# Finite Element Methods for Bellman and Isaacs Equations

Bartosz Jaroszkowski

UNIVERSITY
OF SUSSEX

A thesis presented for the degree of

*Doctor of Philosophy*

School of Mathematical and Physical Sciences
University of Sussex

August 2021

# Declaration

I hereby declare that this thesis has not been and will not be submitted in whole or in part to another University for the award of any other degree. I also declare that this thesis was composed by myself, under the supervision of Dr Max Jensen, and that the work contained therein is my own, except where stated otherwise, such as citations.

Bartosz Jaroszkowski

UNIVERSITY OF SUSSEX

Bartosz Jaroszkowski, Doctor of Philosophy

Finite Element Methods for Bellman and Isaacs Equations

## ABSTRACT

This work concerns the numerical analysis of the Partial Differential Equations (PDEs) with a particular focus on fully nonlinear PDEs. More specifically, the main goal is to provide a finite element method to approximate solutions of Isaacs equations, which come from game theory and can be thought of as generalisation of Hamilton-Jacobi-Bellman (HJB) equations. Both of these classes of problems arise from the stochastic optimal control problems.

Is is widely known that nonlinear PDEs do not in general admit classical solutions. A way to circumvent this issue is to use a relaxed definition of derivative leading to the notion of a generalised solution. One such notion is that of viscosity solution introduced in 1980s by Crandall and Lions. The main idea is to regularise non-smooth functions by using comparison principles and subtractive testing. The theory of viscosity solutions gave rise to novel numerical methods. A general framework of formulating convergent numerical schemes for (possibly degenerate) elliptic PDEs was formulated by Barles and Souganidis in 1991. The main result states that, given a comparison principle depending on the application at hand, a monotone, stable and consistent numerical scheme converges to the unique viscosity solution of a fully nonlinear problem. This framework is used throughout this work to formulate convergent numerical schemes.

The main three contributions of the thesis are as follows. First we present a Finite Element Method to approximate solutions of isotropic parabolic problems of Isaacs type with possibly degenerate diffusions. Second we design a method of numerically approximating isotropic parabolic Hamilton-Jacobi-Bellman equations with nonlinear, mixed boundary conditions where Robin type boundary conditions are imposed via one-sided Dini derivatives. In both cases we prove the convergence of the numerical solution to the unique viscosity solution. The uniqueness of numerical solution is guaranteed by Howard's algorithm. The analysis of the HJB equations with mixed boundary conditions is motivated by option pricing in a financial setting, which leads to our third contribution. We extend the Heston model of mathematical finance to permit the uncertain market price of volatility risk and we interpret it as an HJB equation. Finally, we present a case study investigating the effects of the market price of volatility risk on the option value and its derivatives.

# Acknowledgements

I am not a man of many words but there are people who deserve recognition for their role in the creation of this work.

First and foremost, I would like to thank my supervisor Dr Max Jensen for his unwavering support and patience. His expertise and level-headed approach were invaluable, especially so during the trying times in which this work was being finalised.

I would also like to thank my family for their support and often expressing more concern for myself than I was able to.

I extend my thanks to the Engineering and Physical Sciences Research Council (EPSRC) for providing the funding and support necessary for completion of this research.

Last but not least, I express gratitude to my girlfriend. I would not be here now without you. Thank you, Nathalie.

# Contents

**1 Introduction 1**

**2 Concepts of the discretization of Bellman and Isaacs equations 6**

2.1 Fully nonlinear PDEs and their applications . . . . . . . . . . . . . . . . . . . . . . . . . 6

2.1.1 Derivation of Hamilton-Jacobi-Bellman equations . . . . . . . . . . . . . . . . . . 10

2.1.2 Derivation of Isaacs equations . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

2.2 Theory of viscosity solutions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

2.3 Comparison Principle . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

2.4 Boundary conditions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

2.5 Convergence of numerical schemes in the Barles-Souganidis framework . . . . . . . . . . 23

**3 Finite Element Methods for Bellman problems with the mixed boundary conditions 26**

3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

3.2 Mixed initial boundary value Bellman problem . . . . . . . . . . . . . . . . . . . . . . . 28

3.3 Numerical scheme . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

3.4 Monotonicity . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

3.5 Existence of numerical solutions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

3.6 Consistency . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41

3.7 Stability . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42

3.8 Convergence . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

3.9 Numerical experiments . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

3.A Appendix: Interpretation of the mixed boundary conditions . . . . . . . . . . . . . . . . 51

**4 Model and numerical solution of the Heston equation with the uncertain market price of volatility risk 55**

4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55

4.2 The uncertain Heston model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

4.3 Transformation of the uncertain Heston model . . . . . . . . . . . . . . . . . . . . . . . 61

4.4 Case Study . . . . . . . . . . 64

**5 Finite Element Methods for Isaacs problems 70**

5.1 Introduction . . . . . . . . . . 70

5.2 Isotropic Isaacs equations . . . . . . . . . . 72

5.3 The numerical method . . . . . . . . . . 74

5.4 Monotonicity . . . . . . . . . . 77

5.5 Wellposedness and a solution algorithm . . . . . . . . . . 79

5.6 Stability . . . . . . . . . . 80

5.7 Sub- and supersolutions and uniform convergence . . . . . . . . . . 82

5.8 Boundary conditions . . . . . . . . . . 84

5.9 Construction of barrier functions . . . . . . . . . . 86

5.10 Numerical experiments . . . . . . . . . . 91

**6 Discretization of the fully nonlinear equations in FEniCS 96**

6.1 Mesh creation and processing . . . . . . . . . . 96

6.2 Useful recipes . . . . . . . . . . 98

6.3 Implementation details . . . . . . . . . . 106

6.4 Comparison of the linear algebra libraries . . . . . . . . . . 111

**7 Conclusions 116**

**Bibliography 117**

# Chapter 1

# Introduction

Hamilton-Jacobi-Bellman (HJB) equations are fully non-linear partial differential equations (PDEs). They are derived from the Dynamic Programming Principle, introduced by Richard Bellman, for the solution of optimal control problems. In general, it can be shown that a solution of a first order HJB equation is a value function of a deterministic optimal control problem while the value function of its stochastic extension solves a second order HJB equation. Hamilton-Jacobi-Isaacs (Isaacs in short) equations are also fully nonlinear PDEs, derived by Rufus Isaacs during the study of two-player zero sum games. Analogously, to the HJB case, second order Isaacs equations arise from stochastic versions of zero sum games. In fact, one can think of an HJB equation as a special case of an Isaacs equation, when one of the players' strategy is limited to a singular response.

The real life uses of HJB and Isaacs equations are numerous, as one can guess from the fact they originate in optimal control theory. We present here a non-exhaustive list of areas of application that may be of interest to the reader. In engineering, they are used to solve problems dealing with optimality in aircraft navigation [95] and vehicle fuel consumption [32]. Reinforcement learning is an area of machine learning focused on maximising reward while exploring the environment. One of the steps of the optimisation process is finding estimate of Q-functions which are in fact solutions of HJB equations [75]. Another area of application in computer science is the computer vision, for example the shape from shading problem [100] where the goal is to reconstruct a three dimensional shape from a two-dimensional picture. The Hamilton-Jacobi type equations also find its direct application in financial mathematics, one example would be portfolio optimisation [74]. We will also mention several classes of problems which require solving underlying Hamilton-Jacobi equations. Firstly, we have $H_\infty$ methods which often require solving high dimensional, first order Isaacs equations [51] and which find use for example in robotics [76]. We also have front propagation theory where moving interfaces are modelled with Hamilton-Jacobi equations [94]. The applications include fluid dynamics [89] and geology [60]. Finally, we mention the quickly growing field of mean field game theory [85]. Mean field game theory models $N$-player games as $N \to \infty$. In such an infinite setting optimal behaviour of a single player is still assumed to be governed by an HJB equation. The applications include modelling crowd motion [83] and macroeconomics [20].

Despite the prevalence of HJB and Isaacs equations and nonlinear PDEs in general in real-life applications, a generalised notion of solution was not available for a long time. The main problem was that the nonlinear problems rarely admit classical solutions and weak solutions obtained via multiplicative testing are rarely guaranteed to be unique. It was not until 1980s that a series of articles ( [86], [26], [24]) laid ground for a theory of viscosity solutions. They provide a general framework (summarised in [25]) for dealing with wide range of fully nonlinear problems, including HJB and Isaacs equations. We remark at this point that they are not the only available relaxation of the notion of solution. For example, in [21] a notion of minimax solution was introduced, which is in fact equivalent to the viscosity solution for a wide range of Hamilton-Jacobi problems (see [19]). We also mention a notion of strong solution which satisfies the differential equation almost everywhere. Given that the underlying problem satisfies Cordes condition, one can use Miranda-Talenti estimate to prove existence and uniqueness of such solutions. Even though this notion of a solution is beyond the scope of this dissertation, we inform the reader that a numerical method converging to the strong solution of a fully nonlinear HJB problems can be found in [104].

Even with the emergence of the theory of viscosity solutions, the formulation of numerical methods for second order fully nonlinear PDEs still proved to be difficult. The main issue is that uniqueness of solution is conditional and a numerical scheme has to be designed in a way which allows it to select appropriately between potential candidate solutions in order to converge to the viscosity solution. It is especially difficult due to the fact that notion of viscosity solution is based on a comparison principle, instead of variational one as in the case of weak solutions. A framework which allows to prove the convergence of the numerical scheme to the unique viscosity solution of a wide class of problems appeared for the first time in [6]. The main result of that paper states what conditions have to be satisfied by the scheme to be convergent, however the way in which the numerical scheme is to be constructed is still very much problem dependent.

The aim of this dissertation is to provide a numerical scheme capable of approximating the unique viscosity solution of Hamilton-Jacobi problems in two novel settings. In both cases we extend the results of [68] where method of approximating second order possibly degenerate elliptic HJB equation was presented. We motivate the first case by considering a new Heston model which is an extension of the Black-Scholes model used for option price evaluation. More precisely, for a stock price $S$, a variance $v$ and time $t$, the option value $V(S,v,t)$ is expected to solve the following boundary problem

$$\begin{aligned}
-\partial_t V - \frac{1}{2}\left(S^2 v\frac{\partial^2 V}{\partial S^2} + 2\rho\xi vS\frac{\partial^2 V}{\partial S\partial v} + \xi^2 v\frac{\partial^2 V}{\partial v^2}\right)&\\
-rS\frac{\partial V}{\partial S} - [\kappa(\gamma - v) - \xi\lambda\sqrt{v}]\frac{\partial V}{\partial v} + rV = 0,&\\
V(S,v,T) = \Lambda(S),&\\
V(0,v,t) = \Lambda(0),&\\
\lim_{S\to\infty}\frac{\partial V}{\partial S}(S,v,t) = \lim_{S\to\infty}\frac{\partial \Lambda}{\partial S}(S),&\\
-rS\frac{\partial V}{\partial S}(S,0,t) - \kappa\gamma\frac{\partial V}{\partial v}(S,0,t) +&\\
rV(S,0,t) - \partial_t V(S,0,t) = 0,&\\
\lim_{v\to\infty}\frac{\partial V}{\partial v}(S,v,t) = 0,&
\end{aligned} \tag{1.1}$$

where $\Lambda$ depends on the pay-off function of a considered option and $\rho$, $\xi$, $\kappa$, $\gamma$, $\lambda$ are constant parameters. We then assume $\lambda$ to be an uncertain parameter and we define linear operators satisfying (1.1) for each value of $\lambda$ lying inside of some interval $L$. After truncating the unbounded domain of (1.1), transformation of variables and using controls $\boldsymbol{\lambda} : [0,T] \to L$ one can show that the optimal option value solves the HJB equation with the mixed boundary conditions of the form

$$\begin{aligned}
-\partial_t v + \sup_{\alpha\in A}(L^\alpha\ v - f^\alpha) &= 0 && \text{in } [0,T)\times\Omega,\\
-\partial_t v + \sup_{\alpha\in A}\left(L^\alpha_{\partial\Omega} v - g^\alpha\right) &= 0 && \text{on } [0,T)\times\partial\Omega_t,\\
\sup_{\alpha\in A}\left(L^\alpha_{\partial\Omega} v - g^\alpha\right) &= 0 && \text{on } [0,T)\times\partial\Omega_R,\\
v - g\ &= 0 && \text{on } [0,T)\times\partial\Omega_D,\\
v - v_T &= 0 && \text{on } \{T\}\times\overline{\Omega},
\end{aligned} \tag{1.2}$$

where we denote families of linear operators on the domain $\Omega$ and its boundary by $L^\alpha$ and $L^\alpha_{\partial\Omega}$, respectively. Now the main difficulty is to discretise the boundary operators in such a way that the numerical scheme remains consistent with framework of [6]. Another class of problems whose viscosity solution we would like to approximate are second order Isaacs equations with the non-homogeneous Dirichlet boundary conditions of the form

$$\begin{aligned}
-\partial_t v + \inf_{\beta\in B}\sup_{\alpha\in A}(L^{(\alpha,\beta)}v - f^{(\alpha,\beta)}) = 0 &&& \text{in } (0,T)\times\Omega,\\
v = g &&& \text{on } (0,T)\times\partial\Omega,\\
v = v_T &&& \text{on } \{T\}\times\overline{\Omega}.
\end{aligned} \tag{1.3}$$

Here $L^{(\alpha,\beta)}$ denotes a linear operator on domain $\Omega$. The main difficulty here is that $\inf\sup$ operator is nonconvex and therefore accumulation point of the sequences of numerical solutions is especially difficult to evaluate on boundary. As a result formulating a meaningful comparison principle for a sequence of numerical solutions is a demanding task.

Throughout this dissertation the discretisation of PDEs is achieved via Finite Element Methods (FEM) using P1 elements. The procedure is as follows. We first obtain weak formulation of the differential equation, using the technique of freezing the coefficient described in [68] to deal with second order terms in non-divergence form. We then obtain a discretisation of the domain via triangulation satisfying constraint of strict acuteness. Monotonicity of the scheme is ensured through artificial diffusion. Numerical solutions are known to exist uniquely due to Howard's algorithm.

Basic understanding of the Sobolev spaces is assumed throughout this dissertation so the reader not familiar with this or any other implementation details of the Galerkin type methods is referred to one of [118, Chapter 3], [40, Chapter 1] or [15, Chapters 0-3]. The main advantage of FEM lies in its flexibility and applicability to a wide range of problems. Most importantly, the Finite Element approach can be used on irregularly shaped domains and it can pick up singularities of the solution in crucial parts of the domain given an appropriate choice of mesh.

In this paragraph we present a general summary of numerical methods used for solving second order HJB and Isaacs equations. Reader interested in the literature review of the approaches to obtaining numerical approximation of problems (1.2) and (1.3) specifically is referred to the discussion at the beginning of the chapter dedicated to each of them. We firstly remark that there exists a rich literature on Finite Difference schemes for HJB problems in non-divergence form. For more recent results we refer reader to [5], [79], [47] and for summary to [82]. Numerous different approaches exist, including Discontinuous Galerkin Methods ( [104], [109], [49]) and Finite Element Methods ( [17] for problems admitting classical solution, otherwise [68], [84], [115]). We also point out that a Semi-Lagrangian method was used in [46] to solve the Monge-Ampère problem interpreted as HJB equation with a special choice of controls. The literature regarding Isaacs problems is much more limited. We point to the result on the convergence rates of the Finite Difference schemes obtained in [18] and later extended to the more general case in [80]. The convergence of a Semi-Lagrangian scheme was shown in [31] and a Finite Element Method approximation along with convergence rate estimates can be found in [101]. Additionally, we direct the reader's attention to [48] for a detailed description of the Vanishing Moment Method which can be used as an alternative to Barles-Souganidis framework from [6]. Finally, we recommend the recent review articles [45] and [90] for an overview of contemporary numerical methods for fully nonlinear PDEs. In the latter, HJB and Isaacs problems are treated as prototypical convex and nonconvex nonlinear problems, respectively, and a number of methods for solving them is discussed in a great detail.

The structure of this dissertation is as follows. Each chapter can be read separately, although Chapters 3 and 4 are related to one another as the former provides numerical method used in the latter. We also remark that Chapters 3 and 5 are extensions of the method described in [68], however the contributions made in both of them do not overlap. We now discuss briefly the content of each individual chapter.

In Chapter 2 we take a step back and familiarise the reader with the basic concepts required to carry out the discretisation of HJB and Isaacs problems which are then used in the subsequent chapters. We begin by taking a broad look at the fully nonlinear PDEs. A special attention is paid to HJB and Isaacs equations which we derive from a generic optimal control problems. We then provide alternative definitions of viscosity solutions and we discuss how a comparison principle leads to the uniqueness of the viscosity solution for the Dirichlet problem. We then discuss what is meant by a viscosity solution to the generalised boundary value problem and finally we explain under what conditions a numerical scheme is guaranteed to converge according to the Barles-Souganidis framework. Readers already familiar with those concepts may want to skip this chapter.

In Chapter 3 we show a strong uniform convergence of monotone P1 Finite Element Methods to the viscosity solution of isotropic parabolic Hamilton-Jacobi-Bellman equations with the mixed boundary conditions on unstructured meshes and for possibly degenerate diffusions. Boundary operators, which generally are discontinuous across face boundaries and type changes, are discretised via a lower Dini derivative. In time the Bellman equation is approximated through IMEX schemes. Existence and uniqueness of numerical solutions follows through Howard's algorithm. We present capabilities of the scheme through the numerical solution of a Skorokhod problem on a nonconvex domain.

In Chapter 4 we investigate the Heston option pricing model with the uncertain market price of volatility risk. We reinterpret it as a stochastic optimal control problem and derive the underlying HJB equation with mixed boundary conditions consistent with the scheme described in Chapter 3. Finally, we perform a case study whose goal is to investigate the impact of the market price of volatility risk on the option price and its derivatives.

In Chapter 5 we present a P1 Finite Element scheme converging to the viscosity solution of (1.3). A pointwise convergence of the envelopes of the numerical solution to the Dirichlet boundary conditions is proven subject to the condition that construction of specific family of barrier functions is possible. Finally, we present numerical experiments indicating optimal convergence rates and we show capabilities of the scheme by approximating the value function of a stochastic two player game with degenerate diffusion.

Chapter 6 is focused on the implementation details of the numerical schemes using the Python interface to FEniCS Finite Element library. We present the code snippets which author finds instructive and we discuss how one can solve problems posed in a non-divergence form, which is not the default use case of FEniCS . Finally, we briefly discuss the performance and benchmark different implementations of the same numerical task.

# Chapter 2

# Concepts of the discretization of Bellman and Isaacs equations

## 2.1 Fully nonlinear PDEs and their applications

It is difficult to overstate the importance of the PDEs in the contemporary applied mathematics. It is a basic tool used for modelling processes in diverse settings, including but not limited to financial mathematics (portfolio hedging, risk assessment, optimal execution of trades), chemical and biological sciences (pattern formation, autocatalytic reactions, epidemic dynamics) and physics (fluid dynamics, quantum mechanics). However, the world is in its nature nonlinear and simplified linear models often fail to describe the complexity of a natural process. This leads to increasing complexity of the proposed models which require obtaining solutions to the nonlinear PDEs. Even though many advancements in the field were made throughout 20th century, the nonlinear PDEs are still an area of active research. One of the driving forces of the advance is access to a rapidly increasing computational power, allowing better methods of numerical approximations for increasingly complex nonlinear problems. Before delving into the intricacies of the design of numerical methods we briefly introduce some of the concepts coming from the theory of PDEs which underpin the essence of this work. We first turn our attention to the notion of nonlinearity and what is understood by a fully nonlinear PDE.

Let us consider a functional $F$ which encodes a generalised second order partial differential equation. More explicitly, for a function $u(x,y)$ that is at least twice continuously differentiable we have that

$$F(x,y,u(x,y),u_{xx}(x,y),u_{xy}(x,y),u_{yy}(x,y)) = 0, \tag{2.1}$$

for some $(x,y) \in \mathbb{R}^2$. Note that the functional $F$ could be generalised to include even higher order PDEs and higher dimensional domains. We refrain from doing that since the main focus of this work is on second order PDEs and its current form makes it more convenient to provide concrete examples. We are currently considering a $u$ that is ”smooth” enough to satisfy the Equation (2.1) in a classical sense. In general, solutions of nonlinear problems do not satisfy this condition and often we only hope for a solution which solves the differential problem almost everywhere. In order to be able to search for such a function, a new relaxed notion of a solution is required. This idea is discussed at length in Section 2.2 but for the time being we keep a restrictive assumption of $u$ being twice differentiable.

Having defined the generalised PDE operator $F$ we can now return to the discussion of nonlinearity. Actually, the intuitive notion of linearity coming from linear algebra can be applied in PDE setting as well. We say that $F$ is linear if, given any number of solutions $u_n$ of (2.1), their linear combination $\sum_n c_n u_n$ where $c_n \in \mathbb{R}$ is also a solution. More generally, we can divide the PDEs into three general categories as follows. First, we have linear problems, meaning that $F$ is linear with respect to all derivatives of function $u$. An example would be:

$$a(x,y)\frac{\partial^2 u}{\partial x^2} + b(x,y)\frac{\partial^2 u}{\partial x \partial y} + c(x)u(x,y) + d(x) = 0.$$

Another class of problems are quasi-linear problems. In this case, the differential equation is linear at least in the highest degree derivative of $u$, but not necessarily in the lower degree ones. Consider the following example:

$$(\frac{\partial u}{\partial x})^2 \frac{\partial^2 u}{\partial x^2} + (\frac{\partial u}{\partial x})^2 + d(x) = 0.$$

In this case the coefficient of the highest order term is a nonlinear function but this choice of $F$ is still linear with respect to $\frac{\partial^2 u}{\partial x^2}$. Sometimes the related class of semi-linear PDEs is considered in the literature. Those are defined as quasi-linear PDEs whose highest order coefficient is not $u$ dependent. In other words, the equation

$$a(x,y)\frac{\partial^2 u}{\partial x^2} + (\frac{\partial u}{\partial x})^2 + d(x) = 0$$

is semi-linear, while the previous example is not. Finally, we speak of a fully nonlinear equation to emphasise that we do not make the assumption of quasi linearity. Nonetheless, most literature does not explicitly exclude quasi-linear problems from the class of fully nonlinear problems. An example would be:

$$(\frac{\partial^2 u}{\partial x^2})^2 + d(x,y) = 0.$$

Having established what it means for a PDE to be fully nonlinear, we now turn our attention to several areas of application. It is by no means an exhaustive list but rather a visualisation of the scope of the problems that may require an approximation of a solution to a nonlinear PDE. It is worth noting that no such a thing as a general theory of the fully nonlinear PDEs exists. In general, approaches to solving a nonlinear problem will differ depending on the individual type of an equation under consideration. However, it is often possible (subject to assumptions and limitations) to reformulate equations of one type into equations of another type. Hence the types of equations presented here are not to be thought of as classification into distinct categories but rather labels for types of problems which can be solved with a specifically designed sets of methods.

We begin by explaining the main focus of this work, namely the problems coming from the optimal control theory. The main goal of the optimal control is to optimise some objective function associated to a dynamic system which can be manipulated via controls. By considering the dynamics of a non-deterministic system which are written in a form of a stochastic PDE and applying the Dynamic Programming Principle one can derive so-called Hamilton-Jacobi-Bellman (HJB) equation of the form

$$-\partial_t v + \sup_{\alpha \in A} \left(-a^{\alpha}(t,x)\Delta v(t,x) - b^{\alpha}(t,x) \cdot \nabla v(t,x) - f^{\alpha}(t,x)\right) = 0,$$

where pointwise supremum is taken over the set of available controls $A$. We defer the details of the derivation to Section 2.1.1. Considering a similar optimal control problem but with two independent forces (referred to as players) driving the dynamics of the system, one can formulate so-called Hamilton-Jacobi-Isaacs (Isaacs in short) equation of the form

$$-\partial_t v + \inf_{\beta \in B} \sup_{\alpha \in A} \left(-a^{(\alpha,\beta)}(t,x)\Delta v(t,x) - b^{(\alpha,\beta)}(t,x) \cdot \nabla v(t,x) - f^{(\alpha,\beta)}(t,x)\right) = 0.$$

Again, a more detailed discussion can be found in Section 2.1.2. We only point out that the Isaacs equation can be thought of as generalisation of the HJB equation by considering control set $B$ to be a singleton.

The next area of interest are fully nonlinear problems present in financial mathematics. Probably one of the most known financial equations is the Black-Scholes equation used to evaluate the theoretical value $V$ of a financial option. Under assumption of constancy of volatility $\sigma$ of stock price $S$ and risk-free interest rate $r$ one can estimate $V$ by solving a second order linear PDE of the form

$$\partial_t V + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0.$$

Due to the limitations of the original Black-Scholes model many modifications have been considered including the so-called Heston model. In the Heston model one considers the evolution of $\sigma$ in time to be a mean-reverting process with a long term mean equal to $\gamma$ and a reversion level equal to $\kappa$. We also assume that it is correlated to the stochastic process which represents evolution of the stock price. Considering the square of the volatility $v$, the volatility of volatility $\xi$ and the correlation coefficient $\rho$ we obtain equation of the form

$$\partial_t V + \frac{1}{2}\left(S^2 v \frac{\partial^2 V}{\partial S^2} + 2\rho\xi v S \frac{\partial^2 V}{\partial S \partial v} + \xi^2 v \frac{\partial^2 V}{\partial v^2}\right) - rS\frac{\partial V}{\partial S} - [\kappa(\gamma - v) - \xi\lambda\sqrt{v}]\frac{\partial V}{\partial v} + rV = 0.$$

The details of derivation are available for closer inspection in Chapter 4. Note that this is again a linear PDE.

However, problems like worst case scenario analysis and portfolio optimisation require a fully nonlinear PDEs to be solved. One example would be making one or more parameters in such a PDE uncertain. One can then evaluate how the value of a financial asset can be affected by an incorrect approximation of those parameters. Such a case study is performed in Chapter 4.

We conclude this section with several more examples of nonlinear PDEs to explain the scope of the problem class. Arising from the problems in optimal transport, geometry, reflector shape design (see [71]) or meteorology(see [11]) we have the Monge-Ampère equations of the general form

$$\det D^2 u = f(x,u,\nabla u).$$

A classical example of the Monge-Ampère problem is the prescribed Gaussian curvature equation where $f$ takes the form

$$f = K(x)(1+|\nabla u|^2)^{(n+2)/2},$$

with $K(x)$ being the Gaussian curvature of graph of $u$ at point $x$. For more details on connection to the optimal transport see [96] and for an overview of the classical solution theory see [56, Chapter 17]. Another example of a nonlinear PDE used in a number of applications, including wavefront propagation and geometrical optics, is the Eikonal equation of the form

$$|\nabla u(x)| = f(x). \tag{2.2}$$

For a more detailed discussion we refer reader to [27]. We also mention the classical obstacle problem of the form

$$\min\{-\Delta u(x), u(x) - \phi(x)\} = 0 \tag{2.3}$$

which is discussed in more detail in [99] and the infinity Laplacian equation of the form

$$\Delta_\infty u := \langle Du, D^2 u Du \rangle = u_{x_i} u_{x_j} u x_i x_j = 0,$$

which arises in numerous areas of the nonlinear PDE theory as summarised in [7].

We point out that the setting of the Isaacs equation is broad and since many of the equations stated above are optimisation problems, they can often be reformulated as the Isaacs (or more specifically HJB) equation. As an example we provide a brief discussion of the Monge-Ampère and eikonal problems, noting that reinterpreting the obstacle problem in the form presented in (2.3) as HJB problem is trivial. Let us consider a control set $S := \{M \text{ is a non-negative symmetric matrix} \mid \mathrm{tr} M = 1\}$. It was shown in [78] that for a positive symmetric Hessian and $d$-dimensional domain there is an equivalence between the solution to the

HJB equation of the form

$$\sup_{A\in S}\left(-A\,:\,D^2u(x)+f(x)\sqrt[d]{\det A}\right)=0$$

and the solution to the Monge-Ampère equation of the form

$$\det D^2u(x)-\left(\frac{f}{d}\right)^d=0.$$

In case of the eikonal equation (2.2) we define the control set $A:=\left\{\vec{v}\in\mathbb{R}^2\;\middle|\;|\vec{v}|=1\right\}$ and make use of the fact that Euclidean norm of a gradient can be written as $|\nabla u|=\sup_{\alpha\in A}(\alpha\cdot\nabla u)$. As a result we obtain an HJB problem of the form

$$\sup_{\alpha\in A}\left(\alpha\cdot\nabla u(x)-f(x)\right)=0.$$

### 2.1.1 Derivation of Hamilton-Jacobi-Bellman equations

In this section we follow the discussion in [114] to briefly present how the HJB equation arises from optimal control problems. It is therefore natural to first discuss deterministic optimal control (leading to a first order HJB equation) and then extend the discussion to a non-deterministic, stochastic case which will lead us to the formulation of the general second order HJB equation. We remark that throughout this section we make some strong continuity assumptions which will not hold in general. However, we are considering the case when the HJB equation has a classical, continuous solution and hence such assumptions are justified.

In optimal control we deal with some system evolving in time whose evolution is in the setting of this work represented by the Ordinary Differential Equation (ODE) or a system of ODEs. Additionally, it is possible to influence the behaviour of such system through a time dependent parameter we refer to as *control*. In other words, the state of the system at time $t$ and controls are linked by a differential equation. More explicitly, let $t_0<t<\infty$ and consider initial state $x_0\in\mathbb{R}^d$. Given a compact metric space $A$ we define $\mathcal{A}:=\{\boldsymbol{\alpha}:[t_0,T]\to A\mid\boldsymbol{\alpha}\text{ is measurable}\}$. Then the dynamics of the system are represented by the following differential equation:

$$\begin{cases}\dot{x}(t)=b^{\boldsymbol{\alpha}(t)}(t,x(t)) & t\in(t_0,T],\\ x(t_0)=x_0,\end{cases}\tag{2.4}$$

where a map $\boldsymbol{\alpha}\in\mathcal{A}$ is called an admissible control and $b^{\alpha(t)}:(t_0,T]\times\mathbb{R}^d\to\mathbb{R}^d$ is a map associated to it. Note the use of shorthand notation which for a function $g$ allows us to write $g^\alpha(t)=g(t,\alpha)$ for any $\alpha\in A$. The system of equations (2.4) is called control system and its solution under a control $\boldsymbol{\alpha}$ denoted as $x_{\boldsymbol{\alpha}}(t)$ is a state trajectory. From now on we assume that for a single control, the control system admits a unique trajectory. Note that in general $x_{\boldsymbol{\alpha}}(t)$ depends on the initial tempo-spatial position so it would be more accurate to denote it as $x_{(\boldsymbol{\alpha},t_0,x_0)}(t)$. However, we drop the $t_0$ and $x_0$ from the subscript for the sake of brevity.

We are now interested in finding a control which in some sense optimises the response of the control system. For this purpose, we require a way to measure the performance of a control. In other words, we would like to evaluate the cost of the system evolution starting at time $t_0$ with state $x_0$ until termination time $T$ under state trajectory $x_{\boldsymbol{\alpha}}(t)$. This motivates introduction of the following *cost functional*:

$$\mathcal{J}(t_0,x_0,\boldsymbol{\alpha}) := \int_{t_0}^{T} f^{\boldsymbol{\alpha}(s)}(s,x_{\boldsymbol{\alpha}}(s))ds + v_T(x_{\boldsymbol{\alpha}}(T)), \tag{2.5}$$

where $f^{\boldsymbol{\alpha}(t)} : [t_0,T] \times \mathbb{R}^d \to \mathbb{R}$ is the running cost function and $v_T : \mathbb{R}^d \to \mathbb{R}$ is a function representing the terminal cost of the system. We are now interested in finding a control $\hat{\boldsymbol{\alpha}} \in \mathcal{A}$ which minimises (2.5) or more precisely

$$\mathcal{J}(t_0,x_0,\hat{\boldsymbol{\alpha}}) = \inf_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{J}(t_0,x_0,\boldsymbol{\alpha}). \tag{2.6}$$

Note that for the remainder of this section we could replace inf in (2.6) with min without affecting the analysis. We will however refrain from doing so in order to stay consistent with the subsequent chapters where we work under more restrictive assumptions. Returning to (2.6), if we assume that at least one such $\hat{\alpha}$ exists, we refer to it and its corresponding trajectory as optimal. It is also useful to define a function representing the cost associated to the optimal trajectory $x_{\hat{\boldsymbol{\alpha}},t_0,x_0}$ as follows

$$v(t_0,x_0) := \mathcal{J}(t_0,x_0,\hat{\boldsymbol{\alpha}}). \tag{2.7}$$

We will call such $v$ the *value function*.

We now turn our attention to the fundamental result allowing the formulation of the HJB equation, called Dynamic Programming Principle or Bellman's principle of optimality. It is motivated by the intuitive observation that for any admissible control $\boldsymbol{\alpha} \in \mathcal{A}$, time $t \in [t_0,T)$, position $x$ and $h < T - t$, the value function $v(t,x)$ is bounded from above by a general, not necessarily optimal running cost between times $t$ and $t+h$ added to the value function evaluated at time $t+h$. The following result confirms and refines this observation.

**Theorem 1.** *Let us assume that for all $\alpha \in A$ we have that $b^{\alpha}$ and $f^{\alpha}$ are uniformly continuous, bounded and Lipschitz continuous in space and time. Then for any time $t > t_0$, time increment $h \leq T - t$ and position $x \in \mathbb{R}^d$ we have that*

$$v(t,x) = \inf_{\boldsymbol{\alpha}\in\mathcal{A}} \int_{t}^{t+h} f^{\boldsymbol{\alpha}(s)}(s,x_{\boldsymbol{\alpha}}(s))ds + v(t+h,x_{\boldsymbol{\alpha}}(t+h)). \tag{2.8}$$

*Proof.* We begin the proof by showing that $v(t,x)$ is a lower bound of the right hand side of (2.8). In fact, by the definition of $v$ and $\mathcal{J}$ we have that

$$v(t,x) \leq \mathcal{J}(t,x,\boldsymbol{\alpha}) = \int_{t}^{t+h} f^{\boldsymbol{\alpha}(s)}(s,x_{\boldsymbol{\alpha}}(s))ds + \mathcal{J}(t+h,x_{\boldsymbol{\alpha}}(t+h),\boldsymbol{\alpha}).$$

Taking infimum over $\boldsymbol{\alpha}$ yields the required result. We now need to show that $v(t,x)$ is the largest such lower bound. Due to the continuity of $f^{\alpha}$ and $b^{\alpha}$ we have that for any $\varepsilon > 0$ there exists a control $\boldsymbol{\alpha}_{\varepsilon} \in \mathcal{A}$ such that

$$\begin{aligned} v(t,x)+\varepsilon \geq \mathcal{J}(t,x,\boldsymbol{\alpha}_{\varepsilon}) &= \int_t^{t+h} f^{\boldsymbol{\alpha}_{\varepsilon}(s)}(s,x_{\boldsymbol{\alpha}_{\varepsilon}}(s))ds + \mathcal{J}(t+h,x_{\boldsymbol{\alpha}_{\varepsilon}}(t+h),\boldsymbol{\alpha}_{\varepsilon}) \\ &\geq \int_t^{t+h} f^{\boldsymbol{\alpha}_{\varepsilon}(s)}(s,x_{\boldsymbol{\alpha}_{\varepsilon}}(s))ds + v(t+h,x_{\boldsymbol{\alpha}_{\varepsilon}}(t+h)) \\ &\geq \inf_{\boldsymbol{\alpha}\in\mathcal{A}} \int_t^{t+h} f^{\alpha(s)}(s,x_{\boldsymbol{\alpha}}(s))ds + v(t+h,x_{\boldsymbol{\alpha}}(t+h)). \end{aligned}$$

The equation (2.8) follows directly. □

Another way to formulate this result is to state that if $v$ is optimal over a whole time interval it is also optimal over any of its subintervals. Having established the Dynamic Programming Principle we are now ready to show that the value function of a deterministic optimal control problem is in fact the solution of a first order HJB equation. We formalise this statement in the following result.

**Proposition 1.** *Assume that for all $\alpha \in A$ we have that $b^{\alpha}$ and $f^{\alpha}$ are uniformly continuous, bounded and Lipschitz continuous in space and time and let $v_T$ be bounded and Lipschitz continuous in space. Let us also assume that control system* (2.4) *admits a unique value function $v \in C^1([0,T] \times \mathbb{R}^d)$. Then $v$ is the solution of the following first order final value problem*

$$\begin{cases} -\partial_t v(t,x) + \sup_{\alpha\in A}\left(-b^{\alpha}(t,x)\cdot\nabla v(t,x) - f^{\alpha}(t,x)\right) = 0 & (t,x)\in[0,T)\times\mathbb{R}^d \\ v(T,x) = v_T(x) & x\in\mathbb{R}^d. \end{cases}$$

*Proof.* Fix some $\alpha^* \in A$ and consider a control $\mathcal{A} \ni \boldsymbol{\alpha}(t) \equiv \alpha^*$. Using (2.8) we have that for $h$ small enough

$$\frac{v(t,x) - v(t+h,x_{\boldsymbol{\alpha}}(t+h))}{h} - \frac{1}{h}\int_t^{t+h} f^{\alpha^*}(s,x_{\boldsymbol{\alpha}}(s))ds \leq 0.$$

Using the continuity of $v$, $b^{\alpha}$ and $f^{\alpha}$, letting $h \downarrow 0$ and using (2.4) we get that for any initial choice of $\alpha^*$

$$0 \geq -\partial_t v - b^{\alpha^*}(t,x)\cdot\nabla v(t,x) - f^{\alpha^*}(t,x). \tag{2.9}$$

Note that the above will be also hold true for $\alpha^*$ for which the supremum of right hand side of (2.9) is attained. We now prove the inverse of (2.9). By continuity of $v$, for any positive $\varepsilon > 0$ and $h$ small enough there exists a control $\boldsymbol{\alpha}_{\varepsilon}(\cdot) \in \mathcal{A}$ such that

$$v(t,x) + h\varepsilon \geq \int_t^{t+h} f^{\boldsymbol{\alpha}_{\varepsilon}(s)}(s,x^{\boldsymbol{\alpha}_{\varepsilon}}(s))ds + v(t+h,x_{\boldsymbol{\alpha}_{\varepsilon}}(t+h)).$$

Since we assumed $v$ to be continuously differentiable we have that

$$\begin{aligned}
-\varepsilon &\leq \frac{v(t,x)-v(t+h,x_{\alpha_\varepsilon}(t+h))}{h} - \frac{1}{h}\int_t^{t+h} f^{\alpha_\varepsilon(s)}(s,x_{\alpha_\varepsilon}(s))ds \\
&= \frac{1}{h}\int_t^{t+h} -\partial_t v(s,x_{\alpha_\varepsilon}(s)) - b^{\alpha_\varepsilon(s)}(x_{\alpha_\varepsilon}(s))\cdot\nabla v(s,x_{\alpha_\varepsilon}(s)) - f^{\alpha_\varepsilon(s)}(s,x_{\alpha_\varepsilon}(s))ds \\
&\leq \frac{1}{h}\int_t^{t+h} -\partial_t v(s,x_{\alpha_\varepsilon}(s)) - \sup_{\alpha\in A}\left(b^\alpha(x_\alpha(s))\cdot\nabla v(s,x_\alpha(s)) - f^\alpha(s,x_\alpha(s))\right)ds.
\end{aligned}$$

We now make use of the uniform continuity $b^\alpha$ and $f^\alpha$ for all $\alpha \in A$ and take the limit as $h \to 0$ to obtain

$$0 \leq -\partial_t v(t,x) + \sup_{\alpha\in A}\left(-b^\alpha(t,x)\cdot\nabla v(t,x) - f^\alpha(t,x)\right).$$

This result combined with (2.9) gives us the required result. □

Hence we can obtain the value function of an optimal control problem by solving the related HJB equation. We remark at this point that assumption of $v \in C^1([0,T]\times\mathbb{R}^d)$ does not hold in general and depends on the exact formulation of (2.4). In order to generalise this result, we require notion of viscosity solution which we will introduce in Section 2.2. For the time being, we assume for the simplicity that the underlying optimal control problem satisfies this restriction.

Note that if we know the value function denoted as $\hat{v}$, it is also in principle possible to retrieve the optimal control and the optimal trajectory for each starting time $t_0$ and position $x_0$. The optimal control $\hat{\alpha}$ is the one for which supremum of $\sup_{\alpha\in A}\left(-b^\alpha(t_0,x_0)\cdot\nabla\hat{v}(t_0,x_0) - f^\alpha(t_0,x_0)\right)$ is attained. We can then use $\hat{a}, t_0, x_0$ to find the unique solution of (2.4) which is the optimal trajectory.

Having considered a deterministic optimal control problem we now turn our attention to the stochastic case. Let us consider a standard $d$-dimensional Brownian motion $W$ defined on a given filtered probability space satisfying the usual condition. We can then consider the following stochastic extension of a deterministic controlled system:

$$\begin{cases} dx(t) = b^{\alpha(t)}(x(t))dt + \sigma^{\alpha(t)}(t,x(t))dW(t) \quad t\in(t_0,T], \\ x(t_0) = x_0, \end{cases} \tag{2.10}$$

where $\sigma^{\alpha(t)} : [0,T]\times\mathbb{R}^d \to \mathbb{R}^{d\times d}$. We remark at this point that the set of admissible controls $\mathcal{A}$ will in general depend on the filtered probability space on which $W$ is defined. More precisely, given the underlying sample set $\Omega$ and filtration $\{\mathcal{F}_\sqcup\}_{t\geq 0}$ of the Brownian motion, any admissible control needs to map from $[0,T]\times\Omega$ to $A$ and additionally it needs to be $\{\mathcal{F}_\sqcup\}_{t\geq 0}$-adapted. Notice as well that given any state trajectory $x(\cdot)$ we have that for any $t > 0$ the position $x(t)$ is actually a random variable instead of a fixed point in space. Thus in order to be able to apply Dynamic Programming Principle we would have to use the fact that admissible controls are $\{\mathcal{F}_{\sqcup}\}_{t\geq 0}$-adapted and additionally allow the Brownian motion $W(\cdot)$ and underlying complete probability space to be part of the control. Then using probability measure conditioned upon filtration generated by the Brownian motion we have that the position $x(t)$ is almost surely deterministic. This in turn allows us to consider optimal path or cost given a varying control and probability space. Since the technical details of this process are beyond the scope of this dissertation, the interested reader is referred to [114] for more details.

Analogously to the deterministic case we can define the expected cost functional

$$\mathcal{J}(t_0,x_0,\boldsymbol{\alpha}) := \mathbf{E}_{(t_0,x_0)}\left\{\int_{t_0}^{T} f^{\alpha(s)}(s,x_{\boldsymbol{\alpha}}(s))ds + v_T(x_{\boldsymbol{\alpha}}(T))\right\},$$

optimal control $\hat{\alpha}$ such that

$$\mathcal{J}(t_0,x_0,\hat{\boldsymbol{\alpha}}) = \inf_{\boldsymbol{\alpha}\in\mathcal{A}} \mathcal{J}(t_0,x_0,\boldsymbol{\alpha})$$

and the expected value function $v$ such that

$$v(t_0,x_0) := \mathcal{J}(t_0,x_0,\hat{\boldsymbol{\alpha}}).$$

We note at this point that while the derivation of the second order order HJB equation is in spirit similar to the first order case, the inclusion of the non-deterministic term makes the argument much more technical. Since the stochastic processes are not the focus of this dissertation, we will present the main results analogous to the non-deterministic case and for the detailed proofs we refer reader to [114].

Let us assume that for all $\alpha \in A$ we have that $\sigma^{\alpha}$, $b^{\alpha}$ and $f^{\alpha}$ are uniformly continuous in time and Lipschitz continuous in space.

**Theorem 2.** *For any starting time $t_0 \leq T-h$ and any starting position $x_0 \in \mathbb{R}^d$ we have that*

$$v(x_0,t_0) = \inf_{\boldsymbol{\alpha}\in\mathcal{A}} \mathbf{E}_{(t_0,x_0)}\left\{\int_{t_0}^{t+h} f^{\boldsymbol{\alpha}(s)}(s,x_{\boldsymbol{\alpha}}(s))ds + v(t_0+h,x_{\boldsymbol{\alpha}}(t_0+h))\right\}. \tag{2.11}$$

Note that this is the stochastic version of Theorem 1.

**Proposition 2.** *Let $v_T$ be bounded and Lipschitz continuous in space. Let $v \in C^{1,2}([0,T]\times\mathbb{R}^d)$ be the value function of the control system* (2.13). *Then $v$ is a solution of the following first order final value problem.*

$$\begin{cases} -\partial_t v + \sup_{\alpha\in A}\left(-\frac{1}{2}\mathrm{tr}(\sigma^{\alpha}(\sigma^{\alpha})^T)\Delta v(t,x) - b^{\alpha}(t,x)\cdot\nabla v(t,x) - f^{\alpha}(t,x)\right) = 0 & (t,x)\in[0,T)\times\mathbb{R}^d \\ v(T,x) = v_T(x) & x\in\mathbb{R}^d. \end{cases} \tag{2.12}$$

Note how the introduction of the stochastic terms lead to the emergence of second order terms in HJB equation (2.12). The exact proof of Proposition 2 can be found in [114, Chapter 4, Proposition 3.5].

We would like to remind the reader that in Propositions 1 and 2 we made strong assumptions about the smoothness of the value function $v$. In the later sections we will investigate how the notion of a solution to the HJB equation can be relaxed in order to allow solutions which are not continuously differentiable.

### 2.1.2 Derivation of Isaacs equations

In this section we focus on a model for a two-person, zero-sum differential game. The basic idea is that two players control the dynamics of some evolving system. One of them tries to maximise, while the other tries to minimise, a cost functional that depends upon a trajectory. Our task is then to determine an optimal strategy of each of the players. This is in general a complicated task since strategy of one player depends largely on the strategy adopted by the other player. The dynamics of the optimal problem in the deterministic case are then

$$\begin{cases} \dot{x}(t) = b^{(\alpha(t),\beta(t))}(x(t)) \quad t \in (t_0, T], \\ x(t_0) = x_0. \end{cases} \tag{2.13}$$

We note that this is analogous to the HJB case but this time with a pair of controls $(\boldsymbol{\alpha},\boldsymbol{\beta}) \in A \times B$ where $A$ and $B$ are compact metric spaces. We assume that mapping $b : \mathbb{R}^d \times A \times B \to \mathbb{R}^d$ is continuous. We define the admissible control sets $\mathcal{A}, \mathcal{B}$ to be all measurable functions mapping from $[0,T]$ to $A$ and $B$ respectively. One player chooses $\boldsymbol{\alpha} \in \mathcal{A}$ and tries to maximise the cost functional while the other player chooses $\boldsymbol{\beta} \in \mathcal{B}$ and tries to minimise the cost functional. Analogously to the HJB case we denote the solution of (2.13) as $x_{(\boldsymbol{\alpha},\boldsymbol{\beta},t_0,x_0)}(t)$ and the associated cost functional $\mathcal{J}(t_0,x_0,\boldsymbol{\alpha},\boldsymbol{\beta})$.

The central notion of a well-defined two player game is that of non-anticipating strategy. Informally, given two almost identical choices of control for one of the players, we expect the other player's corresponding responses to be also almost identical. More precisely, we define the set of strategies for the first player as follows

$$\Gamma := \left\{ \boldsymbol{\alpha} : \mathcal{B} \to \mathcal{A} \,\middle|\, \text{ given } t > 0 \text{ and } \beta, \hat{\beta} \in \mathcal{B} \text{ if } \beta(s) = \hat{\beta(s)} \text{ a.e. } \forall s \leq t \text{ then } \alpha[\beta] = \alpha[\hat{\beta}] \text{ a.e. } \forall s \leq t \right\}.$$

The set of strategies for the second player is denoted as $\Delta$ and is defined analogously. This naturally leads us to the definitions of the lower value function

$$v^- := \inf_{\beta \in \Delta} \sup_{\boldsymbol{\alpha} \in \mathcal{A}} \mathcal{J}(t_0, x_0, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

and the upper value function

$$v^+ := \sup_{\boldsymbol{\alpha} \in \Gamma} \inf_{\beta \in \mathcal{B}} \mathcal{J}(t_0, x_0, \boldsymbol{\alpha}, \boldsymbol{\beta}).$$

To gain intuition about the meaning of the above definitions one may think about them in terms of a discretised temporal space. In the first case, the advantage is given to the second player as he responds to the actions of the first player and thus has more information at his disposal. The situation is reversed in the second case. In general, we have that $v^- \leq v^+$. In the specific case that $v^- = v^+$ we say that the game has a value. However, lower and upper value functions may differ significantly.

We will now list some of the properties of the lower and the upper value functions remarking that the proofs can be found in [41]. Rather than with the technical details we are concerned with pointing out the analogy to the HJB case which should be apparent.

Firstly, let us assume that for all $(\alpha,\beta) \in A \times B$ we have that $b^{(\alpha,\beta)}$, $f^{(\alpha,\beta)}$ are uniformly continuous, bounded and Lipschitz continuous in space and time and $v_T$ is bounded and Lipschitz continuous in space. Then we have the following result regarding the regularity of lower and upper value functions.

**Proposition 3.** *Both $v^-$ and $v^+$ are bounded and Lipschitz continuous in time and space*

The next result is the equivalent of the Dynamic Programming Principle stated in the previous section but this time in the Isaacs setting.

**Theorem 3.** *For any starting time $t_0 \leq T-h$ and starting position $x_0 \in \mathbb{R}^d$ we have that*

$$v^-(x_0,t_0) = \inf_{\beta\in\Delta} \sup_{\alpha\in\mathcal{A}} \int_{t_0}^{t+h} f^{(\alpha(s),\beta(s))}(s,x_{(\alpha,\beta)}(s))ds + v^-(t_0+h, x_{(\alpha,\beta)}(t_0+h)) \tag{2.14}$$

*and*

$$v^+(x_0,t_0) = \sup_{\alpha\in\Gamma} \inf_{\beta\in B} \int_{t_0}^{t+h} f^{(\alpha(s),\beta(s))}(s,x_{(\alpha,\beta)}(s))ds + v^+(t_0+h, x_{(\alpha,\beta)}(t_0+h)). \tag{2.15}$$

Consider now the final value problem of the following form

$$\begin{cases} -\partial_t v + H(\nabla v, x, t)v &= 0 \qquad (t,x) \in [0,T) \times \mathbb{R}^d, \\ v(T,x) &= v_T(x) \qquad x \in \mathbb{R}^d. \end{cases} \tag{2.16}$$

where either $H = H^-$ or $H = H^+$ with

$$H^-(p,x,t) := \inf_{\beta\in B} \sup_{\alpha\in A} \left(-b^{(\alpha,\beta)}(t,x)\cdot p - f^{(\alpha,\beta)}(t,x)\right)$$

and

$$H^+(p,x,t) := \sup_{\alpha\in A} \inf_{\beta\in B} \left(-b^{(\alpha,\beta)}(t,x)\cdot p - f^{(\alpha,\beta)}(t,x)\right).$$

We will refer to such problems as the upper and lower Isaacs equations. Note that by definition $H^+(p,x,t) \geq H^-(p,x,t)$ for any choice of $(p,x,t) \in \mathbb{R}^d \times \mathbb{R}^d \times [0,T)$. In the case when the equality holds we say that the Isaacs condition is satisfied.

The following result proves that the solutions of the upper and lower Isaacs equations are indeed the value function of the underlying optimal control problem. The exact proof can be found in the discussion following the statement of Corollary 2.5 in [106].

**Proposition 4.** *Let $v^- \in C^1(\mathbb{R}^d \times [0,T))$ be a lower value function of the control system* (2.13). *Then $v^-$ solves* (2.16) *with $H = H^-$. Similarly, $v^+ \in C^1(\mathbb{R}^d \times [0,T))$ solves* (2.16) *with $H = H^+$.*

We skip the details of the extension to the stochastic case which lead to the derivation of the second order Isaacs equation. The proof is analogous to the HJB case with the exception of the stochastic version of the Dynamic Programming Principles 2.14 and 2.15 which actually require the notion of a viscosity solution introduced in the next section. Reader interested in the detailed derivation is referred to [52].

## 2.2 Theory of viscosity solutions

As highlighted in the previous section, in the case of fully nonlinear problems the generalised partial differential equations of the form

$$F(x,u,Du,D^2u) = 0 \quad \text{in } \Omega \tag{2.17}$$

do not always admit classical solutions. Often we are forced to look for a solution among function which do not admit second or first order differentiation at all points of the domain. Therefore we would like to relax the notion of a solution in such a manner that the problem still admits the unique solution. One of the approaches is the idea of the multiplicative testing where integration by parts with smooth test functions is used to define weak derivatives. Given functions $u, v$ which are locally integrable on an open set $U$ and infinitely smooth test function $\phi$ we say that $v$ is a (first order) weak derivative of $u$ with respect to $x_i$ if and only if

$$\int_U u(x)\partial_i\phi(x)\mathrm{d}x = -\int_U v(x)u(x)\mathrm{d}x.$$

This idea extends to higher order derivatives and thus we no longer require $u$ to be differentiable. It is easy to sea that by replacing derivatives in (2.17) with their weak counterparts we increase the size of the potential solution space. This relaxes the notion of a solution and gives us a new set of candidate solutions to the problem. We refer to them as weak solutions.

However, given even a relatively simple non-linear problem, the notion of a weak solution obtained via the multiplicative testing is not sufficient. In this setting we may obtain multiple weak solutions which are almost everywhere classical solution to the problem and yet we do not have a way of selecting the correct one. In such case it is unclear how to tell apart correct and spurious solutions as we have no control over the PDE on null sets. Additionally, if the problem is posed in a non-divergence form, the integration by parts is impossible from purely calculational point of view. Hence we require an alternative method of obtaining information about the derivatives of non-smooth functions. Such a method is provided through the notion of viscosity solution, a term first introduced in 1980s in a series of papers by Crandall, Lions and Ishii. In this section we introduce all the basic notions required to work with viscosity solutions when designing numerical methods but the reader interested in more comprehensive overview is advised to read the summarizing work [25]. Throughout this section we follow this setting quite closely, making departures when necessary.

In order to define a viscosity solution we first need to restrict the set of allowed differential operators. While it may seem at first that it limits the scope of the theory, one can see in [25] that it still applies to a wide range of problems, most importantly including HJB and Isaacs equations. Consider the differential operator analogous to the one in (2.17) with $F : \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \times \mathcal{S}(d) \to \mathbb{R}$ where $\mathcal{S}(d)$ is the set of symmetric $d \times d$ matrices. We will require $F$ to satisfy two additional conditions. Firstly, we have that

$$F(x,r,p,X) \leq F(x,r,p,Y) \quad \text{whenever} \quad Y \leq X \tag{2.18}$$

where by $Y \leq X$ we mean that $X - Y$ is positive semi-definite. If the above condition is satisfied we say that $F$ is degenerate elliptic. One may think of such operators as family of PDEs which treat inequality in the derivatives in a consistent manner. Additionally, we require that

$$F(x,r,p,X) \leq F(x,s,p,X) \quad \text{whenever} \quad r \leq s. \tag{2.19}$$

If the above condition is satisfied and if additionally $F$ is degenerate elliptic, we say that $F$ is proper. We also assume that $F$ is continuous.

The main idea is to use the subtractive instead of the multiplicative testing. We will initially assume $u$ to be smooth in order to show a connection between a classical and a viscosity solution. Let us consider a proper differential operator $F$ and assume that $u$ is a classical subsolution of $F = 0$. We also have an infinitely smooth test function $\phi$ such that $u - \phi$ has local maximum at some $\hat{x}$. Then by basic calculus we have that $Du(\hat{x}) = D\phi(\hat{x})$ and $D^2u(\hat{x}) \leq D^2\phi(\hat{x})$. Using the properties of $F$ we have that

$$F(\hat{x}, u(\hat{x}), D\phi(\hat{x}), D^2\phi(\hat{x})) \leq F(\hat{x}, u(\hat{x}), Du(\hat{x}), D^2u(\hat{x})) \leq 0.$$

Note that this way we express abstract information about the derivatives of $u$ in terms of $\phi$ which is smooth. In order to obtain a similar comparison for non-smooth $u$ we need to relax the notion of the derivatives, similarly as in the case of the multiplicative testing.

We now allow $u$ to be non-smooth. We first observe that $u(x) \leq u(\hat{x}) - \phi(\hat{x}) + \phi(x)$ for $x$ in the vicinity of $\hat{x}$ and hence, by using Taylor series and letting $x \to \hat{x}$ we get

$$u(x) \leq u(\hat{x}) + \langle p, x - \hat{x} \rangle + \frac{1}{2}\langle X(x - \hat{x}, x - \hat{x}) \rangle + O((x - \hat{x})^3), \tag{2.20}$$

for some $p \in \mathbb{R}^d$ and $X \in \mathcal{S}(d)$. Note that the vector $p$ and the matrix $X$ could be replaced by $D\phi(x)$ and $D^2\phi(x)$ respectively. We also notice that if we allowed $u$ to be smooth then the choice $p = Du(\hat{x})$ and

$X = D^2u(\hat{x})$ would be valid as well. Thus we are interested in finding the set of all pairs $(p,X)$ for which the inequality (2.20) holds. More precisely, we consider region $\Omega$ such that $\hat{x} \in \Omega$. Then the set of all $(p,X)$ such that for any sequence $x_k \to \hat{x}$ the inequality (2.20) is satisfied, is called the superjet of $u$ at point $\hat{x}$ and is denoted by $J_{\Omega}^{2,+}u(x)$. Subjets are defined in an analogous manner but with direction of inequality in (2.20) reversed. We also remark that there exists an alternative definition of super- and subjets equivalent to the one presented above. It can be actually shown that

$$J_{\Omega}^{2,+}u(x) := \{(D\phi(\hat{x}), D^2\phi(\hat{x})) : \phi \in C^2(\Omega) \text{ and } u - \phi \text{ has a local maximum at } \hat{x}\}.$$

The subjets are defined analogously but with the maximum replaced by the minimum.

Having defined a relaxed version of the second order derivatives we are now interested in defining the notion of the viscosity solution of differential problem (2.17).

**Definition 1.** *Let differential operator $F$ be proper and $\Omega \subset \mathbb{R}^d$. Any function $u$ which is upper semicontinuous in $\Omega$ and satisfies*

$$F(x,u(x),p,X) \leq 0 \quad \forall x \in \Omega \quad \textit{and} \quad (p,X) \in J_{\Omega}^{2,+}u(x) \tag{2.21}$$

*is called a* viscosity subsolution *of* (2.17)*. Analogously, any function $u$ which is lower semicontinuous in $\Omega$ and satisfies*

$$F(x,u(x),p,X) \geq 0 \quad \forall x \in \Omega \quad \textit{and} \quad (p,X) \in J_{\Omega}^{2,-}u(x) \tag{2.22}$$

*is called a* viscosity supersolution *of* (2.17)*. If a function $u$ is simultaneously a viscosity sub- and supersolution of* (2.17)*, then we say it is viscosity solution of* (2.17)*.*

Note that using the alternative definition of sub- and superjets we can actually provide an equivalent definition without explicit mention of the jets. As this definition avoids the technical point of explaining the notion of jets it is often used in the literature. Hence, for the sake of completeness, we present it here.

**Definition 2.** *We call an upper semi-continuous function (lower semi-continuous) $u$ a viscosity subsolution (resp., supersolution) of* (2.17) *if, for any $\phi \in C^2(\mathbb{R} \times \mathbb{R}^d)$,*

$$F(x,\phi(x),D\phi(x),D^2\phi(x)) \leq 0 \quad (\textit{resp.}, \geq 0),$$

*provided that $u - \phi$ attains its maximum (resp., minimum) at $x \in \Omega$. We call $u$ a viscosity solution of* (2.17) *if it is simultaneously a viscosity sub- and supersolution of* (2.17)*.*

At this point we would also like to point out that there exists an alternative definition of a viscosity solution which allows a richer set of candidate solutions. More precisely, the potential viscosity solution is only required to be locally bounded instead of semicontinuous. The alternative definition requires the notion of a semicontinuous envelope defined as follows. For a function $z:\Omega \to \mathbb{R}$ the upper semi-continuous envelope $z^*$ is

$$z^* := \limsup_{r\to 0}\{u(y):\ y\in\Omega \text{ and } |y-x|\le r\}$$

and the lower semi-continuous envelope $z_*$ is

$$z_* := \liminf_{r\to 0}\{u(y):\ y\in\Omega \text{ and } |y-x|\le r\}.$$

Then we may define viscosity solution as follows.

**Definition 3.** *A locally bounded function u is called a viscosity subsolution (resp., supersolution) of* (2.17) *if for all $\phi \in C^2(\Omega)$ and all $x\in\Omega$ such that $u^*-\phi$ (resp., $u_*-\phi$) has a local maximum (resp., minimum) at x it holds that*

$$F_*(x,u^*(x),D\phi(x),D^2\phi(x)) \le 0 \tag{2.23}$$

*(resp.,*

$$F^*(x,u_*(x),D\phi(x),D^2\phi(x)) \ge 0). \tag{2.24}$$

*The function u is said to be a viscosity solution of* (2.17) *if it is both sub- and supersolution of* (2.17)*.*

As already mentioned, this definition of a viscosity solution is actually a relaxation of the first two in the sense that any function that satisfies the first two definitions also satisfies this one, but not the other way around. Indeed, it is easy to see that a semi-continuous function is locally bounded and its semicontinuous envelope is the function itself. We also point out that this definition allows us to consider discontinuous differential operators $F$. We remark that one has to be mindful of which definition one chooses to employ and that it remains the same throughout the argument. Failing to do so could potentially lead to erroneous results. One area where such distinction is important is the idea of comparison principle which we discuss in the next section.

## 2.3 Comparison Principle

Having defined the notion of a viscosity solution we would now like to show the uniqueness of the solution of the problem (2.17). However, in order to build a general theory of such proofs we require one more set of tools which are referred to as comparison principles. As it was the case with viscosity solutions, we motivate usefulness of this notion by first considering the case of smooth solutions. Let us consider $u,v\in C^2(\Omega)$ which are classical sub- and supersolution of (2.17) respectively, meaning that $F(x,u(x),Du(x),D^2u(x))\le 0$ and $F(x,v(x),Dv(x),D^2(x))\ge 0$. Let us also assume that $u\le v$ on the boundary of domain $\Omega$ denoted as $\partial\Omega$. Let us now consider the function $w:=u-v$ and let us assume that it has a local minimum at $\hat{x}\in\Omega$. By a basic calculus, this implies that $Du(\hat{x})=Dv(\hat{x})$ and $D^2u(\hat{x})\le D^2v(\hat{x})$. We can now use the degenerate ellipticity of $F$ to obtain the following inequality

$$F(x,u(\hat{x}),Du(\hat{x}),D^2u(\hat{x})) \leq 0 \leq F(x,v(\hat{x}),Dv(\hat{x}),D^2v(\hat{x})) \leq F(x,v(\hat{x}),Du(\hat{x}),D^2u(\hat{x})). \tag{2.25}$$

Due to the fact that $F$ is proper we can now conclude that $v(\hat{x}) \geq u(\hat{x})$ and therefore $w(\hat{x}) \leq 0$ at any local maximum in $\Omega$. Let us now assume that $u \geq v$ at some point in $\Omega$. However, this a contradiction since due to the fact that $u \leq v$ on $\partial\Omega$, this would imply a positive local maximum of $w(\hat{x})$ in $\Omega$. Hence we conclude that $u \leq v$ in $\overline{\Omega}$.

We are now interested in providing an analogous result for $u,v$ which are sub- and supersolution in a viscosity sense. Conceptually, we would like to replace the pairs $(Du(\hat{x}),D^2u(\hat{x}))$ and $(Dv(\hat{x}),D^2v(\hat{x}))$ with pairs belonging to sub- and superjets respectively. However, for non-differentiable functions there may exist points where jets are not large enough or even empty. Let us consider a function which is non-differentiable only on null sets. Then in the neighbourhood of any point at which such a function is non-differentiable, we can expect points where jets are in fact non-empty. This motivates us to define the closures of the jets which instead of considering values of $(p,X)$ in a pointwise sense, consider sequences of such pairs for any sequence $x_n \to x$. Here $x$ could possibly be at a point of non-differentiability and the closure of a jet will be non-empty even in such a case. More formally, we define the closure of a superjet as

$$\begin{aligned}\overline{J}^{2,+}_{\Omega}u(x) := \Big\{ (p,X) \in \mathbb{R}^d \times \mathcal{S}(d) \ : \ &\exists (x_n,p_n,X_n) \in \Omega \times \mathbb{R}^d \times \mathcal{S}(d) \ni \\ &(p_n,X_n) \in J^{2,+}_{\Omega}u(x_n) \text{ and } (x_n,u(x_n),p_n,X_n) \to (x,u(x),p,X) \Big\}.\end{aligned}$$

Note that for a continuous $F$, Definition 8 is still valid when jets are replaced with their closures.

We are now ready to present the main result of this section which is a comparison principle for the Dirichlet problem. Note that it is not possible to formulate a general comparison principle and it needs to be proven depending on the problem at hand. In this dissertation we only list and motivate necessary assumptions and present the final result. Reader interested in the exact proof is referred to the discussion preceding [25, Theorem 3.2] which makes explicit use of the closures of sub- and superjets.

**Assumption 1.** *There exists $\gamma > 0$ such that*

$$F(x,r,p,X) - F(x,s,p,X) \geq \gamma(r-s) \quad \textit{for } r \geq s.$$

Notice the similarities to the condition (2.19). One could think of it as assuming $F$ to be uniformly proper.

**Assumption 2.** *There exists a function $\omega : [o,\infty) \mapsto [0,\infty)$ such that $\lim_{x\downarrow 0} \omega(x) = 0$ and*

$$F(y,r,\alpha(x-y),Y) - F(x,r,\alpha(x-y),X) \leq \omega(\alpha|x-y|2 + |x-y|) \tag{2.26}$$

*whenever*

$$-3\alpha\begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} \leq \begin{pmatrix} X & 0 \\ 0 & -Y \end{pmatrix} \leq 3\alpha\begin{pmatrix} I & -I \\ -I & I \end{pmatrix}. \tag{2.27}$$

Again, note the similarity of (2.26) to (2.18). In fact, it is shown in [25] that former condition implies the latter. One interpretation is that (2.26) additionally ensures uniform continuity with respect to gradient. By multiplying (2.27) from the left and from the right by an arbitrary non-zero column vector $z$ we see that it is actually generalisation of the condition $Y \leq X$ as it imposes lower bound on the term $z^TXz - z^TYz$ apart from the usual upper bound of 0.

We now state the main result of this section.

**Theorem 4.** *Suppose that a differential operator $F$ is continuous and satisfies Assumptions 1 and 2. Consider $\Omega$ to be an open bounded subset of $\mathbb{R}^d$. Let $v \in LSC(\overline{\Omega})$, $u \in USC(\overline{\Omega})$ be a viscosity super- and subsolution of* (2.17) *respectively. Let $u \leq v$ on $\partial\Omega$. Then we have that $u \leq v$ on $\overline{\Omega}$.*

Note how the above result proves uniqueness of the viscosity solution. Indeed, assume that there are two distinct viscosity solutions $u$ and $v$. Then the comparison principle implies that $u \leq v$ and $v \leq u$ and hence the result follows. The next natural step would be to prove the existence of the viscosity solution via Perron's method which is indeed what is done in [25]. However, since our focus in this work is on the formulation of the numerical methods, an alternative approach is available due to [6]. The main result included there allows to prove the convergence of a numerical scheme to the unique viscosity solution of a Boundary Value Problem (BVP) in a very general framework. This approach allows us to bypass many technical difficulties posed by the study of the original PDE. It is discussed in detail in Section 2.5 but now we need to explain what is exactly meant by a viscosity solution of a BVP.

## 2.4 Boundary conditions

Let us consider boundary operator $B(x,u(x),p,X)$. Note that Hessian typically does not occur on the boundary as it conflicts with well-posedness of the problem so in most cases we use a boundary operator of the form $B(x,u(x),p)$. Let us also consider the BVP of the following form

$$G(x,u(x),Du(x),D^2u(x)) = \begin{cases} F(x,u(x),Du(x),D^2u(x)) & \text{in } \Omega, \\ B(x,u(x),Du(x),D^2u(x)) & \text{on } \partial\Omega. \end{cases} \tag{2.28}$$

Note that the above operator is in general discontinuous, and so we can no longer apply Definition 8 even if we use closures of jets. In order to gain intuition about why it is the case, let us consider a point $x \in \partial\Omega$. Recalling the definition of the closures of superjets, consider two sequences $x_n, y_n$ such that $\Omega \ni x_n \to x$ and

$\partial\Omega \ni y_n \to y$. Now pairs of $(p_n, X_n)$ which are in a certain sense generated by the sequence $x_n$ will satisfy the inequality $F(x_n, u(x_n), p_n, X_n) \leq 0$. Similarly, a sequence $(p_n, X_n)$ generated by $y_n$ will instead satisfy $B(y_n, u(y_n), p_n, X_n) \leq 0$. However when selecting a pair $(p, X) \in \overline{J}_{\overline{\Omega}}^{2,+} u(x)$ we have no way of knowing which sequence generated it and hence we can only say that either $F(x, u(x), p, X) \leq 0$ or $B(x, u(x), p, X) \leq 0$. This is not sufficient to satisfy the boundary condition in the pointwise sense, at least not in a general case. Instead we need to make use of Definition 3. We recall the notion of a semi-continuous envelope and use the fact that upper and lower semi-continuous envelopes of $G$ are defined as follows:

$$G^*(x, u(x), Du(x), D^2u(x)) = \begin{cases} F(x, u(x), Du(x), D^2u(x)) & \text{in } \Omega, \\ \max F(x, u(x), Du(x), D^2u(x)), B(x, u(x), Du(x), D^2u(x)) & \text{on } \partial\Omega, \end{cases}$$

$$G_*(x, u(x), Du(x), D^2u(x)) = \begin{cases} F(x, u(x), Du(x), D^2u(x)) & \text{in } \Omega, \\ \min F(x, u(x), Du(x), D^2u(x)), B(x, u(x), Du(x), D^2u(x)) & \text{on } \partial\Omega. \end{cases}$$

Note the presence of min and max operators. This allows us to formulate the following definition of a viscosity solution of (2.28).

**Definition 4.** *Let differential operators F and B be proper and $\Omega \subset \mathbb{R}^d$. We say that function u is a viscosity solution of* (2.28) *if it is a viscosity solution of $G = 0$ in $\overline{\Omega}$ in the sense of Definition 3.*

Note that we could equivalently use the statement of Definition 1 but replace $F$ with $G^*$, jets with their closures and $\Omega$ with $\overline{\Omega}$. We once again stress that the general boundary condition $B$ is not imposed in the strong, pointwise sense since it is not possible in general. Note that since the boundary condition is considered in a relaxed sense, jets at the boundary may become larger, potentially increasing the set of candidate solutions. When dealing with Dirichlet boundary conditions it is actually possible to impose them in the pointwise sense, by assuming the candidate solutions to be contained in a suitable function space. Note that in this case the pointwise and viscosity approach come with two different interpretations of the underlying optimal control problem. In the first case, we assume that the trajectory of the system terminates whenever boundary is reached and the final cost is incurred immediately. In the latter case, evolution of the system terminates only if it is optimal. Otherwise, the trajectory can return to the interior of the domain.

## 2.5 Convergence of numerical schemes in the Barles-Souganidis framework

Having discussed the notion of a solution we want to find, we need to be able to formulate a numerical scheme that is capable of selecting the viscosity solution among the set of candidate solutions. An abstract framework in which convergence of numerical schemes to viscosity solutions can be guaranteed was introduced in [6]. In this section we present the main result of this paper.

Let us consider an arbitrary mesh $\mathcal{T}_h$ discretising the domain $\overline{\Omega}$, where $h$ denotes the maximum mesh element diameter. We will study general numerical schemes of the form $S: \mathbb{R}_+ \times \overline{\Omega} \times \mathbb{R} \times B(\overline{\Omega}) \mapsto \mathbb{R}$, $(h,x,s,\phi) \mapsto S(h,x,s,\phi)$ where $x$ is some point in the closure $\overline{\Omega}$, $\phi$ is the numerical approximation of a viscosity solution and $s$ denotes the function value of the numerical approximation at $x$, i.e. $s = \phi(x)$. Note that at the moment definition of $s$ introduces redundancy into the notation, but it will actually be useful later. While noting that for a non-smooth function $u$ the value of operand $F(x,u(x),\nabla u(x),D^2u(x))$ is not defined, we conceptually want to formulate conditions for which

$$S(h,x,u(x),u) \approx hF(x,u(x),\nabla u(x),D^2u(x)).$$

We will now give abstract definitions of the conditions which have to be satisfied in order to ensure convergence of an abstract scheme $S$ to the unique viscosity solution of (2.28). Recall the notion of ellipticity of a differential operator $F$ which in a sense preserved ordering in the argument of the Hessian. We would like a similar "ordering principle" to apply to $S$. In order to achieve that we introduce the following notion of a monotone scheme.

**Definition 5.** *A numerical scheme $S$ is monotone if for all $h > 0$, $x \in \overline{\Omega}$, $s \in \mathbb{R}$ and $\phi, \psi \in B(\overline{\Omega})$*

$$S(h,x,s,\phi) \leq S(h,x,s,\theta) \text{ whenever } \phi \geq \theta$$

Since we want the numerical scheme $S$ to approximate a bounded function it is natural to expect the numerical solutions to be bounded for each refinement of the mesh. We would like to formalise this notion. Let us consider a subset $N_h$ of the domain $\overline{\Omega}$ such that $\forall r > 0, x \in \overline{\Omega}$ there exists $H > 0$ such that $\forall h \in (0,H) B(x,r) \cap N_h \neq \emptyset$. In other words, given a ball centred around any point in the domain, if we decrease $h$ sufficiently there will be a node of the mesh contained inside that ball.

**Definition 6.** *For every $h > 0$ there exists a solution $u_h \in B(\overline{\Omega})$ of*

$$S(h,x,u_h(x),u_h) = 0 \text{ for } x \in N_h,$$

*with $u_h|_{N_h} \in C(N_h)$ and an upper bound $\|u_h\|_{L^\infty(\overline{\Omega})}$ independent of $h$.*

Note that the above definition does not assume the uniqueness of a numerical solution nor the boundedness of all numerical solutions that satisfy the equation. We only require that for every $h$ at least one such solution exists and hence we can create a sequence of bounded solutions as $h$ decreases.

Conceptually, the consistency means that the schemes needs to be robust with respect to perturbation in all arguments. Moreover, the largest and smallest accumulation point of the sequence of numerical solutions has to be contained within upper and lower semicontinuous envelopes of (2.28). We formalise this statement by the following definition.

**Definition 7.** *The scheme $S$ is consistent if and only if for all $x \in \overline{\Omega}$ and $\phi \in C^\infty(\overline{\Omega})$*

$$\limsup_{h\to 0, N_h \ni y \to x, \xi \to 0} S(h, y, \phi(y)+\xi, \phi+\xi)/h \leq G^*(x, \phi(x), D\phi(x), D^2\phi(x)),$$

$$\liminf_{h\to 0, N_h \ni y \to x, \xi \to 0} S(h, y, \phi(y)+\xi, \phi+\xi)/h \geq G_*(x, \phi(x), D\phi(x), D^2\phi(x)).$$

Our final requirement is the existence of a comparison principle.

**Assumption 3.** *If $u$ is a viscosity subsolution and $v$ viscosity supersolution of BVP*

$$G(x, \phi(x), D\phi(x), D^2\phi(x)) = 0$$

*then $u \leq v$.*

Note that a specific proof of a comparison principle depends on the form of the underlying BVP and its formulation is in general a non-trivial task.

Having discussed all the building blocks we can now state the main result of this section. The exact proof can be found in [6, Theorem 2.1].

**Theorem 5.** *Assume that scheme $S$ is monotone and stable. Consider a BVP of the form*

$$G(x, u(x), Du(x), D^2u(x)) = 0 \quad \text{for } x \in \overline{\Omega}, \tag{2.29}$$

*which admits a comparison principle and such that $S$ is consistent. Then* (2.29) *has the unique viscosity solution $u$. Moreover, on each compact $K \subset \overline{\Omega}$ the solutions $u_h$ of $S$ converge on $N_h$ uniformly to $u$ as $h \to 0$:*

$$\lim_{h\to 0} \sup_{x \in N_h \cap K} |u(x) - u_h(x)| = 0.$$

Hence if we are able to construct a stable, monotone, consistent scheme and prove a comparison principle for a given BVP then, by Theorem 5, this proves convergence of this scheme to the unique viscosity solution of the BVP.

One final remark is that the original Barles-Souganidis argument treats boundary conditions in a viscosity sense, so in order to consider Dirichlet conditions in the pointwise sense, one needs to adapt the argument carefully. More precisely, it has to be shown that envelopes of the numerical solution satisfy the boundary conditions in the pointwise sense. Otherwise, application of a comparison principle may lead to incorrect results.

# Chapter 3

# Finite Element Methods for Bellman problems with the mixed boundary conditions

## 3.1 Introduction

This chapter extends results of [68] with the inclusion of mixed, fully nonlinear boundary conditions. More explicitly, we consider the numerical solution of HJB equations with mixed boundary conditions of the form:

$$-\partial_t v + \sup_{\alpha\in A}(L^\alpha \ v - f^\alpha) = 0 \qquad \text{in } [0,T)\times\Omega, \tag{3.1a}$$

$$-\partial_t v + \sup_{\alpha\in A}\left(L^\alpha_{\partial\Omega} v - g^\alpha\right) = 0 \qquad \text{on } [0,T)\times\partial\Omega_t, \tag{3.1b}$$

$$\sup_{\alpha\in A}\left(L^\alpha_{\partial\Omega} v - g^\alpha\right) = 0 \qquad \text{on } [0,T)\times\partial\Omega_R, \tag{3.1c}$$

$$v - g \ = 0 \qquad \text{on } [0,T)\times\partial\Omega_D, \tag{3.1d}$$

$$v - v_T \ = 0 \qquad \text{on } \{T\}\times\overline{\Omega}. \tag{3.1e}$$

Here $L^\alpha$ and $L^\alpha_{\partial\Omega}$ denote operators on the domain $\Omega$ and its boundary, respectively. The sets $\partial\Omega_t$, $\partial\Omega_R$ and $\partial\Omega_D$ form a decomposition of $\partial\Omega$. While leaving further details of the notation to the next section, it is already apparent how the basic fully nonlinear structure of the PDE operator, meaning the left-hand side of (3.1a), is mirrored in the Robin-type boundary conditions (3.1b) and (3.1c). But there is a crucial, additional complication of the boundary operators $L^\alpha_{\partial\Omega}$ - they will in general depend on the full gradient $\nabla v$ and not just on the tangential gradient $\nabla_{\partial\Omega} v$, meaning that $L^\alpha_{\partial\Omega} v$ cannot be evaluated with knowledge of $v|_{\partial\Omega}$ only.

Recalling the connection between optimal control and HJB equations, Bellman-type equations as in (3.1b) naturally arise on sections $\partial\Omega_t$ of the boundary. Indeed, the boundary condition (3.1b) expresses the possibility of controlling the particle or agent on the boundary through processes which are implicitly described by the $L^{\alpha}_{\partial\Omega}$. In contrast, Dirichlet conditions (3.1d) are appropriate for those section $\partial\Omega_D$ of $\partial\Omega$ where the possibility to control may cease in exchange for the reward or cost of $g_D$.

Boundary conditions of type (3.1c) arise from the Skorokhod control problem, which models particle reflection at the boundary [82, 87, 102]. Moreover, they have recently been used for the numerical solution of optimal transport problems in the setting of Monge-Ampère equations, where the transport boundary conditions are examined in Hamilton-Jacobi form [12, 73].

For the author the problem of primary interest is the Heston model of financial interest rates with uncertain market price of volatility risk discussed in Chapter 4. The Heston equation is most naturally posed on an unbounded domain, where already with certain market price of volatility risk it appears with mixed boundary terms corresponding to (3.1b), (3.1c) as well as (3.1d). All those types of boundary conditions remain when introducing uncertainty and when truncating the domain for the purposes of numerical approximation.

The aim of this chapter is to introduce a Finite Element Method capable of computing approximations to viscosity solutions for the aforementioned problems. The presented method permits degenerate diffusions. Boundary operators may exhibit discontinuities across face boundaries and where the type of boundary condition changes. A challenge for problems of this type is the discretisation of the first-order directional derivatives in (3.1b) and (3.1c) which is simultaneously consistent and monotone. On the one hand establishing monotonicity with an artificial diffusion approximating the Laplace-Beltrami operator of $\partial\Omega$ would not be sufficient because of the normal component in the directional derivatives of (3.1b) and (3.1c). On the other hand an artificial diffusion approximating the Laplace operator of $\Omega$ would not vanish under refinement due to different scaling of boundary and domain terms, thus leading to an inconsistent method. Our formulation is based on the observation that lower Dini directional derivatives exist for all functions in the P1 approximation space whenever the direction in question points out of the tangent cone of $\Omega$ at the position of interest.

A benefit of the Finite Element approach is that besides $L^\infty$ also $L^2(H^1)$ convergence can be established on unstructured meshes, as was shown in [67, 68] for the Dirichlet problem. The $L^2(H^1)$ convergence is for instance important for the above mentioned Heston model discussed in Chapter 4 as Delta hedging requires knowledge of partial derivatives of the value function.

The numerical analysis of HJB equations with Neumann and Robin conditions encompasses only few works. First results were provided by the Finite Difference community; we refer to the text book [82]. More recently, the transport boundary conditions of optimal transport were in [12] approximated with a filtered wide stencil scheme. In [1] a nonlinear Neumann boundary operator is approximated by extending the boundary into a strip of positive thickness, allowing the boundary conditions to be treated like a PDE operator. Within the finite element setting one line of research has developed around the approximation of Cordes solutions, in [54] with a mixed, non-conforming Finite Element Method while in [73] with a Discontinuous Galerkin Finite Element Method. Both [54] and [73] concentrate on the linear setting in non-divergence form. For a general review of the approximation of fully nonlinear equations with other types of boundary conditions we refer to [45, 90].

The structure of this chapter is as follows. In Section 3.2 we formulate the HJB problem with mixed boundary conditions. In Section 3.3 we define the numerical method. In Section 3.4 we prove monotonicity properties of the discretised operators. In Section 3.5 we show the existence and uniqueness of numerical solutions. In Sections 3.6 and 3.7 we establish consistency and stability, respectively, leading us to our main result of convergence in Section 3.8. Finally, we present numerical experiments in Section 3.9.

## 3.2 Mixed initial boundary value Bellman problem

In this section we introduce time-dependent Bellman equations with mixed boundary conditions. We consider a polytope domain $\Omega \subset \mathbb{R}^d$ with $d \geq 2$, i.e. a bounded, connected, closed domain, whose interior is non-empty and whose boundary is formed of flat faces. We allow $\Omega$ to be nonconvex. Let $\mathcal{F}_k$ denote set of open $k$-dimensional faces of $\Omega$ contained in $\partial\Omega$.

We consider Dirichlet and Robin boundary conditions on disjoint subsets of boundary $\partial\Omega$. Additionally, the region of the Robin boundary conditions breaks into two parts, one with and one without time derivative. We denote those three disjoint regions as $\partial\Omega_D$, $\partial\Omega_R$ and $\partial\Omega_t$, respectively. Therefore $\partial\Omega_D \cap \partial\Omega_R \cap \partial\Omega_t = \emptyset$ and $\partial\Omega_D \cup \partial\Omega_R \cup \partial\Omega_t = \partial\Omega$.

It is convenient to define the notion of a *generalised face* as the intersection of an $\omega' \in \mathcal{F}_{d-1}$ and a region linked to a boundary condition:

$$\mathcal{F} := \left\{ \omega \subset \partial\Omega : \, \omega = \omega' \cap \partial\Omega_X \text{ where } \omega' \in \mathcal{F}_{d-1}, \partial\Omega_X \in \{\partial\Omega_D, \partial\Omega_R, \partial\Omega_t\} \right\}$$

We assume that the boundary conditions are continuous on each $\omega$; however, discontinuities across (generalised) face boundaries may occur.

We introduce the normed space of piecewise continuous functions

$$PC(\partial\Omega, \mathbb{R}^k) := \{ g \in L^\infty(\partial\Omega, \mathbb{R}^k) : \, g|_\omega \in C(\omega, \mathbb{R}^k) \quad \forall \omega \in \mathcal{F} \},$$

equipped with the $L^\infty(\partial\Omega, \mathbb{R}^k)$ norm. If $k = 1$, we simply write $PC(\partial\Omega)$. We denote the standard inner product of $L^2(\Omega)$ and $L^2(\Omega, \mathbb{R}^d)$ by $\langle \cdot, \cdot \rangle$.

Let $A$ be a compact metric space, $\alpha \in A$ and let $L^\alpha$ be a linear operators of the following form:

$$L^\alpha : \overline{\Omega} \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R} \to \mathbb{R},\ (x,q,p,s) \mapsto -a^\alpha(x)\,q - b^\alpha(x) \cdot p + c^\alpha(x)\,s.$$

The interpretation as differential operator follows with $q = \Delta w(x)$, $p = \nabla w(x)$ and $s = w(x)$ for $w \in C^2(\overline{\Omega})$. The mapping

$$A \to C(\overline{\Omega}) \times C(\overline{\Omega}, \mathbb{R}^d) \times C(\overline{\Omega}) \times C(\overline{\Omega}), \alpha \mapsto (a^\alpha, b^\alpha, c^\alpha, f^\alpha),$$

is assumed to be continuous such that the families of functions $\{a^\alpha\}_{\alpha\in A}$, $\{b^\alpha\}_{\alpha\in A}$, $\{c^\alpha\}_{\alpha\in A}$ and $\{f^\alpha\}_{\alpha\in A}$ are equicontinuous. We require that $a^\alpha(x) \geq 0$ for all $\alpha \in A$ so that all $L^\alpha$ are degenerate elliptic. Frequently we abbreviate $L^\alpha(x, \Delta w(x), \nabla w(x), w(x))$ by $L^\alpha w(x)$ and $x \mapsto L^\alpha w(x)$ by $L^\alpha w$.

For $\alpha \in A$ the Robin operators $L^\alpha_{\partial\Omega}$ are defined as

$$L^\alpha_{\partial\Omega} : \partial\Omega \times \mathbb{R}^d \times \mathbb{R} \to \mathbb{R},\ (x,p,s) \mapsto -b^\alpha_{\partial\Omega}(x) \cdot p + c^\alpha_{\partial\Omega}(x)s. \tag{3.2}$$

with $b^\alpha_{\partial\Omega} \in PC(\partial\Omega, \mathbb{R}^d)$, $c^\alpha_{\partial\Omega} \in PC(\partial\Omega)$. The abbreviations $L^\alpha_{\partial\Omega} w(x)$ and $L^\alpha_{\partial\Omega} w$ are used analogously to $L^\alpha$.

We can now pose the Hamilton-Jacobi-Bellman (HJB) problem, whose numerical solution is the subject of this chapter:

$$\begin{aligned}
-\partial_t v + \sup_{\alpha\in A}(L^\alpha\ \ v - f^\alpha) &= 0 \qquad \text{in } [0,T) \times \Omega, && \text{(3.3a)}\\
-\partial_t v + \sup_{\alpha\in A}\big(L^\alpha_{\partial\Omega} v - g^\alpha\big) &= 0 \qquad \text{on } [0,T) \times \partial\Omega_t, && \text{(3.3b)}\\
\sup_{\alpha\in A}\big(L^\alpha_{\partial\Omega} v - g^\alpha\big) &= 0 \qquad \text{on } [0,T) \times \partial\Omega_R, && \text{(3.3c)}\\
v - g \ &= 0 \qquad \text{on } [0,T) \times \partial\Omega_D, && \text{(3.3d)}\\
v - v_T \ &= 0 \qquad \text{on } \{T\} \times \overline{\Omega}, && \text{(3.3e)}
\end{aligned}$$

with $g \in C(\partial\Omega)$, $g^\alpha \in PC(\partial\Omega)$, $v_T \in C(\overline{\Omega})$ and $T \in (0,\infty)$. The suprema are applied pointwise. An interpretation of (3.3) in the context of optimal control is given in the Appendix 3.A. Observe that the data terms $g^\alpha$ of the Robin conditions are $\alpha$-dependent, while the corresponding Dirichlet data $g$ are not. We assume equicontinuity of mapping

$$A \to PC(\partial\Omega, \mathbb{R}^d) \times PC(\partial\Omega) \times PC(\partial\Omega),\ \alpha \mapsto (b^\alpha_{\partial\Omega}, c^\alpha_{\partial\Omega}, g^\alpha)$$

in $\alpha \in A$. Additionally, we require the sign-conditions $v_T$, $c^\alpha_\Omega$, $c^\alpha_{\partial\Omega}$, $g$, $g^\alpha \geq 0$. It follows from continuity that

$$\sup_{\alpha\in A} \| (b^{\alpha}_{\partial\Omega}, c^{\alpha}_{\partial\Omega}, g^{\alpha}) \|_{L^{\infty}(\partial\Omega,\mathbb{R}^d)\times L^{\infty}(\partial\Omega)\times L^{\infty}(\partial\Omega)} < \infty. \tag{3.4}$$

We require $v_T$ to satisfy the Dirichlet boundary conditions on $\partial\Omega_D$.

It is useful to formulate the operator used in (3.3) more succinctly as

$$F(x,q,p,r,s) = \begin{cases} -r + \sup_{\alpha}\big(L^{\alpha}(x,q,p,s) - f^{\alpha}(x)\big) & \text{on } [0,T)\times\overline{\Omega}, \\ -r + \sup_{\alpha}\big(L^{\alpha}_{\partial\Omega}(x,p,s) - g^{\alpha}(x)\big) & \text{on } [0,T)\times\partial\Omega_t, \\ \sup_{\alpha}\big(L^{\alpha}_{\partial\Omega}(x,p,s) - g^{\alpha}(x)\big) & \text{on } [0,T)\times\partial\Omega_R, \\ s - g(x) & \text{on } [0,T)\times\partial\Omega_D, \\ s - v_T(x) & \text{on } \{T\}\times\overline{\Omega}. \end{cases}$$

We conclude the section with a definition of a viscosity solution similar to the setting of [6] which will be used throughout the chapter. To this end, let us consider a bounded function $v : [0,T]\times\overline{\Omega}\to\mathbb{R}$ and its upper and lower semi-continuous envelopes, defined respectively as

$$v^*(t,x) := \limsup_{\substack{(s,y)\to(t,x)\\(s,y)\in[0,T]\times\overline{\Omega}}} v(s,y)$$

and

$$v_*(t,x) := \liminf_{\substack{(s,y)\to(t,x)\\(s,y)\in[0,T]\times\overline{\Omega}}} v(s,y).$$

We analogously extend the definition of lower- and upper semicontinuous envelopes to $F$.

**Definition 8.** *A bounded function $v$ is a viscosity supersolution (respectively, subsolution) of* (3.3) *if, for any test function $\psi\in C^2(\mathbb{R}\times\mathbb{R}^d)$,*

$$F^*(x,\Delta\psi(t,x),\nabla\psi(t,x),\partial_t\psi(t,x),v_*(t,x)) \geq 0,$$

*(respectively,*

$$F_*(x,\Delta\psi(t,x),\nabla\psi(t,x),\partial_t\psi(t,x),v^*(t,x)) \leq 0,)$$

*provided that $v^*-\psi$ attains a local minimum (respectively, $v_*-\psi$ attains a local maximum) at $(t,x)\in[0,T]\times\overline{\Omega}$. Finally, we call $v:[0,T]\times\overline{\Omega}\to\mathbb{R}$ a viscosity solution of* (3.3) *if it is simultaneously a viscosity sub- and supersolution of* (3.3).

Notice how the above definition is conceptually close to the one in Definition 3, but we are now considering operators allowing time derivative term. Here the value of

$$F^*(x, \Delta\psi(t,x), \nabla\psi(t,x), \partial_t\psi(t,x), v_*(t,x))$$

should only depend on $\psi$'s restriction to $[0,T] \times \overline{\Omega}$. However, generally for lower-dimensional faces $F \in \mathcal{F}_{d-2}$ one finds test functions $\psi, \phi \in C^2(\mathbb{R} \times \mathbb{R}^d)$ with $\psi|_{[0,T]\times\overline{\Omega}} = \phi|_{[0,T]\times\overline{\Omega}}$ such that $\nabla\psi(t,x) \neq \nabla\phi(t,x)$ for $x \in F$.

We therefore demand that the coefficient $b^\alpha_{\partial\Omega}(x)$ of (3.2) belongs to the tangent cone:

$$b^\alpha_{\partial\Omega}(x) \in K(x) \qquad \forall\, \alpha \in A, x \in \partial\Omega. \tag{3.5}$$

Here, because of the polytopic nature of the domain, we define the tangent cone $K(x)$ as

$$K(x) := \left\{x' \in \mathbb{R}^d \,\middle|\, \exists \Lambda \in (0,\infty)\,\forall \lambda \in [0,\Lambda] : x + \lambda\, x' \in \overline{\Omega}\right\},$$

i.e. $x'$ is in the cone if there is a line segment from $x$ in the direction of $x'$ which is contained in $\overline{\Omega}$. Indeed, for $b^\alpha_{\partial\Omega}(x) \in K(x) \setminus \{0\}$ we observe how

$$-b^\alpha_{\partial\Omega}(x) \cdot \nabla\psi(t,x) = \partial_{-b^\alpha_{\partial\Omega}(x)}\psi(t,x) = \lim_{\substack{\lambda\to 0\\ \lambda>0}} \frac{\psi(t,x) - \psi(t, x + \lambda\, b^\alpha_{\partial\Omega}(x))}{\lambda} \tag{3.6}$$

is expressed only referring to $\psi$ on $[0,T] \times \overline{\Omega}$ and thus independently of $\psi$'s extension to $\mathbb{R} \times \mathbb{R}^d$. The limit on the right-hand side of (3.6) is known as the lower Dini derivative of $\psi$ in direction $-b^\alpha_{\partial\Omega}(x)$. On smooth sections of the boundary and at outward pointing corners the requirement (3.5) corresponds to an outflow condition, while at re-entrant corners (3.5) may permit an inflow term. In that sense, (3.5) is less restrictive than oblique boundary conditions such as [25, (7.35)]. We remark, however, that strengthened versions such as [25, (7.35)] may be necessary to ensure the existence of a comparison principle for the initial boundary value problem of the specific application of interest.

## 3.3 Numerical scheme

For the discretisation of (3.3) we consider a sequence $V_i, i \in \mathbb{N}$, of piecewise linear, simplicial, shape-regular finite element spaces. Let $\mathcal{T}_i$ be the mesh corresponding to the finite element space $V_i$. The boundary mesh $\mathcal{B}_i$ consists of the $(d-1)$-dimensional faces $F$ of elements $K \in \mathcal{T}_i$ with $F \subset \partial\Omega$. We make the assumption that $\mathcal{B}_i$ is subordinate to $\mathcal{F}$, i.e. every open set $F^*$ such that $\overline{F^*} \in \mathcal{B}_i$ is contained entirely in exactly one generalised face $\omega \in \mathcal{F}$.

Let $V_i^g \subset V_i$ be the affine subspace of functions which interpolate the Dirichlet boundary data on $\partial\Omega_D$ and $V_i^0 \subset V_i$ be the vector subspace of functions which interpolate 0 on $\partial\Omega_D$. The nodes of the finite element mesh are denoted by $y_i^\ell$. Here the index $\ell$ ranges over the nodes in the interior first, then the nodes on $\partial\Omega_t$, then $\partial\Omega_R$ and finally $\partial\Omega_D$. Therefore, $y_i^\ell \in \Omega$ for $\ell \leq N_i^\Omega$ for some $N_i^\Omega \in \mathbb{N}$ denoting number of interior nodes, $y_i^\ell \in \Omega \cup \partial\Omega_t$ for $\ell \leq N_i^t$ for some $N_i^t \in \mathbb{N}$ and, lastly, $y_i^\ell \in \Omega \cup \partial\Omega_t \cup \partial\Omega_R$ for $\ell \leq N_i := \dim V_i^g$. These nodes $y_i^\ell \in \Omega \cup \partial\Omega_t \cup \partial\Omega_R$ are called non-Dirichlet nodes.

The associated hat functions $\phi_i^\ell \in V_i$ are chosen so that $\phi_i^\ell(y_i^\ell) = 1$ while $\phi_i^\ell(y_i^s) = 0$ for $\ell \neq s$. Set $\hat{\phi}_i^\ell := \phi_i^\ell / \|\phi_i^\ell\|_{L^1(\Omega)}$. Therefore, the $\phi_i^\ell$ are normalised in the $L^\infty(\overline{\Omega})$ norm whilst the $\hat{\phi}_i^\ell$ are normalised in the $L^1(\overline{\Omega})$ norm.

The mesh size, i.e. the largest diameter of an element, is denoted $\Delta x_i$. It is assumed that $\Delta x_i \to 0$ as $i \to \infty$. The uniform time step size is denoted $h_i$ with the constraint that $T/h_i \in \mathbb{N}$. It is assumed that $h_i \to 0$ as $i \to \infty$. Let $s_i^k$ be the $k$th time step at the refinement level $i$. Then the set of time steps is $S_i := \{s_i^k : k = T/h_i, \ldots, 0\}$.

We introduce the operator $d_i$, which approximates the time derivative on $\Omega$ and $\partial\Omega_t$ but which is 0 on the remaining boundary. More precisely, we let the $\ell$th entry of $d_i w(s_i^k, \cdot)$ be

$$(d_i w(s_i^k, \cdot))_\ell = \begin{cases} \frac{w(s_i^{k+1}, y_i^\ell) - w(s_i^k, y_i^\ell)}{h_i} & \ell \leq N_i^t, \\ 0 & \text{otherwise.} \end{cases}$$

Observe how $(d_i w(s_i^k, \cdot))_\ell = 0$ for nodes $y_i^\ell \in \partial\Omega_R$ is consistent with the structure of (3.3c).

For each discretisation of (3.3) we allow a splitting of $L^\alpha$ and $L^\alpha_{\partial\Omega}$ into an explicit and an implicit part. For each $\alpha$ and for each $i$, we introduce the explicit operator $E^\alpha_{\Omega,i}$ and the implicit operator $I^\alpha_{\Omega,i}$ such that

$$E^\alpha_{\Omega,i} : C^2(\overline{\Omega}) \to C(\overline{\Omega}),\ w \mapsto -\bar{a}^\alpha_{\Omega,i} \Delta w - \bar{b}^\alpha_{\Omega,i} \cdot \nabla w + \bar{c}^\alpha_{\Omega,i} w,$$

$$I^\alpha_{\Omega,i} : C^2(\overline{\Omega}) \to C(\overline{\Omega}),\ w \mapsto -\bar{\bar{a}}^\alpha_{\Omega,i} \Delta w - \bar{\bar{b}}^\alpha_{\Omega,i} \cdot \nabla w + \bar{\bar{c}}^\alpha_{\Omega,i} w,$$

where $\bar{a}^\alpha_{\Omega,i}, \bar{\bar{a}}^\alpha_{\Omega,i}, \bar{c}^\alpha_{\Omega,i}, \bar{\bar{c}}^\alpha_{\Omega,i} \in C(\overline{\Omega})$ and $\bar{b}^\alpha_{\Omega,i}, \bar{\bar{b}}^\alpha_{\Omega,i} \in C(\overline{\Omega}, \mathbb{R}^d)$. Assumption 4 below shows that the explicit and implicit operators are chosen such that $I^\alpha_{\Omega,i} + E^\alpha_{\Omega,i}$ approximates $L^\alpha$. Analogously, we introduce non-negative $f_i^\alpha \in C(\overline{\Omega})$ which approximate $f^\alpha$.

The discretisations of $E^\alpha_{\Omega,i}$ and $I^\alpha_{\Omega,i}$ are the mappings from $V_i$ to $\mathbb{R}^{N_i}$ which are given by

$$(\mathsf{E}^\alpha_{\Omega,i} w)_\ell := \bar{a}^\alpha_{\Omega,i}(y_i^\ell) \langle \nabla w, \nabla \hat{\phi}_i^\ell \rangle + \langle -\bar{b}^\alpha_{\Omega,i} \cdot \nabla w + \bar{c}^\alpha_{\Omega,i} w, \hat{\phi}_i^\ell \rangle, \tag{3.7a}$$

$$(\mathsf{I}^\alpha_{\Omega,i} w)_\ell := \bar{\bar{a}}^\alpha_{\Omega,i}(y_i^\ell) \langle \nabla w, \nabla \hat{\phi}_i^\ell \rangle + \langle -\bar{\bar{b}}^\alpha_{\Omega,i} \cdot \nabla w + \bar{\bar{c}}^\alpha_{\Omega,i} w, \hat{\phi}_i^\ell \rangle, \tag{3.7b}$$

$$(\mathsf{F}^\alpha_{\Omega,i})_\ell := \langle f_i^\alpha, \hat{\phi}_i^\ell \rangle, \tag{3.7c}$$

where $\ell$ ranges over all internal nodes, i.e. $\ell \leq N_i^\Omega$. For boundary nodes $\ell > N_i^\Omega$ we set $(\mathsf{E}^\alpha_{\Omega,i} w)_\ell = (\mathsf{I}^\alpha_{\Omega,i} w)_\ell =$

$(\mathsf{F}^{\alpha}_{\Omega,i})_{\ell} = 0$. Because of the scaling of the $\hat{\phi}^{\ell}_{i}$, integration-by-parts gives for smooth $w$ and large $i$ that $\langle \nabla w, \nabla \hat{\phi}^{\ell}_{i} \rangle \approx -\Delta w(x)$ if $x \approx y^{\ell}_{i}$ away from the boundary.

Similarly, we define operators $\mathsf{E}^{\alpha}_{\partial\Omega,i}$ and $\mathsf{I}^{\alpha}_{\partial\Omega,i}$ on the boundary to discretise $L^{\alpha}_{\partial\Omega}$ as the sum of an explicit and implicit part. Starting point is the observation that the directional derivative $\partial_{-b^{\alpha}_{\partial\Omega}} w$ is well-defined in the sense of (3.6) for functions $w \in V_i$ even though $w$ is in general not continuously differentiable.

For $0 \leq \ell \leq N^{\Omega}_{i}$ we set $(\mathsf{E}^{\alpha}_{\partial\Omega,i} w)_{\ell} = (\mathsf{I}^{\alpha}_{\partial\Omega,i} w)_{\ell} = (\mathsf{F}^{\alpha}_{\partial\Omega,i})_{\ell} = 0$. More interestingly, for $N^{\Omega}_{i} < \ell \leq N_i$ ranging over the nodes of the Robin boundary conditions, we define the mappings from $V_i$ to $\mathbb{R}^{N_i}$ by

$$(\mathsf{E}^{\alpha}_{\partial\Omega,i} w)_{\ell} := \partial_{-\bar{b}^{\alpha}_{\partial\Omega,i}(y^{\ell}_{i})} w(y^{\ell}_{i}) + \bar{c}^{\alpha}_{\partial\Omega,i}(y^{\ell}_{i})\, w(y^{\ell}_{i}), \tag{3.8a}$$

$$(\mathsf{I}^{\alpha}_{\partial\Omega,i} w)_{\ell} := \partial_{-\bar{\bar{b}}^{\alpha}_{\partial\Omega,i}(y^{\ell}_{i})} w(y^{\ell}_{i}) + \bar{\bar{c}}^{\alpha}_{\partial\Omega,i}(y^{\ell}_{i})\, w(y^{\ell}_{i}), \tag{3.8b}$$

$$(\mathsf{F}^{\alpha}_{\partial\Omega,i})_{\ell} := g^{\alpha}_{i}(y^{\ell}_{i}), \tag{3.8c}$$

where $\bar{c}^{\alpha}_{\partial\Omega,i}, \bar{\bar{c}}^{\alpha}_{\partial\Omega,i}, g^{\alpha}_{i} \in PC(\partial\Omega)$ and $\bar{b}^{\alpha}_{\partial\Omega,i}, \bar{\bar{b}}^{\alpha}_{\partial\Omega,i} \in PC(\partial\Omega, \mathbb{R}^d)$. Here $\partial_{-\bar{b}^{\alpha}_{\partial\Omega,i}}$ and $\partial_{-\bar{\bar{b}}^{\alpha}_{\partial\Omega,i}}$ are understood as lower Dini derivatives as in (3.6).

On the Dirichlet boundary the mappings $\mathsf{E}^{\alpha}_{\partial\Omega,i}$, $\mathsf{I}^{\alpha}_{\partial\Omega,i}$ and $\mathsf{F}^{\alpha}_{\partial\Omega,i}$ implement nodal interpolation. For $\ell > N_i$ we set

$$(\mathsf{E}^{\alpha}_{\partial\Omega,i} w)_{\ell} := 0, \tag{3.9a}$$

$$(\mathsf{I}^{\alpha}_{\partial\Omega,i} w)_{\ell} := w(y^{\ell}_{i}), \tag{3.9b}$$

$$(\mathsf{F}^{\alpha}_{\partial\Omega,i})_{\ell} := g(y^{\ell}_{i}). \tag{3.9c}$$

We assume a fully implicit discretisation of the region $\partial\Omega_R$; additionally, suppose that $\bar{\bar{c}}^{\alpha}_{\partial\Omega,i}$ is chosen positive on $\partial\Omega_R$, even if $c^{\alpha}_{\partial\Omega} = 0$.

In summary, we require that the following assumption holds.

**Assumption 4.** *The coefficients satisfy*

$$\begin{aligned}\limsup_{i\to\infty} \sup_{\alpha\in A} \Big( &\sup_{0\leq\ell\leq N_i} \big\| a^{\alpha} - \big(\bar{a}^{\alpha}_{\Omega,i}(y^{\ell}_{i}) + \bar{\bar{a}}^{\alpha}_{\Omega,i}(y^{\ell}_{i})\big) \big\|_{L^{\infty}(\operatorname{supp} \hat{\phi}^{\ell}_{i})} \\ &+ \big\| b^{\alpha} - \big(\bar{b}^{\alpha}_{\Omega,i} + \bar{\bar{b}}^{\alpha}_{\Omega,i}\big) \big\|_{L^{\infty}(\Omega,\mathbb{R}^d)} + \big\| c^{\alpha} - \big(\bar{c}^{\alpha}_{\Omega,i} + \bar{\bar{c}}^{\alpha}_{\Omega,i}\big) \big\|_{L^{\infty}(\Omega)} \\ &+ \big\| f^{\alpha} - f^{\alpha}_{i} \big\|_{L^{\infty}(\Omega)} \Big) = 0\end{aligned}$$

*and*

$$\begin{aligned}\limsup_{i\to\infty} \sup_{\alpha\in A} \Big( &\big\| b^{\alpha}_{\partial\Omega} - \big(\bar{b}^{\alpha}_{\partial\Omega,i} + \bar{\bar{b}}^{\alpha}_{\partial\Omega,i}\big) \big\|_{L^{\infty}(\partial\Omega)} + \big\| c^{\alpha}_{\partial\Omega} - \big(\bar{c}^{\alpha}_{\partial\Omega,i} + \bar{\bar{c}}^{\alpha}_{\partial\Omega,i}\big) \big\|_{L^{\infty}(\partial\Omega)} \\ &+ \big\| f^{\alpha} - f^{\alpha}_{i} \big\|_{L^{\infty}(\partial\Omega)} + \big\| g - g_i \big\|_{L^{\infty}(\partial\Omega)} + \big\| g^{\alpha} - g^{\alpha}_{i} \big\|_{L^{\infty}(\partial\Omega)} \Big) = 0.\end{aligned}$$

*We require that the family*

$$\{(\bar{a}^{\alpha}_{\Omega,i}, \bar{b}^{\alpha}_{\Omega,i}, \bar{c}^{\alpha}_{\Omega,i}, \bar{b}^{\alpha}_{\partial\Omega,i}, \bar{c}^{\alpha}_{\partial\Omega,i}, \bar{\bar{a}}^{\alpha}_{\Omega,i}, \bar{\bar{b}}^{\alpha}_{\Omega,i}, \bar{\bar{c}}^{\alpha}_{\Omega,i}, \bar{\bar{b}}^{\alpha}_{\partial\Omega,i}, \bar{\bar{c}}^{\alpha}_{\partial\Omega,i}, f^{\alpha}_i, g^{\alpha}_i)\}_{\alpha \in A}$$

*is equicontinuous and depends continuously on $\alpha$. We impose $\bar{a}^{\alpha}_{\Omega,i} = \bar{c}^{\alpha}_{\Omega,i} = \bar{c}^{\alpha}_{\partial\Omega,i} = 0 \in \mathbb{R}$ and $\bar{b}^{\alpha}_{\Omega,i} = \bar{b}^{\alpha}_{\partial\Omega,i} = 0 \in \mathbb{R}^d$ as well as $\bar{\bar{c}}^{\alpha}_{\partial\Omega,i} > 0$ on the restriction to $\partial\Omega_R$, $i \in \mathbb{N}$.*

We define

$$\mathsf{E}^{\alpha}_i = \mathsf{E}^{\alpha}_{\Omega,i} + \mathsf{E}^{\alpha}_{\partial\Omega,i}, \quad \mathsf{I}^{\alpha}_i = \mathsf{I}^{\alpha}_{\Omega,i} + \mathsf{I}^{\alpha}_{\partial\Omega,i}, \quad \mathsf{F}^{\alpha}_i = \mathsf{F}^{\alpha}_{\Omega,i} + \mathsf{F}^{\alpha}_{\partial\Omega,i}.$$

We also use the notation $\mathsf{I}^{\alpha}_i, \mathsf{E}^{\alpha}_i$ and $\mathsf{F}^{\alpha}_i$ for the matrix representations of exactly these $\mathsf{I}^{\alpha}_i, \mathsf{E}^{\alpha}_i$ and $\mathsf{F}^{\alpha}_i$ with respect to the nodal basis $\{\phi^{\ell}_i\}_{\ell}$ for the trial functions. Moreover, we assume that the supremum operator is applied componentwise, i.e. $(\sup_{\alpha} v^{\alpha})_{\ell} = \sup_{\alpha} v^{\alpha}_{\ell}$ for $v \in \mathbb{R}^n$. The expression $a \lesssim b$ means that there exists a generic constant $C > 0$, independent of $i$ and $\alpha$, such that $a \leq Cb$. Relation $a \gtrsim b$ is defined analogously.

We can now state the numerical scheme used to approximate the solution of (3.3). We initialise the scheme by the nodal interpolation of $v_T$ so that $v_i(T, \cdot) \in V_i$. Then, in order to find the numerical solution $v_i(s^k_i, \cdot) \in V_i$, we proceed inductively over the remaining timesteps $k \in \{T/h_i - 1, \ldots, 1, 0\}$:

$$-d_i v_i(s^k_i, \cdot) + \sup_{\alpha \in A} \left( \mathsf{E}^{\alpha}_i v_i(s^{k+1}_i, \cdot) + \mathsf{I}^{\alpha}_i v_i(s^k_i, \cdot) - \mathsf{F}^{\alpha}_i \right) = 0. \tag{3.10}$$

We also use an alternative formulation of the numerical scheme. The matrices $\mathsf{E}^{k,w}_i$, $\mathsf{I}^{k,w}_i$ and $\mathsf{F}^{k,w}_i$ are constructed row-wise out of the matrices $\mathsf{E}^{\alpha}_i$, $\mathsf{I}^{\alpha}_i$ and $\mathsf{F}^{\alpha}_i$. More precisely, given a node $y^{\ell}_i$, timestep $s^k_i$ and a function $w(s^k_i, \cdot) \in H^1(\overline{\Omega})$, let $\hat{\alpha}$ be a maximiser of

$$\sup_{\alpha \in A} \left( \mathsf{E}^{\alpha}_i w(s^{k+1}_i, \cdot) + \mathsf{I}^{\alpha}_i w(s^k_i, \cdot) - \mathsf{F}^{\alpha}_i \right)_{\ell} = 0.$$

Note that choice of $\hat{\alpha}$ is not necessarily unique; the analysis is valid for any choice of such $\hat{\alpha}$. We also remark that solving maximisation problems of that form can be approached in multiple ways with differing computational cost and complexity. In numerical experiments throughout this work we use a method popular in the literature (see for example [42]) where maximisation is performed by comparing values over a finite subset of $A$. Although relatively simple, it poses a problem of choosing discretization of $A$ in such a way that any error incurred by it is negligible when compared to the approximation error of the numerical scheme. As pointed out in [69] alternative algorithms may be preferable in general.

Let the $\ell$th row of $\mathsf{E}^{k,w}_i$, $\mathsf{I}^{k,w}_i$ and $\mathsf{F}^{k,w}_i$ be equal to the $\ell$th row of $\mathsf{E}^{\hat{\alpha}}_i$, $\mathsf{I}^{\hat{\alpha}}_i$ and $\mathsf{F}^{\hat{\alpha}}_i$, respectively. In a non-ambiguous case we will omit explicit mention of $k$ and simply write $\mathsf{E}^{w}_i$, $\mathsf{I}^{w}_i$ and $\mathsf{F}^{w}_i$. We can now reformulate (3.10) using the newly constructed operators. We initialise the scheme with the interpolant $v_i(T, \cdot)$. Then

$v_i \in V_i$ for each $k \in \{T/h_i - 1, \dots, 1, 0\}$ and for $0 \leq \ell \leq N_i^t$ solves

$$\Big( (h_i \mathsf{I}_i^{k,v_i} + \mathsf{Id})\, v_i(s_i^k, \cdot) + (h_i \mathsf{E}_i^{k,v_i} - \mathsf{Id})\, v_i(s_i^{k+1}, \cdot) - h_i \mathsf{F}_i^{k,v_i} \Big)_\ell = 0 \tag{3.11a}$$

and for each $k \in \{T/h_i - 1, \dots, 1, 0\}$ and for $N_i^t < \ell$ solves

$$\Big( h_i \mathsf{I}_i^{k,v_i}\, v_i(s_i^k, \cdot) - h_i \mathsf{F}_i^{k,v_i} \Big)_\ell = 0, \tag{3.11b}$$

recalling the implicit discretisation on $\partial\Omega_R \cup \partial\Omega_D$, enforced through (3.9) and Assumption 4.

For the sake of convenience let us also introduce the operators $\hat{\mathsf{I}}_i^{k,v_i}$, $\hat{\mathsf{E}}_i^{k,v_i}$ and $\hat{\mathsf{F}}_i$ which combine spatial and temporal terms. The $\ell$th row of $\hat{\mathsf{I}}_i^{k,v_i}$, $\hat{\mathsf{E}}_i^{k,v_i}$ and $\hat{\mathsf{F}}_i$ is equal to that of $(h_i \mathsf{I}_i^{k,v_i} + \mathsf{Id})$, $(h_i \mathsf{E}_i^{k,v_i} - \mathsf{Id})$ and $h_i \mathsf{F}_i^{k,v_i}$, respectively, if $\leq \ell \leq N_i^t$. If $N_i^t < \ell$, the $\ell$th row is equal to $(h_i \mathsf{I}_i^{k,v_i})$, a zero vector and $h_i \mathsf{F}_i^{k,v_i}$, respectively. For a fixed control $\alpha$, the operators $\hat{\mathsf{I}}_i^{\alpha}$, $\hat{\mathsf{E}}_i^{\alpha}$ and $\hat{\mathsf{F}}_i^{\alpha}$ are constructed in an analogous manner. Then for all timesteps $s_i^k$ and each node $y_i^\ell$ solution $v_i$ of (3.11) solves also:

$$\hat{\mathsf{I}}_i^{k,v_i}\, v_i(s_i^k, y_i^\ell) + \hat{\mathsf{E}}_i^{k,v_i}\, v_i(s_i^{k+1}, y_i^\ell) - \hat{\mathsf{F}}_i^{k,v_i} = 0. \tag{3.12}$$

**Remark 1.** *To implement the lower Dini derivative $\partial_{-\bar{b}^{\alpha}_{\partial\Omega,i}(y_i^\ell)} w(y_i^\ell)$ in a computer code, we note that for $\lambda >$ $0$ sufficiently small there is an element $K \in \mathcal{T}_i$ whose closure contains both $y_i^\ell$ and $y_i^\ell + \lambda\, \bar{b}^{\alpha}_{\partial\Omega,i}(y_i^\ell)$. Indeed, choosing $\lambda$ smaller than the shortest edge length of the mesh achieves this (edge meaning* 1*-dimensional face). We then have*

$$\partial_{-\bar{b}^{\alpha}_{\partial\Omega,i}(y_i^\ell)} w(y_i^\ell) = \frac{w(y_i^\ell) - w(y_i^\ell + \lambda\, \bar{b}^{\alpha}_{\partial\Omega,i}(y_i^\ell))}{\lambda}.$$

*Importantly, because $w \in V_i$ is affine on $\overline{K}$, even without taking a limit $\lambda \to 0$ as on the right-hand side of* (3.6) *the Dini derivative is obtained exactly.*

## 3.4 Monotonicity

In this section we consider monotonicity properties of the discrete differential operators defined in the previous section. Monotonicity is crucial for proving the existence of a unique numerical solution of (3.10) as well as for establishing convergence to the viscosity solution.

**Definition 9.** *Let us consider $v \in V_i$ that has a local non-positive minimum at a node $y_i^\ell$. We say that an operator $F$ satisfies a Local Monotonicity Property (LMP) if for any such $v$ it follows that $(Fv)_\ell \leq 0$. Additionally, the operator $F$ satisfies the weak Discrete Maximum Principle (wDMP) provided that, for any $v \in V_i$,*

$$\big(Fv\big)_\ell \geq 0 \;\; \forall \ell \in \{1, \dots, N_i\} \implies \min_{\Omega \cup \partial\Omega_R \cup \partial\Omega_t} v \geq \min\{\min_{\partial\Omega_D} v, 0\}. \tag{3.13}$$

We now describe a method for choosing the artificial diffusion coefficients to impose the LMP on the matrices $\mathsf{I}_i^\alpha$ and $\mathsf{E}_i^\alpha$. It is based on the assumption of strict acuteness of the mesh. Consider an element $K \in \mathcal{T}_i$ with diameter $\Delta x_K$. For a bounded function $g : \overline{\Omega} \to \mathbb{R}^d$ we define $g$'s norm on the restriction to $K$ as

$$|g|_K := \Big(\sum_{j=1}^{d} \big\|g_j\big\|^2_{L^\infty(K)}\Big)^{\frac{1}{2}}.$$

Then by strict acuteness of the meshes we mean that there exists a $\theta \in (0, \frac{\pi}{2})$ such that the following holds:

$$\nabla\phi_i^\ell \cdot \nabla\phi_i^l\big|_K \leq -\sin(\theta)\,|\nabla\phi_i^\ell|_K\,|\nabla\phi_i^l|_K \qquad \forall \ell, l \leq N_i,\ \ell \neq l,\ \forall K \in \mathcal{T}_i. \tag{3.14}$$

We say that the family of meshes $\{\mathcal{T}_i\}_i$ is uniformly strictly acute if $\theta$ does not depend on $i$. As discussed in [16], for $d = 2$ and $d = 3$ the angle $\theta$ can be interpreted geometrically as $\frac{\pi}{2}$ minus the largest angle between the pairs of $(d-1)$-dimensional faces of the element $K$. For higher dimensions, we refer reader to [113] where a weakened but sufficient for the sake of our analysis mesh condition and its geometrical interpretation for arbitrarily large dimension is presented.

### 3.4.1 The LMP of $\mathsf{E}_{\Omega,i}^\alpha$, $\mathsf{E}_{\partial\Omega,i}^\alpha$, $\mathsf{E}_{\Omega,i}^\alpha$ and $\mathsf{I}_{\partial\Omega,i}^\alpha$

Let the functions $\tilde{a}_{\Omega,i}^\alpha, \tilde{\tilde{a}}_{\Omega,i}^\alpha, \bar{c}_{\Omega,i}^\alpha, \bar{\bar{c}}_{\Omega,i}^\alpha \in C(\overline{\Omega})$ and $\bar{b}_{\Omega,i}^\alpha, \bar{\bar{b}}_{\Omega,i}^\alpha \in C(\overline{\Omega}, \mathbb{R}^d)$ be given. These functions may be chosen freely as long as Assumption 4 holds. Conceptually $\tilde{a}_{\Omega,i}^\alpha + \tilde{\tilde{a}}_{\Omega,i}^\alpha \approx a^\alpha$ is the splitting of the second-order coefficients into explicit and implicit part *without* the addition of artificial diffusion. With the addition of artificial diffusion, the coefficients $\bar{a}_i^\alpha$ and $\bar{\bar{a}}_i^\alpha$ of Assumption 4 are obtained.

Indeed, as $\bar{b}_{\Omega,i}^\alpha, \bar{\bar{b}}_{\Omega,i}^\alpha, \bar{c}_{\Omega,i}^\alpha, \bar{\bar{c}}_{\Omega,i}^\alpha$ are bounded, we can select non-negative artificial diffusion coefficients $\bar{\nu}_{\Omega,i}^{\alpha,\ell}$ and $\bar{\bar{\nu}}_{\Omega,i}^{\alpha,\ell}$ so that we have for all interior nodes $y_i^\ell$ and mesh elements $K$ with $y_i^\ell$ as vertex that

$$\begin{aligned} |\bar{b}_{\Omega,i}^\alpha|_K + \Delta x_K \|\bar{c}_{\Omega,i}^\alpha\|_{L^\infty(K)} &\leq \bar{\nu}_{\Omega,i}^{\alpha,\ell}\,\sin(\theta)\,|\nabla\hat{\phi}_i^\ell|_K\,\mathrm{vol}(K), \\ |\bar{\bar{b}}_{\Omega,i}^\alpha|_K + \Delta x_K \|\bar{\bar{c}}_{\Omega,i}^\alpha\|_{L^\infty(K)} &\leq \bar{\bar{\nu}}_{\Omega,i}^{\alpha,\ell}\,\sin(\theta)\,|\nabla\hat{\phi}_i^\ell|_K\,\mathrm{vol}(K). \end{aligned} \tag{3.15}$$

Now, choosing $\bar{a}_{\Omega,i}^\alpha, \bar{\bar{a}}_{\Omega,i}^\alpha \in C(\overline{\Omega})$ such that

$$\bar{a}_{\Omega,i}^\alpha(y_i^\ell) \geq \max\{\tilde{a}_{\Omega,i}^\alpha(y_i^\ell), \bar{\nu}_{\Omega,i}^{\alpha,\ell}\}, \qquad \bar{\bar{a}}_{\Omega,i}^\alpha(y_i^\ell) \geq \max\{\tilde{\tilde{a}}_{\Omega,i}^\alpha(y_i^\ell), \bar{\bar{\nu}}_{\Omega,i}^{\alpha,\ell}\}, \tag{3.16}$$

we obtain our splitting of $L^\alpha$ into implicit and explicit part.

**Lemma 1.** *Suppose that the mesh $\mathcal{T}_i$ is strictly acute and that* (3.15) *holds. Then $\mathsf{E}_{\Omega,i}^\alpha$ and $\mathsf{I}_{\Omega,i}^\alpha$ satisfy the LMP for all $\alpha$.*

*Proof.* The argument from [68, Section 8] for the Dirichlet problem carries over unchanged for local minima at interior nodes of $v$ from Definition 9. At boundary nodes the LMP is trivially satisfied as $\mathsf{E}^{\alpha}_{\Omega,i}$ and $\mathsf{I}^{\alpha}_{\Omega,i}$ vanish there. □

We now turn to the monotonicity of the discrete boundary operators.

**Lemma 2.** *The operators $\mathsf{E}^{\alpha}_{\partial\Omega,i}$ and $\mathsf{I}^{\alpha}_{\partial\Omega,i}$ satisfy the LMP for all $\alpha$.*

*Proof.* Let $w \in V_i$ have a local non-positive minimum at a node $y_i^{\ell} \in \partial\Omega_t \cup \partial\Omega_R$. Then we find for the lower Dini derivative $\partial_{-\bar{b}^{\alpha}_{\partial\Omega,i}(y_i^{\ell})} w(y_i^{\ell}) \leq 0$. Also $\bar{c}^{\alpha}_{\partial\Omega,i}(y_i^{\ell})\, w(y_i^{\ell}) \leq 0$ because $c(y_i^{\ell}) \geq 0$. Hence $\mathsf{E}^{\alpha}_{\partial\Omega,i}$ admits the LMP. The argument for $\mathsf{I}^{\alpha}_{\partial\Omega,i}$ is analogous. □

### 3.4.2 Monotonicity properties of the $\mathsf{E}_i^{k,w}$, $\hat{\mathsf{E}}_i^{k,w}$, $\mathsf{I}_i^{k,w}$ and $\hat{\mathsf{I}}_i^{k,w}$

Having examined the basic building blocks of the numerical scheme in the previous two subsections, we can now analyse the monotonicity properties of the derived operators $\hat{\mathsf{E}}_i^{k,w}$ and $\hat{\mathsf{I}}_i^{k,w}$ as they appear in formulation (3.12) of the scheme. We summarise the assumptions made so far in the selection of the artificial diffusion coefficients.

**Assumption 5.** *Suppose that $\mathcal{T}_i$ is strictly acute and that* (3.15) *and* (3.16) *hold.*

First we examine the explicit terms.

**Lemma 3.** *Consider a fixed $w : S_i \times \overline{\Omega} \to \mathbb{R}$ such that $w(s_i^k, \cdot) \in V_i$ for all $s_i^k \in S_i$. Then the operators $v \mapsto \mathsf{E}_i^{k,w} v$ satisfy the LMP and their matrix has non-positive off-diagonal entries. For $h_i$ small enough, $\hat{\mathsf{E}}_i^{k,w}$ is monotone, i.e. all entries of the matrix representation are non-positive.*

*Proof.* For any $i$ and $\alpha$, $\mathsf{E}_i^{\alpha} = \mathsf{E}^{\alpha}_{\Omega,i} + \mathsf{E}^{\alpha}_{\partial\Omega,i}$ satisfies the LMP because its summands do. Let us consider a $v \in V_i$ that has a local non-positive minimum at a node $y_i^{\ell}$. There is an $\alpha \in A$ such that $(\mathsf{E}_i^{k,w} v)_{\ell} = (\mathsf{E}_i^{\alpha} v)_{\ell}$. We know $(\mathsf{E}_i^{\alpha} v)_{\ell} \leq 0$ and therefore that $v \mapsto \mathsf{E}_i^{k,w} v$ satisfies the LMP.

For $j \neq \ell$ the hat function $\phi_i^j$ attains a non-positive minimum at $y_i^{\ell}$. Thus, by the LMP, we have that $(\mathsf{E}_i^{k,w} \phi_i^j)_{\ell} \leq 0$. Hence all the off-diagonal entries of $\mathsf{E}_i^{k,w}$ are non-positive.

Owing to Assumption 1, the discretization on $\partial\Omega_R$ is fully implicit. Thus the rows of $\mathsf{E}_i^{k,w}$ belonging to the discretization on $\partial\Omega_R$ contain only zeros. Similarly, the rows linked to $\partial\Omega_D$ vanish, see (3.9). All other rows include a term arising from the time derivative; their structure is $(h_i \mathsf{E}_i^{k,v_i} - \mathsf{Id})$. Therefore, if $h_i$ is sufficiently small then $\hat{\mathsf{E}}_i^{k,w}$ is monotone. □

Now we turn to the implicit terms.

**Lemma 4.** *Consider a fixed $w : S_i \times \overline{\Omega} \to \mathbb{R}$ such that $w(s_i^k, \cdot) \in V_i$ for all $s_i^k \in S_i$. Then the operators $v \mapsto \mathsf{I}_i^{k,w} v$ satisfy the LMP. Moreover, the $v \mapsto \hat{\mathsf{I}}_i^{k,w} v$ satisfy the wDMP and their matrix representation restricted to $V_i^0$ are strictly diagonally dominant M-matrices.*

*Proof.* Analogously to the proof of Lemma 3, the $v \mapsto \mathsf{l}_i^{k,w} v$ satisfy the LMP and their off-diagonal entries are non-positive.

Before showing the wDMP we verify strict diagonal dominance. By construction, $v \equiv -1$ attains a non-positive local minimum at each node. Since $\mathsf{l}_i^{k,w}$ satisfies the LMP property, we have

$$0 \geq \left(\mathsf{l}_i^{k,w} v\right)_\ell = -\left(\mathsf{l}_i^{k,w}\right)_{\ell\ell} - \sum_{j \neq \ell} \left(\mathsf{l}_i^{k,w}\right)_{\ell j}. \tag{3.17}$$

As the off-diagonal entries of $\mathsf{l}_i^{k,w}$ are non-positive, we conclude the weak diagonal dominance of the $\mathsf{l}_i^{k,w}$:

$$\left(\mathsf{l}_i^{k,w}\right)_{\ell\ell} - \sum_{j \neq \ell} \left| \left(\mathsf{l}_i^{k,w}\right)_{\ell j} \right| \geq 0.$$

The rows of $\hat{\mathsf{l}}_i^{k,w}$ which discretise on $\Omega$ and $\partial\Omega_t$ are equal to the respective rows of the strictly diagonally dominant matrix $h_i \mathsf{l}_i^{k,w} + \mathsf{Id}$.

By Assumption 4 we have $\bar{\bar{c}}_{\partial\Omega,i}^{\alpha} > 0$ on $\partial\Omega_R$. Then

$$0 > \left(\mathsf{l}_i^{k,w} v\right)_\ell = \left(\hat{\mathsf{l}}_i^{k,w} v\right)_\ell \qquad \forall y_i^\ell \in \partial\Omega_R. \tag{3.18}$$

Using the same argument as above, but noting the strict inequality of (3.18) compared to (3.17), we conclude the strict diagonal dominance for rows linked to $\partial\Omega_R$. On $\partial\Omega_D$ the rows resemble an identity matrix, giving also strict diagonal dominance. It follows that the $\hat{\mathsf{l}}_i^{k,w}$ are invertible $M$-matrices because [13, Chapter 6, Theorem 2.3, $(M_{35})$] applies as $\hat{\mathsf{l}}_i^{k,w}$ is a $Z$-matrix.

Finally, consider a $v \in V_i$ with $\min_{\Omega \cup \partial\Omega_t \cup \partial\Omega_t} v < \min\{\min_{\partial\Omega_D} v, 0\}$. Let $y_i^\ell$ be a non-Dirichlet node, where the negative, global minimum of $v$ is attained. Since $\hat{\mathsf{l}}_i^{k,w}$ is a strictly diagonal dominant $M$-matrix it follows that $(\hat{\mathsf{l}}_i^{k,w} v)_\ell < 0$. Hence $\hat{\mathsf{l}}_i^{k,w}$ admits the wDMP. □

### 3.4.3 Scaling of the artificial diffusion coefficients

In order to achieve convergence of the numerical scheme we expect the artificial diffusion coefficients $\bar{\nu}_{\Omega,i}^{\alpha,\ell}$, $\bar{\bar{\nu}}_{\Omega,i}^{\alpha,\ell}$ to vanish in the limit $i \to \infty$.

Suppose that (3.14) holds uniformly for some $\theta$. In this subsection we suppose $\bar{\nu}_{\Omega,i}^{\alpha,\ell}$ are chosen quasi-optimally with regard to (3.15), meaning

$$\bar{\nu}_{\Omega,i}^{\alpha,\ell} \lesssim \sup\left\{ \frac{|\bar{b}_{\Omega,i}^{\alpha}|_K + \Delta x_K \|\bar{c}_{\Omega,i}^{\alpha}\|_{L^\infty(K)}}{\sin(\theta)\,|\nabla\hat{\phi}_i^\ell|_K \operatorname{vol}(K)} \,\middle|\, K \subset \operatorname{supp} \phi_i^\ell \right\} \tag{3.19}$$

$$\bar{\bar{\nu}}_{\Omega,i}^{\alpha,\ell} \lesssim \sup\left\{ \frac{|\bar{\bar{b}}_{\Omega,i}^{\alpha}|_K + \Delta x_K \|\bar{\bar{c}}_{\Omega,i}^{\alpha}\|_{L^\infty(K)}}{\sin(\theta)\,|\nabla\hat{\phi}_i^\ell|_K \operatorname{vol}(K)} \,\middle|\, K \subset \operatorname{supp} \phi_i^\ell \right\}. \tag{3.20}$$

Generally, in implementations of the algorithm quasi-optimally is more easily achieved than optimality.

Because of shape-regularity of the domain one has $|\nabla\hat{\phi}_i^\ell|_K \operatorname{vol}(K) \gtrsim \frac{1}{\Delta x_K}$. We conclude that quasi-optimal artificial diffusion coefficients satisfy

$$\mathsf{O}(\bar{\nu}_{\Omega,i}^{\alpha,\ell}) = \mathsf{O}(\bar{\bar{\nu}}_{\Omega,i}^{\alpha,\ell}) = \Delta x_K. \tag{3.21}$$

We now turn our attention to the time step restrictions imposed through the quasi-optimality (3.19). Recall that in order for the explicit operators to be monotone we require all their entries in matrix representation to be non-positive. This is satisfied trivially for nodes on $\partial\Omega_R \cup \partial\Omega_D$ where we use a fully implicit scheme. Therefore let us consider non-positivity of the diagonal terms of $h_i\mathsf{E}_i^\alpha - \mathsf{Id}$ on the complement $\Omega \cup \partial\Omega_t$. For $y_i^\ell \in \Omega$ this translates into the condition

$$1 \geq h_i \left(\bar{a}_{\Omega,i}^\alpha(y_i^\ell)\langle\nabla\phi_i^\ell,\nabla\hat{\phi}_i^\ell\rangle + \langle -\bar{b}_{\Omega,i}^\alpha \cdot \nabla\phi_i^\ell + \bar{c}_{\Omega,i}^\alpha\,\phi_i^\ell, \hat{\phi}_i^\ell\rangle\right)$$

and for $y_i^\ell \in \partial\Omega_t$

$$1 \geq h_i \left(\partial_{-\bar{b}_{\partial\Omega,i}^\alpha(y_i^\ell)}\phi_i^\ell(y_i^\ell) + \bar{c}_{\partial\Omega,i}^\alpha(y_i^\ell)\right).$$

Because

$$\begin{aligned}
\langle\nabla\phi_i^\ell,\nabla\hat{\phi}_i^\ell\rangle &= \mathsf{O}\big((\Delta x_K)^{-2}\big),\\
\langle\nabla\phi_i^\ell,\hat{\phi}_i^\ell\rangle &= \mathsf{O}\big((\Delta x_K)^{-1}\big),\\
\langle\phi_i^\ell,\hat{\phi}_i^\ell\rangle &= \mathsf{O}\big(1\big),\\
\partial_{-\bar{b}_{\partial\Omega,i}^\alpha(y_i^\ell)}\phi_i^\ell(y_i^\ell) &= \mathsf{O}\big((\Delta x_K)^{-1}\big),
\end{aligned}$$

we find $h_i = \mathsf{O}((\Delta x_K)^2)$ if $\|\bar{a}_{\Omega,i}^\alpha\|_\infty > 0$. Otherwise if $\|\bar{b}_{\Omega,i}^\alpha\|_\infty > 0$ or $\|\bar{b}_{\partial\Omega,i}^\alpha\|_\infty > 0$ we have $h_i = \mathsf{O}(\Delta x_K)$ and if $\bar{a}_{\Omega,i}^\alpha$, $\bar{b}_{\partial\Omega,i}^\alpha$ and $\bar{b}_{\Omega,i}^\alpha$ vanish but not $\bar{c}_{\Omega,i}^\alpha$ or $\bar{c}_{\partial\Omega,i}^\alpha$, then $h_i = \mathsf{O}(1)$. If also $\bar{c}_{\Omega,i}^\alpha = \bar{c}_{\partial\Omega,i}^\alpha = 0$ then there is no restriction on $h_i$, i.e. fully implicit discretisations are monotone for any $h_i > 0$.

## 3.5 Existence of numerical solutions

The discrete non-linear problem (3.12) can be solved by a version of Howard's algorithm discussed in [14]. We now present its formulation in our setting.

**Algorithm 1.** *Given are timestep $k \in \{0,\ldots,T/h_i - 1\}$, solution $v_i(s_i^{k+1},\cdot) \in V_i$ at timestep $k+1$ and an (arbitrary) choice of $\alpha \in A$. Find $w_0 \in V_i$ such that*

$$\hat{\mathsf{I}}_i^\alpha w_0 = \hat{\mathsf{F}}_i^\alpha - \hat{\mathsf{E}}_i^\alpha v_i(s_i^{k+1},\cdot).$$

*Inductively over $m \in \mathbb{N}$, compute $w_{m+1} \in V_i$ such that*

$$\hat{\mathsf{I}}_i^{w_m} w_{m+1} = \hat{\mathsf{F}}_i^{w_m} - \hat{\mathsf{E}}_i^{w_m} v_i(s_i^{k+1}, \cdot). \tag{3.22}$$

To show the convergence of the sequence $(w_m)_m$ to the solution of (3.10) we appeal to an auxiliary problem: for some fixed control $\alpha \in A$ we consider the linear evolution problem associated to it. More precisely, we define $v_i^\alpha \colon S_i \to V_i$ to be such that $v_i^\alpha(T, \cdot) = v_i(T, \cdot)$, the interpolant of $v_T$, and for each $k \in \{0, \dots, T/h_i - 1\}$

$$\hat{\mathsf{I}}_i^\alpha v_i^\alpha(s_i^k, \cdot) + \hat{\mathsf{E}}_i^\alpha v_i^\alpha(s_i^{k+1}, \cdot) - \hat{\mathsf{F}}_i^\alpha = 0. \tag{3.23}$$

Notice that $v_i^\alpha$ is well-defined due to the invertibility of $\hat{\mathsf{I}}_i^\alpha$.

**Theorem 6.** *There exists a unique numerical solution $v_i \colon S_i \to V_i$ which solves* (3.10) *and* (3.12)*. Algorithm 1, provided with the inputs $k$, $v_i(s_i^{k+1}, \cdot)$ and $\alpha$, generates a sequence $(w_m)_m$ which converges superlinearly to $v_i(s_i^k, \cdot)$ as $m \to \infty$. Moreover, $0 \le v_i \le v_i^\alpha$ for all $\alpha \in A$.*

*Proof.* For a fixed timestep $k$, the superlinear convergence of Algorithm 1 to the unique solution $v_i(s_i^k, \cdot)$ is shown in [14, Theorem 2.1] under Assumptions (H1) and (H2) stated therein. Condition (H1) requires the inverse positivity of the operators $\hat{\mathsf{I}}_i^{w_m}$, which holds because according to Lemma 4 every $\hat{\mathsf{I}}_i^{w_m}$ is a non-singular $M$-matrix. Condition (H2) requires that $\alpha \in A \mapsto -\hat{\mathsf{I}}_i^\alpha$ and $\alpha \in A \mapsto \hat{\mathsf{E}}_i^\alpha v_i(s_i^{k+1}, \cdot) - \hat{\mathsf{F}}_i^\alpha$ are continuous, which follows from Assumption 4. Induction over timesteps $k$ gives existence and uniqueness of the solution $v_i$.

We now show that $v_i \ge 0$ on $S_i \times \overline{\Omega}$ by induction over $k$. Firstly, we notice that $v_i(T, \cdot) \ge 0$ because we assumed that $v_T \ge 0$ on $\overline{\Omega}$ and the same holds for its interpolant. Let us assume $v_i(s_i^{k+1}, \cdot) \ge 0$ on $\overline{\Omega}$ for some $s_i^{k+1} \in S_i$. Due to the LMP, all entries of $\hat{\mathsf{E}}_i^{v_i}$ are non-positive and by assumption all entries of $\hat{\mathsf{F}}_i^{v_i}$ are non-negative. Therefore, using (3.12) we have that

$$\hat{\mathsf{I}}_i^{v_i} v_i(s_i^k, \cdot) = -\hat{\mathsf{E}}_i^{v_i} v_i(s_i^{k+1}, \cdot) + \hat{\mathsf{F}}_i^{v_i} \ge 0.$$

We conclude that $v_i(s_i^k, \cdot) \ge 0$ on $\overline{\Omega}$ due to the inverse positivity of $\hat{\mathsf{I}}_i^{v_i}$.

We now prove the last statement, namely $v_i \le v_i^\alpha$ for all $\alpha \in A$, by induction over $k$. Consider any $\alpha \in A$. At time $T$ both $v_i$ and $v_i^\alpha$ interpolate $v_T$ and hence are equal. Let us assume that for some $k \le T/h_i - 1$, $v_i(s_i^{k+1}, \cdot) \le v_i^\alpha(s_i^{k+1}, \cdot)$. From (3.10),

$$\hat{\mathsf{I}}_i^\alpha v_i(s_i^k, \cdot) \le \hat{\mathsf{F}}_i^\alpha - \hat{\mathsf{E}}_i^\alpha v_i(s_i^{k+1}, \cdot).$$

Now subtracting (3.23) from the above inequality, together with the monotonicity of $\hat{\mathsf{E}}_i^\alpha$, gives

$$\hat{\mathsf{I}}_i^\alpha \left( v_i(s_i^k, \cdot) - v_i^\alpha(s_i^k, \cdot) \right) \le \hat{\mathsf{E}}_i^\alpha \left( v_i^\alpha(s_i^{k+1}, \cdot) - v_i(s_i^{k+1}, \cdot) \right) \le 0.$$

Using the inverse positivity of $\hat{\mathsf{I}}_i^\alpha$ gives us $v_i(s_i^k,\cdot) - v_i^\alpha(s_i^k,\cdot) \le 0$ on $\overline{\Omega}$, as required. □

## 3.6 Consistency

We will assume existence of an elliptic projection $P_i$, described in [68], with the properties required in the following assumption.

**Assumption 6.** *There are linear mappings $P_i : C(H^1(\Omega)) \to V_i$ satisfying for all interior hat functions $\hat{\phi}_i^\ell$, $\ell \le N_i^\Omega$,*

$$\langle \nabla P_i w, \nabla \hat{\phi}_i^\ell \rangle = \langle \nabla w, \nabla \hat{\phi}_i^\ell \rangle. \tag{3.24}$$

*There is a constant $C \ge 0$ such that for every $w \in C^\infty(\mathbb{R}^d)$ and $i \in \mathbb{N}$,*

$$\|P_i w\|_{W^{1,\infty}(\Omega)} \le C \|w\|_{W^{1,\infty}(\Omega)} \quad \textit{and} \quad \lim_{i\to\infty} \|P_i w - w\|_{W^{1,\infty}(\Omega)} = 0. \tag{3.25}$$

To state consistency it is convenient to abbreviate the operator of the numerical scheme as

$$F_i\, w(s_i^k, y_i^\ell) := \hat{\mathsf{I}}_i^{k,w}\, w(s_i^k, y_i^\ell) + \hat{\mathsf{E}}_i^{k,w}\, w(s_i^{k+1}, y_i^\ell) - \hat{\mathsf{F}}_i \tag{3.26}$$

for $w(s_i^k,\cdot) \in V_i$. Note that while $F_i$ is the discrete operator approximating the continuous operator $F$ defined in section 3.2, the notationally similar $\hat{\mathsf{F}}_i$ represents the approximation of $f^w$, $g^w$ and $g$ as explained at the end of section 3.3.

**Theorem 7.** *Let $\psi \in C^2(\mathbb{R} \times \mathbb{R}^d)$, $s_i^{k(i)} \to t \in [0,T)$ and $y_i^{\ell(i)} \to x \in \overline{\Omega}$ as $i \to \infty$. Here $s_i^{k(i)}$ is a time step and $y_i^{\ell(i)}$ a node of the i-th refinement. Then*

$$\limsup_{i\to\infty} F_i P_i \psi(s_i^{k(i)}, y_i^{\ell(i)}) \le F^*(x, \Delta\psi(t,x), \nabla\psi(t,x), \partial_t \psi(t,x), \psi(t,x)) \tag{3.27}$$

*and*

$$\liminf_{i\to\infty} F_i P_i \psi(s_i^{k(i)}, y_i^{\ell(i)}) \ge F_*(x, \Delta\psi(t,x), \nabla\psi(t,x), \partial_t \psi(t,x), \psi(t,x)). \tag{3.28}$$

*Proof.* We prove (3.27). The result for (3.28) follows analogously. For ease of notation, the dependence of $k$ and $\ell$ on $i$ is made implicit.

*Step 1:* Standard Finite Difference bounds ensure that if $y_i^\ell \in \Omega \cup \partial\Omega_t$ then

$$\lim_{i\to\infty} d_i P_i \psi(s_i^k, y_i^\ell) = \partial_t \psi(t,x). \tag{3.29}$$

Otherwise, if $y_i^\ell \in \partial\Omega_R \cup \partial\Omega_D$ then

$$\lim_{i\to\infty} d_i P_i \psi(s_i^k, y_i^\ell) = 0. \tag{3.30}$$

*Step 2:* It is shown in [68, Section 4] that if $y_i^\ell \in \Omega$ then

$$\lim_{i\to\infty} \Big( \mathsf{E}_i^\alpha P_i \psi(s_i^\ell, \cdot) + \mathsf{I}_i^\alpha P_i \psi(s_i^k, \cdot) - \mathsf{F}_i^\alpha \Big)_\ell = L^\alpha \psi(t,x) - f^\alpha(x), \tag{3.31}$$

where convergence to the limit is uniform over all $\alpha \in A$. We remark that the orthogonality (3.24) is used in this step.

*Step 3:* Now suppose that $y_i^\ell \in \partial\Omega_D$. Then it follows from (3.25) that

$$\lim_{i\to\infty} F_i P_i \psi(s_i^k, y_i^\ell) = \psi(t,x) - g(x). \tag{3.32}$$

*Step 4:* Let $y_i^\ell \in \partial\Omega_t \cup \partial\Omega_R$. Just like the continuous operators the corresponding first-order terms of the discrete Robin operators employ the lower Dini derivative, giving consistency directly. Thus with

$$\lim_{i\to\infty} \Big| c_{\partial\Omega}^\alpha P_i \psi(t,\cdot) - \bar{\bar{c}}_{\partial\Omega,i}^\alpha P_i \psi(s_i^k,\cdot) - \bar{c}_{\partial\Omega,i}^\alpha P_i \psi(s_i^{k+1},\cdot) \Big| = 0, \tag{3.33}$$

using Assumptions 4 and 6, we conclude that if $y_i^\ell \in \partial\Omega_t \cup \partial\Omega_R$ then

$$\lim_{i\to\infty} \Big( \mathsf{E}_i^\alpha P_i \psi(s_i^\ell, \cdot) + \mathsf{I}_i^\alpha P_i \psi(s_i^k, \cdot) - \mathsf{F}_i^\alpha \Big)_\ell = L_{\partial\Omega}^\alpha \psi(t,x) - g^\alpha(x). \tag{3.34}$$

*Step 5:* Consider the sequence $\{(s_i^k, y_i^\ell)\}_i$ as specified in the statement of the theorem, in particular with $y_i^\ell \in \overline{\Omega}$. We decompose $\{(s_i^k, y_i^\ell)\}_i$ into the subsequences of the $(s_i^k, y_i^\ell)$ where $y_i^\ell$ belongs to $\Omega$, $\partial\Omega_t$, $\partial\Omega_R$ and $\partial\Omega_D$, respectively. Then the conclusions of Steps 1 to 4 above may be applied to the individual subsequences.

□

## 3.7 Stability

In this section we present a lemma which ensures $L^\infty$ stability of the numerical scheme (3.12). The stability statement goes back to the boundedness of a supersolution of the continuous linear problem for a fixed $\alpha$.

Let

$$F^\alpha(x,q,p,r,s) = \begin{cases} -r + L^\alpha(x,q,p,s) - f^\alpha(x) & \text{on } [0,T)\times\overline{\Omega}, \\ -r + L^\alpha_{\partial\Omega}(x,p,s) - g^\alpha(x) & \text{on } [0,T)\times\partial\Omega_t, \\ L^\alpha_{\partial\Omega}(x,p,s) - g^\alpha(x) & \text{on } [0,T)\times\partial\Omega_R, \\ s - g(x) & \text{on } [0,T)\times\partial\Omega_D, \\ s - v_T(x) & \text{on } \{T\}\times\overline{\Omega}. \end{cases}$$

**Assumption 7.** *There exists an $\alpha \in A$ and a $w(t,x) \in C^2(\mathbb{R}\times\mathbb{R}^d)$ which is a strict supersolution of the associated linear problem. More precisely, there is an $\varepsilon > 0$ such that*

$$F^\alpha(x,\Delta w(t,x),\nabla w(t,x),\partial_t w(t,x),w(t,x)) \geq \varepsilon$$

*on* $[0,T]\times\overline{\Omega}$.

The assumption is essentially fulfilled if the linear equation

$$F^\alpha(x,\Delta\psi(t,x),\nabla\psi(t,x),\partial_t\psi(t,x),\psi(t,x)) = 2\varepsilon \tag{3.35}$$

is a well-posed problem in a suitable sense. For example, $\psi$ may be a weak solution of (3.35) as in [92, Chapter 1] which admits a bounded extension to $\mathbb{R}\times\mathbb{R}^d$. In such cases one may pass to a strict supersolution in $C^2(\mathbb{R}\times\mathbb{R}^d)$ through mollification.

**Lemma 5.** *Let $\alpha$ be as in Assumption 7, $s_i^{k(i)} \to t \in [0,T)$ and $y_i^{\ell(i)} \to x \in \overline{\Omega}$ as $i\to\infty$. Here $s_i^{k(i)}$ is a time step and $y_i^{\ell(i)}$ a node of the i-th refinement. Then*

$$\liminf_{i\to\infty}\left[\hat{\mathsf{I}}_i^\alpha P_i w(s_i^k,y_i^\ell) + \hat{\mathsf{E}}_i^\alpha P_i w(s_i^{k+1},y_i^\ell) - (\hat{\mathsf{F}}_i^\alpha)_\ell\right] \geq \varepsilon.$$

*Proof.* We use Theorem 7 for a singleton control set $A=\{\alpha\}$ to cover the linear case. The result now follows because

$$(F^\alpha)_*(x,\Delta w(t,x),\nabla w(t,x),\partial_t w(t,x),w(t,x)) \geq \varepsilon,$$

owing to Assumption 7. □

**Theorem 8.** *The numerical solutions $v_i$ are uniformly bounded in the $L^\infty$ norm. More precisely, there exists* *a finite constant* $C > 0$ *such that*

$$\|v_i\|_{L^\infty(S_i \times \overline{\Omega})} \leq C \qquad \forall i \in \mathbb{N}. \tag{3.36}$$

*Proof.* Recall the solution $v_i^\alpha$ of the linear problem (3.23). We define

$$\begin{aligned} w_i^k &:= P_i w^\alpha(s_i^k, \cdot), \\ \tilde{v}_i^k &:= w_i^k - v_i^\alpha(s_i^k, \cdot). \end{aligned}$$

It is convenient to set

$$e_i^k := \hat{\mathsf{I}}_i^\alpha \tilde{v}_i^k + \hat{\mathsf{E}}_i^\alpha \tilde{v}_i^{k+1} = \hat{\mathsf{I}}_i^\alpha w_i^k + \hat{\mathsf{E}}_i^\alpha w_i^{k+1} - \hat{\mathsf{F}}_i^\alpha. \tag{3.37}$$

Because of Assumptions 6 and 7 the $w_i$ are uniformly bounded in the $L^\infty$ norm. Moreover, $0 \leq v_i \leq v_i^\alpha$ due to Theorem 6. Thus the statement of the theorem is proved once we demonstrate that the $v_i^\alpha$ are bounded from above independently of $i$. This is equivalent to showing a lower bound for the $\tilde{v}_i^k$.

It follows from Lemma 5 that $e^k \geq 0$ for $i$ larger than some constant $M$. It is trivial that there exists a constant $C$ such that the inequality of (3.36) holds for all $i \leq M$. We therefore may assume w.l.o.g. that $e^k \geq 0$ throughout. By possibly modifying $w$ through the addition of a positive constant we can assume that

$$w_i^{T/h_i} = P_i w(T, \cdot) \geq \|v_T(T, \cdot)\|_{L^\infty(\Omega)}$$

while maintaining the supersolution property of $w$ because $c_\Omega^\alpha, c_{\partial\Omega}^\alpha \geq 0$. This implies $\tilde{v}_i^{T/h_i} \geq 0$, i.e. the non-negativity at the final time.

Now suppose $\tilde{v}_i^{k+1} \geq 0$. Then

$$\tilde{v}_i^k = (\hat{\mathsf{I}}_i^\alpha)^{-1} \big(e_i^k - \hat{\mathsf{E}}_i^\alpha \tilde{v}_i^{k+1}\big) \geq 0$$

because also $(\hat{\mathsf{I}}_i^\alpha)^{-1} \geq 0$ and $-\hat{\mathsf{E}}_i^\alpha \tilde{v}_i^{k+1} \geq 0$. Now induction in $k$ completes the proof. □

## 3.8 Convergence

Analogously to the envelopes of functions introduced in Section 3.2 we define envelopes of the numerical solutions as follows

$$\overline{v}^*(t,x) = \sup_{(s_i^k, y_i^\ell) \to (t,x)} \limsup_{i \to \infty} v_i(s_i^k, y_i^\ell), \qquad \underline{v}_*(t,x) = \inf_{(s_i^k, y_i^\ell) \to (t,x)} \liminf_{i \to \infty} v_i(s_i^k, y_i^\ell)$$

where limits are taken over all sequences of nodes in $[0,T]\times\overline{\Omega}$ which converge to $(t,x)\in[0,T]\times\overline{\Omega}$. Owing to Theorem 8, $\overline{v}^*$ and $\underline{v}_*$ attain finite values. By construction, $\overline{v}^*$ is upper and $\underline{v}_*$ lower semi-continuous and $\underline{v}_*\leq\overline{v}^*$.

**Theorem 9.** *The function $\overline{v}^*$ is a viscosity subsolution and $\underline{v}_*$ is a viscosity supersolution.*

*Proof. Step 1 ($\overline{v}^*$ is a subsolution).* To show that $\overline{v}^*$ is a viscosity subsolution, suppose that $w\in C^\infty(\mathbb{R}\times\mathbb{R}^d)$ is a test function such that $\overline{v}^*-w$ has a strict local maximum at $(s,y)\in(0,T)\times\overline{\Omega}$, with $\overline{v}^*(s,y)=w(s,y)$. Note that $(s,y)$ may be on the boundary. Consider a closed neighbourhood $B:=\big\{(t,x)\in(0,T)\times\overline{\Omega}:|t-s|+|x-y|\leq\delta\big\}$ with $\delta>0$ such that

$$\overline{v}^*(s,y)-w(s,y)>\overline{v}^*(t,x)-w(t,x)\quad\forall(t,x)\in B\setminus(s,y).$$

Choose $i$ sufficiently large for $B$ to contain nodes. As in [68] we choose a sequence of nodes $\{(s^k_{i(j)},y^\ell_{i(j)})\}_j$ which maximise $v_i(s^\kappa_{i(j)},y^\lambda_{i(j)})-P_iw(s^\kappa_{i(j)},y^\lambda_{i(j)})$ among all nodes $(s^\kappa_{i(j)},y^\lambda_{i(j)})\in B$ and converge to $(s,y)$. It follows that

$$v_i(s^k_i,y^\ell_i)-P_iw(s^k_i,y^\ell_i)\to\overline{v}^*(s,y)-w(s,y)=0. \tag{3.38}$$

Moreover, because of $(s^k_i,y^\ell_i)\to(s,y)$, the neighbours of the $(s^k_i,y^\ell_i)$ eventually also belong to $B$: for $i$ sufficiently large, we have $(s^\kappa_i,y^\lambda_i)\in B$ if $\kappa\in\{k,k+1\}$ and $y^\lambda_i\in\operatorname{supp}\hat{\phi}^\ell_i$, in which case

$$\begin{aligned}v_i(s^\kappa_i,y^\lambda_i)-P_iw(s^\kappa_i,y^\lambda_i)&\leq v_i(s^k_i,y^\ell_i)-P_iw(s^k_i,y^\ell_i)\\ \Leftrightarrow\ P_iw(s^\kappa_i,y^\lambda_i)+\mu_i&\geq v_i(s^\kappa_i,y^\lambda_i),\end{aligned}$$

with $\mu_i=v_i(s^k_i,y^\ell_i)-P_iw(s^k_i,y^\ell_i)$, and $\mu_i\to0$ as $i\to\infty$ because of (3.38).

Recall that the matrices $\mathsf{E}^\alpha_i$ have non-zero off diagonal entries $(\mathsf{E}^\alpha_i)_{\ell\lambda}$ only if $y^\lambda_i\in\operatorname{supp}\hat{\phi}^\ell_i$ and that $v_i(s^{k+1}_i,\cdot)\leq P_iw(s^{k+1}_i,\cdot)+\mu_i$ on $\operatorname{supp}\hat{\phi}^\ell_i$. Therefore, monotonicity of $h_i\mathsf{E}^\alpha_i-\mathsf{Id}$ for all $\alpha\in A$ implies that

$$\Big((h_i\mathsf{E}^\alpha_i-\mathsf{Id})\Big[P_iw(s^{k+1}_i,\cdot)+\mu_i\Big]\Big)_\ell\leq\Big((h_i\mathsf{E}^\alpha_i-\mathsf{Id})v_i(s^{k+1}_i,\cdot)\Big)_\ell.$$

Applying the LMP and linearity of $\mathsf{I}^\alpha_i$ to $P_iw(s^k_i,\cdot)+\mu_i-v_i(s^k_i,\cdot)$, which has a non-positive local minimum at $y^\ell_i$, yields

$$\Big((h_i\mathsf{I}^\alpha_i+\mathsf{Id})\Big[P_iw(s^k_i,\cdot)+\mu_i\Big]\Big)_\ell\leq\Big((h_i\mathsf{I}^\alpha_i+\mathsf{Id})v_i(s^k_i,\cdot)\Big)_\ell.$$

From the definition of the scheme, with $\gamma := \sup_{\alpha,i} \|\bar{c}_i^\alpha + \bar{\bar{c}}_i^\alpha\|_\infty$,

$$\begin{aligned}
0 =& -d_i v_i(s_i^k, y_i^\ell) + \sup_{\alpha\in A}\Big(\mathsf{E}_i^\alpha v_i(s_i^{k+1},\cdot) + \mathsf{I}_i^\alpha v_i(s_i^k,\cdot) - \mathsf{F}_i^\alpha\Big)_\ell \\
\geq& -d_i\Big(P_i w(s_i^k, y_i^\ell) + \mu_i\Big) \\
&+ \sup_{\alpha\in A}\Big(\mathsf{E}_i^\alpha\Big(P_i w(s_i^{k+1},\cdot) + \mu_i\Big) + \mathsf{I}_i^\alpha\Big(P_i w(s_i^k,\cdot) + \mu_i\Big) - \mathsf{F}_i^\alpha\Big)_\ell \\
=& -d_i P_i w(s_i^k, y_i^\ell) \\
&+ \sup_{\alpha\in A}\Big[\Big(\mathsf{E}_i^\alpha P_i w(s_i^{k+1},\cdot) + \mathsf{I}_i^\alpha P_i w(s_i^k,\cdot) - \mathsf{F}_i^\alpha\Big)_\ell + \mu_i\langle \bar{c}_i^\alpha + \bar{\bar{c}}_i^\alpha, \hat{\phi}_i^\ell\rangle\Big] \\
\geq& -d_i P_i w(s_i^k, y_i^\ell) + \sup_{\alpha\in A}\Big(\mathsf{E}_i^\alpha P_i w(s_i^{k+1},\cdot) + \mathsf{I}_i^\alpha P_i w(s_i^k,\cdot) - \mathsf{F}_i^\alpha\Big)_\ell - \gamma|\mu_i| \\
=& F_i P_i w(s_i^k, y_i^\ell) - \gamma|\mu_i|\,. 
\end{aligned} \tag{3.39}$$

For a fixed $i$, evaluating $F_i P_i w(s_i^k, y_i^\ell)$ may involve a boundary operator even if $(s,y)$ is internal and vice versa may involve the PDE operator even if $(s,y)$ belongs to the boundary. Referring to the semi-continuous envelope $F_*$, it now follows from (3.39), $\lim_i \mu_i = 0$ and Theorem 7 that

$$\begin{aligned}
0 \geq& \liminf_{i\to\infty} F_i P_i w(s_i^k, y_i^\ell) \\
\geq& F_*(y, \Delta w(s,y), \nabla w(s,y), \partial_t w(s,y), \bar{v}^*(s,y)).
\end{aligned}$$

Therefore $\bar{v}^*$ is a viscosity subsolution.

*Step 2 ($\underline{v}_*$ is a supersolution).* Arguments similar to those above show that $\underline{v}_*$ is a viscosity supersolution, where the principal change to the proof is that one considers $w \in C^\infty(\mathbb{R}\times\mathbb{R}^d)$ such that $\underline{v}_* - w$ has a strict local minimum at some $(s,y) \in (0,T)\times\Omega$ with $\underline{v}_*(s,y) = w(s,y)$. With analogous notation, the last line in (3.39) corresponds to

$$0 \leq -d_i P_i w(s_i^k, y_i^\ell) + \sup_{\alpha\in A}\Big(\mathsf{E}_i^\alpha P_i w(s_i^{k+1},\cdot) + \mathsf{I}_i^\alpha P_i w(s_i^k,\cdot) - \mathsf{F}_i^\alpha\Big)_\ell + \gamma|\mu_i|\,,$$

i.e. there is a slight asymmetry in the argument due to the last sign in (3.39). Nevertheless, it is then deduced that

$$0 \leq F^*(x, \Delta w(t,x), \nabla w(t,x), \partial_t w(t,x), \underline{v}_*(t,x)).$$

Thus $\underline{v}_*$ is a viscosity supersolution. □

The above proof is an adaptation of the Barles-Souganidis argument [6] to the Finite Element setting, in line with that in [68] but differing in the treatment of the boundary conditions.

**Assumption 8.** *Let $\bar{v}$ be a lower semi-continuous supersolution and $\underline{v}$ be an upper semi-continuous subsolution. Then $\underline{v} \leq \bar{v}$.*

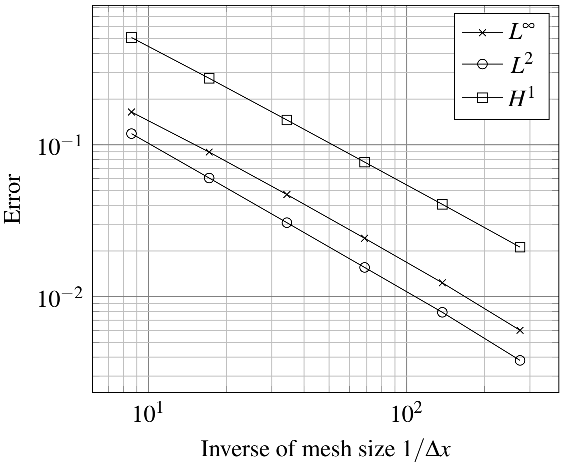


Figure 3.1: Approximation error of Experiment 1

**Theorem 10.** *One has $\underline{v}_* = \overline{v}^* = v$, where $v$ is the unique viscosity solution with $v(T,\cdot) = v_T$. Furthermore*

$$\lim_{i\to\infty} \|v_i - v\|_{L^\infty((0,T)\times\Omega)} = 0. \tag{3.40}$$

*Proof.* Follows as in the proof of Theorem 6.2 in [68]. □

## 3.9 Numerical experiments

The first experiment investigates rates of convergence for a known smooth solution. The remaining experiments examine the approximation of solutions with singularities near type changes of boundary conditions as well as the solution behaviour in the vicinity of nonlinear boundary conditions.

**Experiment 1 (Rates for smooth known solution):** We consider an IBVP on the square domain $\Omega = [-1,1]^2$ with Robin conditions on the right face $\partial\Omega_t = \{1\} \times (-1,1)$ and Dirichlet conditions on the remaining faces $\partial\Omega_D = \partial\Omega \setminus \partial\Omega_t$. We have the control set $A = [0,1]$ and the final time $T = 1$ for the system

$$\begin{aligned} -\partial_t v + \sup_{\alpha\in A}\left(-(\alpha + |x|^2/2)\Delta v + x v_x - f^\alpha\right) &= 0 && \text{in } [0,T)\times\Omega,\\ -\partial_t v + \sup_{\alpha\in A}\left(\alpha v_x - g^\alpha\right) &= 0 && \text{on } [0,T)\times\partial\Omega_t,\\ v \quad &= 0 && \text{on } [0,T)\times\partial\Omega_D,\\ v - (1-x^2)(1-y^2) &= 0 && \text{on } \{T\}\times\overline{\Omega}. \end{aligned} \tag{3.41}$$

We choose $g^\alpha$ and $f^\alpha$ such that

$$v(x,y,t) := t(1-x^2)(1-y^2) + (1-t)\sin(\pi x)\cos\left(\frac{\pi y}{2}\right)$$

is the exact solution of (3.41).

The artificial diffusion coefficients are chosen quasi-optimally, cf. Section 3.4.3. The time dependent Robin boundary condition is treated fully explicitly. The time step size is chosen to ensure monotonicity while permitting a large time step, leading to $O(h_i) = O(\Delta x_i)$. The $L^2$, $H^1$ and $L^\infty$ errors at time $t = 0$, presented also in Figure 3.1, obey in essence the same rates as those observed previously [68] with Dirichlet conditions and $O(h_i) = O(\Delta x_i)$ scaling:

| $\Delta x$ | $L^2$ | Rate | $L^\infty$ | Rate | $H^1$ | Rate |
|---|---|---|---|---|---|---|
| 0.1165 | 1.186e-1 | 0.98 | 1.645e-1 | 0.92 | 5.089e-1 | 0.93 |
| 0.0583 | 6.044e-2 | 0.98 | 8.960e-2 | 0.95 | 2.743e-1 | 0.94 |
| 0.0291 | 3.072e-2 | 0.99 | 4.706e-2 | 0.97 | 1.457e-1 | 0.95 |
| 0.0146 | 1.558e-2 | 0.99 | 2.426e-2 | 0.98 | 7.696e-2 | 0.95 |
| 0.0073 | 7.894e-3 | 1.04 | 1.234e-2 | 1.03 | 4.058e-2 | 0.96 |
| 0.0036 | 3.806e-3 | | 6.006e-3 | | 2.120e-2 | |

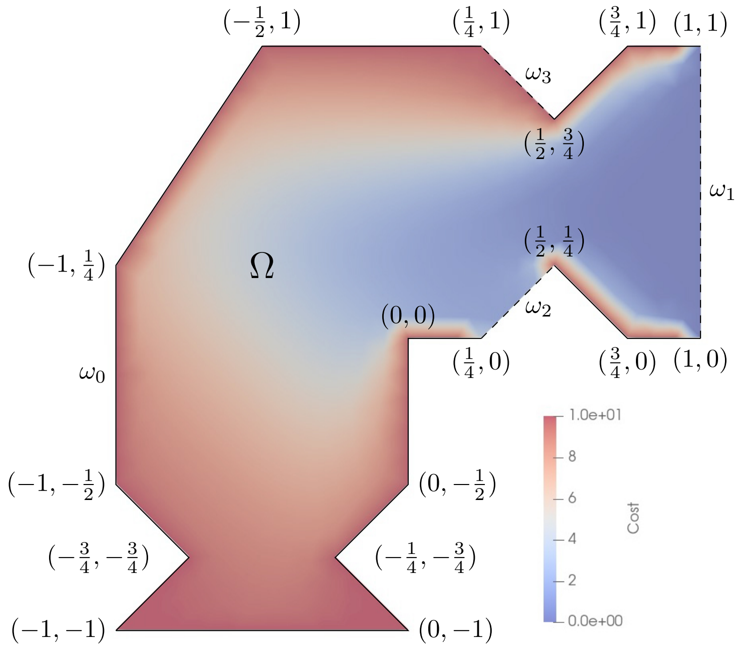


Figure 3.2: Value function of the Skorokhod problem for mesh size $\Delta x \approx 0.12$

**Experiment 2 (Skorokhod problem):** The second numerical experiment is set on a nonconvex, less regular domain, which is depicted in Figure 3.2. The stochastic controlled process is subject to a terminal cost of 10 everywhere apart from $\overline{\omega_1}$ where it is 0. There is no running cost. On $\omega_2$ the particle is transported through a Skorokhod reflection independently of the angle of incidence in the direction of the inner normal vector. Ultimately, a particle can avoid penalisation only by reaching $\overline{\omega_1}$ before the terminal time $T = 1$.

On $\Omega$ the particle may only choose between an upwards drift and drift to the right:

$$
\begin{aligned}
-\partial_t v + \sup_\alpha\,(-a^\alpha \Delta v - b^\alpha \cdot \nabla v) &= 0 && \text{in } [0,T)\times\Omega, && (3.42a)\\
-b_{\partial\Omega}\cdot\nabla v &= 0 && \text{on } [0,T)\times\omega_2 \cup \left\{\left(\tfrac{1}{4},0\right)\right\}, && (3.42b)\\
v &= 0 && \text{on } [0,T)\times\overline{\omega_1}, && (3.42c)\\
v &= 10 && \text{on } [0,T)\times\omega_0\cup\overline{\omega_3}\cup\left\{\left(\tfrac{1}{2},\tfrac{1}{4}\right)\right\}, && (3.42d)\\
v &= v_T && \text{on } \{T\}\times\overline{\Omega}, && (3.42e)
\end{aligned}
$$

where $a^\alpha = 0.1(1-x_2)\alpha$ and $b^\alpha = (-2\alpha, 2(\alpha-1))^T$ for $\alpha \in \{0,1\}$. Moreover, $b_{\partial\Omega} = (1,-1)^T$ and

$$
v_T(x) = \begin{cases} 10 & x \in \overline{\Omega}\setminus\omega_1, \\ 0 & x \in \omega_1. \end{cases} \tag{3.43}
$$

Hence when drifting to the right the particle is exposed to Brownian noise, while the equation is degenerate when the upward drift is selected. The numerical operators are given by

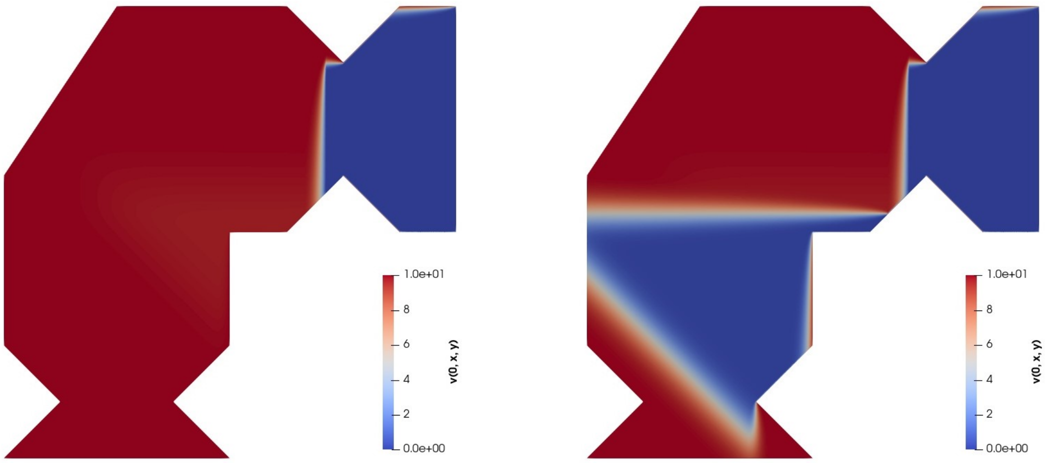
$$
\begin{aligned}
(\mathsf{E}^\alpha_{\Omega,i} v)_\ell &:= \bar{\nu}^{\alpha,\ell}_{\Omega,i}\langle\nabla v, \nabla\hat{\phi}^\ell_i\rangle + \langle -b^\alpha\cdot\nabla v, \hat{\phi}^\ell_i\rangle && (3.44a)\\
(\mathsf{I}^\alpha_{\Omega,i} v)_\ell &:= \max\left(a^\alpha - \bar{\nu}^{\alpha,\ell}_{\Omega,i}, 0\right)\langle\nabla v, \nabla\hat{\phi}^\ell_i\rangle. && (3.44b)\\
(\mathsf{E}^\alpha_{\partial\Omega,i} v)_\ell &:= 0, && (3.44c)\\
(\mathsf{I}^\alpha_{\partial\Omega,i} v)_\ell &:= \frac{v(t,x_1,x_2) - v(t,x_1-\lambda,x_2+\lambda)}{\lambda}. && (3.44d)
\end{aligned}
$$

Figure 3.2 shows the approximation $v_i$ at time $t = 0$ on a coarse mesh. Notice how the introduction of a Skorokhod type boundary gives the particle starting in the vicinity of $\omega_2$ a high probability of reaching the penalty-free exit zone $\overline{\omega_1}$. The node at $(\frac{1}{2},\frac{1}{4})$ already belongs to the Dirichlet boundary. We observe that the penalty of 10 in the boundary segment between $\omega_1$ and $\omega_2$ leads to a layer-like behaviour of the solution of only one element thickness. The related numerical experiments on finer meshes depicted in Figure 3.3 also exhibit this aspect of the numerical solution.

**Experiment 3 (Internal barrier and nonlinear boundary conditions):** The third experiment is an adaptation of the previous one to examine the effect of nonlinear boundary conditions. We break the adaptation into two parts.

*Part (a) (Internal barrier)*: The experiment is identical to the previous one with exception of the drift terms on $\Omega$:

$$
b^\alpha(x) = \begin{cases} \left(-2\alpha, 2(\alpha-1)\right)^T & : |x - \frac{3}{8}| > \frac{1}{20}, \\ \left(0, 2(\alpha-1)\right)^T & : \text{otherwise.} \end{cases}
$$



(a) Skorokhod problem

(b) Nonlinear boundary condition

Figure 3.3: Linear and nonlinear boundary conditions on $\omega_2$ with $\Delta x = 0.0035$.

This means that the strip of all $x$ with $|x - \frac{3}{8}| \leq \frac{1}{20}$ acts as barrier in $\Omega$: Within this strip there is no process with drift to the right. The only way a particle can cross the strip from the left to the right in order to avoid penalisation is by adopting $\alpha = 1$. In this case the particle *might* cross the barrier by means of diffusion; however, there is no drift term to aid the crossing.

The construction results in a value function which at first sight resembles a piecewise constant function. It is close to 10 left of the barrier as the particle is unlikely to reach the penalty-free exit zone $\omega_1$. It is mostly close to 0 right of the barrier; however, reaches 10 at Dirichlet boundary conditions on $\omega_0$ as already indicated in Experiment 2. At the barrier there is an internal layer arising from the possibility of crossing owing to diffusion.

This description of the value function is matched entirely by the numerical computations with the scheme of this chapter: see Figure 3.3(a), where the internal layer as well as the boundary conditions right of the barrier are well resolved.

*Part (b) (Nonlinear boundary condition)*: Now the boundary condition on $\omega_2$ is replaced by a nonlinear operator which corresponds to the choice between the previously used Skorokhod reflection and instantaneous transport along the boundary towards the right. In other words, (3.42b) is replaced by

$$\sup\{-b^0_{\partial\Omega} \cdot \nabla v, -b^1_{\partial\Omega} \cdot \nabla v\} = 0 \quad \text{on } [0,T) \times \omega_2 \cup \left\{ \left(\tfrac{1}{4}, 0\right) \right\}.$$

with $b^0_{\partial\Omega} = (1,-1)$ and $b^1_{\partial\Omega} = (-1,-1)$.

This modification has a striking impact on the behaviour of the controlled system. Now an optimally controlled particle may drift onto the boundary segment $\omega_2$ left of the barrier to be then transported on the boundary past the barrier. Right of the barrier the control is changed to $b^0_{\partial\Omega}$ in order to reflect the particle into $\Omega$ to avoid the penalty at the point $(\frac{1}{2}, \frac{1}{4})$ at the end of $\omega_2$.

An approximation of the resulting value function at time 0 is shown in Figure 3.3(b), which illustrates how this time regions left of the barrier have a value function close to zero since particles located there can now reach the penalty-free exit zone $\omega_1$. Indeed, when computing the solutions for earlier times $t < 0$, one observes further growth of the blue region as more time is available to arrive at $\omega_1$ before termination.

**Experiment 4 (Reflection vs. termination):** We consider an IBVP on the same domain, but now with a nonlinear boundary condition corresponding to a choice between a Skorokhod reflection and termination of the process in exchange for an oscillatory cost $g^\alpha$:

$$\begin{aligned}
-\partial_t v + \sup_\alpha\left(-a^\alpha \Delta v - b^\alpha \cdot \nabla v\right) &= 0 && \text{in } [0,T) \times \Omega, \\
\sup_\alpha\left(-b^\alpha_{\partial\Omega} \cdot \nabla v + c^\alpha_{\partial\Omega} v - g^\alpha\right) &= 0 && \text{on } [0,T) \times \omega_3 \cup \left\{\left(\tfrac{1}{4}, 1\right)\right\}, \\
v &= 0 && \text{on } [0,T) \times \overline{\omega_1}, \\
v &= 10 && \text{on } [0,T) \times \omega_0 \cup \overline{\omega_2} \cup \left\{\left(\tfrac{1}{2}, \tfrac{3}{4}\right)\right\}, \\
v &= v_T && \text{on } \{T\} \times \overline{\Omega},
\end{aligned}$$

where $\alpha \in \{0,1\}$, $v_T$ as in (3.43) and

$$\begin{aligned}
a^\alpha &= 0.2(1-x_2)(1-\alpha) + 0.2(1-x_1)\alpha, \\
b^\alpha &= (-2\alpha, 2(\alpha-1))^T, \quad b^\alpha_{\partial\Omega} = (\alpha,\alpha)^T, \\
c^\alpha &= (1-\alpha), \quad g^\alpha = -(10\cos(160x_1/\pi + 4)(1-\alpha).
\end{aligned}$$

Note that compared to the two previous examples the Robin type boundary has moved to $\omega_3$ while $\omega_2$ is part of the Dirichlet region with value 10. Both operators $L^\alpha$ now have regions of degeneracy, when either $x_1$ or $x_2$ is near 1.

The PDE operator on $\Omega$ is discretised according to (3.44a)–(3.44b), while the Robin operators are approximated implicitly, consistent with Assumption 4. The behaviour of the value function $v$ in the vicinity of $\omega_3$ is depicted in Figure 3.4. One observes how troughs of $g^0$ are attained by the value function, while near peaks of $g^0$ the numerical scheme switches to the reflection principle. Overall the experiment demonstrates how the framework of the chapter not only allows us to approximate nonlinear Robin conditions, but also incorporates a nonlinear switching between Robin conditions on the one hand and Dirichlet conditions on the other hand.

## 3.A Appendix: Interpretation of the mixed boundary conditions

We briefly sketch in a simplified setting how mixed boundary conditions can arise from an underlying optimal control problem. For a start we assume here that the solution $v$ of (3.3) is smooth. In optimal control

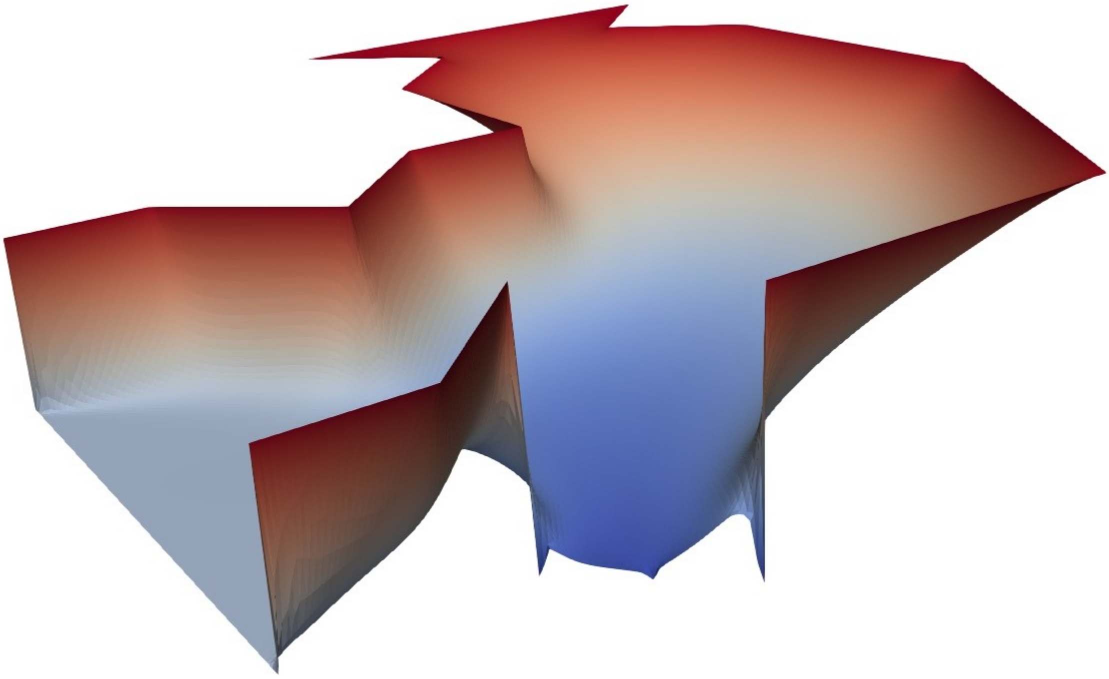


Figure 3.4: Value function with nonlinear boundary condition on $\omega_3$.

formulations the coefficients $c^\alpha$ and $c^\alpha_{\partial\Omega}$ are typically either 0 or they all coincide with some constant in order to model a discounting of cost. We shall assume the former.

We consider a particle, or agent, which occupies the state $\mathbf{x}(t) \in \overline{\Omega}$ at time $t \in [0,T]$. Its movements are described by the following rules:

1. Suppose $\mathbf{x}(t) \in \Omega$, $t < T$ and the control $\alpha \in A$ is selected. Then the particle's immanent movement is described by the SDE

$$\mathrm{d}\mathbf{x} = b^\alpha(\mathbf{x})\,\mathrm{d}t + \sqrt{2\,a^\alpha}\,\mathrm{d}W, \tag{3.45}$$

   where $W$ is a $d$-dimensional Brownian motion. While the particle follows (3.45) it is subject to the cost $f^\alpha \mathrm{d}t$.

2. Suppose $\mathbf{x}(t) \in \partial\Omega_D^\circ$, $t < T$ and the control $\alpha \in A$ is selected. Here $\partial\Omega_D^\circ$ refers to the interior of $\partial\Omega_D$ relative to $\partial\Omega$. Then, in the context of viscosity boundary conditions, the particle may either follow (3.45) at the running cost $f^\alpha \mathrm{d}t$ or terminate its movement at a cost of $g(\mathbf{x}(t))$. If instead pointwise Dirichlet conditions were imposed in Definition 8 then the boundary conditions would correspond to the guarantee that the particle terminates its movement.

3. Suppose $\mathbf{x}(t) \in \partial\Omega_R^\circ$, $t < T$ and the control $\alpha \in A$ is selected. Then the Skorokhod reflection principle may apply. Indeed, as described in [82, Sections 1.4, 3.1.3, ...], upon reaching the boundary the particle may instantaneously be transported a distance $b^\alpha_{\partial\Omega}\delta$ where $\delta > 0$ is small. Alternatively,

because of the context of viscosity boundary conditions, the particle may continue to follow (3.45). To remain within the scope of [82] we assume $g^\alpha = 0$ on $\partial\Omega_R$.

4. Suppose $\mathbf{x}(t) \in \partial\Omega_t^\circ$, $t < T$ and the control $\alpha \in A$ is selected. Then the particle may either move according to

$$\mathrm{d}\mathbf{x} = b^\alpha_{\partial\Omega}(\mathbf{x})\mathrm{d}t \tag{3.46}$$

with running cost $g^\alpha \mathrm{d}t$ or according to (3.45) with running cost $f^\alpha \mathrm{d}t$.

5. Suppose $t = T$ and the particle's movement has not yet terminated at a Dirichlet boundary then the final time cost $v_T(\mathbf{x}(T))$ incurs.

6. Suppose that $\mathbf{x}(t) \in \partial(\partial\Omega_D) \cup \partial(\partial\Omega_R) \cup \partial(\partial\Omega_t)$, $t < T$. Here the outer $\partial$ of $\partial(\partial\Omega_X)$ refers to the boundary of $\partial\Omega_X$ relative to $\partial\Omega$ where $X \in \{D,R,t\}$. Then the particle's behaviour may be selected from multiple of the above scenarios. E.g. if $\mathbf{x}(t) \in \partial(\partial\Omega_D) \cup \partial(\partial\Omega_R)$ then the particle movement may terminate, the particle may be reflected or it may be transported according to (3.45).

All these possible scenarios occur when representing uncertain market price of volatility risk in a Heston model found in Chapter 4.

We now link the above description of the particle by means of the value function to the HJB initial boundary value problem (3.3). Let $\alpha : [0,T] \to A$ represent a choice of controls for each time $s \in [0,T]$. Similarly, let $\xi_\Omega, \xi_{\partial\Omega_D}, \xi_{\partial\Omega_R}, \xi_{\partial\Omega_t} : [0,T] \times \overline{\Omega} \to \{0,1\}$ be indicator functions such that $\operatorname{supp} \xi_{\partial\Omega_X} \subset \overline{\partial\Omega_X}$ for $X \in \{D,R,t\}$, where $\overline{\partial\Omega_X}$ is the closure of $\partial\Omega_X$ relative to $\partial\Omega$. Furthermore,

$$\xi_\Omega + \xi_{\partial\Omega_D} + \xi_{\partial\Omega_R} + \xi_{\partial\Omega_t} \equiv 1.$$

Where $\xi_\Omega = 1$ the particle path $\mathbf{x}$ obeys (3.45), where $\xi_{\partial\Omega_D} = 1$ the particle terminates, where $\xi_{\partial\Omega_R} = 1$ the particle is reflected and where $\xi_{\partial\Omega_t} = 1$ the particle follows (3.46). Since the particle terminates where $\xi_{\partial\Omega_D} = 1$ we requite $\xi_{\partial\Omega_D}(s_1,x) = 1 \Rightarrow \xi_{\partial\Omega_D}(s_2,x) = 1$ for $s_1 \le s_2$. The value function $v$ at $(t,x)$ is the smallest cost realised among all possible choices for $\alpha$ and $\xi = (\xi_\Omega, \xi_{\partial\Omega_D}, \xi_{\partial\Omega_R}, \xi_{\partial\Omega_t})$:

$$\begin{aligned} v(t,x) = \inf_{\alpha,\xi} \mathbf{E}_{xt} \Big( \int_t^\tau & \xi_\Omega(\mathbf{x}(s))\, f^{\alpha(s)}(\mathbf{x}(s)) + \xi_{\partial\Omega_t}(\mathbf{x}(s))\, g^{\alpha(s)}(\mathbf{x}(s))\mathrm{d}s \\ & + \xi_{\partial\Omega_D}(\mathbf{x}(\tau))\, g(\mathbf{x}(\tau)) + \xi_\Omega(\mathbf{x}(\tau))\, v_T(\mathbf{x}(\tau)) \Big), \end{aligned}$$

where $\tau$ is the exit time from $[0,T) \times \Omega$ of $\mathbf{x}$ and $\mathbf{E}_{xt}$ the expectation conditional to $\mathbf{x}(t) = x$.

We fix some $\alpha, \xi$ which are not necessarily optimal. Suppose that $\mathbf{x}(s) \in \operatorname{supp} \xi_{\partial\Omega_t}$ for a short duration

$[t, t+\varepsilon) \ni s$. Then

$$g^{\alpha(t)}(\mathbf{x}(t)) = \lim_{h\to 0} \frac{1}{h} \int_t^{t+h} g^{\alpha(s)}(\mathbf{x}(s))\mathrm{d}s \geq -\lim_{h\to 0} \frac{v(t+h, \mathbf{x}(t+h)) - v(t,x)}{h}$$
$$= -\partial_t v(t,\mathbf{x}) - \nabla v(t,\mathbf{x}) \cdot \dot{\mathbf{x}} = -\partial_t v(t,\mathbf{x}) - \nabla v(t,\mathbf{x}) \cdot b^{\alpha}_{\partial\Omega}(\mathbf{x}).$$

Here the first equality follows from continuity, the inequality from the dynamic programming principle, the second equality from the chain rule and the third from (3.46).

Now, suppose that $\mathbf{x}(s) \subset \operatorname{supp} \xi_\Omega$ for a short duration $[t, t+\varepsilon) \ni s$. Then by a similar argument, detailed in [82], one finds

$$f^{\alpha(t)}(\mathbf{x}(t)) \geq -\partial_t v(t,\mathbf{x}) - \nabla v(t,\mathbf{x}) \cdot b^{\alpha}(\mathbf{x}) - a^{\alpha} \nabla v(t,\mathbf{x}).$$

On $\operatorname{supp} \xi_{\partial\Omega_D}$ we find $g(\mathbf{x}(t)) \geq v(t, \mathbf{x}(t))$ as the minimal cost cannot be more than the cost of termination.

Suppose now that the particle is located at $x \in \operatorname{supp} \xi_{\partial\Omega_R}$. It cannot be more beneficial for the particle to be at $x + b^{\alpha}_{\partial\Omega}\lambda$ as it will immanently be transported there. Thus $v(t,x) \leq v(t, x + b^{\alpha}_{\partial\Omega}\lambda)$ and therefore, with $\lambda \to 0$,

$$-b^{\alpha}_{\partial\Omega}(x) \cdot \nabla v(t,x) \leq 0.$$

When and where-ever the choice of $\alpha, \xi$ is optimal, the respective above inequality turns into an equality. With the compactness of $A$ and the continuous dependence of the coefficients on $\alpha$, such optimal controls exist. Therefore, taking suprema over $A$, one obtains that the value function solves (3.3), at least conceptually, with boundary conditions in the viscosity sense. We refer here to the viscosity sense because the use of semi-continuous envelopes in Definition 1 is interpreted as permitting (3.45) as transport law on all of the closure $\overline{\Omega}$ and as offering at the interfaces between boundary regions multiple boundary operators for the choice of the optimal strategy, like indicated in scenario 6 of the above list. We note that the choice between (3.45) and the various boundary operators will in general be subject to some delicate restrictions, arising from the sub- and superjets. At the boundary these jets are increased in size compared to their counterparts in the domain interior [25, Remark 2.7].

# Chapter 4

# Model and numerical solution of the Heston equation with the uncertain market price of volatility risk

## 4.1 Introduction

The main aim of this chapter is to present an application in finance of the method described in Chapter 3. One of the main challenges in financial mathematics is to determine the fair pricing of options. The classical tool used for that purpose has been Black-Scholes (BS) model, however it comes with certain limitations. Due to the non-realistic assumptions on the properties of the market, it fails to reliably predict the option price behaviour. One of the simplifications underlying the original BS model is the assumption that the volatility of the stock price is constant. Analysis of the real-life data does not support this statement and so numerous attempts were made to differently model the behaviour of the volatility in time. One such approach was proposed in [58] by Heston who modelled the volatility as another, correlated stochastic process, giving rise to the family of the stochastic volatility models. However this approach does not come without critiques. One of the main ones has been that it introduces new parameters like volatility of volatility and market price of volatility risk which may be even more difficult to estimate and predict in practice than the ones from the classical BS model. For instance, in [58] Heston assumes a linear scaling of the market price of volatility risk with variance and while there is an evidence of a positive correlation (see [37]) there does not seem to be a consensus on how to estimate the scaling factor (see [112]). Some authors (see for example [64], [53] and [81]), for the sake of simplicity and arguing that its impact is not significant, assume it to be equal to 0. However, this assumption does not seem to hold up in the realistic setting as discussed for example in [35] and [3]. The attempts of evaluating the market price of volatility in different financial settings can be found in [36] and [112] while its impact on option pricing is investigated in [37] and [88]. At this point it is worth noting that in fact all parameters used in the option pricing calculations are estimates based on the empirical or the historical data. Errors in those estimates can lead to inconsistent results even if the model were to be accurate. This inspired approach taken by Avellaneda, Levy & Parás in [2] where instead of trying to model and predict the behaviour of the parameters, they are assumed to stay within a given tolerance interval. This allows to manage the risk by considering the worst-case scenario leading in fact to an optimal control problem involving non-linear PDE as showed in [72]. This approach is used more specifically in the setting of European options pricing in [23]. In this chapter we aim to take a similar approach with respect to the market price of volatility risk.

Traditionally pricing of European options was achieved by either obtaining a closed form solution via analytical means or by simulation using binomial or Monte Carlo methods. However, in the stochastic volatility setting, the former approach is not always available, especially for more complex options, while the latter is computationally expensive. The other approach is to solve an associated stochastic PDE numerically. In order to address that issue, a number of numerical methods for option pricing emerged over the years. In this chapter we focus on European options, but we would like to point out that rich literature regarding American option pricing exists as well (see for example [119], [22], [62], [116] and [81]). The most common approach is the use of Finite Difference Methods, for example a High-Order Compact method in [108] and Alternating Direction Implicit methods in [65] and [38]. Another area of research is study of Discontinuous Galerkin methods proposed by [61] and recently revisited in [77]. Other areas of research include wavelet methods [93], Fourier methods [44], spectral methods [97] and radial basis methods [4]. As for the Finite Element Methods we refer to [103, Chapter 5] and the references therein. Such methods are discussed in [111] and [59], however no strict convergence results are presented. The main problem when trying to solve the PDE associated with the uncertain Heston model with stochastic volatility is that one requires method which can solve fully nonlinear PDEs with mixed boundary conditions. Such a method in Finite Element setting is provided in Chapter 3. The main advantage of using Finite Elements in this context is that it allows us to investigate convergence of the gradient. This is of particular importance in financial setting due to the role derivatives play in hedging a portfolio.

The outline of this chapter is as follows. In Section 4.2 we briefly state the uncertain Heston model and then show how it can be interpreted as a backward in time stochastic optimal control problem. By combining the methods of stochastic volatility and uncertain parameters we obtain second order non-linear PDE modelling the worst and best case scenarios when a range of values of market price of volatility risk is considered. In Section 4.3 we present a transformation of the Heston equation to the form required by the numerical scheme in Chapter 3. We also discuss the choice of the boundary conditions and the domain truncation for the numerical purposes. Finally, in Section 4.4 we present a case study of a long butterfly option whose main goal is to investigate the impact of the market price of volatility risk on the option price and its derivatives.

## 4.2 The uncertain Heston model

In this section we will consider an extension of the Heston model which includes uncertain parameters, allowing a reformulation into an optimal control problem. The main goal of the Heston model is to provide a theoretical estimate of option prices. An option is a financial contract offering its holder an opportunity to sell (put option) or buy (call option) an underlying asset (throughout this chapter assumed to be a stock) at a specific strike price $K$ at or before a specific time $T \geq 0$. Throughout this chapter we will focus on pricing of European options, which are options that can only be executed at a predefined point in time. At this point we are interested in formulating a model which will allow us to determine an option value $V$ which is fair from the standpoint of both the buyer and the seller. One such choice and the starting point of our discussion is the Heston model which is a modification of the Black-Scholes model. Both of those models will require us to make some further assumptions on the underlying assets and on the market.

In the Black-Scholes setting the change in time of the stock price $S$ is assumed to be represented by a geometric Brownian motion. Considering the Wiener process $W_1(t)$, we assume that we can represent the change of the stock price by the following stochastic differential equation:

$$dS(t) = \mu S(t)dt + \sigma S(t)dW_1(t). \tag{4.1}$$

The variable $\mu$ in (4.1) represents the drift (trend) and is defined to be an average change of the stock price $S$ per unit of time. Additionally, we assume it to be constant and known. The volatility $\sigma$ is the square root of the non-negative variance $v$ between returns from the same stock. In the classical Black-Scholes model $\sigma$ is assumed to be constant but we instead follow [58] and represent it by yet another, correlated Wiener process $W_2(t)$. Then, denoting the volatility of volatility as $\xi$ we obtain the second stochastic differential equation

$$dv(t) = \kappa(\gamma - v(t))dt + \xi\sqrt{v(t)}dW_2(t), \tag{4.2}$$

where we recall that variance $v = \sigma^2$ is a square of the volatility $\sigma$ and $\xi$ is assumed to be a known constant. We denote the correlation coefficient between $W_1$ and $W_2$ as $\rho \in (-1,1)$. Note that (4.2) is a mean-reverting process with a long term mean equal to $\gamma$ and a reversion level equal to $\kappa$.

At this point, we summarise the underlying assumptions of the model. The dividend payouts during the lifetime of an option are set to be 0. We assume it is possible to lend and borrow any amount of a riskless asset at a known constant risk-free interest rate $r$. Moreover, we are allowed to trade any amount, possibly fractional, of the stock $S$ or an option of value $V(t,S,v)$ at any time $0 \leq t \leq T$. We also say that the market is frictionless, which means that no such transaction generates fees. Lastly, we assume that there is no arbitrage possibility by which we mean that it is impossible to achieve an immediate and riskless profit.

Since the stock prices are in general non-deterministic, investment into stocks is inherently risky. Hence, the main idea of the portfolio management is to limit (or rather *hedge*) that risk by an appropriate selection of options, or more generally, financial derivatives. By modelling the change of the stock price $S$ and the volatility $\sigma$ as stochastic processes we introduced two sources of randomness into our model and we need to hedge both of them. In order to do that we consider the portfolio $P_\pi = P_\pi(t) \in \mathbb{R}$ containing the option with value $V$, a quantity $-\delta_1$ of the stock $S$ and a quantity $-\delta_2$ of another option with value $V^{'}$, i.e. $P_\pi := V - \delta_1 S - \delta_2 V^{'}$. By hedging the risk with the choice of $\delta_2 = \frac{\partial V/\partial v}{\partial V'/\partial v}$ and $\delta_1 = \frac{\partial V}{\partial S} + \delta_2 \frac{\partial V'}{\partial S}$ and following the argument in [110, Chapter 51] we conclude the following:

$$\frac{\partial V}{\partial t} + \frac{1}{2} v S^2 \frac{\partial^2 V}{\partial S^2} + S v \xi \rho \frac{\partial^2 V}{\partial S \partial v} + \frac{1}{2} v \xi^2 \frac{\partial^2 V}{\partial v^2} + (\kappa(\gamma - v) - \xi \lambda \sqrt{v}) \frac{\partial V}{\partial v} = 0, \tag{4.3}$$

for some function $\lambda(S, \sigma, t)$ which represents the market price of volatility risk. One now has to face the problem of choosing the $\lambda$. In [58] Heston suggests that it should be $\lambda\sigma$ with $\lambda \in \mathbb{R}$ being a scaling factor, i.e. in this special case $\lambda$ has the meaning of a coefficient and not the whole market price of volatility risk function. At this point one needs to either obtain an estimate of $\lambda$ based on the empirical data or assume it to be uncertain. Since there is no agreed upon method of estimating the market price of volatility, one may argue that any such estimate will be burdened with inaccuracies. We propose a new methodology to investigate how those estimation errors could affect the projected option price lending ideas from the uncertain volatility method discussed in [2]. However, instead of the volatility, we will model the market price of volatility risk as uncertain. In other words, we assume that $\lambda$ is an unknown parameter contained in some interval $L \subset \mathbb{R}$.

For all $\lambda \in L$ we define the linear operators $\mathcal{L}^\lambda$ as

$$\mathcal{L}^\lambda V := -\frac{1}{2} \left( S^2 v \frac{\partial^2 V}{\partial S^2} + 2 \rho \xi v S \frac{\partial^2 V}{\partial S \partial v} + \xi^2 v \frac{\partial^2 V}{\partial v^2} \right) - r S \frac{\partial V}{\partial S} - [\kappa(\gamma - v) - \xi \lambda \sqrt{v}] \frac{\partial V}{\partial v} + r V. \tag{4.4}$$

Consider the set $\mathbb{L}$ of all measurable mappings from $[0, T]$ to $L$. Then the *Heston equation* associated to each control $\boldsymbol{\lambda} \in \mathbb{L}$ is

$$-\partial_t V(t, S, v) + \mathcal{L}^{\boldsymbol{\lambda}(t)} V(t, S, v) = 0. \tag{4.5}$$

In order to make the above problem well-defined we still need to enforce the boundary and the final time

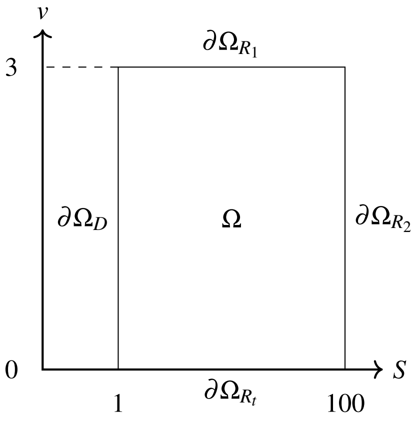


Figure 4.1: Domain $\Omega$ of the untransformed Heston problem

conditions. Throughout this chapter, given maturity $T$, we will use the following boundary conditions

$$V(S,v,T) = \Lambda(S), \tag{4.6}$$

$$V(0,v,t) = \Lambda(0), \tag{4.7}$$

$$\lim_{S\to\infty} \frac{\partial V}{\partial S}(S,v,t) = \lim_{S\to\infty} \frac{\partial \Lambda}{\partial S}(S), \tag{4.8}$$

$$\begin{aligned} -rS\frac{\partial V}{\partial S}(S,0,t) - \kappa\gamma\frac{\partial V}{\partial v}(S,0,t) + \\ rV(S,0,t) - \partial_t V(S,0,t) = 0, \end{aligned} \tag{4.9}$$

$$\lim_{v\to\infty} \frac{\partial V}{\partial v}(S,v,t) = 0. \tag{4.10}$$

where $\Lambda$ is the pay-off function of the option, see Remark 2.

The boundary condition (4.9) for a vanishing variance can be thought of as taking limit $v \to 0$ in Black-Scholes' formula and it is adapted directly from [58]. Notice on the other hand how compared to [58] the Dirichlet condition for a large volatility was replaced by the Neumann condition. We motivate this decision by the fact that as the volatility approaches extremely large values, influence of its oscillations on the option price is expected to be negligible. Thus we impose the change in $v$ direction on this boundary to be 0. In the literature such approach was adopted for pricing of American options in [22] and [63]. For more discussion about the boundary conditions in European and American option pricing see [117].

In practice, the implementation of the numerical scheme will require us to truncate the domain. Generally we choose a rectangular domain $\Omega$. For example, in the setting of the case study presented below we let $S \in [1,\ 100]$ and $v \in [0,\ 3]$ (see Figure 4.1). For the sake of brevity, we will also define $\Phi^{\lambda}$ which we will refer to as the Heston operator as follows:

$$\Phi^{\boldsymbol{\lambda}} V(t,S,v) = \begin{cases} -\partial_t V(t,S,v) + \mathcal{L}^{\boldsymbol{\lambda}(t)} V(t,S,v) & \text{if } (t,S,v) \in [0,T) \times \Omega, \\ V(t,S,v) - \Lambda(0) & \text{if } (t,S,v) \in [0,T) \times \partial\Omega_D, \\ (0\ 1) \cdot \nabla V(t,S,v) & \text{if } (t,S,v) \in [0,T) \times \partial\Omega_{R_1}, \\ (1\ 0) \cdot \nabla V(t,S,v) - \lim_{S\to\infty} \frac{\partial\Lambda}{\partial S}(S) & \text{if } (t,S,v) \in [0,T) \times \partial\Omega_{R_2}, \\ -\partial_t V(t,S,v) - (rS\ \kappa\gamma) \cdot \nabla V(t,S,v) + rV(t,S,v) & \text{if } (t,S,v) \in [0,T) \times \partial\Omega_{R_t}, \\ V(t,S,v) - \Lambda(S) & \text{if } (t,S,v) \in \{T\} \times \overline{\Omega}. \end{cases} \tag{4.11}$$

**Remark 2.** *Notice how the boundary conditions* (4.9) *and* (4.10) *are independent of the type of option under consideration. Therefore, it is relatively easy to select* (4.11) *describing the payoff of a given option through the appropriate choice of a function* $\Lambda$*. For example, for a call option with the strike price* $K$ *one would need to choose*

$$\Lambda(S) = \max\left(0, S-K\right).$$

*In order to calculate the value of a long butterfly position of width* $2a$ *and the strike price* $K$ *the correct choice is*

$$\Lambda(S) = \max\left(0, S-(K-a)\right) - 2\max\left(0, S-K\right) + \max\left(0, S-(K+a)\right).$$

*Similarly, to consider the value of a long straddle with the strike price* $K$ *one requires*

$$\Lambda(S) = \max\left(0, S-K)\right) - \max\left(0, S-K\right).$$

*Since most widely traded options are combinations of basic puts and calls, they can be expressed in an analogous manner.*

Conceptually, we would now like find at each point $(t,S,v)$ of tempo-spatial domain a map $\hat{\boldsymbol{\lambda}} \in \mathbb{L}$ which minimises the solution of (4.5) at $(t,S,v)$. We will denote such an optimal solution as $\overline{V}$. More precisely, we want to find $\overline{V} \in C(\overline{\Omega})$ which solves the following optimal control problem

$$\overline{V}(t,S,v) := \inf_{\boldsymbol{\lambda}\in\mathbb{L}} \{V^{\boldsymbol{\lambda}}(t,S,v) : \Phi^{\boldsymbol{\lambda}} V^{\boldsymbol{\lambda}} = 0 \text{ on } [0,T] \times \overline{\Omega}\}. \tag{4.12}$$

We would now like to find $\overline{V}$ without having to explicitly search through all the $\boldsymbol{\lambda}$. In order to achieve that we will formulate a Hamilton-Jacobi-Bellman (HJB) equation which is uniquely solved by $\overline{V}$.

We will consider the case $(S,v) \in \Omega$, since the boundary operators are fully linear and hence the argument follows trivially there due to the continuity of $\overline{V}$.

Now fix $(t,S,v) \in [0,T) \times \Omega)$. Then,

$$\begin{aligned}
\overline{V}(t,S,v) &= \overline{V}(T,S,v) - \int_t^T \partial_t \overline{V}(\tau,S,v)d\tau && (4.13a)\\
&= \overline{V}(T,S,v) - \int_t^T \mathcal{L}^{\hat{\boldsymbol{\lambda}}(t)} \overline{V}(\tau,S,v)d\tau && (4.13b)\\
&= \Lambda(S,v) - \int_t^T \mathcal{L}^{\hat{\boldsymbol{\lambda}}(t)} \overline{V}(\tau,S,v)d\tau && (4.13c)\\
&= \Lambda(S,v) - \int_t^T \sup_{\lambda \in L} \mathcal{L}^{\lambda} \overline{V}(\tau,S,v)d\tau, && (4.13d)
\end{aligned}$$

where (4.13d) follows because $\overline{V}$ is defined as infimum over all $\boldsymbol{\lambda} \in \mathbb{L}$. As $\mathbb{L}$ consists of measurable functions, this is obtained by taking the supremum pointwise.

Now subtracting $\overline{V}(t+h,S,v) = \Lambda(S,v) - \int_{t+h}^T \sup_{\lambda \in L} \mathcal{L}^{\lambda} \overline{V}(\tau,S,v)d\tau$ from (4.13) we obtain

$$\int_t^{t+h} \sup_{\lambda \in L} \mathcal{L}^{\lambda} \overline{V}(\tau,S,v)d\tau = \overline{V}(t+h,S,v) - \overline{V}(t,S,v).$$

After multiplication with $1/h$ and taking the limit $h \to 0$ we find that

$$\lim_{h \to 0} \frac{1}{h} \int_t^{t+h} \sup_{\lambda \in L} \mathcal{L}^{\lambda} \overline{V}(\tau,S,v)d\tau = \lim_{h \to 0} \frac{\overline{V}(t+h,S,v) - \overline{V}(t,S,v)}{h} = \partial_t \overline{V}(T,S,v).$$

Assuming that $\overline{V}$ is a classical solution so that

$$\lim_{h \to 0} \frac{1}{h} \int_t^{t+h} \sup_{\lambda \in L} \mathcal{L}^{\lambda} \overline{V}(\tau,S,v)d\tau = \sup_{\lambda \in L} \mathcal{L}^{\lambda} \overline{V}(t,S,v)$$

we arrive at

$$-\partial_t V(t,S,V) + \sup_{\lambda \in L} \mathcal{L}_{\lambda} V(t,S,v) = 0. \tag{4.14}$$

Hence in order to find $\overline{V}$ we are now required to solve the equation (4.14) subject to the boundary conditions (4.6)-(4.10). In the following section we will show that this equation can be in fact transformed into the HJB problem conforming to the setting of Chapter 3 thus allowing us to numerically approximate the option value $V$.

## 4.3 Transformation of the uncertain Heston model

In this section will now perform the transformation of the elliptic operators $\mathcal{L}^{\lambda}$ to their isotropic form in order to be consistent with the framework of the numerical method formulated in Chapter 3. Our first goal is to remove the $S$ dependence of the coefficients. In order to do that we let $S = e^x$. Then

$$\frac{\partial V}{\partial x} = S\frac{\partial V}{\partial S}, \quad \frac{\partial^2 V}{\partial x^2} = S^2 \frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial x}, \quad \frac{\partial^2 V}{\partial S \partial v} = \frac{\partial^2 V}{\partial x \partial v}.$$

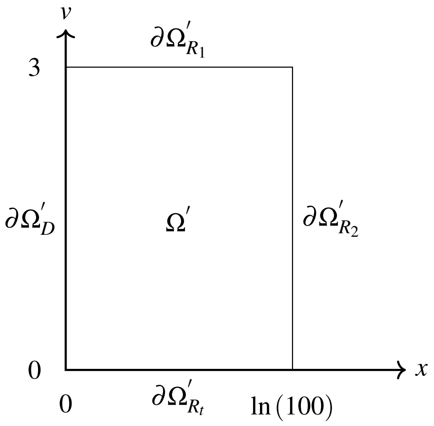


Figure 4.2: Domain $\Omega'$ after the transformation $S = e^x$

Substituting into (4.4) we get

$$\begin{aligned}\mathcal{L}_1^{\lambda} V(x,v,t) :=& -\frac{1}{2}v\left(\frac{\partial^2 V}{\partial x^2} + 2\rho\xi\frac{\partial^2 V}{\partial x \partial v} + \xi^2\frac{\partial^2 V}{\partial v^2}\right) \\ & -(r - \frac{1}{2}v)\frac{\partial V}{\partial x} - [\kappa(\gamma - v) - \xi\lambda\sqrt{v}]\frac{\partial V}{\partial v} + rV. \end{aligned} \tag{4.15}$$

We also need to consider the changes to the boundary conditions. Recall that the boundary condition for large stock price $S$ in untransformed variables is given by $\frac{\partial V(S(x),v,t)}{\partial S}\Big|_{(S,v)\in\partial\Omega_{R_2}} = \lim_{S\to\infty}\frac{\partial\Lambda(S(x))}{\partial S}$. We also have that

$$\frac{\partial V}{\partial S} = \frac{\partial V}{\partial x}\frac{\partial x}{\partial S} = e^{-x}\frac{\partial V}{\partial x}, \quad \frac{\partial \Lambda}{\partial S} = \frac{\partial \Lambda}{\partial x}\frac{\partial x}{\partial S} = e^{-x}\frac{\partial \Lambda}{\partial x}.$$

Hence we conclude that the transformed Neumann boundary condition is

$$\frac{\partial V(x,v,t)}{\partial x}\Big|_{(x,v)\in\partial\Omega'_{R_2}} = \lim_{x\to\infty}\frac{\partial\Lambda(S(x))}{\partial x}. \tag{4.16}$$

The boundary conditions on $\partial\Omega_D$ and $\partial\Omega_{R_1}$ remain unchanged, in the final time condition one substitutes $e^x$ for $S$ and the Robin condition on $\partial\Omega_{R_t}$ is obtained by substituting $v = 0$ into (4.15). The domain $\Omega$ is transformed into $\Omega'$ depicted in Figure 4.2. The Heston operator $\Phi_1$ in the new coordinates is defined by

$$\Phi_1^{\lambda} V(t,x,v) = \begin{cases} -\partial_t V(t,x,v) + \mathcal{L}_1^{\lambda(t)} V(t,x,v) & \text{if } (t,x,v) \in [0,T)\times\Omega', \\ V(t,x,v) - \Lambda(0) & \text{if } (t,x,v) \in [0,T)\times\partial\Omega'_D, \\ (0\ 1)\cdot\nabla V(t,x,v) & \text{if } (t,x,v) \in [0,T)\times\partial\Omega'_{R_1}, \\ (1\ 0)\cdot\nabla V(t,x,v) - \lim_{x\to\infty}\frac{\partial\Lambda(S(x))}{\partial x} & \text{if } (t,x,v) \in [0,T)\times\partial\Omega'_{R_2}, \\ -\partial_t V(t,x,v) - (r\ \kappa\gamma)\cdot\nabla V(t,x,v) + rV(t,x,v) & \text{if } (t,x,v) \in [0,T)\times\partial\Omega'_{R_t}, \\ V(t,x,v) - \Lambda(S(x)) & \text{if } (t,x,v) \in \{T\}\times\overline{\Omega'}. \end{cases} \tag{4.17}$$

In order to remove the second order mixed derivative from $\mathcal{L}_1^{\lambda}$ we consider the following change of variables:

$$y = x - \frac{\rho}{\xi}v, \quad z = \frac{\sqrt{1-\rho^2}}{\xi}v,$$

where $y \in \mathbb{R}$ and $z \geq 0$. The domain $\Omega^{'}$ is transformed into $\Omega^{''}$ whose shape in general depends on the parameters of the numerical experiment. It is depicted in Figure 4.2 with the numerical values of the below case study.

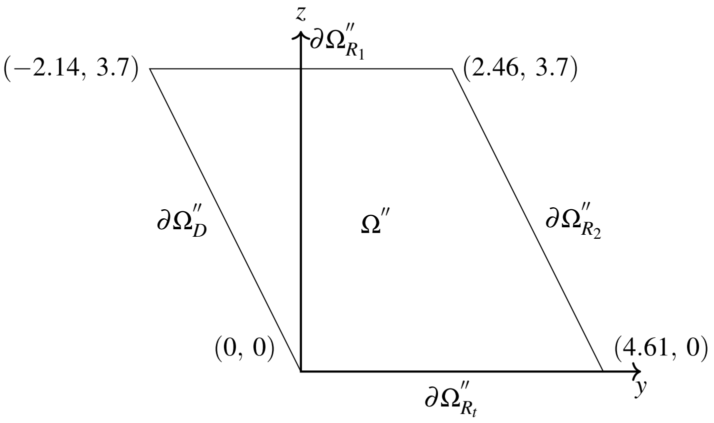

$z$
$\partial\Omega^{''}_{R_1}$
$(-2.14,\ 3.7)$
$(2.46,\ 3.7)$
$\partial\Omega^{''}_{D}$
$\Omega^{''}$
$\partial\Omega^{''}_{R_2}$
$(0,\ 0)$
$(4.61,\ 0)$
$\partial\Omega^{''}_{R_t}$
$y$

Figure 4.3: Domain $\Omega^{''}$ resulting from change of variables with parameter values $\rho = 0.5$, $\xi = 0.7$

We would now like to see how the change of variables affects the Heston operator $\Phi_1^{\boldsymbol{\lambda}}$. For $w$ defined as

$$w(y,z,t) := V\left(x(y,z), v(y,z), t\right)$$

we have that:

$$\begin{aligned}
\frac{\partial V}{\partial x} &= \frac{\partial w}{\partial y}, & \frac{\partial^2 V}{\partial v^2} &= \frac{\rho^2}{\xi^2}\frac{\partial^2 w}{\partial y^2} - \frac{2\rho\sqrt{1-\rho^2}}{\xi^2}\frac{\partial^2 w}{\partial y \partial z} + \frac{1-\rho^2}{\xi^2}\frac{\partial^2 w}{\partial z^2},\\
\frac{\partial V}{\partial v} &= -\frac{\rho}{\xi}\frac{\partial w}{\partial y} + \frac{\sqrt{1-\rho^2}}{\xi}\frac{\partial w}{\partial z}, & \frac{\partial^2 V}{\partial x \partial v} &= -\frac{\rho}{\xi}\frac{\partial^2 w}{\partial y^2} + \frac{\sqrt{1-\rho^2}}{\xi}\frac{\partial^2 w}{\partial y \partial z}.\\
\frac{\partial^2 V}{\partial x^2} &= \frac{\partial^2 w}{\partial y^2}, & &
\end{aligned}$$

Combining those results with (4.15) we obtain the canonical formulation of every interior $\mathcal{L}^{\boldsymbol{\lambda}(\cdot)}$ from (4.5) as required:

$$\begin{aligned}
\mathcal{L}_2^{\boldsymbol{\lambda}(t)} w :=& \frac{-\xi\sqrt{1-\rho^2}}{2} z\Delta w + \left(-r + \frac{\kappa\gamma\rho}{\xi} + \frac{\frac{1}{2}\xi - \kappa\rho}{\sqrt{1-\rho^2}} z - \boldsymbol{\lambda}(t)\rho\sqrt{\frac{\xi z}{\sqrt{1-\rho^2}}}\right)\frac{\partial w}{\partial y}\\
&+ \left(\frac{-\kappa\gamma\sqrt{1-\rho^2}}{\xi} + \kappa z + \boldsymbol{\lambda}(t)\sqrt{\xi z\sqrt{1-\rho^2}}\right)\frac{\partial w}{\partial z} + rw. \qquad (4.18)
\end{aligned}$$

In order to complete the transformation of the initial boundary value problem, we need to apply the same change of variables to the boundary and initial conditions. In other words, our aim is now to reformulate (4.6)-(4.10) accordingly. Since $\frac{\partial V}{\partial x} = \frac{\partial w}{\partial y}$, the Neumann boundary condition (4.8) for large stock prices is obtained by simply substituting $y$ and $z$ into (4.17) on $[0,T) \times \partial\Omega'_{R_2}$ which results in

$$\frac{\partial w}{\partial y}\bigg|_{(y,z)\in\partial\Omega''_{R_2}} = \lim_{y\to\infty} \frac{\partial\Lambda(S(x(y,z)))}{\partial y} \tag{4.19}$$

Under the aforementioned change of variables the Dirichlet boundary condition (4.7) noticing that $\lim_{S\to 0} y = -\infty$ converts straightforwardly to

$$w(y,\, z,\, t)\bigg|_{(y,z)\in\partial\Omega''_D} = \Lambda(0),$$

while for the Neumann condition (4.10) we use the fact that $\frac{\partial V}{\partial v} = -\frac{\rho}{\xi}\frac{\partial w}{\partial y} + \frac{\sqrt{1-\rho^2}}{\xi}\frac{\partial w}{\partial z}$ and $\lim_{v\to\infty} z = \infty$ to obtain

$$(\frac{-\rho}{\xi}\ \frac{\sqrt{1-\rho^2}}{\xi}) \cdot \nabla w = 0.$$

Analogously to the result in [58], the Robin boundary condition for $v \to 0$ is obtained simply by substituting $z = 0$ into (4.18) which gives

$$-\partial_t w + \left(-r + \frac{\kappa\gamma\rho}{\xi}\right)\frac{\partial w}{\partial y} + \left(\frac{-\kappa\gamma\sqrt{1-\rho^2}}{\xi}\right)\frac{\partial w}{\partial z} + rw = 0. \tag{4.20}$$

We summarise the above result by introducing the transformed Heston operator $\Phi_2$ defined as follows

$$\Phi_2^{\boldsymbol{\lambda}} w(t,y,z) = \begin{cases} -\partial_t w(t,y,z) + \mathcal{L}_2^{\boldsymbol{\lambda}(t)} w(t,y,z) & \text{if } (t,y,z) \in [0,T) \times \Omega'', \\ w(t,y,z) - \Lambda(0) & \text{if } (t,y,z) \in [0,T) \times \partial\Omega''_D, \\ (\frac{-\rho}{\xi}\ \frac{\sqrt{1-\rho^2}}{\xi}) \cdot \nabla w(t,y,z) & \text{if } (t,y,z) \in [0,T) \times \partial\Omega''_{R_1}, \\ (1\ 0) \cdot \nabla w(t,y,z) - \lim_{y\to\infty} \frac{\partial\Lambda(S(x(y,z)))}{\partial y} & \text{if } (t,y,z) \in [0,T) \times \partial\Omega''_{R_2}, \\ -\partial_t w(t,y,z) - (r + \frac{\kappa\gamma\rho}{\xi}\ \frac{-\kappa\gamma\sqrt{1-\rho^2}}{\xi}) \cdot \nabla V + rw(t,y,z) & \text{if } (t,y,z) \in [0,T) \times \partial\Omega''_{R_t}, \\ w(t,y,z) - \Lambda(S(x(y,z))) & \text{if } (t,y,z) \in \{T\} \times \overline{\Omega''}. \end{cases} \tag{4.21}$$

We now notice that by replacing the Heston operator $\Phi^{\boldsymbol{\lambda}}$ from (4.4) with its transformed version from (4.21) and following the same argument as in the previous section we obtain an optimal control problem analogous to (4.14) with the structure conforming to the setting of Chapter 3. Note that it resembles the "worst-case scenario" described in [2] but with $\lambda$ instead of $\sigma$ taking the role of the uncertain parameter.

## 4.4 Case Study

Having completed the transformation that allows us to treat the market price of volatility risk as a control in a HJB problem, we now investigate the effects of this parameter on the price of an option. As a starting

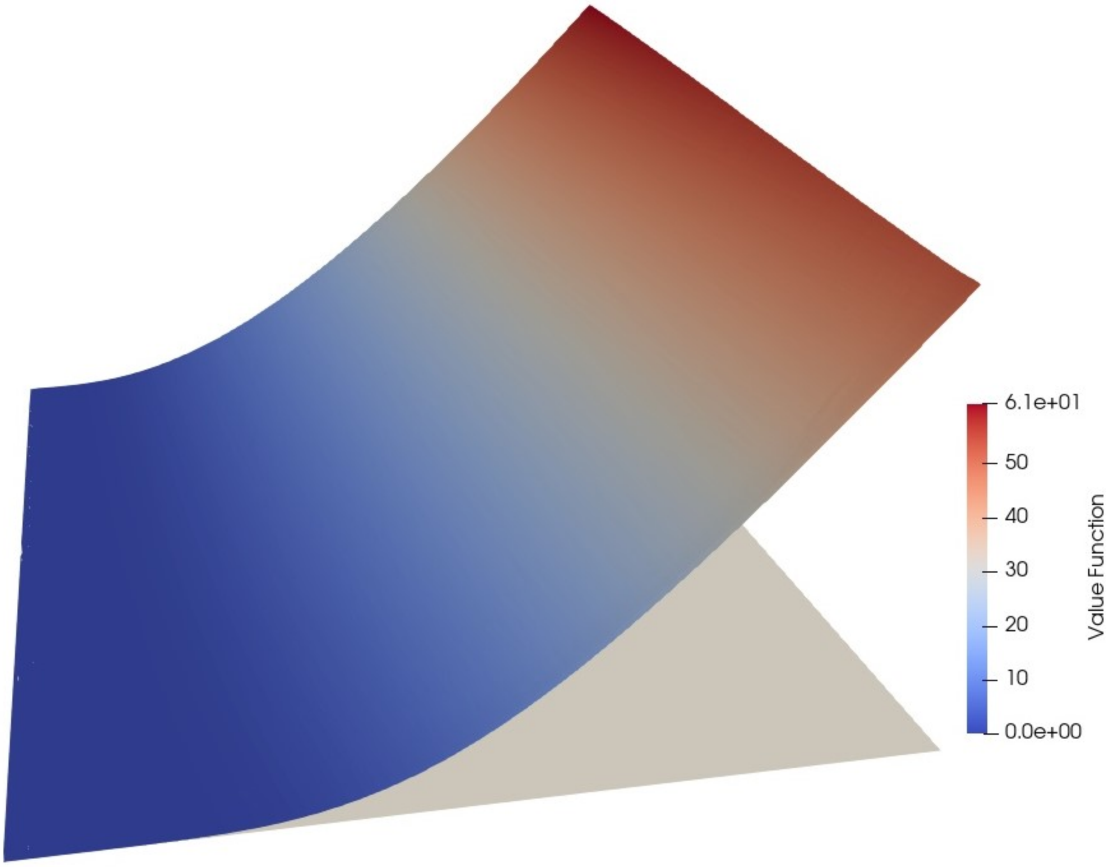


Figure 4.4: Value of a simple call option at $t = 0$ with $T = 0.5$, $K = 50$ and control set $L = [-2.4, -1.6]$.

point of the discussion, we consider the parameters of the experiment described in [34, Table 11]. Following this example we choose the final time $T = 0.5$, the strike price $K = 50$, the volatility of volatility $\xi = 0.7$, the long term volatility mean $\gamma = 0.3$, the mean reversion rate $\kappa = 7$ and the correlation parameter $\rho = 0.5$ (rounded from 0.53). Since the risk free rate $r$ is not stated explicitly, we made a choice of $r = 0.03$. Recalling that the domain was truncated such that $v \in [0, 3]$ and $S \in [1, 100]$, this choice of parameters results in the transformed domain seen in Figure 4.3. A value function of a call option at time $t = 0$ under such choice of parameters and $\lambda = -2.4$ can be seen in Figure 4.4. We now turn our attention to a long butterfly position of width 40 which, as mentioned in Remark 2, is equivalent to choosing

$$\Lambda(S) = \max(0, S - 30) - 2\max(0, S - 50) + \max(0, S - 70).$$

The resulting Bellman final value problem is

$$-\partial_t w + \sup_{\lambda \in L} \mathcal{L}_2^{\lambda} w = 0 \qquad \text{in } (0,T) \times \Omega'', \tag{4.22a}$$

$$w = 0 \qquad \text{on } (0,T) \times \partial\Omega''_D, \tag{4.22b}$$

$$\left(\frac{-\rho}{\xi} \;\; \frac{\sqrt{1-\rho^2}}{\xi}\right) \cdot \nabla w = 0 \qquad \text{on } (0,T) \times \partial\Omega''_{R_1}, \tag{4.22c}$$

$$\frac{\partial w}{\partial y} = 0 \qquad \text{on } (0,T) \times \partial\Omega_{R_2}, \tag{4.22d}$$

$$\begin{aligned} -\partial_t w + \left(-r + \frac{\kappa\gamma\rho}{\xi}\right) \frac{\partial w}{\partial y} + \\ \left(\frac{-\kappa\gamma\sqrt{1-\rho^2}}{\xi}\right) \frac{\partial w}{\partial z} + rw = 0 \end{aligned} \qquad \text{on } (0,T) \times \partial\Omega''_{R_t}, \tag{4.22e}$$

$$\begin{aligned} w = \max\left(0, e^{y - \frac{\rho z}{\sqrt{1-\rho^2}}} - 30\right) - \\ 2\max\left(0, e^{y - \frac{\rho z}{\sqrt{1-\rho^2}}} - 50\right) + \\ \max\left(0, e^{y - \frac{\rho z}{\sqrt{1-\rho^2}}} - 70\right) \end{aligned} \qquad \text{on } \{T\} \times \Omega''. \tag{4.22f}$$

**Result 1: Value Function** Given (4.22) we let $L = [-2.4, -1.6]$. Note how interval $L$ is centred around the market price of volatility risk equal to $-2$ used in [34]. The numerical approximation of the solution to the HJB problem is performed on the transformed domain $\Omega''$ and then the resulting function is cast back to original domain $\Omega$. The outcome is depicted in Figure 4.5. Moreover, one can see in Figure 4.6(a) that the numerical method in fact selects different controls as optimal in different areas of the domain. The difference between the solution of the nonlinear problem compared to the solution of the linear evolution problem associated to one of the controls can be seen in Figure 4.6(b). This indicates the importance of using a nonlinear model.

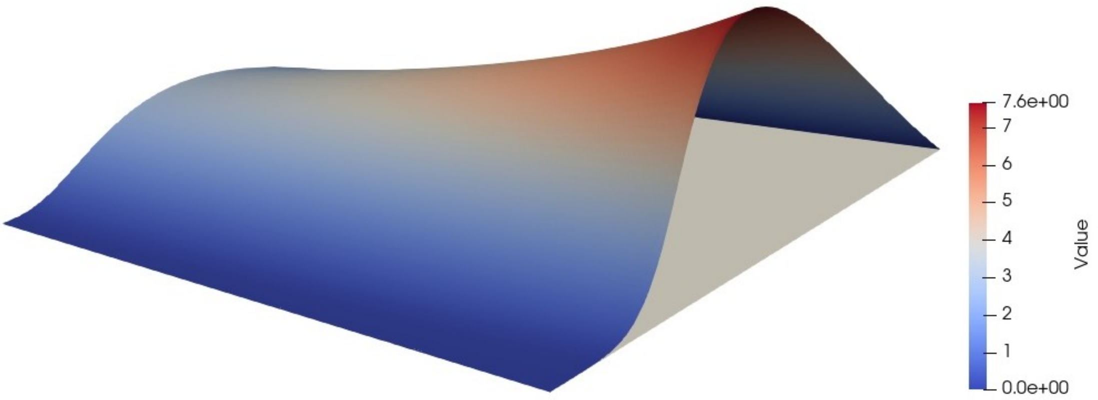
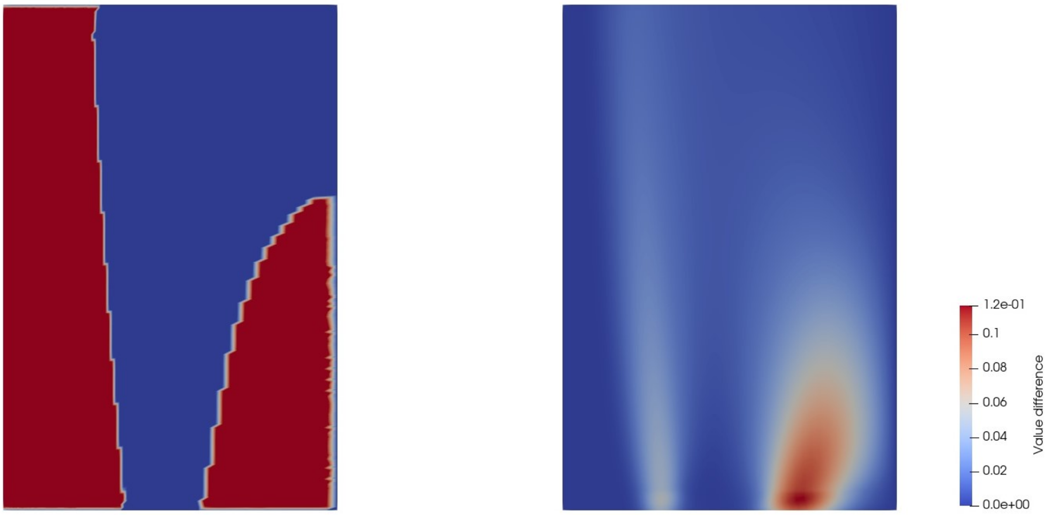


Figure 4.5: Value of a long butterfly position at $t = 0$ with $T = 0.5$, $K = 50$ and control set $L = [-2.4, -1.6]$.



(a) Selected optimal control, Colors represent controls at the extremes of the control set.

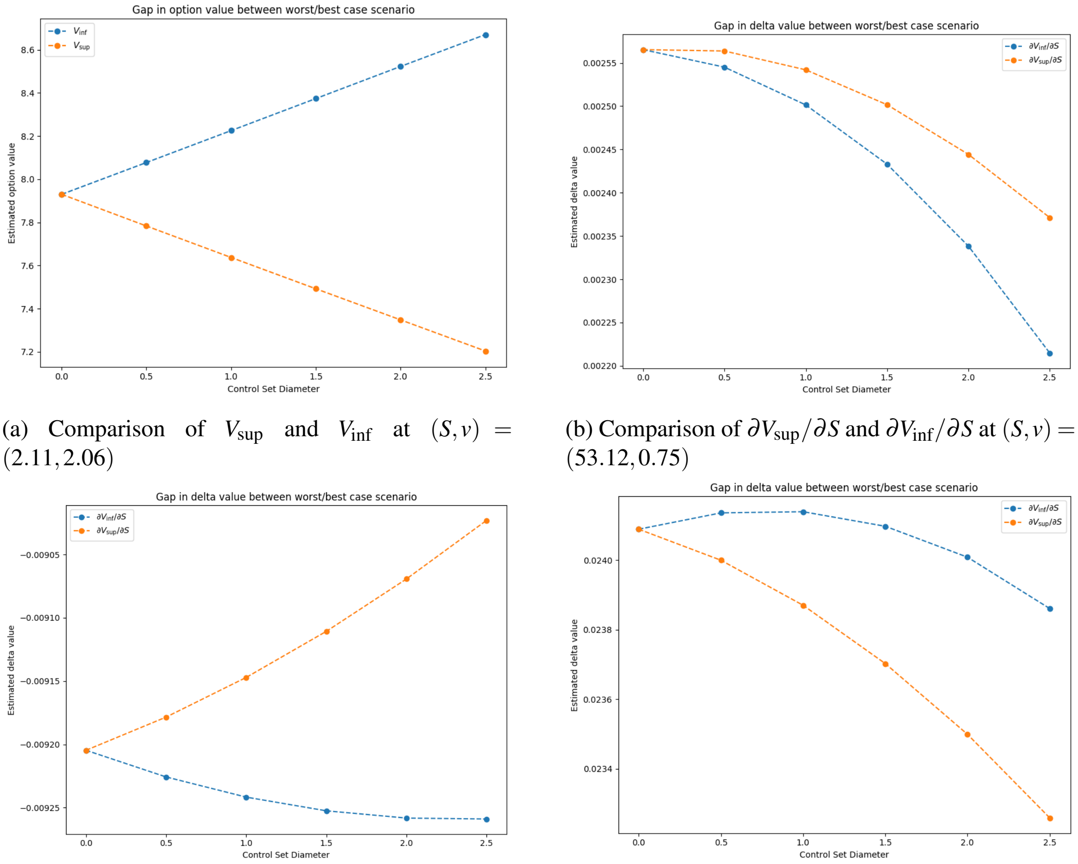
(b) Difference between solutions of a nonlinear problem and linear evolution problem with a fixed control

Figure 4.6: Measurement of the effect of non-linearity for a long butterfly position at $t \approx 0.39$ with $T = 0.5$, $K = 50$ and control set $L = [-2.4, -1.6]$.

**Result 2: $\lambda$ interval testing** We now assess the impact of different choices of control sets $L$ on the option value estimate. In order to do that we consider control sets of increasing diameter and we measure the difference between the value function $V_{\text{inf}}$ of the worst case scenario and the value function $V_{\text{sup}}$ of the best case scenario. In practice, those two cases differ by replacing sup operator by inf operator in (4.22a). The results are shown in Figure 4.7. Firstly, let us point out that our case study confirms the non-trivial impact of the market price of volatility risk on option price. As indicated by Figure 4.7(a) the option value of worse and best case scenario can differ up to 16%. Note that in this case control set contains values ranging between 0 and $-2.5$, which were found to be used in the literature. Given the evidence (see for example findings in [3]) that market price of volatility takes negative values, the simplification of taking $\lambda = 0$ may lead to erroneous estimates. On the other hand, the experiments indicate linear correlation, meaning that in general more negative market prices of volatility risk lead to higher option values. Hence the worst or the best case scenario is achieved by simply choosing $\lambda$ on the verge of a control set.

We instead turn our attention to the partial derivatives of option value $V$, which also known as the deltas. Since they are used to hedge portfolios, they are of special interest to the financial community and a precise estimation of their value is especially important. We now investigate the effect of $\lambda$ on the partial derivative of the option value with respect to $S$. As seen in Figures 4.7(b)-4.7(d) the impact of the value of $\lambda$ on Delta $\partial V / \partial S$ is nonlinear in the vicinity of the strike price $K$. We remark at this point that models which do not guarantee gradient convergence may in general fail to capture this kind of behaviour.

**Result 3: Delta plots** In line with the results of the previous experiment, we continue the investigation of the impact of the market price of volatility risk on the delta values. We again consider the worst and the best case scenarios for control set $L = [-2.5, 0.0]$ at time $t = 0$. Then we plot differences between the



(a) Comparison of $V_{\text{sup}}$ and $V_{\text{inf}}$ at $(S,v) = (2.11, 2.06)$

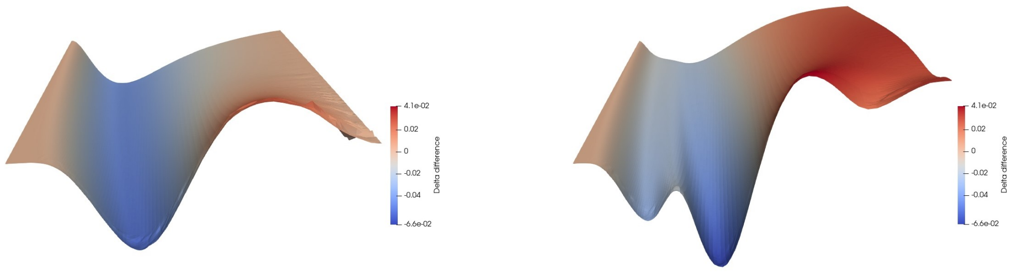
(b) Comparison of $\partial V_{\text{sup}}/\partial S$ and $\partial V_{\text{inf}}/\partial S$ at $(S,v) = (53.12, 0.75)$

(c) Comparison of $\partial V_{\text{sup}}/\partial S$ and $\partial V_{\text{inf}}/\partial S$ at $(S,v) = (51.76, 2.84)$

(d) Comparison of $\partial V_{\text{sup}}/\partial S$ and $\partial V_{\text{inf}}/\partial S$ at $(S,v) = (51.43, 0.23)$

Figure 4.7: Measurement of effect of a diameter of a control set on the value function and its derivative. Control sets are symmetrical and centred at $-1.25$, measurements were made at $t = 0$

Deltas $\partial V_{\text{sup}}/\partial S$ and $\partial V_{\text{inf}}/\partial S$ for all points in $\Omega$ at time $t = 0$. The results for a call option are shown in Figure 4.8(a) and for a long butterfly option in Figure 4.8(b). Note that since $\partial V_{\text{min}}/\partial S$ and $\partial V_{\text{max}}/\partial S$ are both of order 1, the graphs represent a relative as well as an absolute error. We conclude that the impact of the market price of volatility risk on the delta values is significant. In the covered examples, one can expect up to 6% difference between the scenario where $\lambda$ is neglected and the one where a realistic estimate is used.



(a) Call option

(b) Butterfly option

Figure 4.8: Comparison of plots of $\delta(V_{\text{sup}} - V_{\text{inf}})/\delta S$ at time $t = 0$ with control set $[-2.5, 0.0]$

# Chapter 5

# Finite Element Methods for Isaacs problems

## 5.1 Introduction

In this chapter we consider a subclass of fully nonlinear Partial Differential Equations (PDEs) called a second order Hamilton-Jacobi-Isaacs equation (referred to from now on as an Isaacs equation) arising from the optimal control problems. More explicitly, we consider problems of the form

$$-\partial_t v + \inf_{\beta} \sup_{\alpha} (L^{(\alpha,\beta)} v - f^{(\alpha,\beta)}) = 0, \tag{5.1}$$

where $\inf\sup$ is taken over a family of second order linear differential operators $L^{\alpha,\beta}$. A canonical example of such problem is a stochastic two player zero-sum game. Since fully nonlinear PDEs do not generally admit a classical solution, a more relaxed notion of a so-called viscosity solution is required. For a more detailed overview how the differential games, the notion of a viscosity solution and an Isaacs equation are related, we refer reader to Chapter 2 and [107].

The conditions which have to be satisfied by a numerical scheme in order to converge to the unique viscosity solution of an elliptic problem were laid down in [6]. The main difficulty is that the argument is based on existence of the comparison principle. This is especially problematic in the case of an Isaacs equation since the $\inf\sup$ operator is neither concave nor convex. The difficult nature of Isaacs equations is reflected in the challenges apparent in the numerical analysis of these equations. In [101] authors avoid the issues caused by using viscosity solutions and instead employ integro-differential approach. As a result, they are able to formulate a Finite Element Method applicable to the uniformly elliptic problems on convex domains as well as obtain rates of convergence. In [80] author proves an algebraic rate of convergence of Finite Difference schemes approximating a second order Isaacs problem. The result however is limited to highly regular domains. In [31] one can find a formulation of a family of semi-Lagrangian schemes capable of dealing with possibly degenerate second order Isaacs equations. Convergence rates are only obtained for convex case. Also, as is generally the case with semi-Lagrangian schemes, widening of the stencil near the boundary may lead to the loss of accuracy in those areas of the domain.

As far as author of this dissertation is aware this completes the list of currently available methods for numerically approximating second order Isaacs problems. We would also like to point out there exists a rich literature concerning solutions of first order Isaacs equations, often the ones arising in specific applications. Firstly, let us mention [43] where noting the need for approximating high dimensional problems authors present a high-order approximation using a Semi-Lagrangian in time and Finite Volume in space discretisation. One field of the application which requires access to numerical approximations of first order Isaacs equations is the deterministic game theory or more specifically two-player chase problems. We refer reader interested in methods designed specifically for such class of problems to overview in [42]. Another area of application is the so-called $H_\infty$ method coming from control theory. The relation to an Isaacs problem is discussed in [105], while the numerical methods focused on solving high dimensional first orders Isaacs equations arising in $H_\infty$ control design are discussed in [8], [51] and [70].

We point out that solving high-dimensional second order PDEs has been recently area of growing interest. Such problems arise for example in finance in portfolio management where each considered asset increases the dimension of the problem or in applied optimal control problems where dimension of the problem increases with each agent. The main obstacle when solving such problems is so called curse of dimensionality. What is meant by that is that the calculational cost of most of the currently available methods scales exponentially with the dimension of the problem and also the reciprocal of the required accuracy. As a result, most of the methods described so far is unfeasible in case of high dimensional problems, especially so in the case of second order fully nonlinear PDEs. One approach to overcome this issue has been a quickly expanding field of machine learning which uses deep neural networks to obtain approximations of solutions to such problems. In that case nonlinearity is usually treated with the deep splitting approximation method introduced in [9] which approximates solution by solving a series of linear PDEs with separate neural networks. There exist numerous results suggesting that curse of dimensionality can be indeed overcome with neural networks be it for HJB problems (see for example [91], [57], [28], [29]) or Isaacs problems ( [98], [66]). It is however worth noting that even though the numerical results suggest that neural networks possess sufficient expressiveness to overcome curse of dimensionality, the theoretical results are still at best partial. For an overview of contemporary machine learning methods for partial differential equations we refer reader to [10]. We also mention another class of approximation algorithm, so called multilevel Picard approximation methods. Interested reader can find review of those in [39], for a result regarding HJB equations specifically we point to [30].

The main contribution of this chapter is the generalisation of the setting of [68] to Isaacs problems with the non-homogenous Dirichlet boundary conditions. Due to the nonconvexity of the Isaacs operator the solution of a linear evolution problem can no longer be used as a bound of the numerical solution. This poses problem in terms of establishing stability as well as proving point-wise convergence to the boundary conditions. We tackle the first problem by considering combinations of linear problem solutions while in the second case we formulate a framework in which the point-wise convergence to the boundary condition by the envelopes of numerical solutions can be guaranteed, and which hinges on the existence of the barrier functions whose properties are discussed in detail. As a result we formulate a convergent Finite Element Method capable of approximating possibly degenerate second order Isaacs problems with general Dirichlet boundary conditions on bounded, Lipschitz domains.

This chapter is organised as follows. In Section 5.2 we define the class of the problems under consideration and the underlying assumptions. We also define the notion of a viscosity solution used throughout the chapter. In Section 5.3 we introduce the discretization in time and space as well the numerical scheme. In Section 5.4 we prove the monotonicity of the proposed scheme which is a crucial step in proving the convergence to the unique viscosity solution. In Section 5.5 we introduce the algorithm which solves the nonlinear discrete problem at each timestep and which guarantees the existence and uniqueness of the numerical solution. We go on to prove the stability of that solution in Section 5.6. In Section 5.8 we discuss different notions of the boundary condition and introduce the barrier functions which ensure that boundary conditions can be satisfied in the pointwise sense at least on the part of the boundary. We obtain the main result and prove the convergence of the numerical solution to the unique viscosity solution of an Isaacs problem in Section 5.7. Finally, in Section 5.3 we present the numerical experiments verifying the convergence of the scheme and we show application to a two-player stochastic game.

## 5.2 Isotropic Isaacs equations

We consider Isaacs equations on bounded Lipschitz domains $\Omega \in \mathbb{R}^d$ with $d \geq 2$. Let $A$ and $B$ be compact metric spaces and $(\alpha, \beta) \in A \times B$. We introduce the linear operators

$$L^{(\alpha,\beta)} : H^2(\Omega) \to L^2(\Omega),\ w \mapsto -a^{(\alpha,\beta)} \Delta w - b^{(\alpha,\beta)} \cdot \nabla w + c^{(\alpha,\beta)} w \tag{5.2}$$

and data terms $f^{(\alpha,\beta)}, v_T \in C(\overline{\Omega})$, $g \in W^{1,\infty}(\mathbb{R}^d)$. We assume that the mapping

$$A \times B \to C(\overline{\Omega}) \times C(\overline{\Omega}, \mathbb{R}^d) \times C(\overline{\Omega}) \times C(\overline{\Omega}), (\alpha, \beta) \mapsto (a^{(\alpha,\beta)}, b^{(\alpha,\beta)}, c^{(\alpha,\beta)}, f^{(\alpha,\beta)})$$

is assumed to be continuous such that the families of functions $\{a^{(\alpha,\beta)}\}_{(\alpha,\beta)\in A\times B}$, $\{b^{(\alpha,\beta)}\}_{(\alpha,\beta)\in A\times B}$, $\{c^{(\alpha,\beta)}\}_{(\alpha,\beta)\in A\times B}$ and $\{f^{(\alpha,\beta)}\}_{(\alpha,\beta)\in A\times B}$ are equicontinuous. Moreover, $a^{(\alpha,\beta)}(x) \geq 0$ and $f^{(\alpha,\beta)}(x) \geq 0$

for any $(\alpha,\beta) \in A \times B$ and $x \in \Omega$. It follows that all $L^{\alpha,\beta}$ are degenerate elliptic and that

$$\sup_{(\alpha,\beta)\in A\times B} \| (a^{(\alpha,\beta)}, b^{(\alpha,\beta)}, c^{(\alpha,\beta)}, f^{(\alpha,\beta)}) \|_{C(\overline{\Omega})\times C(\overline{\Omega},\mathbb{R}^d)\times C(\overline{\Omega})\times C(\overline{\Omega})} < \infty.$$

For any smooth $w$ we define the Hamiltonian

$$Hw := \inf_{\beta\in B} \sup_{\alpha\in A} (L^{(\alpha,\beta)} w - f^{(\alpha,\beta)}),$$

assuming that the supremum and the infimum are applied pointwise. We wish to study the Isaacs problems of the following form:

$$-\partial_t v + Hv = 0 \qquad \text{in } (0,T)\times\Omega, \tag{5.3a}$$

$$v = g \qquad \text{on } (0,T)\times\partial\Omega, \tag{5.3b}$$

$$v = v_T \qquad \text{on } \{T\}\times\overline{\Omega}. \tag{5.3c}$$

Our aim is to construct a Finite Element Method which approximates the viscosity solution of the Isaacs problem (5.3). We now formalise what is meant by a viscosity solution throughout this chapter. Firstly, let us reformulate (5.3) as $F = 0$ where $F$ is the following differential operator

$$F(x,q,p,r,s) = \begin{cases} -r + \inf_\beta \sup_\alpha \left(-a^{(\alpha,\beta)} q - b^{(\alpha,\beta)} \cdot p + c^{(\alpha,\beta)} s - f^{\alpha}(x)\right) & \text{on } [0,T)\times\overline{\Omega}, \\ s - g(x) & \text{on } [0,T)\times\partial\Omega, \\ s - v_T(x) & \text{on } \{T\}\times\overline{\Omega}. \end{cases}$$

Given a bounded function $v : [0,T]\times\overline{\Omega} \to \mathbb{R}$ we define its upper and lower semi-continuous envelopes, respectively as

$$v^*(t,x) := \limsup_{\substack{(s,y)\to(t,x) \\ (s,y)\in[0,T]\times\overline{\Omega}}} v(s,y)$$

and

$$\underline{v}_*(t,x) := \liminf_{\substack{(s,y)\to(t,x) \\ (s,y)\in[0,T]\times\overline{\Omega}}} v(s,y).$$

We analogously extend the definition of lower- and upper semicontinuous envelopes to $F$.

**Definition 10.** *A bounded function $v$ is a viscosity supersolution (respectively, subsolution) of* (3.3) *if, for any test function $\psi \in C^2(\mathbb{R}\times\mathbb{R}^d)$,*

$$F^*(x, \Delta\psi(t,x), \nabla\psi(t,x), \partial_t\psi(t,x), \underline{v}_*(t,x)) \geq 0,$$

*(respectively,*

$$F_*(x,\Delta\psi(t,x),\nabla\psi(t,x),\partial_t\psi(t,x),v^*(t,x))\leq 0,)$$

*provided that $v^*-\psi$ attains a local minimum (respectively, $v_*-\psi$ attains a local maximum) and additionally $v^*$ (respectively $v_*$) satisfies the initial conditions and the boundary conditions on $\omega\subseteq\partial\Omega$ in the pointwise sense. A function which is simultaneously a viscosity super- and subsolution of* (3.3) *is called a viscosity solution.*

Note how the above definition is in spirit similar to Definition 3 but it additionally requires that the solution satisfies the Dirichlet boundary condition in the pointwise sense on the subset $\omega$ of the boundary.

## 5.3 The numerical method

We consider a sequence $V_i, i\in\mathbb{N}$ of piecewise linear shape-regular finite element spaces. Let us denote the nodes of the finite element mesh by $y_i^\ell$ where $\ell$ corresponds to the position on the mesh. The associated hat functions are denoted $\phi_i^\ell$, i.e. $\phi_i^\ell\in V_i$ such that $\phi_i^s=1$ at the node $y_i^s$, while $\phi_i^\ell=0$ for all the remaining nodes $y_i^\ell$, $\ell\neq s$. Set $\hat{\phi}_i^\ell:=\phi_i^\ell/\|\phi_i^\ell\|_{L^1(\Omega)}$. Therefore, the $\phi_i^\ell$ are normalised in the $L^\infty$ norm whilst the $\hat{\phi}_i^\ell$ are normalised in the $L^1$ norm.

In order to construct a Finite Element Method which is both stable and consistent, care needs to be taken when imposing the boundary conditions. Due to the more delicate stability properties of Isaacs operators compared to for instance Bellman operators, on nonconvex domains a nodal interpolant is not suitable to map the boundary data onto the approximation space. Instead, given $w\in C(H^1(\Omega))$ and letting $V_i^0\subset V_i$ be the subspace of functions which satisfy the homogeneous Dirichlet conditions on $\partial\Omega$, we consider linear mappings $P_i$ which map $w$ into $V_i$ such that for all $\hat{\phi}_i^\ell\in V_i^0$

$$\langle\nabla P_i w,\nabla\hat{\phi}_i^\ell\rangle=\langle\nabla w,\nabla\hat{\phi}_i^\ell\rangle. \tag{5.4}$$

If we chose $P_i w$ such that it interpolates $w$ on the boundary, then $P_i$ becomes the classical elliptic projection of the Laplacian, so we will refer to $P_i$ as an elliptic projection from now on, see also [68, Section 4].

**Assumption 9.** *There are linear mappings $P_i$ satisfying* (5.4) *and there is a constant $C\geq 0$ such that for every $w\in C^\infty(\mathbb{R}^d)$ and $i\in\mathbb{N}$,*

$$\|P_i w\|_{W^{1,\infty}(\Omega)}\leq C_w\,\|w\|_{W^{1,\infty}(\Omega)}\quad \textit{and}\quad \lim_{i\to\infty}\|P_i w-w\|_{W^{1,\infty}(\Omega)}=0. \tag{5.5}$$

It is shown in [33] that (9) holds when $\Omega$ is a bounded convex polyhedral domain in $\mathbb{R}^d$, $d\in\{2,3\}$, when the mesh satisfies a local quasi-uniformity condition and when the test functions vanish on the boundary. To apply the result for nonconvex domains $\Omega$ and general $w\in C^\infty(\mathbb{R}\times\mathbb{R}^d)$, consider for example a convex polyhedral domain $B$ containing $\Omega$ and assume there is a locally quasi-uniform mesh on $B$ which coincides with the original mesh on $\Omega$. Let $\eta$ be a smooth cut-off function with compact support in $B$ such that $\eta \equiv 1$ on $\Omega$. Then the classical elliptic projection on $B$, acting on $\eta w : B \to \mathbb{R}$, has the required properties. Given this construction for $P_i$, it is natural to refer to it as an elliptic projection. This construction provides on nonconvex domains an approximation of the boundary data, which ensures in particular the stability of numerical solutions.

Let $V_i^g \subset V_i$ be the the subspace of functions which attain $P_i g$ on the boundary. We let the index $\ell$ range over the boundary nodes first, in other words, $y_i^\ell \in \Omega$ for $\ell \leq N_i := \dim V_i^0$.

The mesh size, i.e. the largest diameter of an element, is denoted $\Delta x_i$. It is assumed that $\Delta x_i \to 0$ as $i \to \infty$. The uniform time step size is denoted $h_i$ with the constraint that $T/h_i \in \mathbb{N}$. It is assumed that $h_i \to 0$ as $i \to \infty$. Let $s_i^k$ be the $k$th time step at the refinement level $i$. Then the set of time steps is $S_i := \{s_i^k : k = T/h_i, \dots, 0\}$.

The time derivative is approximated by $d_i$ for which we let the $\ell$th entry of $d_i w(s_i^k, \cdot)$ be

$$(d_i w(s_i^k, \cdot))_\ell = \frac{w(s_i^{k+1}, y_i^\ell) - w(s_i^k, y_i^\ell)}{h_i}.$$

For the discretisation of $L^{(\alpha,\beta)}$ we allow splitting into an explicit and an implicit part. For each pair $(\alpha, \beta)$ and for each $i$, we introduce the explicit operator $E_i^{(\alpha,\beta)}$ and the implicit operator $I_i^{(\alpha,\beta)}$ such that $L^{(\alpha,\beta)} \approx E_i^{(\alpha,\beta)} + I_i^{(\alpha,\beta)}$ and

$$\begin{aligned} E_i^{(\alpha,\beta)} &: H^2(\Omega) \to L^2(\Omega),\ w \mapsto -\bar{a}_i^{(\alpha,\beta)} \Delta w - \bar{b}_i^{(\alpha,\beta)} \cdot \nabla w + \bar{c}_i^{(\alpha,\beta)} w, \\ I_i^{(\alpha,\beta)} &: H^2(\Omega) \to L^2(\Omega),\ w \mapsto -\bar{\bar{a}}_i^{(\alpha,\beta)} \Delta w - \bar{\bar{b}}_i^{(\alpha,\beta)} \cdot \nabla w + \bar{\bar{c}}_i^{(\alpha,\beta)} w. \end{aligned}$$

For each $i$ we require a discretisation $f_i^{(\alpha,\beta)}$ of $f^{(\alpha,\beta)}$ which is non-negative and approximates $f^{(\alpha,\beta)}$. We now want to make the conceptual statements $L^{(\alpha,\beta)} \approx E_i^{(\alpha,\beta)} + I_i^{(\alpha,\beta)}$ and $f^{(a,b)} \approx f_i^{(\alpha,\beta)}$ more precise.

**Assumption 10.** *For all sequences of nodes $(y_i^\ell)_{i \in \mathbb{N}}$, where in general $\ell = \ell(i)$ depends on i:*

$$\begin{aligned} \lim_{i \to \infty} \sup_{(\alpha,\beta) \in A \times B} \Big( &\big\| a^{(\alpha,\beta)} - \big(\bar{a}_i^{(\alpha,\beta)}(y_i^\ell) + \bar{\bar{a}}_i^{(\alpha,\beta)}(y_i^\ell)\big) \big\|_{L^\infty(\operatorname{supp} \hat{\phi}_i^\ell)} \\ &+ \big\| b^{(\alpha,\beta)} - \big(\bar{b}_i^{(\alpha,\beta)} + \bar{\bar{b}}_i^{(\alpha,\beta)}\big) \big\|_{L^\infty(\Omega, \mathbb{R}^d)} \\ &+ \big\| c^{(\alpha,\beta)} - \big(\bar{c}_i^{(\alpha,\beta)} + \bar{\bar{c}}_i^{(\alpha,\beta)}\big) \big\|_{L^\infty(\Omega)} + \big\| f^{(\alpha,\beta)} - f_i^{(\alpha,\beta)} \big\|_{L^\infty(\Omega)} \Big) = 0. \end{aligned}$$

*The coefficients $\bar{c}_i^{(\alpha,\beta)}$ and $\bar{\bar{c}}_i^{(\alpha,\beta)}$ are non-negative and that there exists $\gamma \in \mathbb{R}$ such that*

$$\|\bar{c}_i^{(\alpha,\beta)}\|_{L^\infty} + \|\bar{\bar{c}}_i^{(\alpha,\beta)}\|_{L^\infty} \leq \gamma, \qquad \forall i \in \mathbb{N},\ \forall (\alpha,\beta) \in A \times B. \tag{5.6}$$

*The family of mappings*

$$\{(\bar{a}_i^{(\alpha,\beta)},\bar{b}_i^{(\alpha,\beta)},\bar{c}_i^{(\alpha,\beta)},\bar{\bar{a}}_i^{(\alpha,\beta)},\bar{\bar{b}}_i^{(\alpha,\beta)},\bar{\bar{c}}_i^{(\alpha,\beta)},f_i^{\alpha},g_i^{\alpha})\}_{\alpha\in A}$$

*is equicontinuous.*

The splitting into explicit and implicit part is used to define the numerical operators $\mathsf{E}_i^{(\alpha,\beta)}$ and $\mathsf{I}_i^{(\alpha,\beta)}$ as mappings from $H^1(\Omega)$ to $\mathbb{R}^{N_i}$:

$$(\mathsf{E}_i^{(\alpha,\beta)}w)_\ell := \bar{a}_i^{(\alpha,\beta)}(y_i^\ell)\langle\nabla w,\nabla\hat{\phi}_i^\ell\rangle + \langle -\bar{b}_i^{(\alpha,\beta)}\cdot\nabla w + \bar{c}_i^{(\alpha,\beta)}w,\hat{\phi}_i^\ell\rangle, \tag{5.7a}$$

$$(\mathsf{I}_i^{(\alpha,\beta)}w)_\ell := \bar{\bar{a}}_i^{(\alpha,\beta)}(y_i^\ell)\langle\nabla w,\nabla\hat{\phi}_i^\ell\rangle + \langle -\bar{\bar{b}}_i^{(\alpha,\beta)}\cdot\nabla w + \bar{\bar{c}}_i^{(\alpha,\beta)}w,\hat{\phi}_i^\ell\rangle, \tag{5.7b}$$

$$(\mathsf{F}_i^{(\alpha,\beta)})_\ell := \langle f_i^{(\alpha,\beta)},\hat{\phi}_i^\ell\rangle, \tag{5.7c}$$

where $\ell$ ranges over all internal nodes, i.e. $\ell \le N_i$. When restricted to the domain $V_i$, the numerical operators have matrix representations with respect to the nodal bases $\{\phi_i^\ell\}_\ell$, which we also denote by $\mathsf{E}_i^{(\alpha,\beta)}$ and $\mathsf{I}_i^{(\alpha,\beta)}$.

Throughout the chapter, we will make use of the partial ordering of $\mathbb{R}^n$: for $x,y\in\mathbb{R}^n$, we write $x\ge y$ if and only if $x_\ell \ge y_\ell$ for all $\ell\in\{1,\dots,n\}$. For collections $\{x^\alpha\}_\alpha\subset\mathbb{R}^n$ and $\{y^\beta\}_\beta\subset\mathbb{R}^n$, we define the operators $\sup_\alpha$ and $\inf_\beta$ componentwise:

$$\big(\sup_\alpha x^\alpha\big)_\ell = \sup_\alpha x_\ell^\alpha \quad\text{and}\quad \big(\inf_\beta y^\beta\big)_\ell = \inf_\beta y_\ell^\beta.$$

We can now state the numerical scheme for finding an approximate solution to (5.3). We initialise the scheme with the elliptic projection $v_i(T,\cdot) = P_i v_T$. Then, in order to find the numerical solution $v_i(s_i^k,\cdot)\in V_i^g$, we proceed inductively over the remaining timesteps $k\in\{T/h_i-1,\dots,0\}$:

$$-d_i v_i(s_i^k,\cdot) + \inf_{\beta\in B}\sup_{\alpha\in A}\big(\mathsf{E}_i^{(\alpha,\beta)}v_i(s_i^{k+1},\cdot) + \mathsf{I}_i^{(\alpha,\beta)}v_i(s_i^k,\cdot) - \mathsf{F}_i^{(\alpha,\beta)}\big) = 0. \tag{5.8}$$

If all $\mathsf{I}_i^{(\alpha,\beta)}$ vanish then (5.8) is an explicit scheme, otherwise it is implicit.

To show the existence and uniqueness of numerical solutions, it is useful to provide an equivalent reformulation of the scheme. For a function $w: S_i\times\Omega\to\mathbb{R}$ that satisfies $w(s_i^k,\cdot)\in H^1(\Omega)$ for all $s_i^k\in S_i$, let $(\alpha_i^{\ell,k}(w),\beta_i^{\ell,k}(w))\in A\times B$ so that $\alpha_i^{\ell,k}(w)$ is a maximiser of (5.9a) and $\beta_i^{\ell,k}(w)$ is a minimiser of (5.9b):

$$\sup_{\alpha\in A}\Big(\mathsf{E}_i^{(\alpha,\beta)}w(s_i^{k+1},\cdot) + \mathsf{I}_i^{(\alpha,\beta)}w(s_i^k,\cdot) - \mathsf{F}_i^{(\alpha,\beta)}\Big)_\ell, \tag{5.9a}$$

$$\inf_{\beta\in B}\left(\mathsf{E}_i^{(\alpha_i^{\ell,k}(w),\beta)}w(s_i^{k+1},\cdot) + \mathsf{I}_i^{(\alpha_i^{\ell,k}(w),\beta)}w(s_i^k,\cdot) - \mathsf{F}_i^{(\alpha_i^{\ell,k}(w),\beta)}\right)_\ell. \tag{5.9b}$$

Let $\mathsf{I}_i^{k,w}$ and $\mathsf{E}_i^{k,w}$ be the matrices whose $\ell$th row at $k$th time step is equal to that of

$$\mathsf{I}_i^{(\alpha_i^{\ell,k}(w),\beta_i^{\ell,k}(w))} \quad \text{and} \quad \mathsf{E}_i^{(\alpha_i^{\ell,k}(w),\beta_i^{\ell,k}(w))},$$

respectively. Also let the $\ell$th entry of $\mathsf{F}_i^{k,w}$ be

$$\left(\mathsf{F}_i^{(\alpha_i^{\ell,k}(w),\beta_i^{\ell,k}(w))}\right)_\ell.$$

Notice that control $(\alpha_i^{\ell,k}(w),\beta_i^{\ell,k}(w))$ is not necessarily unique. The subsequent analysis is valid regardless of the choice of the control.

We may reformulate the numerical scheme (5.8) with $\mathsf{E}_i^{k,w}$, $\mathsf{I}_i^{k,w}$ and $\mathsf{F}_i^{k,w}$ as follows. Initialise the scheme with the elliptic projection of $v_i(T,\cdot)$, then for each $k \in \{T/h_i - 1,\dots,1,0\}$, $v_i$ solve

$$(h_i\mathsf{I}_i^{k,v_i} + \mathsf{Id})\,v_i(s_i^k,\cdot) + (h_i\mathsf{E}_i^{k,v_i} - \mathsf{Id})\,v_i(s_i^{k+1},\cdot) - h_i\mathsf{F}_i^{k,v_i} = 0. \tag{5.10}$$

### 5.3.1 Consistency properties of elliptic projections

We conclude the section with the consistency properties of linear operators for fixed $(\alpha,\beta) \in A \times B$. The result is unchanged from the Bellman setting in [68].

**Lemma 6.** *Let $w \in C^\infty(\mathbb{R}\times\mathbb{R}^d)$ and let $s_i^{k(i)} \to t \in [0,T]$, $y_i^{\ell(i)} \to x \in \Omega$ as $i \to \infty$. Then*

$$\lim_{i\to\infty} d_i P_i w(s_i^{k(i)},\cdot) = \partial_t w(t,\cdot) \text{ in } W^{1,\infty}(\Omega). \tag{5.11}$$

*Also we have that*

$$\lim_{i\to\infty}\left(\mathsf{E}_i^{(\alpha,\beta)}P_i w(s_i^{k(i)+1},\cdot) + \mathsf{I}_i^{(\alpha,\beta)}P_i w(s_i^{k(i)},\cdot) - \mathsf{F}_i^{(\alpha,\beta)}\right)_{\ell(i)} = L^{(\alpha,\beta)}w(t,x) - f^{(\alpha,\beta)}(x), \tag{5.12}$$

*where convergence to the limit is uniform over all $(\alpha,\beta) \in A \times B$.*

*Proof.* The proof is analogous to the Hamilton-Jacobi-Bellman case described in [68], once control $\alpha \in A$ is replaced by the pair of controls $(\alpha,\beta) \in A \times B$. □

## 5.4 Monotonicity

Monotonicity properties of the numerical scheme play a crucial role in ensuring convergence to the viscosity solution.

**Definition 11.** *An operator $F : V_i \to \mathbb{R}^{N_i}$ is said to satisfy the Local Monotonicity Property (LMP) if for all $v \in V_i$ such that $v$ has a non-positive local minimum at the internal node $y_i^\ell$, we have $(Fv)_\ell \le 0$. The* *operator $F$ satisfies the weak Discrete Maximum Principle (wDMP) provided that for any $v \in V_i$,*

$$\text{if } \big(Fv\big)_\ell \geq 0 \text{ for all } \ell \in \{1,\dots,N_i\}, \qquad \text{then } \min_{\Omega} v \geq \min\{\min_{\partial\Omega} v, 0\}. \tag{5.13}$$

Suppose an operator $F$ satisfies the LMP and $v \in V_i$ has a negative local minimum at an internal node $y_i^\ell$, then $((F+\varepsilon \mathsf{Id})v)_\ell < 0$ for any positive $\varepsilon$. Therefore $F+\varepsilon\mathsf{Id}$ satisfies (wDMP) for any positive $\varepsilon$.

**Assumption 11.** *For each $(\alpha,\beta) \in A\times B$, we assume that $\mathsf{E}_i^{(\alpha,\beta)}$, restricted to $V_i^0$, has non-positive off-diagonal entries. We assume that $h_i$ is small enough so that $h_i\mathsf{E}_i^{(\alpha,\beta)} - \mathsf{Id}$ is monotone for every $(\alpha,\beta)$, i.e. so that all entries of all $h_i\mathsf{E}_i^{(\alpha,\beta)} - \mathsf{Id}$ are non-positive. For each $(\alpha,\beta)\in A\times B$, we suppose that $\mathsf{I}_i^{(\alpha,\beta)}$ satisfies the LMP.*

The above assumption puts a restriction on the size of a time step since we require $h_i\mathsf{E}_i^{(\alpha,\beta)} - \mathsf{Id}$ to be monotone.

We now turn to the matrices $h_i\mathsf{I}_i^{k,w} + \mathsf{Id}$ and $h_i\mathsf{E}_i^{k,w} - \mathsf{Id}$ which will later be used in the proof of the well-posedness of the scheme (5.10).

**Lemma 7.** *Consider a $w : S_i \times \Omega \to \mathbb{R}$ so that $w(s_i^k,\cdot) \in H^1(\Omega)$ for all $s_i^k \in S_i$. Then, the matrices $h_i\mathsf{E}_i^{k,w} - \mathsf{Id}$ are monotone and the matrices of $h_i\mathsf{I}_i^{k,w} + \mathsf{Id}$ restricted to $V_i^0$ are diagonally dominant M-matrices. For fixed $w$, the operators $v \mapsto \mathsf{I}_i^{k,w}v$ and $v \mapsto (h_i\mathsf{I}_i^{k,w} + \mathsf{Id})\,v$ satisfy, respectively, the LMP and wDMP.*

*Proof.* The proof is analogous to that of [68, Lemma 2.3], once control $\alpha \in A$ is replaced by the pair of controls $(\alpha,\beta)\in A\times B$. □

**Corollary 1.** *The non-linear operators $w \mapsto \mathsf{I}_i^{k,w}w$ and $w \mapsto (h_i\mathsf{I}_i^{k,w} + \mathsf{Id})\,w$ satisfy the LMP and wDMP, respectively. Moreover, $w \mapsto -(h_i\mathsf{E}_i^{k,w} - \mathsf{Id})\,w$ is positive: if $w \geq 0$ then $-(h_i\mathsf{E}_i^{k,w} - \mathsf{Id})\,w \geq 0$.*

### 5.4.1 Monotonicity through artificial diffusion

Using the method of artificial diffusion as described in [68] one can ensure that $\mathsf{E}_i^{(\alpha,\beta)}$ and $\mathsf{I}_i^{(\alpha,\beta)}$ satisfy Assumption 11, provided the meshes are strictly acute.

Let $\mathcal{T}_i$ be the mesh corresponding to the finite element space $V_i$. Given a function $g : \Omega \to \mathbb{R}^d$ and an element $K$ of $\mathcal{T}_i$, we denote

$$|g|_K := \Big(\sum_{j=1}^{d} \|g_j\|^2_{L^\infty(K)}\Big)^{\frac{1}{2}}.$$

We say that the meshes $\mathcal{T}_i$ are strictly acute if there exists $\vartheta \in (0,\pi/2)$ such that for all $i \in \mathbb{N}$:

$$\nabla\phi_i^\ell \cdot \nabla\phi_i^l\big|_K \leq -\sin(\vartheta)\; |\nabla\phi_i^\ell|_K\, |\nabla\phi_i^l|_K \qquad \forall \ell, l \leq \dim V_i,\ \ell \neq l,\ \forall K \in \mathcal{T}_i. \tag{5.14}$$

Consider a splitting of the form $a^{(\alpha,\beta)} = \tilde{a}_i^{(\alpha,\beta)} + \tilde{\tilde{a}}_i^{(\alpha,\beta)}$, $b^\alpha = \bar{b}_i^\alpha + \bar{\bar{b}}_i^\alpha$, $c^\alpha = \bar{c}_i^\alpha + \bar{\bar{c}}_i^\alpha$ and $f^{(\alpha,\beta)} = f_i^{(\alpha,\beta)}$, where all terms are in $C(\overline{\Omega})$. Choose non-negative artificial diffusion coefficients $\bar{\nu}_i^{(\alpha,\beta),\ell}$ and $\bar{\bar{\nu}}_i^{(\alpha,\beta),\ell}$ such

**Algorithm 1** Howard's method

```
Require: $\eta > 0$,  $k \in \{0,\dots,T/h_i - 1\}$,  $w \in V_i^g$,  $(\alpha_0,\beta_0) \in A \times B$
$u \leftarrow w$
$\beta \leftarrow \beta_0$
while $\|\Psi^{(\alpha,\beta)}(u,w)\| \geq \eta$ do
    $\alpha \leftarrow \alpha_0$
    while $\|\Psi^{(\alpha,\beta)}(u,w)\| \geq \eta$ do
        $u \leftarrow \Phi^{(\alpha,\beta)}(w)$
        $\alpha \leftarrow \operatorname{argmax}_{\alpha' \in A} \Psi^{(\alpha',\beta)}(u,w)$
    end while
    $u \leftarrow \Phi^{(\alpha,\beta)}(w)$
    $\beta \leftarrow \operatorname{argmin}_{\beta' \in B} \Psi^{(\alpha,\beta')}(u,w)$
end while
return u
```

that for all $K$ that have $y_i^\ell$ as vertex:

$$\big(|\bar{b}_i^\alpha|_K + \Delta x_K \|\bar{c}_i^\alpha\|_{L^\infty(K)}\big) \leq \bar{v}_i^{(\alpha,\beta),\ell} \sin(\vartheta)\,|\nabla \hat{\phi}_i^\ell|_K \operatorname{vol}(K), \tag{5.15a}$$

$$\big(|\bar{\bar{b}}_i^\alpha|_K + \Delta x_K \|\bar{\bar{c}}_i^\alpha\|_{L^\infty(K)}\big) \leq \bar{\bar{v}}_i^{(\alpha,\beta),\ell} \sin(\vartheta)\,|\nabla \hat{\phi}_i^\ell|_K \operatorname{vol}(K). \tag{5.15b}$$

Choose $\bar{a}_i^{(\alpha,\beta)}$ and $\bar{\bar{a}}_i^{(\alpha,\beta)}$ both in $C(\overline{\Omega})$ such that $\bar{a}_i^{(\alpha,\beta)}(y_i^\ell) \geq \max\{\tilde{a}_i^{(\alpha,\beta)}(y_i^\ell), \bar{v}_i^{(\alpha,\beta),\ell}\}$ and $\bar{\bar{a}}_i^{(\alpha,\beta)}(y_i^\ell) \geq \max\{\tilde{\tilde{a}}_i^{(\alpha,\beta)}(y_i^\ell), \bar{\bar{v}}_i^{(\alpha,\beta),\ell}\}$. This way we obtain a new splitting of $L^{(\alpha,\beta)}$ into implicit and explicit parts identified with matrices $\mathsf{I}_i^{(\alpha,\beta)}$ and $\mathsf{E}_i^{(\alpha,\beta)}$ respectively.

It is shown in [68] that (5.15) implies Assumption 11. Moreover, the splittings $a^{(\alpha,\beta)} = \tilde{a}_i^{(\alpha,\beta)} + \tilde{\tilde{a}}_i^{(\alpha,\beta)}$, $b^{(\alpha,\beta)} = \bar{b}_i^{(\alpha,\beta)} + \bar{\bar{b}}_i^{(\alpha,\beta)}$, $c^{(\alpha,\beta)} = \bar{c}_i^{(\alpha,\beta)} + \bar{\bar{c}}_i^{(\alpha,\beta)}$ and $f^{(\alpha,\beta)} = f_i^{(\alpha,\beta)}$ can always be chosen so that also Assumption 10 holds.

## 5.5 Wellposedness and a solution algorithm

In this section we address the existence and uniqueness of numerical solutions and propose a method to compute them.

**Theorem 11.** *There exists a unique numerical solution $v_i \colon S_i \to V_i^0$ that solves* (5.8).

*Proof.* Fix $k$ and $v_i(s_i^{k+1},\cdot)$. Then [14, Theorem 5.2] shows the existence and uniqueness of a solution $v_i(s_i^k,\cdot)$ of (5.8), noting the continuity of $(\alpha,\beta) \to \mathsf{I}_i^{(\alpha,\beta)}$ follows from Assumption 10 and the monotonicity from Assumption 11. □

The proof of Theorem 5.2 of [14], which we used here, relies on the exact solution of the inner Bellman equation (5.9a), which is reached by a Howard's algorithm in the limit. Using these exact solutions of (5.9a) in an outer Howard loop yields a sequence whose limit is the solution of (5.8).

To ensure the completion of the algorithm in finite time we require a termination criterion for the inner and outer loop. In Algorithm 1 (Howard's method) the termination criterion is posed in terms of a tolerance

$\eta$. The statement of the algorithm also refers to

$$\Psi^{(\alpha,\beta)}(u,w) := (h_i\mathsf{I}_i^{(\alpha,\beta)} + \mathsf{Id})\,u + (h_i\mathsf{E}_i^{(\alpha,\beta)} - \mathsf{Id})w - h_i\mathsf{F}_i^{(\alpha,\beta)},$$

given $u, w \in V_i^G$ and $(\alpha,\beta) \in A \times B$. Given just $w$, we let $\Phi^{(\alpha,\beta)}(w)$ be the solution $u$ of $\Psi^{(\alpha,\beta)}(u,w) = 0$. Note that this means that $u$ is updated as a function of $\beta$ selected during the previous iteration of the outer loop.

Suppose that $\sum_\ell \eta_\ell < \infty$ and that $w_\ell$ is the function returned by Algorithm 1 with $\eta = \eta_\ell$, $w = v_i(s_i^{k+1},\cdot)$ and a fixed $(\alpha_0,\beta_0) \in A \times B$. Then Theorem 5.4 of [14] ensures that the $w_\ell$ exist and converge to the unique numerical solution $v_i(s_i^k,\cdot)$ as $\ell \to \infty$.

## 5.6 Stability

For the Hamilton-Jacobi-Bellman equations one can bound the value function by the solution of any linear evolution problem associated to a fixed control $(\alpha,\beta) \in A \times B$. In contrast the value function of an Isaacs equation may lie in part above and in part below the solution of such a linear problem. Instead combinations of such linear operators need to be considered in order to bound the value function. This difference between the Bellman and Isaacs equations extends to the proof of stability of numerical solutions. We begin by adapting [68, Lemma 3.2].

**Lemma 8.** *One has $\|(h_i\mathsf{I}_i^w + \mathsf{Id})^{-1}\|_\infty \le 1$ and $\|h_i\mathsf{E}_i^w - \mathsf{Id}\|_\infty \le 1$ for all $i \in \mathbb{N}$ and $w \in V_i$, where the norms are the matrix $\infty$-norms.*

*Proof.* Define $v = \sum_{\ell=1}^{\dim V_i} \phi_i^\ell \equiv 1$, and $v_0 = \sum_{\ell=1}^{N_i} \phi_i^\ell \in V_i^0$. By Lemma 7, $h_i\mathsf{I}_i^w + \mathsf{Id}$ is an invertible M-matrix on $V_i^0$. Thus, $(h_i\mathsf{I}_i^w + \mathsf{Id})^{-1} \ge 0$ entrywise, so

$$\|(h_i\mathsf{I}_i^w + \mathsf{Id})^{-1}\|_\infty = \max_{1\le\ell\le N_i} \sum_{j=1}^{N_i} (h_i\mathsf{I}_i^w + \mathsf{Id})^{-1}_{\ell j} = \max_{1\le\ell\le N_i} \big((h_i\mathsf{I}_i^w + \mathsf{Id})^{-1}\mathbf{1}\big)_\ell, \tag{5.16}$$

where $\mathbf{1} \in \mathbb{R}^{N_i}$ is the vector with all entries equal to 1. Since $\nabla v \equiv 0$ (as $v \equiv 1$) we have for each $1 \le \ell \le N_i$ that

$$((h_i\mathsf{I}_i^w + \mathsf{Id})v)_\ell = 1 + h_i\langle \bar{\bar{c}}_i^{(\alpha_i^\ell(w),\beta^\ell(w))}, \hat{\phi}_i^\ell\rangle \ge 1,$$

where we have used non-negativity of $\bar{\bar{c}}_i^\alpha$ and defined $(\alpha_i^\ell(w),\beta^\ell(w))$ analogously to $(\alpha_i^{\ell,k}(w),\beta_i^{\ell,k}(w))$ in (5.9). Moreover, since $1 \le \ell \le N_i$ and $\mathsf{I}_i^w$ satisfies the LMP,

$$\big((h_i\mathsf{I}_i^w + \mathsf{Id})\,v\big)_\ell = \big((h_i\mathsf{I}_i^w + \mathsf{Id})\,v_0\big)_\ell + \sum_{j=N_i+1}^{\dim V_i} \underbrace{(h_i\mathsf{I}_i^w)_{\ell j}}_{\le 0} \le \big((h_i\mathsf{I}_i^w + \mathsf{Id})\,v_0\big)_\ell.$$

Because $(h_i\mathsf{I}_i^w + \mathsf{Id})v \ge \mathbf{1}$, we obtain $(h_i\mathsf{I}_i^w + \mathsf{Id})v_0 \ge \mathbf{1}$. So, after applying $(h_i\mathsf{I}_i^w + \mathsf{Id})^{-1}$ to both sides of this inequality, inverse positivity of $h_i\mathsf{I}_i^w+\mathsf{Id}$ gives $1\equiv v\geq v_0\geq (h_i\mathsf{I}_i^w+\mathsf{Id})^{-1}\mathbf{1}$ on $\overline{\Omega}$. This inequality and (5.16) imply $\|(h_i\mathsf{I}_i^w+\mathsf{Id})^{-1}\|_\infty\leq 1$.

One has $\|h_i\mathsf{E}_i^w-\mathsf{Id}\|_\infty=\max_{1\leq\ell\leq N_i}\big(-(h_i\mathsf{E}_i^w-\mathsf{Id})\,v_0\big)_\ell$ because all entries of the matrix $h_i\mathsf{E}_i^w-\mathsf{Id}$ are non-positive. For each $1\leq\ell\leq N_i$,

$$\big((h_i\mathsf{E}_i^w-\mathsf{Id})\,v\big)_\ell=\big((h_i\mathsf{E}_i^w-\mathsf{Id})\,v_0\big)_\ell+\sum_{j=N_i+1}^{\dim V_i}(h_i\mathsf{E}_i^w)_{\ell j}\leq\big((h_i\mathsf{E}_i^w-\mathsf{Id})\,v_0\big)_\ell,$$

so

$$(-(h_i\mathsf{E}_i^w-\mathsf{Id})\,v_0)_\ell\leq(-(h_i\mathsf{E}_i^w-\mathsf{Id})\,v)_\ell=1-h_i\langle\bar{c}_i^{(\alpha_i^\ell(w),\beta^\ell(w))},\hat{\phi}_i^\ell\rangle\leq 1$$

because $\bar{c}_i^{(\alpha_i^\ell(w),\beta^\ell(w))}\geq 0$ where $\bar{c}_i^{(\alpha_i^\ell(w),\beta^\ell(w))}$ is defined analogously to $\bar{\bar{c}}_i^{(\alpha_i^\ell(w),\beta^\ell(w))}$ above. Therefore, $-(h_i\mathsf{E}_i^w-\mathsf{Id})v_0\leq\mathbf{1}$. So $\|h_i\mathsf{E}_i^w-\mathsf{Id}\|_\infty\leq 1$.

□

**Theorem 12.** *The numerical solutions $v_i$ are uniformly bounded in the $L^\infty$ norm:*

$$\begin{aligned}\|v_i\|_{L^\infty(S_i\times\Omega)}\leq{}&\|P_iv_T\|_{L^\infty(\Omega)}+\|P_ig\|_{L^\infty(\Omega)}\\&+T\sup_{(\alpha,\beta)}\Big[\Big(\|\bar{a}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}+\|\bar{\bar{a}}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}\Big)\,\|g\|_{W^{2,\infty}(\Omega)}\\&+\Big(\|\bar{b}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}+\|\bar{\bar{b}}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}\Big)\,\|P_ig\|_{W^{1,\infty}\Omega)}\\&+\Big(\|\bar{c}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}+\|\bar{\bar{c}}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}\Big)\,\|P_ig\|_{L^\infty(\Omega)}+\|f_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}\Big].\end{aligned}$$

*Proof.* We split the numerical solution into two parts

$$v_i(s_i^k,\cdot)=v_i^0(s_i^k,\cdot)+g_i,\qquad P_iv_T=v_T^0+g_i,$$

where $v_i^0(s_i^k,\cdot)\in V_i^0$ and $g_i=P_ig\in V_i^g$. Then (5.10) becomes

$$0=(h_i\mathsf{I}_i^{k,v_i}+\mathsf{Id})\,(v_i^0(s_i^k,\cdot)+g_i)+(h_i\mathsf{E}_i^{k,v_i}-\mathsf{Id})\,(v_i^0(s_i^{k+1},\cdot)+g_i)-h_i\mathsf{F}_i^{k,v_i}$$

or equivalently

$$v_i^0(s_i^k,\cdot)=(h_i\mathsf{I}_i^{k,v_i}+\mathsf{Id})^{-1}\big(h_i\mathsf{F}_i^{k,v_i}-h_i(\mathsf{I}_i^{k,v_i}+\mathsf{E}_i^{k,v_i})\,g_i-(h_i\mathsf{E}_i^{k,v_i}-\mathsf{Id})\,v_i^0(s_i^{k+1},\cdot)\big).$$

Using (5.4), we find

$$\begin{aligned}\|(\mathsf{I}_i^{k,v_i}+\mathsf{E}_i^{k,v_i})\,g_i\|_{L^\infty(\Omega)} \leq & \left(\|\bar{a}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}+\|\bar{\bar{a}}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}\right)\|g\|_{W^{2,\infty}(\Omega)}\\ &+\left(\|\bar{b}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}+\|\bar{\bar{b}}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}\right)\|g_i\|_{W^{1,\infty}\Omega)}\\ &+\left(\|\bar{c}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}+\|\bar{\bar{c}}_i^{(\alpha,\beta)}\|_{L^\infty(\Omega)}\right)\|g_i\|_{L^\infty(\Omega)}.\end{aligned}$$

Now an application of Lemma 8 and inductive argument over the timesteps complete the proof. □

## 5.7 Sub- and supersolutions and uniform convergence

Recall the definition of the upper and lower semi-continuous envelopes $v^*$ and $v_*$ of a function $v$ and consider their numerical equivalent defined as follows

$$\overline{v}^*(t,x)=\sup_{(s_i^k,y_i^\ell)\to(t,x)}\ \limsup_{i\to\infty} v_i(s_i^k,y_i^\ell),\qquad \underline{v}^*(t,x)=\inf_{(s_i^k,y_i^\ell)\to(t,x)}\ \liminf_{i\to\infty} v_i(s_i^k,y_i^\ell). \tag{5.17}$$

Owing to Theorem 11 and Corollary 12, $\overline{v}^*$ and $\underline{v}^*$ attain finite values. By construction, $\overline{v}^*$ is upper and $\underline{v}^*$ lower semi-continuous and $\underline{v}^*\leq\overline{v}^*$.

**Theorem 13.** *The function* $\overline{v}^*$ *is a viscosity subsolution of* (5.3a) *and* $\underline{v}^*$ *is a viscosity supersolution of* (5.3a).

*Proof.* We show that $\overline{v}^*$ is a subsolution. That $\underline{v}^*$ is a supersolution follows analogously up to an asymmetry in the sign of $\gamma|\mu_i|$ in (5.20). Suppose that $w\in C^\infty(\mathbb{R}\times\mathbb{R}^d)$ is a test function such that $\overline{v}^*-w$ has a strict local maximum at $(s,y)\in(0,T)\times\Omega$, with $\overline{v}^*(s,y)=w(s,y)$. Then following the argument in [68] there exists a sequence $\{s_i^k,y_i^\ell\}_i$ such that

$$v_i(s_i^k,y_i^\ell)-P_iw(s_i^k,y_i^\ell)\to\overline{v}^*(s,y)-w(s,y)=0 \tag{5.18}$$

and

$$v_i(s_i^\kappa,y_i^\lambda)-P_iw(s_i^\kappa,y_i^\lambda)\leq v_i(s_i^k,y_i^\ell)-P_iw(s_i^k,y_i^\ell)\ \Leftrightarrow\ P_iw(s_i^\kappa,y_i^\lambda)+\mu_i\geq v_i(s_i^\kappa,y_i^\lambda), \tag{5.19}$$

where $\kappa\in\{k,k+1\}$, $y_i^\lambda\in\operatorname{supp}\hat{\phi}_i^\ell$ and $\mu_i=v_i(s_i^k,y_i^\ell)-P_iw(s_i^k,y_i^\ell)$. Notice that $\mu_i\to0$ as $i\to\infty$ because of (5.18).

As in [68] we conclude that

$$\left((h_i\mathsf{E}_i^{(\alpha,\beta)}-\mathsf{Id})\left[P_iw(s_i^{k+1},\cdot)+\mu_i\right]\right)_\ell\leq\left((h_i\mathsf{E}_i^{(\alpha,\beta)}-\mathsf{Id})v_i(s_i^{k+1},\cdot)\right)_\ell$$

and

$$\left((h_i\mathsf{I}_i^{(\alpha,\beta)}+\mathsf{Id})\left[P_iw(s_i^k,\cdot)+\mu_i\right]\right)_\ell \le \left((h_i\mathsf{I}_i^{(\alpha,\beta)}+\mathsf{Id})v_i(s_i^k,\cdot)\right)_\ell .$$

From the definition of the scheme (5.8),

$$\begin{aligned}
0 =& -d_iv_i(s_i^k,y_i^\ell)+\inf_{\beta\in B}\sup_{\alpha\in A}\left(\mathsf{E}_i^{(\alpha,\beta)}v_i(s_i^{k+1},\cdot)+\mathsf{I}_i^{(\alpha,\beta)}v_i(s_i^k,\cdot)-\mathsf{F}_i^{(\alpha,\beta)}\right)_\ell\\
\ge& -d_i\left(P_iw(s_i^k,y_i^\ell)+\mu_i\right)\\
&+\inf_{\beta\in B}\sup_{\alpha\in A}\left(\mathsf{E}_i^{(\alpha,\beta)}\left(P_iw(s_i^{k+1},\cdot)+\mu_i\right)+\mathsf{I}_i^{(\alpha,\beta)}\left(P_iw(s_i^k,\cdot)+\mu_i\right)-\mathsf{F}_i^{(\alpha,\beta)}\right)_\ell\\
=& -d_iP_iw(s_i^k,y_i^\ell)\\
&+\inf_{\beta\in B}\sup_{\alpha\in A}\left[\left(\mathsf{E}_i^{(\alpha,\beta)}P_iw(s_i^{k+1},\cdot)+\mathsf{I}_i^{(\alpha,\beta)}P_iw(s_i^k,\cdot)-\mathsf{F}_i^{(\alpha,\beta)}\right)_\ell+\mu_i\langle\bar{c}_i^{(\alpha,\beta)}+\bar{\bar{c}}_i^{(\alpha,\beta)},\hat{\phi}_i^\ell\rangle\right]\\
\ge& -d_iP_iw(s_i^k,y_i^\ell)+\inf_{\beta\in B}\sup_{\alpha\in A}\left(\mathsf{E}_i^{(\alpha,\beta)}P_iw(s_i^{k+1},\cdot)+\mathsf{I}_i^{(\alpha,\beta)}P_iw(s_i^k,\cdot)-\mathsf{F}_i^{(\alpha,\beta)}\right)_\ell-\gamma|\mu_i|. \qquad (5.20)
\end{aligned}$$

Using

$$\begin{aligned}
&\left|\inf_{\beta\in B}\sup_{\alpha\in A}\left(\mathsf{E}_i^{(\alpha,\beta)}P_iw(s_i^{k+1},\cdot)+\mathsf{I}_i^{(\alpha,\beta)}P_iw(s_i^k,\cdot)-\mathsf{F}_i^{(\alpha,\beta)}\right)_\ell-\inf_{\beta\in B}\sup_{\alpha\in A}\left(L^{(\alpha,\beta)}w(s,y)-f^{(\alpha,\beta)}(y)\right)\right|\\
\le& \sup_{(\alpha,\beta)\in A\times B}\left|\left(\mathsf{E}_i^{(\alpha,\beta)}P_iw(s_i^{k+1},\cdot)+\mathsf{I}_i^{(\alpha,\beta)}P_iw(s_i^k,\cdot)-\mathsf{F}_i^{(\alpha,\beta)}\right)_\ell-\left(L^{(\alpha,\beta)}w(s,y)-f^{(\alpha,\beta)}(y)\right)\right|,
\end{aligned}$$

the result of Lemma 6 and the fact that $\mu_i\to 0$ we take the limit $i\to\infty$ in inequality (5.20) and conclude that

$$0\ge -\partial_t w(s,y)+\inf_{\beta\in B}\sup_{\alpha\in A}\left(L^{(\alpha,\beta)}w(s,y)-f^{\alpha,\beta}(y)\right). \qquad (5.21)$$

Hence $\overline{v}^*$ is a viscosity subsolution. □

**Lemma 9.** *The sub and supersolutions $\overline{v}^*$ and $\underline{v}^*$ satisfy*

$$\overline{v}^*(T,\cdot)=\underline{v}^*(T,\cdot)=v_T \quad \text{on } \overline{\Omega}. \qquad (5.22)$$

*Proof.* The proof is identical to the Hamilton-Jacobi-Bellman case described in [68] once control $\alpha\in A$ is replaced by the pair of controls $(\alpha^{v_i(\ell)},\beta^{v_i(\ell)})\in A\times B$ similarly as in the proof of Lemma 8. □

**Assumption 12.** *Let $\overline{v}$ be a lower semi-continuous supersolution with $\overline{v}(T,\cdot)=v_T$. Similarly, let $\underline{v}$ be an upper semi-continuous subsolution with $\underline{v}(T,\cdot)=v_T$. Let $\max(-\partial_t\overline{v}+H\overline{v},\overline{v}-g(t,x))\ge 0$ for all $(t,x)\in[0,T]\times\partial\Omega$ and let $\min(-\partial_t\underline{v}+H\underline{v},\underline{v}-g(t,x))\le 0$ for all $(t,x)\in[0,T]\times\partial\Omega$. Then $\underline{v}\le\overline{v}$.*

**Theorem 14.** *One has $\underline{v}^*=\overline{v}^*=v$, where $v$ is the unique viscosity solution of equation* (5.3) *with $v(T,\cdot)=$*

*$v_T$ . Furthermore*

$$\lim_{i\to\infty} \|v_i - v\|_{L^\infty((0,T)\times\Omega)} = 0. \tag{5.23}$$

*Proof.* The proof is identical to the Hamilton-Jacobi-Bellman case described in [68]. □

## 5.8 Boundary conditions

The Dirichlet conditions for Isaacs equations are most commonly imposed in one of two ways: either in the classical sense, by requiring that the solution attains the Dirichlet data at all points or in the viscosity sense meaning that the solution is a sub- and supersolution defined through subtractive testing.

For the proof of the stability of the scheme we assume the existence of two families of barrier functions $(\zeta_{y,\varepsilon})_{\varepsilon>0}$ and $(\xi_{y,\varepsilon})_{\varepsilon>0}$ corresponding to the sets of super- and subsolutions respectively. They will be used in order to prove that the envelopes of the numerical solutions satisfy the Dirichlet boundary conditions in a pointwise sense in the boundary region $\omega$. This $\omega$ also appears in the comparison principle provided with Assumption 12, where $\omega$ characterises the set on which the boundary conditions are imposed pointwise.

Let $s = \vartheta s_i^k + (1-\vartheta)s_i^{k+1} \in [s_i^k, s_i^{k+1}]$ lie between two time steps, $\vartheta \in [0,1]$. Then we interpret $v_i(s,\cdot)$ as the linear interpolant between $v_i(s_i^k,\cdot)$ and $v_i(s_i^{k+1},\cdot)$:

$$v_i(s,\cdot) = \vartheta v_i(s_i^k,\cdot) + (1-\vartheta)v_i(s_i^{k+1},\cdot). \tag{5.24}$$

**Assumption 13.** *Let $\omega \subseteq \partial\Omega$. Let us assume the following:*

1. *Family of upper barriers*

   *For each $y \in \omega$ there exists a family of smooth barrier functions $(\zeta_{y,\varepsilon})_{\varepsilon>0}$ such that for all $\varepsilon$, $t \in [0,T]$ and $i > 0$, a minimiser over $\{t\} \times \Omega$ of $v_i - P_i\zeta_{y,\varepsilon}$ lies on $\{t\} \times \partial\Omega$. Let $q_{t,y,\varepsilon}$ be a minimiser of $g - \zeta_{y,\varepsilon}$ over $\{t\} \times \partial\Omega$. Then, $lim_{\varepsilon\to 0}\, q_{t,y,\varepsilon} = y$ and $\lim_{\varepsilon\to 0} \zeta_{y,\varepsilon}(t,y) - \zeta_{y,\varepsilon}\ (t, q_{t,y,\varepsilon}) \geq 0$.*

2. *Family of lower barriers*

   *For each $y \in \omega$ there exists a family of smooth barrier functions $(\xi_{y,\varepsilon})_{\varepsilon>0}$ such that for all $\varepsilon$, $t \in [0,T]$ and $i > 0$, a maximiser over $\{t\} \times \Omega$ of $v_i - P_i\xi_{y,\varepsilon}$ lies on $\{t\} \times \partial\Omega$. Let $r_{t,y,\varepsilon}$ be a maximiser of $g - \xi_{y,\varepsilon}$ over $\{t\} \times \partial\Omega$. Then, $lim_{\varepsilon\to 0}\, r_{t,y,\varepsilon} = y$ and $\lim_{\varepsilon\to 0} \xi_{y,\varepsilon}(t,y) - \xi_{y,\varepsilon}\ (t, r_{t,y,\varepsilon}) \leq 0$.*

Note that a minimiser (maximiser) over $\{t\} \times \Omega$ of $v_i - P_i\zeta_{y,\varepsilon}$ (resp.,$v_i - P_i\xi_{y,\varepsilon}$) is necessarily attained at a time node since $v_i - P_i\zeta_{y,\varepsilon}$ (resp.,$v_i - P_i\xi_{y,\varepsilon}$) is piecewise linear in time. In order to understand the connection between convergence of the envelopes to the boundary conditions and the barrier functions let us consider the following example.

**Example 1.** *We study a one-dimensional test problem with the homogenous boundary conditions:*

$$\begin{aligned} -\partial_t u + Lu - 1 &= 0, && \text{in } (0,T)\times(0,1),\\ u &= \begin{cases} x-1 & x\neq 0\\ 0 & x=0, \end{cases} && \text{on } \{T\}\times[0,1], \end{aligned}$$

*where $Lu = \nabla u$. In order to find solution we use a fully implicit numerical scheme with artificial diffusion.*

*The exact solution of $-u_t - \upsilon\Delta u + \nabla u - 1 = 0$, interpreting $\upsilon$ as the artificial diffusion coefficient, is:*

$$v_\upsilon(t,x) = x - 1 - \frac{1}{-1+e^{\frac{1}{\upsilon}}} + \frac{e^{\frac{1-x}{\upsilon}}}{-1+e^{\frac{1}{\upsilon}}}.$$

*For a fixed $\upsilon$ we can choose $\Delta x$ small enough such that the numerical solution $\tilde{v}_\upsilon$ with artificial diffusion $\upsilon$ and uniform mesh with meshsize $\Delta x$ satisfies $\|\tilde{v}_\upsilon - v_\upsilon\|_{W^{1,\infty}} < \upsilon$. We note that $v_\upsilon$ attains its minimum at*

$$x_{\min} = 1 + \upsilon \log\left[\frac{1}{\upsilon\left(e^{\frac{1}{\upsilon}}-1\right)}\right]$$

*with minimal value equal to*

$$v_\upsilon(t,x_{\min}) = \frac{1}{1-e^{\frac{1}{\upsilon}}} + \upsilon\left(1+\log\left[\frac{1}{\upsilon\left(e^{\frac{1}{\upsilon}}-1\right)}\right]\right).$$

*In particular, it follows that $x_{\min}\to 0$ and $v_\upsilon(t,x_{\min})\to -1$ as $\upsilon\to 0$. Since $\|\tilde{v}_\upsilon - v_\upsilon\|_{W^{1,\infty}} < \upsilon$, we conclude that the sequence of numerical solutions $\tilde{v}_\upsilon$ has the upper semi-continuous envelope*

$$\overline{v}^*(t,x) = \begin{cases} x-1 & x\neq 0,\\ 0 & x=0. \end{cases} \tag{5.26}$$

*However, since $\overline{v}^*$ has a discontinuity at* 0*, there cannot exist a $C^\infty$ barrier function as described in Assumption 13. As we decrease $\upsilon$, $v_\upsilon$ and hence $\tilde{v}_\upsilon$ can have arbitrarily large gradient in the vicinity of the boundary. As a result, for any barrier function $\zeta$ there exists $i$ for which minimum of $P_i\zeta_{y,\varepsilon} - \tilde{v}_\upsilon$ does not lie on the boundary for all $\varepsilon > 0$. Note that we can construct both upper and lower barrier functions at $x = 1$, e. g. $\zeta_1 = x^2$ and $\xi_1 = (x-1)^2$.*

**Lemma 10.** *Given Assumption 13, we have $v^*(t,x) = \underline{v}^*(t,x) = g(t,x)$ for all $(t,x)\in[0,T]\times\omega$.*

*Proof.* We focus on the case of $v^*(t,x)$ as the other case follows analogously. Let $y\in\omega$. By Assumption 13 we have that

$$\lim_{\varepsilon\to 0} r_{t,y,\varepsilon} = y.$$

Consider a sequence $(s_i^k, y_i^\ell) \to (t,y)$ as $i \to \infty$. We have that

$$\begin{aligned}
\limsup_{i\to\infty} v_i(s_i^k, y_i^\ell) =& \lim_{i\to\infty} P_i \xi_{y,\varepsilon}(s_i^k, y_i^\ell) + \limsup_{i\to\infty} [v_i(s_i^k, y_i^\ell) - P_i \xi_{y,\varepsilon}(s_i^k, y_i^\ell)] \\
\leq & \xi_{y,\varepsilon}(t,y) + \limsup_{i\to\infty} \sup_{z\in\partial\Omega} [v_i(s_i^k, z) - P_i \xi_{z,\varepsilon}(s_i^k, z)] \\
= & \xi_{y,\varepsilon}(t,y) + \limsup_{i\to\infty} \sup_{z\in\partial\Omega} [P_i g(s_i^k, z) - P_i \xi_{y,\varepsilon}(s_i^k, z)] \\
= & \xi_{y,\varepsilon}(t,y) + \limsup_{i\to\infty} \sup_{z\in\partial\Omega} \Big[ P_i g(t,z) - P_i \xi_{y,\varepsilon}(t,z) \\
& + \Big( P_i g(s_i^k, z) - P_i g(t,z) \Big) - \Big( P_i \xi_{y,\varepsilon}(s_i^k, z) - P_i \xi_{y,\varepsilon}(t,z) \Big) \Big] \\
\overset{(a)}{=} & \xi_{y,\varepsilon}(t,y) + \limsup_{i\to\infty} \sup_{z\in\partial\Omega} [P_i g(t,z) - P_i \xi_{y,\varepsilon}(t,z)] \\
= & \xi_{y,\varepsilon}(t,y) + \limsup_{i\to\infty} \sup_{z\in\partial\Omega} [(g(t,z) - \xi_{y,\varepsilon}(t,z)) \\
& + (P_i g(t,z) - g(t,z)) - (P_i \xi_{y,\varepsilon}(t,z) - \xi_{y,\varepsilon}(t,z))] \\
\overset{(b)}{=} & \xi_{y,\varepsilon}(t,y) + \limsup_{i\to\infty} \sup_{z\in\partial\Omega} [g(t,z) - \xi_{y,\varepsilon}(t,z)] \\
\overset{(c)}{=} & \xi_{y,\varepsilon}(t,y) + g(t, r_{t,y,\varepsilon}) - \xi_{y,\varepsilon}(t, r_{t,y,\varepsilon}),
\end{aligned} \tag{5.27}$$

where we get (a) due to Lipschitz continuity of $g$ and $\xi_{y,\varepsilon}$ in time, (b) due to $L^\infty$ convergence of $P_i g(t,z)$ and $P_i \xi_{y,\varepsilon}(t,z)$ as $i \to \infty$ and (c) due to the definition of $r_{t,y,\varepsilon}$. By Assumption 13 we have that $\lim_{\varepsilon\to 0} \xi_{y,\varepsilon}(t,y) - \xi_{y,\varepsilon}\ (t, r_{t,y,\varepsilon}) \leq 0$ for any $t \in [0,T]$, so we can conclude that

$$\limsup_{i\to\infty} v_i(s_i^k, y_i^\ell) \leq g(t,y), \tag{5.28}$$

as $\varepsilon \to 0$. The proof concludes by completing a similar calculation for $\liminf_{i\to\infty} v_i(s_i^k, y_i^\ell)$. This gives us:

$$g(t,y) \geq \limsup_{i\to\infty} v_i(s_i^k, y_i^\ell) \geq \liminf_{i\to\infty} v_i(s_i^k, y_i^\ell) \geq g(t,y),$$

and the final result follows. □

## 5.9 Construction of barrier functions

In order to justify Assumption 13 we now present settings for the construction of barrier functions. First, we introduce a method for ensuring the existence of barrier functions for the simpler case of uniformly parabolic operators on a convex domain for fully implicit numerical schemes. After that we show the extension to general IMEX schemes and allow non-convex domains as well as degenerate operators.

We will focus on constructing lower barrier functions $\xi$, since the argument for upper barrier functions $\zeta$ is symmetric and follows by changing the direction of inequalities and exchanging sup and inf operators where required.

### 5.9.1 Barrier functions for uniformly parabolic equations on convex domains

We assume the existence of a function $\xi \in \{w \in W^{1,\infty}([0,T] \times \overline{\Omega}) : \Delta\xi \in L^\infty\}$ which solves

$$-\lambda\Delta\xi = M + \gamma_1 \|\nabla\xi\|_{L^\infty(\mathbb{R}^d,\Omega)} + \gamma_2\|\xi\|_\infty \qquad \text{on } \Omega, \tag{5.29a}$$

$$\xi = g \qquad \text{on } \partial\Omega, \tag{5.29b}$$

where

$$\gamma_1 := \sup_{\substack{(s_i^k,y_i^\ell)\to(t,x)\\(\alpha,\beta)\in A\times B}} \|\bar{b}_i^{(\alpha,\beta)} + \bar{\bar{b}}_i^{(\alpha,\beta)}\|_\infty, \quad \gamma_2 := \sup_{\substack{(s_i^k,y_i^\ell)\to(t,x)\\(\alpha,\beta)\in A\times B}} \|\bar{c}_i^{(\alpha,\beta)} + \bar{\bar{c}}_i^{(\alpha,\beta)}\|_\infty$$

and

$$\lambda := \inf_{\substack{i\in\mathbb{N}\\(\alpha,\beta)\in A\times B}} \bar{a}_i^{(\alpha,\beta)} + \bar{\bar{a}}_i^{(\alpha,\beta)}, \quad M := \max\Big\{\lambda\|v_T\|_\infty, \sup_{i,\alpha,\beta} \mathsf{F}_i^{(\alpha,\beta)}(y_i^\ell) + 1\Big\}.$$

The Dirichlet boundary conditions of (5.29b) are imposed in the strong sense. Because $g$ does not depend on time it is clear that $\xi$ is constant in $t$. We shall therefore often write $\xi(x)$ instead of $\xi(t,x)$. The function $\xi$ is used as barrier for all $y \in \omega$ and $\varepsilon > 0$: $\xi_{y,\varepsilon} := \xi$.

We assume in this subsection that $\Omega$ is convex. Then, as outlined below Assumption 9, we can construct $P_i$ so that $P_i\xi$ interpolates $\xi$ on the boundary. Furthermore, we require for the construction of this section that $\Delta v_T \in L^\infty(\Omega)$ and $v_T = g$ on $\partial\Omega$.

To show that a maximum of

$$v_i(s_i^k,\cdot) - P_i\xi(x)$$

over $s_i^k \times \overline{\Omega}$ is attained on the boundary for all $s_i^k$ we use that the $h_i\mathsf{I}_i^{k,v_i} + \mathsf{Id}$ are M-matrices. Hence it is enough to prove that

$$\Big((h_i\mathsf{I}_i^{k,v_i} + \mathsf{Id})(P_i\xi - v_i(s_i^k,\cdot))\Big)_\ell \geq 0 \qquad \forall \ell \leq N_i. \tag{5.30}$$

#### Fully implicit numerical scheme

For the sake of clarity, we begin with a fully implicit numerical scheme of the form

$$(h_i\mathsf{I}_i^{k,v_i} + \mathsf{Id})\,v_i(s_i^k,\cdot) - v_i(s_i^{k+1},\cdot) - h_i\mathsf{F}_i^{k,v_i} = 0. \tag{5.31}$$

Recall how $(\alpha_i^{k,\ell}(v_i), \beta_i^{k,\ell}(v_i)) \in A \times B$ is a pair of controls such that $\alpha_i^{k,\ell}(v_i)$ is a maximiser of (5.9a) and $\beta_i^{k,\ell}(v_i)$ is a minimiser of (5.9b) for $w = v_i$. Noting that $-\langle\Delta\xi, \hat{\phi}_i^\ell\rangle = \langle\nabla P_i\xi, \nabla\hat{\phi}_i^\ell\rangle$ is positive due to (5.29), we find that

$$\begin{aligned}
&\Big((h_i\mathsf{I}_i^{k,v_i}+\mathsf{Id})(P_i\xi)\Big)_\ell \\
=\;& (P_i\xi)_\ell - h_i\,\bar{\bar{a}}_i^{(\alpha^{v_i(\ell)},\beta^{v_i(\ell)})}(y_i^\ell)\,\langle\Delta\xi,\hat{\phi}_i^\ell\rangle \\
&+h_i\langle -\bar{\bar{b}}_i^{(\alpha^{v_i(\ell)},\beta^{v_i(\ell)})}\cdot\nabla P_i\xi+\bar{\bar{c}}_i^{(\alpha^{v_i(\ell)},\beta^{v_i(\ell)})}P_i\xi,\hat{\phi}_i^\ell\rangle \\
=\;& (P_i\xi)_\ell - h_i\,\bar{\bar{a}}_i^{(\alpha^{v_i(\ell)},\beta^{v_i(\ell)})}(y_i^\ell)\,\langle\Delta\xi,\hat{\phi}_i^\ell\rangle \\
&+h_i\langle -\bar{\bar{b}}_i^{(\alpha^{v_i(\ell)},\beta^{v_i(\ell)})}\cdot\nabla\xi+\bar{\bar{c}}_i^{(\alpha^{v_i(\ell)},\beta^{v_i(\ell)})}\xi,\hat{\phi}_i^\ell\rangle \\
&+h_i\langle -\bar{\bar{b}}_i^{(\alpha^{v_i(\ell)},\beta^{v_i(\ell)})}\cdot\nabla(P_i\xi-\xi)+\bar{\bar{c}}_i^{(\alpha^{v_i(\ell)},\beta^{v_i(\ell)})}(P_i\xi-\xi),\hat{\phi}_i^\ell\rangle \\
\geq\;& (P_i\xi)_\ell+h_i\langle-\lambda\Delta\xi-\gamma_1\|\nabla\xi\|_\infty-\gamma_2\|\xi\|_\infty,\hat{\phi}_i^\ell\rangle \\
&-h_i\,(\gamma_1\|\nabla P_i\xi-\nabla\xi\|_\infty+\gamma_2\|P_i\xi-\xi\|_\infty) \\
\overset{(5.29)}{=}& (P_i\xi)_\ell+h_iM-h_i\,(\gamma_1\|\nabla P_i\xi-\nabla\xi\|_\infty+\gamma_2\|P_i\xi-\xi\|_\infty)\,.
\end{aligned}$$

Furthermore, there is a $\hat{i}\in\mathbb{N}$ such that for all $i\geq\hat{i}$.

$$\sup_i \gamma_1\|\nabla P_i\xi-\nabla\xi\|_\infty+\gamma_2\|P_i\xi-\xi\|_\infty\leq 1$$

Assuming $i\geq\hat{i}$ for the remainder of the section, and using the definition of $M$

$$\Big((h_i\mathsf{I}_i^{k,v_i}+\mathsf{Id})(P_i\xi)\Big)_\ell\geq(P_i\xi)_\ell+h_i\sup_{j,\alpha,\beta}\mathsf{F}_j^{(\alpha,\beta)}(y_j^\ell).$$

Recall that $\xi(x)=\xi(s_i^k,x)=\xi(s_i^{k+1},x)$. With the numerical scheme (5.31) we obtain

$$\Big((h_i\mathsf{I}_i^{k,v_i}+\mathsf{Id})(P_i\xi-v_i(s_i^k,\cdot))\Big)_\ell\geq(P_i\xi)_\ell-v_i(s_i^{k+1},y_i^\ell). \tag{5.32}$$

Because both $(P_i\xi)_\ell$ and $v_i(s_i^k,y_i^\ell)$ interpolate $g$ on $\partial\Omega$, it follows $P_i\xi-v_i(s_i^k,\cdot)\in V_i^0$. Owing to the M-matrix property of $h_i\mathsf{I}_i^{k,v_i}+\mathsf{Id}$,

$$(P_i\xi)_\ell-v_i(s_i^{k+1},y_i^\ell)\geq 0 \tag{5.33}$$

thus implies condition (5.30). Hence, by induction, (5.30) holds for all $k$ as soon as (5.33) is shown at the final time $s_i^{k+1}=T$.

From (5.29) we know that $-\lambda\Delta\xi\geq M$. Then, using $v_i(T,\cdot)=P_iv_T$ and $\lambda>0$,

$$\begin{aligned}
\langle\nabla\left(P_i\xi^T-v_i(T,\cdot)\right),\nabla\hat{\phi}_i^\ell\rangle&=-\langle\Delta(\xi-v_T),\hat{\phi}_i^\ell\rangle\\
&\geq\langle M/\lambda+\Delta v_T,\hat{\phi}_i^\ell\rangle\geq M/\lambda-\|\Delta v_T\|_\infty.
\end{aligned}$$

The definition of $M$ ensures that $\frac{M}{\lambda} \geq \|\Delta v_T\|_\infty$. Using M-matrix property of the discrete Laplacian formed with respect to the nodal basis $\{\hat{\phi}_i^\ell\}_\ell$ we obtain (5.33) at the final time.

At this point we proved that the maximum of $v_i(s_i^k,\cdot) - P_i\xi$ lies on the boundary as required. It remains to show that for $t \in [0,T]$ we can select a maximiser $r_{t,y,\varepsilon}$ of $g - \xi_{y,\varepsilon}$ over $\{t\} \times \partial\Omega$ such that $\lim_{\varepsilon\to 0}\ r_{t,y,\varepsilon} = y$ and $\lim_{\varepsilon\to 0} \xi_{y,\varepsilon}(t,y) - \xi_{y,\varepsilon}\ (t, r_{t,y,\varepsilon}) \leq 0$. Because $\xi$ attains $g$ on whole the boundary, $r_{t,y,\varepsilon} := y$ is already such a choice.

**IMEX numerical schemes**

We now extend the above argument to general numerical schemes as defined in (5.10). Choosing $\gamma_1, \gamma_2, \lambda$ and $M$ as above, the inequality (5.32) generalises in the IMEX setting to

$$\Big((h_i \mathsf{I}_i^{k,v_i} + \mathsf{Id})(P_i\xi - v_i(s_i^k,\cdot))\Big)_\ell \geq -\Big((h_i \mathsf{E}_i^{k,v_i} - \mathsf{Id})(P_i\xi - v_i(s_i^{k+1},\cdot))\Big)_\ell.$$

According to Lemma 7 we have $-(h_i \mathsf{E}_i^{k,v_i} - \mathsf{Id}) \geq 0$. Therefore, if

$$P_i\xi - v_i(s_i^{k+1},\cdot)) \geq 0$$

implies (5.30). At this point the induction of the previous subsection can be adapted to show that maxima are attained on the boundary. Setting $r_{t,y,\varepsilon} := y$ completes the construction.

### 5.9.2 Barrier functions for degenerate equations and general domains

We now want to remove the two main assumptions of section 5.9.1, namely the requirements that the differential operator is uniformly parabolic and that the domain is convex.

We form $\omega$ of the $y$ for which it is possible to ensure the existence of strict supersolutions in the following sense: for all $\varepsilon > 0$ one can find a $\xi_{y,\varepsilon}$ satisfying

$$\sup_\alpha -a^{(\alpha,\beta)}\Delta\xi_{y,\varepsilon} - b^{(\alpha,\beta)}.\nabla\xi_{y,\varepsilon} + c^{(\alpha,\beta)}\,\xi_{y,\varepsilon} - f^{(\alpha,\beta)} \geq \varepsilon \qquad \text{on } \Omega \tag{5.34}$$

for all $\beta \in C(\Omega;A)$ as well as

$$\sup_{\varepsilon>0}(\|\Delta\xi_{y,\varepsilon}\| + \|\xi_{y,\varepsilon}\|_{W^{1,\infty}(\Omega)}) < \infty$$

and

$$v_T - \xi_{y,\varepsilon} \leq -2\varepsilon \qquad \text{on } \overline{\Omega}\setminus B_y(\delta(\varepsilon)), \tag{5.35a}$$

$$v_T - \xi_{y,\varepsilon} \leq -\ \varepsilon \qquad \text{on } \overline{\Omega}\cap B_y(\delta(\varepsilon)), \tag{5.35b}$$

$$v_T(y) - \xi_{y,\varepsilon}(y) > -2\varepsilon, \tag{5.35c}$$

where $B_y(\delta(\varepsilon))$ is the ball centred at $y$ with radius $\delta(\varepsilon) > 0$, which in turn is a positive parameter depending on $\varepsilon$. Observe that $v_T|_{\partial\Omega} = g|_{\partial\Omega}$ and therefore (5.35) also provides control on the boundary, due to $g$ and $\xi_{y,\varepsilon}$ being time independent. It ensures that the maximum of $y - \xi_{y,\varepsilon}$ is attained in the vicinity of $y$ and is non-positive. We can view (5.34) as generalisation of (5.29a) because $\xi$ of (5.29a) is a strict supersolution of $L^{(\alpha,\beta)}w - f^{(\alpha,\beta)} = 0$ for all $\alpha \in A$, $\beta \in B$.

In light of Assumptions 9 and 10, we may choose $\hat{\imath}$ such that

$$C_1(\hat{\imath})\,(\|\Delta\xi_{y,\varepsilon}\| + \|\xi_{y,\varepsilon}\|_{W^{1,\infty}(\Omega)}) + C_2 \sup_{i\geq\hat{\imath}} \|P_i\xi_{y,\varepsilon} - \xi_{y,\varepsilon}\|_{W^{1,\infty}(\Omega)} \leq \varepsilon. \tag{5.36}$$

where

$$\begin{aligned} C_1(\hat{\imath}) := \sup_{\substack{(\alpha,\beta)\in A\times B\\ i\geq\hat{\imath}}} \Big(\sup_{\ell} \big\|a^{(\alpha,\beta)} - \big(\bar{a}_i^{(\alpha,\beta)}(y_i^\ell) + \bar{\bar{a}}_i^{(\alpha,\beta)}(y_i^\ell)\big)\big\|_{L^\infty(\operatorname{supp}\hat{\phi}_i^\ell)} \\ + \big\|b^{(\alpha,\beta)} - \big(\bar{b}_i^{(\alpha,\beta)} + \bar{\bar{b}}_i^{(\alpha,\beta)}\big)\big\|_{L^\infty(\Omega,\mathbb{R}^d)} \\ + \big\|c^{(\alpha,\beta)} - \big(\bar{c}_i^{(\alpha,\beta)} + \bar{\bar{c}}_i^{(\alpha,\beta)}\big)\big\|_{L^\infty(\Omega)} + \big\|f^{(\alpha,\beta)} - f_i^{(\alpha,\beta)}\big\|_{L^\infty(\Omega)}\Big) \end{aligned}$$

and

$$C_2 := \max\big\{1, \sup_{\substack{(\alpha,\beta)\\ i\in\mathbb{N}}} \|\bar{a}_i^{(\alpha,\beta)} + \bar{\bar{a}}_i^{(\alpha,\beta)}\|_\infty + \|\bar{b}_i^{(\alpha,\beta)} + \bar{\bar{b}}_i^{(\alpha,\beta)}\|_\infty + \|\bar{c}_i^{(\alpha,\beta)} + \bar{\bar{c}}_i^{(\alpha,\beta)}\|_\infty\big\}.$$

Recall the definition of $(\alpha_i^{k,\ell}(w), \beta_i^{k,\ell}(w)) \in A \times B$ in (5.9). For the sake of readability, we write $\hat{\beta}$ in place of $\beta_i^{k,\ell}(P_i\xi_{y,\varepsilon} - v_i)$ and $\hat{\alpha}$ in place of $\alpha_i^{k,\ell,\hat{\beta}}(P_i\xi_{y,\varepsilon} - v_i)$ in this section, i.e. $\hat{\alpha}$ and $\hat{\beta}$ are optimal choices when evaluating the numerical operator at $P_i\xi_{y,\varepsilon} - v_i$.

Then the numerical method, applied to $P_i\xi^{k+1} - v_i$ with the control of the infimum frozen at $P_i\xi_{y,\varepsilon} - v_i$, returns at the node $y_i^\ell$

$$\begin{aligned} &\big((h_i\mathsf{I}_i^{(\hat{\alpha},\hat{\beta})} + \mathsf{Id})(P_i\xi_{y,\varepsilon} - v_i(s_i^k,\cdot)) + (h_i\mathsf{E}_i^{(\hat{\alpha},\hat{\beta})} - \mathsf{Id})(P_i\xi_{y,\varepsilon} - v_i(s_i^{k+1},\cdot))\big)_\ell \\ =&\, h_i\big(-(\bar{a}_i^{(\hat{\alpha},\hat{\beta})}(y_i^\ell) + \bar{\bar{a}}_i^{(\hat{\alpha},\hat{\beta})}(y_i^\ell))\langle\Delta\xi_{y,\varepsilon}, \hat{\phi}_i^\ell\rangle - \langle(\bar{b}_i^{(\hat{\alpha},\hat{\beta})} + \bar{\bar{b}}_i^{(\hat{\alpha},\hat{\beta})}) \cdot \nabla P_i\xi_{y,\varepsilon}, \hat{\phi}_i^\ell\rangle \\ &\quad + \langle(\bar{c}_i^{(\hat{\alpha},\hat{\beta})} + \bar{\bar{c}}_i^{(\hat{\alpha},\hat{\beta})})P_i\xi_{y,\varepsilon} - f_i^{(\hat{\alpha},\hat{\beta})}, \hat{\phi}_i^\ell\rangle\big) \\ \geq&\, h_i\,\langle -a^{(\hat{\alpha},\hat{\beta})}\Delta\xi_{y,\varepsilon} - b^{(\hat{\alpha},\hat{\beta})} \cdot \nabla\xi_{y,\varepsilon} + c^{(\hat{\alpha},\hat{\beta})}\xi_{y,\varepsilon} - f^{(\hat{\alpha},\hat{\beta})}, \hat{\phi}_i^\ell\rangle - h_i\,\varepsilon \\ \geq&\, 0. \end{aligned}$$

We conclude with Lemma 7 that

$$P_i\xi_{y,\varepsilon} - v_i(s_i^{k+1},\cdot) \geq 0 \tag{5.37}$$

implies, for $\ell \le N_i$,

$$\big((h_i \mathsf{I}_i^{(\hat\alpha,\hat\beta)} + \mathsf{Id})(P_i\xi_{y,\varepsilon} - v_i(s_i^k,\cdot))\big)_\ell \ge \big(-(h_i \mathsf{E}_i^{(\hat\alpha,\hat\beta)} - \mathsf{Id})(P_i\xi_{y,\varepsilon} - v_i(s_i^{k+1},\cdot))\big)_\ell \ge 0.$$

Because unlike in the case of subsection 5.9.1 the functions $P_i\xi_{y,\varepsilon} - v_i(s_i^k,\cdot)$ do not vanish on the boundary we need to extend this result to the case when $N_i < \ell \le \dim V_i$. Owing to (5.36) and (5.35b) we know that (5.37) also holds on the boundary for all time steps $s_i^{k+1}$. Furthermore, (5.35b) implies that (5.37) is satisfied on all of $\overline{\Omega}$ at the final time $s_i^{k+1} = T$. In summary we have the induction step that if (5.37) on $\overline{\Omega}$ then $P_i\xi_{y,\varepsilon} - v_i(s_i^k,\cdot) \ge 0$ on $\overline{\Omega}$ and the induction base to guarantee that the maximum of

$$v_i(s_i^k,\cdot) - P_i\xi(x)$$

over $s_i^k \times \overline{\Omega}$ is attained on the boundary for all $s_i^k$.

It remains to show that for $t \in [0,T]$ we can select a maximiser $r_{t,y,\varepsilon}$ of $g - \xi_{y,\varepsilon}$ over $\{t\} \times \partial\Omega$ such that $\lim_{\varepsilon\to 0}\ r_{t,y,\varepsilon} = y$ and $\lim_{\varepsilon\to 0} \xi_{y,\varepsilon}(t,y) - \xi_{y,\varepsilon}\ (t, r_{t,y,\varepsilon}) \le 0$. The former holds because of (5.35a) and (5.35c), while the latter follows from (5.35b).

Similarly we assume the existence of strict subsolutions in the following sense: For all $y \in \omega$ and $\varepsilon > 0$ one can find a $\zeta_{y,\varepsilon}$ satisfying

$$\inf_\beta -a^{(\alpha,\beta)}\Delta\zeta_{y,\varepsilon} - b^{(\alpha,\beta)}.\nabla\zeta_{y,\varepsilon} + c^{(\alpha,\beta)}\,\zeta_{y,\varepsilon} - f^{(\alpha,\beta)} \le -\varepsilon \qquad \text{on } \Omega$$

for all $\alpha \in C(\Omega;A)$ as well as

$$\sup_{\varepsilon>0}(\|\Delta\zeta_{y,\varepsilon}\| + \left\|\zeta_{y,\varepsilon}\right\|_{W^{1,\infty}(\Omega)}) < \infty$$

and

$$\begin{aligned} v_T - \zeta_{y,\varepsilon} &\ge 2\varepsilon && \text{on } \overline{\Omega}\setminus B_y(\delta(\varepsilon)),\\ v_T - \zeta_{y,\varepsilon} &\ge \ \ \varepsilon && \text{on } \overline{\Omega}\cap B_y(\delta(\varepsilon)),\\ v_T(y) - \zeta_{y,\varepsilon}(y) &< 2\varepsilon. \end{aligned}$$

## 5.10 Numerical experiments

In this section we present two numerical experiments showing the viability and an interesting use case of the presented method. In the first experiment, we analyse convergence rates of a fully nonlinear second order Isaacs problem with a known solution and we confirm at least linear convergence in $L^2$, $L^\infty$ and $H^1$ norms. In the second experiment we calculate the value function of a stochastic tag-chase game with asymmetric velocities and vanishing diffusion on a nonconvex domain.

### Isaacs problem with the exact solution

We start by considering a problem with a manufactured solution in order to have a look at the rates of convergence. Let us consider a spatial domain $\Omega$ in $\mathbb{R}^2$ which is an equilateral triangle with vertices $(\pm\sqrt{3},\ \frac{1}{2})$ and $(0,1)$. We will study the following Isaacs problem:

$$-v_t - \inf_{\beta\in[\frac{1}{4},\ \frac{1}{2}]}\{\beta\sqrt{\frac{x^2+y^2}{T-t+1}}\Delta v + \frac{1}{2}\frac{1}{\sqrt{T-t+1}}|\nabla v|\} = -\frac{1}{2}\frac{\sqrt{x^2+y^2}}{(T-t+1)^{3/2}}. \tag{5.38}$$

We can see that (5.38) is indeed an Isaacs problem by using the fact that Euclidean norm of the gradient may be defined alternatively as

$$|\nabla v| = \sup_{\{\alpha\in\mathbb{R}^2:|\alpha|=1\}}\{\alpha\cdot\nabla v\}.$$

One can now verify through a direct calculation that the function

$$v(x,y,t) = \exp\left(-\sqrt{\frac{x^2+y^2}{T-t+1}}\right) + \sqrt{\frac{x^2+y^2}{T-t+1}}$$

solves (5.38) exactly. Our numerical scheme still requires the boundary and final-time data which we obtain by interpolation of the exact solution $v$. The numerical solution is then acquired for the time interval $[0,1)$.

We split the scheme in such a way that the advection term is treated explicitly. Note that in order to ensure the monotonicity of the scheme we may need to introduce some diffusion into explicit terms. In order to improve the rate of convergence it was done locally, i.e. on a node-wise basis. In case the naturally occurring diffusion at the node is not sufficient, we introduce the artificial diffusion. This approach leads to higher values of the artificial diffusion around the origin where degeneracy of the differential operator occurs, while for the majority of nodes in the mesh it is zero or almost zero. We choose the largest timestep guaranteeing the monotonicity of the method which leads to $O(h_i) = O(\Delta x_i)$. If any amount of the diffusion is left after ensuring the monotonicity of explicit operators, it is treated implicitly.

The stability of the scheme is reflected on Figure 5.1 which plots errors at $t = 0$ for different mesh sizes. The rates of convergence are as follows:

| $\Delta x$ | $L^\infty$ | Rate | $L^2$ | Rate | $H^1$ | Rate |
|---|---|---|---|---|---|---|
| 0.4330 | 1.364e-02 | 0.66 | 7.062e-03 | 0.96 | 6.792e-02 | 0.91 |
| 0.2165 | 1.040e-02 | 0.91 | 3.695e-03 | 0.78 | 3.737e-02 | 0.98 |
| 0.1083 | 5.715e-03 | 1.01 | 2.361e-03 | 0.93 | 1.899e-02 | 1.01 |
| 0.0541 | 2.838e-03 | 1.07 | 1.266e-03 | 1.02 | 9.391e-03 | 1.03 |
| 0.0271 | 1.330e-03 | 1.10 | 6.234e-04 | 1.06 | 4.581e-03 | 1.03 |
| 0.0135 | 6.034e-04 | 1.11 | 2.941e-04 | 1.07 | 2.223e-03 | 1.03 |
| 0.0068 | 2.708e-04 | | 1.374e-04 | | 1.082e-03 | |

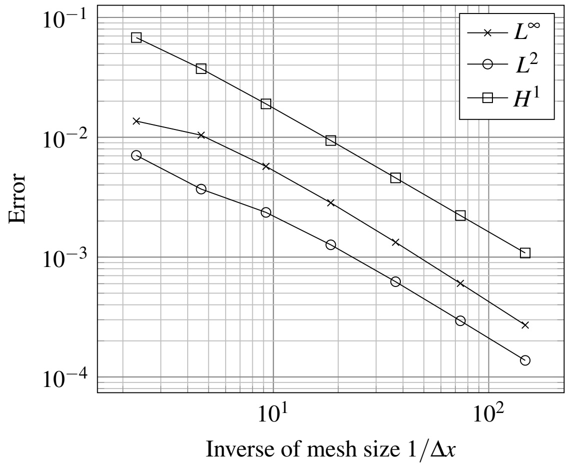


Figure 5.1: Approximation error of Experiment 1

## The Tag-Chase Game with random noise

Imagine two players moving on $\mathbb{R}^2$ plane. One player is the pursuer (we will denote him by P) and tries to catch the other one, who is the evader (we will denote him by E). The pursuer moves with speed $v_P$ while the evader moves with speed $v_E$. Both of them are allowed to choose their direction freely. In other words, their sets of admissible controls are the unit circle. Additionally, some choices of a direction for the pursuer and the evader may be subject to a random noise behaving like a standard Brownian motion. Having specified the setting we are able to formulate the dynamics explicitly as follows

$$\begin{cases} dx_1^P = (v_P)_1 \sin(\beta)dt + \sigma_P^{(\alpha,\beta)}(x_1^P)dW_{(1,P)} \\ dx_2^P = (v_P)_2 \cos(\beta)dt + \sigma_P^{(\alpha,\beta)}(x_2^P)dW_{(2,P)} \\ dx_1^E = (v_E)_1 \sin(\alpha)dt + \sigma_E^{(\alpha,\beta)}(x_1^E)dW_{(1,E)} \\ dx_2^E = (v_E)_2 \cos(\alpha)dt + \sigma_E^{(\alpha,\beta)}(x_2^E)dW_{(2,E)}, \end{cases}$$

where $\alpha, \beta \in [-\pi, \pi]$ and $\alpha$ is the direction chosen by the evader, while $\beta$ is the direction chosen by the pursuer. Additionally, we assume that

$$W_P := \{W_P(t) = (W_{(1,P)}(t), W_{(2,P)}(t)), t \geq 0\},$$
$$W_E := \{W_E(t) = (W_{(1,E)}(t), W_{(2,E)}(t)), t \geq 0\},$$

are two $\mathbb{R}^2$-valued, mutually independent standard Wiener processes. We reduce this 4-dimensional problem to a 2-dimensional one by allowing the origin of the coordinate system to move along with the pursuer.

In this case our dynamics are

$$\begin{cases} d\hat{x}_1 = ((v_P)_1 sin(\beta) - (v_E)_1 \sin(\alpha))dt + \sigma^{(\alpha,\beta)}(\hat{x}_1)d\hat{W}_1 \\ d\hat{x}_2 = ((v_P)_2 cos(\beta) - (v_E)_2 \cos(\alpha))dt + \sigma^{(\alpha,\beta)}(\hat{x}_2)d\hat{W}_2, \end{cases} \tag{5.39}$$

where $\sigma^{(\alpha,\beta)}(\hat{x}_i) = \sqrt{\sigma_P^{(\alpha,\beta)}(x_i^P))^2 + (\sigma_E^{(\alpha,\beta)}(x_i^E))^2}$ and $\hat{W}$ is an $\mathbb{R}^2$-valued standard Wiener process satisfying

$$\sigma^{(\alpha,\beta)}(\hat{x}_i)\hat{W}_i(t) = \sigma_P^{(\alpha,\beta)}(x_i^P)W_{i,P}(t) - \sigma_E^{(\alpha,\beta)}(x_i^E)W_{i,E}(t), \quad t \geq 0,\ i = 1,2.$$

The pursuer catches the evader (and thus wins the game) if they manage to reduce their distance from the evader to some value $r \in \mathbb{R}$. The evader wins the game when they manage to increase the distance to pursuer to some given $R > r$ or if they manage to avoid the capture before some time $T$. Note that in this case the spatial domain of the problem becomes $\overline{\Omega} := \overline{B}(0,R) \setminus B(0,r)$.

Mathematically, the evader receives the pay-out of 1 whenever they win the game and receive zero otherwise. Let $p(\alpha,\beta,x,t)$ be the probability that the path $(\hat{x}_1,\hat{x}_2)$ intersects $\partial B(0,r)$ before either time $T$ is reached or the path intersects $\partial B(0,R)$, assuming that $(\hat{x}_1(t),\hat{x}_2(t)) = x$ and that $(\hat{x}_1,\hat{x}_2)$ admits (5.39) with the $\alpha,\beta$ appearing in the arguments of $p$.

Then $1 - p$ is the expected pay-out to the evader:

$$\mathcal{J}(\alpha,\beta,x,t) := 1 - p(\alpha,\beta,x,t)$$

The value function is then

$$v(x,t) := \inf_{\beta} \sup_{\alpha} \mathcal{J}(\alpha,\beta,x,t)$$

and it solves the following second order Isaacs equation

$$\begin{aligned} -\partial_t v - a^{(\alpha,\beta)}\Delta v(x,t) + \inf_{\beta}\sup_{\alpha}\{b^{(\alpha,\beta)} \cdot \nabla v(x,t)\} &= 0 && \text{in } (0,T) \times \Omega, \\ v &= \begin{cases} 1 : \|\hat{x}\|_{\mathbb{R}^2} = R \\ 0 : \|\hat{x}\|_{\mathbb{R}^2} = r \end{cases} && \text{on } (0,T) \times \partial\Omega, \\ v &= 1 && \text{on } \{T\} \times \overline{\Omega}. \end{aligned}$$

where $a^{(\alpha,\beta)} := \frac{1}{2}\mathrm{tr}(\sigma^{(\alpha,\beta)} \cdot (\sigma^{(\alpha,\beta)})^T)$ and $b^{\alpha,\beta} := \begin{pmatrix} (v_P)_1 sin(\beta) - (v_E)_1 \sin(\alpha) \\ (v_P)_2 cos(\beta) - (v_E)_2 \cos(\alpha) \end{pmatrix}$.

We now assume that the pursuer is faster than the evader when moving in the horizontal direction. Moreover we assume that there is no diffusion in the lower part of the domain, while the diffusion in the upper part of the domain scales with the vertical position $x_2$. This corresponds to the following choice of

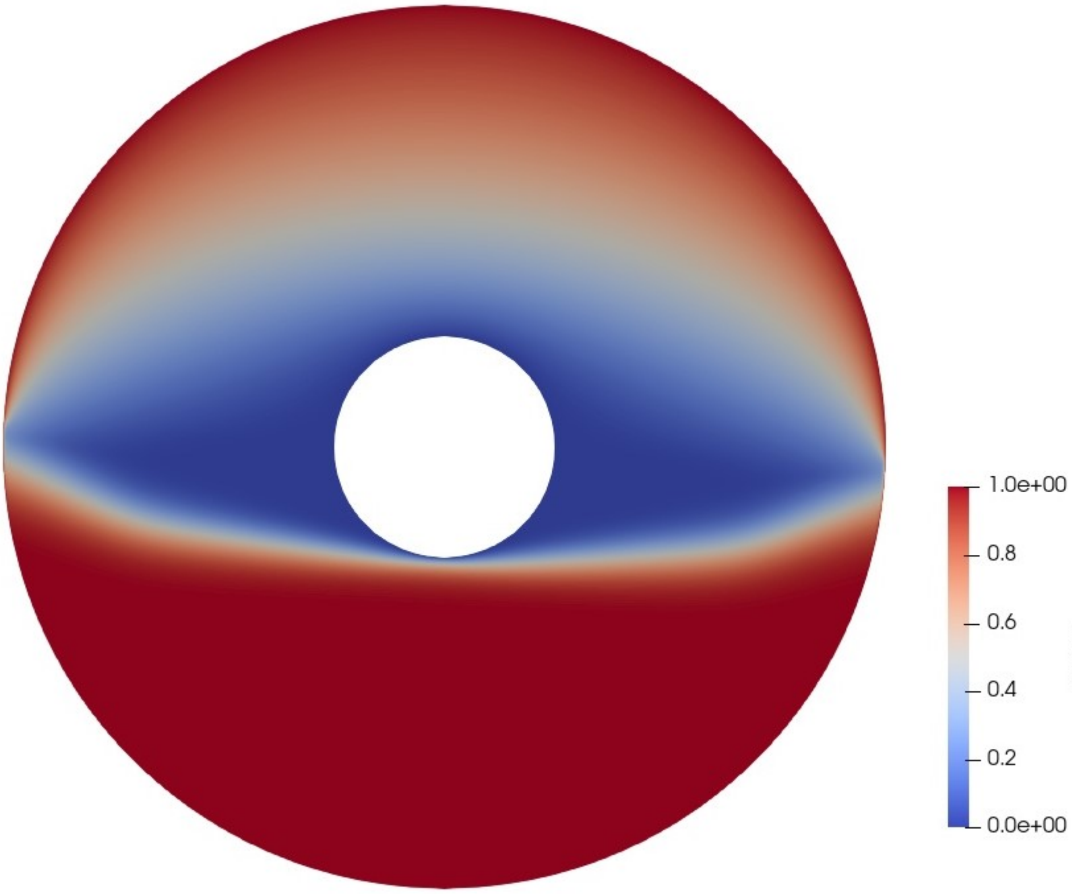


Figure 5.2: The value function at $t = 0$ for asymmetric velocities and the diffusion vanishing in the lower part of $\Omega$

the diffusion and advection coefficients:

$$
\begin{aligned}
b^{(\alpha,\beta)} &= \begin{pmatrix} 4_1 sin(\beta) - 0.5\sin(\alpha) \\ cos(\beta) - \cos(\alpha) \end{pmatrix} \\
a^{(\alpha,\beta)} &= \max\{x_2, 0\}.
\end{aligned}
$$

The numerical approximation of the value function solving the Isaacs problem (5.40) with this choice of coefficients can be found in Figure 5.2. Note the asymmetric nature of the graph, due to the different speeds of the pursuer and the evader in the horizontal direction as well as the effect of the stochastic component of the equation in the upper part of the domain.

# Chapter 6

# Discretization of the fully nonlinear equations in FEniCS

The aim of this section is twofold. Firstly, it is to share the experience gained during the years of using FEniCS library (more information available at [50]) and to facilitate the learning process of anyone wishing to use it in research. This goal is achieved is by briefly introducing the workflow of a FEniCS project as well as presenting a number of generic solutions to common problems which are not readily available through standard FEniCS API. Some of those can be found dispersed through the discussion boards and pieces of documentations while the others are personally written by the author and collected for the reader's convenience. The other goal is to share some of the implementation details of the numerical schemes presented in this work and to give some hints to anybody wishing to reproduce the results. We would like to indicate that the problems discussed in this work are posed in a non-divergence form which is not a setting FEniCS was created for. Therefore it is sometimes unavoidable to directly access and edit the values of the assembled matrices which is in general ill-advised if one wants to keep efficiency at mind. The usual solution in the standard case is to reformulate a problem in such a way that the assembled matrices can be used right away. Therefore author finds it instructive to show how can one proceed when such a reformulation is not an option.

## 6.1 Mesh creation and processing

### Mesh creation in Gmsh

Mesh creation using Gmsh is relatively easy. One needs to define a geometry using the geometric entites such as points and lines, define the planes or volumes inside the lower dimensional contours and finally define the mesh. For a more complicated use cases we refer to the Gmsh documentation [55]. The main difficulty in our case is that we require the mesh to be strictly acute and there is no way to enforce it directly. The simplest solution for 2D meshes is to use Frontal-Delaunay meshing algorithm as it usually results in a strictly acute mesh. However, it is not always the case, especially when one tries later to refine the mesh via element splitting. In some cases it may be necessary to manipulate mesh nodes coordinates by hand. Therefore it is important to include some kind of test within your code that makes sure that strict acuteness condition is satisfied.

## Conversion to xdmf format

By default dolfin-convert command provided by DOLFIN library converts meshes to xml format, which is inefficient both in terms of access time and space. More modern solution is to use xdmf format. One way to convert a msh file is to write a python script using meshio package. It is however worth noting that FEniCS expects meshes to be saved in a certain form that was typically provided by dolfin-convert. This is unfortunately not the case with the default meshio conversion. One needs to do some fine tuning depending on the specifics of your mesh and types of elements used in the Finite Element formulation. One way to avoid this headache, at a cost of computing time, is to simply first convert meshes from msh to xml format using DOLFIN and then from xml to xdmf format using meshio. Beneath we present a short bash script to achieve just that for all msh files in the current folder.

```bash
#!/usr/bin/env bash
ls -1 ${1}*.msh | xargs -n 1 bash -c 'dolfin-convert "$0" "${0%.*}.xml" && meshio-convert "${0%.*}.xml" "${0%.*}.xdmf"'
rm ${1}*.xml
```

## Usage in FEniCS

After the mesh is in the correct form both from a mathematical and computational point of view, its use in FEniCS is relatively easy. First we create a new instance of `Mesh` class, load the mesh into it and it is ready to use.

```python
mesh = Mesh()
with XDMFFile(mesh_filepath) as f:
    f.read(mesh)
```

Although it is not directly connected with meshes we would also like to discuss saving functions to xdmf file at this point. Let us say we are solving a parabolic, backwards in time problem, initialised by the interpolation of the final time data. This is one way to initialise the output file.

```
file = XDMFFile(save_directory)
file.parameters['rewrite_function_mesh'] = False
v = interpolate(exact_solution, V)
file.write(v, T)
```

Note the usage of `'rewrite_function_mesh'` parameter. This assumes that the underlying mesh does not change and hence there is no need to save the mesh data at every time step, thus saving time and disk space. In some cases, for example when using mesh adaptive methods one would set this parameter to true. One useful feature of xdmf files is that they allow to save multiple functions to a single output file. The example showing how to save the value function $v$ and the optimal control $\alpha$ at time $t$ is shown below. Note that `v` and `control` variables should be defined outside the algorithm loop and their values should be updated without assigning new objects to them. Otherwise xdmf file will not be saved correctly, since FEniCS API function differentiates between functions based on their ID set on creation, not the variable name.

```
file.write_checkpoint(
    v, 'value_func', t, XDMFFile.Encoding.HDF5, True)
control.vector()[:] = alpha
file.write_checkpoint(
    control, 'control', t, XDMFFile.Encoding.HDF5, True)
```

## 6.2 Useful recipes

The aim of this section is to provide the reader with simple but not immediately obvious solutions to some common problems one may encounter when dealing with the problems in a non-divergence form in FEniCS . Note that in all code snippets, the objects from FEniCS library are assumed to be imported either by * import (not recommended as it my lead to hard to resolve naming conflicts) or directly. For all the following examples we assume that the mesh has been loaded and function spaces are defined. We also have NumPy and SciPy packages imported. Namely, we assume that at the top of each script we have the following

```
import numpy as np
import scipy as sp
from dolfin import *
```

```
mesh = Mesh()
with XDMFFile(mesh_filepath) as f:
    f.read(mesh)
# we only consider Lagrange P1 elements
V = FunctionSpace(mesh, 'CG', 1)
w = TrialFunction(V)
u = TestFunction(V)
```

## Selecting the boundary nodes

Let us say we wish to obtain the indices of all the boundary nodes. This can be done using the following piece of code:

```
bc = DirichletBC(V, Constant(0), 'on_boundary')
boundary_node_indices = bc.get_boundary_values().keys()
```

This strategy could be used to verify whether you correctly marked the boundary nodes of your domain. Let us assume we have created a subregion of a boundary with a particularly difficult geometry which looks like this:

```
class CustomSubdomain(SubDomain):
    def inside(self, x, on_boundary):
        return complicated_logical_statement
```

You now wish to test whether you made a mistake in *complicated_logical_statement*. This could be done as follows:

```
#use your custom subdomain to mark the boundary edges of the mesh
boundary_markers = MeshFunction('size_t', mesh, 1)
boundary_markers.set_all(99)
CustomSubdomain().mark(boundary_markers, 1)
#set up Dirichlet BC on customain subdomain
bc = DirichletBC(V, Constant(0), boundary_markers, 1)
#Compute positions of nodes contained in subdomain
dof_indices = bc.get_boundary_values().keys()
dof_coordinates = mesh.coordinates()[dof_to_vertex_map(V)]
```

```
    for index in dof_indices:
        print(dof_coordinates[index])
```

The above code simply prints to the console the coordinates of the nodes that were marked as belonging to the custom subdomain, which should be enough to perform an ’eye test’. For a more robust testing you could compare those with a list of coordinates of boundary nodes taken directly from the mesh file. However, remember that in this case the coordinate values will only be correct up to some tolerance. We also remark that even though we are using the term ’boundary nodes’, FEniCS actually marks edges as belonging to the boundary, not single nodes. So if the boundary subdomain does not mark any boundary nodes even though it detects nodes lying in it, this may be due to the fact that there is only one such node or that those nodes are not connected by the mesh edges.

## Assembling lumped mass matrix

Let us say that we wish to perform the mass lumping of a mass matrix. Even though it is not possible to do it directly in the form language we can use the following trick.

```
mass_form = w * u * dx
mass_action_form = action(mass_form, Constant(1))
MM_terms = assemble(mass_action_form)
# if needed we can now perform additional operations on the terms of lumped mass matrix
MM = assemble(mass_form)
MM.zero()
MM.set_diagonal(MM_terms)
```

Note that the matrix *MM* has a sparsity pattern allowing us to perform algebraic operations with other matrices assembled on the same function space. This would not be possible if we started by creating a diagonal matrix.

## Instantiating DOLFIN matrices/vectors

In this recipe we present several methods of instantiating DOLFIN matrices and vectors. In the below code values of all three vectors will be exactly the same, given that *dim* is number of the nodes in the mesh.

```
zero_vector_size_dim = Vector(mesh.mpi_comm(), dim)
vector_fixed_dimension = assemble(u*dx)
vector_fixed_dimension.zero()
function = interpolate(Constant(0), V)
```

```
vector = function.vector()[:]
matrix1 = assemble(w*u*dx)
empty_vector = Vector(mpi_comm(), 0)
empty_vector = matrix1.init_vector(empty_vector, 1)
```

Note how in the above piece of code we instantiated matrix by assembling what is basically a mass matrix. This step is unavoidable because FEniCS does not provide a direct way of manipulating sparsity pattern of a matrix - one would have to construct a new PETSc matrix, using petsc4py. Note that instantiating matrix directly e.g. with `Matrix()` or `PETScMatrix()` returns an interface whose underlying petsc4py array is not assembled. Thus there is no way of populating it with correct values without actually using petsc4py directly. Luckily, sparsity pattern depends on the connectivity of the mesh and will generally be the same for every assembled matrix through the form language. Therefore you can assemble any matrix and simply make a shallow copy of it in one of the following ways.

```
matrix2 = matrix1.copy()
matrix3 = Matrix(matrix2)
```

The next step would be to actually set the values of the newly constructed matrix. A method of doing it is discussed in the next recipe.

## Reading and setting values of DOLFIN matrices

Let us say we want to access a specific value of the assembled DOLFIN matrix. For meshes with small number of vertices you can simply use `matrix.array()` which returns a dense copy of underlying sparse data. For larger matrices this approach should be avoided or is outright impossible as it would cause memory overflow. Instead, FEniCS offers access to sparse data through `matrix.getrow(i)` method which returns an object containing two NumPy arrays. The first one contains column indices of non-zero terms and the second one contains the raw data. In the below example this method is used to set all non-zero terms of a matrix in a given row to a given constant.

```
def set_const_rows(mat, node_indices, const):
    for row_ind in node_indices:
        row = mat.getrow(row_ind)
        mat.set([[const]*len(row[1])], [row_ind], row[0])
        mat.apply('insert')
```

If we instead wanted to iterate through all rows of a DOLFIN matrix or even multiple matrices at the same time consider the following generator.

```
def getrows(*args, ignore=None):
    '''Generates rows of input matrices one-by-one. Works only for matrices of
     the same dimension, supports generic dolfin and PETSc formats.'''
    if ignore is None:
        ignore = set()
    if all(map(lambda x: isinstance(x, Matrix), args)):
        def getter(A, i): return A.getrow(i)
        dim = args[0].size(0)
    elif all(map(lambda x: isinstance(x, PETScMatrix)
                 or isinstance(x, Mat), args)):
        def getter(A, i): return A.getRow(i)
        dim = args[0].getSize()[0]
    else:
        raise Exception("Unsupported or non-matching matrix format."
                        "All matrices should be in the same format."
                        "Supported formats: Matrix, PETScMatrix")

    for i in range(dim):
        if i in ignore:
            continue
        if len(args) == 1:
            yield (getter(args[0], i), i)
        else:
            yield (map(getter, args, [i]*len(args)), i)
```

Note that the above code allows to iterate both through generic DOLFIN matrices and their PETSc implementation. This could be easily extended to any other backends as well by implementing the additional alternative definitions of `getter` and `dim`. The implementation for a single matrix is different since unpacking tuples of length 1 is often counter-intuitive and getting rows of a single matrix is the most common use case. Be advised that iterating through DOLFIN matrices is in general slow and if you find yourself doing it frequently it may be worthwhile to consider using petsc4py or NumPy instead. This way you can access the underlying CSR data directly.

## L1 normalisation of vectors and matrices

This next recipe's usefulness depends on the specific numerical scheme being used but it gives an example of how `getrows()` function could be used.

```
l1_norm = assemble(u*dx)

def l1_normalise(operator, l1_norm):
    if isinstance(operator, Vector):
        operator[:] = [operator[i] / l1_norm[i]
                       if l1_norm[i] else 0.0 for i in range(operator.size())]
        return
    for row, row_ind in getrows(operator):
        vals = []
        for pos in range(len(row[0])):
            vals.append(row[1][pos] / l1_norm[row_ind]
                        if l1_norm[row_ind] else 0.0)
        operator.set([vals], [row_ind], row[0])
        operator.apply('insert')
```

## Matrix multiplication of DOLFIN matrices

This is a short but rather insightful recipe that shows how to obtain the product of two DOLFIN matrices. Similar approach can be taken in order to perform any matrix operation available in petsc4py.

```
def matmult(A, B):
    C = as_backend_type(A).mat().__mul__(as_backend_type(B).mat())
    return todolfin(C, B)

def todolfin(A, B):
    A_d = B.copy()  # A_d is dolfin matrix
    for Arow, row_num in getrows(A):
        A_d.set([Arow[1]], [row_num], Arow[0])
    A_d.apply('insert')
    return A_d
```

Although it may be tempting to return something like `Matrix(PETScMatrix(C))` from `matmult` function, this approach to author's knowledge is incredibly unstable and often results in segmentation fault errors. The presented solution of copying data over to an existing matrix row by row, although admittedly slower, results in a much more stable behaviour.

## Checking monotonicity of an operator

We now present two alternative versions of confirming the monotonicity of an explicit operator - one using PETSc matrices, the other one using SciPy matrices.

```python
def check_monotonicity_petsc(A, ignore=None):
    for row, row_ind in getrows(A, ignore):
        if np.any(row[1] > 0):
            raise Exception(f"Positive Term In row {row_ind}")

def check_monotonicity_scipy(A, ignore=None):
    if ignore is not None:
        check_indices = list(set(range(E.shape[0])) - ignore)
        check = E[check_indices]
    else:
        check = E
    if np.any(check.data > 0):
        raise Exception("Positive Term In Explicit Operator")
```

## Checking diagonal dominance of an operator

This recipe is very similar to the previous one but this time we are confirming the monotonicity properties of implicit matrices.

```python
def implicit_check_petsc(A, ignore):
    for row, row_ind in getrows(A):
        if row_ind in ignore:
            continue
        indices = row[0]
        data = row[1]
        diag_pos = np.where(indices == row_ind)[0][0]
        diag_val = data[diag_pos]
        for pos, x in enumerate(data):
            if diag_val * x > 0.0 and pos != diag_pos:
                raise Exception(f"Wrong sign in implicit matrix row {row}")
        if 2*diag_val <= sum(data):
            raise Exception(
                "Implicit matrix is not strictly diagonally dominant")

def check_implicit_scipy(A):
    nonzero = A.astype(bool).sum(axis=1)
```

```
        sumsigns = np.abs(A.sign().sum(axis=1))
        if np.any(sumsigns != np.abs(2 - nonzero)):
            row = np.where(sumsigns != np.abs(2 - nonzero))[0][0]
            raise Exception(
                f'Wrong sign in impicit matrix row: {row}'
            )

        D = np.abs(A.diagonal())
        D = D.reshape(-1, 1)
        S = np.sum(np.abs(A), axis=1) - D
        if not np.all(D > S):
            row = np.where(D <= S)[0][0]
            raise Exception(f'Matrix is not diagonally dominant - see row: {row}')
```

## Error calculation

This recipe calculates $L^\infty$, $L^2$ and $H^1$ distance between exact solution provided by the user and some DOLFIN function. Note that while $L^\infty$ error only considers values at the nodes, $L^2$ and $H^1$ errors are calculated by projecting provided function onto a higher dimensional space. Thus for a precise measurement, the exact function has to be defined in a space of dimension at least `degree_rise` higher than the provided solution.

```
def error_calc(v, v_e, mesh, mesh_name, save_dir='errors.json'):
    try:
        with open(save_dir, 'r') as f:
            errors = json.load(f)
    except FileNotFoundError:
        errors = {}
    new_errors = {}
    v_int = interpolate(v_e, FunctionSpace(mesh, 'CG', 1))
    new_errors['Linf'] = norm(v.vector() - v_int.vector(), 'linf')
    new_errors['L2'] = errornorm(v_e, v, mesh=mesh, degree_rise=3)
    new_errors['H1'] = errornorm(
        v_e, v, norm_type='H1', mesh=mesh, degree_rise=3)
    errors[mesh_name] = new_errors
    with open(save_dir, 'w') as f:
        json.dump(errors, f)
```

Note that this recipes allows to calculate the errors for multiple meshes within a single experiment, recording all of them in a json file.

## 6.3 Implementation details

In this section we discuss some of the implementation details of the numerical schemes discussed in Chapters 3 and 5.

### Calculating minimal timestep

Recall that our numerical method requires explicit operators $\mathsf{E}_i^{(\alpha,\beta)}$ to be monotone which is guaranteed by choosing a small enough timestep $h$. In general, we want to be able to set the number of timesteps ourselves but the code should correct it in case the chosen number of timesteps was too small. We would also like to be informed what is the minimal required number of timesteps for a given experiment for future reference. This is all accomplished by the following code we will discuss piece by piece.

Firstly, during the assembly of the explicit operators `E` we perform the following at each iteration.

```
diag_vec = Vector(mesh.mpi_comm(), dim)
E.get_diagonal(diag_vec)
h_ab = get_min_timestep(diag_vec, MM_terms,boundary_nodes_list)
h_min = min(h_ab, h_min)
```

The `get_min_timestep` function is implemented as follows:

```
def get_min_timestep(diag, MMdiag, ignore):
    timestep = float('inf')
    for i in range(diag.size()):
        if i in ignore or diag[i] <= 0:
            continue
        timestep = min(MMdiag[i]/diag[i], timestep)
    return timestep
```

After the loop finishes we can simply calculate the minimal number of timesteps with `minM = int(T/h_min) + 1` where $T$ denotes the final time. We also would like to have a way of calculating the number of timesteps ourselves. This is one possible implementation which simply reads the value from a json file.

```
M = get_number_of_timesteps(file_path, mesh_name)

def get_number_of_timesteps(file_path, mesh_name):
    try:
        with open(file_path) as f:
            minM = json.load(f)
            M = minM[mesh_name]
    except (FileNotFoundError, KeyError):
        print('Warning: timestep data not found, setting M to None')
        M = None
    return M
```

Another possible implementation of `get_number_of_timesteps` function would be to fix the ratio between the mesh size and the time step for the sake of convergence testing. Having calculated both minimal and custom number of timesteps we can now compare them and save to the json file for future reference.

```
if not M or M < minM:
    M = minM
    try:
        with open(filename, 'r') as f:
            minM_dict = json.load(f)
    except FileNotFoundError:
        minM_dict = {}
    minM_dict[mesh_name] = M
    with open(filename, 'w') as f:
        json.dump(minM_dict, f)
```

## Calculating the directional derivative at the boundary

Below we present one possible way to calculate and set the values of the directional derivatives at the boundary nodes as required by the scheme described in Chapter 3.

```
    coords = mesh.coordinates()[dof_to_vertex_map(V)]
    def set_directional_derivative(self, operator, region, nodes, control,
                                   time=None):
        for j in self.par.regions[region]:
            adv_x = self.par.robin_adv_x[j](control)
            adv_y = self.par.robin_adv_y[j](control)
            lin = self.par.robin_lin[j](control)
            if time:
                adv_x.t = time
                adv_y.t = time
                lin.t = time
            b = (interpolate(adv_x, self.V),
                 interpolate(adv_y, self.V))
            c = interpolate(lin, self.V)
            for n in nodes:
                # node coordinates
                x = coords[n]
                # evaluate advection at robin node
                b_x = np.array([b[0].vector()[n], b[1].vector()[n]])
                # denominator used to calculate directional derivative
                # denom = self.lamb*np.linalg.norm(b_x)
                if np.linalg.norm(b_x) > 1:
                    lamb = 0.01*self.mesh.hmin()/np.linalg.norm(b_x)
                else:
                    lamb = 0.01*self.mesh.hmin()
                # position of first node of the stencil
                x_prev = x - lamb * b_x
                # Find cell containing first node of stencil and get its
                # dof/vertex coordinates
                try:
                    cell_ind = self.mesh.bounding_box_tree(
                    ).compute_entity_collisions(Point(x_prev))[0]
                except IndexError:
                    raise Exception(
                        "Boundary advection outside tangential cone")
                cell_vertices = self.mesh.cells()[cell_ind]
                cell_dofs = vertex_to_dof_map(self.V)[cell_vertices]
                cell_coords = self.mesh.coordinates()[cell_vertices]
                # calculate weight of each vertex in the cell (using
                #  barycentric coordinates)
                A = np.vstack((cell_coords.T, np.ones(3)))
                rhs = np.append(x_prev, np.ones(1))
                weights = np.linalg.solve(A, rhs)
                dof_to_weight = dict(zip(cell_dofs, weights))
                # calculate directional derivative at each node using
                # weights to interpolate value of numerical solution at
                # x_prev
                row = operator.getrow(n)
                indices = row[0]
                data = row[1]
                for dof in cell_dofs:
                    pos = np.where(indices == dof)[0][0]
                    if dof != n:
                        data[pos] = - dof_to_weight[dof] / lamb
                    else:
                        c_n = c.vector()[dof]
                        # make sure reaction term is positive adding artificial
                        # constant if necessary
                        if not c_n and n in self.robin_nodes:
                            c_n = min(lamb, 1E-4)
                        data[pos] = (1-dof_to_weight[dof]) / lamb + c_n
                # this part is to avoid rounding errors while subtracting and
                # dividing for small lambdas
                sum_data = sum(data)
                if sum_data < 0 and sum_data > -1E-6:
                    pos = np.where(indices == n)[0][0]
                    data[pos] += 1E-6
                operator.set([data], [n], indices)
                operator.apply('insert')
```

## Howard's algorithm

This piece of code shows an implementation of a single step of Howard's algorithm used in numerical approximation of an Isaacs problem.

```
solver = PETScKrylovSolver('gmres', 'sor')
def Howard_outer(v, alpha):
     # v coresponds to v^{k+1}
    alpha = [0]*dim
    ell = 0  # iteration counter
    while ell < howmaxit:
        ell += 1
        how, alpha = Howard_inner(alpha, beta)

    # Create list of vectors (I*v^k+E*v^{k+1} -F) under different controls
    sphow = np.array(how[:])
    spv = np.array(v.vector()[:])
    Ev = [[Exp.dot(spv) for Exp in Exp_b_list]
          for Exp_b_list in Elist]

    multlist = np.empty(
        [csize_beta, csize_alpha, dim])
    for ind_b in range(csize_beta):
        for ind_a in range(csize_alpha):
            multlist[ind_b][ind_a] = \
                (spIlist[ind_b][ind_a].dot(sphow) +
                     Ev[ind_b][ind_a] -
                     Flist[ind_b][ind_a])

        # Loop over vectors of values of (I*v^k+E*v^{k+1} -F) at each node and
        # record control which optimises each of them
        next_ctr = np.argmin(np.max(multlist, axis=1), axis=0)
        return (how, next_ctr)

    def Howard_inner(self, alpha, beta):
        # v coresponds to v^{k+1}, save it to numpy array
        spv = np.array(v.vector()[:])

        Ev = {b: [Ea.dot(spv) for Ea in Elist[b]] for b in set(beta)}
        # construct RHS under input control alfa
        rhs = self.v.vector().copy()
        rhs[:] = [Flist[beta[i]][a][i]
                  - Ev[beta[i]][a][i]
                  for i, a in enumerate(alpha)]
        # initialise vector with correct dimension to store solution
        how = self.v.vector().copy()

        # initalise matrix with a suitable sparsity pattern
        lhs = Ilist[0][0].copy()
        # construct implicit matrix under control (alfa, beta)
        for i, a in enumerate(alpha):
            lhs.set([Ilist[beta[i]][a].getrow(i)[1]],
                    [i], Ilist[beta[i]][a].getrow(i)[0])
        lhs.apply('insert')

        # solve linear problem to get next iterate of u
        for bc in dirichlet_bcs:
            bc.apply(lhs, rhs)
        self.solver.solve(lhs, how, rhs)

        # Create list of vectors (I*v^k+E*v^{k+1} -F) under different controls
        spw = np.array(how[:])
        multlist = np.empty([csize_alpha, dim])
        Iw = {b: [Ia.dot(spw) for Ia in spIlist[b]]
              for b in set(beta)}

        for ind_a, _ in enumerate(ctrset_alpha):
            for j in range(dim):
                multlist[ind_a][j] = Iw[beta[j]][ind_a][j] + Ev[beta[j]][ind_a][j] -
                Flist[beta[j]][ind_a][j]

        # Loop over vectors of values of (I*v^k+E*v^{k+1} -F) at each node and
        # record control which optimizes each of them
        next_ctr = [np.argmax(vector) for vector in zip(*multlist)]

        return (how, next_ctr)
```

Outside of this function one could perform convergence check and see if the distance between the current and the previous iteration of `how` is smaller than the prescribed tolerance. If not, updated value of `alpha` can be used in the next step of Howard's algorithm. From author's experience, this algorithm converges quickly to the required solution and it is often enough to hard code the number of iterations (determined by testing, usually around 5) and forego the tolerance check as done inside `Howard_inner` function.

## 6.4 Comparison of the linear algebra libraries

Having shared the actual FEniCS code we would like to briefly justify some of the implementation choices. The discussion will be rather high level and experience-based but may offer an insight to those wishing to improve the performance of their code or at least provide the idea how to start benchmarking it. The main theme of this section is a comparison of the performance of petsc4py and SciPy sparse matrices when performing different tasks. We use FEniCS as a tool to process meshes and assemble the discretised operators on which experiments are performed. This will hopefully give a rough idea of how one can pick and choose between different backends in order to speed the code up.

### Transformation from DOLFIN matrix to SciPy matrix and vice versa

Transforming DOLFIN matrices to both NumPy / SciPy and petsc4py formats is relatively easy and fast. it can be done in the following way.

```
def toscipy(A):
    mat = as_backend_type(A).mat()
    csr = csr_matrix(mat.getValuesCSR()[::-1], shape=mat.size)
    return csr
```

Note that in order to instantiate SciPy matrix we are first creating petsc4py matrix. However, transforming back to DOLFIN matrix may be costly. For that reason it may be worthwhile to keep copies of both formats at the same time and use different matrices for different purposes. This also comes at a memory cost and

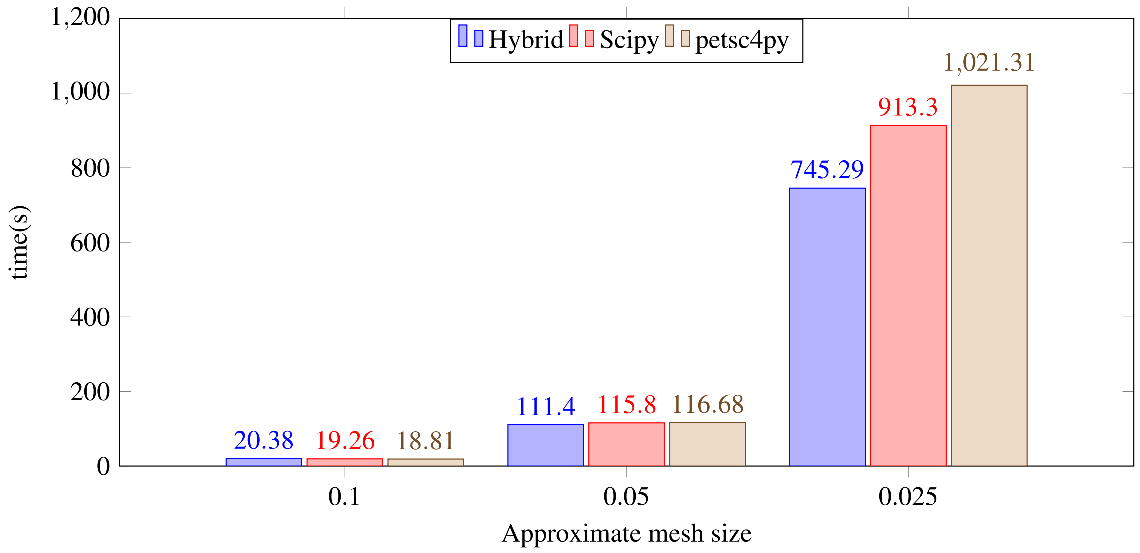


Figure 6.1: A simple benchmark measuring runtime. Implementations in different backends were all tested against the same PDE

needs to be done with caution. Any potential improvements should be confirmed by at least rudimentary testing.

## Performance testing

Each of the formats was created with a specific purpose in mind and therefore they should be used accordingly. The multidimensional NumPy arrays are great for quickly accessing operators assembled under different controls. Also the fact that broadcasting works even for sparse SciPy matrices makes it a good format for checking if assembled matrices satisfy the structural conditions, e.g. if they are monotone. On the other hand, using petsc4py we have a guarantee that the sparsity pattern of matrices will always remain unchanged which is crucial when constructing operators consisting row-wise from other previously assembled operators. With petsc4py we also have a fine-grained control over the linear solver which affects the performance greatly.

We now consider a simple benchmark (see Figure 6.1) performed by author at the early stage of code development. It compares three different implementations of the same program. The comparison here is made between implementation in petsc4py, NumPy /SciPy and a combination of the two approaches. Approximately similar amount of time was spent on each implementation so the learning curve of each library may also be the factor in the final result.

The main take away here is not that petsc4py is the slowest. In our use case where we access,edit and construct operators constantly, broadcasting and multi-dimensional arrays offered by NumPy /SciPy give it an edge even if the matrix operations and linear solvers are not as optimised. The main conclusion is that the hybrid approach where we performed some actions using SciPy matrices and others using petsc4py matrices may be the best. This will differ based on the problem at hand but if performance is a concern then code should always be tested. Beyond a simple benchmarking as presented above author suggests using `cProfile` module on different implementations of the same program in order to get a breakdown of the time spent in different parts of the code. One then can pick parts of the implementations which are the fastest and put them together in a single piece of code. However, it is worth remembering that overhead caused by transitioning between the data types may outweigh any performance boost. Lastly, author recommends fine-tuning of functionalities using `timeit` module. We present several examples below.

First test shows how important it is to regularly check the performance of new versions of code. After experiencing a tremendous slowdown of the program after making seemingly no changes to a function assembling right hand side of the linear system, an effort was made to find the cause. We now present the minimal working example. In Jupyter Notebook let us define:

```python
from dolfin import (assemble, dx, UnitSquareMesh, FunctionSpace, TestFunction,
interpolate, Constant)
import numpy as np

mesh = UnitSquareMesh(1000, 1000)
V = FunctionSpace(mesh, 'CG', 1)
u = TestFunction(V)
rhs = interpolate(Constant(0),V)
```

Let us now consider the output of the two following cells.

```python
%%timeit
np.array(assemble(rhs*u*dx))
```

7.98 *s* ± 60.7 *ms per loop (mean* ± *std. dev. of* 7 *runs,* 1 *loop each)*

```python
%%timeit
np.array(assemble(rhs*u*dx)[:])
```

371 *ms* ± 13.8 *ms per loop (mean* ± *std. dev. of* 7 *runs,* 1 *loop each)*

Quite surprisingly constructing NumPy array from a slice of a DOLFIN vector seems to be an order of magnitude faster compared to using a raw DOLFIN vector. It has to do with the fact that the slice operator applied to a DOLFIN vector actually returns the underlying NumPy array and hence data copying is much more efficient.

Let us now measure the performance of some of the recipes which were implemented both in SciPy and petsc4py. Let us assume that we have defined functions `getrows` , `toscipy` , `check_monotonicity_scipy` and `check_monotonicity_petsc` as previously. In Jupyter Notebook we define

```
from dolfin import (assemble, dx, UnitSquareMesh, FunctionSpace, TestFunction, TrialFunction, Matrix, as_backend_type)
from scipy.sparse import csr_matrix
import numpy as np

mesh = UnitSquareMesh(1000, 1000)
V = FunctionSpace(mesh, 'CG', 1)
u = TestFunction(V)
w = TrialFunction(V)
matrix = assemble(u*w*dx)
matrix_size = matrix.size(0)
```

and then we consider the following cells and their evaluations.

```
%%timeit
check_monotonicity_petsc(matrix)
```

```
7.56 s ± 105 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
%%timeit
check_monotonicity_scipy(toscipy(matrix))
```

```
32.8 ms ± 444 µs per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

```
%%timeit
check_monotonicity_petsc(matrix, ignore=set(range(matrix.size(0)//2)))
```

```
3.83 s ± 134 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
%%timeit
check_monotonicity_scipy(toscipy(matrix), ignore=set(range(matrix.size(0)//2)))
```

```
336 ms ± 14 ms per loop (mean ± std. dev. of 7 runs, 1 loop each)
```

```
%%timeit
check_monotonicity_petsc(matrix, ignore=set(range(matrix.size(0))))
```

```
114 ms ± 3.8 ms per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

```
%%timeit
check_monotonicity_scipy(toscipy(matrix), ignore=set(range(matrix.size(0))))
```

294 *ms* ± 14.8 *ms per loop (mean* ± *std. dev. of* 7 *runs,* 1 *loop each)*

As expected direct iteration through the rows of petsc4py matrix is inefficient compared to direct access to underlying data paired with broadcasting available to SciPy matrix. We finish this section with a result which is often forgotten by beginners, namely that standard loops should be avoided when dealing with NumPy /SciPy matrices. We now consider an alternative implementation of `check_monotonicity_scipy` function

```
def check_monotonicity_scipy2(E, ignore=None):
    i = 0
    if ignore is None:
        ignore = set()
    for i in range(E.shape[0]):
        if i in ignore:
            continue
        row = E.getrow(i)
        if np.any(row.data > 0):
            i += 1
```

and run the following cell.

```
%%timeit
check_monotonicity_scipy2(toscipy(matrix))
```

1*min* 3*s* ± 2.08 *s per loop (mean* ± *std. dev. of* 7 *runs,* 1 *loop each)*

As we can see the performance is significantly worse even when compared to petsc4py matrix. This should be an emphatic argument against using loops when dealing with NumPy /SciPy arrays.

This concludes the section about benchmarking. If performance is still an issue, author's recommendation is to directly use low-level C or C++ code. However, this is beyond the scope of this chapter whose sole focus is FEniCS Python interface.

# Chapter 7

# Conclusions

The main goal of this dissertation was to extend the formulation and design of P1 Finite Element Methods to two novel settings, namely to approximate the unique viscosity solution of the second order Hamilton-Jacobi-Bellman equations with fully nonlinear mixed boundary conditions and the second order Isaacs equation with the homogeneous Dirichlet boundary conditions. We now briefly summarise the contributions of each of the chapters, not including the introductory Chapter 2.

In Chapter 3 we provided the first Finite Element Method which can approximate the viscosity solution of the Hamilton-Jacobi-Bellman equation with fully nonlinear mixed boundary conditions in considerable generality on possibly nonconvex, non-smooth domains. We remind the reader that for the problems of this type a challenge is posed by the discretisation of the first order directional derivatives in (3.1b) and (3.1c). On the one hand establishing monotonicity with an artificial diffusion approximating the Laplace-Beltrami operator of $\partial\Omega$ would not be sufficient because of the normal component. On the other hand, an artificial diffusion approximating the Laplace operator of $\Omega$ would not vanish under refinement due to different scaling of boundary and domain terms, thus leading to an inconsistent method. This was achieved by discretising the boundary differential operators via lower Dini derivative which ensured that the scheme stayed monotone and consistent. As a result we obtained a method which works on the unstructured meshes and for possibly degenerate diffusions. This allowed us to treat nonlinear problems arising from stochastic optimal control like for example the Skorokhod problem on nonconvex domain.

In Chapter 4 we presented a novel model of the uncertain market price of volatility risk in extension of Heston's equation, which takes the form of a Hamilton-Jacobi-Bellman equation with mixed boundary conditions. This together with findings of Chapter 3 provided a viable method of option value estimation in the setting of the uncertain market price of volatility risk. Additionally, thanks to this formulation one can easily assume any of the parameters to be uncertain and take some confidence interval around the estimated value of parameter to be the control set of optimisation problem. One can then proceed and calculate the value function of the worst case scenario, which may be of interest to the financial community given that the parameter has a nonlinear impact on the option value or its derivatives. In this chapter we chose the market price of volatility risk as the uncertain parameter due to relative shortage of information about it in the existing literature. We then conducted a case study on butterfly option and confirmed that the option value scales linearly with the market price of risk and the magnitude of the impact is non-trivial. Additionally, we found some evidence that the market price of volatility risk may have non-trivial nonlinear impact on the financial derivatives under certain conditions.

In Chapter 5 we provided a novel Finite Element Method for solving the second order Isaacs equation with degenerate anisotropic diffusion on possibly nonconvex domains. We extended the results of [68] to obtain a discretisation of linear operators projecting the non-homogeneous Dirichlet data and satisfying the monotonicity conditions required for Howard's algorithm to converge to the unique solution. Due to the nonconvexity on inf sup operators we no longer could use that solutions of linear evolution problems associated to each control are an upper bound for the numerical solution. Despite that, we proved the stability of the scheme, using that at least one such control exists and making the necessary adaptation to accommodate the non-homogeneous boundary conditions. The convergence to the boundary conditions remained a difficult problem but we provided the reader with a framework that assures convergence of the envelopes of the numerical solution to the boundary data. The result is subject to the existence of a family of barrier functions on the boundary. As a consequence a class of comparison principles was introduced which allows to flexibly combine the pointwise and the viscosity boundary conditions. The convergence rates were investigated in a numerical experiment. We also studied a numerical approximation of a value function of an asymmetric, stochastic tag-chase game with degenerate diffusion. To author's knowledge this is one of the few currently formulated numerical methods proven to work for a second order Isaacs problem and the first one which can simultaneously deal with unstructured meshes, nonconvex domain and degenerate ellipticity.

In Chapter 6 we provided some of the implementation details of the presented methods. The focus was on providing the advice for researchers interested in implementing numerical methods for problems in the non-divergence form in FEniCS . We discussed a general workflow in a standard FEniCS project, with the special attention paid to creating, transforming and loading the meshes. We also showed pieces of code used by author to solve specific, reoccurring problems, especially having to do with manually manipulating the discretised operators. We finished the chapter with a quick discussion about code performance, how it is influenced by the choice of linear algebra backend and how preliminary benchmarking may be performed. As far as author is aware, most of the written material regarding FEniCS is rather outdated and none of it discusses problems in non-divergence form. Thus hopefully this chapter will be a valuable contribution to some of the readers.

# Bibliography

[1] Y. Achdou and M. Falcone, *A semi-Lagrangian scheme for mean curvature motion with nonlinear Neumann conditions*, Interfaces and Free Boundaries, 14 (2012), pp. 455–485, https://doi.org/10.4171/IFB/288.

[2] M. Avellaneda, A. Levy, and A. Parás, *Pricing and hedging derivative securities in markets with uncertain volatilities*, Applied Mathematical Finance, 2 (1995), pp. 73–88, https://doi.org/10.1080/13504869500000005.

[3] G. Bakshi and N. Kapadia, *Delta-hedged gains and the negative volatility risk premium*, Review of Financial Studies, 16 (2003), pp. 527–566, https://doi.org/10.2139/ssrn.267106.

[4] L. V. Ballestra and G. Pacelli, *Pricing European and American options with two stochastic factors: A highly efficient radial basis function approach*, Journal of Economic Dynamics and Control, 37 (2013), pp. 1142–1167, https://doi.org/10.1016/j.jedc.2013.01.013.

[5] G. Barles and E. R. Jakobsen, *On the convergence rate of approximation schemes for Hamilton-Jacobi-Bellman equations*, ESAIM: Mathematical Modelling and Numerical Analysis, 36 (2002), pp. 33–54, https://doi.org/10.1051/m2an:2002002.

[6] G. Barles and P. E. Souganidis, *Convergence of approximation schemes for fully nonlinear second order equations*, Asymptotic Analysis, 4 (1991), pp. 271–283, https://doi.org/10.3233/ASY-1991-4305.

[7] E. N. Barron, L. C. Evans, and R. Jensen, *The infinity Laplacian, Aronsson's equation and their generalizations*, Transactions of the American Mathematical Society, 360 (2008), pp. 77–101, https://doi.org/10.1090/S0002-9947-07-04338-3.

[8] R. W. Beard and T. W. McLain, *Successive Galerkin approximation algorithms for nonlinear optimal and robust control*, International Journal of Control, 71 (1998), pp. 717–743, https://doi.org/10.1080/002071798221542.

[9] C. Beck, S. Becker, P. Cheridito, A. Jentzen, and A. Neufeld, *Deep splitting method for parabolic PDEs*, arXiv preprint arXiv:1907.03452, (2019).

[10] C. BECK, M. HUTZENTHALER, A. JENTZEN, AND B. KUCKUCK, *An overview on deep learning-based approximation methods for partial differential equations*, arXiv preprint arXiv:2012.12348, (2020).

[11] J.-D. BENAMOU AND Y. BRENIER, *Weak existence for the semigeostrophic equations formulated as a coupled Monge–Ampère/transport problem*, SIAM Journal on Applied Mathematics, 58 (1998), `https://doi.org/10.1137/S0036139995294111`.

[12] J.-D. BENAMOU, B. D. FROESE, AND A. M. OBERMAN, *A viscosity solution approach to the Monge-Ampère formulation of the Optimal Transportation Problem*, arXiv, (2012), `http://arxiv.org/abs/1208.4873v2`, `https://arxiv.org/abs/1208.4873v2`.

[13] A. BERMAN AND R. J. PLEMMONS, *M-matrices*, in Nonnegative Matrices in the Mathematical Sciences, Academic Press, 1979, ch. 6, pp. 132–164, `https://doi.org/10.1016/B978-0-12-092250-5.50013-8`.

[14] O. BOKANOWSKI, S. MAROSO, AND H. ZIDANI, *Some convergence results for Howard's algorithm*, SIAM Journal on Numerical Analysis, 47 (2009), pp. 3001–3026.

[15] S. BRENNER AND R. SCOTT, *The Mathematical Theory of Finite Element Methods*, vol. 15 of Texts in Applied Mathematics, Springer-Verlag, New York, 2008, `https://doi.org/10.1007/978-0-387-75934-0`.

[16] E. BURMAN AND A. ERN, *Nonlinear diffusion and discrete maximum principle for stabilized Galerkin approximations of the convection–diffusion-reaction equation*, Computer Methods in Applied Mechanics and Engineering, 191 (2002), pp. 3833–3855.

[17] K. BÖHMER, *On Finite Element Methods for fully nonlinear elliptic equations of second order*, SIAM Journal on Numerical Analysis, 46 (2008), pp. 1212—–1249, `https://doi.org/10.1137/040621740`.

[18] L. CAFFARELLI AND P. SOUGANIDIS, *A rate of convergence for monotone Finite Difference approximations to fully nonlinear, uniformly elliptic PDEs*, Communications on Pure and Applied Mathematics, 61 (2008), pp. 1 – 17, `https://doi.org/10.1002/cpa.20208`.

[19] F. CARDIN AND O. BERNARDI, *Minimax and viscosity solutions of Hamilton-Jacobi equations in the convex case*, Communications on Pure and Applied Analysis - COMMUN PURE APPL ANAL, 5 (2006), pp. 793–812, `https://doi.org/10.3934/cpaa.2006.5.793`.

[20] P. CHAN AND R. SIRCAR, *Bertrand and Cournot Mean Field Games*, Applied Mathematics & Optimization, 71 (2015), pp. 533—569, `https://doi.org/10.1007/s00245-014-9269-x`.

[21] M. Chaperon, *Lois de conservation et géométrie symplectique*, Comptes rendus de l'Académie des sciences. Série 1, Mathématique, 312 (1991), pp. 345–348.

[22] N. Clarke and K. Parrott, *Multigrid for American option pricing with stochastic volatility*, Applied Mathematical Finance, 6 (1999), pp. 177–195, https://doi.org/10.1080/135048699334528.

[23] S. N. Cohen and M. Tegnér, *European option pricing with stochastic volatility models under parameter uncertainty*, in Frontiers in Stochastic Analysis–BSDEs, SPDEs and their Applications, vol. 289 of Springer Proceedings in Mathematics & Statistics, Cham, 2019, Springer, pp. 123–167, https://doi.org/10.1007/978-3-030-22285-7_5.

[24] M. G. Crandall, L. C. Evans, and P. L. Lions, *Some properties of viscosity solutions of Hamilton-Jacobi equations*, Transactions of the American Mathematical Society, 282 (1984), pp. 487–502, https://doi.org/10.2307/1999247.

[25] 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 (1992), pp. 1–67, https://doi.org/10.1090/S0273-0979-1992-00266-5.

[26] M. G. Crandall and P.-L. Lions, *Viscosity solutions of Hamilton-Jacobi equations*, Transactions of the American Mathematical Society, 277 (1983), pp. 1–42, https://doi.org/10.2307/1999343.

[27] B. Dacorogna and P. Marcellini, *Implicit Partial Differential Equations*, Progress in Nonlinear Differential Equations and Their Applications, Birkhäuser, Boston, MA, 1999, https://doi.org/10.1007/978-1-4612-1562-2.

[28] J. Darbon, G. P. Langlois, and T. Meng, *Overcoming the curse of dimensionality for some Hamilton-Jacobi partial differential equations via neural network architectures*, Research in the Mathematical Sciences, 7 (2020), pp. 1–50, https://doi.org/https://doi.org/10.1007/s40687-020-00215-6.

[29] J. Darbon and T. Meng, *On some neural network architectures that can represent viscosity solutions of certain high dimensional Hamilton-Jacobi partial differential equations*, Journal of Computational Physics, 425 (2021), p. 109907, https://doi.org/https://doi.org/10.1016/j.jcp.2020.109907.

[30] J. Darbon and S. Osher, *Algorithms for overcoming the curse of dimensionality for certain Hamilton-Jacobi equations arising in control theory and elsewhere*, Research in the Mathematical Sciences, 3 (2016), pp. 1–26, https://doi.org/10.1186/s40687-016-0068-7.

[31] K. Debrabant and E. R. Jakobsen, *Semi-Lagrangian schemes for linear and fully non-linear diffusion equations*, Mathematics of Computation, 82 (2013), pp. 1433–1462.

[32] L. del Re and G. Steinmaurer, *Computation of minimum achievable fuel consumption for serial hybrids*, SAE Transactions, 108 (1999), pp. 3263–3270, https://doi.org/10.4271/1999-01-2945.

[33] A. Demlow, D. Leykekhman, A. H. Schatz, and L. B. Wahlbin, *Best approximation property in the $W^1_\infty$ norm for Finite Element Methods on graded meshes*, Mathematics of Computation, 81 (2012), pp. 743–764, https://doi.org/10.1090/S0025-5718-2011-02546-9.

[34] J. S. Doran, *On the market price of volatility risk*, PhD thesis, The University of Texas at Austin, 2004.

[35] J. S. Doran, *The influence of tracking error on volatility risk premium estimation*, Journal of Risk, 9 (2007), pp. 1–36, https://doi.org/10.21314/JOR.2007.149.

[36] J. S. Doran and E. I. Ronn, *Computing the market price of volatility risk in the energy commodity markets*, Journal of Banking & Finance, 32 (2008), pp. 2541–2552, https://doi.org/10.1016/j.jbankfin.2008.04.003.

[37] J. Duarte and C. S. Jones, *The price of market volatility risk*, AFA 2009 San Francisco Meetings Paper, (2007), https://doi.org/10.2139/ssrn.1106960.

[38] B. Düring and J. Miles, *High-order ADI scheme for option pricing in stochastic volatility models*, Journal of Computational and Applied Mathematics, 316 (2017), pp. 109–121, https://doi.org/10.1016/j.cam.2016.09.040.

[39] W. E, J. Han, and A. Jentzen, *Algorithms for solving high dimensional PDEs: From nonlinear Monte Carlo to machine learning.*, arXiv preprint arXiv:2008.13333, (2020).

[40] A. Ern and J.-L. Guermond, *Theory and Practice of Finite Elements*, vol. 159 of Applied Mathematical Sciences, Springer-Verlag, New York, 2004, https://doi.org/10.1007/978-1-4757-4355-5.

[41] L. C. Evans and P. Souganidis, *Differential games and representation formulas for solutions of Hamilton-Jacobi-Isaacs equations*, Indiana University Mathematics Journal, 33 (1984), pp. 773–797.

[42] M. Falcone, *Numerical methods for differential games based on Partial Differential Equations*, International Game Theory Review, 8 (2006), pp. 231–272, https://doi.org/10.1142/S0219198906000886.

[43] M. Falcone and D. Kalise, *A high-order semi-Lagrangian/Finite Volume scheme for Hamilton-Jacobi-Isaacs equations*, in System Modeling and Optimization. CSMO 2013. IFIP Advances in Information and Communication Technology, C. Pötzsche, C. Heuberger, B. Kaltenbacher, and F. Rendl, eds., vol. 443, Springer, Berlin, Heidelberg, 2014, pp. 105–117, https://doi.org/10.1007/978-3-662-45504-3_10.

[44] F. Fang and C. W. Oosterlee, *A Fourier-based valuation method for Bermudan and barrier options under Heston's model*, SIAM Journal on Financial Mathematics, 2 (2011), pp. 439—463, https://doi.org/10.1137/100794158.

[45] X. Feng, R. Glowinski, and M. Neilan, *Recent developments in numerical methods for fully nonlinear second order Partial Differential Equations*, SIAM Rev., 55 (2013), pp. 205–267, https://doi.org/10.1137/110825960.

[46] X. Feng and M. Jensen, *Convergent semi-Lagrangian methods for the Monge-Ampère equation on unstructured grids*, SIAM J. Numer. Anal., 55 (2017), pp. 691–712, https://doi.org/10.1137/16M1061709.

[47] X. Feng and T. Lewis, *A narrow-stencil Finite Difference Method for approximating viscosity solutions of fully nonlinear elliptic partial differential equations with applications to Hamilton-Jacobi-Bellman equations*, 2019, https://arxiv.org/abs/1907.10204.

[48] X. Feng and M. Neilan, *The Vanishing Moment Method for fully nonlinear second order partial differential equations: Formulation, theory, and numerical analysis*, 2011, https://arxiv.org/abs/1109.1183.

[49] X. Feng, M. Neilan, and S. Schnake, *Interior penalty Discontinuous Galerkin methods for second order linear non-divergence form elliptic PDEs*, Journal of Scientific Computing, 74 (2018), pp. 1651—-1676, https://doi.org/10.1007/s10915-017-0519-3.

[50] FEniCS Community, *FEniCS project*, https://fenicsproject.org (accessed 20-12-2020).

[51] H. C. Ferreira, P. H. Rocha, and R. M. Sales, *On the convergence of successive Galerkin approximation for nonlinear output feedback $H_\infty$ control*, Nonlinear Dynamics, 60 (2010), pp. 651–660, https://doi.org/10.1007/s11071-009-9622-9.

[52] W. H. Fleming and P. Souganidis, *On the existence of value functions of two-player, zero-sum stochastic differential games*, Indiana University Mathematics Journal, 38 (1989), pp. 293–314.

[53] P. A. Forsyth, K. R. Vetzal, and R. Zvan, *Penalty methods for American options with stochastic volatility*, Journal of Computational and Applied Mathematics, 91 (1998), pp. 199–218, https://doi.org/10.1016/S0377-0427(98)00037-5.

[54] D. GALLISTL, *Numerical approximation of planar oblique derivative problems in nondivergence form*, Math. Comp., 88 (2019), pp. 1091–1119, https://doi.org/10.1090/mcom/3371.

[55] C. GEUZAINE AND J.-F. REMACLE, *Gmsh reference manual*, https://gmsh.info/doc/texinfo/gmsh.html (accessed 20-12-2020).

[56] D. GILBARG AND N. S. TRUDINGER, *Elliptic Partial Differential Equations of Second Order*, Springer, Berlin, 2001.

[57] J. HANA, A. JENTZEN, AND W. E, *Solving high-dimensional partial differential equations using deep learning*, Proceedings of the National Academy of Sciences, 115 (2018), pp. 8505–8510, https://doi.org/10.1073/pnas.1718942115.

[58] S. L. HESTON, *A closed-form solution for options with stochastic volatility with application to bond and currency options*, The Review of Financial Studies, 6 (1993), pp. 327–343, https://doi.org/10.1093/RFS/6.2.327.

[59] N. HILBER, O. REICHMANN, C. SCHWAB, AND C. WINTER, *Computational Methods for Quantitative Finance*, Springer-Verlag, Berlin Heidelberg, 2013, https://doi.org/10.1007/978-3-642-35401-4.

[60] Ø. HJELLE AND S. A. PETERSEN, *A Hamilton–Jacobi framework for modeling folds in structural geology*, Mathematical Geosciences, 43 (2011), https://doi.org/10.1007/s11004-011-9357-2.

[61] J. HOZMAN AND T. TICHÝ, *A discontinuous Galerkin method for numerical pricing of European options under Heston stochastic volatility*, in AIP Conference Proceedings, vol. 1789, 2016, https://doi.org/10.1063/1.4968449.

[62] S. IKONEN AND J. TOIVANEN, *Efficient numerical methods for pricing American options under stochastic volatility*, Numerical Methods for Partial Differential Equations, 24 (2007), pp. 104–126, https://doi.org/10.1002/num.20239.

[63] S. IKONEN AND J. TOIVANEN, *Effcient numerical methods for pricing American options under stochastic volatility*, Numerical Methods for Partial Differential Equations, 24 (2008), pp. 104–126.

[64] S. IKONEN AND J. TOIVANEN, *Operator splitting methods for pricing American options under stochastic volatility*, Numerische Mathematik, 113 (2009), pp. 299–324, https://doi.org/10.1002/num.20239.

[65] K. IN 'T HOUT AND S. FOULON, *ADI Finite Difference schemes for option pricing in the Heston model with correlation*, International Journal of Numerical Analysis and Modeling, 7 (2010), pp. 303–320.

[66] K. Ito, C. Reisinger, and Y. Zhang, *A neural network-based policy iteration algorithm with global $H^2$-superlinear convergence for stochastic games on domains*, Foundations of Computational Mathematics, 21 (2021), pp. 331–374, `https://doi.org/https://doi.org/10.1007/s10208-020-09460-1`.

[67] M. Jensen, *$L^2(H^1\gamma)$ Finite Element convergence for degenerate isotropic Hamilton–Jacobi–Bellman equations*, IMA Journal of Numerical Analysis, 37 (2017), pp. 1300–1316, `https://doi.org/10.1093/imanum/drw055`.

[68] M. Jensen and I. Smears, *On the convergence of Finite Element Methods for Hamilton-Jacobi-Bellman equations*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 137–162, `https://doi.org/10.1137/110856198`.

[69] D. Kalise, A. Kröne, and K. Kunisch, *Local minimization algorithms for dynamic programming equations*, SIAM Journal on Scientific Computing, 38 (2016), pp. A1587–A1615, `https://doi.org/https://doi.org/10.1137/15M1010269`.

[70] D. Kalise, S. Kundu, and K. Kunisch, *Robust feedback control of nonlinear PDEs by numerical approximation of high-dimensional Hamilton–Jacobi–Isaacs equations*, SIAM Journal on Applied Dynamical Systems, 19 (2020), pp. 1496—–1524, `https://doi.org/10.1137/19M1262139`.

[71] A. Karakhanyan and X.-J. Wang, *On the reflector shape design*, J. Differential Geom., 84 (2010), pp. 561–610, `https://doi.org/10.4310/jdg/1279114301`.

[72] N. E. Karoui, S. Peng, and M. C. Quenez, *Backward Stochastic Differential Equations in finance*, Mathematical Finance, 7 (1997), pp. 1–71, `doi:10.1111/1467-9965.00022`.

[73] E. L. Kawecki, *A Discontinuous Galerkin Finite Element Method for uniformly elliptic two dimensional oblique boundary-value problems*, SIAM J. Numer. Anal., 57 (2019), pp. 751–778, `https://doi.org/10.1137/17M1155946`.

[74] S. Kilianová and M. Trnovska, *Robust portfolio optimization via solution to the Hamilton–Jacobi–Bellman equation*, International Journal of Computer Mathematics, 93 (2014), pp. 1–10, `https://doi.org/10.1080/00207160.2013.871542`.

[75] J. Kim and I. Yang, *Hamilton-Jacobi-Bellman equations for Q-learning in continuous time*, in Proceedings of the 2nd Conference on Learning for Dynamics and Control, A. M. Bayen, A. Jadbabaie, G. Pappas, P. A. Parrilo, B. Recht, C. Tomlin, and M. Zeilinger, eds., vol. 120 of Proceedings of Machine Learning Research, PMLR, 2020, pp. 739–748.

[76] M. J. Kim, Y. Choi, and W. K. Chung, *Bringing nonlinear $H_\infty$ optimality to robot controllers*, IEEE Transactions on Robotics, 31 (2015), pp. 682–698, https://doi.org/10.1109/TRO.2015.2419871.

[77] S. Kozpinar, M. Uzunca, and B. Karasözen, *Pricing European and American options under Heston model using Discontinuous Galerkin Finite Elements*, Mathematics and Computers in Simulation, 177 (2020), pp. 568–587, https://doi.org/10.1016/j.matcom.2020.05.022.

[78] N. V. Krylov, *Nonlinear Elliptic and Parabolic Equations of the Second Order*, Springer, New York, 1987.

[79] N. V. Krylov, *On the rate of convergence of Finite-Difference approximations for uniformly nondegenerate elliptic Bellman's equations*, Appl. Math. Optim., 69 (2014), pp. 431–458, https://doi.org/10.1007/s00245-013-9228-y.

[80] N. V. Krylov, *On the rate of convergence of Finite-Difference approximations for elliptic Isaacs equations in smooth domains*, Communications in Partial Differential Equations, 40 (2015), pp. 1393–1407.

[81] A. Kunoth, C. Schneider, and K. Wiechers, *Multiscale methods for the valuation of American options with stochastic volatility*, International Journal of Computer Mathematics, 89 (2012), pp. 1145–1163, https://doi.org/10.1080/00207160.2012.672732.

[82] H. J. Kushner and P. G. Dupuis, *Numerical methods for stochastic control problems in continuous time*, in Applications of Mathematics, vol. 24, Springer-Verlag, New York, 2 ed., 2001.

[83] A. Lachapelle and M.-T. Wolfram, *On a mean field game approach modeling congestion and aversion in pedestrian crowds*, Transportation Research Part B: Methodological, 45 (2011), pp. 1572–1589, https://doi.org/10.1016/j.trb.2011.07.011.

[84] O. Lakkis and T. Pryer, *A Finite Element Method for nonlinear elliptic problems*, SIAM J. Sci. Comput., 35 (2013), pp. A2025–A2045.

[85] J.-M. Lasry and P.-L. Lions, *Mean field games*, Japanese Journal of Mathematics, 2 (2007), pp. 229—260, https://doi.org/10.1007/s11537-007-0657-8.

[86] P.-L. Lions, *Generalized solutions of Hamilton-Jacobi equations*, in Research Notes in Mathematics, vol. 69, Pitman Advanced Publishing Program, Boston, 1982.

[87] P. L. Lions, *Neumann type boundary conditions for Hamilton-Jacobi equations*, Duke Mathematical Journal, 52 (1985), pp. 793–820, https://doi.org/10.1215/S0012-7094-85-05242-1.

[88] M. Mielkie and M. Davison, *Investigating the market price of volatility risk for options in a regime-switching market*, Econometric Modeling: Capital Markets - Risk eJournal, (2013), http://dx.doi.org/10.2139/ssrn.2326534.

[89] K. A. Mitchell and J. R. Mahoney, *Invariant manifolds and the geometry of front propagation in fluid flows*, Chaos: An Interdisciplinary Journal of Nonlinear Science, 22 (2012), p. 037104, https://doi.org/10.1063/1.4746039.

[90] M. Neilan, A. J. Salgado, and W. Zhang, *Numerical analysis of strongly nonlinear PDEs*, Acta Numerica, 26 (2017), pp. 137–303, https://doi.org/10.1017/S0962492917000071.

[91] N. Nüsken and L. Richter, *Solving high-dimensional Hamilton-Jacobi-Bellman PDEs using neural networks: perspectives from the theory of controlled diffusions and measures on path space*, Partial Differential Equations and Applications, 2 (2021), pp. 1–48, https://doi.org/10.1007/s42985-021-00102-x.

[92] O. A. Oleĭnik and E. V. Radkevič, *Second order equations with nonnegative characteristic form*, Plenum Press, New York-London, 1973. Translated from the Russian by Paul C. Fife.

[93] L. Ortiz-Gracia and C. W. Oosterlee, *Robust pricing of European options with wavelets and the characteristic function*, SIAM Journal on Scientific Computing, 35 (2013), pp. B1055—-B1084, https://doi.org/10.1137/130907288.

[94] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, Journal of Computational Physics, 79 (1988), pp. 12–49, https://doi.org/https://doi.org/10.1016/0021-9991(88)90002-2.

[95] C. Parzani and S. Puechmorel, *On a Hamilton-Jacobi-Bellman approach for coordinated optimal aircraft trajectories planning*, Optimal Control Applications and Methods, 39 (2018), pp. 933–948, https://doi.org/10.1002/oca.2389.

[96] G. D. Philippis and A. Figalli, *The Monge-Ampère equation and its link to Optimal Transportation*, Bulletin of the American Mathematical Society, 51 (2014), pp. 527–580, https://doi.org/10.1090/S0273-0979-2014-01459-4.

[97] E. Pindza, K. C. Patidar, and E. Ngounda, *Implicit-explicit predictor-corrector methods combined with improved spectral methods for pricing European style vanilla and exotic options*, Electronic transactions on numerical analysis, 40 (2013), pp. 268–293.

[98] C. Reisinger and Y. Zhang, *Rectified deep neural networks overcome the curse of dimensionality for nonsmooth value functions in zero-sum games of nonlinear stiff systems*, Analy-

sis and Applications, 18 (2020), pp. 951–999, https://doi.org/https://doi.org/10.1142/S0219530520500116.

[99] X. Ros-Oton, *Obstacle problems and free boundaries: An overview*, SeMA Journal, 75 (2018), pp. 399–419, https://doi.org/10.1007/s40324-017-0140-2.

[100] E. Rouy and A. Tourin, *A viscosity solutions approach to shape-from-shading*, SIAM Journal on Numerical Analysis, 29 (1992), pp. 867–884, https://doi.org/10.1137/0729053.

[101] A. J. Salgado and W. Zhang, *Finite Element approximation of the Isaacs equation*, ESAIM Mathematical Modelling and Numerical Analysis, 53 (2019), pp. 351–374.

[102] O.-S. Serea, *On reflecting boundary problem for optimal control*, SIAM Journal on Control and Optimization, 42 (2003), pp. 559–575, https://doi.org/10.1137/S0363012901395935, https://doi.org/10.1137/S0363012901395935.

[103] R. Seydel, *Tools for Computational Finance*, Universitext, Springer-Verlag, London, 2012, https://doi.org/10.1007/978-1-4471-2993-6.

[104] I. Smears and E. Süli, *Discontinuous Galerkin Finite Element approximation of Hamilton–Jacobi–Bellman equations with Cordes coefficients*, SIAM J. Numer. Anal., 52 (2014), pp. 993–1016, https://doi.org/10.1137/130909536.

[105] P. Soravia, *Equivalence between nonlinear $H_\infty$ control problems and existence of viscosity solutions of Hamilton—Jacobi—Isaacs equations*, Applied Mathematics and Optimization, 39 (1999), pp. 17–32, https://doi.org/10.1007/s002459900096.

[106] P. E. Souganidis, *Two-player, zero-sum differential games and viscosity solutions*, in Stochastic and Differential Games. Annals of the International Society of Dynamic Games, vol. 4, Birkhäuser, Boston, MA, 1999, ch. 2, pp. 69–104, https://doi.org/10.1007/978-1-4612-1592-9_2.

[107] P. E. Souganidis, *Two-player, zero-sum differential games and viscosity solutions*, in Stochastic and Differential Games. Annals of the International Society of Dynamic Games, M. Bardi, T. Raghavan, and T. Parthasarathy, eds., vol. 4, Birkhäuser, Boston, MA, 1999, ch. 2, pp. 69–104.

[108] D. Y. Tangman, A. Gopaul, and M. Bhuruth, *Numerical pricing of options using high-order compact Finite Difference schemes*, Journal of Computational and Applied Mathematics, 218 (2008), pp. 270–280, https://doi.org/10.1016/j.cam.2007.01.035.

[109] C. Wang and J. Wang, *A primal-dual weak Galerkin Finite Element Method for second order elliptic equations in non-divergence form*, Mathematics of Computation, 87 (2018), pp. 515–545, https://doi.org/10.1090/mcom/3220.

[110] P. WILMOTT, *Paul Wilmott on Quantitative Finance*, John Wiley & Sons Ltd., Chichester, 2 ed., 2006.

[111] G. WINKLER, T. APEL, AND U. WYSTUP, *Valuation of options in Heston's stochastic volatility model using Finite Element Methods*, Foreign Exchange Risk, (2001), pp. 278–313.

[112] X. WU, H. ZHOU, AND S. WANG, *Estimation of market prices of risks in the G.A.R.C.H. diffusion model*, Economic Research-Ekonomska Istraživanja, 31 (2018), pp. 15–36, https://doi.org/10.1080/1331677X.2017.1421989.

[113] J. XU AND L. ZIKATANOV, *A monotone Finite Element scheme for convection-diffusion equations*, Mathematics of Computation, 68 (1999), pp. 1429–1446.

[114] J. YONG AND X. Y. ZHOU, *Dynamic Programming and HJB equations*, in Stochastic Controls: Hamiltonian Systems and HJB Equations, Springer New York, 1999, ch. 4, pp. 157–215.

[115] W. ZHANG AND R. H. NOCHETTO, *Discrete ABP estimate and convergence rates for linear elliptic equations in non-divergence form*, Foundations of Computational Mathematics, 18 (2018), pp. 537–593, https://doi.org/10.1007/s10208-017-9347-y.

[116] S.-P. ZHU AND W.-T. CHEN, *A predictor–corrector scheme based on the ADI method for pricing American puts with stochastic volatility*, Computers and Mathematics with Applications, 62 (2011), pp. 1–26, https://doi.org/10.1016/j.camwa.2011.03.101.

[117] S.-P. ZHU AND W.-T. CHEN, *A predictor–corrector scheme based on the ADI method for pricing American puts with stochastic volatility*, Computers and Mathematics with Applications, 62 (2011), pp. 1–26.

[118] O. ZIENKIEWICZ, R. TAYLOR, AND J. ZHU, *The Finite Element Method: Its Basis and Fundamentals (Seventh Edition)*, Butterworth-Heinemann, Oxford, seventh edition ed., 2013, https://doi.org/10.1016/C2009-0-24909-9.

[119] R. ZVAN, P. A. FORSYTH, AND K. R. VETZAL, *Penalty methods for American options with stochastic volatility*, Journal of Computational and Applied Mathematics, 91 (1998), pp. 199–218, https://doi.org/10.1016/S0377-0427(98)00037-5.