. A generalisation of the convergence criteria, e.g., to the case of non-symmetric matrices, can be found in [2]; however, most of these more general results then require detailed knowledge of the structure of A, and some properties are non-trivial to check.

Results on how to optimally choose ω can also be found in [20]; since these results are imperfect, and the optimal estimates of ω are usually related to the spectral radius of A, which then may have to be computed first, the necessity of choosing ω correctly for optimal performance is a clear drawback of PSOR.

1.2.2 Penalisation

A penalty approach for the numerical solution of discrete LCPs in an option pricing context was introduced in [29]. The penalty approach is different in nature from the other methods discussed in this section since it consist of two steps: first, the penalty approximation of the considered equation, and, second, the iterative solution of the regularised problem.

Penalisation has long been used for the approximation and regularisation of variational inequalities (cf. [44, 9]), but, most importantly, as was shown in [29], when approximating a discrete LCP by penalisation, the resulting non-linear problem can powerfully be solved by Newton's method.

The penalty approximation to (1.2.1) is given by

$$Ax_\rho - b - \rho \max\{c - x_\rho\,, 0\} = 0, \tag{1.2.2}$$

where $\rho > 0$ is some sufficiently large penalty parameter. For $x \in \mathbb{R}^N$, we define the Jacobian

$$J(x) := A + \rho\, \mathbf{I}_{\{c-x>0\}},$$

where $\mathbf{I}_{\{c-x>0\}} \in \mathbb{R}^{N\times N}$ is defined to be an identity matrix in which we set a row i, $1 \le i \le N$, to be equal to zero if the i-th element of vector $\max\{c-x, 0\}$ equals zero.

Algorithm 1.2.3. *(Newton-like Method for Penalised Eq.) Let $x^0 \in \mathbb{R}^N$ be some starting value. Then, for known x^n, $n \ge 0$, find x^{n+1} such that*

$$J(x^n)(x^{n+1} - x^n) = -(Ax^n - b - \rho \max\{c - x^n, 0\}).$$

Theorem 1.2.4. *Suppose A is an M-matrix. There exists a constant $C > 0$, independent of ρ, such that*

$$\|x^* - x_\rho\| \le C/\rho, \tag{1.2.3}$$

where x^* and x_ρ are the solutions to (1.2.1) *and* (1.2.2)*, respectively. Let $(x^n)_{n\ge 0}$ be the sequence generated by Algorithm 1.2.3. We have $x^{n+1} \ge x^n$ for $n \ge 1$. As $n \to \infty$, x^n converges (in finitely many steps) to a limit x^* which solves* (1.2.1)*.*

Proof. See [29] or [69]. □

1.2.3 Policy Iteration

The method of policy iteration in its current form originates in early works on Markov chain approximation (cf. [46]). More recently, in [28], it has been introduced to the case of discrete HJB equations, like (1.1.7), and it can by now be considered as the standard approach for the solution of such equations.

Algorithm 1.2.5. *(Policy Iteration for HJB Eq.) For $1 \le i \le N$ and $y \in \mathbb{R}^N$, define $u_i^{min}(y) \in \mathbf{U}$ to be such that*

$$u_i^{min}(y) = \arg\min\{(A_v\, y - b_v)_i : v \in \mathbf{U}\},$$

and set $A^{min}(y) \in \mathbb{R}^{N\times N}$ and $b^{min}(y) \in \mathbb{R}^N$ to be matrix and vector consisting of rows $(A_{u_i^{min}(y)})_i$ and $(b_{u_i^{min}(y)})_i$, $1 \le i \le N$, respectively. Let $x^0 \in \mathbb{R}^N$ be some starting value. Then, for known x^n, $n \ge 0$, find x^{n+1} such that

$$A^{min}(x^n)\, x^{n+1} = b^{min}(x^n). \tag{1.2.4}$$

Theorem 1.2.6. *Suppose $\{A_u\,; u \in \mathbf{U}\}$ is a family of M-matrices, with $\mathbf{U}$ being a compact set. Suppose $u \mapsto A_u$ is continuous. Let $(x^n)_{n\ge 0}$ be the sequence generated by Algorithm 1.2.5. We have $x^{n+1} \ge x^n$ for $n \ge 1$. As $n \to \infty$, x^n converges to a limit x^* which solves* (1.1.7)*. If $|\mathbf{U}| < \infty$, we have convergence in finitely many steps.*

Proof. See [28] or [50]. □

1.2.4 Nested Policy Iteration

In [50], a version of policy iteration for the solution of discretised Isaacs equations, i.e. problems of the form (1.1.8), was introduced. Together with value iteration (cf. [46, 28]), this is one of the few existing algorithms for the numerical solution of such problems in their general form. (As discussed in [28], especially for fine grids, policy iteration can be expected to be much superior to value iteration.)

Algorithm 1.2.7. *For $1 \leq i \leq N$ and $x \in \mathbb{R}^N$, define $v_i^{min}(x) \in \mathbf{V}$ such that*

$$\max_{u \in \mathbf{U}} (A_{u,v_i^{min}(x)}\, x - b_{u,v_i^{min}(x)})_i = \min_{v \in \mathbf{V}} \Big\{ \max_{u \in \mathbf{U}} (A_{u,v}\, x - b_{u,v})_i \Big\}.$$

Let $x^0 \in \mathbb{R}^N$ be some starting value. Then, for known x^n, $n \geq 0$, find x^{n+1} such that

$$\max_{u \in \mathbf{U}} (A_{u,v_i^{min}(x^n)}\, x^{n+1} - b_{u,v_i^{min}(x)})_i = 0, \quad 1 \leq i \leq N. \tag{1.2.5}$$

Theorem 1.2.8. *Suppose $\{A_{u,v}\,; u \in \mathbf{U},\, v \in \mathbf{V}\}$ is a family of M-matrices, with $\mathbf{U}$ and $\mathbf{V}$ being compact sets. Suppose $(u, v) \mapsto A_{u,v}$ is continuous. Let $(x^n)_{n \geq 0}$ be the sequence generated by Algorithm 1.2.7. We have $x^{n+1} \geq x^n$ for $n \geq 1$. As $n \to \infty$, x^n converges to a limit x^* which solves* (1.1.8)*. If $|\mathbf{U}| + |\mathbf{V}| < \infty$, we have convergence in finitely many steps*

Proof. See [50]. □

It is interesting to note that, in case of equation (1.2.1), the two apparently distinct approaches outlined in Sections 1.2.2 and 1.2.3, namely penalisation and policy iteration, are closely related by an evergreen: Newton's method. It has been shown in [50] that policy iteration is equivalent to the primal-dual active set strategy, and, in turn, it has been shown in [43] that the latter is equivalent to a semi-smooth Newton method. Similarly, penalty approximation is generally only the first step for a subsequent application of a Newton-like scheme (see Algorithm 1.2.3 above); i.e., in this case, penalty approximation can in essence be viewed as a pre-processor. Also, in Section 2.4, we discuss how policy iteration can be interpreted as a special kind of Newton-based penalty iteration.

PSOR is different in that it cannot be extended to more general HJB equations, which, in contrast, is possible for policy iteration (see Section 1.2.4) as well as penalisation (see Chapters 3 and 4).

1.2.5 What is Missing

Generally, it can be said that the numerical solution of discretised HJB equations has not been studied extensively. With the possible exception of the discrete LCP, for which several algorithms exist, to the best of our knowledge, only one method has been brought forward for all other equations, namely policy iteration. Therefore, there is a clear need for devising competitive approaches, measuring relative performance, broadening the general understanding of how such equations can be solved, and deciding when a particular approach is preferable.

1.3 Structure of this Thesis

Throughout this thesis, we study a number of algorithms for the numerical solution of non-linear equations arising in finance. We generally put our emphasis on fully-implicit time stepping schemes since stability of the discretisation is one of the main requirements when proving convergence to the continuous equations in a viscosity setting. We start out by studying an American option problem, the simplest non-trivial HJB equation possible, and we then look at more general HJB equations with discrete and continuous control sets; we finish off by studying an HJB obstacle problem, which is a special case of an Isaacs equation. While cross-referencing and avoiding redundancies, the chapters are kept largely self-contained and allow for seperate reading.

Chapter 2. We consider the pricing of American options in a Black-Scholes model, i.e. we numerically solve equation (1.1.2), which, in discretised form, becomes (1.2.1), which is a special case of (1.1.7). We solve (1.2.1) by the method of policy iteration (cf. [28]), and we derive upper and lower bounds for the maximum number of required iterations. We present numerical results demonstrating the competitiveness of the method compared to PSOR (cf. [20]) and penalisation (cf. [29]). The chapter is based on our work in [68].

Chapter 3. We consider finitely controlled HJB equations, i.e. we study the numerical solution of equations of the form (1.1.3) with $|\mathbf{U}| < \infty$, which, in discretised form, becomes (1.1.7). We propose the penalty approximation

$$A_{u_0}\, x_\rho - b_{u_0} - \rho \sum_{u\in\mathbf{U}\setminus\{u_0\}} \max\{b_u - A_u\, x_\rho\, , 0\} = 0, \tag{1.3.1}$$

where $u_0 \in \mathbf{U}$ and $\rho > 0$ large. We show that (1.3.1) approximates (1.1.7) as $\rho \to \infty$, and we compute x_ρ in (1.3.1) using a Newton-like method. We demonstrate numerical competitiveness of our combined approach compared to policy iteration (cf. [28]) by pricing a European option in a model with different borrowing/lending rates and stock borrowing fees. The chapter is based on our work in [69].

Chapter 4. We consider continuously controlled HJB equations, i.e. we study the numerical solution of equations of the form (1.1.3) where, in contrast to Chapter 3, we now assume $\mathbf{U}$ to be a (possibly infinite) compact set; in addition, we also consider the HJB obstacle problem

$$\inf\Big\{ \sup_{u\in\mathbf{U}} \{\mathcal{L}_u V\}, \tilde{\mathcal{L}} V \Big\} = 0, \tag{1.3.2}$$

where $\tilde{\mathcal{L}}$ is another affine differential operator, which is a special case of the Isaacs equation in (1.1.4). In discretised form, we then obtain (1.1.7) for the HJB equation and for the HJB obstacle problem

$$\min\Big\{ \max_{u\in\mathbf{U}} \{A_u\, x - b_u\}, \tilde{A}x - \tilde{b} \Big\} = 0, \tag{1.3.3}$$

where (1.3.3) is again a special case of (1.1.8). We propose penalty approximations

$$A_{u_0} x_\rho - b_{u_0} - \rho \max_{u \in \mathbf{U} \setminus \{u_0\}} \Big\{ \max \big\{ b_u - A_u x_\rho , 0 \big\} \Big\} = 0 \tag{1.3.4}$$

$$\text{and} \quad \max_{u \in \mathbf{U}} \big\{ A_u x_\rho - b_u \big\} - \rho \max \big\{ \tilde{b} - \tilde{A} x_\rho , 0 \big\} = 0, \tag{1.3.5}$$

where $u_0 \in \mathbf{U}$ and $\rho > 0$ large. We show that (1.3.4) and (1.3.5) approximate (1.1.7) and (1.3.3), respectively, as $\rho \to \infty$, and, as already in Chapter 3, we use Newton-like methods for the solution of the penalised equations. Since, clearly, every finite set is also compact, the situation of Chapter 3 is also covered by this chapter, and (1.3.4) is an alternative penalty approximation to (1.2.1); however, the treatment of infinite compact sets $\mathbf{U}$ in this chapter requires paying close attention to the properties of the maps $u \mapsto A_u$ and $u \mapsto b_u$, and many of the techniques applied in Chapter 3 are conceptually much simpler. As already in Chapter 3, we demonstrate numerical competitiveness of our techniques compared to policy iteration (cf. [28, 50]); here, we solve an indifference pricing problem and an early exercise problem in an incomplete market. The chapter is based on our work in [70].

Chapter 5. We summarise the results presented in this thesis, conceptually link the different approaches considered in the previous chapters, and discuss the continuation of our work.

Chapter 2

Policy Iteration for American Options

In this chapter, we study policy iteration, introduced in the context of Hamilton-Jacobi-Bellman (HJB) equations in [28], as a tool for solving problems of the form (1.2.1), which result from the finite difference and finite element approximation of American options. We show that, in general, $O(N)$ is an upper and lower bound on the number of iterations needed to solve a system of size N (i.e. when having a space discretisation with N nodes where the discretisation matrix $A \in \mathbb{R}^{N \times N}$). If embedded in a class of standard discretisations with M time steps, the overall complexity of American option pricing is then only $O(N(M+N))$, and, therefore, for $M \sim N$, identical to the pricing of European options, which is $O(MN)$. We also discuss the numerical properties and robustness with respect to model parameters in relation to penalty and projected relaxation methods.

2.1 Introduction

An American option is a financial instrument that gives its buyer the right, but not the obligation, to buy (or sell) an asset at an agreed price at any time up to a certain time T. When working in a partial differential equation (PDE) framework, the option value $V = V(t, S)$, where $t \in [0, T]$ and $S \in \mathbb{R}^+$ denote time and value of the underlying stock, respectively, is usually (e.g. cf. [23, 61]) the solution of a linear complementarity problem (LCP)

$$\begin{aligned} \mathcal{L}V &\geq 0, \\ V &\geq P, \\ \text{and} \quad \mathcal{L}V \cdot (V - P) &= 0 \end{aligned}$$

with terminal condition $V(T, S) = P(S)$, where $\mathcal{L}$ is a linear (parabolic) differential operator and $P = P(S)$ denotes the payoff of the option; we note that the problem can equivalently be formulated as the HJB equation

$$\min\{\mathcal{L}V, V - P\} = 0, \tag{2.1.1}$$

which is a generalisation of the Black-Scholes case (1.1.2). If a fully implicit or weighted time-stepping scheme (as briefly introduced in (1.1.6)) is applied to the operator $\mathcal{L}$ in the above LCP, one usually has to solve a discrete LCP in the form of Problem 2.1.1 at every time step (cf. [23, 60, 67]). Here, $N \in \mathbb{N}$ denotes the length of the space grid. A simple example is given at the start of Section 2.3.

Problem 2.1.1. *Let $A \in \mathbb{R}^{N\times N}$ be an M-matrix, and let b, $c \in \mathbb{R}^N$ be vectors. Find $x \in \mathbb{R}^N$ such that*

$$\begin{aligned} Ax &\geq b, \\ x &\geq c, \\ \text{and} \quad (Ax-b)_i \cdot (x-c)_i &= 0, \quad 1 \leq i \leq N. \end{aligned}$$

In this context, for $z \in \mathbb{R}^N$ and $1 \leq i \leq N$, $(z)_i$ is used to denote the i-th element of the vector z. Additionally, throughout this chapter, for $Z \in \mathbb{R}^{N\times N}$ and $1 \leq i \leq N$, we will use $(Z)_i$ to denote the i-th row of the matrix Z. The definition of an M-matrix can be found in Definition 1.1.1 or [25]; in particular, an M-matrix Z is non-singular with $Z^{-1} \geq 0$, i.e. every element of Z^{-1} is non-negative. Matrices of this form arise naturally from most discretisation schemes for partial differential equations, and, as briefly outlined in Chapter 1, they are crucial when proving convergence of a finite difference scheme by viscosity solution techniques. For the existence and uniqueness of a solution to Problem 2.1.1, e.g., see [20] or [69].

For more general classes of matrices A, LCPs like the one in Problem 2.1.1 have been widely studied from the point of view of linear and quadratic programming (cf. [19]), but these often make no particular use of the special structure of A arising from the discretisation of a differential operator (cf. [2, 20]). This makes the projected successive over-relaxation method (PSOR) – introduced in [20] – the most widely used approach in practice for finite difference matrices (cf. [23, 60, 67]), in spite of its relatively slow convergence (cf. [38]); see also Section 1.2.3, where an outline of PSOR has been given.

We aim to demonstrate that the method of policy iteration, developed in [28] for the numerical solution of HJB equations, yields a powerful and beautifully simple method for the solution of Problem 2.1.1 in a financial context. Policy iteration is based on the interpretation of Problem 2.1.1 as the discrete HJB equation (1.2.1), or, equivalently and component-wise,

$$\min_{\phi\in\{0,1\}} \{\phi(Ax-b)_i + (1-\phi)(x-c)_i\} = 0, \tag{2.1.2}$$

where ϕ is a control parameter: 0 for exercise, 1 for continuation in state i. The availability of policy iteration as a direct algorithm for American option pricing does not seem to be widely known; after completing the results presented in this chapter, however, we were made aware of some results derived independently in [50] for

“Howard’s algorithm” (motivated by Markov decision processes, [36]), which also includes specific results on American option pricing.

We begin by an analysis in Section 2.2 which is similar to [50], for the convergence of the method in no more than N steps. We then extend the results of [50] by proving that this bound is sharp, and we illustrate this behaviour numerically in Section 2.3; in a slightly different setting, a numerical example where policy iteration requires N steps to convergence can also be found in [59]. We also include an example where A is not an M-matrix and policy iteration fails, highlighting the importance of monotonicity in addition to the requirement when proving convergence of a finite difference approximation in the viscosity sense (see Chapter 1). Implementationally, our method is equivalent to an iterative scheme for variational inequalities in [35] and the primal-dual active set strategy used in [43], a connection also observed in [50]. As we discuss, it can also be seen as the limit for infinite penalty parameter of an adaptation of the standard penalty method in [69] to this case. (The link between semi-smooth Newton methods and policy iteration has been noted before, e.g. in [50, 43], but viewing policy iteration as the limit of a Newton iteration of a penalty method adds a new perspective.) Interestingly, the formal limit of the standard method in [29] does not lead to a feasible policy iteration. The absence of any penalty parameter in the method presented here has some advantages, since it averts the question about the intensity of the penalisation. It is conceptually a simpler and arguably more intuitive approach.

We benchmark our approach against PSOR (cf. [20] and Section 1.2.1) and a penalty method (cf. [29] and 1.2.2), showing the competitiveness of the method in practice.

We point out that, for particular cases (e.g. cf. [21] for the case of tridiagonal M-matrices), there exist specific algorithms improving over PSOR; we restrict ourselves to a comparison with penalisation and PSOR since they are the most widely used and most versatile approaches.

The PSOR method is proven to converge for most matrices arising in option pricing applications – specifically, positive definite ones (cf. [2]) – but, as already stated above, the number of necessary iterations may grow substantially for decreasing mesh size, especially for a suboptimal relaxation parameters (cf. [38]).

The penalty method, combined with a Newton-type algorithm to solve the penalised equation as proposed in [29], requires only the solution of a linear system per iteration (in one dimension), where the number of iterations is usually very small in applications; this is, subject to limitations that we will discuss, inherited by the policy iteration presented here.

For completeness, we now briefly discuss the relation to various other methods – not, or only loosely based on the solution of the discrete LCP – which have been brought forward for pricing American options in a finite difference framework.

In the financial industry, a standard approach is to apply the early exercise right 'explicitly', which results in a decoupling of the two inequalities by first computing a continuation value, $\hat{x}$ say, i.e. the value of holding and hedging the option, and taking the maximum of that and the exercise value,

$$A\hat{x} = b, \quad x = \max(\hat{x}, c).$$

Except for fully explicit time stepping schemes, where A is the identity matrix, $\hat{x}$ is only an approximate solution to the discrete LCP at every individual time step. However, the approach is often remarkably accurate in practice for sufficiently small time steps since convergence of the full discretisation (i.e. time and space simultaneously) can be proved using results in [6].

One class of methods is based on a more sophisticated splitting of the operators, which approximate the continuous LCP by a sequence of simpler problems in every time step, rather than solving the discretised LCP directly, e.g. see [37, 38] and references therein; these methods are typically efficient if the underlying splitting for the European counterpart is accurate. However, with these approaches, there is no sense of solving a discrete LCP, which means that the convergence properties of the discretised LCP to the continuous one (e.g. see [24, 49] and references therein) cannot be utilised, but convergence has to be analysed afresh for the splitted scheme. To our knowledge, there is no comprehensive convergence analysis for these methods to date.

Another class of methods exploits explicit knowledge of the topology of the exercise and continuation regions, either by front-fixing (cf. [62]), method of lines coupled with Riccati transformations (cf. [53, 52]), or – on the discrete level – by partition of the index set of the linear program as in [12] and [14], to name just a few. The method proposed here is similar to [14] in the sense that it is a direct method for the solution of the LCP with a finite number of iterations, but, differing from [14], it is not dependent on any structure of the discrete exercise region.

Structure of this Chapter

In Section 2.2, we introduce policy iteration in the context of American option pricing, and we prove finite convergence in at most $N+1$ steps, where $N \times N$ is the size of the discretisation matrices; we also prove that this bound is sharp and that monotonicity is an essential requirement. In Section 2.3, we present detailed numerical results. Finally, in Section 2.4, we discuss the relation to semi-smooth Newton iterations for a penalty approximation.

2.2 The Method of Policy Iteration

In this section, we describe how to use policy iteration as an algorithm for solving Problem 2.1.1. We adapt algorithm and proof from [28] to the specific situation of the discrete LCP considered here. We use I_N to denote the identity matrix in $\mathbb{R}^{N\times N}$, and we consider

$$(\phi_i(Ax)_i + (1-\phi_i)(x)_i) - (\phi_i(b)_i + (1-\phi_i)(c)_i) = 0, \quad 1 \le i \le N,$$

where $\phi_i \in \arg\min_{\phi\in\{0,1\}} \{\phi(Ax-b)_i+(1-\phi)(x-c)_i\}$, $1 \le i \le N$, is *an* optimal policy in state i (but not necessarily unique); put differently, the solution x^* to Problem 2.1.1 solves

$$A^*x^* = b^*,$$

where $(A^*)_i = \phi_i(A)_i + (1-\phi_i)(I_N)_i$ and $(b^*)_i = \phi_i(b)_i + (1-\phi_i)(c)_i$, $1 \le i \le N$.

Algorithm 2.2.1. *Let $x^0 \in \mathbb{R}^N$. For x^n given, let $\phi^n \in \mathbb{R}^N$, $A^n \in \mathbb{R}^{N\times N}$ and $b^n \in \mathbb{R}^N$ be such that, for $1 \le i \le N$, we have*

$$\begin{aligned} (\phi^n)_i &\in \arg\min_{\phi\in\{0,1\}} \{\phi(Ax^n-b)_i + (1-\phi)(x^n-c)_i\}, \\ (A^n)_i &= (\phi^n)_i(A)_i + (1-(\phi^n)_i)(I_N)_i \\ \textit{and} \quad (b^n)_i &= (\phi^n)_i(b)_i + (1-(\phi^n)_i)(c)_i\,, \end{aligned}$$

from which follows

$$(A^nx^n - b^n)_i = \min_{\phi\in\{0,1\}} \{\phi(Ax^n-b)_i + (1-\phi)(x^n-c)_i\}. \tag{2.2.1}$$

Find $x^{n+1} \in \mathbb{R}^N$ such that

$$A^nx^{n+1} = b^n. \tag{2.2.2}$$

In essence, in each step of the iteration in Algorithm 2.2.1, we check pointwise which inequality is violated the most, and solve that one with equality. (The only exception to this is when both inequalities are non-negative, in which case we solve the smaller one with equality; however, as we will see in (2.2.3), this can only happen when computing x^1.)

2.2.1 Finite Termination and Linear Complexity

Theorem 2.2.2. *Let $(x^n)_{n=0}^\infty$ be the sequence generated by Algorithm 2.2.1. Then*

$$\begin{aligned} x^{n+1} &\ge x^n, \quad n\in\mathbb{N}, \\ \textit{and} \quad x^n &= x^*, \quad n \ge \kappa, \end{aligned}$$

where x^* is the solution to Problem 2.1.1 and $\kappa \in \mathbb{N}$ is a constant independent of x^0.

Proof. It can be found in [25] that, since A is an M-matrix by assumption, A^n as defined in (2.2.1) is an M-matrix; this means in particular that A^n is non-singular, and, hence, Algorithm 2.2.1 is well defined. Independent of $n \in \mathbb{N}$, there are only 2^N different compositions that can be assumed by A^n and b^n, as follows from (2.2.1), consequently there are only 2^N different possible values for x^n, $n \in \mathbb{N}$. From (2.2.1) and (2.2.2) follows

$$0 = A^{n-1}x^n - b^{n-1} \geq \min\{Ax^n - b, x^n - c\} = A^n x^n - b^n, \tag{2.2.3}$$

and therefore

$$\begin{aligned} A^n(x^{n+1} - x^n) &= (A^n x^{n+1} - b^n) - (A^n x^n - b^n) \\ &\geq (A^n x^{n+1} - b^n) - (A^{n-1} x^n - b^{n-1}) = 0. \end{aligned}$$

The M-matrix property of A_n, $(A^n)^{-1} \geq 0$, implies that $x^{n+1} \geq x^n$ for $n \in \mathbb{N}$. Having established that $(x^n)_{n=1}^{\infty}$ is a monotone sequence in a finite set, we may set $\kappa = 2^N$, and we can deduce the existence of a limit x^*. It follows directly from (2.2.1) and (2.2.2) that x^* solves Problem 2.1.1. □

In the following corollary, we will equip Algorithm 2.2.1 with a tie-breaker: if, in (2.2.1), we have a tie of the form

$$(Ax^n - b)_i = (x^n - c)_i = \min\big\{(Ax^n - b)_i\,, (x^n - c)_i\big\},$$

then we set $(\phi^n)_i = 1$,

$$(A^n)_i = (A)_i \quad \text{and} \quad (b^n)_i = (b)_i\,, \tag{2.2.4}$$

such that $(A^n x^n - b^n)_i = (Ax^n - b)_i$. Now, based on (2.2.4), the following corollary provides a much sharper bound for the maximum number of steps of Algorithm 2.2.1. In the next section, we will give an example which shows that, in common applications, the obtained order is indeed sharp.

Corollary 2.2.3. *If we use the tie-breaker* (2.2.4)*, then the result of Theorem 2.2.2 holds for $\kappa = N + 1$, i.e. Algorithm 2.2.1 has finite termination in at most $N + 1$ steps.*

Proof. From (2.2.3), for $1 \leq i \leq N$, $n \in \mathbb{N}$, we have that

$$\min\{(Ax^n - b)_i\,, (x^n - c)_i\} \leq 0. \tag{2.2.5}$$

From (2.2.4) and (2.2.5) and the monotonicity of $(x^n)_{n=1}^{\infty}$, for $1 \leq i \leq N$, we may deduce that if $(x^n)_i \geq (c)_i$ and $(A^n)_i = (A)_i$, then

$$(Ax^{n+1} - b)_i = 0 \quad \text{and} \quad (x^{n+1})_i \geq (x^n)_i \geq (c)_i\,,$$

which implies that $(A^{n+1})_i = (A)_i$; iterating the argument, we obtain that

$$\text{if } (x^n)_i \geq (c)_i \ \wedge \ (A^n)_i = (A)_i \text{ for } n \in \mathbb{N}, \text{ then } (A^m)_i = (A)_i \ \forall\, m \geq n. \tag{2.2.6}$$

Additionally, we know that

$$(x^n)_i \geq (c)_i \quad \text{for } 1 \leq i \leq N, \ n \geq 2, \tag{2.2.7}$$

since $(x^2)_i \geq (x^1)_i$, and, if $(x^1)_i < (c)_i$, then we have $(Ax^1 - b)_i = 0$, which implies $(x^2)_i = (c)_i$. However, we also note that $A^{n+1} = A^n$ for $n \in \mathbb{N} \cup \{0\}$ implies $x^{n+1} = x^*$. Altogether, we may say that

- as long as Algorithm 2.2.1 has not yet converged, the linear system solved in (2.2.2) must change in every step,
- and, based on (2.2.6) and (2.2.7), from x^2 onwards, every row of the linear system in (2.2.2) can change at most once before convergence,

which means that Algorithm 2.2.1 must converge in no more than $N+2$ steps. But the maximum number of $N+2$ steps can only be reached if $A^2 = I_N$ and $A^* = A$, where $(A^*x^* - b^*) = \min\left\{(Ax^* - b), (x^* - c)\right\}$, which is a contradiction because we would have had $x^1 = x^*$. We may conclude that $\kappa = N+1$. $\square$

We point out that, in practice, we find the performance of Algorithm 2.2.1 to be indifferent to the use of a tie-breaker as introduced in (2.2.4), which is not surprising since – numerically – we would expect a tie to be rare.

2.2.2 Example: American Put and Other Standard Payoffs

It follows from the results in Section 2.2.1 (see also [50]) that policy iteration converges in at most $N+1$ steps, where N is the problem size. We will now show that in the valuation of the standard American put, the number of iterations is indeed proportional to N normally. Specifically, this behaviour results from the fact that for certain linear segments of the obstacle, as are a common feature of option payoffs, policy iteration moves the discrete exercise boundary only by at most one node per iteration. We demonstrate this on the example of the put payoff, but the result generalises to more general piecewise linear payoffs.

Proposition 2.2.4. *Let the tridiagonal matrix A result from the finite difference discretisation of the Black-Scholes operator, which is only assumed to be of at least first order consistent. Let $b_i = c_i = P(S_i) = \max(K - S_i, 0)$ be the put payoff with strike K. Let $q = \max\{i : S_i \le K\}$ and, for the exact solution x^* to Problem 2.1.1, let $e = \min\{i : (Ax^* - b)_i = 0\} \le q$ (a discrete exercise boundary). If $x^0 = c$, then Algorithm 2.2.1 terminates in no less than $q - e$ steps.*

Proof. The proof can be found in [68]. For completeness, it is also included in Appendix A. $\square$

The proof of Proposition 2.2.4 demonstrates that, for a put payoff, policy iteration reduces to the following simple method: nodes are picked systematically as candidates for the discrete exercise boundary, and, if the value function with this exercise strategy solves the linear complementarity problem, the solution is found; if not, the next candidate node is tested. This procedure is similar in spirit to the approaches in [12] and [14]. The advantage of policy iteration, however, is that it does not require any knowledge of the topology of exercise and continuation regions, but these arise as a by-product.

Remark 2.2.5. *In Proposition 2.2.4, the number of nodes $q - e$ between the free boundary and the strike will be approximately $N(K - S_e)/S_{max} = O(N)$, if S_e is the exercise boundary and S_{max} the largest node on a uniform grid with N nodes.*

In the present setting, where the discretisation matrix A can be expected to be tridiagonal, the linear system in each step of Algorithm 2.2.1 can be solved by the Thomas algorithm (cf. [4]) in computational complexity $O(N)$. Since, by Corollary 2.2.3, Algorithm 2.2.1 converges in at most N steps, the overall complexity for the solution of Problem 2.1.1 is then at most $O(N^2)$. If, additionally, Problem 2.1.1 results from a time discretisation with M time steps, the overall complexity for the American option pricing algorithm will be at most $O(MN^2)$. Interestingly, it can be shown that, for a relevant class of (monotone) discretisations, the *overall* number of policy iterations, i.e. over all time steps, is bounded by $M + N$, such that the overall complexity of the method is $O(N(N + M))$.

Corollary 2.2.6. *Consider a monotone, at least first order consistent scheme for the Black-Scholes PDE (e.g. implicit Euler, or Crank-Nicolson under a time step constraint, and central differences on a sufficiently fine mesh). Assume that, in each time step, the solution from the previous time step is used as initial value for the policy iteration. Then the total number of policy iterations over all time steps $1, \dots, M$ is exactly $q-e+(M-m)$, where e is the discrete exercise boundary for the American put at time step M, and m the number of time steps in which the discrete free boundary has moved at least one grid point.*

Proof. The proof can be found in [68]. For completeness, it is also included in Appendix A. □

Although we clearly have implicit schemes in mind, the above results also hold for explicit schemes (where we use explicit time-stepping and also evaluate the optimal control explicitly), for which the free boundary can only move one grid cell per time step by construction.

2.2.3 The Role of Monotonicity and the M-Matrix Structure

The proof of Theorem 2.2.2 crucially requires A to be an M-matrix. For general matrices A, if Algorithm 2.2.1 is well defined and converges, it follows from (2.2.2) that the limit still solves Problem 2.1.1. However, generally, i.e. without monotonicity, we obtain no more than subsequence convergence, and in that case we may not find a solution. We will now construct such an example.

It is clear that finite difference matrices resulting from a central difference discretisation of non-degenerate one-dimensional linear elliptic equations are M-matrices, as long as the grid size is sufficiently small; the discretisation matrices, in every time step, of standard implicit schemes for the corresponding linear parabolic problem, e.g. the θ-scheme, are then also M-matrices. The M-matrix property can thus only fail for large drifts and coarse meshes. For example, suppose

$$-\mathcal{L}V = \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + \mu(S) S \frac{\partial V}{\partial S} - rV, \tag{2.2.8}$$

where $\sigma > 0$ and μ are to be determined later. If, for the sake of this example, we use a four point equidistant space grid consisting of S_0, S_1, S_2, S_3, where $S_0 < S_1 < S_2 < S_3$, and we have known terminal data at $t = T > 0$ and Dirichlet boundary conditions $V(S_0, t) = V(S_3, t) = 0$, $t \geq 0$, at S_0 and S_3, then discretising $\mathcal{L}V$ fully implicitly in time (with step size k) and with central differences in space (with step size h) results in a discretisation matrix

$$\begin{pmatrix} 1 + \frac{k}{h^2}\sigma^2 S_1^2 + r & -\frac{k}{2h^2}\sigma^2 S_1^2 - \frac{k}{2h}\mu(S_1)S_1 \\ -\frac{k}{2h^2}\sigma^2 S_2^2 + \frac{k}{2h}\mu(S_2)S_2 & 1 + \frac{k}{h^2}\sigma^2 S_2^2 + r \end{pmatrix}. \tag{2.2.9}$$

If we pick μ sufficiently large and such that $\mu(S_1) < 0$ and $\mu(S_2) > 0$, the off-diagonals will be positive and the M-matrix property will fail. It is easy to check that, for an appropriate combination of parameters k, h, σ, μ, S_1, S_2, r and payoff vector c, Problem 2.1.1 may be given by

$$A = \begin{pmatrix} 6 & 8 \\ 16 & 8 \end{pmatrix}, \; b = \begin{pmatrix} 58 \\ 64 \end{pmatrix} \quad \text{and} \quad c = \begin{pmatrix} 1 \\ 5 \end{pmatrix}.$$

Since there are only four candidate solutions, $w = (0.6, 6.8)^T$, $x = (1, 5)^T$, $y = (1, 6)^T$ and $z = (3, 5)^T$, where $Aw = b$, $x = c$,

$$\begin{pmatrix} 1 & 0 \\ 16 & 8 \end{pmatrix} y = \begin{pmatrix} 1 \\ 64 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 6 & 8 \\ 0 & 1 \end{pmatrix} z = \begin{pmatrix} 58 \\ 5 \end{pmatrix},$$

one verifies that z is the unique solution of $f(z) := \min\{Az - b,\, z - c\} = (0, 0)^T$:

$$f(w) = \begin{pmatrix} -0.4 \\ 0 \end{pmatrix}, \quad f(x) = \begin{pmatrix} -6 \\ -1 \end{pmatrix}, \quad f(y) = \begin{pmatrix} -2 \\ 0 \end{pmatrix}, \quad \text{and} \quad f(z) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Using w as initial guess, Algorithm 2.2.1 alternates between w and y, never finding the correct value z.

The chosen 'mean-repelling' drift μ is clearly somewhat unconventional; fortunately, for a constant drift as in the Black-Scholes model or mean-reverting drift μ as in the pricing of currency options (cf. [71]), say, or for any (more realistic) discretisation in which h is sufficiently small, the discretisation matrix (2.2.9) will still turn out to be an M-matrix, and, thus, our theory applies once again.

In more than one dimension, the M-matrix property is typically lost for discretisations of cross-derivatives, independent of the mesh size. In such cases, convergence cannot be guaranteed by the present theory; however, experimentally, we have found this not to be a problem for fine enough meshes.

2.3 Performance for Numerical Examples

In this section, we compare the numerical performance of Algorithm 2.2.1 with two other approaches, namely PSOR and penalisation (see [20, 29] and Sections 1.2.1 and 1.2.2, respectively).

We price an American put with strike K in a standard Black-Scholes framework (cf. [23]), i.e. we have

$$-\mathcal{L}V = \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV \tag{2.3.1}$$

and $P(S) = \max\{K - S, 0\}$, and we restrict the allowed asset range to $S \in [0, S_{max}]$, where $S_{max} > K$ is taken to be large. We discretise (2.3.1) following a standard textbook approach (e.g. cf. [60]), using a one-sided difference in time and central differences in space, working with M time steps and N space steps, and solving backwards in time with a fully implicit scheme. This means that, for every time step, we have to solve a discrete LCP as given in Problem 2.1.1, where A represents the finite difference discretisation matrix, b represents the solution vector known from the previous time step, and c represents the payoff vector. The exact parameters used in our computations can be found in Table 2.1. Regardless of which algorithm we use for the solution of Problem 2.1.1, we only terminate the algorithm if we have found a vector $x \in \mathbb{R}^N$ satisfying

$$\begin{aligned} \frac{Ax-b}{\|b\|_\infty} &\geq -tol, \\ \frac{x-c}{\|c\|_\infty} &\geq -tol, \\ \text{and} \quad \left[\frac{|(Ax-b)_i|}{\|b\|_\infty} \leq tol \quad \vee \quad \frac{|(x-c)_i|}{\|c\|_\infty} \right. &\left. \leq tol \right], \quad 1 \leq i \leq N, \end{aligned}$$

for some given tolerance $tol > 0$. We implement PSOR exactly as described in Section 1.2.1, testing all over-relaxation parameters $w_R \in \{1, 1.025, 1.05, \ldots, 1.875, 1.9\}$, and we use a non-linear penalty iteration with penalty parameter ρ as introduced in Section 1.2.2.

Remark 2.3.1. *The convergence of the proposed finite difference scheme for an American option in a standard Black-Scholes model as considered here can be justified in a number of ways. First, it can be shown that similar discretisation matrices also result from a finite element discretisation, convergence of which can be proved in the framework of Sobolev spaces (cf. [24]). Alternatively, convergence can be proved in a viscosity setting, as outlined in Chapter 1; we explain this in more detail in Section 3.5.1, and the current example is a special case of Theorem 3.5.8.*

r	σ	T	K	S_{max}	tol	ρ'
0.05	0.4	1	100	600	1e-08	1e06

Table 2.1: *The parameters used for the numerical computations. We use penalty parameter $\rho = \rho'/k$, where $k = T/M$ is the size of the time step, to correct for the scaling implicitly resulting from the time discretisation (cf. [32]).*

2.3.1 Dependence on Grid Size and Time Step

The results of our computations for the different methods are summarised in Table 2.2. For different grid sizes, the table shows the maximum number and the average number of iterations needed to solve the discrete LCP to an accuracy of *tol* at one time step, and it also includes the overall runtime needed to price the put when solving the discrete LCP for all time steps. For PSOR, w_R^* denotes the over-relaxation parameter that performed best for that particular grid size. (The fact that w_R^* changes with the grid size and is not known in advance is certainly a drawback of the PSOR method, as we have already stated in Section 1.2.1)

PSOR	Max Iterations	∅ Iterations	Runtime	w_R^*
$M, N = 200$	3	2.17	0.23s	1.050
$M, N = 400$	5	2.67	1.07s	1.075
$M, N = 800$	5	3.15	4.88s	1.175
$M = 50,\ N = 800$	28	17.00	1.61s	1.650
$M = 800,\ N = 50$	1	1.00	0.14s	1.000
Penalty Method	Max Iterations	∅ Iterations	Runtime	-
$M, N = 200$	2	1.06	0.04s	-
$M, N = 400$	2	1.06	0.08s	-
$M, N = 800$	3	1.07	0.28s	-
$M = 50,\ N = 800$	5	1.82	0.03s	-
$M = 800,\ N = 50$	2	1.00	0.07s	-
Policy Iteration	Max Iterations	∅ Iterations	Runtime	-
$M, N = 200$	3	1.07	0.03s	-
$M, N = 400$	4	1.06	0.09s	-
$M, N = 800$	6	1.07	0.29s	-
$M = 50,\ N = 800$	18	2.08	0.04s	-
$M = 800,\ N = 50$	2	1.00	0.07s	-

Table 2.2: *Our computational results for different numbers of time and space steps, denoted by M and N, respectively. We see that, for all chosen grid sizes, policy iteration and penalisation perform virtually identically and clearly outperform PSOR.*

We see that, for all considered grid sizes, policy and penalised iteration have very similar numerical performances and clearly improve over PSOR. Since, for the three considered algorithms, we always solve the discrete LCP at every time step to the same accuracy *tol*, the numerical results are equivalent to this accuracy, and the computational runtime is the only criterion that is left to be considered; hence, policy iteration or penalisation should be the methods of choice, with policy iteration having the (mostly conceptual) advantage of being an exact solver of the discrete LCP rather than an approximation.

Policy Iteration	$N = 100$	$N = 200$	$N = 400$	$N = 800$
$M = 100$	2/1.05/0.01s	4/1.13/0.03s	7/1.26/0.03s	14/1.54/0.05s
$M = 200$	2/1.03/0.02s	3/1.07/0.03s	5/1.13/0.05s	11/1.27/0.08s
$M = 400$	2/1.01/0.04s	2/1.03/0.06s	4/1.06/0.09s	8/1.14/0.15s
$M = 800$	2/1.01/0.06s	2/1.02/0.11s	3/1.03/0.16s	6/1.07/0.29s

Table 2.3: *Our computational results for different numbers of time and space steps, denoted by M and N, respectively. The three numbers in every cell represent 'Max Iterations', '∅ Iterations' and 'Runtime'.*

Table 2.3 shows that the maximum number of iterations increases linearly with N if M is kept fixed, as predicted by the theory; it increases with $\sqrt{N}$, if $N = M$, which we will explain in the next subsection.

In the light of the results in Section 2.2.2, we now consider in more detail the number of policy iterations needed to solve a single discrete LCP. To this end, we set $M = 1$ and consider different grid sizes N, using c as initial guess. In Figure 2.1, we see that the number of policy iterations required is clearly linear in N.

Since, in Figure 2.1, we used one time step only, the starting value of the iteration can be expected to merely be a rather coarse approximation of the solution. However, the results of Table 2.2 suggest that the average number of iterations is almost constant if there is a good initial guess, as is the case when using a sufficiently fine time grid where the active set changes only for a few grid points around the exercise boundary; more precisely, based on Table 2.2, the observed complexity of American option pricing by penalisation and policy iteration is $O(MN)$, using M time steps and requiring $O(N)$ for solving a tridiagonal system.

In Table 2.4, we see the average number of policy iterations required when setting $M = N$ and running a fully implicit and a Crank-Nicolson scheme. Again, in both cases, the average number of iterations is clearly bounded (and very small), here suggesting that the discrete free boundary only moves a small number of grid cells per time step. The American option pricing algorithm thus shows complexity $O(N^2)$ in practice.

As a Crank-Nicolson central difference scheme is hoped to converge with second order in time and space (at least with a time step selector as in [29]), $M = N$ is usually the

Policy Iteration, $M = N$	100	200	300	400	500	600	700	800
Fully Implicit, ∅ Iterations	1.05	1.07	1.07	1.07	1.07	1.07	1.07	1.07
C.N. , ∅ Iterations	1.02	1.03	1.05	1.04	1.03	1.03	1.03	1.04

Table 2.4: *The average number of policy iterations required per time step when setting $M = N$ appears to be constant, for fully implicit as well as Crank-Nicolson (C.N.) time stepping. Hence, in practice, we can price American options in $O(N^2)$, which is the same complexity as for European options.*

natural choice. Altogether, the observed complexity of an American option pricing algorithm based on policy iteration (or penalisation) is seen to be the same as for a European pricing code.

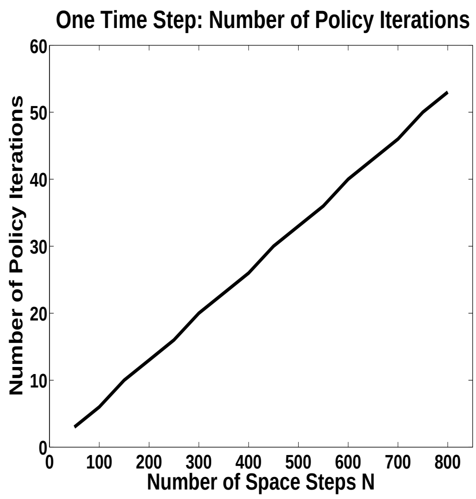

Figure 2.1: *We fix $M = 1$, solving exactly one discrete LCP, and we consider different grid sizes N. In this setup, the starting value is then always a discretisation of the payoff function on a space grid of N points. We see that the number of policy iterations required for solving the discrete LCP is clearly linear in the matrix dimension N.*

2.3.2 Dependence on Volatility

It is known from [58] that, close to expiry, $\sigma^2(T-t) \ll 1$, the exercise boundary S_e behaves like

$$\frac{S_e(t) - K}{K} \sim -\sigma\sqrt{-(T-t)\log(T-t)}.$$

As the number of policy iterations is proportional (cf. Remark 2.2.5) to the displacement of the free boundary from the initial guess (typically the solution from the previous time step), we expect that the maximum number of iterations to be attained at the first time step, where the free boundary moves fastest, and to be proportional to σ (cf. Table 2.5). If k/h, and therefore M/N, is held constant, one further expects that the maximum number of iterations increases with $\sqrt{N}$, as can be observed in Table 2.3. For large enough T, the average number of iterations for all time steps between $t = T$ and $t = 0$ will depend weakly on σ because of the upper bound N on the total number of iterations over all time steps (cf. Proposition 2.2.4 and Table 2.5).

$M = 1$, $N = 200$	$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.8$
Policy Iteration	6/6	13/13	23/23
Penalty Method	4/4	6/6	7/7
PSOR	39/39	302/302	990/990
$M = 200$, $N = 200$	$\sigma = 0.2$	$\sigma = 0.4$	$\sigma = 0.8$
Policy Iteration	2/1.03	3/1.07	6/1.11
Penalty Method	2/1.03	2/1.06	2/1.09
PSOR	2/1.99	4/2.18	9/2.92

Table 2.5: *Our computational results for the performance in dependence on σ. The two numbers in every cell represent 'Max Iterations' and '∅ Iterations'. For $M = 1$, policy iteration has a clear linear dependence on σ, penalisation is almost unaffected by the change in σ, and PSOR seems very sensitive to σ. For $M = N$, all three schemes show only a mild dependence on σ.*

2.4 The Relation of Policy Iteration to Penalisation

Here, we discuss the relation of penalty and policy iterations for American options, and how they can be combined to exploit the advantages of both.

2.4.1 Basic Properties

The canonical and most common penalty approximation for American options, briefly outlined in Section 1.2.2, used e.g. in [29] and in the numerical examples of Section 2.3, is obtained by replacing Problem 2.1.1 by

$$Ax_\rho - b - \rho \max\{c - x_\rho\,, 0\} = 0, \tag{2.4.1}$$

for $\rho > 0$ large. A semi-smooth Newton iteration for (2.4.1) can be defined (see [29]) as per Alorithm 2.2.1, but with

$$\begin{aligned} (\phi_\rho^n)_i &\in \operatorname*{arg\,min}_{\phi \in \{0,1\}} \big\{(1-\phi)(x_\rho^n - c)_i\big\}, && (2.4.2)\\ (A_\rho^n)_i &= (A)_i + \rho\,(1 - (\phi_\rho^n)_i)(I_N)_i && (2.4.3)\\ \text{and} \quad (b_\rho^n)_i &= (b)_i + \rho\,(1 - (\phi_\rho^n)_i)(c)_i\,. && (2.4.4) \end{aligned}$$

The number of iterations needed to solve (2.4.1) can be shown to be bounded by N, as for the policy iteration algorithm, and this convergence is also monotone under the same assumptions (see [29]). This, and most of the numerical results in Section 2.3.1, suggest similar properties of the two methods, although there is a visible difference in the iteration numbers for large N. Indeed, the iterates generated by the two methods are distinctly different, even for large penalty parameter, which is clear by inspection of the policies ϕ_ρ^n and ϕ^n.

One may be tempted to formally set $\rho = \infty$ in the above equations and define a different policy iteration by Algorithm 2.2.1, but with the policy ϕ replaced by

$$(\widehat{\phi}^n)_i \in \operatorname*{arg\,min}_{\phi \in \{0,1\}} \big\{(1-\phi)(x^n - c)_i\big\}. \tag{2.4.5}$$

We will now show why this does not result in a convergent algorithm, essentially because the policy does not take into account the PDE constraint. First, we have to define a tie-breaker for the case $(x^n)_i = (c)_i$, when (2.4.5) does not uniquely define $(\widehat{\phi}^n)_i$. For the choice $(\widehat{\phi}^n)_i = 0$, the iteration may get 'stuck', i.e. $(x^m)_i = (c)_i$ for all $m \geq n$, and will not find any solution with $(x^*)_i > (c)_i$. For the choice $(\widehat{\phi}^n)_i = 1$, however, one observes that the solution x^* to Problem 2.1.1 is generally not a fixed point of this policy iteration: if $x^0 = x^*$ and $(Ax^* - b)_i > 0$ for some i, then $x_i^0 = x_i^* = c_i$, and in the next iteration $(Ax^1 - b)_i = 0$.

We will show in the following section that a slightly modified penalty scheme has policy iteration as its formal limit.

It remains to investigate why the number of Newton steps for the penalised system does not grow in the same way as for policy iteration when the grid is refined, although the policy algorithm can be viewed as a Newton method for the LCP (see [50]). Here, it is helpful to take the viewpoint of [41], who analyse a semi-smooth Newton method for the underlying continuous variational inequality in a suitable function space and show superlinear convergence. The penalty iteration above can be seen as discretisation of the iteration in [41], and therefore has a well-defined limit for vanishing grid size. This is not true for the policy iteration, which is in some sense inherently discrete. For a discussion of the regularity of the iterates for the unpenalised (continuous) variational inequality, we refer to [41], who also remark on the propagation of discrete active sets, related to the discussion in Section 2.2.2.

2.4.2 Policy Iteration as a Limit of Penalisation

We briefly show how policy iteration can be seen as limit of a different penalised iteration.

Problem 2.4.1. *Let A, b, and c be as in Problem 2.4.1. Let $\rho > 0$ be large. Find $x_\rho \in \mathbb{R}^N$ such that*

$$Ax_\rho - b + \rho \min\{Ax_\rho - b, x_\rho - c\} = 0. \tag{2.4.6}$$

Algorithm 2.4.2. *Let $x^0 \in \mathbb{R}^N$. For x_ρ^n given, use A^n and b^n as introduced in Algorithm 2.2.1, and find $x_\rho^{n+1} \in \mathbb{R}^N$ such that*

$$(A + \rho A^n)x_\rho^{n+1} = b + \rho b^n. \tag{2.4.7}$$

Theorem 2.4.3. *Problem 2.4.1 has a unique solution x_ρ^* satisfying*

$$\|x_\rho^* - x^*\|_\infty \leq \frac{C}{\rho}$$

for a constant $C > 0$ independent of ρ, where x^* is the solution to Problem 2.1.1. Furthermore, denoting by $(x_\rho^n)_{n=0}^\infty$ the sequence generated by Algorithm 2.4.2, it is

$$\begin{aligned} x_\rho^{n+1} &\geq x_\rho^n\,, \quad n \in \mathbb{N}, \\ \text{and} \quad x_\rho^n &= x_\rho^*\,, \quad n \geq \kappa, \end{aligned}$$

where κ is the same as in Theorem 2.2.2.

Proof. The results can easily be shown by using the same techniques as in [69], where more complex penalty approximations are dealt with. □

Now, we have seen that Problem 2.4.1 is a viable penalty approximation of Problem 2.1.1 and that Algorithm 2.4.2 has properties very similar to those of Algorithm 2.2.1. In addition, equation (2.4.7) bears strong resemblance to equation (2.2.1), especially if we expect the terms multiplied by ρ to be dominant.

Lemma 2.4.4. *The sequence $(x_\rho^n)_{n=0}^\infty$ generated by Algorithm 2.4.2 is uniformly bounded in ρ, i.e.*

$$\|x_\rho^n\|_\infty \leq C, \quad n \in \mathbb{N},$$

for a constant $C > 0$ independent of ρ.

Proof. For $n \in \mathbb{N}$ and $1 \leq i \leq N$, it is

$$(Ax_\rho^{n+1} - b)_i = 0 \tag{2.4.8}$$

or

$$(Ax_\rho^{n+1} + \rho x_\rho^{n+1} - b - \rho c)_i = 0. \tag{2.4.9}$$

For equation (2.4.9), we consider two cases. First, if $(Ax_\rho^{n+1} - b)_i > (x_\rho^{n+1} - c)_i$, then we have

$$(x_\rho^{n+1} - c)_i \leq 0 \leq (Ax_\rho^{n+1} - b)_i\,. \tag{2.4.10}$$

Second, if $(Ax_\rho^{n+1} - b)_i \leq (x_\rho^{n+1} - c)_i$, then it is

$$(Ax_\rho^{n+1} - b)_i \leq 0 \leq (x_\rho^{n+1} - c)_i\,. \tag{2.4.11}$$

Now, there is no ρ in equations (2.4.8), (2.4.10), and (2.4.11), and we can deduce the existence of M-matrices A^* and A^{**} such that $\min\{b, c\} \leq A^*x_\rho^{n+1}$ and $A^{**}x_\rho^{n+1} \leq \max\{b, c\}$, where all rows of A^* and A^{**} are either taken from A or I_N; since there are only finitely many compositions that can be assumed by either of A^* and A^{**}, we may conclude the proof. □

Based on Lemma 2.4.4, we can now show that, given identical starting values, one step of policy iteration does in fact correspond to the limit $\rho \to \infty$ of one step of penalty iteration.

Lemma 2.4.5. *Consider the sequences $(x^n)_{n=0}^{\infty}$ and $(x_\rho^n)_{n=0}^{\infty}$ generated by Algorithms 2.2.1 and 2.4.2, respectively. If $x^n = x_\rho^n$ for some $n \in \mathbb{N}$, then*

$$\|x^{n+1} - x_\rho^{n+1}\|_\infty \leq \frac{C}{\rho},$$

where $C > 0$ is a constant independent of n and ρ.

Proof. We have

$$(A + \rho A^n)x_\rho^{n+1} = b + \rho b^n \quad \text{and} \quad A^n x^{n+1} = b^n,$$

which implies

$$\frac{1}{\rho} A x_\rho^{n+1} + A^n(x_\rho^{n+1} - x^{n+1}) = \frac{1}{\rho} b,$$

and we get the desired result by using the uniform boundedness from Lemma 2.4.4. □

No matter how large ρ is, Lemma 2.4.5 cannot be applied iteratively to compare the whole sequences generated by the two schemes since it assumes identical starting values. However, we can make the following instructive remark.

Remark 2.4.6. *Loosely speaking, the estimate of Lemma 2.4.5 can be interpreted to relate Algorithms 2.2.1 and 2.4.2 in the following sense: if both algorithms use the same starting value x^0 and we set $\rho = \infty$, then the generated sequences are identical.*

2.4.3 A Hybrid Method

Policy iteration has the conceptual advantage that it solves the discrete LCP exactly, whereas, with penalisation, there is an additional (small) penalisation error that has to be controlled. The Newton iteration for the penalised problem, conversely, has the advantage that it converges faster in practice, and the speed-up can be substantial if N is large. This is an effect of the mechanism by which policy iteration detects the free boundary by a pointwise search. This suggests a hybrid approach where we use a semi-smooth Newton method to solve the penalised equation and then use this solution as initial value for policy iteration. The following proposition shows that the total number of linear equation solves in this combined algorithm is only by one larger than the Newton steps, with the advantage that the LCP is then solved exactly.

Proposition 2.4.7. *If x_ρ is the solution of the penalised equation (2.4.1), then, for sufficiently large ρ, policy iteration with initial value x_ρ converges in a single step.*

Proof. The proof can be found in [68]. For completeness, it is also included in Appendix A. □

2.5 Conclusion

In this chapter, we have shown that the method of policy iteration, devised in [28] for the solution of discretised HJB equations, is a natural fit to American option pricing. It is extremely simple in structure, and finite convergence in at most $N+1$ steps can easily be proved. Numerical results show that, in practically relevant situations, it performs identically to a penalty scheme and improves over PSOR.

The simplicity advantage of the proposed method is especially noticeable in one dimension, where the algorithm is a small modification of a tridiagonal linear solver; in this case, the overall complexity for solving the linear complementarity problem is $O(N^2)$ and, if used as part of an instationary American option solver, $O(N(N+M))$ overall. In higher dimensions, the proposed method is still a direct method in the sense that the basic iteration has finite termination, but the linear systems required by the algorithm might most suitably be solved by an iterative solver.

Finally, we discuss how our approach can be interpreted as the formal limit of a Newton-type iteration applied to a penalised equation, the advantage of the policy iteration being that, in the limit, the penalisation error vanishes. However, it is also noted that policy iteration is inherently finite-dimensional, and the number of iterations increases for refined meshes, whereas, for semi-smooth Newton methods for a zero-order penalty, term this is not the case, as there is an underlying continuous iteration which is approximated. A consequence of these two points combined is that although two penalty approximations to the same discrete HJB equation may have theoretically identical properties for fixed finite dimensions, the infinite dimensional limit is a helpful orientation in that it provides robustness of the method as the grid is refined, a point related to that made in [41].

Chapter 3

Hamilton-Jacobi-Bellman (HJB) Equations: Finite Control Sets

In the previous chapter, we have used the method of policy iteration for the solution of the discrete non-linear Problem 2.1.1, equivalent to (1.2.1) and (2.1.2), which results from the finite difference discretisation of continuous problems like (1.1.2) or (2.1.1). We have then, amongst others, compared the policy iteration approach to a Newton-like method applied to the penalty approximation described in Section 1.2.2.

In this chapter, we generalise the setting from the previous chapter, and we consider HJB equations of the form (1.1.3), for finite control sets $\mathbf{U}$. Discretising by finite differences as briefly explained in Chapter 1, we then face problems of the form (1.1.7). Here, we introduce a penalty method which approximates these non-linear discrete systems to an order of $O(1/\rho)$, where $\rho > 0$ is the penalty parameter, and we show that an iterative scheme can be used to solve the penalised discrete problem in finitely many steps. We include a number of examples from mathematical finance for which the described approach yields a rigorous numerical scheme and present numerical results. Our penalty approach can be seen as an extension to HJB equations of the method presented in [29] for American options (cf. Section 1.2.2).

3.1 Introduction

In their most general form, HJB equations like (1.1.3) arise when applying Bellman's principle of optimality to problems of stochastic optimal control, in which case the solution to the HJB equation can often be found to solve the control problem (cf. [72, 56]). There is a wide range of practical applications to optimal control, amongst others in mathematical finance, and many examples are listed in [26, 46]. Naturally, the high practical relevance gives rise to the question of how to best solve stochastic control problems numerically.

Early (and still relevant) methods rely on Markov chain approximation, of which an extensive overview can be found in [45, 46, 26].

The notion of viscosity solution, introduced in [40], together with the most helpful results of [6], permits another line of attack, namely the direct solution of the corresponding HJB equations by discretisation methods; for details, see [6, 8, 7] and references therein; in [6], it was – roughly speaking – shown that every stable and consistent discretisation converges to the financially relevant solution; a brief conceptual outline of these arguments has also been given in Chapter 1.

Even though both approaches can be found to be closely related (e.g. cf. [63]), in particular for explicit time stepping methods, there is a rather clear conceptual difference: Markov chain approximations make explicit use of the underlying stochastic model and solve for transition densities, while finite difference methods deal solely with the HJB equation and solve for the unique viscosity solution directly (cf. [40]).

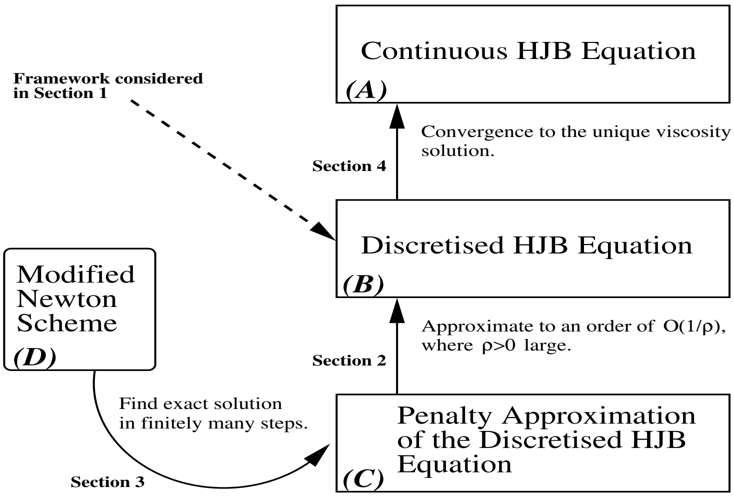

Figure 3.1: *To numerically approximate the HJB equation in (A), we consider a discrete version (B). Given some conditions that have to be checked for every particular example, the grid convergence of (B) to (A) can be guaranteed. We then approximate (B) by the penalty problem (C), which in turn can be solved exactly using algorithm (D).*

In this chapter, we are concerned with the numerical solution of a certain kind of HJB equation with a finite control set, namely equation (1.1.3) for $|\mathbf{U}| < \infty$, which, when applying a partially or fully implicit finite difference discretisation, becomes (1.1.7). Existing algorithms are, in one way or another, all related to policy iteration techniques, which have been used in the setting of Markov chain approximation [46, 59] as well as for HJB equations [28, 30, 66]. In this chapter, we introduce a novel approach by which the non-linear discrete system can be solved using a penalisation technique. More precisely, we show how a penalty approach that was devised for American option problems in [29] can be extended to solve discretised HJB equations provided the discretisation matrices obey a certain structure. (Penalty methods in general have a long history and have, for example, already been used extensively in the setting of variational inequalities in [9].) Examples from mathematical finance that fit into our framework include uncertain volatility models [48], transaction cost models [47], and unequal borrowing/lending rates and stock borrowing fees [3, 10]; a succinct overview of these and similar models can be found in [28, 30].

Structure of this Chapter

We aim to devise a penalty-based numerical scheme for the solution of a rather general class of HJB equations; the broad concept is depicted in Figure 3.1; here, using the overview from Chapter 1, *(A)* corresponds to (1.1.3), *(B)* corresponds to (1.1.7), and *(C)* corresponds to (1.3.1).

Section 3.2 We define the considered class of HJB equations (denoted by *(A)* in Fig. 3.1) and – having fully implicit and weighted time stepping schemes in mind – a fairly general non-linear discrete problem (denoted by *(B)* in Fig. 3.1). The structure of *(B)* is chosen such that we have an unconditionally stable numerical scheme which converges to *(A)* if the latter has a unique viscosity solution and satisfies a strong comparison principle (cf. Chapter 1).

Section 3.3 The non-linear discrete problem *(B)* is non-trivial to solve and will be our main consideration. In a first step, we introduce a penalised problem (denoted by *(C)* in Fig. 3.1) which approximates *(B)* to an order of $O(1/\rho)$, where $\rho > 0$ is the penalty parameter. It is worth pointing out that as the discretisation in *(B)* is stable (due to the monotonicity of the discretisation matrices), any errors introduced by the penalisation will stay within bounds and, hence, will not compromise the overall numerical scheme. Since *(C)* is a non-linear discrete problem itself, the feasibility of the penalty approach crucially depends on the possibility of conveniently solving *(C)* in a second step; in Section 3.4, we present an algorithm doing exactly that.

Section 3.4 We present an iterative procedure (denoted by *(D)* in Fig. 3.1) with finite termination which finds the exact solution to *(C)*. Even though the theoretical bound on the number of iterations depends on the the number of grid points and the cardinality of the finite control, the algorithm always converges – independently of the grid size and the penalty parameter – in less than four steps for our numerical examples.

Section 3.5 We give a detailed outline of how various examples from mathematical finance fit into our framework. Furthermore, for the HJB equation arising in European option pricing with different borrowing/lending rates and stock borrowing fees, we show in detail how convergence of *(B)* to the unique viscosity solution of *(A)* can be guaranteed.

Section 3.6 We conclude by presenting numerical results and compare our approach to the method of policy iteration (cf. Section 1.2.3).

3.2 Problem Formulation

We begin by (slightly informally) introducing the type of HJB equation (see Problem 4.1.1) considered in this chapter and for which we subsequently develop a fully coherent numerical scheme. In Section 3.5, we will then show that the problems from mathematical finance listed in the Section 3.1 all match our framework and that rigorous problem formulations can be obtained by employing the theory of viscosity solutions (as introduced in [40]).

Problem 3.2.1. *Let $\mathbb{S}$, $|\mathbb{S}| < \infty$, be a finite set of controls. Let $\Omega \subset \mathbb{R}^n$ (with $n \in \mathbb{N}$) be an open set. Let $\mathcal{L}_s$, $s \in \mathbb{S}$, be a number of linear differential operators on Ω. Find a function $V : \Omega \to \mathbb{R}$ such that*

$$\min\{\mathcal{L}_s V : s \in \mathbb{S}\} = 0. \tag{3.2.1}$$

At this stage, we are not yet concerned with possible boundary/initial conditions needed for the uniqueness of a solution to Problem 3.2.1 since formulation (3.2.1) is primarily motivational. Nevertheless, we point out that (if we have an additional linear differential operator $\mathcal{L}^*$) formulation (3.2.1) is sufficiently general to include problems of the form

$$\mathcal{L}^* V + \min\{\tilde{\mathcal{L}}_s V : s \in \mathbb{S}\} = 0 \tag{3.2.2}$$

when defining $\mathcal{L}_s := \tilde{\mathcal{L}}_s + \mathcal{L}^*$, $s \in \mathbb{S}$. In many applications, $\mathcal{L}^* V$ in (3.2.2) is taken to be the time derivative of V (i.e. $\mathcal{L}^* V = \partial V / \partial t$). Also, the approach described in this paper works analogously when replacing the "min" in equations (3.2.1) and (3.2.2) by "max", which will be explained in Remark 3.2.4.

In particular, we note that (2.1.1) is a special case of (3.2.1) for $|\mathbb{S}| = 2$, i.e. the American option pricing problem from the previous chapter is a subset of the class of finitely controlled HJB equations considered here.

Since the matrix and vector operations of this chapter are slightly more complex than those of the previous chapter, we make the following remark on the notation we will be using.

Remark 3.2.2. *Let $N \in \mathbb{N}$ and define $\mathcal{N} := \{1, \dots, N\}$. Consider two vectors x, $y \in \mathbb{R}^N$ and a matrix $A \in \mathbb{R}^{N \times N}$. For any $i \in \mathcal{N}$, we denote by $(x)_i$ and $(A)_i$ the i-th coordinate and the i-th row of vector x and matrix A, respectively. When writing $x \geq 0$, we generally mean that $(x)_i \geq 0$ for all $i \in \mathcal{N}$, and we take $z := \min\{x, y\}$ to be the vector satisfying $(z)_i = \min\{(x)_i , (y)_i\}$ for all $i \in \mathcal{N}$. The definitions extend trivially to other relational operators and to the maximum of two vectors.*

In the following, whenever the choice of $N \in \mathbb{N}$ is clear from the context, we will use the set $\mathcal{N}$ as introduced in Remark 3.2.2 without explicitly saying so.

We make the assumption that we can find a sensible (i.e. stable and convergent) discretisation of Problem 4.1.1 that results in discrete systems of the following form.

Problem 3.2.3. *Find $x \in \mathbb{R}^N$ such that*

$$\min\{A_s x - b_s : s \in \mathbb{S}\} = 0, \tag{3.2.3}$$

where

- *$b_s \in \mathbb{R}^N$, $s \in \mathbb{S}$, are vectors,*
- *$A_s \in \mathbb{R}^{N\times N}$, $s \in \mathbb{S}$, are tridiagonal matrices with non-positive off-diagonal entries, positive diagonal entries, non-negative row sums, and at least one positive row sum,*
- *and the positive row sum is equally located in all matrices A_s, $s \in \mathbb{S}$.*

In Section 3.5, we will show that stable and convergent discretisations satisfying the setup of Problem 3.2.3 can be found in many relevant cases (and, most notably, for many examples from finance), justifying our assumptions. The conditions on the matrices are motivated by the fact that a tridiagonal matrix with non-positive off-diagonal entries, positive diagonal entries, non-negative row sums, and at least one positive row sum satisfies the M-matrix property (cf. Remark 1.1.2), which means, in particular, that it has a non-negative inverse; we shall make frequent use of this fact.

In particular, we note that the finite difference discretisation used in Section 2.3 in the previous chapter matches the matrix conditions stated in the above problem.

Remark 3.2.4. *If the "min" in equation* (3.2.1) *is replaced by "max", equation* (3.2.3) *becomes*

$$\min\{\tilde{A}_s x - \tilde{b}_s : s \in \mathbb{S}\} = 0, \tag{3.2.4}$$

with $\tilde{A}_s := -A_s$ and $\tilde{b}_s := -b_s$. In this case, $-\tilde{A}_s$, $s \in \mathbb{S}$, is an M-matrix, which means that $\tilde{A}_s$ has non-positive inverse $\tilde{A}^{-1} \leq 0$; therefore, all proofs that follow (Theorem 3.2.5 and Sections 3.3 and 3.4) can be shown to hold for both situations as soon as they hold for one. However, for the sake of simplicity, we solely focus on situation (3.2.3).

Theorem 3.2.5. *There exists a unique solution to Problem 3.2.3.*

Proof. The proof of existence will be given in Corollary 3.3.5. Suppose we have two solutions x_1 and x_2. For every $i \in \mathcal{N}$, there exists an $s_i \in \mathbb{S}$ such that

$$(A_{s_i})_i\, x_2 - (b_{s_i})_i = 0,$$

and it also is

$$(A_{s_i})_i \, x_1 - (b_{s_i})_i \geq 0.$$

Denote by $A^* \in \mathbb{R}^{N\times N}$ the matrix consisting of the rows $(A_{s_i})_i$, $i \in \mathcal{N}$, which is an M-matrix by our assumptions on the discretisation (cf. Problem 3.2.3). We have

$$A^*(x_1 - x_2) \geq 0,$$

and $x_1 - x_2 \geq 0$ follows since $(A^*)^{-1} \geq 0$; swapping the arguments, we get $x_2 - x_1 \geq 0$, which then proves the theorem. □

3.3 Penalisation of the Discrete Problem

In this section, we discuss how the non-linear discrete system in Problem 3.2.3 can be approximated by another (also non-linear) problem using a penalisation technique, and it will turn out that the penalised formulation can very efficiently be solved numerically. Our penalty approximation to the discrete HJB equation can be seen as an extension of the penalty scheme for American options as introduced in [29] (see also Section 1.2.2, where we have briefly summarised these results).

3.3.1 The Penalised Problem

Our penalty approximation looks as follows.

Problem 3.3.1. *Let $s_0 \in \mathbb{S}$, $\rho > 0$. Find $x_\rho \in \mathbb{R}^N$ such that*

$$(A_{s_0} \, x_\rho - b_{s_0}) - \rho \sum_{s\in\mathbb{S}\backslash\{s_0\}} \max\{b_s - A_s \, x_\rho \, , 0\} = 0. \tag{3.3.1}$$

In this, $\rho > 0$ is called the penalty parameter, and (3.3.1) can be thought of as applying a penalty of intensity ρ whenever an expression $A_s - b_s \, x_\rho$, $s \in \mathbb{S}\backslash\{s_0\}$, turns negative; by pushing up the expression $\rho \sum_{s\in\mathbb{S}\backslash\{s_0\}} \max\{b_s - A_s \, x_\rho \, , 0\}$, the penalty parameter causes $(A_{s_0} \, x_\rho - b_{s_0})$ and, implicitly, x_ρ to grow until no constraints are violated anymore.

Theorem 3.3.2. *There exists a unique solution to Problem 3.3.1.*

Proof. The proof of existence will be given in Theorem 3.4.5. Suppose we have two solutions $x_{\rho,1}$ and $x_{\rho,2}$. It is

$$(A_{s_0} \, x_{\rho,1} - b_{s_0}) - \rho \sum_{s\in\mathbb{S}\backslash\{s_0\}} \max\{b_s - A_s \, x_{\rho,1} \, , 0\} = 0 \tag{3.3.2}$$

and

$$(A_{s_0} \, x_{\rho,2} - b_{s_0}) - \rho \sum_{s\in\mathbb{S}\backslash\{s_0\}} \max\{b_s - A_s \, x_{\rho,2} \, , 0\} = 0, \tag{3.3.3}$$

and, subtracting (3.3.3) from (3.3.2), we get

$$A_{s_0}(x_{\rho,1} - x_{\rho,2}) - \rho \sum_{s \in \mathbb{S} \setminus \{s_0\}} \max\{b_s - A_s\, x_{\rho,1}\,, 0\} \\ + \rho \sum_{s \in \mathbb{S} \setminus \{s_0\}} \max\{b_s - A_s\, x_{\rho,2}\,, 0\} = 0. \quad (3.3.4)$$

Now, let $i \in \mathcal{N}$ and $r \in \mathbb{S} \setminus \{s_0\}$, and consider that

$$\begin{aligned} & - \max\{(b_r)_i - (A_r\, x_{\rho,1})_i\,, 0\} + \max\{(b_r)_i - (A_r\, x_{\rho,2})_i\,, 0\} \\ = \ & \min_{\phi \in \{0,1\}} \phi[(A_r\, x_{\rho,1})_i - (b_r)_i] - \min_{\phi \in \{0,1\}} \phi[(A_r\, x_{\rho,2})_i - (b_r)_i] \\ \leq \ & \max_{\phi \in \{0,1\}} \phi[(A_r)_i (x_{\rho,1} - x_{\rho,2})]. \end{aligned} \quad (3.3.5)$$

Hence, noting that the M-matrix property of A_{s_0} is preserved if we add rows of the matrices A_s , $s \in \mathbb{S} \setminus \{s_0\}$, to some of its rows, expression (3.3.4) implies, in the light of (3.3.5), that there exists an M-matrix $A^* \in \mathbb{R}^{N \times N}$ satisfying

$$A^*(x_{\rho,1} - x_{\rho,2}) \geq 0,$$

which then gives $(x_{\rho,1} - x_{\rho,2}) \geq 0$ since $(A^*)^{-1} \geq 0$; by the same reasoning, we can also get $(x_{\rho,2} - x_{\rho,1}) \geq 0$, which then completes the proof. $\square$

3.3.2 Properties of the Penalised Problem

What follows are properties of the penalised problem. We prove that the penalty approximation converges to the original problem as $\rho \to \infty$ and provide an estimate for the penalisation error.

Lemma 3.3.3. *Let x_ρ be the solution to Problem 3.3.1. There exists a constant $C_1 > 0$, independent of ρ, such that $\|x_\rho\|_\infty \leq C_1$.*

Proof. We rewrite equation (3.3.1) to get

$$A_{s_0}\, x_\rho = \rho \sum_{s \in \mathbb{S} \setminus \{s_0\}} \max\{b_s - A_s\, x_\rho\,, 0\} + b_{s_0}. \quad (3.3.6)$$

Hence, for every component $i \in \mathcal{N}$, we have that

$$(A_{s_0}\, x_\rho)_i = (b_{s_0})_i \quad \text{or} \quad \exists\ s \in \mathbb{S} \setminus \{s_0\} \text{ s.t. } (A_s\, x_\rho)_i < (b_s)_i\,.$$

Therefore, we may, for $i \in \mathcal{N}$, define $s_i \in \mathbb{S}$ to be such that $(A_{s_i}\, x_\rho)_i \leq (b_{s_i})_i$ is satisfied, and, introducing $A^* \in \mathbb{R}^{N \times N}$ to be the matrix having as i-th row the i-th row of A_{s_i} , $i \in \mathcal{N}$, and defining $b^* \in \mathbb{R}^N$ correspondingly, it is

$$A^*\, x_\rho \leq b^*.$$

Since A^* is an M-matrix, it is $(A^*)^{-1} \geq 0$, and we get $x_\rho \leq (A^*)^{-1}b^*$, in which $(A^*)^{-1}$ and b^* can be bounded independently of ρ because there are only finitely many compositions that can be assumed. To see that the negative part of x_ρ is bounded as well, we note that from (3.3.6) we have

$$x_\rho \geq (A_{s_0})^{-1} b_{s_0} \,,$$

in which $(A_{s_0})^{-1} \geq 0$. □

Lemma 3.3.4. *Let x_ρ be the solution to Problem 3.3.1. There exists a constant $C_2 > 0$, independent of ρ, such that*

$$A_{s_0}\, x_\rho - b_{s_0} \geq 0, \tag{3.3.7}$$

$$A_s\, x_\rho - b_s \geq -\frac{C_2}{\rho}, \quad s \in \mathbb{S}\backslash\{s_0\}, \tag{3.3.8}$$

and, for every $i \in \mathcal{N}$, it is

$$(A_{s_0}\, x_\rho - b_{s_0})_i = 0 \tag{3.3.9}$$

or $\exists\ s \in \mathbb{S}\backslash\{s_0\}$ such that

$$(A_s\, x_\rho - b_s)_i \leq \frac{C_2}{\rho}. \tag{3.3.10}$$

Combined, properties (3.3.7)–(3.3.10) *imply that* $\| \min\{A_s\, x_\rho - b_s : s \in \mathbb{S}\}\|_\infty \leq C_2/\rho$.

Proof. For $s \in \mathbb{S}\backslash\{s_0\}$, it is

$$\begin{aligned} \rho(b_s - A_s\, x_\rho) &\leq \rho \sum_{r \in \mathbb{S}\backslash\{s_0\}} \max\{b_r - A_r\, x_\rho\,, 0\} \\ &= A_{s_0}\, x_\rho - b_{s_0}\,, \end{aligned}$$

and, applying Lemma 3.3.3 to the last expression, we get that

$$A_s\, x_\rho - b_s \geq -\frac{C_2}{\rho} \tag{3.3.11}$$

for some constant $C_2 > 0$ independent of ρ. Furthermore, we have that, for every component $i \in \mathcal{N}$, it is

$$(A_{s_0}\, x_\rho - b_{s_0})_i = 0$$

or $\exists\ s \in \mathbb{S}\backslash\{s_0\}$ such that

$$(b_s - A_s\, x_\rho)_i \geq 0,$$

which, together with (3.3.11), extends to

$$|(A_s x - b_s)_i| \leq \frac{C_2}{\rho}.$$

□

Corollary 3.3.5. *Let x_ρ be the solution to Problem 3.3.1. As $\rho \to \infty$, x_ρ converges to a limit x^* which solves Problem 3.2.3.*

Proof. Since $(x_\rho)_{\rho>0}$ is bounded (as seen in Lemma 3.3.3), it has a convergent subsequence, which we will not distinguish notationally; we denote the limit of this subsequence by x^*. From (3.3.7) and (3.3.8), we may deduce that

$$A_s\, x^* - b_s \geq 0, \quad s \in \mathbb{S}.$$

Now, consider a component $i \in \mathcal{N}$. From (3.3.9) and (3.3.10), we know that, for every $\rho > 0$, there exists an $s_{i,\rho} \in \mathbb{S}$ such that

$$|(A_{s_{i,\rho}}\, x_\rho - b_{s_{i,\rho}})_i| \leq \frac{C_2}{\rho}.$$

Recalling that $|\mathbb{S}| < \infty$, we can then infer that there exists once more a subsequence of $(x_\rho)_{\rho>0}$, again not notationally distinguished, and an $s_i^* \in \mathbb{S}$ such that

$$|(A_{s_i^*}\, x_\rho - b_{s_i^*})_i| \leq \frac{C_2}{\rho} \quad \forall\, \rho > 0, \tag{3.3.12}$$

which means

$$|(A_{s_i^*}\, x^* - b_{s_i^*})_i| = 0. \tag{3.3.13}$$

Altogether, we see that x^* solves Problem 3.2.3. Since Problem 3.2.3 has a unique solution (cf. Theorem 3.2.5), the just given result holds not only for subsequences of $(x_\rho)_{\rho>0}$, but the whole sequence converges. □

Theorem 3.3.6. *Let x_ρ and x^* be the solutions to Problems 3.3.1 and 3.2.3, respectively. There exists a constant $C_3 > 0$, independent of ρ, such that*

$$\|x^* - x_\rho\|_\infty \leq \frac{C_3}{\rho}.$$

Proof. We continue in the setup of the proof of Corollary 3.3.5. From (3.3.12) and (3.3.13), we know that the original sequence $(x_\rho)_{\rho>0}$ of solutions to Problem 3.3.1 has a subsequence $(x_{\rho'})_{\rho'>0}$ such that, for every component $i \in \mathcal{N}$, there exists an $s_i^* \in \mathbb{S}$ satisfying

$$|(A_{s_i^*})_i\, (x_{\rho'} - x^*)_i| \leq \frac{C_2}{\rho'} \quad \forall\, \rho' > 0.$$

Denote the matrix consisting of the rows $(A_{s_i^*})_i$, $i \in \mathcal{N}$, by $A^* \in \mathbb{R}^{N\times N}$. Noting that A^* is an M-matrix and hence invertible, we obtain that

$$|x^* - x_{\rho'}| \leq \frac{\|(A^*)^{-1}\|_\infty\, C_2}{\rho'} \quad \forall\, \rho' > 0, \tag{3.3.14}$$

in which $\|(A^*)^{-1}\|_\infty$ can be bounded independently of ρ because there are only finitely many compositions it can assume. Since, by the same arguments, every subsequence of the original sequence $(x_\rho)_{\rho>0}$ can be shown to have a subsequence satisfying (3.3.14), we deduce that the whole sequence satisfies

$$|x^* - x_\rho| \leq \frac{C_3}{\rho} \quad \forall\, \rho > 0$$

for some constant $C_3 > 0$ independent of ρ. □

3.4 An Iterative Method for Solving the Penalised Problem

Having seen in the previous section how penalisation can be used to approximate Problem 3.2.3, it remains to answer the question how to numerically solve the non-linear discrete system (3.3.1) appearing in the penalised formulation. To this end, we present an iterative method – distantly resembling Newton's method – which has finite termination and converges to the solution to Problem 3.3.1. In the situation of American options, a similar algorithm was used in [29].

We are trying to find $y \in \mathbb{R}^N$ which satisfies (3.3.1); hence, defining

$$G(y) := (A_{s_0}\, y - b_{s_0}) - \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} \max\{b_s - A_s\, y, 0\}, \quad y \in \mathbb{R}^N,$$

we need to solve $G(y) = 0$.

Definition 3.4.1. *Let ξ be a vector (or a matrix) in $\mathbb{R}^N$ (or $\mathbb{R}^{N\times N}$). For $y \in \mathbb{R}^N$, we denote by ξ_s^y, $s \in \mathbb{S}\setminus\{s_0\}$, the vector (or matrix) consisting of*

- *all rows of ξ where the corresponding rows of $b_s - A_s\, y$ are positive*
- *and where all other rows are set to zero.*

Algorithm 3.4.2. *Define*

$$J_G(y) := A_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} A_s^y\,, \quad y \in \mathbb{R}^N.$$

Let $x^0 \in \mathbb{R}^N$ be some starting value. Then, for known x^n, $n \geq 0$, find x^{n+1} such that

$$J_G(x^n)(x^{n+1} - x^n) = -G(x^n). \tag{3.4.1}$$

Clearly, the just introduced algorithm bears strong resemblance to Newton's method. However, since (as we will see) it converges globally in finitely many steps, it is more powerful than standard Newton iteration, for which convergence can only be guaranteed in a sufficiently small neighbourhood of the solution. Furthermore, the function $G(\cdot)$ is not differentiable, which means standard Newton iteration is not directly applicable. For general results on extensions of Newton methods to semi-smooth Newton methods for the solution of particular non-smooth equations, see [41, 42] and references therein.

The algorithm is well defined since J_G is a non-singular matrix. Also, equation (3.4.1) can be written as

$$\big(A_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} A_s^{x^n}\big)(x^{n+1} - x^n) = -(A_{s_0}x^n - b_{s_0}) + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} (b_s^{x^n} - A_s^{x^n} x^n),$$

which is equivalent to

$$\big(A_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} A_s^{x^n}\big)x^{n+1} = b_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} b_s^{x^n}. \tag{3.4.2}$$

Lemma 3.4.3. *Let $x^0 \in \mathbb{R}^N$ be some starting value, and let $(x^n)_{n=0}^{\infty}$ be the sequence generated by Algorithm 4.4.4. It is $x^n \le x^{n+1}$ for $n \ge 1$.*

Proof. Writing (3.4.2) for x^n and x^{n+1}, we obtain

$$\begin{aligned} &\big(A_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} A_s^{x^n}\big)x^{n+1} = b_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} b_s^{x^n} \\ \text{and}\quad &\big(A_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} A_s^{x^{n-1}}\big)x^{n} = b_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} b_s^{x^{n-1}}, \end{aligned}$$

and subtracting yields

$$\begin{aligned} &\big(A_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} A_s^{x^n}\big)(x^{n+1} - x^n) \\ &\quad = \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} b_s^{x^n} - \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} A_s^{x^n} x^n - \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} b_s^{x^{n-1}} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} A_s^{x^{n-1}} x^n\,. \end{aligned} \tag{3.4.3}$$

In this, $A_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} A_s^{x^n}$ is an M-matrix (and has a non-negative inverse), and, therefore, the proof is complete if we can show that the right hand side of expression (3.4.3) is non-negative; to do this, we consider the rows $i \in \mathcal{N}$ of the right hand side of (3.4.3) separately. Let $s \in \mathbb{S}\setminus\{s_0\}$.

(i) If $(A_s^{x^n})_i = 0$ and $(A_s^{x^{n-1}})_i = 0$, we have $\rho(b_s^{x^n})_i - \rho(A_s^{x^n} x^n)_i - \rho(b_s^{x^{n-1}})_i + \rho(A_s^{x^{n-1}} x^n)_i = 0$.

(ii) If $(A_s^{x^n})_i \neq 0$ and $(A_s^{x^{n-1}})_i \neq 0$, we have $(A_s^{x^n})_i = (A_s^{x^{n-1}})_i$ and $(b_s^{x^n})_i = (b_s^{x^{n-1}})_i$, and, hence, we have $\rho(b_s^{x^n})_i - \rho(A_s^{x^n} x^n)_i - \rho(b_s^{x^{n-1}})_i + \rho(A_s^{x^{n-1}} x^n)_i = 0$.

(iii) If $(A_s^{x^n})_i \neq 0$ and $(A_s^{x^{n-1}})_i = 0$, we have $\rho(b_s^{x^n})_i - \rho(A_s^{x^n} x^n)_i - \rho(b_s^{x^{n-1}})_i + \rho(A_s^{x^{n-1}} x^n)_i = \rho(b_s^{x^n})_i - \rho(A_s^{x^n} x^n)_i \ge 0$.

(iv) If $(A_s^{x^n})_i = 0$ and $(A_s^{x^{n-1}})_i \neq 0$, we have $\rho(b_s^{x^n})_i - \rho(A_s^{x^n} x^n)_i - \rho(b_s^{x^{n-1}})_i + \rho(A_s^{x^{n-1}} x^n)_i = -\rho(b_s^{x^{n-1}})_i + \rho(A_s^{x^{n-1}} x^n)_i \ge 0$, where the last inequality holds because $\max\{(b_s)_i - (A_s)_i\, x^n\,, 0\} = 0$.

This completes the proof. $\square$

Lemma 3.4.4. *There exists a number $\kappa \in \mathbb{N}$ such that, for every starting value x^0, Algorithm 3.4.2 terminates after at most κ steps.*

Proof. Since $A_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} A_s^{x^n}$ is a non-singular matrix, we may write (3.4.2) as

$$x^{n+1} = \big(A_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} A_s^{x^n}\big)^{-1}\big(b_{s_0} + \rho \sum_{s\in\mathbb{S}\setminus\{s_0\}} b_s^{x^n}\big). \tag{3.4.4}$$

There is only a finite number of compositions for the right hand side of (3.4.4) as it consists of given matrices of which some rows are set to zero. Hence, there exists a set $\mathcal{C}$, $|\mathcal{C}| < \infty$, $\mathcal{C}$ independent of x^0, such that $x^n \in \mathcal{C}$ for all $n \in \mathbb{N}$. Also, by the properties of the algorithm, if $x^{n'+1} = x^{n'}$ for some $n' \in \mathbb{N}$, then it is $x^n = x^{n'}$ for all $n \geq n'$. This, together with the monotonicity of the algorithm (cf. Lemma 3.4.3), proves the lemma. □

Theorem 3.4.5. *Independently of the starting value x^0, Algorithm 3.4.2 converges (in finitely many steps) to a limit x^* which solves Problem 3.3.1.*

Proof. As the proof of convergence has already been given in Lemma 3.4.4, it only remains to show that the limit x^* satisfies $G(x^*) = 0$. This follows easily from the fact that $x^* = x^{n+1} = x^n$ for all $n \geq \kappa$, which means that (3.4.1) reduces to $-G(x^*) = -G(x^n) = J_G(x^n)(x^{n+1} - x^n) = 0$. □

3.5 Examples of HJB Equations Arising in Finance

Let $r,\ q \geq 0$, $\sigma > 0$ denote interest rate, (continuously paid) dividend rate, and volatility, respectively. We introduce the Black-Scholes operator

$$\mathcal{L}_{BS}^{r,q,\sigma} V := \frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + (r-q)S\frac{\partial V}{\partial S} - rV, \tag{3.5.1}$$

where $V = V(S,t)$ is some function of S and t.

Remark 3.5.1. *Let $M,\ N \in \mathbb{N}$. Suppose we want to solve equation (3.5.1) numerically backwards in time on the interval $[0, S_{max}] \times [0, T]$. Also, suppose that a terminal value and boundary conditions have been fixed. Let time and space grids be given by $\{jk : j+1 \in \mathcal{M}\}$ and $\{ih : i+1 \in \mathcal{N}\}$, respectively, where $\mathcal{M} := \{1, \ldots, M\}$, $\mathcal{N} := \{1, \ldots, N\}$, $k := T/(M-1)$ and $h := S_{max}/(N-1)$. Using a fully implicit finite difference method with a one-sided difference in time and central differences in space, we obtain the following scheme: For every time step $j \in \mathcal{M}\setminus\{1\}$, we have to find $V^{j-1} \in \mathbb{R}^N$ such that*

$$-A_{BS}^{r,q,\sigma} V^{j-1} + V^j = 0,$$

where $V^j \in \mathbb{R}^N$ is the vector known from the previous time step and $A_{BS}^{r,q,\sigma} \in \mathbb{R}^{N\times N}$ is a tridiagonal matrix with $(b_i)_{i=1}^N$ on the diagonal, $(c_i)_{i=1}^{N-1}$ on the upper diagonal, and $(a_i)_{i=2}^N$ on the lower diagonal; more precisely, for $i \in \mathcal{N}\setminus\{1, N\}$, the coefficients are *given by*

$$\begin{aligned} a_i^{r,q,\sigma} &= -\frac{1}{2}i^2\sigma^2 k + \frac{1}{2}i(r-q)k, \\ b_i^{r,q,\sigma} &= 1 + i^2\sigma^2 k + rk, \\ \text{and} \quad c_i^{r,q,\sigma} &= -\frac{1}{2}i^2\sigma^2 k - \frac{1}{2}i(r-q)k, \end{aligned}$$

and $b_1^{r,q,\sigma}$, $c_1^{r,q,\sigma}$, $a_N^{r,q,\sigma}$ *and* $b_N^{r,q,\sigma}$ *are chosen to accommodate the boundary conditions. In finance, boundary conditions can usually be found such that* $A_{BS}^{r,q,\sigma}$ *is an M-matrix.*

Proof. See [60]. □

The examples from mathematical finance listed in Section 3.2 (uncertain volatility models [48], transaction cost models [47], and unequal borrowing/lending rates and stock borrowing fees [3, 10, 34]), all result in HJB equations of the form (3.2.1) where every differential operator $\mathcal{L}_s$, $s \in \mathbb{S}$, can be interpreted to be of Black-Scholes type (3.5.1) for some set of parameters r, q and σ. (For details, we refer to [28, 30], where the models have been described at length.) As seen in Remark 3.5.1, a finite difference discretisation of (3.5.1) results, for every time step, in a linear system of equations where the defining matrix is an M-matrix, and, therefore, the above problems all fit into the framework of Problem 3.2.3; a detailed outline of one representative case will be given in the next section.

3.5.1 Example: European Option Pricing with Stock Borrowing Fees and Unequal Borrowing/Lending Rates

We would like to price a European option that pays $P(S)$ at time $T > 0$, where S denotes the price of the underlying. We assume the standard Black-Scholes model with the extension that the cash borrowing rate (denoted by r_b) and the cash lending rate (denoted by r_l) are not necessarily equal, and, also, that there are stock borrowing fees of r_f reflecting the fact that shorting the stock may have a certain cost. To preclude arbitrage, it must be $r_b \geq r_l \geq r_f$. Different borrowing/lending rates have been discussed in [3, 10], and stock borrowing fees are studied in [34]; the resulting mathematical model has been explained in [28].

To price a short position in the option, we have to solve the following non-linear problem. We define $S_{max} > 0$ to be a large upper limit on the value of the underlying S and set $\Omega := (0, S_{max})$. For simplicity of presentation, we assume that $P(0) = P(S_{max}) = 0$; most other boundary conditions can be treated similarly.

Problem 3.5.2. *Let* $\sigma > 0$ *and* $r_b \geq r_l \geq r_f \geq 0$. *Find* $V : \overline{\Omega} \times [0,T] \to \mathbb{R}$ *such that* $V(S,t) = P(S)$ *for* $(S,t) \in \partial\Omega \times (0,T]$, $V(S,T) = P(S)$ *for* $S \in \overline{\Omega}$, *and*

$$\max\left\{\mathcal{L}_{BS}^{r,q,\sigma} V : (r,q) \in \mathbb{S}\right\} = 0 \tag{3.5.2}$$

on $\Omega \times (0,T)$*; here, it is* $\mathbb{S} := \{(r_l, 0), (r_b, 0), (r_l, r_f), (r_b, r_b - r_l + r_f)\}$.

We use the notion of viscosity (sub-/super-) solution as developed in [40], and, in that framework, the following strong comparison principle – which guarantees the uniqueness of any viscosity solution – can be shown to hold for the pricing problem.

Theorem 3.5.3. *Suppose u and v are, respectively, viscosity sub- and supersolution of Problem 3.5.2. If the payoff satisfies $P \in \mathcal{C}(\overline{\Omega})$, then it is $u \leq v$ on $\overline{\Omega} \times (0, T]$.*

Proof. After transforming the terminal value problem into an initial value problem, this can be found directly by checking the necessary conditions in [40] or [16]. □

It now seems natural to apply the discretisation technique from Remark 3.5.1 to every Black-Scholes operator in equation (3.5.2) individually to obtain a non-linear discrete approximation of Problem 3.5.2.

Problem 3.5.4. *Let M, $N \in \mathbb{N}$ and use the notation of Remark 3.5.1. For σ and $\mathbb{S}$ as in Problem 3.5.2, find $(V^j)_{j \in \mathcal{M}} \subset \mathbb{R}^N$ such that*

$$\min \left\{ A_{BS}^{r,q,\sigma} V^{j-1} - V^j : (r, q) \in \mathbb{S} \right\} = 0, \quad j \in \mathcal{M} \backslash \{1\}, \tag{3.5.3}$$

where

- *$V^M \in \mathbb{R}^N$ is the function P evaluated on the grid $\{ih : i + 1 \in \mathcal{N}\}$*
- *and $b_1^{r,q,\sigma} = b_N^{r,q,\sigma} = 1$ and $c_1^{r,q,\sigma} = a_N^{r,q,\sigma} = 0$ in all matrices $A_{BS}^{r,q,\sigma}$.*

What follows are a few properties – stability, consistency, and monotonicity, used as defined in [6] – of the just introduced numerical scheme; based on these, we will then be able to deduce convergence.

Lemma 3.5.5. *If $\sigma^2 \geq r_b$, the discretisation in Problem 3.5.4 is unconditionally stable in the maximum norm.*

Proof. For any $(r, q) \in \mathbb{S}$ and $i \in \mathcal{N} \backslash \{1, N\}$, it is $b_i^{r,q,\sigma} \geq 1$ and $a_i^{r,q,\sigma}$, $c_i^{r,q,\sigma} \leq 0$. Hence, from (3.5.3), for any given $j \in \mathcal{M} \backslash \{1\}$ and $i \in \mathcal{N} \backslash \{1, N\}$, there is choice of $(r, q) \in \mathbb{S}$ such that

$$\begin{aligned} |V_i^{j-1}| &= \frac{1}{b_i^{r,q,\sigma}} |V_i^j - a_i^{r,q,\sigma} V_{i-1}^{j-1} - c_i^{r,q,\sigma} V_{i+1}^{j-1}| \\ &\leq \frac{1 - a_i^{r,q,\sigma} - c_i^{r,q,\sigma}}{b_i^{r,q,\sigma}} \max \left(|V_i^j|, |V_{i-1}^{j-1}|, |V_{i+1}^{j-1}| \right), \end{aligned}$$

and we infer that

$$\max_{1 \leq i \leq N} |V_i^{j-1}| \leq \max \left(|V_1^{j-1}|, |V_N^{j-1}|, \max_{1 \leq i \leq N} |V_i^j| \right) \leq \max_{1 \leq i \leq N} P(ih), \quad j \in \mathcal{M} \backslash \{1\}.$$

□

We point out that the constraint on the size of σ required in Lemma 3.5.5 can be avoided when using upwinding instead of central differences for the space derivatives. However, since, in practice, the implicit scheme runs stably regardless of the choice of σ, we do not investigate this point any further.

Lemma 3.5.6. *The discretisation in Problem 3.5.4 is consistent.*

Proof. Let $\phi : \overline{\Omega} \times [0,T] \to \mathbb{R}$, and, for $i \in \mathcal{N}$ and $j \in \mathcal{M}$, write $\phi_i^j := \phi(ih, jk)$. We then define

$$\mathfrak{S}(h,k,ih,jk,\phi_i^j,\phi) :=$$
$$\begin{cases} a_i^{r,q,\sigma}\phi_{i-1}^j + b_i^{r,q,\sigma}\phi_i^j + c_i^{r,q,\sigma}\phi_{i+1}^j - \phi_i^{j+1} & \text{if } i \in \mathcal{N}\backslash\{1,N\} \wedge j \in \mathcal{M}\backslash\{M\}, \\ \phi_i^j - P(ih) & \text{if } i \in \{1,N\} \vee j = M, \end{cases}$$

and, for general $(S,t) \in \Omega \times (0,T)$ – which is not necessarily on the grid – we set

$$\mathfrak{S}(h,k,S,t,\phi(S,t),\phi) := \mathfrak{S}(h,k,ih,jk,\phi_i^j,\phi)$$

using $i = \lceil S/h \rceil h$ and $j = \lceil t/k \rceil k$. This definition of the function $\mathfrak{S}$ is in line with the notation in [6], and, clearly, our numerical scheme from Remark 3.5.1 corresponds to $\mathfrak{S}(h,k,ih,jk,V_i^j,(V_r^j)_{r\neq i,\, j\in\mathcal{M}}) = 0$. Now, if ϕ is smooth on $\overline{\Omega} \times [0,T]$, we have

$$\liminf \frac{\mathfrak{S}(h,k,S,t,\phi(S,t),\phi)}{k} \geq \zeta(S_0\,,t_0)$$

as $h \to 0$, $k \to 0$, $t \to t_0$ and $S \to S_0$, where

$$\zeta(S_0\,,t_0) := \begin{cases} (\mathcal{L}_{BS}^{r,q,\sigma}\phi)(S_0\,,t_0) & \text{if } (S_0\,,t_0) \in \Omega \times (0,T), \\ \min\left\{(\mathcal{L}_{BS}^{r,q,\sigma}\phi)(S_0\,,t_0)\,,\ \phi(S_0\,,t_0) - P(S_0)\right\} & \text{if } S_0 \in \partial\Omega \text{ or } t_0 = T, \end{cases}$$

and the relation also holds when replacing "lim inf", "$\geq$" and "min" by "lim sup", "$\leq$" and "max", respectively. This proves the lemma. □

Lemma 3.5.7. *The discretisation in Problem 3.5.4 is monotone.*

Proof. We use the notation of Lemma 3.5.6. It is

$$\mathfrak{S}(h,k,S,t,z,\phi) \leq \mathfrak{S}(h,k,S,t,z,\psi)$$

if $\phi \geq \psi$ since $a_i^{r,q,\sigma}$, $c_i^{r,q,\sigma} \leq 0$. □

Finally, we have done enough preparatory work to formulate the following theorem on the convergence properties of our numerical scheme.

Theorem 3.5.8. *In the limit h, $k \to 0$, the solution $(V^j)_{j\in\mathcal{M}}$ to Problem 3.5.4 converges uniformly on $\overline{\Omega} \times [0,T]$ to the unique viscosity solution to Problem 3.5.2.*

Proof. Using Lemmas 3.5.5, 3.5.6 and 3.5.7, this follows from results in [6]. □

r_b	r_l	r_f	σ	T	S_{max}	M	N	tol	ρ
0.15	0.1	0.08	0.4	1	600	400	400	1e-08	1e04

Table 3.1: *The parameters used in the numerical computations. (For illustrative purposes, we take unrealistically high interest rates.)*

3.6 Numerical Results

In this section, we numerically solve Problem 3.5.2 using the penalty approach presented in Section 3.3. To have a non-trivial financial instrument, we choose

$$P(S) = \begin{cases} S/4 - 25 & \text{if } S \in [100, 200), \\ -S/4 + 75 & \text{if } S \in [200, 300), \\ 0 & \text{else}, \end{cases}$$

which is the payoff function of a butterfly spread; unless stated differently, all other parameters are set as given in Table 3.1.

After discretising the problem (cf. Section 3.5), we are left with the non-linear discrete equation in Problem 3.5.4 (which corresponds directly to Problem 3.2.3). To solve it, we employ (i) the penalty approach from Section 3.3 (which means we solve Problem 3.3.1 instead) and (ii) the policy iteration as devised in [28, 30]; a detailed outline of policy iteration for HJB equations was given in Section 1.2.3 ; our analysis in Chapter 2 is about special case of this more general form of policy iteration.

Remark 3.6.1. *The methods used to solve Problems 3.2.3 and 3.3.1 are of iterative nature, which means we need a valid check for convergence.*

- *When solving Problem 3.3.1 by Algorithm 3.4.2, we abort the iteration as soon as we find an x_ρ^n such that*

$$\frac{\|(A_{s_0}\, x_\rho^n - b_{s_0}) - \rho \sum_{s \in \mathbb{S}\setminus\{s_0\}} \max\{b_s - A_s\, x_\rho^n\,, 0\}\|_\infty}{\|\max\{b_s : s \in \mathbb{S}\}\|_\infty} \leq tol.$$

- *When solving Problem 3.2.3 by policy iteration, we abort the iteration as soon as we find an x^n such that*

$$\frac{(A_s x^n - b_s)_l}{\|\max\{b_s : s \in \mathbb{S}\}\|_\infty} \geq -tol \quad \forall\ (l, s) \in \mathcal{N} \times \mathbb{S}$$

and, for every $i \in \mathcal{N}$, there exists an $r \in \mathbb{S}$ such that

$$\frac{(A_r x^n - b_r)_i}{\|\max\{b_s : s \in \mathbb{S}\}\|_\infty} \leq tol.$$

Policy Iteration	$n = 1$	$n = 2$	$n = 3$	$n = 4$
$M,\, N = 400$	90.50%	9.50%	-	-
$M,\, N = 1000$	91.20%	8.80%	-	-
$M = 900,\, N = 30$	99.67%	0.33%	-	-
$M = 30,\, N = 900$	-	96.67%	3.33%	-
Penalty Method ($\rho = 4e03$)	$n = 1$	$n = 2$	$n = 3$	$n = 4$
$M,\, N = 400$	-	-	78.75%	21.25%
$M,\, N = 1000$	-	-	78.40%	21.60%
$M = 900,\, N = 30$	-	-	92.56%	7.44%
$M = 30,\, N = 900$	-	-	66.67%	33.33%
Penalty Method ($\rho = 1e06$)	$n = 1$	$n = 2$	$n = 3$	$n = 4$
$M,\, N = 400$	-	-	79.00%	21.00%
$M,\, N = 1000$	-	-	78.20%	21.80%
$M = 900,\, N = 30$	-	-	92.44%	7.56%
$M = 30,\, N = 900$	-	-	70.00%	30.00%

Table 3.2: *The table shows the number of iterations (denoted by n) needed when using policy iteration and Algorithm 3.4.2 to solve the non-linear discrete Problems 3.2.3 and 3.3.1, respectively. Since the number of required iterations might differ between time steps, the percentages denote how frequently a certain number occurred. Indisputably, both schemes converge astonishingly well (since generally $n \leq 4$), and, in many cases, policy iteration seems to even need only one step, whereas Algorithm 3.4.2 predominantly needs three.*

In Figures 3.2 and 3.3, we see the solution to Problem 3.5.2 for different values of r_b, r_l and r_f. To understand the effects, it helps to see the butterfly spread as having a *call side* (the left hand side) and a *put side* (the right hand side). For a short position in a vanilla call option, the hedged portfolio is long the underlying and short the bank account. Conversely, for a short position in a vanilla put option, the hedged portfolio is short the underlying and long the bank account. Therefore, the call side of the butterfly spread rises when the borrowing rate is increased (cf. Figure 3.2), and the put side rises when the borrowing rate is increased or stock borrowing fees are introduced (cf. Figures 3.2, 3.3).

To see how well the penalty approximation (cf. Section 3.3) works in practice, we observe the difference between Problems 3.2.3 and 3.3.1 in dependence on the penalty parameter ρ, where we solve the former using policy iteration (see [28, 30] and Section 1.2.3) and the latter using Algorithm 3.4.2. We measure the difference in the maximum norm at time zero and keep the time and space grid fixed at $M,\, N = 400$. Clearly, Figure 3.4 shows that the observed convergence is of first order, which confirms the result of Theorem 3.3.6. (We find almost identical convergence rates when using $(M, N) \in \{(600, 600), (1000, 1000), (900, 30), (30, 900)\}$, i.e. the observed penalty convergence is independent of the grid size.)

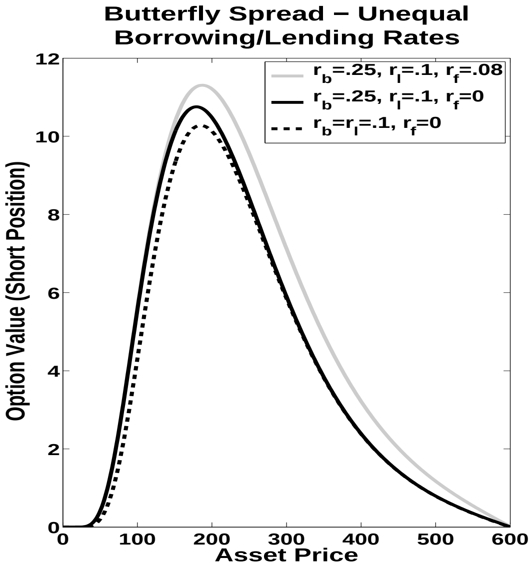

Figure 3.2: *The value of a short position in a butterfly spread. When raising the borrowing rate, the value on the left hand side (where a short position in the bank account is needed for hedging) increases and the right hand side does not change. When introducing a fee for stock borrowing, the right hand side (where a short position in the underlying is needed for hedging) increases and the left hand side does not change.*

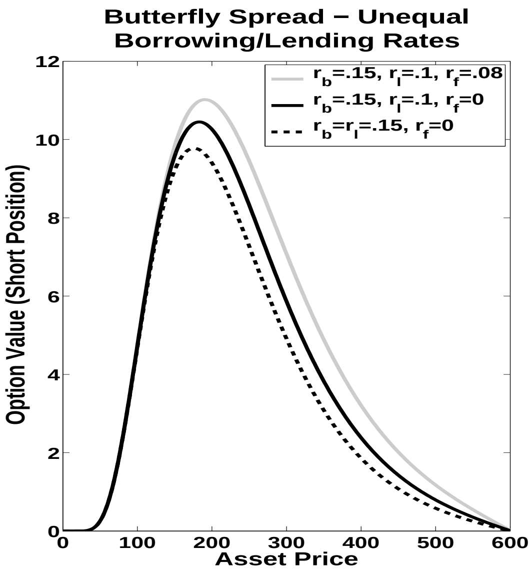

Figure 3.3: *The value of a short position in a butterfly spread. When decreasing the lending rate, the value on the right hand side (where a long position in the bank account is needed for hedging) increases and the left hand side does not change. When introducing a fee for stock borrowing, the right hand side (where a short position in the underlying is needed for hedging) increases again and the left hand side again does not change.*

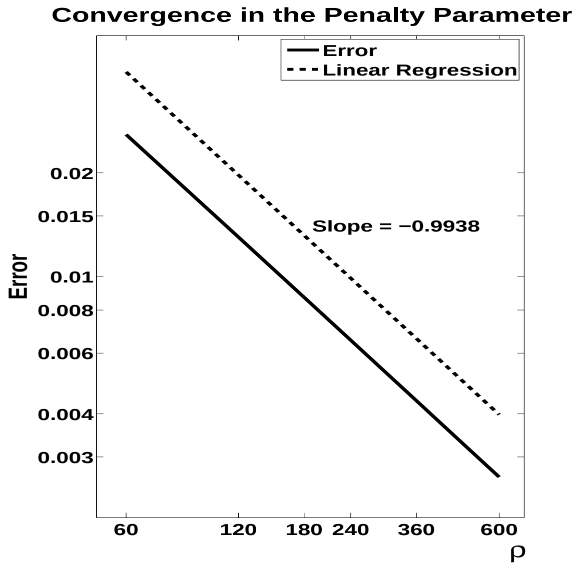

Figure 3.4: *We test the penalisation technique introduced in Section 3.3. The plot (which is in log-log coordinates) shows the difference between the solutions to Problems 3.2.3 and 3.3.1 in dependence of the penalty parameter $\rho > 0$. As ρ is increased, the two problems converge nicely. Linear regression suggests the error is of order $O(1/\rho)$, confirming the analytical results (cf. Theorem 3.3.6).*

Finally, we come to analyse the numerical efficiency of the penalty method devised in this chapter. Especially in comparison with policy iteration, there are two relevant aspects to this: (i) the number of iterations needed to solve the non-linear discrete system and (ii) the overall runtime of the method.

The necessary numbers of iterations are listed in Table 3.2, and we see that, for the examples chosen, neither method ever needs more than four iterations; for the majority of cases, policy iteration needs only one iteration and Algorithm 3.4.2 needs merely three. The overall runtimes of both approaches are listed in Table 3.3, and, in consistency with the number of iterations needed for solving the non-linear system,

Grid Size	Policy	Penalty ($\rho = 4e03$)	Penalty ($\rho = 1e06$)
$M, N = 400$	$0.216s$	$0.733s$	$0.740s$
$M, N = 1000$	$1.148s$	$4.021s$	$4.093s$
$M = 900, N = 30$	$0.142s$	$0.473s$	$0.475s$
$M = 30, N = 900$	$0.0624s$	$0.116s$	$0.121s$

Table 3.3: *We see the computational time needed to solve Problems 3.2.3 and 3.3.1 using policy iteration and Algorithm 3.4.2, respectively. The penalty scheme needs roughly three to four times longer than the policy iteration, a difference which reflects the different numbers of iterations listed in Table 3.2.*

we see that the penalty method runs up to three to four times as long as the policy iteration; nicely, the runtime of the penalty scheme does not depend on the size of the penalty parameter.

Since we generally expect finite termination (as stated in Lemma 3.4.4 and Section 1.2.3, respectively), one could, instead of proceeding as in Remark 3.6.1, also test for convergence by checking to what degree $x^n = x^{n+1}$ is satisfied. Interestingly, this is a disadvantage if a scheme converges in very few steps since one needs an extra step for verification once a certain accuracy is reached. (Using Remark 3.6.1 to test for convergence, Table 3.2 shows that, in most cases, the policy iteration reaches the desired accuracy in only one step. In [28], where it is tested for $|x^n - x^{n+1}| < tol$, the authors generally need two steps.)

In total, for the current example, both approaches clearly work very well, with the policy iteration being computationally slightly more efficient (cf. Tables 3.2, 3.3) than the penalty scheme. Since, analytically, both algorithms have very similar properties, i.e. both algorithms converge monotonically (cf. Lemma 3.4.3, [28]) and in finitely many steps (cf. Lemma 3.4.4 and Section 1.2.3), it is hard to say generally which scheme should be faster, and the question of which algorithm is advantageous might simply be down to the exact nature of the problem; e.g., we will see in Chapter 4 that policy iteration is more sensitive to the choice of the initial guess than penalisation, but, for good initial choices, it runs very competitively, as also seen above.

3.7 Conclusion

We have shown that penalty methods can be a powerful tool for the solution of HJB equations with finitely many controls, and we have included a rigorous analysis of the convergence properties of the overall approach. More precisely, we have discretised the HJB equation using standard finite differences and have approximated the resulting non-linear discrete problem using a penalisation technique. If the differential operators of the equation obey a certain structure, we have been able to prove stability and convergence of the finite difference approximation in the viscosity sense and to estimate the penalisation error to be of order $O(1/\rho)$, where $\rho > 0$ is the penalty parameter. For the numerical solution of the penalised problem, we have devised an iterative method which has finite termination in theory and, for the considered examples, converges in less than four steps in practice. There is a wide range of problems in mathematical finance matching our framework; in particular, we have presented numerical results for European option pricing in a model with different borrowing/lending rates and stock borrowing fees and have demonstrated the competitiveness of our approach.

Chapter 4

Hamilton-Jacobi-Bellman (HJB) Equations: Continuous Control and Obstacle Problems

In the previous chapter, we have presented a penalty scheme for the solution of the finite difference discretisation of an HJB equation with a finite control set, i.e. we have solved non-linear discrete systems of the form (1.1.7) with $|\mathbf{U}| < \infty$, which result from the finite difference discretisation of problems like (1.1.3) with the same control set.

In this chapter, we present a novel penalty approach for the numerical solution of continuously controlled HJB equations and HJB obstacle problems, i.e. we stay within the concepts of the previous chapter, but we now solve problems of the form (1.1.7) and (1.3.3) with $\mathbf{U}$ compact, which we obtain from continuous equations like (1.1.3) and (1.3.2), respectively. Our results include estimates of the penalisation error for a class of penalty terms, and we show that variations of Newton's method can again be used to obtain globally convergent iterative solvers for the penalised equations. Furthermore, we discuss under what conditions local quadratic convergence of the iterative solvers can be expected. We include numerical results demonstrating the competitiveness of our methods.

4.1 Introduction

We begin by introducing the two kinds of non-linear equations we aim to solve numerically in this chapter. Like in Chapter 3, the problem formulations are initially slightly informal and primarily motivational, and any equation of the described type permitting a discretisation with certain properties fits into our framework.

Problem 4.1.1. *Let $\mathbf{U} \subset \mathbb{R}$ be a compact*[1] *interval. Let $\Omega \subset \mathbb{R}^n$ (with $n \in \mathbb{N}$) be an*

[1] Our results can trivially be extended to any finite union of compact intervals in $\mathbb{R}^N$, which, in particular, includes any finite set. For simplicity of presentation, we work with $\mathbf{U}$ as given.

open set, and let $\mathcal{L}_u$, $u \in \mathbf{U}$, be a given family of affine differential operators on Ω. Find a function $V : \Omega \to \mathbb{R}$ such that

$$\inf_{u\in\mathbf{U}}\{\mathcal{L}_u V\} = 0. \tag{4.1.1}$$

Problem 4.1.2. *In the situation of Problem 4.1.1, let $\tilde{\mathcal{L}}$ be an additional affine differential operator. Find a function $W : \Omega \to \mathbb{R}$ such that*

$$\inf\left\{\sup_{u\in\mathbf{U}}\{\mathcal{L}_u W\},\ \tilde{\mathcal{L}}W\right\} = 0. \tag{4.1.2}$$

Examples in which the control set $\mathbf{U}$ in (4.1.1) is truly infinite and does not reduce to a finite set include problems of indifference pricing in finance (cf. [15] and references therein) and gas storage valuation (cf. [22, 17]). The second formulation, Problem 4.1.2, is an obstacle problem (cf. [24]) involving an HJB equation, and it bears some resemblance to a simplified form of Isaacs' equation (cf. [72, 40, 39]); considering an example from [54], we will later see that the computation of an early exercise indifference price in an incomplete market model leads to an equation of the form (4.1.2), a context in which the similarity to an obstacle problem can clearly be traced back to the American exercise feature (cf. [23]).

Existing work on this area of HJB equations includes [28, 66, 30, 50, 31], where a policy iteration algorithm is used, and our own results from Chapter 3, where a penalty approximation for HJB equations is introduced. For the solution of the penalised equations, we again consider Newton-like iterative solvers, similar but much more delicate to use than the ones devised in Chapter 3, many properties of which we prove based on results in [55, 57]. See also [43, 41, 42] regarding the use of semi-smooth Newton methods in the context of variational inequalities and their interpretation as primal-dual active set strategies, which in turn are closely related to the method of policy iteration.

We extend the previous results in two ways. First, we consider a compact set of controls; until now, except for [50], all algorithms were based on the assumption of a finite control set. Second, we also solve an HJB obstacle problem, which arises for example in mathematical finance when pricing early exercise claims in incomplete markets, e.g. in [54]; an efficient algorithm capable of handling this problem is the Ho-3 algorithm introduced in [50], which can be considered as a nested policy iteration; a brief outline of this method has been given in Section 1.2.4. The discussed techniques have many applications in mathematical finance. Besides the applications already considered in Chapter 3 and [28, 30], which include uncertain volatility models, transaction cost models, and unequal borrowing/lending rates and stock borrowing fees, our methods are applicable to many problems from portfolio optimisation, including indifference pricing [73, 15] and indifference pricing with early exercise features [54].

Structure of this Chapter

We aim to devise numerical schemes for the solution of (continuously controlled) HJB equations and HJB obstacle problems. The conceptual structure is again similar to Figure 3.1 in Chapter 3, but this chapter's setup contains many additional caveats that demand very careful treatment.

Section 4.2 We relate the two non-linear equations of Problems 4.1.1 and 4.1.2 each to a non-linear discrete system; as already in Chapter 3, the latter are chosen such that they are monotone and correspond to what naturally arises when applying a fully implicit or weighted time stepping discretisation to the equations (since, as before, this setup will later allow us to conclude convergence to the unique viscosity solution; cf. Chapter 1, Section 3.5.1, or [6, 8, 7]).

Section 4.3 We present a penalised problem which approximates the discrete HJB equation to an accuracy of $O(1/\rho)$, where $\rho > 0$ is the penalty parameter. A similar approach was used in [29] for American options, and we have already extended it to discretely controlled HJB equations in Chapter 3. The novelty of the approach described in this chapter is that we do not penalise for every control individually, as was done in Chapter 3, but only penalise the maximum violation; this significantly simplifies the penalty formulation and allows for compact control sets.

Section 4.4 We study the iterative solution of the penalised HJB equation presented in Section 4.3. We introduce two globally convergent Newton-type solvers, and we show that, when using a smooth penalty term, the classical Newton scheme can be expected to have local quadratic convergence.

Sections 4.5 and 4.6 In these two sections, we extend the techniques from Sections 4.3 and 4.4 to show that the use of penalisation (Section 4.5) with subsequent non-linear iteration (Section 4.6) is also a powerful strategy for the solution of the obstacle version of a continuously controlled HJB equation. In particular, in Section 4.6, we present two Newton-type iterative methods, one that converges globally and one that converges locally quadratically.

Section 4.7 The numerical strategies developed in Sections 4.3–4.6 are designed for HJB equations and HJB obstacle problems with a compact set of controls and continuous dependence of the differential operators on the controls. For some equations, it may be convenient to approximate the dependence on the controls, e.g. by piecewise linearisation; in this section, we prove that the previously introduced algorithms are stable with respect to such perturbations.

Section 4.8 We conclude by presenting numerical results, solving an HJB equation taken from [73] and an HJB obstacle problem found in [54]; in both cases, we compare our approach to the method of policy iteration (cf. [28, 50]).

4.2 Problem Formulation

Following [25], we introduce Z_N, $N \in \mathbb{N}$, to be the set of all real square matrices of dimension N whose off-diagonal entries are all non-positive, i.e.

$$Z_N := \{A = (a_{rs}) \in \mathbb{R}^{N\times N} : a_{rs} \le 0,\, r \ne s\},$$

and define

$$K_N := \{A \in Z_N : \exists\, x \ge 0 \text{ s.t. } Ax > 0\}$$

to be the set of M-matrices in $\mathbb{R}^N$; it can be found in [25] that $A^{-1} \ge 0$ for all matrices $A \in K_N$. Furthermore, we introduce the set

$$K_N^o := \Big\{A = (a_{r,s}) \in K_N : a_{ss} > \sum_{r \ne s} |a_{rs}|\Big\} \subset K_N\,,$$

which has the following properties.

- If, for $1 \le i \le N$, we replace the i-th row of $A \in K_N^o$ by the i-th row of some $B \in K_N^o$, the resulting matrix will still be in K_N^o.
- $A + B \in K_N^o$ for two matrices A, $B \in K_N^o$.

Throughout this chapter, the set K_N^o will be one of our main building blocks since it allows us to deduce the M-matrix property of a matrix that was constructed from a number of other matrices.

We recall the notation on relational operators introduced in Remark 3.2.2 in the previous chapter; we will use exactly the same notation here.

Making use of the definition of K_N^o, we assume, for now, that we can find sensible discretisations of Problems 4.1.1 and 4.1.2 of the following form; we will give a more rigorous justification of this assumption later on, in Section 4.8; as in previous chapters, the choice of matrices used here is partially motivated by the need for monotone finite difference discretisation to prove convergence in a viscosity setting (see Chapter 1 or Section 3.5.1).

Problem 4.2.1. *Let*

$$\mathfrak{A} : \mathbf{U} \to \mathbb{R}^{N\times N} : u \mapsto A_u \tag{4.2.1}$$

$$\textit{and}\quad \mathfrak{B} : \mathbf{U} \to \mathbb{R}^{N} : u \mapsto b_u \tag{4.2.2}$$

be continuous functions, with $A_u \in K_N^o$, $u \in \mathbf{U}$. Find $x \in \mathbb{R}^N$ such that

$$\min_{u\in\mathbf{U}}\{A_u\, x - b_u\} = 0. \tag{4.2.3}$$

The continuity requirement of $\mathfrak{A}$ and $\mathfrak{B}$ above is to be understood in the sense of any vector norm, say $\|\cdot\|_2$ or $\|\cdot\|_\infty$, in $\mathbb{R}^{N\times N}$ and $\mathbb{R}^N$.

Problem 4.2.2. *In the setting of Problem 4.2.1, let also $\tilde{A} \in K_N^o$ and $\tilde{b} \in \mathbb{R}^N$. Find $z \in \mathbb{R}^N$ such that*

$$\min\left\{ \max_{u\in\mathbf{U}}\{A_u\, z - b_u\},\ \tilde{A}z - \tilde{b}\right\} = 0. \tag{4.2.4}$$

We will, without always mentioning so explicitly, make frequent use of the fact that every function which is continuous on a compact interval attains its minimum and maximum on the interval.

Theorem 4.2.3. *The discrete Problems 4.2.1 and 4.2.2 are both uniquely solvable.*

Proof. Corollary 4.3.6 and Theorem 4.4.1 give the existence of a solution to Problem 4.2.1, and, similarly, Corollary 4.5.5 and Theorem 4.6.1 (or, alternatively, Algorithm 1.2.7) give the existence of a solution to Problem 4.2.2. Hence, it remains to show the uniqueness of the solutions. To this end, suppose we have two solutions x_1 and x_2 to Problem 4.2.1. For every $i \in \mathcal{N}$, there exists a $u_i \in \mathbf{U}$ such that

$$(A_{u_i})_i\, x_2 - (b_{u_i})_i = 0,$$

and we also have

$$(A_{u_i})_i\, x_1 - (b_{u_i})_i \geq 0.$$

Denote by $A^* \in \mathbb{R}^{N\times N}$ the matrix consisting of the rows $(A_{u_i})_i$, $i \in \mathcal{N}$. We have $A^* \in K_N^o$ and $A^*(x_1 - x_2) \geq 0$, from which we get $x_1 - x_2 \geq 0$ since $(A^*)^{-1} \geq 0$. Conversely, using the same arguments but swapping x_1 and x_2, we can also get $x_2 - x_1 \geq 0$, which then proves the uniqueness of a solution to Problem 4.2.1. Now, suppose we have two solutions z_1 and z_2 to Problem 4.2.2. For $i \in \mathcal{N}$, let u_i^1, $u_i^2 \in \mathbf{U}$ be such that

$$\max_{u\in\mathbf{U}}(A_u\, z_j - b_u)_i = (A_{u_i^j}\, z_j - b_{u_i^j})_i\,, \quad j \in \{1,2\}. \tag{4.2.5}$$

From (4.2.4), we then get that, for $i \in \mathcal{N}$, we have

$$\begin{aligned}
0 =\ & \min\{(A_{u_i^1}\, z_1 - b_{u_i^1})_i\,, (\tilde{A}z_1 - \tilde{b})_i\} - \min\{(A_{u_i^2}\, z_2 - b_{u_i^2})_i\,, (\tilde{A}z_2 - \tilde{b})_i\} \\
=\ & \min_{\phi\in\{0,1\}}[\phi(A_{u_i^1}\, z_1 - b_{u_i^1})_i + (1-\phi)(\tilde{A}z_1 - \tilde{b})_i] \\
& \qquad\qquad - \min_{\phi\in\{0,1\}}[\phi(A_{u_i^2}\, z_2 - b_{u_i^2})_i + (1-\phi)(\tilde{A}z_2 - \tilde{b})_i] \\
\leq\ & \max_{\phi\in\{0,1\}}[\phi(A_{u_i^1}\, z_1 - b_{u_i^1})_i - \phi(A_{u_i^2}\, z_2 - b_{u_i^2})_i + (1-\phi)(\tilde{A})_i(z_1 - z_2)] \\
\leq\ & \max_{\phi\in\{0,1\}}\ \max_{u\in\mathbf{U}}[\phi(A_u)_i(z_1 - z_2) + (1-\phi)(\tilde{A})_i(z_1 - z_2)].
\end{aligned}$$

At this point, we can show the uniqueness of a solution to Problem 4.2.2 by arguing analogously to the first half of this proof (where we introduced the matrix A^* and used its M-matrix properties). ☐

Next, we show that the discretisation matrices appearing in (4.2.3) and (4.2.4) have some convenient boundedness properties.

Lemma 4.2.4. *For $\mathfrak{A}$ as in Problem 4.2.1, there exists a constant $C > 0$ such that*

$$0 < \|A_u\|_\infty\,, \|A_u^{-1}\|_\infty \leq C, \quad u \in \mathbf{U}.$$

Proof. Since continuity of $\mathfrak{A}$ implies directly the existence of a constant $C_1 > 0$ such that $\|A_u\|_\infty \leq C_1$ for all $u \in \mathbf{U}$, it only remains to prove the corresponding estimate for the inverse matrices. An M-matrix is, in particular, non-singular with positive determinant (cf. [25]), and, hence, we have that $u \mapsto \det(A_u) > 0$ is a continuous and positive function on $\mathbf{U}$. As every continuous function on a compact interval assumes its minimum, there exists an $\epsilon > 0$ such that $\det(A_u) > \epsilon$ for all $u \in \mathbf{U}$. Now, by Cramer's rule, for a square non-singular matrix $A = (a_{rs}) \in \mathbb{R}^{N\times N}$, we have $A^{-1} = (\alpha_{rs})$ if we define

$$\alpha_{rs} := \frac{A_{sr}}{\det(A)}\,, \quad 1 \leq r, s \leq N,$$

where A_{sr} is the cofactor of a_{sr} in the matrix A. Since the calculation of the cofactor is, like the determinant, a continuous operation on the entries of the matrix, there exists a constant $C_2 > 0$ such that, for $u \in \mathbf{U}$ and $(A_u)^{-1} = \big((\alpha_u)_{rs}\big)$, we have

$$|(\alpha_u)_{rs}| = \left|\frac{(A_u)_{sr}}{\det(A_u)}\right| \leq \left|\frac{(A_u)_{sr}}{\epsilon}\right| \leq \frac{C_2}{\epsilon}\,, \quad 1 \leq r, s \leq N.$$

This proves the lemma. □

Corollary 4.2.5. *Suppose that, in the situation of Problem 4.2.1, we define*

$$\mathcal{U} := \underbrace{\mathbf{U} \times \mathbf{U} \times \ldots \times \mathbf{U}}_{N\text{-}times} \subset \mathbb{R}^N$$
$$\text{and} \quad \mathfrak{A}^{\mathcal{U}} : \mathcal{U} \to \mathbb{R}^{N\times N} : (u_1\,, u_2\,, \ldots, u_N) \mapsto A_{u_1 u_2 \ldots u_N}\,,$$

where $A_{u_1 u_2 \ldots u_N}$ denotes the matrix having as i-th row, $i \in \mathcal{N}$, the i-th row of A_{u_i}. In this case, $\mathcal{U}$ is a compact set, $\mathfrak{A}^{\mathcal{U}}$ is a continuous function, and $A_{u_1 u_2 \ldots u_N} \in K_N^o$ for every $(u_1\,, u_2\,, \ldots, u_N) \in \mathcal{U}$. Furthermore, there exists a constant $C > 0$ such that

$$\|A_{u_1 u_2 \ldots u_N}\|_\infty + \|A^{-1}_{u_1 u_2 \ldots u_N}\|_\infty \leq C, \quad (u_1\,, u_2\,, \ldots, u_N) \in \mathcal{U}.$$

Proof. The proof is identical to the one of Lemma 4.2.4, the only difference being that we have to deal with a continuous function $\mathcal{U} \to \mathbb{R}$ instead of $\mathbf{U} \to \mathbb{R}$. □

4.3 Penalising the Discrete HJB Equation

We begin by studying Problem 4.2.1, which we call the *HJB equation*; for nominal differentiation, Problem 4.2.2, which we discuss subsequently, will be referred to as the *HJB obstacle problem.* The penalty approximation used in this chapter is an extension of ideas used for discretely controlled HJB equations in Chapter 3 and for American options in [29]. The penalised equation is a non-linear equation itself and can be solved by an iterative scheme (cf. Section 4.4).

We introduce the penalty term

$$\Pi : \mathbb{R}^N \to \mathbb{R}^N : y \mapsto \big(\pi_1(y_1), \pi_2(y_2), \dots, \pi_N(y_N)\big)^{tr}. \tag{4.3.1}$$

Assumption 4.3.1. *For $i \in \mathcal{N}$, we take $\pi_i \in C(\mathbb{R})$ to be a non-decreasing function satisfying $\pi_i|_{(-\infty,0]} = 0$ and $\pi_i|_{(0,\infty)} > 0$.*

Problem 4.3.2. *Let $u_0 \in \mathbf{U}$, $\rho > 0$. Find $x_\rho \in \mathbb{R}^N$ such that*

$$(A_{u_0}\, x_\rho - b_{u_0}) - \rho \max_{u \in \mathbf{U}} \Pi(b_u - A_u\, x_\rho) = 0. \tag{4.3.2}$$

In (4.3.2), contrary to Chapter 3, a penalty is applied only to the maximum violation of the constraints, which is – in essence – a refinement of the approach in Chapter 3; since the solution to Problem 4.3.2 can be shown to be bounded independently of ρ (cf. Lemma 4.3.4), this guarantees all constraints to be satisfied as $\rho \to \infty$ (cf. Corollary 4.3.6).

We point out that, in Problem 4.3.2, u_0 does not require any further specification, and all the results of this section can be shown to hold for any choice of $u_0 \in \mathbf{U}$. In Section 4.8.1, we will study the practical effects of u_0 in a numerical example.

Lemma 4.3.3. *If there exists a solution to Problem 4.3.2, then it is unique.*

Proof. Suppose we have two solutions $x_{\rho 1}$ and $x_{\rho 2}$. From (4.3.2), we get that

$$A_{u_0}(x_{\rho 1} - x_{\rho 2}) - \rho \max_{u \in \mathbf{U}} \Pi(b_u - A_u\, x_{\rho 1}) + \rho \max_{u \in \mathbf{U}} \Pi(b_u - A_u\, x_{\rho 2}) = 0. \tag{4.3.3}$$

Now, let $i \in \mathcal{N}$, and define u_{i1}, $u_{i2} \in \mathbf{U}$ to be such that

$$\max_{u \in \mathbf{U}} \pi_i\big((b_u - A_u\, x_{\rho j})_i\big) = \pi_i\big((b_{u_{ij}} - A_{u_{ij}}\, x_{\rho j})_i\big), \quad j \in \{1, 2\}.$$

For the second and third term in (4.3.3), we have

$$\begin{aligned} & -\rho \max_{u \in \mathbf{U}} \pi_i\big((b_u - A_u\, x_{\rho 1})_i\big) + \rho \max_{u \in \mathbf{U}} \pi_i\big((b_u - A_u\, x_{\rho 2})_i\big) \\ = & -\rho \pi_i\big((b_{u_{i1}} - A_{u_{i1}}\, x_{\rho 1})_i\big) + \rho \pi_i\big((b_{u_{i2}} - A_{u_{i2}}\, x_{\rho 2})_i\big) \\ \leq & -\rho \pi_i\big((b_{u_{i2}} - A_{u_{i2}}\, x_{\rho 1})_i\big) + \rho \pi_i\big((b_{u_{i2}} - A_{u_{i2}}\, x_{\rho 2})_i\big). \end{aligned} \tag{4.3.4}$$

At this point, the idea is that if there exists a matrix $A^* \in K_N^o$ satisfying

$$A^*(x_{\rho 1} - x_{\rho 2}) \geq 0,$$

then we can deduce $(x_{\rho 1} - x_{\rho 2}) \geq 0$ since $(A^*)^{-1} \geq 0$, and we get uniqueness as in Theorem 4.2.3. We distinguish between the following two cases.

- If $\pi_i\big((b_{u_{i2}} - A_{u_{i2}}\, x_{\rho 1})_i\big) \geq \pi_i\big((b_{u_{i2}} - A_{u_{i2}}\, x_{\rho 2})_i\big) \geq 0$, then (4.3.4) is non-positive, which means the first term in (4.3.3) must be non-negative, and we can use $(A^*)_i = (A_{u_0})_i$.
- If $0 \leq \pi_i\big((b_{u_{i2}} - A_{u_{i2}}\, x_{\rho 1})_i\big) \leq \pi_i\big((b_{u_{i2}} - A_{u_{i2}}\, x_{\rho 2})_i\big)$, then $(b_{u_{i2}} - A_{u_{i2}}\, x_{\rho 2})_i \geq (b_{u_{i2}} - A_{u_{i2}}\, x_{\rho 1})_i$, which is equivalent to $\big(A_{u_{i2}}(x_{\rho 1} - x_{\rho 2})\big)_i \geq 0$, and we can set $(A^*)_i = (A_{u_{i2}})_i$.

□

In Section 4.4, we will discuss several choices of the penalty function Π that allow to conclude the existence of a solution to Problem 4.3.2. Here, we study the approximation of Problem 4.2.1 by Problem 4.3.2.

Lemma 4.3.4. *Suppose there exists a solution x_ρ to Problem 4.3.2. There exists a constant $C > 0$, independent of ρ and Π, such that $\|x_\rho\|_\infty \leq C$.*

Proof. We rewrite (4.3.2) to get

$$A_{u_0}\, x_\rho = \rho \max_{u \in \mathbf{U}} \Pi\big(b_u - A_u\, x_\rho\big) + b_{u_0}. \tag{4.3.5}$$

Hence, for every component $i \in \mathcal{N}$, we have that

$$(A_{u_0}\, x_\rho)_i = (b_{u_0})_i \quad \text{or} \quad \exists\, u \in \mathbf{U} \text{ s.t. } (A_u\, x_\rho)_i < (b_u)_i\,.$$

Therefore, we may, for $i \in \mathcal{N}$, define $u_i \in \mathbf{U}$ to be such that $(A_{u_i}\, x_\rho)_i \leq (b_{u_i})_i$ is satisfied, and, introducing $A^* \in K_N^o$ to be the matrix having as i-th row the i-th row of A_{u_i} , $i \in \mathcal{N}$, and defining $b^* \in \mathbb{R}^N$ correspondingly, we have

$$A^*\, x_\rho \leq b^*.$$

From Corollary 4.2.5, we then get $x_\rho \leq (A^*)^{-1} b^* \leq C$ for some constant C independent of ρ and Π. To see that the negative part of x_ρ is bounded as well, we note that, from (4.3.5), we have

$$x_\rho \geq (A_{u_0})^{-1} b_{u_0}\,,$$

in which $(A_{u_0})^{-1} \geq 0$. □

Lemma 4.3.5. *Suppose there exists a solution x_ρ to Problem 4.3.2 for every $\rho > 0$. There exists a constant $C > 0$, independent of ρ and Π, such that*

$$\Big\| \min_{u \in \mathbf{U}} \Big[\max\{A_u\, x_\rho - b_u\,, 0\} - \Pi(b_u - A_u\, x_\rho) \Big] \Big\|_\infty \leq C/\rho. \tag{4.3.6}$$

Proof. For $u \in \mathbf{U}$, we have

$$\begin{aligned} \rho \Pi(b_u - A_u\, x_\rho) &\leq \rho \max_{v \in \mathbf{U}} \Pi\big(b_v - A_v\, x_\rho\big) \\ &= A_{u_0}\, x_\rho - b_{u_0}\,, \end{aligned}$$

and, applying Lemma 4.3.4 to the last expression, we get that

$$0 \leq \Pi(b_u - A_u\, x_\rho) \leq C/\rho \tag{4.3.7}$$

for some constant $C > 0$ independent of ρ. Furthermore, for every $i \in \mathcal{N}$, we either have $(A_{u_0}\, x_\rho - b_{u_0})_i = 0$, in which case

$$\max\{(A_{u_0}\, x_\rho - b_{u_0})_i\,, 0\} - \pi_i\big((b_{u_0} - A_{u_0}x_\rho)_i\big) = 0 \leq C/\rho,$$

or $\exists\; u^* \in \mathbf{U}$ such that $\pi_i\big((b_{u^*} - A_{u^*}\, x_\rho)_i\big) \geq 0$, which, based on (4.3.7), gives

$$\max\{(A_{u^*}\, x_\rho - b_{u^*})_i\,, 0\} - \pi_i\big((b_{u^*} - A_{u^*}x_\rho)_i\big) = -\pi_i\big((b_{u^*} - A_{u^*}x_\rho)_i\big) \geq -C/\rho.$$

□

For the canonical choice $\Pi(\cdot) = \max\{\cdot, 0\}$, (4.3.6) reduces to

$$\big\| \min_{u\in\mathbf{U}}\{A_u\, x_\rho - b_u\}\big\|_\infty \leq C/\rho, \tag{4.3.8}$$

stating that x_ρ satisfies Problem 4.2.1 to an order $O(1/\rho)$; if the set $\mathbf{U}$ is discrete, estimate (4.3.8) matches similar results in [29] and Chapter 3.

Corollary 4.3.6. *Suppose there exists a solution x_ρ to Problem 4.3.2 for every $\rho > 0$. As $\rho \to \infty$, x_ρ converges to a limit x^* which solves Problem 4.2.1.*

Proof. Since $(x_\rho)_{\rho>0}$ is bounded (as seen in Lemma 4.3.4), it has a convergent subsequence, which we do not distinguish notationally; we denote the limit of this subsequence by x^*. From Lemma 4.3.5, we may deduce that

$$A_u\, x^* - b_u \geq 0, \quad u \in \mathbf{U}. \tag{4.3.9}$$

Now, consider a component $i \in \mathcal{N}$. Using again Lemma 4.3.5, we know that, for every $\rho > 0$, there exists a $u_{i\rho} \in \mathbf{U}$ such that

$$0 \leq \begin{cases} (A_{u_{i\rho}}\, x_\rho - b_{u_{i\rho}})_i \leq C/\rho & \text{if } (A_{u_{i\rho}}\, x_\rho - b_{u_{i\rho}})_i \geq 0, \\ \pi_i\big((b_{u_{i\rho}} - A_{u_{i\rho}}\, x_\rho)_i\big) \leq C/\rho & \text{if } (A_{u_{i\rho}}\, x_\rho - b_{u_{i\rho}})_i < 0. \end{cases} \tag{4.3.10}$$

Recalling that $\mathbf{U}$ is compact and $(u_{i\rho})_{\rho>0} \subset \mathbf{U}$, we can then infer that there exists once more a subsequence of $(x_\rho)_{\rho>0}$, again not notationally distinguished, and a $u_i^* \in \mathbf{U}$ such that $u_{i\rho} \to u_i^*$; hence, recalling the continuity of $\mathfrak{A}$ and $\mathfrak{B}$ in the definition of Problem 4.2.1, it follows from (4.3.10) that

$$|(A_{u_i^*}\, x^* - b_{u_i^*})_i| = 0. \tag{4.3.11}$$

Altogether, combining (4.3.9) and (4.3.11), we see that x^* solves Problem 4.2.1. Since Problem 4.2.1 has a unique solution (cf. Theorem 4.2.3), the just given result holds not only for subsequences of $(x_\rho)_{\rho>0}$, but the whole sequence converges. □

We have finally done enough preparatory work to show that – if we choose the penalty function Π correctly – the solution to Problem 4.3.2 is indeed a good approximation of the solution to Problem 4.2.1; in fact, we can obtain an approximation that is of first order in the penalty parameter, which is the same order we already had for the penalty approximation in Chapter 3.

Theorem 4.3.7. *Suppose there exists a solution x_ρ to Problem 4.3.2 for every $\rho > 0$. Let x^* be the solution to Problem 4.2.1. If, in addition to Assumption 4.3.1, we have $\pi_i|_{(0,\infty)} \in C^1((0,\infty))$, $i \in \mathcal{N}$, and $\inf\{\frac{\partial \pi_i}{\partial y}(y) : y \in (0,\infty),\ i \in \mathcal{N}\} \geq c_{min} > 0$, then there exists a constant $C > 0$, independent of ρ and Π but depending on c_{min}, such that*

$$\|x^* - x_\rho\|_\infty \leq C/\rho.$$

Proof. Without loss of generality, we may assume that $c_{min} < 1$. Let $i \in \mathcal{N}$. We use u_i^* as introduced in the proof of Corollary 4.3.6, which means, in particular, that (4.3.11) holds. Furthermore, for $\rho > 0$, we define $u_{i\rho}^{min} \in \mathbf{U}$ to be such that

$$u_{i\rho}^{min} = \begin{cases} \arg\max_{u\in\mathbf{U}} \pi_i\big((b_u - A_u\, x_\rho)_i\big) & \text{if } (A_{u_0} x_\rho - b_{u_0})_i > 0, \\ u_0 & \text{if } (A_{u_0} x_\rho - b_{u_0})_i = 0, \end{cases}$$

which means, as seen in the proof of Lemma 4.3.5, that

$$\begin{aligned} &|(A_{u_{i\rho}^{min}}\, x_\rho - b_{u_{i\rho}^{min}})_i| \\ \leq\ &\frac{1}{c_{min}}\big|\max\{(A_{u_{i\rho}^{min}}\, x_\rho - b_{u_{i\rho}^{min}})_i\,, 0\} - \pi_i\big((b_{u_{i\rho}^{min}} - A_{u_{i\rho}^{min}} x_\rho)_i\big)\big| \leq C_1/\rho \end{aligned}$$

for some constant $C_1 > 0$ independent of ρ and Π. Applying (4.3.10), (4.3.11) and the definition of $u_{i\rho}^{min}$, we now have that

$$(A_{u_{i\rho}^{min}})_i\,(x_\rho - x^*) \leq (A_{u_{i\rho}^{min}}\, x_\rho - b_{u_{i\rho}^{min}})_i \leq C_1/\rho$$

and

$$\begin{aligned} (A_{u_i^*})_i\,(x^* - x_\rho) &\leq (A_{u_i^*}\, x^* - b_{u_i^*})_i - (A_{u_{i\rho}^{min}}\, x_\rho - b_{u_{i\rho}^{min}})_i \\ &= -\,(A_{u_{i\rho}^{min}}\, x_\rho - b_{u_{i\rho}^{min}})_i \leq C_1/\rho. \end{aligned}$$

Denoting by A_1^*, $A_2^* \in K_N^o$ the matrices having as i-th row, $i \in \mathcal{N}$, the i-th rows of $A_{u_{i\rho}^{min}}$ and $A_{u_i^*}$, respectively, we get that

$$x_\rho - x^* \leq \frac{\|(A_1^*)^{-1}\|_\infty\, C_1}{\rho} \quad \text{and} \quad x^* - x_\rho \leq \frac{\|(A_2^*)^{-1}\|_\infty\, C_1}{\rho},$$

and, using Corollary 4.2.5, we may infer that

$$\|x^* - x_\rho\|_\infty \leq C_2/\rho$$

for some constant $C_2 > 0$ independent of ρ and Π. □

4.4 Solving the Penalised HJB Equation by Iteration

In the previous section, we have seen how Problem 4.2.1 can be approximated by penalisation. We will now discuss iterative methods for the solution of the penalised problem, i.e. algorithms for the computation of $y \in \mathbb{R}^N$ satisfying (4.3.2); we will see that the properties of our iterative schemes depend much on the choice of our penalty function Π.

Introducing

$$G(y) := (A_{u_0}\, y - b_{u_0}) - \rho \max_{u \in \mathbf{U}} \Pi\big(b_u - A_u\, y\big), \quad y \in \mathbb{R}^N,$$

we need to solve $G(y) = 0$. For $i \in \mathcal{N}$, $y \in \mathbb{R}^N$, we define

$$u_{\pi_i}^{min}(y) := \arg\max_{v \in \mathbf{U}} \pi_i\big((b_v - A_v\, y)_i\big),$$

and we assume that $\pi_i|_{(0,\infty)} \in C^1((0,\infty))$. For some of the following theorems, we require further that, for $i,\, j \in \mathcal{N}$,

$$\frac{\partial}{\partial y_j}\Big[\max_{u \in \mathbf{U}} \pi_i\big((b_u - A_u\, y)_i\big)\Big] = \frac{\partial \pi_i}{\partial y}\big((b_{u_{\pi_i}^{min}(y)} - A_{u_{\pi_i}^{min}(y)}\, y)_i\big)\big(A_{u_{\pi_i}^{min}(y)}\big)_{ij} \tag{4.4.1}$$

holds whenever $\frac{\partial \pi_i}{\partial y}$ is well defined, with $u_{\pi_i}^{min} \in C(\mathbb{R}^N, \mathbf{U})$; the Jacobian of G is then given by

$$\big(J_G(y)\big)_i = (A_{u_0})_i \; + \rho \frac{\partial \pi_i}{\partial y}\big((b_{u_{\pi_i}^{min}(y)} - A_{u_{\pi_i}^{min}(y)}\, y)_i\big)\big(A_{u_{\pi_i}^{min}(y)}\big)_i \tag{4.4.2}$$

for $i \in \mathcal{N}$, $y \in \mathbb{R}^N$; if $\frac{\partial \pi_i}{\partial y}$ does not exist at 0, we set $\frac{\partial \pi_i}{\partial y}(0) := \lim_{y \uparrow 0} \frac{\partial \pi_i}{\partial y}(y) = 0$. For the numerical examples following later in this chapter, (4.4.1) is generally satisfied.

The next theorem states that a globally convergent iterative scheme for the solution of Problem 4.3.2 exists whenever Π is a smooth and non-decreasing function.

Theorem 4.4.1. *If, in addition to Assumption 4.3.1, for $i \in \mathcal{N}$, $\pi_i \in C^1(\mathbb{R})$ and $\frac{\partial \pi_i}{\partial y} \geq 0$, and (4.4.1) is satisfied, then Newton's method with line search as introduced by J.-S. Pang in [55] presents a globally convergent iterative scheme for the solution of Problem 4.3.2. In particular, this means that there exists a solution to Problem 4.3.2.*

Proof. We will show that conditions (a)-(d) in Theorem 4 in [55] are satisfied. First of all, since G is continuously partially differentiable, it is differentiable as a function $\mathbb{R}^N \to \mathbb{R}^N$, which, in particular, guarantees B-differentiability of G. In (a), we need to show that, for arbitrary $y^0 \in \mathbb{R}^N$, the set $\{y \in \mathbb{R}^N : \|G(y)\|_\infty \leq \|G(y^0)\|_\infty\}$ is bounded; since, for $i \in \mathcal{N}$, we have $(A_{u_0}\, y)_i \geq (b_{u_0})_i$, and either $(A_{u_0}\, y)_i \leq \big(G(y^0) +$

$b_{u_0}\big)_i$ or $(A_{u^{min}_{\pi_i}(y)}\, y)_i \leq (b_{u^{min}_{\pi_i}(y)})_i$, this can easily be inferred by applying Corollary 4.2.5. The partial derivatives of Π are non-negative, which means $J_G(y) \in K^o_N$, $y \in \mathbb{R}^N$, and, hence, (b) holds. Given the continuity of the partial derivatives of G, we can apply Lemma 1 and Theorem 2 in [55] to obtain (c). Finally, in (d), it is sufficient to show that, for any compact set $\mathcal{B} \subset \mathbb{R}$, there exists a constant $c > 0$ such that

$$\|J_G(y)v\|_\infty \geq c \tag{4.4.3}$$

for all $y \in \mathcal{B}$, $v \in \mathbb{R}^N$, $\|v\|_\infty = 1$. We prove (4.4.3) by contradiction, i.e. suppose there exist sequences $(y^n)_{n\geq 1} \subset \mathcal{B}$ and $(v^n)_{n\geq 1} \subset \mathbb{R}^N$, $\|v^n\|_\infty = 1$ for all $n \geq 1$, such that $\lim_{n\to\infty} \|J_G(y^n)v^n\|_\infty = 0$. Based on the boundedness of the sequences $(y^n)_{n\geq 1}$ and $(v^n)_{n\geq 1}$, we infer the existence of subsequences – not notationally distinguished – converging to limits $y^* \in \mathcal{B}$ and $v^* \in \mathbb{R}^N$, respectively, where $\|v^*\|_\infty = 1$, and we get $\lim_{n\to\infty} \|J_G(y^n)v^n\|_\infty = \|J_G(y^*)v^*\|_\infty = 0$. Now, from $\frac{\partial \pi_i}{\partial y} \geq 0$, $i \in \mathcal{N}$, we have that $J_G(y^*) \in K^o_N$, and we note that $J_G(y^*)v^* = 0$ implies $v^* = 0$, which is a contradiction to $\|v^*\|_\infty = 1$. Hence, we may conclude that (4.4.3) must hold true, which completes the proof. □

Having seen that Newton's method with line search can lead to a globally convergent scheme, we next come to look at Newton's method in its classical form.

Algorithm 4.4.2. *(Newton's Method for the pen. HJB Equation) Let $x^0 \in \mathbb{R}^N$ be some starting value. Then, for known x^n, $n \geq 0$, find x^{n+1} such that*

$$J_G(x^n)(x^{n+1} - x^n) = -G(x^n). \tag{4.4.4}$$

Theorem 4.4.3. *Suppose the conditions of Theorem 4.4.1 are satisfied, and let x_ρ be the solution to Problem 4.3.2. There exists a neighbourhood $\mathcal{B}$ of x_ρ such that, for any starting value $x^0 \in \mathcal{B}$, the Newton sequence $(x^n)_{n\geq 0}$ generated by Algorithm 4.4.2 is well defined, remains in $\mathcal{B}$ and converges to x_ρ ; the rate of convergence is quadratic.*

Proof. If G has a strong and non-singular F-derivative at x_ρ, then the result can be found in Theorem 3 in [55]. Given the continuity of the partial derivatives of G, we can apply Theorem 2 in [55] to obtain the existence of a strong F-derivative of G, which coincides with J_G. Since $\frac{\partial \pi_i}{\partial y} \geq 0$, $i \in \mathcal{N}$, implies $J_G(x_\rho) \in K^o_N$, the strong F-derivative of G at x_ρ is indeed non-singular. □

Now, if we use $\Pi(\cdot) = \max\{\cdot, 0\}$ (in which case, based on (4.4.2), J_G is notationally still well defined), (4.4.4) simplifies, and we get the following special case of Algorithm 4.4.2.

Algorithm 4.4.4. *(Newton-like Method for pen. HJB Eq.) For $i \in \mathcal{N}$ and $y \in \mathbb{R}^N$, define $u^{min}_i(y) \in \mathbf{U}$ to be such that*

$$u^{min}_i(y) = \arg\max_{v\in\mathbf{U}} \Big[\max\{(b_v - A_v\, y)_i\, , 0\} \Big], \tag{4.4.5}$$

and set $A^{min}(y) \in \mathbb{R}^{N\times N}$ and $b^{min}(y) \in \mathbb{R}^N$ to be matrix and vector consisting of

- *rows* $(A_{u_i^{min}(y)})_i$ *and* $(b_{u_i^{min}(y)})_i$ *,* $i \in \mathcal{N}$*, respectively, if we have* $(b_{u_i^{min}(y)} - A_{u_i^{min}(y)}\, y)_i > 0$
- *and having zero rows if* $(b_{u_i^{min}(y)} - A_{u_i^{min}(y)}\, y)_i \leq 0$.

If $\Pi(y) = \max\{y, 0\}$*, we now have* $J_G(y) = A_{u_0} + \rho A^{min}(y) \in K_N^o$ *, and (4.4.4) becomes*

$$\big(A_{u_0} + \rho A^{min}(x^n)\big)(x^{n+1} - x^n) = -(A_{u_0}\, x^n - b_{u_0}) + b^{min}(x^n) - \rho A^{min}(x^n)x^n\,,$$

which is equivalent to

$$\big(A_{u_0} + \rho A^{min}(x^n)\big)x^{n+1} = b_{u_0} + \rho\, b^{min}(x^n). \tag{4.4.6}$$

Lemma 4.4.5. *Let* $x^0 \in \mathbb{R}^N$ *be some starting value, and let* $(x^n)_{n=0}^{\infty}$ *be the sequence generated by Algorithm 4.4.4. We have* $x^n \leq x^{n+1}$ *for* $n \geq 1$.

Proof. Writing (4.4.6) for x^n and x^{n+1}, we obtain

$$\begin{aligned} &\big(A_{u_0} + \rho\, A^{min}(x^n)\big)x^{n+1} = b_{u_0} + \rho\, b^{min}(x^n) \\ \text{and}\quad &\big(A_{u_0} + \rho\, A^{min}(x^{n-1})\big)x^n = b_{u_0} + \rho\, b^{min}(x^{n-1}), \end{aligned}$$

and subtracting yields

$$\begin{aligned} &\big(A_{u_0} + \rho\, A^{min}(x^n)\big)(x^{n+1} - x^n) \\ &\qquad = \rho\, b^{min}(x^n) - \rho\, A^{min}(x^n)\, x^n - \rho\, b^{min}(x^{n-1}) + \rho\, A^{min}(x^{n-1})\, x^n\,. \end{aligned} \tag{4.4.7}$$

In this, $A_{s_0} + \rho\, A^{min}(x^n) \in K_N^o$ has a non-negative inverse, and, therefore, the proof is complete if we can show that the right-hand side of expression (4.4.7) is non-negative; to do this, we consider the rows $i \in \mathcal{N}$ of the right-hand side of (4.4.7) separately.

(i) If $\big(b^{min}(x^{n-1}) - A^{min}(x^{n-1})\, x^{n-1}\big)_i = 0$, the right-hand side of (4.4.7) equals $\big(\rho\, b^{min}(x^n) - \rho\, A^{min}(x^n)\, x^n\big)_i \geq 0$.

(ii) If $\big(b^{min}(x^{n-1}) - A^{min}(x^{n-1})\, x^{n-1}\big)_i > 0$, the right-hand side of (4.4.7) equals

$$\begin{aligned} &\big(\rho\, b^{min}(x^n) - \rho\, A^{min}(x^n)\, x^n - \rho\, b^{min}(x^{n-1}) + \rho\, A^{min}(x^{n-1})\, x^n\big)_i \\ \geq\ &\big(\rho\, b^{min}(x^n) - \rho\, A^{min}(x^n)\, x^n - \rho\, b^{min}(x^n) + \rho\, A^{min}(x^n)\, x^n\big)_i = 0. \end{aligned}$$

This completes the proof. $\square$

Lemma 4.4.6. *Let* $(x^n)_{n=0}^{\infty}$ *be the sequence generated by Algorithm 4.4.4. There exists a constant* $C > 0$ *such that, for every starting value* x^0 *,* $\|x^n\|_\infty \leq C$*,* $n \geq 1$.

Proof. We may write (4.4.6) as

$$x^{n+1} = \big(A_{u_0} + \rho A^{min}(x^n)\big)^{-1}\big(b_{u_0} + \rho\, b^{min}(x^n)\big). \tag{4.4.8}$$

The boundedness of the right-hand side of (4.4.8) follows from Corollary 4.2.5. $\square$

Theorem 4.4.7. *Independently of the starting value x^0, Algorithm 4.4.4 converges to a limit x_ρ which solves Problem 4.3.2.*

Proof. From Lemmas 4.4.5 and 4.4.6, we have the existence of a limit x_ρ such that $x^n \to x_\rho$ as $n \to \infty$. Hence, it only remains to show that the limit x_ρ satisfies $G(x_\rho) = 0$. This follows easily from (4.4.4), the boundedness of J_G and the continuity of the function G. □

We emphasise that Algorithm 4.4.4 is based on the particular choice $\Pi = \max\{\cdot, 0\}$ and that all convergence results of Theorem 4.4.7 have been shown independently of assumption (4.4.1).

4.5 Penalising the Discrete HJB Obstacle Problem

Having dealt with Problem 4.2.1 (termed *HJB equation*) in the previous sections, we now consider Problem 4.2.2 (termed *HJB obstacle problem*), which has to be treated differently due to the combination of "min" and "max" operators. We use the same penalty term Π as introduced in (4.3.1), and we again make Assumption 4.3.1.

Problem 4.5.1. *Let $\rho > 0$. Find $z_\rho \in \mathbb{R}^N$ such that*

$$\max_{u\in\mathbf{U}}\{A_u\, z_\rho - b_u\} - \rho\Pi\big(\tilde{b} - \tilde{A}z_\rho\big) = 0. \tag{4.5.1}$$

If one treats the "max" inside the "min" in (4.2.4) as if $|\mathbf{U}| = 1$, then the penalty formulation (4.5.1) corresponds to the penalisation technique introduced in [29], and, thus, (4.5.1) has a heuristic interpretation; however, mathematically, the situation is a bit more subtle, and we will now study various properties of Problem 4.5.1.

Lemma 4.5.2. *If there exists a solution to Problem 4.5.1, then it is unique.*

Proof. Suppose we have two solutions $z_{\rho 1}$ and $z_{\rho 2}$. For $i \in \mathcal{N}$, we use u_i^1, $u_i^2 \in \mathbf{U}$ as defined in (4.2.5). From equation (4.5.1), we then get that, for $i \in \mathcal{N}$,

$$\begin{aligned}&\big(A_{u_i^1}(z_{\rho 1} - z_{\rho 2})\big)_i - \rho\pi_i\big((\tilde{b} - \tilde{A}z_{\rho 1})_i\big) + \rho\pi_i\big((\tilde{b} - \tilde{A}z_{\rho 2})_i\big)\\ &\qquad \geq (A_{u_i^1}z_{\rho 1} - b_{u_i^1} - A_{u_i^2}z_{\rho 2} + b_{u_i^2})_i - \rho\pi_i\big((\tilde{b} - \tilde{A}z_{\rho 1})_i\big) + \rho\pi_i\big((\tilde{b} - \tilde{A}z_{\rho 2})_i\big) = 0.\end{aligned}$$

At this point, we have an expression similar to (4.3.3), and we can finish off by following the lines of the proof of Lemma 4.3.3. □

Lemma 4.5.3. *Assume there exists a solution z_ρ to Problem 4.5.1 for every $\rho > 0$. There exists a constant $C > 0$, independent of ρ and Π, such that*

$$\|(z_\rho)_{\rho>0}\|_\infty \leq C.$$

Proof. Applying Corollary 4.2.5, we can argue as in the proof of Lemma 4.3.4. □

Lemma 4.5.4. *Assume there exists a solution z_ρ to Problem 4.5.1 for every $\rho > 0$. There exists a constant $C > 0$, independent of ρ and Π, such that*

$$\Big\| \min\Big\{ \max_{u\in\mathbf{U}}\{A_u\, z_\rho - b_u\}, \max\{\tilde{A}z_\rho - \tilde{b}, 0\} - \Pi\big(\tilde{b} - \tilde{A}z_\rho\big)\Big\}\Big\|_\infty \le C/\rho. \tag{4.5.2}$$

Proof. Based on Lemma 4.5.3, we can directly infer from (4.5.1) that

$$0 \le \Pi\big(\tilde{b} - \tilde{A}z_\rho\big) \le C/\rho.$$

At this stage, we obtain estimate (4.5.2) by pointing out that, for $i \in \mathcal{N}$, either $\max_{u\in\mathbf{U}}(A_u\, z_\rho - b_u)_i = 0$ (which also means $\pi_i\big((\tilde{b} - \tilde{A}z_\rho)_i\big) = 0$) or $\max\{(\tilde{A}z_\rho - \tilde{b})_i\,, 0\} - \pi_i\big((\tilde{b} - \tilde{A}z_\rho)_i\big) \ge -C/\rho$. □

For the canonical choice $\Pi(\cdot) = \max\{\cdot, 0\}$, (4.5.2) becomes

$$\Big\| \min\Big\{ \max_{u\in\mathbf{U}}\{A_u\, z_\rho - b_u\}, \tilde{A}z_\rho - \tilde{b}\Big\}\Big\|_\infty \le C/\rho,$$

meaning that z_ρ satisfies Problem 4.2.2 to an order $O(1/\rho)$; this property is consistent with the penalty approximation of the HJB equation discussed in Section 4.3 (and also with the penalty approximations in Chapter 3 and [29].)

Corollary 4.5.5. *Suppose there exists a solution z_ρ to Problem 4.5.1 for every $\rho > 0$. As $\rho \to \infty$, z_ρ converges to a limit z^* which solves Problem 4.2.2.*

Proof. This can be shown by arguing as in the proof of Corollary 4.3.6. □

Theorem 4.5.6. *Suppose there exists a solution z_ρ to Problem 4.5.1 for every $\rho > 0$. Let z^* be the solution to Problem 4.2.2. If, in addition to Assumption 4.3.1, we have $\pi_i \in C^1((0,\infty))$, $i \in \mathcal{N}$, and $\inf\{\frac{\partial \pi_i}{\partial y}(y) : y \in (0,\infty),\ i \in \mathcal{N}\} \ge c_{min} > 0$, then there exists a constant $C > 0$, independent of ρ and Π but depending on c_{min}, such that*

$$\|z^* - z_\rho\|_\infty \le C/\rho.$$

Proof. We proceed similarly to the proof of Theorem 4.3.7. Without loss of generality, we may assume that $c_{min} < 1$. Let $i \in \mathcal{N}$. Let $(A^*_i\,, b^*_i) \in \{(\tilde{A}, \tilde{b})\} \cup \{(A_u\,, b_u) : u \in \mathbf{U}\}$ be such that

$$\min\Big\{ \max_{u\in\mathbf{U}}(A_u\, z^* - b_u)_i\,, (\tilde{A}z^* - \tilde{b})_i \Big\} = (A^*_i\, z^* - b^*_i)_i = 0.$$

Similarly, applying Lemma 4.5.4, let $(A_{i\rho}\,, b_{i\rho}) \in \{(\tilde{A}, \tilde{b})\} \cup \{(A_u\,, b_u) : u \in \mathbf{U}\}$ be such that

$$\begin{aligned}
|(A_{i\rho}\, z_\rho - b_{i\rho})_i| &= \Big| \min\Big\{ \max_{u\in\mathbf{U}}(A_u\, z_\rho - b_u)_i\,, (\tilde{A}z_\rho - \tilde{b})_i \Big\}\Big| \\
&\le \frac{1}{c_{min}}\Big| \min\Big\{ \max_{u\in\mathbf{U}}(A_u\, z_\rho - b_u)_i\,, \max\{(\tilde{A}z_\rho - \tilde{b})_i\,, 0\} - \pi_i\big((\tilde{b} - \tilde{A}z_\rho)_i\big)\Big\}\Big| \\
&\le \frac{C_1}{c_{min}\,\rho}
\end{aligned}$$

for some constant $C_1 > 0$ independent of ρ and Π. Furthermore, as already in (4.2.5), let u_i^*, $u_i^\rho \in \mathbf{U}$ such that $(A_{u_i^*} z^* - b_{u_i^*})_i = \max_{u\in\mathbf{U}}(A_u z^* - b_u)_i$ and $(A_{u_i^\rho} z_\rho - b_{u_i^\rho})_i = \max_{u\in\mathbf{U}}(A_u z_\rho - b_u)_i$. We first consider $z_\rho - z^*$, for which we distinguish two cases. ($C_2, C_3 > 0$ are taken to be constants independent of ρ and Π.)

(i) If $A_{i\rho} = \tilde{A}$, we have $\big(A_{i\rho}(z_\rho - z^*)\big)_i \leq (\tilde{A} z_\rho - \tilde{b})_i - (A_i^* z^* - b_i^*)_i \leq C_2/\rho$.

(ii) If $A_{i\rho} = A_{u_i^\rho}$, we have $\big(A_{u_i^*}(z_\rho - z^*)\big)_i \leq (A_{u_i^\rho} z_\rho - b_{u_i^\rho})_i - (A_i^* z^* - b_i^*)_i \leq C_3/\rho$.

Now, for the reverse case, $z^* - z_\rho$, similar estimates can be obtained by swapping the roles of "$*$" and "ρ". The proof can be completed by following the lines of the proof of Theorem 4.3.7. $\square$

4.6 Solving the Penalised HJB Obstacle Problem by Iteration

In Section 4.4, we have discussed how to iteratively solve the penalty approximation of the discrete HJB equation. Similarly, in this section, we discuss iterative methods for the penalty approximation of the discrete HJB obstacle problem. We find that, again, results depend much on the choice of Π, and the obtained algorithms do not share all of the properties of those in Chapter 3 and Section 4.4.

Defining

$$H(y) := \max_{u\in\mathbf{U}}\{A_u\, y - b_u\} - \rho\Pi\big(\tilde{b} - \tilde{A}y, 0\big) = 0$$

for $y \in \mathbb{R}^N$, to solve (4.5.1), we need to compute y such that $H(y) = 0$. For $i \in \mathcal{N}$, $y \in \mathbb{R}^N$, we define

$$u_i^{max}(y) := \arg\max_{v\in\mathbf{U}}\,(A_v\, y - b_v)_i\,, \tag{4.6.1}$$

and we assume that $\pi_i|_{(0,\infty)} \in C^1((0,\infty))$, $u_i^{max} \in C(\mathbb{R}^N, \mathbf{U})$, and

$$\frac{\partial}{\partial y_j}\Big[\max_{u\in\mathbf{U}}(A_u\, y - b_u)_i\Big] = \big(A_{u_i^{max}(y)}\big)_{ij} \in C(\mathbb{R}^N, \mathbb{R}) \tag{4.6.2}$$

for i, $j \in \mathcal{N}$. The Jacobian of H is then given by

$$\big(J_H(y)\big)_i := \big(A_{u_i^{max}(y)}\big)_i + \rho\frac{\partial \pi_i}{\partial y}\big((\tilde{b} - \tilde{A}y)_i\big)(\tilde{A})_i \tag{4.6.3}$$

for $i \in \mathcal{N}$, $y \in \mathbb{R}^N$, whenever $\frac{\partial \pi_i}{\partial y}$ is well defined; if $\frac{\partial \pi_i}{\partial y}$ does not exist at 0, we set $\frac{\partial \pi_i}{\partial y}(0) := \lim_{y\uparrow 0}\frac{\partial \pi_i}{\partial y}(y) = 0$. Like in Section 4.4, we point out that, for the numerical examples following later in this chapter, (4.6.2) is generally satisfied.

As already in Theorem 4.4.1, whenever Π is a smooth and non-decreasing function, we can find a globally convergent iterative scheme for the solution of Problem 4.5.1.

Theorem 4.6.1. *If, in addition to Assumption 4.3.1, for $i \in \mathcal{N}$, $\pi_i \in C^1(\mathbb{R})$ and $\frac{\partial \pi_i}{\partial y} \geq 0$, then Newton's method with line search as introduced by J.-S. Pang in [55] presents a globally convergent iterative scheme for the solution of Problem 4.5.1. In particular, this means that there exists a solution to Problem 4.5.1.*

Proof. The proof is a minor modification of the proof of Theorem 4.4.1. □

Knowing that Problem 4.5.1 has a solution for sufficiently smooth penalty terms Π, we can now proceed to showing that there also exists a solution if we penalise using the *max*-function.

Corollary 4.6.2. *If $\Pi(\cdot) = \max\{\cdot, 0\}$, then there exists a solution to Problem 4.5.1.*

Proof. Let $(\Pi^\epsilon)_{\epsilon>0}$ be a sequence of penalty functions satisfying $\pi_i^\epsilon \in C^\infty(\mathbb{R})$, $0 \leq \frac{\partial \pi_i^\epsilon}{\partial y} \leq 1$ and $\| \max\{y, 0\} - \Pi^\epsilon(y)\|_\infty \leq \epsilon$ for $i \in \mathcal{N}$, $y \in \mathbb{R}$, $\epsilon > 0$, and let $(z_\epsilon^*)_{\epsilon>0}$ be the sequence of solutions to Problem 4.5.1 corresponding to $(\Pi^\epsilon)_{\epsilon>0}$. Based on Lemma 4.5.3, we know that there exists a subsequence of $(z_\epsilon^*)_{\epsilon>0}$, not notationally distinguished, that converges to a limit $z^* \in \mathbb{R}^N$ as $\epsilon \to 0$. We aim to show that z^* solves Problem 4.5.1 for $\Pi(\cdot) = \max\{\cdot, 0\}$. We have

$$\begin{aligned}
&\| \max_{u\in\mathbf{U}}\{A_u\, z^* - b_u\} - \rho \max\{\tilde{b} - \tilde{A}z^*, 0\}\|_\infty \\
&= \| \max_{u\in\mathbf{U}}\{A_u\, z_\epsilon - b_u\} - \rho\Pi^\epsilon\big(\tilde{b} - \tilde{A}z_\epsilon\big) - \max_{u\in\mathbf{U}}\{A_u\, z^* - b_u\} + \rho \max\{\tilde{b} - \tilde{A}z^*, 0\}\|_\infty \\
&\leq \| \max_{u\in\mathbf{U}}\{A_u\, z_\epsilon - b_u\} - \max_{u\in\mathbf{U}}\{A_u\, z^* - b_u\}\|_\infty \\
&\qquad\qquad\qquad + \|\rho\Pi^\epsilon\big(\tilde{b} - \tilde{A}z_\epsilon\big) - \rho \max\{\tilde{b} - \tilde{A}z^*, 0\}\|_\infty\,,
\end{aligned}$$

in which we obtain

$$\lim_{\epsilon\to 0} \| \max_{u\in\mathbf{U}}\{A_u\, z_\epsilon - b_u\} - \max_{u\in\mathbf{U}}\{A_u\, z^* - b_u\}\|_\infty = 0$$

if we argue as in the proof of Corollary 4.3.6, and

$$\begin{aligned}
&\|\rho\Pi^\epsilon\big(\tilde{b} - \tilde{A}z_\epsilon\big) - \rho \max\{\tilde{b} - \tilde{A}z^*, 0\}\|_\infty \\
&\leq \|\rho\Pi^\epsilon\big(\tilde{b} - \tilde{A}z^*\big) - \rho \max\{\tilde{b} - \tilde{A}z^*, 0\}\|_\infty + \|\rho\Pi^\epsilon\big(\tilde{b} - \tilde{A}z^*\big) - \rho\Pi^\epsilon\big(\tilde{b} - \tilde{A}z_\epsilon\big)\|_\infty \\
&\leq \rho\epsilon\ + \rho\|z_\epsilon - z^*\|_\infty
\end{aligned}$$

as $\epsilon \to 0$ by the properties of $(\Pi^\epsilon)_{\epsilon>0}$; hence, we have $\max_{u\in\mathbf{U}}\{A_u\, z^* - b_u\} - \rho \max\{\tilde{b} - \tilde{A}z^*, 0\} = 0$. □

Having, as in Section 4.4, established that Newton's method with line search can be used to solve the penalised HJB obstacle problem, we again proceed to Newton's method in its classical form.

Algorithm 4.6.3. *(Newton's Method for the pen. HJB Obstacle Prob.) Let $z^0 \in \mathbb{R}^N$ be some starting value. Then, for known z^n, $n \geq 0$, find z^{n+1} such that*

$$J_H(z^n)(z^{n+1} - z^n) = -H(z^n). \tag{4.6.4}$$

Theorem 4.6.4. *Suppose the conditions of Theorem 4.6.1 are satisfied, and let z_ρ be the solution to Problem 4.5.1. There exists a neighbourhood $\mathcal{B}$ of z_ρ such that, for any starting value $z^0 \in \mathcal{B}$, the Newton sequence $(z^n)_{n\geq 0}$ generated by Algorithm 4.6.3 is well defined, remains in $\mathcal{B}$ and converges to z_ρ ; the rate of convergence is quadratic.*

Proof. The result follows from Theorem 3 in [55] if H is B-differentiable and has a strong and non-singular F-derivative at z_ρ. Recalling (4.6.2), we can argue as in the proof of Theorem 4.4.3 to obtain B-differentiability of H. From Theorem 2 in [55], it follows that H has a strong F-derivative, which is non-singular since $J_H(y) \in K_N^o$ for all $y \in \mathbb{R}^N$. □

We point out that, based on the assumptions of Theorem 4.6.1, the Lipschitz property of J_H , additionally required in Theorem 4.6.4, depends entirely on the regularity of $y \mapsto \max_{u\in\mathbf{U}}\{A_u\, y - b_u\}$ in (4.6.2).

If we use $\Pi(\cdot) = \max\{\cdot, 0\}$ (in which case, based on (4.6.3), J_H is notationally still well defined), (4.6.4) simplifies, and we get the following special case of Algorithm 4.6.3.

Algorithm 4.6.5. *(Newton-like Method for the pen. HJB Obstacle Prob.) For $i \in \mathcal{N}$ and $y \in \mathbb{R}^N$, define $u_i^{max}(y) \in \mathbf{U}$ as in (4.6.1), and set*

$$u^{max}(y) := (u_1^{max}(y)\,, u_2^{max}(y)\,, \ldots, u_N^{max}(y))^{tr}.$$

Furthermore, define $A^+(y) \in \mathbb{R}^{N\times N}$ and $b^+(y) \in \mathbb{R}^N$ to be matrix and vector consisting of

- *rows $(\tilde{A})_i$ and $(\tilde{b})_i$, $i \in \mathcal{N}$, respectively, if* $\max\{(\tilde{b} - \tilde{A}y)_i\,, 0\} > 0$
- *and having zero rows if* $\max\{(\tilde{b} - \tilde{A}y)_i\,, 0\} = 0$.

If $\Pi(y) = \max\{y, 0\}$, we now have $J_H(y) = A_{u^{max}(y)} + \rho A^+(y) \in K_N^o$, and (4.6.4) becomes

$$\begin{aligned}\big(A_{u^{max}(z^n)} + \rho A^+(z^n)\big)(z^{n+1} - z^n) = -\big(A_{u^{max}(z^n)}(z^n)\, z^n - b_{u^{max}(z^n)}\big)\\ - \rho A^+(z^n)z^n + \rho b^+(z^n),\end{aligned}$$

which is equivalent to

$$\big(A_{u^{max}(z^n)} + \rho A^+(z^n)\big)z^{n+1} = b_{u^{max}(z^n)} + \rho b^+(z^n). \tag{4.6.5}$$

From (4.6.5), it is easy to see that Algorithm 4.6.5 is well defined. The next theorem states that, given certain conditions on the matrix $\tilde{A}$, local quadratic convergence of Algorithm 4.6.5 to the solution of Problem 4.5.1 for $\Pi(\cdot) = \max\{\cdot, 0\}$ (see also Corollary 4.6.2) can be guaranteed.

Theorem 4.6.6. *Let $(z^n)_{n=0}^{\infty}$ be the sequence generated by Algorithm 4.6.5. There exists a constant $C > 0$ such that, for every starting value z^0, it is $\|z^n\|_\infty \le C$, $n \ge 1$. Furthermore, if $\tilde{A} = I_N$, where $I_N \in \mathbb{R}^{N\times N}$ denotes the identity matrix, then there exists a neighbourhood $\mathcal{B}$ of z_ρ such that, for any starting value $z^0 \in \mathcal{B}$, $(z^n)_{n\ge 0}$ remains in $\mathcal{B}$ and converges to the solution z_ρ at a quadratic rate.*

Proof. The boundedness of $(z^n)_{n\ge 1}$ follows from (4.6.5) by arguing as in the proof of Lemma 4.4.6. The quadratic local convergence property of $(z^n)_{n\ge 0}$ follows from Theorem 3.2 in [57] if H is semi-smooth and all $V \in \partial H(z_\rho)$ are non-singular, where ∂H denotes the generalised Jacobian as introduced in [18]. By the assumptions at the beginning of this section, H is continuously partially differentiable (and, in particular, semi-smooth) everywhere except on the set $\mathcal{D} := \{y \in \mathbb{R}^N : \exists\, i \in \mathcal{N} \text{ s.t. } y_i = (\tilde{b})_i\}$, since the only critical term is $\max\{\tilde{b} - \tilde{A}y, 0\} = \max\{\tilde{b} - y, 0\}$. Denoting the directional derivative at $y \in \mathbb{R}^N$ in direction $h \in \mathbb{R}^N$ by $H'(y; h)$, it can easily be verified that

$$\lim_{h\to 0} \frac{H'(y+h;h) - H'(y;h)}{\|h\|_\infty} = 0, \quad y \in \mathcal{D}, \tag{4.6.6}$$

which, by Theorem 2.3 in [57], gives semi-smoothness on $\mathcal{D}$, and, since the generalised Jacobian at $\tilde{b}$ is given by

$$\partial H(\tilde{b}) = \{A_{u^{max}(\tilde{b})} + \lambda \tilde{A} : \lambda \in [0,1]\} \subset K_N^o\,,$$

we see that all $V \in \partial H(\tilde{b})$ are in fact non-singular. □

Remark 4.6.7. *Since, in practice, Problem 4.5.1 frequently results from a time-stepping routine, the solution from the previous time step can be used as a starting value z^0, and, for small enough time-steps, $z^0 \in \mathcal{B}$ as required by Theorems 4.6.4 and 4.6.6 can be expected to hold.*

4.7 Evaluating the Continuous Control in Practice

If we solve Problems 4.2.1 and 4.2.2 using penalisation, we have to numerically deal with the continuous control, e.g. when computing $u_i^{min}(x^n)$ and $u_i^{max}(z^n)$ in (4.4.5) and (4.6.1) for given x^n and z^n, respectively. Now, if the control is well behaved (as in the examples following later in this chapter), the exact minimum/maximum can be found by differentiating and using analytical techniques; however, this is not necessarily always possible, i.e. we might have to approximate $(A_u)_{u\in\mathbf{U}}$ and $(b_u)_{u\in\mathbf{U}}$ by functions that are easier to handle numerically. The following remark states that such an approximation is legitimate within the framework of our algorithms.

Remark 4.7.1. *(Stability in the Control) Suppose we have families of functions $(A_u^\epsilon)_{u\in\mathbf{U}}$ and $(b_u^\epsilon)_{u\in\mathbf{U}}$, $\epsilon > 0$, satisfying the following properties.*

- *For every $\epsilon > 0$, $(A_u^\epsilon)_{u\in\mathbf{U}}$ and $(b_u^\epsilon)_{u\in\mathbf{U}}$ satisfy the same conditions as required in the definition of Problem 4.2.1 and in the setup of Sections 4.4 and 4.6.*

- *For every $u \in \mathbf{U}$, we have $\lim_{\epsilon\to 0} A_u^\epsilon = A_u$ and $\lim_{\epsilon\to 0} b_u^\epsilon = b_u$.*

Suppose we solve Problems 4.3.2 and 4.5.1 by using the approximations $(A_u^\epsilon)_{u\in\mathbf{U}}$ and $(b_u^\epsilon)_{u\in\mathbf{U}}$ for some $\epsilon > 0$, running any of the algorithms discussed in Sections 4.4 and 4.6, obtaining solutions $x^{*,\epsilon}$ and $z^{*,\epsilon}$, respectively. In the limit $\epsilon \to 0$, we have $x^{*,\epsilon} \to x^*$ and $z^{*,\epsilon} \to z^*$, where x^* and z^*, respectively, denote the solutions to Problems 4.3.2 and 4.5.1 for $(A_u)_{u\in\mathbf{U}}$ and $(b_u)_{u\in\mathbf{U}}$.

Proof. For fixed $\epsilon > 0$, all results from the previous sections hold, and it only remains to show that $x^{*,\epsilon} \to x^*$ and $z^{*,\epsilon} \to z^*$ as $\epsilon \to 0$. Since all involved functions are continuous on the compact interval $\mathbf{U}$, the convergence $\lim_{\epsilon\to 0} A_u^\epsilon = A_u$ and $\lim_{\epsilon\to 0} b_u^\epsilon = b_u$ is uniform in $u \in \mathbf{U}$; hence, we can reproduce results of Corollary 4.2.5 with constants independent of $\epsilon > 0$, which, following Lemmas 4.3.4 and 4.5.3, means that $x^{*,\epsilon}$ and $z^{*,\epsilon}$ are bounded independently of $\epsilon > 0$. We then infer the existence of converging subsequences, not notationally distinguished, of $(x^{*,\epsilon})_{\epsilon>0}$ and $(z^{*,\epsilon})_{\epsilon>0}$, respectively. Since the convergence in ϵ is uniform in $u \in \mathbf{U}$, we may swap "$\lim_{\epsilon\to 0}$" and "$\max_{u\in\mathbf{U}}$" in expressions (4.3.2) and (4.5.1), and we see that the limits of the sequences $(x^{*,\epsilon})_{\epsilon>0}$ and $(z^{*,\epsilon})_{\epsilon>0}$ do indeed solve Problems 4.3.2 and 4.5.1, respectively. The uniqueness of the solutions (see Lemmas 4.3.3 and 4.5.2) means that not only subsequences but the whole sequences converge. □

We point out that the rather general definition of approximating functions used in the above remark includes most common numerical approximations, e.g. piecewise constant, piecewise linear, and other finite subspace approximations.

4.8 Numerical Results

Finally, we come to test and analyse the applicability of the previously introduced numerical techniques by solving two models from mathematical finance, presented in Sections 4.8.1 and 4.8.2, which lead directly to equations as given in Problems 4.1.1 and 4.1.2, respectively.

4.8.1 Example: An Incomplete Market Investment Problem

In this section, we solve an incomplete market problem taken from [73]. More precisely, we look at an optimal investment model in which an agent has to distribute his money between a risk free bond and a risky stock; the market incompleteness arises from the stochastic volatility of the stock price process.

Let b, a, $\sigma : \mathbb{R} \to \mathbb{R}$ be bounded and globally Lipschitz functions, and suppose that there exists a constant c, independent of y, such that $\sigma(y) \geq c$. Let $T > 0$ be some finite time horizon. Let $\mu > r > 0$ and B_0 , S_0 , $Y_0 > 0$. Suppose we have a bond price process $(B_t)_{0\leq t\leq T}$, a stochastic volatility process $(Y_t)_{0\leq t\leq T}$, and a stock price

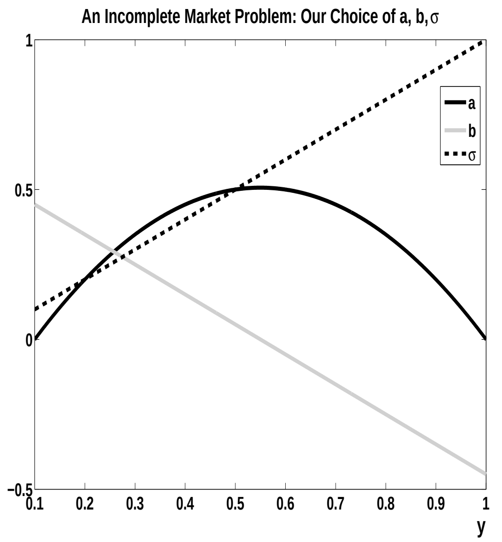

Figure 4.1: *The functions a, b and σ as chosen for the volatility process in the investment problem introduced in Section 4.8.1. As can easily be observed, whenever y approaches κ (or 1), $a(y)$ goes to zero and $b(y)$ is positive (negative); this guarantees that the diffusion $dY_t = b(Y_t)\,dt + a(Y_t)\,dW_t^1$ stays in $[\kappa, 1]$. Additionally, when (numerically) solving the two equations (4.8.2) and (4.8.3) on the interval $[\kappa, 1]$, outgoing characteristics (cf. [64]) replace our boundary conditions at κ and 1.*

process $(S_t)_{0\le t\le T}$ solving, respectively,

$$\begin{aligned} dB_t &= rB_t\,dt\,, \\ dY_t &= b(Y_t)\,dt + a(Y_t)\,dW_t^1\,, \\ \text{and} \quad dS_t &= \mu S_t\,dt + \sigma(Y_t)S_t\,dW_t^2\,, \end{aligned}$$

where $(W_t^i)_{0\le t\le T}$, $i \in \{1,2\}$, are two Brownian motions defined on a probability space $(\Omega, \mathcal{F}, P)$ with a correlation coefficient $\varrho \in [-1,1]$.

Staying exactly in the framework of [73], we consider an investor who can invest in the stock and in the bond. We suppose that the investor has initial wealth $X_0 \ge 0$ and that he may rebalance his portfolio $X_t = \pi_t^0 + \pi_t$ at any time $t \in [0,T]$; here, π_t^0 and π_t denote the amounts invested, respectively, in the bond and in the stock. The investor's wealth process solves

$$dX_t = rX_t\,dt + (\mu - r)\pi_t\,dt + \sigma(Y_t)\pi_t\,dW_t^1\,, \quad t \in [0,T],$$

and must satisfy $X_t \ge 0$ for every $t \in [0,T]$. We take the investor's utility function to be of CRRA-type and given by

$$U(x) = \frac{1}{\gamma}x^\gamma, \quad x \in \mathbb{R},$$

for some constant $0 < \gamma < 1$. Now, trying to maximise the final utility, the investor's value function is given by

$$\phi(x,y,t) := \sup_{\pi \in \mathcal{A}} \mathbb{E}[U(X_T)|X_t = x, Y_t = y], \quad (x,y,t) \in [0,\infty) \times \mathbb{R} \times [0,T], \quad (4.8.1)$$

where $\mathcal{A}$ is the set of admissible trading strategies (for details, see [73]). We cite the following result, which shows how this utility maximisation problem can be solved.

Proposition 4.8.1. *The value function introduced in* (4.8.1) *can be written as*

$$\phi(x,y,t) = \frac{x^\gamma}{\gamma}\varphi(y,t), \quad (x,y,t) \in [0,\infty) \times \mathbb{R} \times [0,T],$$

where φ solves

$$\frac{1}{\gamma}\big[\varphi_t + \frac{1}{2}a^2(y)\varphi_{yy} + b(y)\varphi_y\big] + r\varphi \\ + \max_{u \in \mathbf{U}}\big[\frac{1}{2}(\gamma-1)\sigma^2(y)u^2\varphi + \varrho\sigma(y)a(y)u\varphi_y + (\mu - r)u\varphi\big] = 0, \quad (4.8.2)$$

with $\varphi(y,T) \equiv 1$ and $\mathbf{U} \subset \mathbb{R}$ an appropriately chosen compact set, and $\tilde{\varphi}(\cdot,\cdot) := \varphi(\cdot,\cdot)^{\frac{1-\gamma+\varrho^2\gamma}{1-\gamma}}$ satisfies

$$\tilde{\varphi}_t + \frac{1}{2}a(y)^2\tilde{\varphi}_{yy} + \Big[b(y) + \varrho\frac{\gamma(\mu-r)a(y)}{(1-\gamma)\sigma(y)}\Big]\tilde{\varphi}_y \\ + \frac{\gamma(1-\gamma+\varrho^2\gamma)}{1-\gamma}\Big[r + \frac{(\mu-r)^2}{2\sigma^2(a)(1-\gamma)}\Big]\tilde{\varphi} = 0. \quad (4.8.3)$$

The notion of solution used in this context has to be understood in the viscosity sense (cf. [40]), and φ, $\tilde{\varphi}$ are the unique solutions to (4.8.2) *and* (4.8.3)*, respectively.*

Proof. The main result can be found in [73]. See [8] for details on the existence of the viscosity solutions. □

In (4.8.2), we make the assumption of $\mathbf{U}$ being a compact set since, for every time $t \in [0,T)$, the maximum in (4.8.2) is assumed at $u = \pi_t^*$, where $(\pi_t^*)_{0 \le t \le T}$ is the investor's optimal trading policy (cf. [73]), which should not reach infinity in a meaningful financial model.

Clearly, from Theorem 4.8.1, to find ϕ, we need to compute φ or $\tilde{\varphi}$ and solve either equation (4.8.2) or (4.8.3); we have deliberately chosen a problem which can be linearised such that we can obtain a reference solution by standard methods. In the next few sections, we will select parameters and present and compare several approaches of computing ϕ.

4.8.1.1 Choosing Model Parameters and Functions

We set $r = 0.3$, $\mu = 0.7$, $\varrho = -0.2$, $\gamma = 0.5$, and $T = 1$. Furthermore, we introduce $y_{min} := \kappa := 0.1$ and $y_{max} := 1$, and, for $y \in [y_{min}, y_{max}]$, we use

$$\begin{aligned} a(y) &= -\,2.5(y - 0.5 - 0.5\kappa)^2 + 2.5(-0.5 + 0.5\kappa)^2, \\ b(y) &= -\,y + 0.55, \\ \text{and} \quad \sigma(y) &= y. \end{aligned}$$

Functions a, b and σ are shown in Figure 4.1; they satisfy the technical conditions listed in the previous section and guarantee $(Y_t)_{t\geq 0} \subset [\kappa, 1]$.

4.8.1.2 Discretisation of the Continuous Equations

Numerically, we solve equations (4.8.2) and (4.8.3) backwards in time, starting at expiry T, on a grid $\{(ih + \kappa, jk) : 0 \leq i \leq N,\ 0 \leq j \leq M\}$, where $h := (1 - \kappa)/N$ and $k := T/M$. We perform a fully implicit finite difference discretisation, using one-sided differences for all first derivatives (including the time derivative) and central differences for all second derivatives. (For a general overview of basic finite difference concepts for PDEs, e.g. see [60].) In particular, when discretising $\mathfrak{d}\varphi_y$ (or $\mathfrak{d}\tilde{\varphi}_y$), where $\mathfrak{d} = \mathfrak{d}(y, u)$ denotes the combined coefficient of the first y-derivative in (4.8.3) (or (4.8.2)), we switch between left-sided and right-sided differences in accordance with a positive or negative sign of the coefficient $\mathfrak{d}$; this way, we can guarantee the following two properties.

- The tridiagonal discretisation matrix implied by our fully implicit scheme has positive entries on the diagonal and non-positive entries on the upper/lower diagonals.
- Since $a(\kappa) = a(1) = 0$ and $b(\kappa) > 0 > b(1)$, the boundary conditions at κ and 1 are replaced by finite differences pointing inwards. (cf. [64]).

Proceeding as just described, the linear parabolic PDE in (4.8.3) is approximated by a simple linear system of equations. Furthermore, taking $\mathbf{U}$ to be $[-l, l]$, with $l = 150$, our discretisation of equation (4.8.2) matches Problem 4.2.1, with (4.2.1) and (4.2.2) satisfying all assumptions.

Remark 4.8.2. *The choice of backward/forward differencing in accordance with a positive or negative sign of the coefficient $\mathfrak{d}$ may sometimes – even though the problem does not arise for the examples considered in this thesis – result in $u \mapsto A_u$ being a continuous but non-smooth function, which, in turn, will affect some of the theoretical results in Sections 4.4 and 4.6.*

Remark 4.8.3. *Based on our choice of functions a and b (cf. Figure 4.1), the diffusion $(Y_t)_{0\leq t\leq T}$ will always stay in $[\kappa, 1]$. Conceptually, this means that no 'outside' information – like Dirichlet boundary conditions – is required for the PDE to be completely specified on the interval $[\kappa, 1]$, since the flow of information can be thought* *of as coming from 'within'. Mathematically, this means that, by taking $a(y) := 0$, $y \in \mathbb{R}\backslash[\kappa, 1]$, and using one-sided inwards pointing finite difference stencils at κ and 1, we obtain a consistent (and monotone) discretisation scheme without requiring any Dirichlet boundary conditions.*

Remark 4.8.4. *Convergence of the finite difference approximation to the unique viscosity solution can be shown as described in Chapter 1 and Section 3.5.1. For the current example, consistency is straightforward to prove (e.g. cf. Lemma 3.5.6), stability can be shown following [28] since we have what they call a "positive coefficient discretisation", and a strong comparison result can be found in [8]; hence, altogether, the results of [6] are applicable, and convergence to the unique viscosity solutions of* (4.8.3) *and* (4.8.2) *can be guaranteed.*

4.8.1.3 Solution of the Discrete Systems

In this section, we will compare the following three ways of numerically solving the incomplete market problem represented by the equations in Theorem 4.8.1. All computations are done in Matlab.

- We solve the linear system of equations resulting – for every time step – from a discretisation of the linear parabolic PDE in (4.8.3).
- We solve Problem 4.2.1 (corresponding to one time step of a fully implicit discretisation of the non-linear parabolic PDE in (4.8.2)) by
  - the penalty method devised in Sections 4.3 and 4.4 of this paper, and by
  - the method of policy iteration.

The use of policy iteration has been studied in [28, 50] and has been briefly summarised in Section 1.2.3. Following Section 4.7, we approximate $\mathbf{U} = [-l, l] = [-150, 150]$ by $\widetilde{\mathbf{U}} := \{-150 + \frac{3r}{10} : r = 0, 1, \ldots, 1000\}$, thereby discretising $\mathbf{U}$ by using a very fine grid of 1001 points. (Effectively, in the notation of Remark 4.7.1, we approximate $(A_u)_{u \in \mathbf{U}}$ and $(b_u)_{u \in \mathbf{U}}$ by piecewise constant functions.) Numerically, when computing candidate solutions to equations (4.2.3) and (4.3.2) by policy iteration and Algorithm 4.4.4, respectively, we terminate the iterations according to the following two checks of accuracy, using tol =1e-08.

- For (4.2.3), we terminate if our candidate solution $x^n \in \mathbb{R}^N$ satisfies

$$\frac{\|A_{\tilde{u}^{min}(x^n)}\, x^n - b_{\tilde{u}^{min}(x^n)}\|_\infty}{\|b_{\tilde{u}^{min}(x^n)}\|_\infty} \leq tol, \tag{4.8.4}$$

  where $\tilde{u}^{min}(x^n)$ satisfies $A_{\tilde{u}^{min}(x^n)}\, x^n - b_{\tilde{u}^{min}(x^n)} = \min\{A_{\tilde{u}}\, x^n - b_{\tilde{u}} : \tilde{u} \in \widetilde{\mathbf{U}}\}$.

- For (4.3.2), we terminate if our candidate solution $x^n_\rho \in \mathbb{R}^N$ satisfies

$$\frac{\|(A_{u_0}\, x^n_\rho - b_{u_0}) - \rho \max\{b_{\tilde{u}^{min}_\rho(x^n_\rho)} - A_{\tilde{u}^{min}_\rho(x^n_\rho)}\, x^n_\rho\, , 0\}\|_\infty}{\|b_{u_0} + \rho\, b^+_{\tilde{u}^{min}_\rho(x^n_\rho)}\|_\infty} \leq tol, \tag{4.8.5}$$

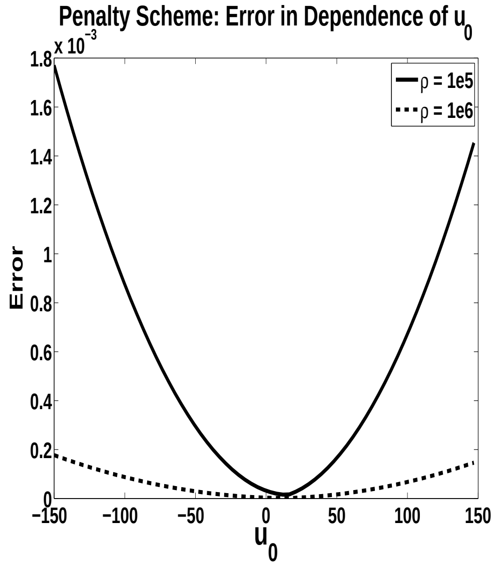

Figure 4.2: *The incomplete market investment problem of Section 4.8.1. For $M = N = 200$ and different choices of u_0, we see the difference between the solution computed by the penalty scheme and the policy iteration method. The error is measured in the max-norm. We observe that, for the quality of the penalty approximation, it is advantageous to pick u_0 close to the optimal control value.*

where $\tilde{u}_\rho^{min}(x^n)$ satisfies $b_{\tilde{u}_\rho^{min}(x^n)} - A_{\tilde{u}_\rho^{min}(x^n)}\, x_\rho^n = \max\{b_{\tilde{u}_\rho} - A_{\tilde{u}_\rho}\, x_\rho^n : \tilde{u}_\rho \in \widetilde{\mathbf{U}}\}$ and

$$b^+_{\tilde{u}_\rho^{min}(x_\rho^n)} := \begin{cases} b_{\tilde{u}_\rho^{min}(x_\rho^n)} & \text{if } b_{\tilde{u}_\rho^{min}(x^n)} - A_{\tilde{u}_\rho^{min}(x^n)}\, x_\rho^n > 0, \\ 0 & \text{else.} \end{cases}$$

We use the numerical solution of the linear parabolic PDE (4.8.3) as a *reference solution* for the incomplete market problem. For $M = N = 200$ and $\rho = 1e06$, measured in the maximum norm, the difference between the numerical solution of (4.8.3) and the penalty approximation (4.3.2) is 2e-03, and the difference between the penalty approximation (4.3.2) and the policy iteration solution of (4.2.3) is 2e-04.

Figure 4.2 shows the impact of the a priori unspecified parameter u_0 in Problem 4.3.2 on the penalisation error. We first remark that, if u_0 could be chosen to be the optimal control of the discretised problem, the penalisation error in Problem 4.3.2 would be identical to zero by construction. Now, clearly, the optimal control is unknown, and generally a function of the state variable and time, and, therefore, a constant value u_0 generally gives a non-zero (in fact, negative) penalisation error, which we know to converge to zero of first order in $1/\rho$. This is the underlying convergence mechanism of the proposed method and does not require a cunning choice of u_0. Figure 4.2 does show, however, that the penalisation error for fixed ρ can be reduced by diligent choice of u_0. It is thereby sufficient to choose u_0 of the same order of magnitude as the optimal control. One can thus take advantage of a priori knowledge

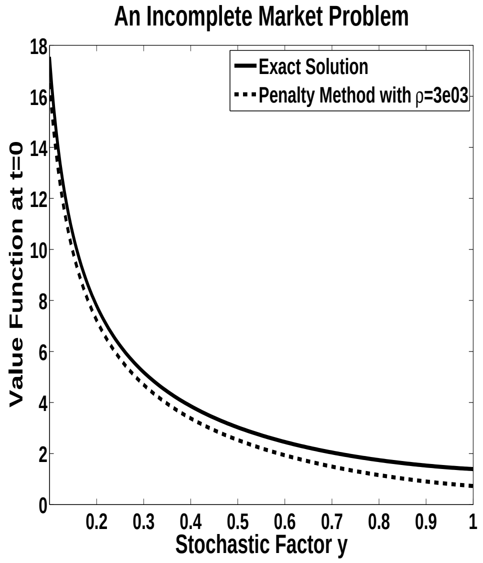

Figure 4.3: *The incomplete market investment problem of Section 4.8.1. For* $M = N = 200$*, we see the solution to* (4.8.3)*, referred to as "reference solution" since it avoids the treatment of non-linearities, and the solution to the penalised equation* (4.3.2) *for* $\rho = 1e03$*; for the current choice of* ρ*, the penalty approximation is still coarse.*

of the approximate control size. If such an estimate is not available, one might first determine a rough approximation by producing a crude version of Figure 4.2 on a coarse mesh – coarse in parameter space as well as time and state space – which is computationally cheap, and pick u_0 accordingly. We do not take advantage of this information in the following computations, and, throughout, we use $u_0 = -l = -150$, which appears to be the worst-case choice. We also tested the impact of u_0 on the Newton method, and found the required number of iterations virtually unaffected in all settings.

The results of our numerical tests are summarised in Figures 4.3 and 4.4 and in Tables 4.1 and 4.2.

In Figure 4.3, we see the penalty approximation for $\rho = 1e03$. Given that the penalty parameter is still relatively small, the penalty approximation is still below the reference solution, but, clearly, the curves are similarly shaped already. In Figure 4.4, we measure rate of the convergence in ρ; more precisely, for different sizes of ρ, starting out with the $t = k$ value of the reference solution (i.e. the value at the penultimate time step), we compute a time zero value using the penalty scheme (thus solving exactly one discrete LCP) and compare it to the time zero value of the reference solution. As expected, based on Theorem 4.3.7, the convergence in ρ is of first order. In Table 4.1, we see the number of iterations needed by Algorithm 4.4.4 and policy iteration, averaged over all time steps; for the two schemes, the numbers are almost identical, and – in both cases – we never need more than two iterations to reach the

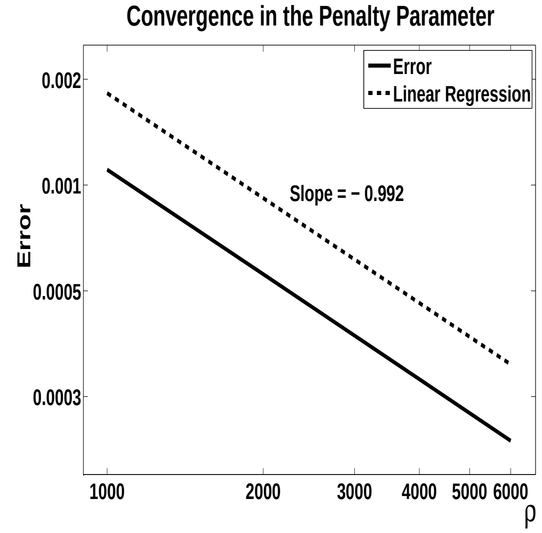

Figure 4.4: *The incomplete market investment problem of Section 4.8.1. For M=N=200, we measure the speed of convergence in the penalty parameter ρ. The error is measured in the max-norm. The plot is log-log, and we observe a convergence rate of 0.992, close to one, confirming the results of Theorem 4.3.7.*

desired accuracy *tol*. Finally, in Table 4.2, we see the computation times for the two schemes, which, as is to be expected based on the iteration numbers, are virtually the same.

4.8.2 Example: Early Exercise Options in an Incomplete Market

In this section, we value an early exercise contract in an incomplete market, in which the incompleteness stems from the fact that the asset on which the early exercise contracts are written is not traded; the example in its entirety is taken from [54], and we use it to demonstrate that the arising non-linear equations can be solved by a fully implicit finite difference scheme when using the results of Section 4.5.

Let b, $a : \mathbb{R} \to \mathbb{R}$ be bounded and globally Lipschitz functions. Let $T > 0$ be some finite time horizon. Let μ, $\sigma > 0$ and $S_0 > 0$, $Y_0 \in \mathbb{R}$. Suppose we have a traded asset price process $(S_t)_{0 \le t \le T}$ and a non-traded asset price process $(Y_t)_{0 \le t \le T}$ solving, respectively,

$$\begin{aligned} dS_t &= \mu S_t \, dt + \sigma S_t \, dW_t^1 \\ \text{and} \quad dY_t &= b(Y_t) \, dt + a(Y_t) \, dW_t^2 \, , \end{aligned}$$

where $(W_t^i)_{0 \le t \le T}$, $i \in \{1, 2\}$, are two Brownian motions defined on a probability space $(\Omega, \mathcal{F}, P)$ with a correlation coefficient $\varrho \in [-1, 1]$. Additionally, we assume the existence of a riskless bond with interest rate $r = 0$.

Policy Iteration	$n=1$	$n=2$
M, $N=50$	6%	94%
M, $N=200$	11%	89%
$M=200$, $N=50$	53%	47%
$M=50$, $N=200$	-	100%
Penalty Method ($\rho=4e03$)	$n=1$	$n=2$
M, $N=50$	8%	92%
M, $N=200$	13%	87%
$M=200$, $N=50$	49.5%	50.5%
$M=50$, $N=200$	-	100%
Penalty Method ($\rho=1e06$)	$n=1$	$n=2$
M, $N=50$	6%	94%
M, $N=200$	11%	89%
$M=200$, $N=50$	55%	45%
$M=50$, $N=200$	-	100%

Table 4.1: *The incomplete market investment problem of Section 4.8.1. For different time and space grids, we see the number of iterations needed by penalty approximation – or, more precisely, by Algorithm 4.4.4 when solving* (4.3.2) *– and policy iteration. For the different schemes, the numbers are very similar, and, generally, we never need more than two steps.*

Similar to the previous example, we consider an investor who can invest in the traded asset and in the bond. We suppose that the investor has initial wealth $X_0 \in \mathbb{R}$ and that he may rebalance his portfolio $X_t = \pi_t^0 + \pi_t$ at any time $t \in [0,T]$; here, π_t^0 and π_t denote the amounts invested, respectively, in the bond and in the stock. The investor's wealth process solves

$$dX_t = \mu\pi_t\,dt + \sigma\pi_t\,dW_t^1\,, \quad t \in [0,T].$$

We take the investor's utility function to be of exponential type and given by

$$U(x) = -e^{-\gamma x}, \quad x \in \mathbb{R},$$

with risk aversion parameter $\gamma > 0$. Now, we suppose the investor holds an early exercise contract with payoff $P(y)$, $y \in \mathbb{R}$, on the non-traded asset, and we would like to find the indifference price (cf. [54]) of the instrument; we cite the following result.

Proposition 4.8.5. *The buyer's early exercise indifference price $\psi(y,t)$, where $(y,t) \in \mathbb{R}\times[0,T]$, is the unique bounded viscosity solution to*

$$\min\Big\{-\psi_t - \mathcal{L}^b\psi + \frac{1}{2}\gamma(1-\varrho^2)a^2(y)\psi_y^2\,, \psi - P(y)\Big\} = 0, \qquad (4.8.6)$$

where $\psi(\cdot,T) = P(\cdot)$ and

$$\mathcal{L}^b\psi := \frac{1}{2}a^2(y)\psi_{yy} + \big(b(y) - \varrho\frac{\mu}{\sigma}a(y)\big)\psi_y\,.$$

Grid Size	Policy	Penalty ($\rho = 4e03$)	Penalty ($\rho = 1e06$)
M, $N = 50$	$1.08s$	$1.08s$	$1.09s$
M, $N = 200$	$35.07s$	$34.76s$	$35.16s$
$M = 200$, $N = 50$	$2.74s$	$2.82s$	$2.76s$
$M = 50$, $N = 200$	$10.89s$	$10.88s$	$10.91s$

Table 4.2: *The incomplete market investment problem of Section 4.8.1. For different time and space grids, we see the computation times needed by penalty approximation and policy iteration. In all cases, the computational effort of the two schemes is very similar.*

Proof. The main result can be found in [54]. See [8] for details on the existence of the viscosity solution. □

It can easily be shown that $\psi_y^2 = \max_{u \in \mathbb{R}}\{2u\psi_y - u^2\}$, and, if we assume ψ_y to be bounded, (4.8.6) can be rewritten as

$$\min\big\{\max\{\mathcal{L}_u^b \psi : u \in \mathbf{U}\}, \psi - P(y)\big\} = 0, \tag{4.8.7}$$

where $\mathbf{U} \subset \mathbb{R}$ is a suitably chosen compact set and, for $u \in \mathbf{U}$, we define

$$\mathcal{L}_u^b \psi := -\psi_t - \frac{1}{2}a^2(y)\psi_{yy} - \big(b(y) - \varrho\frac{\mu}{\sigma}a(y)\big)\psi_y + \frac{1}{2}\gamma(1-\varrho^2)a^2(y)(2u\psi_y - u^2).$$

Hence, to compute the early exercise indifference price of the considered option, we have to solve (4.8.7), which has the same structure as (4.1.2).

4.8.2.1 Choosing Model Parameters and Functions

We set $\mu/\sigma = 1$, $\varrho = 0.1$, $\gamma = 1$ and $T = 1$. Furthermore, we introduce $y_{min} := 0$ and $y_{max} := 5$, and, for $y \in [y_{min}, y_{max}]$, we use

$$\begin{aligned} a(y) &= y, \\ b(y) &= 0.3y, \\ \text{and} \quad P(y) &= \max\{1-y, 0\}. \end{aligned}$$

In particular, the choice of $P(\cdot)$ means that we are dealing with an American put with strike one.

4.8.2.2 Discretisation of the Continuous Equations and Solution of the Discrete Systems

Numerically, we solve (4.8.6) and (4.8.7) similarly to Section 4.8.1.2, i.e. we proceed backwards in time, starting at expiry T, on a grid $\{(ih, jk) : 0 \le i \le N,\ 0 \le j \le M\}$, where $h := 5/N$ and $k := T/M$. In space, we again apply a finite difference discretisation guaranteeing for the discretisation matrices to be in K_N^o. Since we are dealing with a put option, we use $\psi(0) = 1$ and $\psi(5) = 0$ as boundary conditions, and we take the set $\mathbf{U}$ in (4.8.7) to be $[-1, 0]$. We compare the following three numerical approaches in Matlab.

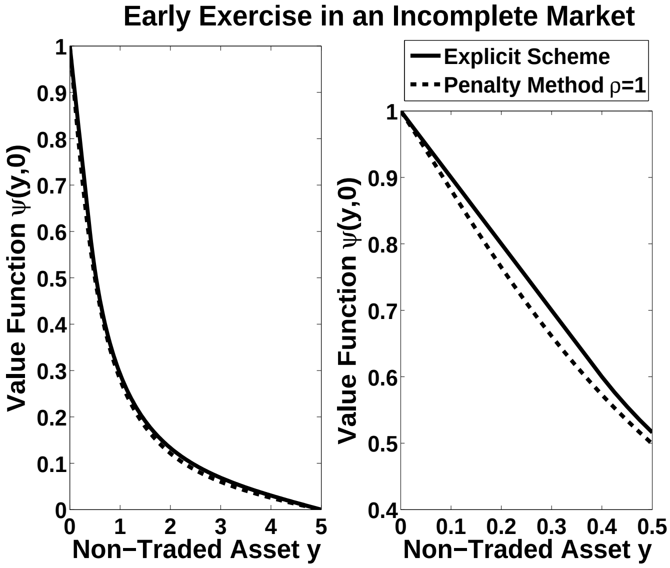

Figure 4.5: *The incomplete market early exercise pricing problem of Section 4.8.2. We see the penalty approximation (for $M = N = 200$ and $\rho = 1$) and the solution of the explicit scheme (for $M =$ 4e04 and $N = 200$). Even though the penalty parameter is still very small, the penalty solution already seems to be a reasonable approximation.*

- For (4.8.6), we use an explicit time stepping scheme, meaning all non-linearities can be dealt with easily.
- We solve Problem 4.2.2 (corresponding to one time step of a fully implicit discretisation of the non-linear parabolic PDE in (4.8.7)) by
  - the penalty method devised in Sections 4.5 and 4.6 of this paper, and by
  - the method of policy iteration (see Section 1.2.4 or [50]).

As already in Section 4.8.1.3, we employ Remark 4.7.1 and approximate $\mathbf{U} = [-1, 0]$ by $\widetilde{\mathbf{U}} := \{-1 + \frac{r}{101} : r = 0, 1, \ldots 101\}$. When solving the penalised equation (4.5.1) by Algorithm 4.6.5, we use a test for accuracy of the kind (4.8.5), setting *tol* =1e-08 as before. Similarly, we use a test for accuracy of the kind (4.8.4) with the same tolerance when solving an equation of the form (4.2.4) by policy iteration.

Remark 4.8.6. *As already in Remark 4.8.4, convergence of the fully implicit discretisation of* (4.8.7) *to the unique viscosity solution can be guaranteed by noting that we have stability, monotonicity, and consistency of the discretisation, and by using a strong comparison principle (cf. [8]). The fully explicit discretisation of* (4.8.6) *converges similarly provided we have stability.*

In our numerical tests, we find the explicit scheme to require a relatively large number of time steps for stability, making it difficult to use. In Table 4.3, fixing ρ =1e06, we see the difference between the explicit scheme and the penalty approximation

Explicit Scheme	*Time*	*Penalty* ($\rho = 1e06$)	*Time*	*Difference*
$M = 2500$, $N = 50$	0.78s	$M = 50$, $N = 50$	0.23s	1.3e-03
$M = 4e04$, $N = 200$	16.49s	$M = 200$, $N = 200$	3.71s	3.5e-04

Table 4.3: *The incomplete market early exercise pricing problem of Section 4.8.2. For different time and space grids, we see the difference between the explicit scheme and the penalty method, and the respective runtimes. The explicit scheme runs much longer due to the high number of time steps needed to guarantee stability for a given space discretisation. Furthermore, since our space discretisation contains one-sided differences for reasons of monotonicity, the expected consistency order is $O(1/M) + O(1/N)$, which also makes the choice $M = N$ desirable.*

for different grid sizes; for the explicit scheme, for a given space discretisation, we always choose the number of time steps such that the solution plot does not show any instabilities, whereas for the penalty scheme we take identical numbers of time and space steps. For $N = 50$ and $N = 200$, the explicit and the penalty scheme differ by 1.5e-03 and 3.6e-04 in the max-norm, respectively (cf. Table 4.3). We point out that the explicit scheme runs substantially longer due to the high number of time steps required for stability; picking the time and space steps proportional to each other for the fully implicit scheme is optimal experimentally because of the observed first order convergence in both time and space. In Figure 4.5, we see the penalty approximation for $\rho = 1$ and the explicit solution; even though the penalty parameter is very small, the two graphs are extremely close. The difference in the max-norm between the policy iteration and penalty approximation solutions is 1.6165e-05 and 2.6011e-05 for grid sizes $M = N = 50$ and $M = N = 200$, respectively. In Table 4.4, for different grid sizes and penalty parameters, we see the maximum and the average number of iterations needed by Algorithms 4.6.5 (penalty approximation) and 1.2.7 (nested policy iteration) for solving the discrete systems at every time step, as well as the corresponding runtimes for the full schemes. (In case of Algorithm 1.2.7, we count one iteration whenever line (1.2.5) is executed.) Throughout, the average numbers of iterations are small, and both schemes run very fast, with the penalty scheme being faster by about a factor two; the effect appears to be due to the fact that – whilst the average number of iterations is small – policy iteration requires a large number of iterations in a few instances (as seen by the $Max\ Iterations$ in Table 4.4); we will investigate this effect more closely below. Finally, in Figure 4.6, we measure the rate of convergence in ρ, and confirm first order convergence as predicted by Theorem 4.5.6; the implementation is precisely as in Section 4.8.1, except that we use a $\rho = 1e08$ penalty approximation as reference solution; as before, the error is measured in the max-norm.

4.8.2.3 Sensitivity with Respect to the Initial Guess

We have seen above that, unlike penalty approximation, policy iteration requires many iterations in a few instances and that this effect appears to correlate with the grid size N (cf. Table 4.4). To investigate if the phenomenon relates to the quality of the initial guess for the non-linear iterations, we set $M = 1$ and consider different grids sizes N; the results can be seen in Figure 4.7. (Since, in our implementations, we use the solution from the previous time step as initial guess for the next, setting $M = 1$ can be interpreted as solving a single non-linear discrete system with a poor initial guess.) Clearly, the penalty approximation is almost unaffected by the increase of N, whereas the number of iterations of the policy iterations grows linearly in N. (In [50], it has already been observed that even for a simple American option problem, the best obtainable bound on the number of iterations of policy iteration is linear, i.e. $O(N)$.)

Now, it is easy to see that equation (4.2.4) is scalable, i.e., for any $\delta > 0$, it can equivalently be rewritten as

$$\min\left\{\max_{u\in\mathbf{U}}\{A_u\, z - b_u\}, \delta(\tilde{A}z - \tilde{b})\right\} = 0,$$

and it has been pointed out in [31, 33, 32] that a different choice of δ will generally lead to different policy iterations; more precisely, in our context, depending on the choice of δ, we obtain different adaptations of Algorithm 1.2.7, all converging to the

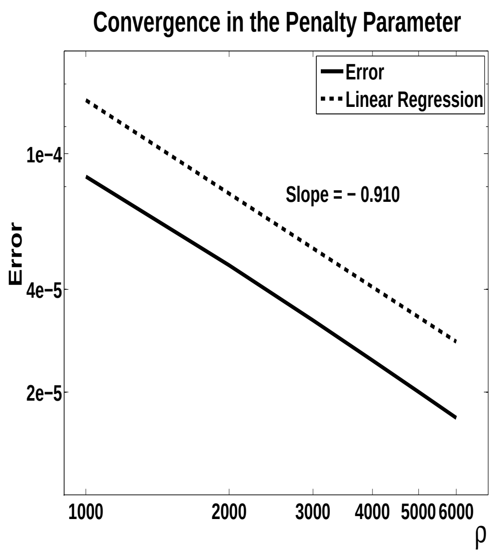

Figure 4.6: *The incomplete market early exercise pricing problem of Section 4.8.2. For M=N=200, we measure the speed of convergence in the penalty parameter ρ. The error is measured in the max-norm. The plot is log-log, and we observe a convergence rate of 0.910, close to one, confirming the results of Theorem 4.5.6.*

Policy Iteration	Max Iterations	∅ Iterations	Runtime
$M, N = 50$	4	2.20	0.38s
$M, N = 200$	11	2.17	6.42s
$M = 200$, $N = 50$	4	1.83	0.86s
$M = 50$, $N = 200$	18	2.88	2.12s
Penalty Method ($\rho = 4e03$)	Max Iterations	∅ Iterations	Runtime
$M, N = 50$	3	1.98	0.25s
$M, N = 200$	3	1.21	4.09s
$M = 200$, $N = 50$	3	1.15	0.66s
$M = 50$, $N = 200$	4	2.16	1.47s
Penalty Method ($\rho = 1e06$)	Max Iterations	∅ Iterations	Runtime
$M, N = 50$	2	1.10	0.17s
$M, N = 200$	3	1.08	3.82s
$M = 200$, $N = 50$	2	1.02	0.61s
$M = 50$, $N = 200$	4	1.38	1.13s

Table 4.4: *The incomplete market early exercise pricing problem of Section 4.8.2. For different time and space grids, we see the number of iterations needed by Algorithm 4.6.5 when solving the penalised equation* (4.5.1)*. Independently of the grid size, the absolute and the average number of required iterations is small.*

same solution. For theoretical considerations on the best choice of δ, we refer to [33]; numerically, we find the following.

- Simply introducing a scaling factor δ does not yield an improvement.
- Changing the initial guess from the payoff $P(\cdot)$ to $z^0 \equiv 1$ does not yield an improvement.
- Using a scaling factor $\delta = 1e06$, combined with initial guess $z^0 \equiv 1$, significantly reduces the number of iterations needed (cf. Figure 4.7).

Further analysis shows that a large scaling factor δ yields an improvement whenever the initial guess is such that $\tilde{A}z^0 - \tilde{b} > 0$, which is necessary for the multiplication by the scaling factor to have an effect.

In summary, we can conclude that the penalty approximation appears to have a generic advantage when dealing with poor starting values, whereas – to obtain equally good results by policy iteration – prudent implementation is inevitable. The main reason for the different performance of policy iteration seems to be that it does *not* show Newton-type behaviour, i.e. it does not converge to the solution in steps with rapidly decreasing size; in particular, when using the payoff as initial guess, it shifts the solution upwards node by node until the free boundary is found, resulting in the linear dependence on N observed in Figure 4.7.

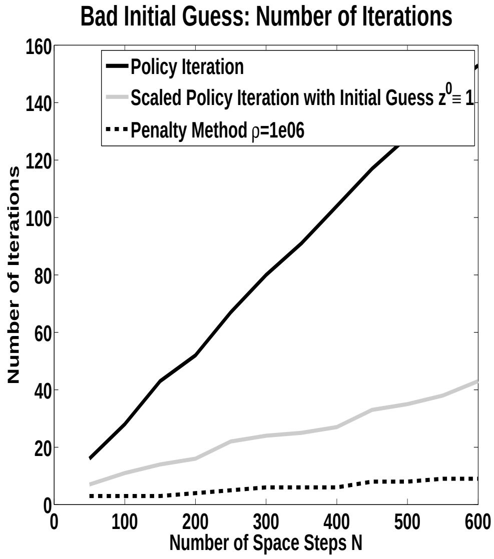

Figure 4.7: *The incomplete market early exercise pricing problem of Section 4.8.2. For $M = 1$ and varying N, we measure the number of non-linear iterations required by (scaled) policy iteration and penalty approximation, respectively.*

4.9 Conclusion

In this chapter, we have considered the numerical solution of continuously controlled HJB equations and HJB obstacle problems – which trivially includes finitely controlled equations as solved in Chapter 3 – and we show that penalisation is again a powerful means for solving the non-linear discrete problems resulting from fully implicit finite difference discretisations. More precisely, when applying a monotone and stable finite difference scheme to the considered non-linear equations, convergence to the unique viscosity solution can be guaranteed (cf. Chapter 1 and Section 3.5.1), and, if implicit time stepping is used to secure stability, one is left with the tricky task of solving the resulting non-linear discrete systems. Generally, this can be done by policy iteration or value iteration (cf. [28, 50]) or – as we show – by penalisation combined with a Newton-like iteration. For both problems, continuously controlled HJB equation and HJB obstacle problem, we show that the accuracy achieved by penalisation is $O(1/\rho)$, where ρ is the penalty parameter; i.e., we have the same approximation accuracy as obtained in the simpler setting of Chapter 3. We include numerical examples from (early exercise) incomplete market pricing, demonstrating the competitiveness of our algorithms as fast and easy-to-use numerical schemes.

Chapter 5

Conclusion

Finally, we come to discuss and link the different results presented in the previous chapters. We summarise present achievements, while at the same time outlining current limitations of our work and detailing plans of overcoming these.

5.1 This Thesis

In this thesis, we have introduced and studied numerical algorithms for the solution of discrete Hamilton-Jacobi-Bellman (HJB) equations. More precisely, we have contributed to the understanding of policy iteration in the context of American option pricing (cf. Chapter 2), and we have devised different penalty approaches for the solution of more general HJB equations (cf. Chapters 3 and 4).

In all cases, we have mathematically proved the convergence of our schemes, and we have numerically established competitiveness compared to state-of-the-art methods.

We believe that, at this stage, we have achieved several things.

- By studying policy iteration for the case of the discretised American option linear complementarity problem (cf. Chapter 2) and proving that, in standard situations, it has complexity $O(MN)$, which is the same as for European options, we have made a strong case for the use of policy iteration in American option pricing. We have shown that, numerically, policy iteration hugely outperforms PSOR, the current standard approach.

- By introducing penalty methods for discretised HJB equations (cf. Chapters 3 and 4) and carefully benchmarking them against policy iteration methods, we have demonstrated the usefulness of penalisation as a numerical tool in the context of certain non-linear equations. Analytically as well as numerically, our suggested penalty schemes are on par with policy iteration (the current standard approach), and, thus, are found to be extremely competitive.

More precisely, in most standard situations where discretised HJB equations result from a time stepping routine (with sufficiently small time steps), policy iteration and penalisation perform almost identically (cf. Sections 2.3, 3.6, and 4.8.1), with policy iteration occasionally having a small edge. However, unlike penalisation, policy iteration is sensitive to a bad initial guess (cf. Sections 2.3 and 4.8.2.3), and, in the situation of the HJB obstacle problem, penalisation improves over policy iteration even when running it as part of a time stepping routine (cf. Section 4.8.2); in the latter case, this can probably be attributed to the fact that, even for a small time step, the payoff is a rather coarse approximation of the solution to the incomplete market early exercise price before expiry, driving up the number of policy iterations in the very first step of the numerical scheme.

- By computing several examples and discussing their numerical implementation in detail (cf. Sections 2.3, 3.5, and 4.8), we have raised awareness for the availability of rigorous, efficient, and easy-to-use numerical schemes for the solution of HJB equations in mathematical finance, which includes our own as well as competing methods. Since HJB equations arise numerously in financial models, a thorough understanding of their numerical solution is important and adds to the practical applicability of the models.

5.2 Future Work

In this final section, we briefly discuss our ideas for extending our work to date.

5.2.1 Penalty Approximation of Isaacs Equations

Isaacs equations appear in the theory of differential games (cf. [39]), where two or more competing players are considered; the resulting equations contain several controls, one for each player, and a combination of 'min' and 'max' operators, representing their conflicting goals.

To date, the nested policy iteration, summarised in Section 1.2.4, is the only existing algorithm for the numerical solution of (1.1.8); hence, it would be desirable to devise a competing method as a benchmark, possibly a penalisation-based one.

Making similar assumptions to the ones in Chapter 4, we can prove that

$$\max_{u\in\mathbf{U}}\left\{A_{u,v_0}\,x_\rho - b_{u,v_0}\right\} - \rho \max_{(u,v)\in\mathbf{U}\times\mathbf{V}}\Big\{\max\left\{b_{u,v} - A_{u,v}\,x_\rho\,,0\right\}\Big\} = 0 \qquad (5.2.1)$$

approximates (1.1.8) to first order in the penalty parameter ρ, for any $v_0 \in \mathbf{V}$. Unfortunately, at this stage, we have not yet been able to devise an iterative algorithm solving the penalised equation (5.2.1), and the techniques applied in Chapters 3 and 4 cannot directly be transferred. However, at the very least locally, a Newton-based approach applied to (5.2.1) should be successful in many cases, and, based on results from Chapter 4, we will study the required properties.

An alternative route would be to replace the first term in (5.2.1) by a second penalty approximation (based on Chapter 4), which would still result in a combined penalty approximation of equation (1.1.8), using two penalty terms; however, again, the viability of such an approach depends on the solvability of the resulting approximation by iterative methods, and has to be tested carefully.

5.2.2 Monotone Discretisation of Cross Derivatives

All of the numerical methods discussed in the previous chapters heavily rely on monotone discretisations; this is important for the convergence to the unique viscosity solution, and it also ties in closely with the need for M-matrices in the iterative schemes.

The M-matrix requirement for the iterative solvers is less of a worry, since, experimentally, the schemes can frequently be found to converge nonetheless. (In Chapter 2, we created an example where policy iteration fails when the M-matrix property is violated; however, for matrices close to M-matrix structure, often achieved for fine grid sizes, the iterations usually converge.)

Regarding the convergence to the viscosity solution, there are some specific problems for which convergence of a non-monotone discretisation has been proved, but, so far, a uniform framework is missing. (For example, in [51], for a first order HJB equation in one space dimension, a converging non-monotone finite difference scheme has been constructed, even for discontinuous initial data. In [5], for a first order HJB equation in two space dimensions, it has been shown that convergence can still be guaranteed when monotonicity is relaxed to the slightly weaker notion of ϵ-monotonicity, introduced in the paper.)

The monotonicity requirement for the discretisation is the only reason we cannot trivially extend our results to higher dimensions, as there are no straightforward monotone discretisation schemes for operators containing cross derivatives

$$\frac{\partial^2 f}{\partial x \partial y} \tag{5.2.2}$$

of a function $f(x, y)$. The problem with (5.2.2) is that all common finite difference discretisations contain 'corner' terms with 'bad' signs, like

$$-f(x + \Delta x, y - \Delta y) \quad \text{or} \quad -f(x - \Delta x, y + \Delta y), \tag{5.2.3}$$

which cannot be compensated for by the finite difference stencils of $\partial^2 f/\partial x^2$ and $\partial^2 f/\partial y^2$, and which conflict with the monotonicity requirements.

The definition of monotonicity most commonly used, as also in this thesis, is slightly stricter than absolutely necessary; specific results allowing for small deviations of the classic monotonicity definition (cf. [6]) include, for example, [5, 51].

In future work, we intend to approach the question of cross derivatives twofold.

- The existing relaxations (e.g. cf. [5, 51]) of the monotonicity requirement have not been studied explicitly with regard to cross derivatives, and we feel there is a need for establishing to what extend the monotonicity violation of cross derivatives affects the convergence properties of the finite difference scheme.
- It can be shown (e.g. cf. [13]) that a monotone discretisation can be recovered if a corrector term is added to the equation. The drawback of this approach is that, even if the actual affects of the corrector are small, we still loose consistency of our scheme. We aim to study if a corrector can be added as a regularisation which converges to the original equation in a certain limit.

Altogether, a unified approach how to deal with cross derivative terms in higher dimensional HJB equations would be hugely beneficial in that it would make all of the methods discussed in this thesis directly applicable. (Of course, in higher dimensions, we need an efficient sparse matrix solver for the linear systems of equations arising from the iterative schemes.)

Appendix A

Addition to Chapter 2

Proof of Proposition 2.2.4. We follow the proof of Proposition 2.4 in [68]. We prove by induction that

$$x_i^n = c_i \text{ for all } i \leq q - n, \tag{A.0.1}$$

where n denotes the n-th step of Algorithm 2.2.1. This is true for $n = 0$ by choice of x^0. Now assume that (A.0.1) holds for some $n \geq 0$. Noting that $\mathcal{L}P = rK > 0$ for $S < K$, and that any first order consistent discretisation is exact for linear functions, it follows that $(Ax^n - b)_i = rK > 0$ if $x_{i-1}^n = x_i^n = x_{i+1}^n = c_i$ and $i < q$, which is given for $i \leq q - n - 1$ by (A.0.1). Hence, $\min\{(Ax^n - b)_i, (x - c)_i\} = (x - c)_i = 0$, and, for $i \leq q - n - 1 = q - (n + 1)$, we have $(A^n)_i = (I_N)_i$ and $(b^n)_i = c_i$; consequently, $x_i^{n+1} = c_i$ for those i, which proves (A.0.1) for step $n + 1$. That is to say the discrete free boundary moves by no more than one index in each iteration, $(Ax^n - b)_e > 0$ for $n < q - e$, and the iteration terminates not before step $q - e$. □

Proof of Corollary 2.2.6. We follow the proof of Corollary 2.6 in [68]. That $q - e + (M - m)$ is an upper bound can be shown as in the proof of Proposition 4.6 in [50] when accounting for the fact that there may be time steps where the discrete free boundary is stationary. Arguments similar to those in the proof of Proposition 2.2.4 show that it is also a lower bound. □

Proof of Proposition 2.4.7. We follow the proof of Proposition 4.7 in [68]. It suffices to show that the discrete continuation region of the penalty solution $\{i : (x_\rho)_i > c_i\}$ is identical to $\{i : x_i > c_i\}$, i.e.

$$x_i > c_i \quad \Rightarrow \quad \exists\, \rho_0 > 0\ \forall\, \rho \geq \rho_0 : \quad (x_\rho)_i > c_i.$$

But this follows from $\|x - x_\rho\|_\infty \leq C/\rho$ (see [29], [69]). □

Bibliography

[1] R. A. Adams. *Sobolev spaces.* Amsterdam, Oxford: Academic Press, 2nd edition, 2003.

[2] B. H. Ahn. Solution of nonsymmetric linear complementarity problems by iterative methods. *Journal of Optimization Theory and Applications*, 33(2):175–185, 1981.

[3] A. L. Amadori. Nonlinear integro-differential evolution problems arising in option pricing: A viscosity solution approach. *Journal of Differential and Integral Equations*, 16(7):787–811, 2003.

[4] W. F. Ames. *Numerical methods for partial differential equations.* Academic Press: New York, 2nd edition, 1977.

[5] S. Augoula and R. Abgrall. High order numerical discretization for Hamilton-Jacobi equations on triangular meshes. *Journal of Scientific Computing*, 15(2):197–229, 2000.

[6] G. Barles. Convergence of numerical schemes for degenerate parabolic equations arising in finance. In L. C. Rogers and D. Talay, editors, *Numerical Methods in Finance*, pages 1–21. Cambridge: Cambridge University Press, 1997.

[7] G. Barles and E. R. Jakobsen. On the convergence rate of approximation schemes for Hamilton-Jacobi-Bellman equations. *Mathematical Modelling and Numerical Analysis*, 36(1):33–54, 2002.

[8] G. Barles and E. R. Jakobsen. Error bounds for monotone approximation schemes for parabolic Hamilton-Jacobi-Bellman equations. *Mathematics of Computation*, 76(260):1861–1893, 2007.

[9] A. Bensoussan and J. L. Lions. *Applications of variational inequalities in stochastic control*, volume 12 of *Studies in Mathematics and its Applications.* Amsterdam, New York, Oxford: North-Holland Pub. Co. , 1982.

[10] Y. Z. Bergman. Option pricing with differential interest rates. *The Review of Financial Studies*, 8(2):475–500, 1995.

[11] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3):637–654, 1973.

[12] A. Borici and H.-J. Lüthi. Fast solutions of complementarity formulations in American put pricing. *Journal of Computational Finance*, 9(1), 2005.

[13] F. Bouchut and F. Hermano. Finite difference schemes with cross derivatives correctors for multidimensional parabolic systems. *Journal of Hyperbolic Differential Equations*, 3(1):27–52, 2006.

[14] M. J. Brennan and E. S. Schwartz. The valuation of American put options. *The Journal of Finance*, 32(2):449–462, 1977.

[15] R. Carmona, editor. *Indifference pricing: theory and applications*. Princeton, N.J., Oxford: Princeton University Press, 2009.

[16] S. Chaumont. Uniqueness to elliptic and parabolic Hamilton-Jacobi-Bellman equations with non-smooth boundary. *Comptes Rendus Mathematique*, 339(8):555–560, 2004.

[17] Z. Chen and P. A. Forsyth. A semi-Lagrangian approach for natural gas storage valuation and optimal operation. *SIAM Journal on Scientific Computing*, 30(1):339–368, 2007.

[18] F. H. Clarke. *Optimization and nonsmooth analysis*. Philadelphia: SIAM, 1990.

[19] R. W. Cottle and G. B. Dantzig. Complementary pivot theory of mathematical programming. *Linear Algebra and its Applications*, 1:103–125, 1968.

[20] C. W. Cryer. The solution of a quadratic programming problem using systematic overrelaxation. *SIAM Journal on Control*, 9(3):385–392, 1971.

[21] C. W. Cryer. The efficient solution of linear complementarity problems for tridiagonal Minkowski matrices. *ACM Transactions on Mathematical Software*, 9(2):199–214, 1983.

[22] M. Thompson, M. Davison and H. Rasmussen. Natural gas storage valuation and optimization: A real options application. *Naval Research Logistics*, 56(3):226–238, 2009.

[23] P. Wilmott, J. Dewynne and S. Howison. *Option pricing: mathematical models and computation*. Oxford: Oxford Financial Press, 1993.

[24] C. M. Elliott and J. R. Ockendon. *Weak and variational methods for moving boundary problems*. Boston: Pitman Advanced Publishing Program, 1982.

[25] M. Fiedler. *Special matrices and their applications in numerical mathematics*. Lancaster: Nijhoff, 1986.

[26] W. H. Fleming and H. M. Soner. *Controlled Markov processes and viscosity solutions*. New York: Springer, 2nd edition, 2005.

[27] D. M. Pooley, P. A. Forsyth and K. R. Vetzal. Numerical convergence properties of option pricing PDEs with uncertain volatility. *IMA Journal of Numerical Analysis*, 23(2):241–267, 2003.

[28] P. A. Forsyth and G. Labahn. Numerical methods for controlled Hamilton-Jacobi-Bellman PDEs in finance. *The Journal of Computational Finance*, 11(2):1–44, 2007.

[29] P. A. Forsyth and K. R. Vetzal. Quadratic convergence for valuing American options using a penalty method. *SIAM Journal on Scientific Computing*, 23(6):2095–2122, 2002.

[30] P. A. Forsyth and K. R. Vetzal. Numerical methods for non-linear PDEs in finance. Working paper, University of Waterloo, http://www.cs.uwaterloo.ca/∼paforsyt/entry.pdf, 2010.

[31] Y. Huang, P. A. Forsyth and G. Labahn. Combined fixed point iteration for HJB equations in finance. Working paper, University of Waterloo, https://www.cs.uwaterloo.ca/∼paforsyt/hjb_combined.pdf, 2010.

[32] Y. Huang, P. A. Forsyth and G. Labahn. Inexact arithmetic considerations for direct control and penalty methods: American options under jump diffusion. Working paper, University of Waterloo, https://www.cs.uwaterloo.ca/∼paforsyt/inexact.pdf, 2011.

[33] Y. Huang, P. A. Forsyth and G. Labahn. Methods for pricing American options under regime switching. *SIAM Journal on Scientific Computing*, 33(5):2144–2168, 2011.

[34] D. Duffie, N. Gârleanu and L. H. Pedersen. Securities lending, shorting, and pricing. *Journal of Financial Economics*, 66(2-3):307–339, 2002.

[35] R. H. Hoppe. Multigrid algorithms for variational inequalities. *SIAM Journal on Numerical Analysis*, 24(5):1046–1065, 1987.

[36] R. A. Howard. *Dynamic programming and Markov processes*. MIT Technology Press, 1960.

[37] S. Ikonen and J. Toivanen. Operator splitting methods for American option pricing. *Applied Mathematics Letters*, 17:809–814, 2004.

[38] S. Ikonen and J. Toivanen. Operator splitting methods for pricing American options under stochastic volatility. *Numerische Mathematik*, 113(2):229–324, 2009.

[39] R. Isaacs. *Differential games. A mathematical theory with applications to warfare and pursuit, control and optimization.* New York: John Wiley and Sons, 1965.

[40] M. G. Crandall, H. Ishii and P.-L. Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27(1):1–67, 1992.

[41] K. Ito and K. Kunisch. Semi-smooth Newton methods for variational inequalities of the first kind. *Mathematical Modelling and Numerical Analysis*, 37(1):41–62, 2003.

[42] K. Ito and K. Kunisch. On a semi-smooth Newton method and its globalization. *Mathematical Programming*, 118(2):347–370, 2007.

[43] M. Hintermüller, K. Ito and K. Kunisch. The primal-dual active set strategy as a semismooth Newton method. *SIAM Journal on Optimization*, 13(3):865–888, 2002.

[44] D. Kinderlehrer and G. Stampacchia. *An introduction to variational inequalities and their applications.* New York: Academic Press, 1980.

[45] H. J. Kushner. Numerical methods for stochastic control problems in continuous time. *SIAM Journal on Control and Optimization*, 28(5):99–1048, 1990.

[46] H. J. Kushner and P. G. Dupuis. *Numerical methods for stochastic control problems in continuous time.* New York: Springer, 2nd edition, 2001.

[47] H. E. Leland. Option pricing and replication with transactions costs. *The Journal of Finance*, 40(5):1283–13–1, 1985.

[48] M. Avellaneda, A. Levy and A. Parás. Pricing and hedging derivative securities in markets with uncertain volatilities. *Applied Mathematical Finance*, 2(2):73–888, 1995.

[49] W. Allegretto, Y. Lin and H. Yang. Finite element error estimates for a nonlocal problem in American option valuation. *SIAM Journal on Numerical Analysis*, 39(3):834–857, 2002.

[50] O. Bokanowski, S. Maroso and H. Zidani. Some convergence results for Howard's algorithm. *SIAM Journal on Numerical Analysis*, 47(4):3001–3026, 2009.

[51] O. Bokanowski, N. Megdich and H. Zidani. Convergence of a non-monotone scheme for Hamilton-Jacobi-Bellman equations with discontinous initial data. *Numerische Mathematik*, 115(1):1–44, 2010.

[52] C. Chiarella, B. Kang, G. H. Meyer and A. Ziogas. The evaluation of American option prices under stochastic volatility and jump-diffusion dynamics using the method of lines. *International Journal of Theoretical and Applied Finance*, 12(3):393–425, 2009.

[53] G. H. Meyer and J. Van Der Hoek. The evaluation of American options with the method of lines. *Advances In Futures and Options Research*, 9:265–285, 1997.

[54] A. Oberman and T. Zariphopoulou. Pricing early exercise contracts in incomplete markets. *Computational Management Science*, 1(1):75–107, 2003.

[55] J.-S. Pang. Newton's method for B-differentiable equations. *Mathematics of Operations Research*, 15(2):311–341, 1990.

[56] H. Pham. *Continuous-time stochastic control and optimization with financial applications*. London: Springer, 2009.

[57] L. Qi and J. Sun. A nonsmooth version of Newton's method. *Mathematical Programming*, 58:353–367, 1993.

[58] G. Barles, J. Burdeau, M. Romano and N. Samsoen. Critical stock price near expiration. *Mathematical Finance*, 47(4):77–95, 1995.

[59] M. S. Santos and J. Rust. Convergence properties of policy iteration. *SIAM Journal on Control and Optimization*, 42:2094–2115, 2004.

[60] R. Seydel. *Tools for computational finance*. Universitext. Berlin: Springer, 3rd edition, 2006.

[61] S. Shreve. *Stochastic calculus for finance II: continous-time models*. New York: Springer, 2008.

[62] B. F. Nielsen, O. Skavhaug and A. Tveito. Penalty and front-fixing methods for the numerical solution of American option problems. *Journal of Computational Finance*, 5(2):69–97, 2002.

[63] Q. S. Song. Convergence of Markov chain approximation on generalized HJB equation and its applications. *Automatica*, 44(3):761–766, 2008.

[64] J. C. Strikwerda. *Finite difference schemes and partial differential equations*. Philadelphia: Society for Industrial and Applied Mathematics, 2nd edition, 2004.

[65] R. S. Varga. *Matrix iterative analysis*. Berlin, London: Springer, 2000.

[66] J. Wang and P. Forsyth. Maximal use of central differencing for Hamilton-Jacobi-Bellman PDEs in finance. *SIAM Journal on Numerical Analysis*, 46(3):1580–1601, 2009.

[67] P. Wilmott. *Paul Wilmott introduces quantitative finance*. Chichester: Wiley, 2nd edition, 2007.

[68] J. H. Witte and C. Reisinger. On the use of policy iteration as an easy way of pricing American options. Submitted, http://arxiv.org/abs/1012.4976, 2011.

[69] J. H. Witte and C. Reisinger. A penalty method for the numerical solution of Hamilton-Jacobi-Bellman (HJB) equations in finance. *SIAM Journal on Numerical Analysis*, 49(1):213–231, 2011.

[70] J. H. Witte and C. Reisinger. Penalty methods for the solution of discrete HJB equations – continuous control and obstacle problems. Submitted, http://arxiv.org/abs/1105.5954, 2011.

[71] C. H. Hui, C. F. Lo, V. Yeung and L. Fung. Valuing foreign currency options with a mean-reverting process: a study of Hong Kong dollar. *International Journal of Finance and Economics*, 13(1):118–134, 2008.

[72] J. Yong and X. Y. Zhou. *Stochastic controls: Hamiltonian systems and HJB equations.* New York, London: Springer, 1999.

[73] T. Zariphopoulou. A solution approach to valuation with unhedgeable risks. *Finance and Stochastics*, 5(1):61–82, 2001.