Hierarchies d'autoroutes pour les équations d'Hamilton-Jacobi-Bellman (HJB)

Shanqing Liu

INSTITUT POLYTECHNIQUE DE PARIS

ÉCOLE POLYTECHNIQUE

IP PARIS

2023IPPAX107

Thèse de doctorat

Highway Hierarchies for HJB Equations

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École polytechnique

École doctorale n°574 École doctorale de mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques Appliquées

Thèse présentée et soutenue à Palaiseau, le 22 Décembre 2023, par

SHANQING LIU

Composition du Jury :

Hasnaa Zidani Professeur, INSA Rouen Normandie	Rapporteur
Roberto Ferretti Professeur associé, Université Roma Tre	Rapporteur
Jérome Darbon Professeur associé, Brown University	Examinateur
Jean-Marie Mirebeau Directeur de recherche, CNRS et ENS Paris-Saclay	Président du jury
Alexander Vladimirsky Professeur, Cornell University	Examinateur
Marianne Akian Directeur de recherche, INRIA et Ecole polytechnique (CMAP)	Directeur de thèse
Stéphane Gaubert Directeur de recherche, INRIA et Ecole polytechnique (CMAP)	Directeur de thèse

Acknowledgments

***

I would like to thank all of those who accompanied me during the three years of the thesis.

First of all, my sincerest thanks go to Marianne Akian and Stéphane Gaubert. I will always remember the day of the "dynamic programming" class, when I came to Marianne to ask for a chance for potential research topics on that field, and when I was ignorant master student who was not knowing and unprepared for what will happens ahead. Then the next days, it was Marianne who opened the door of her office to me, and the door to everything I have learned, accomplished and established during those three years and all beyond. Days were challenging in the beginning, especially within the situation of COVID when we started to work. But Marianne generously devoted plenty of her time and energy to helping me navigate the initial stages. More after, she was always there for the three years, remained patient and attentive for every single details of either naive or ill-formed questions of me. When look back to my first draft of paper at the very beginning, I realized it was disastrous–even the fundamental mathematical structure was poorly constructed, much like myself at the time. It was with Mariannes' guidance that I was able to work and process my research during the thesis in these three years, and it is also her mentorship which played a crucial role in shaping my journey in research. Of course Stéphane is always there with his enthusiasm and boundaries knowledge in mathematics. I've always held my sincerest respect for Stéphane. Your passion for research and mathematics, your divergent and creative thought which origins from your deep knowledge across variety of subjects have always been a motivation and inspiration for my research. I felt confident to try and explore my ideas because I have you as a "cornerstone" for everything in the three years. I will remember the discussion we had, your guidance on how to approach thinking and working, and your attention to every technical detail. I see you as a role model for me to chase in my academic road.

I want to thank Hasnaa Zidani and Roberto Ferretti for accepting to review this manuscript. Your insightful and constructive feedback greatly contributed to improving the quality of the manuscript. My special thanks to Hasnaa, as during SAMI-MODE 2022 in Limoges, we had a short discussion on the convergence rate of semi-Lagrangian scheme, which serves as part of the initial motivation of Chapter 4 of this manuscript.

I want to thank Jean-Marie Mirebeau for accepting to be the president of the defense. It was during your talk at the Mater Optimisation seminar where I first time learned about the fast-marching method. Additionally, thanks for your kind and helpful responses to my questions during the early stages of my thesis. I also wish to thank Jérome Darbon and Alexander Vladimiesky for their interesting in my work, and for accepting to be the examiners of my thesis. I am especially grateful for all of the jury members to present in the defence, which is scheduled on the last workday of 2023.

Thanks to the member of team Tropical for numerous support, discussion and encouragement all the time. My special thank goes to Yang, for our discussion, work and everything beyond. You teach and guide me a lot and I see you as a minor advisor of thesis! Also special thanks

to Hanadi, who is always serious and efficient for all the administrarive staffs I have. Thanks to CMAP! It has been nearly 4 years since I first came to CMAP for an internship. I will miss everything here. Also thanks to 91.06 which brings me more than 1000 times between apartment and CMAP, though can be later sometimes.

Thanks to all friends and colleagues here in CMAP and Ecole Polytechnique, especially to the members of office 20.03 who form a lovely and attractive place to work, and to Claire, Constantin, Chen, Haoyang, Liding, Manon, Songbo, Thibault, Wanqing, who have shared an important moment and period in my life. Thanks to the team Discrete Mathematics, François, Anne, Apolline, Claire, Ariane and Benoit. And lastly, of course, my deepest thanks to Kang, with whom I believe I have spend the most time within these three years. We share and talk evetything and for sure this will last forever.

Finally, thanks to my family. This thesis is dedicated to you.

Contents

***

Chapter 1

Introduction

***

1.1 Introduction

1.1.1 Deterministic Optimal Control Theory

The concept of *optimal control theory* involves finding an optimal strategy to optimize a certain objective functional, where this objective functional depends on the trajectory of both the control and state variables along time. In continuous-time deterministic cases, the evolution of the state is determined by an ordinary differential equation, and the objective involves an integral of a certain function of state and control over time. The formulation of objective functional is quite flexible. For instance, one may seek to optimize an integral over a fixed time horizon, or until a time horizon where the controlled state first reaches a certain target, or over an infinite time horizon, for which a discounted rate is typically used. The dynamics and objective functional may also depend on time [BC08].

In various contexts, the optimum of control problem depends on the initial state of the system. Thus, it is natural to consider the *value function* which is a mapping from the state space of the problem to $\mathbb{R}$. The value function then maps an initial state to the optimal value of the control problem associated with that state. A closely related approach to the concept of value function for analyzing the optimal control problem is the *dynamic programming principle*, which was first formulated by Richard Bellman in $1950's$ [BCC57]. It asserts that, the optimal control for the problem will remain an optimal control at any successive states along the optimal trajectory. Let us begin by considering that the value function is sufficiently smooth, that is, differentiable everywhere. By dynamic programming principle, one obtains that the value

function is a solution of a non-linear partial differential equation, the so-called *Hamilton-Jacobi-Bellman* equation, of the form:

$$F(x, v(x), \nabla v(x)) = 0 \ . \tag{1.1}$$

Equation (1.1) indeed provides a sufficient and necessary optimality condition for the control problem. Moreover, once (1.1) is solved, it allows one to compute a *closed-loop* optimal control, meaning that the optimal control is expressed as a function of the state. In practical applications, this leads to a solution that is robust against system perturbations.

The dynamic programming principle and HJB equation provide powerful tools addressing optimal control problems. However, the assumption that the value function is differentiable everywhere is too restrictive. In the early 1980s, Crandall and Lions introduced the notion of *viscosity solution* [CL83; CEL84]. Uniqueness results for the first order equation (1.1) are also established. Since then, a wide range of deterministic optimal control problems has been related to HJB equations of the form (1.1) in the viscosity sense: under certain regularity assumptions, the value function of a deterministic optimal control problem is the unique viscosity solution of the associated HJB equation.

For optimal control problems, it is natural and important to consider state constraints, as they often arise in practical applications. In these contexts, the state of the system is required to remain within the closure of a certain open domain $\overline{\Omega}$. For these problems, some controllability assumptions on the dynamics and on boundaries of state constraints are required. Additionally, the characterization of value functions by means of HJB equations should also be addressed in state constrained sense. Among the first efforts on state constrained optimal control problems, we mention the works of Soner [Son86a; Son86b]. In these works, Soner introduced the so-called *inward pointing qualification condition* (IPQ), which indeed require that at every point of the boundary of Ω, there exists a field of the system pointing inward the domain Ω. Assuming this condition, along with other regular assumptions on $\overline{\Omega}$, typically that it is compact with $\mathcal{C}^2$ boundary, the value function is bounded and uniformly continuous on $\overline{\Omega}$. Moreover, the notion of *constrained viscosity solution* is also proposed, which is defined as a viscosity subsolution on Ω and a viscosity supersolution on $\overline{\Omega}$. The property that the value function is a viscosity supersolution on $\overline{\Omega}$ imposes a boundary condition. Then, the value function of the state constrained problem can be characterized as the unique constrained viscosity solution of the HJB equation. Afterward, as the IPQ condition may not hold in some situations, Frankowska and co-workers introduced another controllability assumption in a series of works [Fra93; FV00; FP00], known as the *outward pointing qualification condition* (OPQ). This condition requires that every point on the boundary of Ω can be reached by a trajectory originating from a point within the interior of Ω. In this framework, the value function of the control problem is characterized as the unique lower semicontinuous solution of the associated HJB equation. We should also mention the recent works of [BFZ10; BFZ11] which characterized the value function of the state constrained problems without any controllability assumptions. In this thesis, we shall consider in particular the exit time problem (in Chapter 4). In this problem, in addition to the boundary of the state constraint, the boundary condition for the target set should be well defined. We also refer to [CL90] for reference.

1.1.2 Numerical Approximation: Grid Based Methods

Up to rare cases, optimal control problems and HJB equations can only be solved approximately using numerical methods. Various numerical schemes have been proposed to these problems since the pioneering works in [CL84; Cap83; Fal87]. First of all, since HJB equation is itself a non-linear partial differential equation, it can be numerically approximated using *finite difference schemes*, which are among the most common approaches for numerically solving PDEs. In finite

difference schemes, the state space is discretized using a grid, and the PDEs are solved by approximating the partial derivatives using the values on grid nodes. These methods are referred to as *backward* or *forward* finite difference schemes, depending on the nodes that are used for approximation. However, since the primary goal is to approximate the value function of an optimal control problem, the convergence of these numerical schemes should also be understood in the context of viscosity solutions. In essence, achieving convergence in numerical schemes requires satisfying three conditions: monotonicity, consistency, and stability (see [BS91]). To attain this convergence, a technique known as *upwind* correction is typically employed. This correction helps in selecting the appropriate neighboring grid points for approximating derivatives.

Another classical numerical method used to approximate HJB PDEs is the *Semi-Lagrangian* scheme. This scheme arises along with the idea of approximating the continuous optimal control problem with a discrete time control problem [Cap83; DI84; Fal87; FF14]. Namely, one can first apply an Euler time-discretization on the dynamics, with a time step Δt. Next, the cost functional, typically represented as an integral, is approximated using the trapezoidal-like sum. This results in a discrete time optimal control problem, and the dynamic programming equation associated with this discrete time problem serves as an approximation of the HJB equation. With sufficient regularity assumptions, one can expect a convergence rate in the order of Δt. However, it is important to note that this method involves only semi-discretization in time and is defined among the entire state space, making it impractical for direct implementation. To make it feasible for practical computations, further discretization in space is required. Consider a regular grid with a mesh step Δx for discretizing the state space, the fully discretized scheme involves applying the Semi-Lagrangian scheme at the grid nodes. Moreover, when the point in the next step, derived from the dynamics, does not fall within the grid, which is often the case, the value at this node is computed through interpolation based on its neighboring nodes. A convergence result is obtained when both Δt and $\frac{\Delta x}{\Delta t}$ tend to 0. One interesting property of the fully discretized scheme is that it can be thought of as a dynamic programming equation of a stochastic optimal control problem [KD01]. In this context, the grid nodes can be interpreted as the state space, and interpolation parameters can be interpreted as the transition probabilities of a controlled Markov chain. As we shall see in Chapter 5, we make use of this property to demonstrate an improved convergence result.

1.1.3 Towards Mitigating the Curse of Dimensionality

Following the previous discussion, one can think that these researches on the characterization of the value function as the unique viscosity solution of the HJB equation, as well as approximating the solution using numerical schemes are rather complete. However, one major difficulty that prevents this approach to be used on real applications is the well known *curse-of-dimensionality*, which was first expressed by Richard Bellman in [BCC57]:

- *"...what casts the pall over our victory celebration? It is the curse of dimensionality, a malediction that has plagued the scientist from the earliest days."*

Indeed, the HJB equation is formulated in the same dimension as the state space, which typically has a very high dimension in real applications. Setting aside the problems related to the regularity of the value function, solving high dimensional PDEs is a challenging problem in itself and constitutes a field of study. After discretization, the size of the (nonlinear) systems to be solved is exponential in the dimension, making the numerical computation untractable even on modern computers. Moreover, reading, writing and storing the grid nodes as well as the solution is in itself a problem due to the huge size. In practice, numerical computation is feasible only in a dimension $d \leqslant 4$ in modern computers. Mitigating the curse of dimensionality is the primary motivation driving all the researches in this thesis.

One additional difficulty is that solving the discretized equation requires an *iterative* procedure. At each iteration, a computation is executed at every node of the grid, and this process is repeated until convergence. These iterations are rather expensive. An interesting acceleration method, know as the fast marching method (FMM), and also referred as a *single pass* method, has been proposed originally by Sethian [Set96] and by Tsitsiklis [Tsi95], then further studied for instance in [SV01; SV03; CF07; For09; CCV14; Mir19]. These methods aim to reduce computational effort and obtain an approximate solution in such a way that, at every point of the discretization grid, the value is computed at most k times, where k is a bound unrelated to the discretization mesh. Such an approach was initially introduced to solve the front propagation problem, then extended to more general stationary Hamilton-Jacobi equations. It takes advantage of the property that the evolution of the region behind a "propagation front" is monotonically increasing, the so called "causality" property. Depending on the choice of update operators, which correspond to the discretization schemes that are used, one can distinguish between the finite difference fast marching method and the Semi-Lagrangian fast marching method. In particular, for the update operator, the size of the neighborhood of each node should be adjusted accordingly.

One of the interests of fast marching method appears in the computational complexity. In general, the fast marching method implemented in a $d-$dimensional grid with M points requires a number of arithmetic operations in the order of $K_d M \log(M)$, in which the constant $K_d \in [2d, D^d]$ depends on the type of discrete neighborhood that is considered (and D is the maximal diameter of neighborhoods). To efficiently achieve this complexity bound, a particular date structure should be adjusted. For the classical fast marching method, the data of nodes are normally stored using two types of data structure: A full $d-$dimensional matrix (or tensor), which contains the information of the whole discretization grid and the value functions computed at each step; A dynamical linked list, which contains the information of the narrow band nodes using the value function. We give more details of the data structure for fast marching method in Chapter 4, where we introduce a new data structure for our method, in which the grid is dynamically constructed around the optimal trajectory.

1.1.4 Max-plus Numerical Methods

As previously mentioned, both the finite difference scheme and the fully discretized Semi-Lagrangian scheme lead to equations that can be interpreted dynamic programming equations for stochastic optimal control problems with discrete time and state spaces. More recently, max-plus based discretization schemes have been developed for solving first-order HJB equations. These schemes rely on a max-plus basis representation of the discretized value function, which leads to a discrete time deterministic optimal control problem. In a broad sense, these methods take advantage of the *max-plus linearity* of the evolution semigroup of the HJB PDE, the so called *Lax-Oleinik semigroup*. After a time discretization, this allows one to approximate the value function for a given time horizon, by a supremum of appropriate "basis functions", for instance quadratic forms. These supremums are propagated through the action of the Lax-Oleinik semigroup between two successive time steps. In particular, in the work of Fleming and McEneaney [FM00], it is demonstrated that any semiconvex function can be represented as the max-plus linear combination of quadratic functions centered at a dense countable subset. Then, assuming semiconvexity of the value function, they approximate it at a given time horizon using the max-plus linear combination of quadratic functions. The propagation between two successive time steps is done by applying a max-plus linear operator on the coefficients of max-plus linear combination. This approximation can be interpreted as a dynamic programming equation for a discrete time deterministic optimal control problem.

Alternatively in [AGL08], a similar form of approximation for the value function was pro-

posed. To establish the recursive equations for the scalar coefficients, the authors introduced a family of "test functions". The iterative computation of these scalar coefficients in this case involves the application of a nonlinear operator. This operator can be thought of as a projection onto the space of basis functions and then a projection onto the space of test functions. This approximation scheme can be interpreted as a dynamic programming equation of a deterministic zero-sum two player game. Furthermore, the authors provide a convergence result with detailed error estimates and demonstrate that the computational complexities of their approaches remain comparable to those of classical grid-based methods. In Chapter 6, we shall combine these max-plus approximations with direct methods, leading to a higher degree of accuracy.

One way to overcome the curse of dimensionality is to assume some structure of the control problem. This is particularly what was proposed by McEneaney's curse of dimensionality free method in [McE07]. In this work, McEneaney considers infinite horizon switched optimal control problems, for which the Hamiltonian is expressed as a maximum of finite many "simpler" Hamiltonians. Each of these Hamiltonians is a linear-quadratic form originating from a linear quadratic optimal control problem. The author demonstrates that the complexity exhibits cubic growth in dimension (of the state)(see also [McE09]). This complexity, however, is bounded by a number that is exponential in the number of time steps, which is referred to as the "curse of complexity". Several "pruning" methods are then proposed to improve such a complexity bound, for instance, in [GMQ11; Qu14b].

1.1.5 Concentrating on Optimal Trajectories

Another efficient way to overcome the curse-of-dimensionality is to replace the general problem of solving the HJ equation and approximating the value function in the entire state space with the computation of only one or several optimal trajectories with a fixed initial state. The latter problem can be solved, under some convexity assumptions, by the Pontryagin Maximum Principle approach [RZ98; RZ99; Tré05], or by direct methods[Tré05; Bon+06]. In the same inspiration, in discrete time setting, one can use the stochastic dual dynamic programming (SDDP) method, which was first introduced in [PP91]. It is designed to solve deterministic or stochastic control problems with a specific structure where the costs are jointly convex with respect to state and control, in the case of minimization, and the dynamics are linear with respect to both state and control. Such a special structure guarantees that the value function is convex at every time horizon. Thus, the value function is approximated by a finite supremum of affine maps (that is, a piecewise affine convex map), and the approximated value function, together with the optimal trajectories starting from the fixed initial state, can then be computed efficiently using linear programming solvers. We refer to [Sha11; GLP15] for the convergence of SDDP. In cases where the assumptions on costs and dynamics are not satisfied, meaning in the absence of convexity, the SDDP method typically only leads to a local optimum. In Chapter 7, we develop a new numerical method, which can be thought of as an extension of SDDP method to semiconcave problems.

More recently, other methods consist in exploiting the structure of the problem, in particular to reduce the set of possible trajectories among which the optimization is done. For instance, in [AFS19; AFS20], the authors introduced a tree-structured discretization, taking advantage of the Lipschitz continuity of the value function. In [BGZ22], the authors introduced an adaptive discretization in the control space, which has been shown to be efficient when the dimension of the control space is low.

Concentrating on optimal trajectories is the very initial idea of our works in Chapter 4, Chapter 6 and Chapter 7.

1.1.6 Recent Development of Numerical Methods

The development of efficient numerical methods for solving HJB equations remains a prominent research topic, in particular with several recent studies aiming to overcome the curse of dimensionality. One recent method, introduced in [DO16], is based on the Hopf formula. This method focus on the HJ equation with Hamiltonian depends only on the gradient of the value function. Instead of discretization, the authors propose a method to solve HJ equation by combining the Hopf formula with the split Bregman iterative approach ([GO09]). The authors also show that their method has a complexity bound that is polynomial in the dimension. Deep learning and neural network techniques are also applied to solve the HJB equation and to find a feedback control law, for instance in [Kan+21; DDM23; BPW23].

Other recent methods are based on tensor decomposition. Among them we can cite the approximation of HJB equation using low-rank hierarchical tensor product approximation together with Monte-Carlo method proposed in [OSS22]. Additionally, in [DKK21], the authors introduced a tensor train approximation for the value function of the control problem. Subsequently, the resulting nonlinear system is solved using a Newton iterative method. Developing a tropical analogue of low-rank tensor approximation for the value function to solve the HJB equation is one of the motivations for the studies in Chapter 7.

1.2 Contributions

In this thesis, we develop new numerical methods, for solving the deterministic optimal control problems and the associated first order Hamilton-Jacobi-Bellman equations. Additionally, we analyze the convergence, computational complexity, and regularity properties of these methods. Our primary objective is to tackle and mitigate the curse of dimensionality. One common idea to address this challenge is to concentrate on identifying one or several optimal trajectories with fixed initial and/or final conditions.

1.2.1 Summary and Organization

- In Chapter 3, we give essential background on deterministic optimal control theory and the numerical methods that will be used in the rest of this thesis.
- In Chapter 4, we focus on a particular and fundamental problem, the minimum time problem. We present our new algorithm, considering both continuous aspects and numerical approximation. We show the convergence and we establish a computational complexity bound w.r.t. certain error bound. We present numerical tests up to dimension 7, confirming the speed of the method.
- In Chapter 5, we analyze a particular Semi-Lagrangian scheme for the minimum time problem and the associated eikonal equation. We prove a regularity property for the discretized value function. We establish the convergence rate of both the semi-discretized scheme and fully discretized scheme. In particular, we apply the result to derive a sufficient condition for the ideal complexity bound of Chapter 4.
- In Chapter 6, we consider general finite horizon deterministic optimal control problems. First, we combine direct methods with max-plus finite element method, leading to an algorithm with a higher degree of accuracy. Then, we adapt the idea of Chapter 4 to this context, that allows one to obtain the ideal complexity bound with a relaxed condition.
- In Chapter 7, we introduce a novel method to approximate the problem with a fixed initial state. This method is inspired by, and can be thought of as, a generalization of

the (Stochastic) Dual Dynamic Programming algorithm adapted to semiconcave problems. We show that our method converges to the global optimum under certain regularity assumptions. We present numerical benchmarks on $N-$body problems.

1.2.2 Contribution of Chapters

We now present in details the chapters that contain original works and that constitute the contributions of this thesis.

1.2.2.1 Contribution of Chapter 4: A multilevel Fast Marching Method For the Minimum Time Problem

In Chapter 4, we introduce a new algorithm to approximate the solutions of a class of stationary Hamilton-Jacobi PDEs arising from minimum time problem optimal control problems. In particular, we focus on finding the minimum traveling time between two given sets $\mathcal{K}_{\text{src}}$ and $\mathcal{K}_{\text{dst}}$ in a given domain Ω, along with the optimal trajectories.

For this purpose, we address two problems, one involving the usual time direction called "arrival to $\mathcal{K}_{\text{dst}}$" and the other involving the reverse direction called "start from $\mathcal{K}_{\text{src}}$". We characterize the value functions $v_{\text{s}\to}, v_{\to\text{d}}$ of these two problems using two state constrained HJB equations in their respective directions. We then characterize the geodesic points using $v_{\text{s}\to}$ and $v_{\to\text{d}}$, in Proposition 4.3.3 and Lemma 4.3.5. Moreover, based on these two value functions, we define an open subdomain $\mathcal{O}_\eta$ of Ω,

$$\mathcal{O}_\eta = \{x \in (\Omega \setminus (\mathcal{K}_{\text{src}} \cup \mathcal{K}_{\text{dst}})) \mid \mathcal{F}_v(x) < \inf_{y\in\Omega}\{\mathcal{F}_v(y) + \eta\} \} \ , \tag{1.2}$$

where $\mathcal{F}_v(x) = v_{\text{s}\to}(x) + v_{\to\text{d}}(x) - v_{\text{s}\to}(x)v_{\to\text{d}}(x)$. We show the equivalence between the subdomain $\mathcal{O}_\eta$ and the $\delta-$geodesic points in Lemma 4.3.14 and Lemma 4.3.15. Based on these properties, we establish in Theorem 4.3.16 that, if we reduce the state space from Ω to $\mathcal{O}_\eta$, then for every x in the set of $\delta-$geodesic points with $\delta < \eta$, the new value functions $v^\eta_{\text{s}\to}(x)$ and $v^\eta_{\to\text{d}}(x)$ are equal to $v_{\text{s}\to}(x)$ and $v_{\to\text{d}}(x)$, respectively.

Our new algorithm takes advantage of the aforementioned properties. We rely on nested grid approximations, and look for the optimal trajectories by using the coarse grid approximations to reduce the search space in fine grids. More precisely, following a coarse approximation in the coarse grid, two approximate value functions $v^H_{\text{s}\to}, v^H_{\to\text{d}}$ are computed in the grid nodes. We then select the *active* nodes using an approximation formula derived from (1.2), where v is replaced by v^H and η is replaced by a parameter η_H. The active nodes can be thought of as neighborhood points around the optimal trajectory. The fine grid is dynamically constructed "around" these active nodes. Then, computations of the approximate value functions $v^h_{\text{s}\to}, v^h_{\to\text{d}}$ are only performed on the selected fine grid nodes. We give in Algorithm 4.2 the details of the two level method (2LFMM). Then, we prove in Theorem 4.4.3 the convergence of two level fast marching method, stating that if η_H is sufficiently large, the error estimate is as good as the one obtained by directly discretizing the whole domain with the fine grid. The concept of coarse-fine approximation can be extended to the multi-level case. Given a family of successive mesh steps $H_1 \leqslant H_2 \ldots \leqslant H_N = h$, and a family of real positive parameters $\{\eta_1, \eta_2, \ldots, \eta_{N-1}\}$, the multi-level fast marching method (MLFMM) is presented in Algorithm 4.3, and the convergence result is presented in Theorem 4.4.4.

We provide a computational complexity bound with respect to the error ε, in order to show the improvement of our algorithm compared to other grid based methods. To begin, we bound the space complexity of 2LFMM by the volume of tubular neighborhoods around optimal trajectories in Proposition 4.5.2. More precisely, given a coarse mesh step H, a fine mesh step

h and parameter η_H, the space complexity can be expressed as follows

$$\mathcal{C}_{spa}(H,h) = \widetilde{O}\Big(C^d\Big(\frac{1}{H^d} + \frac{(\eta_H)^{\beta(d-1)}}{h^d}\Big)\Big) \ , \tag{1.3}$$

where β is the "stiffness" of the value function around optimal trajectories, see Assumption (A6). This space complexity indeed involves nodes in coarse grid, included in a ball of volume of $O(1)$, and the nodes in fine grid, included in a tubular neighborhood of the optimal trajectory with volume of $O((\eta_H)^{\beta(d-1)})$. The same analysis as in two level case also works for the $N-$level case, for which the space complexity is deduced from a hierarchy of tubular neighborhood. One can observe that the complexity bound in (1.3), as well as the one for $N-$level case, is a function of the family of coarse mesh steps when h is fixed. Then, the mesh steps are chosen such that the complexity bound achieves it's minimum. Moreover, once the mesh steps are chosen, the complexity becomes a function of the number of levels N. Thus, N is again chosen to minimize the complexity. We present in Theorem 4.5.5 the main result for this complexity bound. To summarize, for our multi-level method, the number of arithmetic operations is in the order of $\widetilde{O}(C^d\varepsilon^{-\frac{1+(d-1)(1-\gamma\beta)}{\gamma}})$, where $C>1$ is a constant depending on the problem characteristics and $0<\gamma\leqslant 1$ is the convergence rate of the classical fast marching method. Hence, considering the dependence in ε only, we reduce the complexity bound from $\widetilde{O}(\varepsilon^{-\frac{d}{\gamma}})$ to $\widetilde{O}(\varepsilon^{-\frac{1+(d-1)(1-\gamma\beta)}{\gamma}})$. In typical situations in which the value function is smooth with a nondegenerate Hessian in the neighborhood of an optimal trajectory, one has $\beta = 1/2$. In exceptional cases, with a L_1-type geometry, one can get $\beta = 1$. Observe that the complexity bound reduces to $\widetilde{O}(C^d\varepsilon^{-1})$ when $\gamma=\beta=1$. Thus, considering the dependence in ε only, the complexity bound becomes $\widetilde{O}(\varepsilon^{-1})$ and is thus of same order as for one dimensional problems.

To numerically implement the algorithm, we introduce a dedicated data structure, a "hash table", to store the successive constrained (dynamically constructed) grids, for every level. Every time a new node is selected, we compute its slot by a hash function, then store the necessary information for computation in that slot, see details in Section 4.4.4. We present numerical tests up to dimension 7, and we analyze the effective complexity of our multilevel algorithm.

1.2.2.2 Contribution of Chapter 5: Convergence and Error Estimates of a Semi-Lagrangian Scheme for the Eikonal Equation

In Chapter 5, we consider a particular Semi-Lagrangian scheme for numerically solving the eikonal equation arising from the minimum time problem of reaching a target set $\mathcal{K}$, in which the time step varies depending on the state. The aim is to establish a sufficient condition for achieving a convergence rate of order 1 for both the semi-discretized scheme and the fully discretized scheme, where in the case of the semi-discretized scheme, the convergence rate is expressed in terms of the time step, while for the latter one, it is in terms of both the time step and the mesh step. This result is also applied to derive sufficient conditions for achieving a convergence rate of 1 for Semi-Lagrangian type fast marching method. To the best of our knowledge, this is the first time that such a convergence rate is established. We then apply this result to obtain the ideal complexity bound of the method proposed in Chapter 4.

We first consider the semi-discretized scheme. We represent the solution of the discretized system as the value function of a discrete time deterministic optimal control problem. The first main result is presented in Proposition 5.3.4, in which we show that under particular semiconcave assumptions on the dynamics and on the distance function to the target set (see details in Assumption (A8)), we obtain that the discrete time value function is semiconcave, that is

$$v^h(x+z) - 2v^h(x) + v^h(x-z) \leqslant C\|z\|^2, \quad \text{for every } x,z \in \mathbb{R}^d \setminus \mathcal{K} \ . \tag{1.4}$$

Sketch of Proof of Proposition 5.3.4. We derive this property by considering the discrete time optimal control problems with initial states $x + z$, $x - z$ and x, respectively. Let y^* denote the optimal trajectory for the problem with initial state x. We construct feasible trajectories, denoted as y^+ and y^- for the problems with initial states $x + z$ and $x - z$, in the following manner: y^+ and y^- initially follow the same control trajectory as for y^*; if y^* first reaches $\mathcal{K}$, then y^+ and y^- continue along a straight line to $\mathcal{K}$. Otherwise, if y^- reaches $\mathcal{K}$ first, after then, y^+ repeats two times the control of trajectory y^* until y^* reaches $\mathcal{K}$. After that, y^+ follows a straight line trajectory. (1.4) is subsequently deduced by calculating the costs of the trajectories y^-, y^+ and y^*, respectively. □

For the convergence rate, one side, namely $v - v^h$, is easier to bound from above since the discrete set of trajectories is a subset of the continuous one. In the other direction, bounding $v^h - v$ involves viscosity techniques and the semiconcavity of v^h. We present this result as the second main result in Theorem 5.3.5.

We then consider the fully discretized scheme, and we begin by considering a simple P_1 (piecewise linear) interpolation operator I_1. Our aim is to establish an upper bound on $\|w^h - v^h\|_\infty$, where w^h is the solution of the fully discretized scheme. For the upper bound of $w^h - v^h$, we use the fact that both the interpolation operator and Bellman operator associated with the discrete time deterministic control problem are non-expensive, with the Bellman operator exhibiting a contraction rate of $(1 - \frac{h}{f})$. Additionally, when v^h exhibits semiconcavity, the supremum over x of $(I_1[v^h] - v^h)(x)$ is bounded by Ch^2. For the upper bound of $v^h - w^h$, we first represent the solution of the fully discretized system as the value function of a stochastic control problem. Then, we show that, under semiconvex assumptions on the dynamics and on the distance function to the target set (see details in Assumption (A9)), we can derive an error in the order of h is. This result is presented in Proposition 5.4.4, which in short is as follows,

$$\sup_{x \in \mathbb{R}^d} (v^h - w^h)(x) \leqslant Ch \ . \tag{1.5}$$

Sketch of Proof. We consider a controlled Markov chain with initial state x. For any strategy σ^h, we construct a deterministic trajectory that follows the same control as the one associated with σ^h in the stochastic case. This trajectory is indeed a feasible trajectory for the deterministic discrete system. We then derive (1.5) by calculating the costs of the stochastic control problem with strategy σ^h, and of the deterministic problem with the constructed trajectory, which is mainly based on two properties: (i) the states of the Markov process have the property that the expectation of $\xi_{k+1} - \xi_k$ is $h\alpha$, and the covariance is bounded by h^2 (see (5.69)); (ii) A property related to the expectation of semiconvex functions (see Lemma 5.4.3). □

We then apply the convergence result of the fully discretized scheme to show that the fast marching methods, using update operators derived from a Semi-Lagrangian type discretization, have a convergence rate of order h under the assumptions we introduced, where h represents the mesh grid. As a consequence, the computational complexity of the multilevel fast marching method introduced in Chapter 4 depends solely on β, the stiffness of the optimal trajectory.

1.2.2.3 Contribution of Chapter 6: An Adaptive Multi-Level Max-Plus Method for Deterministic Optimal Control Problems

In Chapter 6, we consider finite horizon deterministic optimal control problems that involve both initial and final costs. First, we combine max-plus approximations with direct methods, leading to a numerical method with a higher degree of accuracy. Then, we extend the idea of dynamic grid refinement around tubular neighborhood of optimal trajectories, which was introduced in Chapter 4.

We characterize the optimality conditions by considering a pair of HJB PDEs associated to two optimal control problems: one involving forward dynamics with fixed initial state and free final state, and a dual one involving backward dynamics with fixed final state and free initial state. We adapt the max-plus finite element method to approximate the two value functions $v^t_{\mathrm{s}\rightarrow}$ and $v^t_{\rightarrow\mathrm{d}}$, for every $t \in \{0, \delta, \dots, T\}$. In more details, considering first $v^t_{\rightarrow\mathrm{d}}$, and given a finite family of basis functions $\{w_i\}_{1\leqslant i\leqslant p}$, $v^t_{\rightarrow\mathrm{d}}$ is approximated by a max-plus linear combination of the basis functions with coefficients $\{\lambda_i^{\rightarrow\mathrm{d},t}\}_{1\leqslant i\leqslant p}$. The recursive equation of scalars between two successive time steps is obtained by introducing a set of test functions $\{z_j\}_{1\leqslant j\leqslant q}$ (see details in Proposition 6.3.1). Our first work is based on the observation that the small time propagation of basis functions leads to a new optimal control problem,

$$\langle z_j, S^\delta[w_i]\rangle = \max\left\{ z_j(x(0)) + \int_0^\delta \ell(x(s), u(s))ds + w_i(x(\delta))\right\} . \tag{1.6}$$

We show in Proposition 6.3.2 and Lemma 6.3.3 that, under certain regularity assumptions on f and ℓ (see details in Assumption (A11)), choosing strongly concave basis functions and test functions w_i, z_j (for instance quadratic functions), within a given time horizon $\delta \leqslant \bar{\delta}$, the problem (1.6) is actually a concave program with respect to the trajectory $(x(\cdot), u(\cdot))$. This property can be explained by the "propagation" of the strong concavity of the initial and/or terminal costs in (1.6) over a small time horizon. It implies that Problem (1.6) can be solved exactly, or with an error that is negligible compared to δ, by employing a direct method. We propose to approximate this problem by a direct method. The complete algorithm is presented in Algorithm 6.1. The error estimate is also presented in Theorem 6.3.4, which is then a direct consequence of the results of [Lak07].

After obtaining the approximation of the two value functions, we apply a similar approach as in Chapter 4 to approximate the optimal trajectory. In this case, the value of the problem is represented (approximately) by the scalars $\lambda^t_{\mathrm{s}\rightarrow}$, $\lambda^t_{\rightarrow\mathrm{d}}$ in two directions, that is

$$v^* \approx \sup_{1\leqslant i,j\leqslant p} \left\{ \lambda_i^{\mathrm{s}\rightarrow,t} + \lambda_j^{\rightarrow\mathrm{d},t} + \langle w_i^{\mathrm{s}\rightarrow}, w_j^{\rightarrow\mathrm{d}}\rangle \right\}, \ \forall\, t \in [0, T] . \tag{1.7}$$

We then select sets of indices i and j that are $\eta-$optimal in (1.7). These indices are indeed in correspondance with neighborhoods in $\mathbb{R}^d$ of some dual optimal trajectories for $v^t_{\mathrm{s}\rightarrow}$ and $v^t_{\rightarrow\mathrm{d}}$, respectively. The (primal) optimal trajectories can then be identified based on these sets of indices, see Theorem 6.4.7.

We then extend the idea of dynamic grid refinement around the tubular neighborhood of optimal trajectories in Chapter 4. In particular, we use a hierarchy of finer and finer irregular grids to generate the basis functions and test functions. In the two level case, we first use two coarse grids to generate the basis functions and test functions for approximating the value functions in two directions. Then, given a parameter η^H, we identify the "active" nodes in coarse grids for both directions. These active nodes indeed correspond to the indices (i, j) which are η^H-optimal in (1.7). The coarse approximation of optimal trajectories is obtained using these active nodes. Then, we construct the fine grids around the active nodes. The basis functions and test functions for fine approximation will be generated by these grids. We present the complete two level method in Algorithm 6.2. The concept of coarse-fine approximation can be extended to the multi-level case. Given a family of successive mesh grids $\{\hat{G}^{H_l}, G^{H_l}\}_{1\leqslant l\leqslant m}$, and a family of real positive parameters $\{\eta_l\}_{1\leqslant l\leqslant m-1}$, the Adaptive $m-$level Max-Plus Approximation method is presented in Algorithm 6.3.

We show that using our algorithm, the number of basis functions needed to get a certain error ε is considerably reduced. Indeed, for a $d-$dimensional problem, under certain regularity assumptions, we get a complexity bound of $C^d(1/\varepsilon)^{\frac{1}{2}}$ arithmetic operations, for some constant

$C > 1$. This should be compared with methods based on regular grids, which yield complexity bounds of order $O(1/\varepsilon^{ad})$ in which $a > 0$ depends on regularity assumptions and on the order of the scheme. With our adaptive method, the curse of dimensionality remains only present in the term C^d. We present the main complexity result in Theorem 6.6.4. To compare with the computational complexity in Chapter 4, the use of max-plus approximations combined with direct methods leads to a higher degree of accuracy. Indeed, under appropriate regularity assumptions (in particular the assumptions we used in Chapter 5 to obtain a convergence rate of order h for the fast marching method), the method of Chapter 4 has a computational complexity of order $\mathcal{O}(\varepsilon^{-1-(d-1)(1-\beta)})$, in which the parameter $0 < \beta \leqslant 1$ measures the "stiffness" of the value functions near optimal trajectories. Typical instances are moderately stiff, and have a parameter $\beta = 1/2$, leading to a complexity of order $\mathcal{O}(\varepsilon^{-1-(d-1)/2})$. In contrast, we get here a complexity of order $\mathcal{O}(\varepsilon^{-\frac{1}{2}})$, with less demanding assumptions.

1.2.2.4 Contribution of Chapter 7: Semiconcave Dual Dynamic Programming and Its Application to Tropical Low-Rank Approximation of N-body System

In Chapter 7, we introduce a novel algorithm for numerically finding the value function, along with the optimal trajectory, for a class of finite horizon deterministic optimal control problems with a fixed initial initial state. In particular, the reward function (in maximization case) is only required to be semiconcave with respect to the state x.

We look for a tight approximation of the value function along the optimal trajectories starting from a given initial point x_0. We start with a (arbitrary) feasible trajectory for the control problem, and construct an initial upper approximate for the value function. At every iteration step, for the maximization problem, after a discretization in time, we approximate the value function, in a given time horizon, by a minimum of quadratic "basis" functions (see definition in (7.14)). However, the evolutionary semigroup associated with the maximization problem is max-plus linear instead of min-plus linear. To propagate the basis functions to the next time horizon, we solve a dual problem of the maximization problem (see the formulation of the dual problem in (7.19)). We then construct a new upper approximation of the value function based on the dual problem. The trajectory is then updated to an optimal trajectory derived from the current approximate value function. Thus, in every iteration, we add one more basis functions for the approximation. We present our new algorithm in Algorithm 7.1.

We also present a slight variant of the algorithm in Algorithm 7.2, which involves two loops in time: a backward-in-time loop for updating the approximate value function, and a forward-in-time loop for updating the trajectory. We show that our algorithm can be compared to, and can be thought of as an extension of, the (S)DDP method, in particular to handle situations involving the semiconcavity condition on the running reward. Indeed, it can be explained by adding a quadratic terms of "regularization" for the semiconcavity of the value function and of the running rewards. Following the SDDP method's approach, in every iteration, we indeed solve a new dual problem in the form of (7.31). We should the equivalence of this algorithm as the one in the approximation point of view in Proposition 7.3.6.

We show that our method converges to the global maximum under certain regularity assumptions. This is based on the property that the small time propagation preserves the semiconcavity, which we present in Proposition 7.4.4. Moreover, denoting $v_m^{t,h}$ the approximate value function obtained from our algorithm in iteration step m, we establish in Proposition 7.4.7 that this approximation is monotone with respect to the iterative step, and is upper and lower bounded. We present the convergence result in Theorem 7.4.9.

As an application, we employ our algorithm to construct a tropical low-rank tensor approximation, which can be thought of as a tropical analogue of the classical low-rank tensor decomposition (see in (7.59)), for a N-body system. The action functional of this system con-

sists of individual potential energy and kinetic energy, and the Coulomb interaction energy. We interpret this system using the framework of optimal control and Hamilton-Jacobi equation, based on the principle of least action (see in (7.61)). We present numerical benchmarks to determine the optimal trajectory and the grand state of each individual in Section 7.5.4.

Chapter 2

Introduction (en français)

***

2.1 Introduction

2.1.1 Théorie du Contrôle Optimal Déterministe

Le concept de *théorie du contrôle optimal* implique la recherche d'une stratégie optimale pour optimiser une certaine fonction objective, où cette fonction objective dépend de la trajectoire des variables de contrôle et d'état dans le temps. Dans les cas déterministes à temps continu, l'évolution de l'état est déterminée par une équation différentielle ordinaire, et l'objectif implique une intégrale d'une certaine fonction de l'état et du contrôle dans le temps. La formulation de l'objectif fonctionnel est très souple. Par exemple, on peut chercher à optimiser une intégrale sur un horizon temporel fixe, ou jusqu'à un horizon temporel où l'état contrôlé atteint pour la première fois une certaine cible, ou sur un horizon temporel infini, pour lequel un taux d'actualisation est généralement utilisé. La dynamique et l'objectif fonctionnel peuvent également dépendre du temps [BC08].

Dans divers contextes, l'optimum d'un problème de contrôle dépend de l'état initial du système. Il est donc naturel de considérer la *fonction de valeur* qui est un mappage de l'espace d'état du problème vers $\mathbb{R}$. La fonction de valeur fait alors correspondre un état initial à la valeur optimale du problème de contrôle associé à cet état. Une approche étroitement liée au concept de fonction de valeur pour l'analyse du problème de contrôle optimal est le *principe de programmation dynamique*, qui a été formulé pour la première fois par Richard Bellman dans [BCC57]. Elle affirme que le contrôle optimal du problème restera un contrôle optimal à tous les états successifs le long de la trajectoire optimale. Commençons par considérer que la fonction de valeur est suffisamment lisse, c'est-à-dire différentiable partout. Par le principe de la

programmation dynamique, on obtient que la fonction de valeur est une solution d'une équation différentielle partielle non linéaire, l'équation dite de *Hamilton-Jacobi-Bellman*, de la forme:

$$F(x, v(x), \nabla v(x)) = 0 \ . \tag{2.1}$$

L'équation (2.1) fournit en effet une condition d'optimalité suffisante et nécessaire pour le problème de contrôle. En outre, une fois que (2.1) est résolue, elle permet de calculer un contrôle optimal *boucle fermée*, ce qui signifie que le contrôle optimal est exprimé en tant que fonction de l'état. Dans les applications pratiques, cela permet d'obtenir une solution robuste aux perturbations du système.

Le principe de programmation dynamique et l'équation HJB constituent des outils puissants pour résoudre les problèmes de contrôle optimal. Cependant, l'hypothèse selon laquelle la fonction de valeur est partout différentiable est trop restrictive. Au début des années 1980, Crandall et Lions ont introduit la notion de *solution de viscosité* [CL83; CEL84]. Des résultats d'unicité pour l'équation du premier ordre (2.1) sont également établis. Depuis lors, un large éventail de problèmes de contrôle optimal déterministe a été relié aux équations HJB de la forme (1.1) dans le sens de la viscosité : sous certaines hypothèses de régularité, la fonction de valeur d'un problème de contrôle optimal déterministe est l'unique solution de viscosité de l'équation HJB associée.

Pour les problèmes de contrôle optimal, il est naturel et important de considérer les contraintes d'état, car elles apparaissent souvent dans les applications pratiques. Dans ces contextes, l'état du système doit rester dans la fermeture d'un certain domaine ouvert $\overline{\Omega}$. Pour ces problèmes, certaines hypothèses de contrôlabilité sur la dynamique et sur les limites des contraintes d'état sont nécessaires. En outre, la caractérisation des fonctions de valeur au moyen des équations HJB devrait également être abordée dans le sens des contraintes d'état. Parmi les premiers efforts sur les problèmes de contrôle optimal sous contrainte d'état, nous mentionnons les travaux de Soner [Son86a; Son86b]. Dans ces travaux, Soner a introduit ce que l'on appelle la condition de qualification du pointage vers l'intérieur (IPQ), qui exige en effet qu'en tout point de la frontière de Ω, il existe un champ du système pointant vers l'intérieur du domaine Ω. En supposant cette condition, ainsi que d'autres hypothèses régulières sur $\overline{\Omega}$, typiquement qu'il est compact avec $\mathbb{C}^2$ de frontière, la fonction de valeur est bornée et uniformément continue sur $\overline{\Omega}$. De plus, la notion de *solution de viscosité contrainte* est également proposée, qui est définie comme une sous-solution de viscosité sur Ω et une supersolution de viscosité sur $\overline{\Omega}$. La propriété selon laquelle la fonction de valeur est une supersolution de viscosité sur $\overline{\Omega}$ impose une condition limite. La fonction de valeur du problème sous contrainte d'état peut alors être caractérisée comme l'unique solution de viscosité sous contrainte de l'équation HJB. Par la suite, comme la condition IPQ peut ne pas être valable dans certaines situations, Frankowska et ses collègues ont introduit une autre hypothèse de contrôlabilité dans une série de travaux [Fra93; FV00; FP00], connue sous le nom de *condition de qualification du pointage vers l'extérieur* (OPQ). Cette condition exige que chaque point de la frontière de Ω puisse être atteint par une trajectoire provenant d'un point situé à l'intérieur de Ω. Dans ce cadre, la fonction de valeur du problème de contrôle est caractérisée comme l'unique solution semi-continue inférieure de l'équation HJB associée. Nous devons également mentionner les travaux récents de [BFZ10; BFZ11] qui ont caractérisé la fonction de valeur des problèmes à contraintes d'état sans aucune hypothèse de contrôlabilité. Dans cette thèse, nous considérerons en particulier le problème du temps de sortie (dans Chapter 4). Dans ce problème, en plus de la limite de la contrainte d'état, la condition limite pour l'ensemble cible doit être bien définie. Nous nous référons également à [CL90] pour référence.

2.1.2 Approximation numérique : Méthodes basées sur la grille

Jusqu'à de rares cas, les problèmes de contrôle optimal et les équations HJB ne peuvent être résolus qu'approximativement à l'aide de méthodes numériques. Divers schémas numériques ont été proposés pour ces problèmes depuis les travaux pionniers de [CL84; Cap83; Fal87]. Tout d'abord, l'équation HJB étant elle-même une équation aux dérivées partielles non linéaire, elle peut être approchée numériquement à l'aide de *schémas de différences finies*, qui font partie des approches les plus courantes pour la résolution numérique des EDP. Dans les schémas de différences finies, l'espace d'état est discrétisé à l'aide d'une grille et les EDP sont résolues en approximant les dérivées partielles à l'aide des valeurs sur les nœuds de la grille. Ces méthodes sont appelées schémas de différences finies *backward* ou *forward*, en fonction des nœuds utilisés pour l'approximation. Cependant, comme l'objectif principal est d'approximer la fonction de valeur d'un problème de contrôle optimal, la convergence de ces schémas numériques doit également être comprise dans le contexte des solutions de viscosité. Par essence, la convergence des schémas numériques nécessite de satisfaire trois conditions : monotonicité, cohérence et stabilité (voir [BS91]). Pour atteindre cette convergence, une technique connue sous le nom de correction *upwind* est généralement employée. Cette correction permet de sélectionner les points de grille voisins appropriés pour l'approximation des dérivées. Une autre méthode numérique classique utilisée pour approximer les EDP HJB est le schéma *Semi-Lagrangien*. Ce schéma est né avec l'idée d'approximer le problème de contrôle optimal continu par un problème de contrôle en temps discret [Cap83; DI84; Fal87; FF14]. On peut d'abord appliquer une discrétisation temporelle d'Euler à la dynamique, avec un pas de temps Δt. Ensuite, la fonction de coût, généralement représentée sous la forme d'une intégrale, est approximée à l'aide de la somme de type trapézoïdal. Il en résulte un problème de contrôle optimal en temps discret, et l'équation de programmation dynamique associée à ce problème en temps discret sert d'approximation à l'équation HJB. Avec des hypothèses de régularité suffisantes, on peut s'attendre à un taux de convergence de l'ordre de Δt. Cependant, il est important de noter que cette méthode n'implique qu'une semi-discrétisation en temps et qu'elle est définie sur l'ensemble de l'espace d'état, ce qui la rend impraticable pour une mise en œuvre directe. Une discrétisation supplémentaire dans l'espace est nécessaire pour permettre des calculs pratiques. Considérons une grille régulière avec un pas de maille Δx pour discrétiser l'espace d'état, le schéma entièrement discrétisé implique l'application du schéma semi-lagrangien aux nœuds de la grille. En outre, lorsque le point de l'étape suivante, dérivé de la dynamique, ne se trouve pas dans la grille, ce qui est souvent le cas, la valeur à ce nœud est calculée par interpolation sur la base de ses nœuds voisins. Un résultat de convergence est obtenu lorsque Δt et $\frac{\Delta x}{\Delta t}$ tendent tous deux vers 0. Une propriété intéressante du schéma entièrement discrétisé est qu'il peut être considéré comme une équation de programmation dynamique d'un problème de contrôle optimal stochastique [KD01]. Dans ce contexte, les nœuds de la grille peuvent être interprétés comme l'espace d'état, et les paramètres d'interpolation comme les probabilités de transition d'une chaîne de Markov contrôlée. Comme nous le verrons dans Chapter 5, nous utilisons cette propriété pour démontrer un résultat de convergence amélioré.

2.1.3 Vers l'atténuation de la malédiction de la dimensionnalité

Suite à la discussion précédente, on peut penser que les recherches sur la caractérisation de la fonction de valeur en tant que solution unique de viscosité de l'équation HJB, ainsi que l'approximation de la solution à l'aide de schémas numériques, sont assez complètes. Cependant, une difficulté majeure qui empêche cette approche d'être utilisée dans des applications réelles est le fameux *curse-of-dimensionality*, qui a été exprimé pour la première fois par Richard Bellman dans [BCC57] :

- *"...what casts the pall over our victory celebration? It is the curse of dimensionality, a malediction that has plagued the scientist from the earliest days."*

En effet, l'équation HJB est formulée dans la même dimension que l'espace d'état, qui a généralement une dimension très élevée dans les applications réelles. Si l'on met de côté les problèmes liés à la régularité de la fonction de valeur, la résolution d'EDP de haute dimension est un problème difficile en soi et constitue un domaine d'étude. Après discrétisation, la taille des systèmes (non linéaires) à résoudre est exponentielle en fonction de la dimension, ce qui rend le calcul numérique irréalisable, même sur les ordinateurs modernes. En outre, la lecture, l'écriture et le stockage des nœuds de la grille ainsi que de la solution constituent en soi un problème en raison de leur taille considérable. En pratique, le calcul numérique n'est possible que dans une dimension $d \leqslant 4$ sur les ordinateurs modernes. Atténuer la malédiction de la dimensionnalité est la motivation première de toutes les recherches menées dans le cadre de cette thèse. Une difficulté supplémentaire réside dans le fait que la résolution de l'équation discrétisée nécessite une procédure *itérative*. À chaque itération, un calcul est exécuté à chaque nœud de la grille, et ce processus est répété jusqu'à convergence. Ces itérations sont assez coûteuses. Une méthode d'accélération intéressante, connue sous le nom de méthode de "fast-marching" (FMM), et également appelée méthode *single pass*, a été proposée à l'origine par Sethian [Set96] et par Tsitsiklis [Tsi95], puis étudiée plus avant, par exemple dans [SV01; SV03; CF07; For09; CCV14; Mir19]. Ces méthodes visent à réduire l'effort de calcul et à obtenir une solution approximative de telle sorte qu'en chaque point de la grille de discrétisation, la valeur soit calculée au maximum k fois, où k est une limite non liée à la maille de discrétisation. Cette approche a été initialement introduite pour résoudre le problème de la propagation des fronts, puis étendue à des équations de Hamilton-Jacobi stationnaires plus générales. Elle tire parti de la propriété selon laquelle l'évolution de la région située derrière un "front de propagation" est monotone et croissante, ce que l'on appelle la propriété de "causalité". Selon le choix des opérateurs de mise à jour, qui correspondent aux schémas de discrétisation utilisés, on peut distinguer la méthode de fast-marching par différences finies et la méthode de fast-marching par semi-lagrangienne. En particulier, pour l'opérateur de mise à jour, la taille du voisinage de chaque nœud doit être ajustée en conséquence.

L'un des intérêts de la méthode de fast-marching réside dans sa complexité de calcul. En général, la méthode de fast-marching mise en oeuvre dans une grille à $d-$dimension avec M points nécessite un nombre d'opérations arithmétiques de l'ordre de $K_d M \log(M)$, dans laquelle la constante $K_d \in [2d, D^d]$ dépend du type de voisinage discret qui est considéré (et D est le diamètre maximal des voisinages). Pour atteindre efficacement cette limite de complexité, il convient d'adapter une structure de données particulière. Pour la méthode classique de fast-marching, les données des nœuds sont normalement stockées à l'aide de deux types de structures de données : Une matrice complète à $d-$dimension (ou tenseur), qui contient les informations de l'ensemble de la grille de discrétisation et les fonctions de valeur calculées à chaque étape ; Une liste chaînée dynamique, qui contient les informations des nœuds de la bande étroite utilisant la fonction de valeur. Nous donnons plus de détails sur la structure de données pour la méthode de marche rapide dans Chapter 4, où nous introduisons une nouvelle structure de données pour notre méthode, dans laquelle la grille est construite dynamiquement autour de la trajectoire optimale.

2.1.4 Méthodes numériques Max-plus

Comme indiqué précédemment, le schéma aux différences finies et le schéma semi-lagrangien entièrement discrétisé conduisent tous deux à des équations qui peuvent être interprétées comme des équations de programmation dynamique pour des problèmes de contrôle optimal stochas-

tique avec des espaces de temps et d'état discrets. Plus récemment, des schémas de discrétisation basés sur le max-plus ont été développés pour résoudre les équations HJB du premier ordre. Ces schémas reposent sur une représentation de la fonction de valeur discrétisée sur une base max-plus, ce qui conduit à un problème de contrôle optimal déterministe en temps discret. D'une manière générale, ces méthodes tirent parti de la linéarité *max-plus* du semigroupe d'évolution de l'EDP HJB, le soi-disant *semigroupe de Lax-Oleinik*. Après une discrétisation temporelle, cela permet d'approximer la fonction de valeur pour un horizon temporel donné, par une somme supérieure de "fonctions de base" appropriées, par exemple des formes quadratiques. Ces suprêmes sont propagés par l'action du semigroupe de Lax-Oleinik entre deux pas de temps successifs. En particulier, dans les travaux de Fleming et McEneaney [FM00], il est démontré que toute fonction semi-convexe peut être représentée comme la combinaison linéaire max-plus de fonctions quadratiques centrées sur un sous-ensemble dense et dénombrable. Ensuite, en supposant que la fonction de valeur est semi-convexe, ils en font une approximation à un horizon donné en utilisant la combinaison linéaire max-plus de fonctions quadratiques. La propagation entre deux pas de temps successifs est effectuée en appliquant un opérateur linéaire max-plus sur les coefficients de la combinaison linéaire max-plus. Cette approximation peut être interprétée comme une équation de programmation dynamique pour un problème de contrôle optimal déterministe à temps discret.

Dans [AGL08], une forme similaire d'approximation pour la fonction de valeur a été proposée. Pour établir les équations récursives des coefficients scalaires, les auteurs ont introduit une famille de "fonctions de test". Le calcul itératif de ces coefficients scalaires dans ce cas implique l'application d'un opérateur non linéaire. Cet opérateur peut être considéré comme une projection sur l'espace des fonctions de base, puis une projection sur l'espace des fonctions de test. Ce schéma d'approximation peut être interprété comme une équation de programmation dynamique d'un jeu déterministe à somme nulle à deux joueurs. En outre, les auteurs fournissent un résultat de convergence avec des estimations d'erreur détaillées et démontrent que les complexités de calcul de leurs approches restent comparables à celles des méthodes classiques basées sur une grille. Dans Chapter 6, nous combinerons ces approximations max-plus avec des méthodes directes, ce qui conduira à un degré de précision plus élevé.

Une façon de surmonter la malédiction de la dimensionnalité est de supposer une certaine structure du problème de contrôle. C'est en particulier ce qui a été proposé par la méthode de McEneaney sans malédiction de la dimensionnalité dans [McE07]. Dans ce travail, McEneaney considère problèmes de contrôle optimal avec commutation à horizon infini, pour lesquels l'hamiltonien est exprimé comme un maximum d'un nombre fini d'hamiltoniens "plus simples". Chacun de ces hamiltoniens est une forme linéaire quadratique issue d'un problème de contrôle optimal linéaire quadratique. L'auteur démontre que la complexité présente une croissance cubique en dimension (de l'état) (voir aussi [McE09]). Cette complexité est toutefois limitée par un nombre exponentiel dans le nombre de pas de temps, que l'on appelle le "raccourcissement de la complexité". Plusieurs méthodes d'"élagage" sont alors proposées pour améliorer cette limite de complexité, par exemple dans [GMQ11; Qu14b].

2.1.5 Concentration sur les trajectoires optimales

Un autre moyen efficace de surmonter la malédiction de la dimensionnalité consiste à remplacer le problème général de la résolution de l'équation HJ et de l'approximation de la fonction de valeur dans l'ensemble de l'espace d'état par le calcul d'une ou de plusieurs trajectoires optimales avec un état initial fixe. Ce dernier problème peut être résolu, sous certaines hypothèses de convexité, par l'approche du principe du maximum de Pontryagin [RZ98; RZ99; Tré05], ou par des méthodes directes[Tré05; Bon+06]. Dans la même inspiration, en temps discret, on peut utiliser la méthode de programmation dynamique duale stochastique (SDDP), qui a été

introduite pour la première fois dans [PP91]. Il est conçu pour résoudre des problèmes de contrôle déterministes ou stochastiques avec une structure spécifique où les coûts sont conjointement convexes par rapport à l'état et au contrôle, dans le cas de la minimisation, et où la dynamique est linéaire par rapport à l'état et au contrôle. Cette structure spéciale garantit que la fonction de valeur est convexe à chaque horizon temporel. Ainsi, la fonction de valeur est approximée par un supremum fini de cartes affines (c'est-à-dire une carte convexe affine par morceaux), et la fonction de valeur approximée, ainsi que les trajectoires optimales à partir de l'état initial fixe, peuvent alors être calculées efficacement à l'aide de solveurs de programmation linéaire. Nous nous référons à [Sha11; GLP15] pour la convergence du SDDP. Dans les cas où les hypothèses sur les coûts et la dynamique ne sont pas satisfaites, c'est-à-dire en l'absence de convexité, la méthode SDDP ne conduit généralement qu'à un optimum local. Dans Chapter 7, nous développons une nouvelle méthode numérique, qui peut être considérée comme une extension de la méthode SDDP aux problèmes semi-concaves.

Plus récemment, d'autres méthodes consistent à exploiter la structure du problème, en particulier pour réduire l'ensemble des trajectoires possibles parmi lesquelles l'optimisation est effectuée. Par exemple, dans [AFS19; AFS20], les auteurs ont introduit une discrétisation structurée en arbre, en tirant parti de la continuité Lipschitz de la fonction de valeur. Dans [BGZ22], les auteurs ont introduit une discrétisation adaptative dans l'espace de contrôle, qui s'est avérée efficace lorsque la dimension de l'espace de contrôle est faible.

La concentration sur les trajectoires optimales est l'idée initiale de nos travaux dans Chapter 4, Chapter 6 et Chapter 7.

2.1.6 Développement récent des méthodes numériques

Le développement de méthodes numériques efficaces pour résoudre les équations HJB reste un sujet de recherche important, en particulier avec plusieurs études récentes visant à surmonter la malédiction de la dimensionnalité. Une méthode récente, introduite dans [DO16], est basée sur la formule de Hopf. Cette méthode se concentre sur l'équation HJ dont l'hamiltonien ne dépend que du gradient de la fonction de valeur. Au lieu de la discrétisation, les auteurs proposent une méthode pour résoudre l'équation HJ en combinant la formule de Hopf avec l'approche itérative de Bregman divisée ([GO09]). Les auteurs montrent également que leur méthode a une limite de complexité qui est polynomiale dans la dimension. Les techniques d'apprentissage profond et de réseaux neuronaux sont également appliquées pour résoudre l'équation HJB et pour trouver une loi de contrôle par rétroaction, par exemple dans [Kan+21; DDM23; BPW23].

D'autres méthodes récentes sont basées sur la décomposition tensorielle. Parmi elles, on peut citer les l'approximation de l'équation HJB en utilisant l'approximation du produit tensoriel hiérarchique de faible rang avec la méthode de Monte-Carlo proposée dans [OSS22]. En outre, dans [DKK21], les auteurs ont introduit une approximation de train tensoriel pour la fonction de valeur du problème de contrôle. Le système non linéaire résultant est ensuite résolu à l'aide d'une méthode itérative de Newton. Le développement d'un analogue tropical de l'approximation tensorielle de faible rang pour la fonction de valeur afin de résoudre l'équation HJB est l'une des motivations des études présentées dans Chapter 7.

2.2 Contributions

Dans cette thèse, nous développons de nouvelles méthodes numériques pour résoudre les problèmes de contrôle optimal déterministe et les équations de Hamilton-Jacobi-Bellman du premier ordre associées. En outre, nous analysons la convergence, la complexité de calcul et les propriétés de régularité de ces méthodes. Notre objectif principal est de lutter contre la malédiction de la

dimensionnalité et de l'atténuer. Une idée commune pour relever ce défi est de se concentrer sur l'identification d'une ou plusieurs trajectoires optimales avec des conditions initiales et/ou finales fixes.

2.2.1 Résumé et organisation

- Dans Chapter 3, nous donnons des informations essentielles sur la théorie du contrôle optimal déterministe et les méthodes numériques qui seront utilisées dans le reste de cette thèse.
- Dans Chapter 4, nous nous concentrons sur un problème particulier et fondamental, le problème du temps minimum. Nous présentons notre nouvel algorithme, qui tient compte à la fois des aspects continus et de l'approximation numérique. Nous montrons la convergence et nous établissons une limite de complexité de calcul avec une certaine limite d'erreur. Nous présentons des tests numériques jusqu'à la dimension 7, confirmant la rapidité de la méthode.
- Dans Chapter 5, nous analysons un schéma semi-lagrangien particulier pour le problème du temps minimum et l'équation eikonale associée. Nous prouvons une propriété de régularité pour la fonction de valeur discrétisée. Nous établissons le taux de convergence du schéma semi-discrétisé et du schéma entièrement discrétisé. En particulier, nous appliquons le résultat pour dériver une condition suffisante pour la limite de complexité idéale de Chapter 4.
- Dans Chapter 6, nous considérons des problèmes généraux de contrôle optimal déterministe à horizon fini. Tout d'abord, nous combinons les méthodes directes avec la méthode des éléments finis max-plus, ce qui permet d'obtenir un algorithme plus précis. Ensuite, nous adaptons l'idée de Chapter 4 à ce contexte, ce qui permet d'obtenir la limite de complexité idéale avec une condition assouplie.
- Dans Chapter 7, nous introduisons une nouvelle méthode pour approximer le problème avec un état initial fixe. Cette méthode s'inspire et peut être considérée comme une généralisation de l'algorithme de programmation dynamique duale (stochastique) adapté aux problèmes semiconcaves. Nous montrons que notre méthode converge vers l'optimum global sous certaines hypothèses de régularité. Nous présentons des benchmarks numériques sur des problèmes à $N-$corps.

2.2.2 Contribution des chapitres

Nous présentons maintenant en détail les chapitres qui contiennent des travaux originaux et qui constituent les contributions de cette thèse.

2.2.2.1 Contribution de Chapter 4: A multilevel Fast Marching Method For the Minimum Time Problem

Dans Chapter 4, nous introduisons un nouvel algorithme pour approximer les solutions d'une classe d'EDP de Hamilton-Jacobi stationnaires découlant de problèmes de contrôle optimal en temps minimum. En particulier, nous nous concentrons sur la recherche du temps de parcours minimum entre deux ensembles donnés $\mathcal{K}_{\text{src}}$ et $\mathcal{K}_{\text{dst}}$ dans un domaine donné Ω, ainsi que sur les trajectoires optimales.

À cette fin, nous traitons deux problèmes, l'un impliquant la direction temporelle habituelle appelée "arrivée à $\mathcal{K}_{\text{dst}}$" et l'autre impliquant la direction inverse appelée "départ de $\mathcal{K}_{\text{src}}$". Nous

caractérisons les fonctions de valeur $v_{\mathrm{s}\to}, v_{\to\mathrm{d}}$ de ces deux problèmes en utilisant deux équations HJB contraintes par l'état dans leurs directions respectives. Nous caractérisons ensuite les points géodésiques en utilisant $v_{\mathrm{s}\to}$ et $v_{\to\mathrm{d}}$, dans Proposition 4.3.3 et Lemma 4.3.5. De plus, sur la base de ces deux fonctions de valeur, nous définissons un sous-domaine ouvert $\mathcal{O}_\eta$ de Ω,

$$\mathcal{O}_\eta = \{x \in (\Omega \setminus (\mathcal{K}_{\mathrm{src}} \cup \mathcal{K}_{\mathrm{dst}})) \mid \mathcal{F}_v(x) < \inf_{y\in\Omega}\{\mathcal{F}_v(y) + \eta\} \} \ , \tag{2.2}$$

où $\mathcal{F}_v(x) = v_{\mathrm{s}\to}(x) + v_{\to\mathrm{d}}(x) - v_{\mathrm{s}\to}(x)v_{\to\mathrm{d}}(x)$. Nous montrons l'équivalence entre le sous-domaine $\mathcal{O}_\eta$ et les points géodésiques $\delta-$ dans Lemma 4.3.14 et Lemma 4.3.15. Sur la base de ces propriétés, nous établissons dans Theorem 4.3.16 que, si nous réduisons l'espace d'état de Ω à $\mathcal{O}_\eta$, alors pour chaque x dans l'ensemble des points $\delta-$géodésiques avec $\delta < \eta$, les nouvelles fonctions de valeur $v^\eta_{\mathrm{s}\to}(x)$ et $v^\eta_{\to\mathrm{d}}(x)$ sont égales à $v_{\mathrm{s}\to}(x)$ et $v_{\to\mathrm{d}}(x)$, respectivement.

Notre nouvel algorithme tire parti des propriétés susmentionnées. Nous nous appuyons sur des approximations de grilles imbriquées et recherchons les trajectoires optimales en utilisant les approximations de grilles grossières pour réduire l'espace de recherche dans les grilles fines. Plus précisément, après une approximation grossière dans la grille grossière, deux fonctions de valeur approximative $v^H_{\mathrm{s}\to}, v^H_{\to\mathrm{d}}$ sont calculées dans les nœuds de la grille. Nous sélectionnons ensuite les nœuds *actifs* à l'aide d'une formule d'approximation dérivée de (2.2), où v est remplacé par v^H et η par un paramètre η_H. Les nœuds actifs peuvent être considérés comme des points de voisinage autour de la trajectoire optimale. La grille fine est construite dynamiquement "autour" de ces nœuds actifs. Ensuite, les calculs des fonctions de valeur approximative $v^h_{\mathrm{s}\to}, v^h_{\to\mathrm{d}}$ ne sont effectués que sur les nœuds de la grille fine sélectionnés. Nous donnons dans Algorithm 4.2 les détails de la méthode à deux niveaux (2LFMM). Ensuite, nous prouvons dans Theorem 4.4.3 la convergence de la méthode de marche rapide à deux niveaux, en affirmant que si η_H est suffisamment grand, l'estimation de l'erreur est aussi bonne que celle obtenue en discrétisant directement l'ensemble du domaine avec la grille fine. Le concept d'approximation grossière-fine peut être étendu au cas multi-niveaux. Étant donné une famille de maillages successifs $H_1 \leqslant H_2 \ldots \leqslant H_N = h$, et une famille de paramètres positifs réels $\{\eta_1, \eta_2, \ldots, \eta_{N-1}\}$, la méthode de marche rapide multi-niveaux (MLFMM) est présentée dans Algorithm 4.3, et le résultat de la convergence est présenté dans Theorem 4.4.4.

Nous fournissons une limite de complexité de calcul par rapport à l'erreur ε, afin de montrer l'amélioration de notre algorithme par rapport à d'autres méthodes basées sur la grille. Pour commencer, nous limitons la complexité spatiale de 2LFMM par le volume des voisinages tubulaires autour des trajectoires optimales dans Proposition 4.5.2. Plus précisément, étant donné un pas de maille grossière H, un pas de maille fine h et le paramètre η_H, la complexité spatiale peut être exprimée comme suit

$$\mathcal{C}_{spa}(H,h) = \widetilde{O}\Big(C^d\Big(\frac{1}{H^d} + \frac{(\eta_H)^{\beta(d-1)}}{h^d}\Big)\Big) \ , \tag{2.3}$$

où β est la "rigidité" de la fonction de valeur autour des trajectoires optimales, voir Assumption (A6). Cette complexité spatiale implique en effet des nœuds dans la grille grossière, inclus dans une boule d'un volume de $O(1)$, et des nœuds dans la grille fine, inclus dans un voisinage tubulaire de la trajectoire optimale d'un volume de $O((\eta_H)^{\beta(d-1)})$. La même analyse que dans le cas à deux niveaux fonctionne également pour le cas à $N-$niveaux, pour lequel la complexité spatiale est déduite d'une hiérarchie de voisinage tubulaire. On peut observer que la limite de complexité dans (2.3), ainsi que celle pour le cas $N-$level, est une fonction de la famille des pas de maille grossiers lorsque h est fixé. Ensuite, les pas de maille sont choisis de manière à ce que la limite de complexité atteigne son minimum. De plus, une fois les pas de maille choisis, la complexité devient une fonction du nombre de niveaux N. Ainsi, N est à nouveau choisi pour

minimiser la complexité. Nous présentons dans Theorem 4.5.5 le résultat principal de cette limite de complexité. En résumé, pour notre méthode à plusieurs niveaux, le nombre d'opérations arithmétiques est de l'ordre de $\widetilde{O}(C^d\varepsilon^{-\frac{1+(d-1)(1-\gamma\beta)}{\gamma}})$, où $C > 1$ est une constante dépendant des caractéristiques du problème et $0 < \gamma \leqslant 1$ est le taux de convergence de la méthode classique de marche rapide. Ainsi, en considérant la dépendance dans ε seulement, nous réduisons la limite de complexité de $\widetilde{O}(\varepsilon^{-\frac{d}{\gamma}})$ à $\widetilde{O}(\varepsilon^{-\frac{1+(d-1)(1-\gamma\beta)}{\gamma}})$. Dans les situations typiques où la fonction de valeur est lisse avec un hessien non dégénéré dans le voisinage d'une trajectoire optimale, on a $\beta = 1/2$. Dans des cas exceptionnels, avec une géométrie de type L_1-, on peut obtenir $\beta = 1$. Observez que la limite de complexité se réduit à $\widetilde{O}(C^d\varepsilon^{-1})$ lorsque $\gamma = \beta = 1$. Ainsi, en considérant uniquement la dépendance de ε, la limite de complexité devient $\widetilde{O}(\varepsilon^{-1})$ et est donc du même ordre que pour les problèmes unidimensionnels.

Pour mettre en œuvre numériquement l'algorithme, nous introduisons une structure de données dédiée, une "table de hachage", pour stocker les grilles successives contraintes (construites dynamiquement), pour chaque niveau. Chaque fois qu'un nouveau nœud est sélectionné, nous calculons son emplacement par une fonction de hachage, puis nous stockons les informations nécessaires au calcul dans cet emplacement, voir les détails dans Section 4.4.4. Nous présentons des tests numériques jusqu'à la dimension 7, et nous analysons la complexité effective de notre algorithme multiniveau.

2.2.2.2 Contribution de Chapter 5: Convergence and Error Estimates of a Semi-Lagrangian Scheme for the Eikonal Equation

Dans Chapter 5, nous considérons un schéma semi-lagrangien particulier pour résoudre numériquement l'équation d'Eikonal découlant du problème du temps minimum pour atteindre un ensemble cible $\mathcal{K}$, dans lequel le pas de temps varie en fonction de l'état. L'objectif est d'établir une condition suffisante pour atteindre un taux de convergence d'ordre 1 à la fois pour le schéma semi-discrétisé et le schéma entièrement discrétisé, où dans le cas du schéma semi-discrétisé, le taux de convergence est exprimé en termes de pas de temps, tandis que pour le dernier, il est en termes de pas de temps et de pas de maille. Ce résultat est également appliqué pour dériver des conditions suffisantes pour atteindre un taux de convergence de 1 pour la méthode de marche rapide de type semi-lagrangien. A notre connaissance, c'est la première fois qu'un tel taux de convergence est établi. Nous appliquons ensuite ce résultat pour obtenir la limite de complexité idéale de la méthode proposée dans Chapter 4.

Nous considérons d'abord le schéma semi-discrétisé. Nous représentons la solution du système discrétisé comme la fonction de valeur d'un problème de contrôle optimal déterministe en temps discret. Le premier résultat principal est présenté dans Proposition 5.3.4, dans lequel nous montrons que sous des hypothèses semiconcaves particulières sur la dynamique et sur la fonction de distance à l'ensemble cible (voir les détails dans Assumption (A8)), nous obtenons que la fonction de valeur en temps discret est semiconcave, c'est-à-dire

$$v^h(x+z) - 2v^h(x) + v^h(x-z) \leqslant C\|z\|^2, \quad \text{for every } x, z \in \mathbb{R}^d \setminus \mathcal{K} \,. \tag{2.4}$$

Esquisse de la preuve de Proposition 5.3.4. Nous dérivons cette propriété en considérant les problèmes de contrôle optimal en temps discret avec les états initiaux $x+z$, $x-z$ et x, respectivement. Soit y^* la trajectoire optimale pour le problème avec l'état initial x. Nous construisons des trajectoires réalisables, désignées par y^+ et y^- pour les problèmes avec des états initiaux $x+z$ et $x-z$, de la manière suivante : y^+ et y^- suivent initialement la même trajectoire de contrôle que pour y^* ; si y^* atteint d'abord $\mathcal{K}$, alors y^+ et y^- continuent le long d'une ligne droite jusqu'à $\mathcal{K}$. Sinon, si y^- atteint $\mathcal{K}$ en premier, alors y^+ et y^- continuent en ligne droite jusqu'à $\mathcal{K}$, y^+ répète deux fois le contrôle de la trajectoire y^* jusqu'à ce que y^* atteigne $\mathcal{K}$.

Ensuite, y^+ suit une trajectoire en ligne droite. (2.4) est ensuite déduite en calculant les coûts des trajectoires y^-, y^+ et y^*, respectivement. □

Pour le taux de convergence, un côté, à savoir $v - v^h$, est plus facile à borner par le haut puisque l'ensemble discret des trajectoires est un sous-ensemble de l'ensemble continu. Dans l'autre sens, la limitation de $v^h - v$ fait appel à des techniques de viscosité et à la semiconcavité de v^h. Nous présentons ce résultat comme le deuxième résultat principal dans Theorem 5.3.5.

Nous considérons ensuite le schéma entièrement discrétisé, et nous commençons par considérer un simple opérateur d'interpolation P_1 (linéaire par morceaux) I_1. Notre objectif est d'établir une borne supérieure sur $\|w^h - v^h\|_\infty$, où w^h est la solution du schéma entièrement discrétisé. Pour la borne supérieure de $w^h - v^h$, nous utilisons le fait que l'opérateur d'interpolation et l'opérateur de Bellman associés au problème de contrôle déterministe en temps discret ne sont pas coûteux, l'opérateur de Bellman présentant un taux de contraction de $(1 - \frac{h}{f})$. De plus, lorsque v^h présente une semiconcavité, le supremum sur x de $(I_1[v^h] - v^h)(x)$ est borné par Ch^2. Pour la borne supérieure de $v^h - w^h$, nous représentons d'abord la solution du système entièrement discrétisé comme la fonction de valeur d'un problème de contrôle stochastique. Ensuite, nous montrons que, sous des hypothèses semi-convexes sur la dynamique et sur la fonction de distance à l'ensemble cible (voir les détails dans Assumption (A9)), nous pouvons dériver une erreur de l'ordre de h est. Ce résultat est présenté dans Proposition 5.4.4, qui se résume comme suit,

$$\sup_{x \in \mathbb{R}^d} (v^h - w^h)(x) \leqslant Ch \ . \tag{2.5}$$

Sketch of Proof. Nous considérons une chaîne de Markov contrôlée avec un état initial x. Pour toute stratégie σ^h, nous construisons une trajectoire déterministe qui suit le même contrôle que celui associé à σ^h dans le cas stochastique. Cette trajectoire est en effet une trajectoire réalisable pour le système discret déterministe. Nous dérivons ensuite (1.5) en calculant les coûts du problème de contrôle stochastique avec la stratégie σ^h, et du problème déterministe avec la trajectoire construite, qui est principalement basée sur deux propriétés : (i) les états du processus de Markov ont la propriété que l'espérance de $\xi_{k+1} - \xi_k$ est $h\alpha$, et la covariance est limitée par h^2 (voir (5.69)) ; (ii) une propriété liée à l'espérance des fonctions semi-convexes (voir Lemma 5.4.3). □

Nous appliquons ensuite le résultat de convergence du schéma entièrement discrétisé pour montrer que les méthodes de marche rapide, utilisant des opérateurs de mise à jour dérivés d'une discrétisation de type semi-lagrangien, ont un taux de convergence d'ordre h sous les hypothèses que nous avons introduites, où h représente la grille de maillage. En conséquence, la complexité de calcul de la méthode de marche rapide à plusieurs niveaux introduite dans Chapter 4 dépend uniquement de β, la rigidité de la trajectoire optimale.

2.2.2.3 Contribution de Chapter 6: An Adaptive Multi-Level Max-Plus Method for Deterministic Optimal Control Problems

Dans Chapter 6, nous considérons des problèmes de contrôle optimal déterministe à horizon fini qui impliquent à la fois des coûts initiaux et finaux. Tout d'abord, nous combinons les approximations max-plus avec des méthodes directes, ce qui conduit à une méthode numérique avec un degré de précision plus élevé. Ensuite, nous étendons l'idée du raffinement dynamique de la grille autour du voisinage tubulaire des trajectoires optimales, qui a été introduite dans Chapter 4.

Nous caractérisons les conditions d'optimalité en considérant une paire d'EDP HJB associées à deux problèmes de contrôle optimal : l'un impliquant une dynamique vers l'avant avec un état initial fixe et un état final libre, et l'autre impliquant une dynamique vers l'arrière avec un état final fixe et un état initial libre. Nous adaptons la méthode des éléments finis max-plus pour approximer les deux fonctions de valeur $v^t_{\mathrm{s}\to}$ et $v^t_{\to\mathrm{d}}$, pour chaque $t \in \{0, \delta, \dots, T\}$. Plus

précisément, si l'on considère d'abord $v^t_{\to \mathrm{d}}$, et compte tenu d'une famille finie de fonctions de base $\{w_i\}_{1\leqslant i\leqslant p}$, $v^t_{\to \mathrm{d}}$ est approximé par une combinaison linéaire max-plus des fonctions de base à coefficients $\{\lambda_i^{\to \mathrm{d},t}\}_{1\leqslant i\leqslant p}$. les fonctions de base à coefficients $\{\lambda_i^{\to \mathrm{d},t}\}_{1\leqslant i\leqslant p}$. L'équation récursive des scalaires entre deux pas de temps successifs est obtenue en introduisant un ensemble de fonctions de test $\{z_j\}_{1\leqslant j\leqslant q}$ (voir les détails dans Proposition 6.3.1). Notre premier travail est basé sur l'observation que la propagation en petit temps des fonctions de base conduit à un nouveau problème de contrôle optimal,

$$\langle z_j, S^\delta[w_i]\rangle = \max\left\{ z_j(x(0)) + \int_0^\delta \ell(x(s),u(s))ds + w_i(x(\delta))\right\} . \tag{2.6}$$

Nous montrons dans Proposition 6.3.2 et Lemma 6.3.3 que, sous certaines hypothèses de régularité sur f et ℓ (voir les détails dans Assumption (A11)), en choisissant des fonctions de base fortement concaves et des fonctions de test w_i, z_j (par exemple des fonctions quadratiques), dans un horizon temporel donné $\delta \leqslant \bar{\delta}$, le problème (1.6) est en fait un programme concave par rapport à la trajectoire $(x(\cdot), u(\cdot))$. Cette propriété peut s'expliquer par la "propagation" de la forte concavité des coûts initiaux et/ou terminaux dans (1.6) sur un petit horizon temporel. Cela implique que le problème (1.6) peut être résolu exactement, ou avec une erreur négligeable par rapport à δ, en employant une méthode directe. Nous proposons d'approximer ce problème par une méthode directe. L'algorithme complet est présenté dans Algorithm 6.1. L'estimation de l'erreur est également présentée dans Theorem 6.3.4, qui est alors une conséquence directe des résultats de [Lak07].

Après avoir obtenu l'approximation des deux fonctions de valeur, nous appliquons une approche similaire à celle utilisée dans Chapter 4 pour approximer la trajectoire optimale. Dans ce cas, la valeur du problème est représentée (approximativement) par les scalaires $\lambda^t_{\mathrm{s}\to}$, $\lambda^t_{\to \mathrm{d}}$ dans deux directions, c'est-à-dire

$$v^* \approx \sup_{1\leqslant i,j\leqslant p} \left\{ \lambda_i^{\mathrm{s}\to,t} + \lambda_j^{\to \mathrm{d},t} + \langle w_i^{\mathrm{s}\to}, w_j^{\to \mathrm{d}}\rangle \right\}, \ \forall\ t\in[0,T] . \tag{2.7}$$

Nous sélectionnons ensuite des ensembles d'indices i et j qui sont $\eta-$optimaux dans (1.7). Ces indices correspondent en effet aux voisinages dans $\mathbb{R}^d$ de certaines trajectoires optimales duales pour $v^t_{\mathrm{s}\to}$ et $v^t_{\to \mathrm{d}}$, respectivement. Les trajectoires optimales (primaires) peuvent alors être identifiées sur la base de ces ensembles d'indices, voir Theorem 6.4.7.

Nous étendons ensuite l'idée du raffinement dynamique de la grille autour du voisinage tubulaire des trajectoires optimales dans Chapter 4. En particulier, nous utilisons une hiérarchie de grilles irrégulières de plus en plus fines pour générer les fonctions de base et les fonctions de test. Dans le cas à deux niveaux, nous utilisons d'abord deux grilles grossières pour générer les fonctions de base et les fonctions d'essai pour l'approximation des fonctions de valeur dans deux directions. Ensuite, étant donné un paramètre η^H, nous identifions les nœuds "actifs" dans les grilles grossières pour les deux directions. Ces noeuds actifs correspondent en effet aux indices (i,j) qui sont η^H-optimaux dans (2.7). L'approximation grossière des trajectoires optimales est obtenue à l'aide de ces nœuds actifs. comme dans (6.54). Ensuite, nous construisons les grilles fines autour des nœuds actifs. (voir dans (6.56)). Les fonctions de base et les fonctions de test pour l'approximation fine seront générées par ces grilles. Nous présentons la méthode complète à deux niveaux dans Algorithm 6.2. Le concept d'approximation grossière et fine peut être étendu au cas multi-niveaux. Étant donné une famille de grilles successives $\{\hat{G}^{H_l}, G^{H_l}\}_{1\leqslant l\leqslant m}$, et une famille de paramètres positifs réels $\{\eta_l\}_{1\leqslant l\leqslant m-1}$, la méthode d'approximation adaptative Max-Plus à m-niveaux est présentée dans Algorithm 6.3.

Nous montrons qu'en utilisant notre algorithme, le nombre de fonctions de base nécessaires pour obtenir une certaine erreur ε est considérablement réduit. En effet, pour un problème à

$d-$dimension, sous certaines hypothèses de régularité, nous obtenons une limite de complexité de $C^d(1/\varepsilon)^{\frac{1}{2}}$ opérations arithmétiques, pour une certaine constante $C > 1$. Ceci doit être comparé aux méthodes basées sur des grilles régulières, qui donnent des limites de complexité d'ordre $O(1/\varepsilon^{ad})$ dans lesquelles $a > 0$ dépend des hypothèses de régularité et de l'ordre du schéma. Avec notre méthode adaptative, la malédiction de la dimensionnalité n'est présente que dans le terme C^d. Nous présentons le principal résultat de complexité dans Theorem 6.6.4. Par rapport à la complexité de calcul dans Chapter 4, l'utilisation d'approximations max-plus combinées à des méthodes directes permet d'obtenir un degré de précision plus élevé. En effet, sous des hypothèses de régularité appropriées (en particulier les hypothèses que nous avons utilisées dans Chapter 5 pour obtenir un taux de convergence d'ordre h pour la méthode de marche rapide), la méthode de Chapter 4 présente une complexité de calcul d'ordre $\mathcal{O}(\varepsilon^{-1-(d-1)(1-\beta)})$, dans laquelle le paramètre $0 < \beta \leqslant 1$ mesure la "rigidité" des fonctions de valeur près des trajectoires optimales. Les cas typiques sont modérément rigides et ont un paramètre $\beta = 1/2$, ce qui conduit à une complexité d'ordre $\mathcal{O}(\varepsilon^{-1-(d-1)/2})$. En revanche, nous obtenons ici une complexité d'ordre $\mathcal{O}(\varepsilon^{-\frac{1}{2}})$, avec des hypothèses moins exigeantes.

2.2.2.4 Contribution de Chapter 7: Semiconcave Dual Dynamic Programming and Its Application to Tropical Low-Rank Approximation of N-body System

Dans Chapter 7, nous introduisons un nouvel algorithme pour trouver numériquement la fonction de valeur, ainsi que la trajectoire optimale, pour une classe de problèmes de contrôle optimal déterministe à horizon fini avec un état initial fixe. En particulier, la fonction de récompense (dans le cas de la maximisation) doit seulement être semiconcave par rapport à l'état x.

Nous cherchons une approximation serrée de la fonction de valeur le long des trajectoires optimales à partir d'un point initial donné x_0. Nous commençons par une trajectoire réalisable (arbitraire) pour le problème de contrôle, et construisons une approximation supérieure initiale pour la fonction de valeur. A chaque pas d'itération, pour le problème de maximisation, après une discrétisation dans le temps, nous approximons la fonction de valeur, dans un horizon temporel donné, par un minimum de fonctions quadratiques "de base" (voir la définition dans (7.14)). Cependant, le semigroupe évolutionnaire associé au problème de maximisation est max-plus linéaire au lieu de min-plus linéaire. Pour propager les fonctions de base à l'horizon temporel suivant, nous résolvons un problème dual du problème de maximisation (voir la formulation du problème dual dans (7.19)). Nous construisons ensuite une nouvelle approximation supérieure de la fonction de valeur basée sur le problème dual. La trajectoire est alors ensuite mise à jour vers une trajectoire optimale dérivée de la fonction de valeur approximative actuelle. Ainsi, à chaque itération, nous ajoutons une fonction de base supplémentaire pour l'approximation. Nous présentons notre nouvel algorithme dans Algorithm 7.1.

Nous présentons également une légère variante de l'algorithme dans Algorithm 7.2, qui implique deux boucles dans le temps : une boucle en arrière dans le temps pour la mise à jour de la fonction de valeur approximative, et une boucle en avant dans le temps pour la mise à jour de la trajectoire. Nous montrons que notre algorithme peut être comparé à la méthode (S)DDP et peut être considéré comme une extension de celle-ci, en particulier pour traiter les situations impliquant la condition de semiconcavité sur la récompense courante. En effet, il peut être expliqué par l'ajout d'un terme quadratique de "régularisation" pour la semiconcavité de la fonction de valeur et des récompenses en cours d'exécution. En suivant l'approche de la méthode SDDP, à chaque itération, nous résolvons en effet un nouveau problème dual sous la forme de (7.31). Nous devrions démontrer l'équivalence de cet algorithme avec celui du point de vue de l'approximation dans Proposition 7.3.6.

Nous montrons que notre méthode converge vers le maximum global sous certaines hypothèses de régularité. Ceci est basé sur la propriété que la propagation en petit temps préserve

la semiconcavité, que nous présentons dans Proposition 7.4.4. De plus, en dénotant $v_m^{t,h}$ la fonction de valeur approximative obtenue par notre algorithme au pas d'itération m, nous établissons dans Proposition 7.4.7 que cette approximation est monotone par rapport au pas d'itération, et qu'elle est bornée par le haut et par le bas. Nous présentons le résultat de convergence dans Theorem 7.4.9.

En guise d'application, nous utilisons notre algorithme pour construire une approximation tensorielle tropicale de faible rang, qui peut être considérée comme un analogue tropical de la décomposition tensorielle classique de faible rang (voir in (7.59)), pour un système à N corps. La fonctionnelle d'action de ce système se compose de l'énergie potentielle et de l'énergie cinétique individuelles, ainsi que de l'énergie d'interaction de Coulomb. Nous interprétons ce système dans le cadre du contrôle optimal et de l'équation de Hamilton-Jacobi, sur la base du principe de moindre action (voir in (7.61)). Nous présentons des repères numériques pour déterminer la trajectoire optimale et le grand état de chaque individu dans Section 7.5.4.

Chapter 3

Preliminaries

***

In this preliminary chapter, we introduce four main concepts - deterministic optimal control, the Semi-Lagrangian scheme for HJB equations, the fast marching method and tropical-based numerical methods. The first three concepts form the foundation of part I of this thesis, the last concept is for the part II of this thesis.

3.1 Optimal Control Problem and Hamilton-Jacobi-Bellman Equation

The purpose of this section is to provide a concise introduction to *deterministic optimal control* problems, and it's relevance with the concept of *viscosity solution* of a class of nonlinear partial

differential equations of the form

$$F(x, v(x), \nabla v(x)) = 0 \ . \tag{3.1}$$

This equation is commonly known as the *Hamilton-Jacobi-Bellman* equation, and is derived from the *dynamic programming principle*, which was first formulated by Richard Bellman in 1950's. The material of this chapter is mainly based on the reference books of Fleming and Soner [FS06], and of Bardi and Capuzzo-Dolcetta [BC08].

3.1.1 Deterministic Optimal Control Problem

An optimal control problem can be described as finding an optimal strategy $u(\cdot) \in \mathcal{U}$, where $\mathcal{U} = \{u : [0, +\infty) \to U \subseteq \mathbb{R}^m$ such that $u(\cdot)$ is measurable$\}$, that optimize a certain objective functional, where the objective functional depends on the strategy $u(\cdot)$ and the states of the system $y(\cdot)$, and the state $y : [0, +\infty) \to \Omega \subseteq \mathbb{R}^d$ is governed by a dynamical system. We call U the control space and Ω the state space. We consider the following deterministic optimal control problems, in which the evolution of the system is determined by an ordinary differential equation,

$$\begin{cases} \dot{y}(s) = f(y(s), u(s)), \quad \forall\ s > 0 \ , \\ y(0) = x \ , \end{cases} \tag{3.2}$$

where $f : \Omega \times \mathcal{U} \to \mathbb{R}^d$ is called the dynamics. Let us denote by $y_u(x; s)$ the solution of (3.2). In this chapter, we consider the infinite horizon discounted control problem, with a *discount rate* $\lambda > 0$. More precisely, the objective functional has the form

$$J(x, u(\cdot)) := \int_0^\infty e^{-\lambda s} \ell(y_u(x; s), u(s)) ds \ , \tag{3.3}$$

and we consider a minimization problem, i.e., our objective is to minimize $J(x, u(\cdot))$ over all $u(\cdot) \in \mathcal{U}$. In that case we call ℓ the *running cost*. Notice that the objective functional J may have a different form. For instance, the objective functional can be replaced by the integral until a time horizon τ, at which the controlled state $y_u(x; \tau)$ first reaches a certain target $\mathcal{K}$. This is the one we explore in Chapter 4 in the particular case of a minimum time problem, such that $\ell \equiv 1$. Another variant involves a fixed time horizon, for instance T, together with a finial cost $\phi(y(T))$. This is the one we consider in Chapter 6 without the discount rate. Moreover, the dynamics and running cost may depend on time. In that case, one may consider a new state $z = (y, t) \in \Omega \times [0, +\infty)$ and the dynamics of the new state $\dot{z} = (f(t, y, u), 1)$, then treat it as for the previous form.

3.1.2 Dynamic Programming Principle and Hamilton-Jacobi-Bellman Equation

In this section, we always take $\Omega = \mathbb{R}^d$. In the optimal control problem, we observe that the minimum value of the cost functional depends on the initial state x. Thus, we shall define the *value function* $v : \mathbb{R}^d \to \mathbb{R}$ as follows:

$$v(x) := \inf_{u(\cdot) \in \mathcal{U}} J(x, u(\cdot)) \ . \tag{3.4}$$

We first assume that $v(x) > -\infty$, and that the infimum in (3.4) is achieved in some $u_x^*(\cdot)$. The dynamic programming principle asserts that, as Bellman said, the optimal control u_x^* remains

an optimal control for any successive states $y_{u_x^*}(x;t)$. Namely, consider an arbitrary $t > 0$, we can rewrite the optimality condition as

$$v(x) = \int_0^t e^{-\lambda s}\ell(y_{u_x^*}(x;s), u_x^*(s))ds + \int_t^\infty e^{-\lambda s}\ell(y_{u_x^*}(x;s), u_x^*(s))ds \ . \tag{3.5}$$

By a simple change of variable, i.e., let $\tau = s - t$, we have the second part of the sum in (3.5) is equal to

$$e^{-\lambda t}\int_0^\infty e^{-\lambda\tau}\ell(y_{u_x^*}(x,\tau+t), u_x^*(\tau+t))ds \ . \tag{3.6}$$

Using the dynamic programming principle, we deduce that (3.5) is equivalent to

$$v(x) = \int_0^t e^{-\lambda s}\ell(y_{u_x^*}(x;s), u_x^*(s))ds + e^{-\lambda t}v(y_{u_x^*}(x;t)) \ , \tag{3.7}$$

that is the integral part in (3.6) is equal to the value function with initial state $y_{u_x^*}(x;t)$. In the general case, that is when the optimal control u_x^* is not necessarily achieved, we replace (3.7) by

$$v(x) = \inf_{u(\cdot)\in\mathcal{U}}\left\{\int_0^t e^{-\lambda s}\ell(y_u(x;s), u(s))ds + e^{-\lambda t}v(y_u(x;t))\right\} \ . \tag{3.8}$$

The dynamic programming principle provides a tool for analyzing the optimality conditions of the control problem. To derive the equation that the value function should satisfy, let us begin by assuming that v is differentiable at all points $x \in \mathbb{R}^d$. Then, taking the limit as $t \to 0$ in (3.8), we obtain that the value function is a solution of the following partial differential equation, the so called *Hamilton-Jacobi-Bellman* (HJB) equation

$$\lambda v(x) + H(x, \nabla v(x)) = 0 \ , \tag{3.9a}$$

where ∇v denotes the gradient of v w.r.t. x, and the *Hamiltonian* H is given by:

$$H(x,p) = \sup_{u\in U}\{-p\cdot f(x,u) - \ell(x,u)\} \ , x,p \in \mathbb{R}^d \ . \tag{3.9b}$$

The HJB equation (3.9) provides a sufficient and necessary optimality condition for the control problem. Indeed, once (3.9) is solved, one can get the optimal control as a maximizer of the Hamiltonian,

$$u^*(x) = \text{Argmin}_{u\in\mathcal{U}}\{-\nabla v(x)\cdot f(x,u) - \ell(x,u)\} \ . \tag{3.10}$$

Notice that in (3.10), the optimal control is expressed as a function of the state x. It is also called a *feedback* optimal control, or *closed-loop* optimal control, which has desirable advantages in real applications, for instance the solution is robust against system perturbations.

3.1.3 Viscosity solutions

The dynamic programming principle and HJB equation provide powerful tools addressing the optimal control problems, as long as equation (3.9) holds everywhere for the value function v. However, the assumption that v be everywhere differentiable is too restrictive. Furthermore, if we relax this condition and consider only that (3.9) is satisfied almost everywhere, there may be other solutions than the value function. Thus, a good formulation of weak solution of (3.9) is needed to relate it with the value function of the control problem.

In the early 1980s, Crandall and Lions introduced the notion of *viscosity solution* in papers [CL83; CEL84], an uniqueness result for the first order equations was also provided. The

value function of a large class of optimal control problems is then characterized as the unique solution of the associated HJB equation in the viscosity sense.

Following [CL83; CEL84], we briefly introduce the notion of viscosity solution. We assume that v is continuous and bounded in $\mathbb{R}^d$, which is a mild condition typically satisfied under general assumptions on f and ℓ. Let us define, respectively, the *superdifferential* of v at x as

$$\nabla^+ v(x) = \left\{ p \in \mathbb{R}^d \mid \limsup_{y \to x} \frac{v(y) - v(x) - p \cdot (y - x)}{|y - x|} \leqslant 0 \right\} , \tag{3.11a}$$

and the *subdifferential* of v at x as

$$\nabla^- v(x) = \left\{ q \in \mathbb{R}^d \mid \liminf_{y \to x} \frac{v(y) - v(x) - q \cdot (y - x)}{|y - x|} \geqslant 0 \right\} . \tag{3.11b}$$

Then viscosity solutions are defined as follows.

Definition 3.1.1. Let v be continuous and bounded from $\mathbb{R}^d$ to $\mathbb{R}$, then

- v is a *viscosity subsolution* of (3.9) if

$$\lambda v(x) + H(x, p) \leqslant 0, \quad \forall\, x \in \mathbb{R}^d,\ \forall\, p \in \nabla^+ v(x) . \tag{3.12a}$$

- v is a *viscosity supersolution* of (3.9) if

$$\lambda v(x) + H(x, p) \geqslant 0, \quad \forall\, x \in \mathbb{R}^d,\ \forall\, q \in \nabla^- v(x) . \tag{3.12b}$$

- v is a viscosity solution of (3.9) if it is a viscosity subsolution and viscosity supersolution of (3.9).

Notice that when v is differentiable at x, we have $\nabla^+ v(x) = \nabla^- v(x) = \{\nabla v(x)\}$. In particular if v is differentiable everywhere in $\mathbb{R}^d$, the notion of viscosity solution is consistent with the notion of classical solution. There is another equivalent definition of viscosity solution in terms of *test functions* as follows.

Definition 3.1.2. Let v be continuous and bounded from $\mathbb{R}^d$ to $\mathbb{R}$, then

- v is a *viscosity subsolution* of (3.9) if for every test function $\phi \in \mathcal{C}^1(\mathbb{R}^d, \mathbb{R})$, and for all local maximum points $x_0 \in \mathbb{R}^d$ of the function $v - \phi$, we have

$$\lambda v(x_0) + H(\nabla \phi(x_0), p) \leqslant 0 . \tag{3.13a}$$

- v is a *viscosity supersolution* of (3.9) if for every test function $\phi \in \mathcal{C}^1(\mathbb{R}^d, \mathbb{R})$, and for all local minimum points $x_0 \in \mathbb{R}^d$ of the function $v - \phi$, we have

$$\lambda v(x_0) + H(\nabla \phi(x_0), p) \geqslant 0 . \tag{3.13b}$$

- v is a viscosity solution of (3.9) if it is a viscosity subsolution and viscosity supersolution of (3.9).

We refer to [CEL84] for the proof of the equivalence between Definition 3.1.1 and Definition 3.1.2. Additionally, we cite the following lemma as an intuition of this equivalence.

Lemma 3.1.3. *Let v be continuous and bounded from $\mathbb{R}^d$ to $\mathbb{R}$. We have*

(i) *$p \in \nabla^+ v(x_0)$ if and only if there exists $\phi \in \mathcal{C}^1(\mathbb{R}^d, \mathbb{R})$ such that x_0 is a local maximum point of $v - \phi$, and $p = \nabla \phi(x_0)$.*

(ii) *$q \in \nabla^- v(x_0)$ if and only if there exists $\phi \in \mathcal{C}^1(\mathbb{R}^d, R)$ such that x_0 is a local minimum point of $v - \phi$, and $q = \nabla \phi(x_0)$.*

The concept of viscosity solution bridges the gap between the regularity of the value function and the HJB equation. More precisely, under general assumptions on f and ℓ, the value function v of the optimal control problem is the unique viscosity solution of the HJB equation.

3.1.4 State Constrained Control Problem and HJB equation

For the optimal control problem, it is natural to consider that the state of the system is required to remain within the closure $\overline{\Omega}$ of a certain open domain Ω, which is also often the case in practical applications. In that case, in addition to the dynamics of the system (3.2), we put an additional constraint for the optimal control problem,

$$y(s) \in \overline{\Omega},\ \forall s > 0\ . \tag{3.14}$$

In that case, the continuity of the value function is not easily obtained, and the notion of state constrained HJB equation together with the state constrained viscosity solution are needed.

Soner introduced in [Son86a; Son86b] a controllability condition in the dynamics and in the boundary of the state space to establish the uniform continuity of the value function. Namely, the followings assumptions on Ω and f are needed.

Assumption (A1) Let $\partial\Omega$ denote the boundary of Ω,

(i) $\partial\Omega$ is compact.

(ii) $\partial\Omega$ is of class $\mathcal{C}^2$.

(iii) There exists a positive constant C such that, for every $x \in \partial\Omega$, there exists $u \in U$,

$$f(x,u) \cdot n(x) \leqslant -C < 0\ , \tag{3.15}$$

where $n(x)$ denotes the exterior normal vector to $\overline{\Omega}$ at x.

The points (i) and (ii) in Assumption (A1) can be relaxed under specific conditions. The point (iii) in Assumption (A1), also called the *inward pointing qualification condition*, indeed require that at every point of the boundary of Ω, there exists a field of the system pointing inward the domain Ω. Under Assumption (A1), the value function is bounded and uniformly continuous on $\overline{\Omega}$, and it is obtained as the *constrained viscosity solution* of the HJB equation defined as follows.

Definition 3.1.4. A bounded uniformly continuous function v on $\overline{\Omega}$ is a constrained viscosity solution of (3.9) on $\overline{\Omega}$ if it is a viscosity subsolution on Ω and a viscosity supersolution on $\overline{\Omega}$.

The property that v is a viscosity supersolution on $\overline{\Omega}$ imposes a boundary condition. Using the definition of viscosity subsolution and supersolution in (3.12a) and (3.12b) (or in (3.13a) and (3.13b)), the value function of the state constrained problem can be characterized as the following state constrained HJB equation:

$$\begin{cases} \lambda v(x) + H(x, \nabla v(x)) = 0, & x \in \Omega\ , \\ \lambda v(x) + H(x, \nabla v(x)) \geqslant 0, & x \in \partial\Omega\ . \end{cases} \tag{3.16}$$

The uniqueness result is also established under Assumption (A1) and general assumptions on f and ℓ.

It is worth noting when addressing the exit time problem, that in addition to the boundary condition due to the state constraint, a boundary condition due to the target set should be defined. This is the case we explore in Chapter 4. We also refer to [CL90] for reference.

3.2 Classical Numerical Methods for HJB Equations

Up to rare cases, optimal control problems and HJB equations can only be solved approximately using numerical methods. We should mention that the HJB equation is itself a nonlinear partial differential equation, thus it can be approximated using *finite difference scheme*, which is among the most common approach for numerically solving PDEs. The purpose of this chapter is to present another classical numerical scheme for approximating the HJB equations, the so called *Semi-Lagrangian schemes*. This scheme somehow arises by applying the dynamic programming principle to a discrete time optimal control problem obtained after an Euler time-discretization of the dynamics. The material of this chapter is mainly based on the reference paper of Capuzzo-Dolcetta and Ishii [DI84], and of the reference book of Falcone and Ferretti [FF14].

3.2.1 Discrete Time Optimal Control Problem

In this section, we intend to approximate the optimal control problem presented in Section 3.1.1 by a discrete time optimal control problem. Let us consider the Euler approximation scheme in time, with step Δt, of the dynamical system (3.2), that is,

$$\begin{cases} y^h(k+1) = y^h(k) + \Delta t f(y^h(k), u(k\Delta t)), \quad \forall\ k = 0, 1, 2, \dots, \\ y^h(0) = x\ . \end{cases} \tag{3.17}$$

Here, h stands for the discretization. Moreover, we consider a subset $\mathcal{U}^h$ of $\mathcal{U}$, containing the controls that take piecewise constant values, that is

$$\mathcal{U}^h := \{u \in \mathcal{U} \mid u(s) \equiv u(k\Delta t),\ \text{for all } s \in [k\Delta t, (k+1)\Delta t[\ \}\ . \tag{3.18}$$

Given a $u(\cdot) \in \mathcal{U}^h$, the solution of the system (3.17) is denoted by $y^h_u(x;k)$, $k = 0, 1, 2, \dots$. We shall also consider a discrete cost functional, defined as follows,

$$J^h(x, u(\cdot)) := \Delta t \sum_{k=0}^{\infty} (1 - \lambda\Delta t)^k \ell(y^h_u(x;k), u(k\Delta t))\ . \tag{3.19}$$

Notice that (3.19) can be thought of as a rectangular approximation of the integral function in the continuous cost functional (3.3), for which $y(t)$ is defined by $y(t) = y^h_u(x;k)$ for every $t \in [k\Delta t, (k+1)\Delta t)$. Similarly as in continuous case, the minimum over all $u(\cdot) \in \mathcal{U}^h$ of the discrete cost functional (3.19) depends on the initial state x. We shall define the discrete value function $v^h : \mathbb{R}^d \to \mathbb{R}$ as follows:

$$v^h(x) := \inf_{u(\cdot)\in\mathcal{U}^h} J^h(x, u(\cdot))\ . \tag{3.20}$$

3.2.2 Semi-Lagrangian Scheme

We apply the dynamic programming principle to the discrete time problem constructed in Section 3.2.1, this leads to the following equation

$$v^h(x) + \max_{u\in\mathcal{U}} \Big\{ -(1 - \lambda\Delta t) v^h(x + \Delta t f(x, u)) - \Delta t \ell(x, u) \Big\} = 0\ , \tag{3.21}$$

for every $x \in \mathbb{R}^d$. The existence, uniqueness and regularity property of the discretized equation (3.21) are well studied, for which we refer to the book of [FF14]. In a nutshell, under general regular assumptions on ℓ and f, we have that the value function of the discrete optimal control problem is the unique solution of the equation (3.21).

The discrete equation (3.21) can be thought of as an approximation of the HJB equation (3.9). This approximation is commonly referred to as the Semi-Lagrangian scheme, with only a semi-discretization in time. One advantage of this scheme, compared to finite difference schemes, is that the time step can be chosen independently of the space discretization (as will be detailed later). Indeed, if (3.9) admits a classical solution, that is $v \in \mathcal{C}^1$, the convergence of v^h to v as $h \to 0$ is relatively intuitive. However, as the essential purpose is to approximate the value function of the optimal control problem, the convergence should be understood in the viscosity sense. We present the convergence result below.

Proposition 3.2.1. *Let us denote*

$$\liminf_{h\to 0, y\to x} v^h(y) = \underline{v}(x), \quad \limsup_{h\to 0, y\to x} v^h(x) = \overline{v}(x) \ . \tag{3.22}$$

Under general assumptions on ℓ and f, $\underline{v}$ is a viscosity supersolution of the HJB equation (3.9), $\overline{v}(x)$ is a viscosity subsolution of the HJB equation (3.9). Thus, $\{v^h\}$ converges uniformly to the viscosity solution of (3.9) on any compact subset of $\mathbb{R}^d$ as $h \to 0$.

As part of our interests to solve the optimal control problem, we also propose a method to approximate the optimal trajectory and the feedback optimal control. Indeed, once (3.17) is solved, one can compute a discrete optimal control w.r.t. every $x \in \mathbb{R}^d$ as follows:

$$u^{h,*}(x) \in \text{Argmax}_{u\in U} \left\{ -(1-\lambda\Delta t) v^h(x + \Delta t f(x,u)) - \Delta t f(x,u) \right\} \ . \tag{3.23}$$

Starting from the initial sate x, one can iteratively find the approximate optimal trajectory $y^{h,*}$, that is,

$$\begin{cases} y^{h,*}(k+1) = y^{h,*}(k) + \Delta t f(y^{h,*}(k), u^{h,*}(y^{h,*}(k))), \quad k = 0, 1, 2, \dots, \\ y^{h,*}(0) = x \ . \end{cases} \tag{3.24}$$

Then, one can approximate the optimal control process of the continuous problem as follows:

$$u^{h,*}(s) := u^{h,*}(y^{h,*}(\lfloor \frac{s}{\Delta t} \rfloor)), \quad s \in [0, \infty) \ , \tag{3.25}$$

where $\lfloor a \rfloor$ denotes the greatest integer smaller or equal to a. The control process $u^{h,*}$ constructed in (3.25) can thought of as a piecewise constant approximation of the optimal control.

The approximation scheme (3.21) provides a method for approximating the value function. However, it only involves a semi-discretization in time, and is defined in all $x \in \mathbb{R}^d$, making it impractical to be implemented. Thus, for practical computations, we need to further discretize the state space. For easy expression, we make the following assumption

Assumption (A2) There exists a bounded polyhedral domain $X \subset \mathbb{R}^d$ such that,

$$x + \Delta t f(x,u) \in X, \quad \forall\ x \in X \text{ and } \forall\ u \in U \ , \tag{3.26}$$

with Δt small enough.

We should note that when there is a state constraint, as in the case we discussed in Section 3.1.4, the state space $\overline{\Omega}$ can act as X, and Assumption (A2) may be replaced by a weaker condition. This typically occurs under sufficiently regular assumptions on the boundary of Ω. We refer to the book of [FF14] for further studies. We also explore the numerical approximation in the case of state constrained problems, as in Chapter 4 and Chapter 5.

Consider now a regular triangular discretization of X with diameter Δx, and denote it by X^h. Let us construct an approximation of the value function, w^h, obtained as follows:

we apply the Semi-Lagrangian scheme (3.21) to all the grid nodes $x_i \in X^h$, while when the point $(x_i + \Delta t f(x_i, u))$ is not in the grid X^h, we compute the value of $w^h(x_i + \Delta t f(x_i, u))$ by an interpolation of the values of it's neighborhood nodes. We assume given an interpolation operator $I[\cdot]$ to be used in (3.21) when $x_i \in X^h$, then consider the following *fully discretized Semi-Lagrangian scheme*. Define $z^h : X^h \to \mathbb{R}$ by

$$z^h(x_i) + \max_{u \in \mathcal{U}} \Big\{ -(1 - \lambda \Delta t) I[z^h](x_i + \Delta t f(x_i, u)) - \Delta t \ell(x_i, u) \Big\} = 0, \quad \forall\, x \in X^h \ , \tag{3.27a}$$

then w^h is obtained by

$$w^h = I[z^h] \ . \tag{3.27b}$$

Let us now go into more details by considering a simple interpolation operator, denoted by I_1, which is the P_1 (piecewise linear) interpolation operator in the simplices of the triangular. For every $x \in X$, let $Y^h(x) = \{y_k\}_{k=1,2,\dots,d+1}$ denote the set of vertices of the simplex that contains x. Then we have

$$I_1[z^h](x) = \sum_{y_k \in Y^h(x))} \lambda(x; y_k) z^h(y_k) \ , \tag{3.28a}$$

where the coefficients $\lambda(x; y_k)$ depend on x and y_k, and are uniquely determined by the following condition:

$$\begin{cases} 0 \leqslant \lambda(x; y_k) \leqslant 1, \text{ for every } y_k \ , \\ \displaystyle\sum_{y_k \in Y^h(x)} \lambda(x; y_k) = 1 \text{ and } \sum_{y_k \in Y^h(x)} \lambda(x; y_k) y_k = x \ . \end{cases} \tag{3.28b}$$

We observe that, in the formulation (3.28b), the coefficients $\{\lambda_k(x; y_k)\}$ can be interpreted as the transition probabilities of a controlled Markov chain. In this interpolation, the state space is defined as the set of nodes in X^h, and the transition probability from the node x_i to the node y_k under the dynamics $f(x_i, u)$ is $\lambda(x_i + \Delta f(x_i, u); y_k)$. One interest of this property arises when considering the Semi-Lagrangian scheme as a stochastic control problems. Moreover, we use this property to demonstrate a convergence result in Chapter 5.

3.2.3 A Glance of Convergence Analysis

We already give the convergence result of the semi-discretization scheme. In this section, we will provide some convergence rate estimates for both the semi-discretization scheme and the fully discretized scheme.

A first convergence rate estimate for the semi-discretization scheme, which typically occurs under mild assumptions on v, is stated as follows:

$$\|v - v^h\|_\infty \leqslant C_1 (\Delta t)^{\frac{1}{2}} \ , \tag{3.29}$$

where C is a constant depending on various regularity bounds of ℓ and f but not on Δt. For the fully discretized scheme, the following estimate holds in the sense of sup-norm over X

$$\|v^h - w^h\|_{\infty, X} \leqslant C_2 (\frac{\Delta x}{\Delta t}) \ , \tag{3.30}$$

which typically happens by showing the Lipschitz continuity of v^h. Thus, combining (3.29) and (3.30), we have the following result.

Proposition 3.2.2. *Under general assumptions on ℓ and f, w^h convergences uniformly to v on x as $\Delta t \to 0$ and $\frac{\Delta x}{\Delta t} \to 0$, and the following estimate holds:*

$$\|v - w^h\|_{\infty, X} \leqslant C_1 (\Delta t)^{\frac{1}{2}} + C_2 (\frac{\Delta x}{\Delta t}) \ . \tag{3.31}$$

The estimate in (3.29) can be improved, for instance, under semi-concavity assumptions. However, improving the estimate in (3.30) is challenging, since it typically requires regularity conditions on v^h. For the interest of the readers, we refer to [Fer02] for some improvement of such an estimate under a strong convexity assumption on the Hamiltonian. In Chapter 5, we also improve this estimate using the connection between the fully discretized scheme and the controlled Markov process.

3.2.4 Curse-of-dimensionality

Following the previous discussion, we can think that the theoretical studies of the dynamic programming principle approach for solving optimal control problems is rather complete, ranging from characterization of the value function as the unique viscosity solution of HJB equation, to the approximation of the equation using semi-discretization schemes and fully discretized schemes. However, one major difficulty that prevents this approach to be used in real applications is the well known *curse-of-dimensionality*, which was first expressed by Richard Bellman in [BCC57]:

- *"...what casts the pall over our victory celebration? It is the curse of dimensionality, a malediction that has plagued the scientist from the earliest days."*

Indeed, as we can see in (3.9), the HJB equation is formulated in the same dimension as the state space, which typically has a very high dimension in real applications. Setting aside the problems related to the regularity of the value function, solving high dimensional PDEs is a challenging problem in itself and constitutes a field of study. As for the Semi-Lagrangian scheme, even in the favorable cases where (3.29) holds with an order of Δt, one can only expect the total error estimate of the fully discretized scheme is in the order of $(\Delta t + \frac{\Delta x}{\Delta t})$. Therefore, by choosing $\Delta t = (\Delta x)^{\frac{1}{2}}$, one can obtain an error estimate equal to $O((\Delta x)^{\frac{1}{2}})$. In other words, to achieve an error less or equal to $\varepsilon > 0$, one would need to take $\Delta x = O(\varepsilon^2)$, thus resulting in a computational complexity of $O((\frac{1}{\varepsilon})^{2d})$. In practice, numerical computation is feasible only in a dimension $d \leqslant 4$ in modern computers. We should note that this computational complexity estimate explains on of the interests of the studies in high order schemes. Indeed, if the order of convergence can be improved, for instance from $(\Delta x)^{\frac{1}{2}}$ to (Δx), this can lead to a significant reduction in complexity or enables the handling of dimensions that are twice as large. Mitigating the curse of dimensionality is the primary motivation driving all the researches in the subsequent chapters.

3.3 The Fast Marching Method

In this section, we present a numerical method known as the Fast Marching Method (FMM), introducued in [Tsi95; Set96]. This method was originally proposed for solving the *isotropic eikonal equation* which is of the form

$$\begin{cases} c(x)\|T(x)\| = 1, & x \in \mathbb{R}^d \setminus \Omega_0 \ , \\ T(x) = 0, & x \in \partial\Omega_0 \ , \end{cases} \tag{3.32}$$

where $\|\cdot\|$ denotes the Euclidean norm and $c > 0$ for every x. The equation (3.32) arises from the monotone front propagation problem, where the front of interface evolves monotonically along its normal direction with speed $c(x)$. In particular, the viscosity solution T of (3.32) also represents the value function of a minimum time problem, and the level set $\{x \in \mathbb{R}^d \mid T(x) = t\}$ of T models the interface at time t. In this case, the value function is monotone non-decreasing in the direction of propagation. This section is based on a series of works in [Tsi95; Set96; SV03; CF07; For09; CCV14; Mir18; Mir19]

3.3.1 Minimum Time Problem and Eikonal Equation

In this section we introduce the eikonal equation by showing it's connection with a particular optimal control problem, the minimum time problem. Consider a compact set $\Omega_0 \subset \mathbb{R}^d$. Our objective is to find the minimum time necessary to travel from a point $x \in \mathbb{R}^d \setminus \Omega_0$ to the boundary of Ω_0, with regular speed function $c : \mathbb{R}^d \to \mathbb{R}_{>0}$. Here, regularity means that c is Lipschitz continuous, bounded and strictly positive. The control represents the direction of motion. More precisely, let S_1 be the unit sphere in $\mathbb{R}^d$, i.e., $S_1 = \{x \in \mathbb{R}^d, \|x\| = 1\}$. We consider an optimal control problem such that the set of controls is in the unit sphere, that is $\mathcal{U} = \{u : \mathbb{R} \to S_1 : u(\cdot) \text{ is measurable}\}$, and the evolution of the system is determined by the following equation

$$\begin{cases} \dot{y}(s) = c(y)u(s), \quad \forall\ s > 0\ , \\ y(0) = x\ . \end{cases} \tag{3.33}$$

Let us denote $y_u(x; s)$ the solution of (3.33). The objective functional has the form

$$J(u(\cdot), x) = \inf\{\tau \geqslant 0 \mid y_u(x; \tau) \in \Omega_0\}\ . \tag{3.34}$$

The value function $T : \mathbb{R}^d \to \mathbb{R}$ is defined by

$$T(x) = \inf_{\alpha \in \mathcal{U}} J(u(\cdot), x), \quad \forall\ x \in \mathbb{R}^d\ . \tag{3.35}$$

Then, restricted to $\overline{\mathbb{R}^d \setminus \Omega_0}$, T is a viscosity solution of the following stationary Hamilton-Jacobi-Bellman equation

$$\begin{cases} \max\limits_{u \in \mathcal{U}} \{-(\nabla T(x) \cdot u)c(x) - 1\} = 0, & x \in \mathbb{R}^d \setminus \Omega_0\ , \\ T(x) = 0, & x \in \partial\Omega_0\ . \end{cases} \tag{3.36}$$

Notice that the maximum of the first equation of (3.36) is achieved at

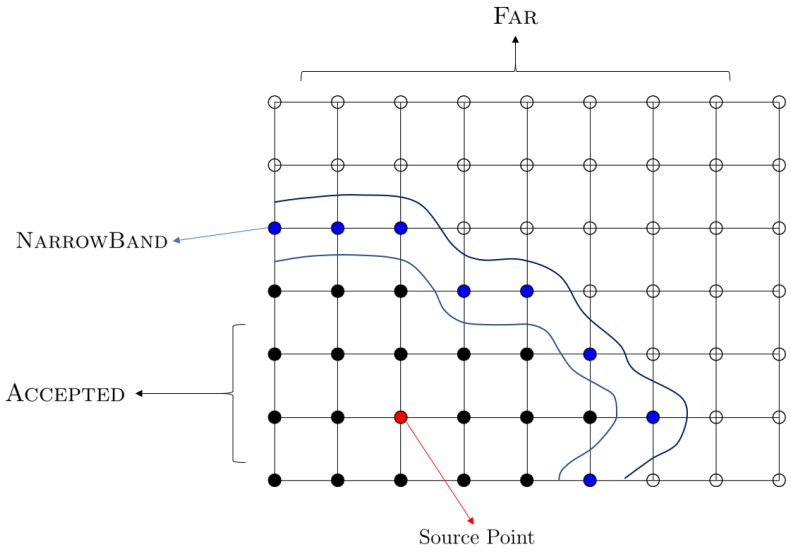
$$u^*(x) := \text{Argmax}_{u \in S_1}\{-\nabla T(x) \cdot u\} = -\frac{\nabla T(x)}{\|\nabla T(x)\|}\ , \tag{3.37}$$

thus (3.36) can be written as in (3.32).

3.3.2 Finite Difference Fast Marching Method

To simplify the notion, we will present the results in dimension $d = 2$. The first fast marching method, initially introduced in [Tsi95; Set96], is based on the finite difference scheme for the equation (3.32). Assuming we discretize the whole domain by a grid X^h and then discretize the equation by finite difference scheme, the classical method to solve the resulting equation involves preforming iterative computations. At each iteration, the approximate solution is improved everywhere in the grid nodes of X^h. The fast marching method, which can be viewed as an acceleration of theses iterative steps, visits the nodes of X^h in a special ordering and computes the approximate value function in just one iteration. This special ordering, constructed by fast marching method, is such that the value function is monotone non-increasing in the direction of propagation. This construction is done by dividing the nodes into three groups (as illustrated in Section 3.3.2),

- Far, which contains the nodes that have not been searched yet;
- Accepted, which contains the nodes at which the value function has been already computed and settled – by the monotone property, in the subsequent search, we do not need to update the value function at those nodes;



- NarrowBand, which contains the nodes "around" the front – at each step, the value function is updated only at these nodes.

Let us consider, for simplicity, a grid of $N \times N$ nodes, $X^h = \{(x_i, y_j)\}_{1 \leqslant i,j \leqslant N}$, with Δx as the discretization step in both the x and y axes. For an element $(x, y) \in \mathbb{R}^2$, let us denote T_x, T_y the partial derivatives of T with respect to x and y, respectively. Then, equation (3.32) can be written as follows

$$T_x^2 + T_y^2 = \frac{1}{(c(x,y))^2} \ . \tag{3.38}$$

We then use a first order finite difference on the grid nodes to approximate the partial derivatives T_x and T_y. However, as mentioned in Section 3.2, it is important to note that the convergence of the numerical approximation for HJB equation should be understood in the viscosity sense. Therefore, we need to apply an *up-wind* finite difference scheme. Let us use $T_{i,j}$ and $c_{i,j}$ to represent the values of T and c at the grid points (x_i, y_j), respectively. Then, the up-wind finite difference scheme for (3.38) is given as follows:

$$\begin{aligned} &\frac{1}{(\Delta x)^2}\Big(\max\Big\{(T_{i,j} - T_{i-1,j})^+, -(T_{i+1,j} - T_{i,j})^-\Big\}\Big)^2 \\ &+ \frac{1}{(\Delta x)^2}\Big(\max\Big\{(T_{i,j} - T_{i,j-1})^+, -(T_{i,j+1} - T_{i,j})^-\Big\}\Big)^2 = \frac{1}{c_{i,j}^2} \ , \end{aligned} \tag{3.39}$$

where we denote $(a)^+ := \max(a, 0)$ and $(a)^- := \min(a, 0)$. Moreover, we define the set of neighborhood nodes of $x = (x_i, x_j)$ in the first order finite difference scheme as follows:

$$N_{FD}(x) = \{(x_{i+1}, y_j), (x_{i-1}, y_j), (x_i, y_{j+1}), (x_i, y_{j-1})\} \ . \tag{3.40}$$

Then, a sketch of finite difference fast marching method is presented as follows,

Initialization. The algorithm starts by labeling only nodes in $\Omega_0 \cap X^h$ as Accepted, and the value of these nodes are set to 0. Moreover, we set the nodes that are in the neighborhood of the Accepted nodes, as defined in (3.40), and that are not accepted to NarrowBand. The value of the initial NarrowBand nodes are set to $\frac{\Delta x}{c(x)}$. The remaining nodes are set to Far, and the value of Far nodes are set to $+\infty$.

Main Loop. The main loop of the fast marching method consists of the following steps

- Select the node in NarrowBand with the smallest value, denoted by x^*. Add x^* into the set of Accepted nodes.
- Add the neighborhood nodes of x^*, $N_{FD}(x^*)$, that are in the set Far to NarrowBand.
- Update values of the nodes in $N_{FD}(x^*)$ that are in the NarrowBand set using equation (3.39).

Then main loop stops when NarrowBand is empty.

An interesting property of the fast marching method is that NarrowBand can be thought of as an approximation of the interface front.

3.3.3 Semi-Lagrangian Fast Marching Method

Under enough regularity assumptions, the Semi-Lagrangian scheme is known to be more accurate than finite difference scheme, thus it is natural to adapt the fast marching method using Semi-Lagrangian discretization. A basic tool in the studies of the minimum time problem is the change of variable

$$v(x) = 1 - e^{-T(x)} \ , \tag{3.41}$$

which was first used by Kruzkov [Kru75]. The function $v(x)$ is bounded and Lipschitz continuous. The minimum time is recovered by $T(x) = -\log(1 - v(x))$. Notice that the Kruzkov transform is monotone, thus when we focus on approximating v, it will not change the order constructed by the fast marching method. Moreover, this approach can deal with the cases in which $c = 0$, for instance when obstacles are present. In these cases the value v remains bounded by 1. By doing so, we change the system (3.36) to

$$\begin{cases} v(x) + \max\limits_{u \in \mathcal{U}} \{-(\nabla v(x) \cdot u)c(x) - 1\} = 0, & x \in \mathbb{R}^d \setminus \Omega_0 \ , \\ v(x) = 0, & x \in \partial\Omega_0 \ . \end{cases} \tag{3.42}$$

We follow the discretization steps as in Section 3.2.2 and, in particular, we take the time step $\Delta t = h$. Given an interpolation operator I, a fully discretized scheme is given as follows:

$$\begin{aligned} z^h(x_i) &= \min_{u \in S_1} \left\{ (1-h) I[z^h](x_i + hc(x)u) + h \right\}. \quad \forall x_i \in X^h \setminus \Omega_0 \ , \\ z^h(x_i) &= 0, \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad x \in X^h \cap \Omega_0 \ , \end{aligned} \tag{3.43}$$

and $w^h = I[z^h]$. Notice that an efficient method to compute the minimum in (3.43) is to set $h(x) = \frac{\Delta x}{c(x)}$, in which the time step varies with respect to the state. Then, by using a piecewise linear interpolation operator, the minimum can be computed within a sphere with radius Δx. However, though most of the numerical experiments reveal accurate results, the convergence requires further verification. As we can see from Section 3.2.3, in general, the fully discretized Semi-Lagrangian scheme converges when both Δt and $\frac{\Delta x}{\Delta t}$ tend to 0. This is one of the motivations of Chapter 5. For the moment, let us take $h(x) = \frac{\Delta x}{c(x)}$, and use a piecewise linear interpolation operator that uses the following neighborhood of nodes of $x = (x_i, y_j)$

$$N_{SL}(x) = N_{FD} \cup \{(x_{i+1}, y_{j+1}), (x_{i+1}, y_{j-1}), (x_{i-1}, y_{j+1}), (x_{i-1}, y_{j-1})\} \ . \tag{3.44}$$

The Semi-Lagrangian fast marching is then a slight modification of the one proposed in Section 3.3.2. We follow the same steps as in the finite difference case, except that when updating the value function in the grid nodes, we use (3.43) with the neighborhood $N_{FD}(\cdot)$ replaced by $N_{SL}(\cdot)$.

3.3.4 Computational Complexity and Data Structure

Let us focus on the computational complexity of the fast marching method. Denote $M = N \times N$ the number of nodes in the grid. In both Semi-Lagrangian and finite difference cases, during each iteration of the main loop, we add one node to the set ACCEPTED. Thus, the loop can be repeated at most M times. At each iteration, we need to select the point that has the minimum value among the NARROWBAND nodes. To efficiently select this node, the nodes of NARROWBAND should be arranged and ordered by their values using a particular data structure, the *binary heap*. This arrangement ensures that the minimum value resides at the root of the heap and thus selecting the node with the minimum value has complexity $O(1)$. Moreover, adding a new node to the NARROWBAND has complexity bound $O(\log(M))$, as there are at most M nodes in NARROWBAND set. The number of new nodes added to the NARROWBAND depends on the size of the neighborhood, which is 4 for the finite difference discretization according to the definition in (3.40), and 8 for the Semi-Lagrangian discretization according to the definition in (3.44). In summary, for two dimensional fast marching method, the computational complexity is $O(M \log(M))$. This complexity can be straightforwardly extended to $d-$dimensional cases, which gives $O(K_d M \log(M))$ with M still be the number of nodes, and the constant $K_d \in [2d, D^d]$ depends on the type of discrete neighborhood that is considered (and D is the maximal diameter of neighborhoods).

The implementation efficiency of fast marching method relies on the data structure used to store the grid nodes, as discussed above. The data of the classical fast marching method are normally stored using two types of structures:

- a full $d-$dimensional table (or tensor), which contains all the values of the current approximate value function on the whole discretization grid (the values are updated at each step);
- a binary heap (min-heap), which contains the information on the NARROWBAND nodes with the current approximate value function.

These particular data structures align with the theoretical complexity estimates mentioned earlier. In Chapter 4, we also explore in details the data structure of fast marching method.

3.3.5 Causality, Anisotropicity and Extension

One crucial assumption that ensures the fast marching techniques work is the so-called causality property, which indeed requires that when applying the update operator to a node, the computation depends only on the nodes with values less than or equal to this node. This assumption also appears in Dijkstra's Algorithm in the discrete case, which is then automatically satisfied when the costs are nonnegative. However, this assumption is quite restrictive when solving general discretized HJB equations. Indeed the causality property naturally holds for usual discretizations of isotropic equations. This is due to the physical interpolation of isotropic front propagation problems, which shows that the characteristic curves of the equation coincide with the gradient lines of its solution. In anisotropic cases, this is not true. Several studies intended to overcome this difficulty. In particular, in [SV03; Vla06], the authors extended the fast-marching method to handle some amount of anisotropicity by increasing the sets of neighborhood points for every node in the grid. Considering now the speed function $c(x, u)$, so depends on the direction of the motion. The authors define a coefficient to measure the anisotropy as follows

$$\gamma := \frac{\max_{x,u} c(x,u)}{\min_{x,u} c(x,u)} \,. \tag{3.45}$$

During computations, the neighborhood of one node is defined by a large set of nodes, which, essentially, contains the nodes of simplices that are at most a distance of γh from the given node. As a consequence, the author show that their methods could handle a certain class of equations, but the larger neighborhood increases the computational complexity. More recently, in [Mir14; Mir18; Mir19], the author extended the fast marching method to some $2-D$ and $3-D$ elliptic anisotropic cases, as well as other types of degenerate anelliptic cases related to curvature penalization. His method is based on discretization using adaptive stencil adapted to the Hamiltonian and associated Voronoi's first reduction of quadratic forms. The computational complexity of the algorithm in [Mir14; Mir18; Mir19] is $O(M \ln M + M \ln k)$, where k represents the maximum anisotropic ratio. Other works intending to generalize the fast marching method include [CFM11; For09; FLG08]. Up to some of the special cases as mentioned above, the anisotropy remains a major difficulty that prevents the generalization of fast marching.

3.4 Max-Plus Based Numerical Methods

In this chapter, we introduce a class of numerical methods for solving deterministic optimal control problems based on the max-plus algebra or tropical algebra. To be consistent with most of the initial references, we will focus on the following finite horizon undiscounted control problem

$$\max \int_0^T \ell(y(s), u(s))ds + \phi(y(T)) \ . \tag{3.46}$$

where the maximum is taken over the trajectories $(y(\cdot), u(\cdot))$ satisfying the constraint (3.2) and $y(\cdot) \in \Omega, u(\cdot) \in \mathcal{U}$. We will not go into the details of the state constraint, for which we can adapt the method of Section 3.1.4. This class of numerical methods are inspired by the max-plus linearity of the dynamic programming equation of the control problem, and has shown advantages in solving problems under specific regularity conditions. The context of this chapter is mainly based on the reference papers of Fleming and McEneaney [FM00], of Akian, Gaubert and Lakhoua [AGL08], of McEneaney [MDG08b] and on the reference book of McEneaney [McE06].

3.4.1 Max-Plus Semifield

The max-plus semifield is set $\mathbb{R}_{\max} := \mathbb{R} \cup \{-\infty\}$ equipped with two operations: for every $a, b \in \mathbb{R}_{\max}$,

$$\begin{aligned} a \oplus b &:= \max\{a, b\} \ , \\ a \otimes b &:= a + b \ , \end{aligned} \tag{3.47}$$

with $-\infty$ as the 0 element, that is $a \oplus -\infty = a$ for all $a \in \mathbb{R}_{\max}$, and with 0 as the unit element, that is $a \otimes 0 = a$ for all $a \in \mathbb{R}_{\max}$. $\oplus$ and $\otimes$ are often referred as tropical addition and tropical multiplication, respectively. One can then define the classical algebraic computations in $\mathbb{R}_{\max}$, for instance the matrix computations, exponentiation, scalar products, integrals, ...

Every non-zero element x in $\mathbb{R}_{\max}$ has an inverse $-x$ for $\otimes$. However, $\oplus$ is not invertible. This results in a key property of max-plus algebra, that is, a system of linear equations, even when the number of equations coincides with the number of degrees of freedom, and when the system is nonsingular, may have no solution. Thus, in order to solve max-plus linear systems, the notion of *residuation* should be used.

Definition 3.4.1. For every ordered sets $(S, \leqslant)$, $(T, \leqslant)$ and map $f : S \to T$, we say that f is residuated if there exists a map $f^{\#} : T \to S$ such that

$$f(s) \leqslant t \Leftrightarrow s \leqslant f^{\#}(t) \ . \tag{3.48}$$

The map f is residuated if and only if, for all $t \in T$, $\{s \in S \mid f(s) \leqslant t\}$ has a maximum element in S. The residuated map is then defined by

$$f^{\#}(t) := \max\{s \in S \mid g(s) \leqslant t\} \ . \tag{3.49}$$

Let us denote by $\overline{\mathbb{R}}_{\max}$ the complete idempotent semiring obtained by extending $\mathbb{R}_{\max}$ with $+\infty$, $\overline{\mathbb{R}}_{\max} = \mathbb{R} \cup \{+\infty\}$. Any max-plus linear operator $M : \overline{\mathbb{R}}_{\max}^{p} \to \overline{\mathbb{R}}_{\max}^{q}$ has a residuated map. Thus, to solve a max-plus linear system

$$Mx = b \ , \tag{3.50}$$

we consider the maximal subsolution of the inequality $Mx \leqslant b$, which is given by

$$M^{\#}b = \max\{x \mid Mx \leqslant b\} \ . \tag{3.51}$$

In usual notations, it is given by

$$(M^{\#}b)_j = \min_{1 \leqslant i \leqslant p} (-M_{i,j} + b_i), \quad \forall\ j \in \{1, 2, \ldots, q\} \ . \tag{3.52}$$

Then $M^{\#}$ is the residuated map of M.

3.4.2 Max-Plus Variational Formulation and Approximation of HJB Equation

For the optimal control problem proposed in (3.46), let us consider the value function v defined as follows, for every $(x, t) \in \Omega \times [0, T]$,

$$v(x,t) = \sup_{u(\cdot) \in \mathcal{U}} \left\{ \int_0^t \ell(y(s), u(s)) ds + \phi(y(t)) \right\} \ , \tag{3.53}$$

under the same constraint as the control problem. Then, v is the viscosity solution of the following Hamilton-Jacobi-Bellman equation

$$\begin{cases} \dfrac{\partial v}{\partial t} - H(x, \nabla v) = 0 \ , & (x,t) \in \Omega \times [0,T] \ , \\ v(x,0) = \phi(x) \ , & x \in \Omega \ , \end{cases} \tag{3.54a}$$

where the Hamiltonian is defined by

$$H(x,p) = \sup_{u \in \mathcal{U}} \{p \cdot f(x,u) + \ell(x,u)\} \ . \tag{3.54b}$$

Let us denote by S^t the *Lax Oleinik semigroup* of Equation (3.54), i.e., the evolution semigroup of this PDE, meaning that for all $0 \leqslant t \leqslant T$, S^t is the map sending the initial cost function ϕ to the value function $v(\cdot, t)$:

$$S^t[\phi] := v(\cdot, t) \ , \tag{3.55}$$

so that the semigroup property $S^{t_1+t_2} = S^{t_1} \circ S^{t_2}$. Moreover, the map S^t is *max-plus linear*, meaning that, for all scalar $\lambda \in \mathbb{R}_{\max}$ and functions $\phi_1, \phi_2 : \Omega \to \mathbb{R}_{\max}$, we have

$$\begin{aligned} S^t[\phi^1 \oplus \phi^2] &= S^t[\phi^1] \oplus S^t[\phi^2] \ , \\ S^t[\lambda \otimes \phi^1] &= \lambda \otimes S^t[\phi^1] \ , \end{aligned} \tag{3.56}$$

where for any functions ϕ^1 and ϕ^2, $\lambda \otimes \phi^1$ is the function $x \in X \mapsto \lambda + \phi^1(x)$ and $\phi^1 \oplus \phi^2$ is the function $x \in \Omega \mapsto \sup(\phi^1(x), \phi^2(x))$, in the usual sense.

The max-plus numerical methods to solve (3.54) take advantage of the max-plus linearity of S^t. Let us first discretize the time horizon by $\frac{T}{\delta}$ steps. For a given time horizon $t \in \{0, \delta, \dots, T\}$, the value function v^t is approximated by a max-plus linear combination of a family of finitely many "basic functions", $\{w_i\}_{1 \leqslant i \leqslant p}$, with the coefficients, $\{\lambda_i^t\}_{1 \leqslant i \leqslant p} \in \mathbb{R}_{\max}^p$, that is

$$v^t \approx v^{t,h} := \bigoplus_{1 \leqslant i \leqslant p} \{\lambda_i^t \otimes w_i\} \ : \ x \mapsto \max_{1 \leqslant i \leqslant p} \{\lambda_i^t + w_i(x)\} \ . \tag{3.57}$$

Natural choices of the family of basis functions are the Lipschitz functions of the form $w_i(x) := -c\|x - x_i\|_1$, and the quadratic functions of the form $w_i(x) = -\frac{c}{2}\|x - a\|_2^2$. Indeed, let $\mathcal{W}$ be a complete $\mathbb{R}_{\max}$-semimodule of functions $w : \Omega \to \overline{\mathbb{R}}_{\max}$, meaning that $\mathcal{W}$ is stable under the operation of taking the supremum of an arbitrary family of functions, and by the addition of a constant. The semimodule $\mathcal{W}$ is chosen in such a way that $v^t \in \mathcal{W}$ for all $t \in [0, T]$. The family of quadratic functions with Hessian c generates, in max-plus sense, the complete semimodule of lower-semicontinuous $c-$semiconvex functions. In many applications, the value function v^t is known to be $c-$semiconvex for all $t \in [0, T]$ for some constant $c \geqslant 0$, and then $\mathcal{W}$ can be taken as the set of $c-$semiconvex functions, which is a complete module. Let us also denote $\mathcal{W}_h \subset \mathcal{W}$ the semimodule generated by the finite family of basis functions $\{w_i\}_{1 \leqslant i \leqslant p}$. The following Figure 3.1 is a sketch of the approximation of a semiconvex function by the maximum of quadratics.

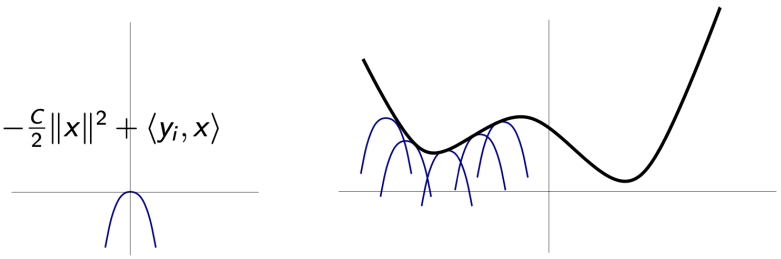


Figure 3.1: Approximation of a $c-$semiconvex function by maximum of quadratics.

After approximating the value function at a certain time horizon t, we need to propagate the approximation of the value function. By the semigroup property, we have

$$\begin{cases} v^{t+\delta} = S^\delta[v^t], \quad \forall \ t = 0, \delta, \dots, T - \delta \ , \\ v^0 = \phi. \end{cases} \tag{3.58}$$

We expect that the approximation of $v^{t+\delta}$ has a similar form as (3.57), using the same family of basis functions $\{w_i\}_{1 \leqslant i \leqslant p}$ together with a family of scalars $\{\lambda_i^{t+\delta}\}_{1 \leqslant i \leqslant p}$, with v^t being approximated by (3.57). Replacing v^t and $v^{t+\delta}$ by such approximations, and using the max-plus linearity of S^δ, the propagation in (3.58) should satisfy

$$\begin{aligned} v^{t+\delta,h} &= \max_{1 \leqslant i \leqslant p} \left\{\lambda_i^{t+\delta} + w_i(x)\right\} \\ &= S^\delta[v^{t,h}](x) = \max_{1 \leqslant i \leqslant p} \left\{\lambda_i^t + S^\delta[w_i](x)\right\}, \quad \forall \ x \in \Omega \ . \end{aligned} \tag{3.59}$$

3.4.3 The Max-Plus Basis Method of Fleming and McEneaney

In [FM00], Fleming and McEneaney propose a recursive update for λ^t by applying a max-plus linear operator at each time step. This method is adapted to problems that exhibit a specific structure: the lagrangian ℓ is quadratic with respect to u and the dynamics f is linear with respect to u. This approach use a semiconvex duality representation of semiconvex functions.

Definition 3.4.2. Let $R > 0$ and $v : \bar{B}_R \to \mathbb{R}_{\max}$ be a $c'-$semiconvex, $L-$Lipschitz function on $\bar{B}_R$. The $c-$semiconvex dualities of v, $\hat{v} : \hat{X} \to \mathbb{R}$, are defined as follows

$$v(x) = \sup_{\hat{x}\in\hat{X}} \left\{ -\frac{c}{2}\|x - \hat{x}\|^2 + \hat{v}(\hat{x}) \right\} := \mathcal{D}_c^{\#}(\hat{v})(x), \quad x \in \bar{B}_R \ , \tag{3.60}$$

and for every $\hat{x} \in \hat{X}$,

$$\hat{v}(\hat{x}) = \inf_{x\in\bar{B}_R} \left\{ \frac{c}{2}\|x - \hat{x}\|^2 + v(x) \right\} := \mathcal{D}_c(v)(\hat{x}) \ . \tag{3.61}$$

Then if $c > c'$, $\hat{v}$ is uniquely defined on $\bar{B}_{D_R}$ with $D_R \geqslant R + \frac{L}{c}$, and $v = \mathcal{D}_c^{\#}\mathcal{D}_c(v)$.

In other words, any semiconvex function ϕ can be represented as the max-plus linear combination of quadratic functions centered at a dense countable subset. Consider the right hand side of the approximation in (3.59), and let us denote $W_h : \mathbb{R}^p_{\max} \to \mathcal{W}$ the max-plus linear operator such that

$$W_h(\lambda) = \oplus_{1\leqslant i\leqslant p}\{\lambda_i \otimes w_i\} \ , \text{ for all } \lambda \in \mathbb{R}^p_{\max} \ . \tag{3.62}$$

Then, the scalars λ_i^t can be inductively computed as follows

$$\begin{cases} W_h\lambda^0 = \phi \ , \\ W_h\lambda^{t+\delta} = S^\delta W_h\lambda^t, \ \forall\ t \in \{0, \delta, \dots, T\} \ . \end{cases} \tag{3.63}$$

The above formula can be expressed using an operator $A : \mathbb{R}^p_{\max} \to \mathbb{R}^p_{\max}$ apply to λ^t, with entries

$$A_{j,i} = \inf \left\{ -w_j(x) + S^\delta[w_i](x) \right\}, \quad \text{for all } i, j \in \{1, 2, \dots, p\} \ . \tag{3.64}$$

In usual notation, we have

$$\lambda_i^{t+\delta} = \max_{1\leqslant j\leqslant p} \left\{ A_{i,j} + \lambda_j^t \right\}, \quad \text{for all } i \in \{1, 2, \dots, p\} \ . \tag{3.65}$$

The recursive equation (3.65) may be interpolated as the dynamic programming equation of a deterministic optimal control problem, with finite state spaces $\{1, 2, \dots, p\}$, and to each state $i \in \{1, 2, \dots, p\}$ there is one possible action $j \in \{1, 2, \dots, p\}$. Given the state j and action i, the reward is $A_{i,j}$.

3.4.4 The Max-Plus Finite Element Method of Akian, Gaubert and Lakhoua

A more involved approximation method was introduced in [AGL08]. The authors also introduce $\mathcal{Z}$, a complete $\mathbb{R}_{\max}$-semimodule of test functions $z : \Omega \mapsto \overline{\mathbb{R}}_{\max}$. Let us define the max-plus scalar product of $u \in \mathcal{W}$ and $v \in \mathcal{Z}$ by

$$\langle u, v\rangle = \sup_{x\in\Omega}\{u(x) + v(x)\} \ . \tag{3.66}$$

If the space of test functions $\mathcal{Z}$ is large enough, then (3.58) is equivalent to:

$$\forall z \in \mathcal{Z}, \quad \begin{cases} \langle z, v^{t+\delta}\rangle = \langle z, S^\delta[v^t]\rangle, \quad \forall\ t \in \{0, \delta, T - \delta\} \ , \\ \langle z, v^0\rangle = \langle z, \phi\rangle \ , \end{cases} \tag{3.67}$$

Let us now consider $\mathcal{Z}_h \subset \mathcal{Z}$, a semimodule generated by a finite family of test functions $\{z_j\}_{1\leqslant j\leqslant q}$, and, instead of requiring (3.67) to hold for all $z \in \mathcal{Z}$, we only require that it holds

for generators, leading to a finite system of equations. Therefore, the approximation $v^{t+\delta,h}$ and v^0 should satisfy:

$$\forall j \in \{1,2,\dots,q\}, \quad \begin{cases} \langle z_j, v^{t+\delta,h}\rangle = \langle z_j, S^\delta[v^{t,h}]\rangle, & \forall\ t \in \{0,\delta,T-\delta\}\ , \\ \langle z_j, v^T\rangle = \langle z_j, \phi_T\rangle\ . \end{cases} \tag{3.68}$$

Let us denote $Z_h^* : \mathcal{W} \to \mathbb{R}^q_{\max}$ such that

$$(Z_h^*(w))_j = \langle z_j, w\rangle, \quad \forall\ 1 \leqslant j \leqslant q\ . \tag{3.69}$$

Notice that the transpose off Z_h^*, Z_h has a similar definition as W_h in (3.62). Then, the scalars can be inductively computed as follows

$$\begin{cases} \lambda^0 = W_h^{\#}\phi\ , \\ \lambda^{t+\delta} = (Z_h^* W_h)^{\#}(Z_h^* S^\delta W_h \lambda^t),\ \forall\ t \in \{0,\delta,\dots,T\}\ . \end{cases} \tag{3.70}$$

The above formula can be expressed as using the linear operators $M_h := Z_h^* W_h$ and $K_h := Z_h^* S^\delta W_h$ applied to λ^t, with entries:

$$(M_h)_{j,i} = \langle z_j, w_i\rangle\ , \tag{3.71a}$$

$$(K_h)_{j,i} = \langle z_j, S^\delta w_i\rangle\ . \tag{3.71b}$$

The matrices M_h and K_h may be thought of as max-plus analogues of the *mass* and *stiffness* matrices arising in the finite element method. In usual notations, we have

$$\lambda_i^{t+\delta} = \min_{1\leqslant j\leqslant q}\left\{-(M_h)_{j,i} + \max_{1\leqslant k\leqslant p}\{(K_h)_{j,k} + \lambda_k^t\}\right\}\ . \tag{3.72}$$

The recursive equation (3.72) may be interpreted as the dynamic programming equation of a deterministic zero-sum two player game, with finite state space $\{1,2,\dots,p\}$. Then, to each state $i \in \{1,2,\dots,p\}$, there is one possible action $j \in \{1,2,\dots,q\}$ for the first player and one possible action $k \in \{1,2,\dots,p\}$ for the second player. Given the state i and the actions j,k, the cost of the first player, which is the reward of the second player, is $(-(M_h)_{j,i} + (K_h)_{j,k})$.

Notice that in the recursive equation (3.72), we assume that the matrices M_h and K_h are exactly known. Hence this method is also called the "ideal" max-plus finite element method. Indeed, computing M_h is relatively easy, given that the basis functions and test functions are chosen to be "nice" concave functions, for instance quadratic functions. As a result, the scalar product in (3.71a) can be computed using standard optimization methods, and in some cases, it can be solved analytically. Computing K_h involves to solve the small time propagation of the basis functions, thus typically it can only be solved approximately. The method with K_h replaced by an approximation is called the "effective" max-plus finite element method.

One way to approximate (3.71b) is to use the HJB equation. Indeed, the equation (3.54) suggests approximating (3.71b) as follows,

$$(K_h)_{j,i} \approx (\tilde{K}_h^1)_{j,i} := \sup_{x\in X}\{z_j(x) + w_i(x) + \delta H(x, \nabla w_i(x))\}\ . \tag{3.73}$$

Indeed, (3.73) can be thought of as a perturbation of the optimization problem associated with the computation of $(M_h)_{j,i}$. Observing also δ is small, we then further approximate the approximation in (3.73) by first computing the scalar product $\langle z_j, w_i\rangle$. Denote $O_{j,i}$ the set where the optimum pf $z_j(x) + w_i(x)$ is obtained, an computational efficient way to approximate K_h is given as follows,

$$(K_h)_{j,i} \approx (\tilde{K}_h^2)_{j,i} := \langle z_j, w_i\rangle + \delta \sup_{x\in O_{j,i}}\{H(x, \nabla w_i(x))\}\ . \tag{3.74}$$

In Chapter 6, we show that, under certain assumptions with small enough δ, the small time propagation is itself a concave optimization problem. Thus it can be solved exactly, or with a negligible error, by direct methods.

3.4.5 A Glance of Convergence Analysis and Error Estimate

For the max-plus finite element method, the error consists of two part: the approximation error for approximating the small time propagation, and the projection error resulting from different choices of finite elements. Under certain technical assumptions, the approximation error is as follows,

$$\|K_h - \tilde{K}_h^1\|_\infty \leqslant C_K^1 \delta^2, \quad \|K_h - \tilde{K}_h^2\|_\infty \leqslant C_K^2 \delta^{\frac{3}{2}} \ . \tag{3.75}$$

The projection error depends on the choices of basis functions and test functions. To clarify, let us define X^h as a grid that discretizes X. The basis and test functions are generated based on the points of the grid X^h. The projection error is characterized by the maximal radius of the Voronoi cells of the space X divided by the points of X^h, denoted by Δx. Under mild regularity assumptions, meaning that v^t is $c-$semiconvex and $L-$Lipschitz continuous for every t, and by choosing quadratic basis functions with Hessian c and $L-$Lipschitz test functions, we can obtain the following error bound when K_h is approximate by $\tilde{K}_h^1$

$$\|v_h^T - v^T\|_\infty \leqslant C_1(\delta + \frac{\Delta x}{\delta}) \ . \tag{3.76a}$$

Moreover, we can obtain the following error bound when K_h is approximate by $\tilde{K}_h^2$

$$\|v_h^T - v^T\|_\infty \leqslant C_2(\sqrt{\delta} + \frac{\Delta x}{\delta}) \ . \tag{3.76b}$$

The error estimate in (3.76) shows that the computational complexities of these methods remain comparable to those of classical grid-based methods.

Chapter 4

A Multilevel Fast-Marching Method for The Minimum Time Problem

***

This chapter is based on the submitted paper [AGL23a].

Abstract. We introduce a new numerical method to approximate the solutions of a class of stationary Hamilton-Jacobi (HJ) partial differential equations arising from minimum time optimal control problems. We rely on several grid approximations, and look for the optimal trajectories by using the coarse grid approximations to reduce the search space in fine grids. This may be thought of as an infinitesimal version of the "highway hierarchy" method which has been developed to solve shortest path problems (with discrete time and discrete state). We obtain, for each level, an approximate value function on a sub-domain of the state space. We show that the sequence obtained in this way does converge to the viscosity solution of the HJ equation. Moreover, for our multi-level algorithm, if $0 < \gamma \leqslant 1$ is the convergence rate of the classical numerical scheme, then the number of arithmetic operations needed to obtain an error in $O(\varepsilon)$ is in $\widetilde{O}(\varepsilon^{-\theta})$, with $\theta < \frac{d}{\gamma}$, to be compared with $\widetilde{O}(\varepsilon^{-\frac{d}{\gamma}})$ for ordinary grid-based methods. Here d is the dimension of the problem, and θ depends on d, γ and on the "stiffness" of the value function around optimal trajectories, and the notation $\widetilde{O}$ ignores logarithmic factors. When $\gamma = 1$ and the stiffness is high, θ is equal to 1, meaning that the theoretical complexity is in $\widetilde{O}(\varepsilon^{-1})$. We describe such special cases. In more general cases such that $\gamma = 1$, θ equals $(d+1)/2$, although in practice θ is rather independent of d and much smaller ($\leqslant 1.7$). We illustrate the approach by solving HJ equations of eikonal type up to dimension 7.

4.1 Introduction

4.1.1 Motivation and context

We consider a class of optimal control problems, consisting of finding the minimum traveling time between two given sets in a given domain. Such optimal control problems are associated to a stationary Hamilton-Jacobi (HJ) equation via the Bellman dynamic programming principle (see for instance [FS06]). In particular, the value function is characterized as the solution of the associated HJ equation in the viscosity sense [CL83; CEL84; FS06]. Problems with state constraints can be addressed with the notion of constrained viscosity solution [Son86a; Son86b].

Various classes of numerical methods have be proposed to solve this problem. The *Finite difference schemes* are based on a direct discretization of the HJ equation (see for example [CL84]). The *Semi-Lagrangian schemes*, as in [Fal87; FF14], arise by applying the Bellman dynamic programming principle to the discrete time optimal control problem obtained after an Euler time discretization of the dynamics. In both cases, the discretized system of equations can be interpreted as the dynamic programming equation of a stochastic optimal control problem [KD01] with discrete time and state space. More recently, max-plus based discretization schemes have been developed in [FM00; McE06; McE07; AGL08]. These methods take advantage of the max-plus linearity of the Lax-Oleinik evolution operator of the HJ equation. They are based on a max-plus basis representation of the discretized value function, leading to a discrete time deterministic optimal control problem.

Once a discretization is done, traditional numerical methods apply iterative steps to solve the discretized stationary HJ equation, for instance value iteration or policy iteration. At each step, the value function is computed in the whole discretization grid. In the particular case of the shortest path problem (with discrete time and state space), with a nonnegative cost function, one can obtain the solution of the stationary dynamic programming equation by Dijkstra's algorithm.

The fast marching method originally introduced by Sethian in [Set96] and by Tsitsiklis in [Tsi95], then further developed for instance in [**mirebeau2014efficient**; SV01; CF07; CFM11; CCV14; Mir14; Mir18; Mir19], is based on that observation. It provides a Dijkstra-type algorithm for a monotone finite difference or semi-lagrangian discretization of the HJ equation of a shortest path problem with continuous time and state. It is called "single pass" because at every point of the discretization grid, the value is computed at most k times, where k is a bound not related to the discretization mesh. Such an approach was initially introduced to solve the front propagation problem, then extended to more general stationary Hamilton-Jacobi equations. It takes advantage of the property that the evolution of the region behind a "propagation front" is monotonely increasing, the so called the "causality" property. In general, the fast marching method implemented in a d-dimensional grid with M points requires a number of arithmetic operations in the order of $K_d M \log M$, in which the constant $K_d \in [2d, D^d]$ depends on the type of discrete neighborhood that is considered (and D is the maximal diameter of neighborhoods).

Although the fast-marching is a fast, single pass, method, it remains a grid-based method, and hence still suffers from the "curse of dimensionality". Indeed, the number of the grid nodes grows exponential with the dimension d, making the reading, writing, storing and computation untractable even on modern computers. Several types of discretizations or representations have been developed recently to overcome the curse of dimensionality for HJ equations. One may cite the sparse grids approximations of Bokanowski, Garcke, Griebel and Klompmaker [Bok+13], or of Kang and Wilcox [KW17], the tensor decompositions of Dolgov, Kalise and Kunisch [DKK21] or of Oster, Sallandt and Schneider [OSS22], the deep learning techniques applied by Nakamura-Zimmerer, Gong, and Kang [NGK21] or by Darbon and Meng [DM21]. In the case of structured problems, one may also cite the max-plus or tropical numerical method of McEneaney [McE06; McE07], see also [MDG08a; Qu13; Qu14a; Dow18] for further developments, and the Hopf formula of Darbon and Osher [DO16], see also [Cho+19; Kir+18; YD21a].

Another way to overcome the curse of dimensionality is to replace the general problem of solving the HJ equation by the one of computing the optimal trajectories between two given points. The latter problem can be solved, under some convexity assumptions, by the Pontryagin Maximum Principle approach [RZ98; RZ99; BZ99], which is normally done by searching for the zero of a certain shooting function (the dimensionality of the systems to be solved for such a shooting method is normally $2d$). Another method is the stochastic dual dynamic programming (SDDP) [PP91; Sha11; GLP15], in which the value function is approximated by a finite supremum of affine maps, and thus can be computed efficiently by linear programming solvers. In the absence of convexity assumptions, these methods may only lead to a local minimum. In that case, more recent methods consist in exploiting the structure of the problem, in order to reduce the set of possible trajectories among which the optimization is done. For instance, Alla, Falcone and Saluzzi [AFS19] (see also [AFS20]) introduced a tree structure discretization, taking advantage of the Lipschitz continuity of the value function. Also, Bokanowski, Gammoudi, and Zidani [BGZ22] introduced an adaptive discretization in the control space, which has shown to be efficient when the dimension of control space is low.

4.1.2 Contribution

In this chapter, we intend to find the optimal trajectories between two given sets, for the minimal time problem. We develop an adaptive multi-level and causal discretization of the state space, leading to a new algorithm.

Our method is inspired by the recent development of the "Highway Hierarchies" algorithm [Del+06; SS12] for the (discrete) shortest path problems. The latter algorithm accelerates the Dijkstra's algorithm, to compute the shortest path between any two given points. It first performs a pre-processing computing "highways" in which the optimal paths should go through,

then computing the shortest path between two given points using such highways. The highways are themselves computed using a partial application of the Dijkstra's algorithm. By doing so, one can find the exact shortest path, as by using Dijkstra's algorithm, but in a much shorter time. However, the original highway hierarchy method is difficult to implement in the case of a discretized HJ equation, even when this equation is associated to a shortest path problem. Indeed regularity properties may prevent the existence of highways in an exact sense. Also the computation and storage complexity of the pre-processing procedure (computing highways) is equivalent to the one of Dijkstra's algorithm, making the highway hierarchy method being more suitable to problems in which numerous queries have to be solved quickly, on a fixed graph (a typical use case is GPS path planning). We shall mention the works in [PC08; CCV14], which draw upon somehow similar inspiration of the present work, that is to extend the acceleration methods for Dijkstra's algorithm to continuous case.

Our approach combines the idea of highway hierarchy with the one of multi-level grids, in order to reduce the search space. Indeed, we compute (approximate) "highways" by using a coarse grid. Then, we perform a fast marching in a finer grid, restricted to a neigborhood of the highways. This method is iterated, with finer and finer grids, and smaller and smaller neighborhoods, untill the desired accuracy is reached.

We show that, by using our algorithm, the final approximation error is as good as the one obtained by discretizing directly the whole domain with the finest grid. Moreover, the number of elementary operations and the size of the memory needed to get an error of ε are considerably reduced. Indeed, recall that the number of arithmetic operations of conventional grid-based methods is in the order of $\widetilde{O}(\varepsilon^{-\frac{d}{\gamma}} K_d)$, where d is the dimension of the problem, $0 < \gamma \leqslant 1$ is the convergence rate of the classical fast marching method, $K_d \in [2d, L^d]$ for some constant L depending on the diameter of discrete neighborhoods, and $\widetilde{O}(x)$ ignores the logarithmic factors. For our multi-level method, with suitable parameters, the number of arithmetic operations is in the order of $\widetilde{O}(C^d \varepsilon^{-\frac{1+(d-1)(1-\gamma\beta)}{\gamma}})$, where $C > 1$ is a constant depending on the problem characteristics, and $0 < \beta \leqslant 1$ measures the "stiffness" of the value function around optimal trajectories. Then, our complexity bound reduces to $\widetilde{O}(C^d \varepsilon^{-1})$ when $\gamma = \beta = 1$. Hence, considering the dependence in ε only, we reduce the complexity bound from $\widetilde{O}(\varepsilon^{-\frac{d}{\gamma}})$ to $\widetilde{O}(\varepsilon^{-\frac{1+(d-1)(1-\gamma\beta)}{\gamma}})$. Moreover, under some regularity assumptions, the complexity bound becomes $\widetilde{O}(\varepsilon^{-1})$ and is thus of same order as for one-dimensional problems.

This chapter is organized as follows: In Section 4.2, we give some preliminary results on the HJ equation and the minimum time optimal control problem. In Section 4.3, we present our original idea from the continuous point of view, and give some results which will be useful to prove the correctness of our algorithm. In Section 4.4, we present our algorithm, from the discretization method to the implementation. We also describe a specific data storage structure, with hashing techniques, which is essential to implement our algorithm in an efficient way. In Section 4.5, we give the computational complexity of our new algorithm, by providing error bound. Finally, in Section 4.6, we present numerical tests, for problems of dimension 2 to 7.

4.2 Hamilton-Jacobi equation for the Minimum Time Problem

4.2.1 The Minimum Time Problem

Let Ω be an open, bounded domain in $\mathbb{R}^d$. Let S_1 be the unit sphere in $\mathbb{R}^d$, i.e., $S_1 = \{x \in \mathbb{R}^d, \|x\| = 1\}$ where $\|\cdot\|$ denotes the Euclidean norm. Let $\mathcal{A} = \{\alpha : \mathbb{R}_{\geqslant 0} \mapsto S_1 : \alpha(\cdot) \text{ is measurable}\}$ denote the set of controls, every $\alpha \in \mathcal{A}$ is then the unit vector determining the direction of motion. We denote by f the speed function, and assume the following basic

regularity properties:

Assumption (A3)

(i) $f : \overline{\Omega} \times S_1 \mapsto \mathbb{R}_{>0}$ is continuous.

(ii) f is bounded on $\overline{\Omega} \times S_1$, i.e., $\exists M_f > 0$ s.t. $\|f(x,\alpha)\| \leqslant M_f$, $\forall x \in \overline{\Omega}, \forall \alpha \in S_1$.

(iii) There exists constants $L_f, L_{f,\alpha} > 0$ such that $|f(x,\alpha) - f(x^{'},\alpha)| \leqslant L_f|x - x^{'}|, \forall \alpha \in S_1, \forall x, x^{'} \in \Omega$ and $|f(x,\alpha) - f(x,\alpha^{'})| \leqslant L_{f,\alpha}|\alpha - \alpha^{'}|, \forall x \in \Omega, \forall \alpha, \alpha^{'} \in S_1$.

It is worth noting that the original fast marching method presented in [Tsi95; Set96] focuses on cases where $f(x,\alpha) = f(x)$, meaning the speed function does not depend on the direction, a property known as "isotropy". Furthermore, Assumption (A3) applies to both Riemannian and Finslerian geometry but excludes sub-Riemannian systems and, more broadly, the dynamics of nonholonomic systems.

Let $\mathcal{K}_{\text{src}}$ and $\mathcal{K}_{\text{dst}}$ be two disjoint compact subsets of Ω (called the *source* and the *destination* resp.). Our goal is to find the minimum time necessary to travel from $\mathcal{K}_{\text{src}}$ to $\mathcal{K}_{\text{dst}}$, and the optimal trajectories between $\mathcal{K}_{\text{src}}$ and $\mathcal{K}_{\text{dst}}$, together with the optimal control α.

We intend to solve the following optimal control problem:

$$\begin{aligned} &\inf\ \tau \geqslant 0 \\ &s.t. \begin{cases} \dot{y}(t) = f(y(t),\alpha(t))\alpha(t),\ \forall t \in [0,\tau]\ , \\ y(0) \in \mathcal{K}_{\text{src}},\ y(\tau) \in \mathcal{K}_{\text{dst}}\ , \\ y(t) \in \overline{\Omega},\ \alpha(t) \in \mathcal{A},\ \forall t \in [0,\tau]\ . \end{cases} \end{aligned} \tag{4.1}$$

4.2.2 HJ Equation for the Minimum Time Problem.

A well known sufficient and necessary optimality condition for the above problem is given by the Hamilton-Jacobi-Bellman equation, which is deduced from the dynamic programming principle. Indeed, one can consider the following controlled dynamical system:

$$\begin{cases} \dot{y}(t) = f(y(t),\alpha(t))\alpha(t),\ \forall t \geqslant 0\ , \\ y(0) = x\ . \end{cases} \tag{4.2}$$

We denote by $y_\alpha(x;t)$ the solution of the above dynamical system (4.2) with $\alpha \in \mathcal{A}$ and $y_\alpha(x;s) \in \overline{\Omega}$ for all $0 \leqslant s \leqslant t$. More precisely, we restrict the set of control trajectories so that the state y stays inside the domain $\overline{\Omega}$, i.e., we consider the following set of controls:

$$\mathcal{A}_{\Omega,x} := \{\alpha \in \mathcal{A} \mid y_\alpha(x;s) \in \overline{\Omega},\ \text{for all } s \geqslant 0\}\ , \tag{4.3}$$

and we further assume $A_{\Omega,x} \neq \emptyset$. In other words, the structure of the control set $A_{\Omega,x}$ is adapted to the state constraint " $y \in \overline{\Omega}$ ". Let us define the cost functional by:

$$J_{\to \text{d}}(\alpha(\cdot),x) = \inf\{\tau \geqslant 0 \mid y_\alpha(x;\tau) \in \mathcal{K}_{\text{dst}}\}\ , \tag{4.4}$$

"$_{\to \text{d}}$" means arrival to destination. The value function $T_{\to \text{d}} : \bar{\Omega} \mapsto \mathbb{R} \cup \{+\infty\}$ is defined by

$$T_{\to \text{d}}(x) = \inf_{\alpha \in \mathcal{A}_{\Omega,x}} J_{\to \text{d}}(\alpha(\cdot),x)\ . \tag{4.5}$$

Then, restricted to $\overline{\Omega \setminus \mathcal{K}_{\text{dst}}}$, informally $T_{\to \text{d}}$ is a solution of the following state constrained Hamilton-Jacobi-Bellman equation:

$$\begin{cases} -\left(\min_{\alpha \in S_1}\{(\nabla T_{\to \text{d}}(x) \cdot \alpha) f(x,\alpha)\} + 1\right) = 0, & x \in \Omega \setminus \mathcal{K}_{\text{dst}} \ , \\ -\left(\min_{\alpha \in S_1}\{(\nabla T_{\to \text{d}}(x) \cdot \alpha) f(x,\alpha)\} + 1\right) \geqslant 0, & x \in \partial\Omega \ , \\ T_{\to \text{d}}(x) = 0 & x \in \partial\mathcal{K}_{\text{dst}} \ . \end{cases} \tag{4.6}$$

In order to relate the minimum time function and the HJB equation, we use the following definition for the viscosity solution of the following state constrained HJ equation with continuous hamiltonian $F : \mathbb{R}^d \times \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$, open bounded domain $\mathcal{O} \subset \mathbb{R}^d$, and "target" $\partial_{\text{t}}\mathcal{O} \subset \partial\mathcal{O}$:

$$\begin{cases} F_0(x, u(x), Du(x)) = 0, & x \in \mathcal{O} \ , \\ F_0(x, u(x), Du(x)) \geqslant 0, & x \in \partial\mathcal{O} \setminus (\partial_{\text{t}}\mathcal{O}) \ , \\ u(x) = 0, & x \in \partial_{\text{t}}\mathcal{O} \ . \end{cases} \qquad SC(F_0, \mathcal{O}, \partial_{\text{t}}\mathcal{O})$$

When there is no target set, that is $\partial_{\text{t}}\mathcal{O} = \emptyset$, the following definition corresponds to the one introduced first by Soner in [Son86a] (see also [BC08]), the case with a nonempty closed target set $\partial_{\text{t}}\mathcal{O}$ is inspired by the results of [CL90].

Definition 4.2.1 (compare with [Son86a; BC08; CL90])**.** Let $u : \overline{\mathcal{O}} \to \mathbb{R}$ be continuous.

(i) The function u is a viscosity subsolution of $(SC(F_0, \mathcal{O}, \partial_{\text{t}}\mathcal{O}))$ if for every test function $\psi \in \mathcal{C}^1(\overline{\mathcal{O}})$, for all local maximum points $x_0 \in \overline{\mathcal{O}}$ of the function $u - \psi$, we have:

$$\begin{cases} F(x_0, u(x_0), D\psi(x_0)) \leqslant 0 & \text{if } x_0 \in \mathcal{O} \ , \\ u(x_0) \leqslant 0 & \text{if } x_0 \in \partial_{\text{t}}\mathcal{O} \ . \end{cases}$$

(ii) The function u is a viscosity supersolution of $(SC(F_0, \mathcal{O}, \partial_{\text{t}}\mathcal{O}))$ if for every test function $\psi \in \mathcal{C}^1(\overline{\mathcal{O}})$, for all local minimum points $x_0 \in \overline{\mathcal{O}}$ of the function $u - \psi$, we have:

$$\begin{cases} F(x_0, u(x_0), D\psi(x_0)) \geqslant 0 & \text{if } x_0 \notin \partial_{\text{t}}\mathcal{O} \ , \\ u(x_0) \geqslant 0 & \text{otherwise.} \end{cases}$$

(iii) The function u is a viscosity solution of $(SC(F_0, \mathcal{O}, \partial_{\text{t}}\mathcal{O}))$ if and only if it is a viscosity subsolution and supersolution of $(SC(F_0, \mathcal{O}, \partial_{\text{t}}\mathcal{O}))$.

A basic method in the studies of the above system (see [Vla06], [Bar89], [BC08, Chapter-IV]) is the change of variable:

$$v_{\to \text{d}}(x) = 1 - e^{-T_{\to \text{d}}(x)} \ , \tag{4.7}$$

which was first used by Kruzkov [Kru75]. By doing so, $v_{\to \text{d}}(x)$ is automatically bounded and Lipschitz continuous. Once $v_{\to \text{d}}$ is computed, we can directly get the minimum time for x by $T_{\to \text{d}}(x) = -\log(1 - v_{\to \text{d}}(x))$.

In fact, consider a new control problem associated to the dynamical system (4.2), and the discounted cost functional defined by

$$J'_{\to \text{d}}(\alpha(\cdot), x) = \inf\left\{ \int_0^\tau e^{-t} dt \mid \tau \geqslant 0, \ y_\alpha(x;\tau) \in \mathcal{K}_{\text{dst}} \right\} \ , \tag{4.8}$$

for $\alpha \in \mathcal{A}_{\Omega,x}$. Then, the value function v of the control problem given by

$$v(x) = \inf_{\alpha \in \mathcal{A}_{\Omega,x}} J'_{\to \text{d}}(\alpha(\cdot), x) \tag{4.9}$$

coincides with $v_{\to \mathrm{d}}$ in (4.7). Let now

$$F(x,r,p) = -\min_{\alpha\in S_1}\{p\cdot f(x,\alpha)\alpha + 1 - r\} \ . \tag{4.10}$$

This Hamiltonian corresponds to the new control problem (4.2,4.8,4.9), and the restriction of the value function $v_{\to \mathrm{d}}$ to $\overline{\Omega\setminus\mathcal{K}_{\mathrm{dst}}}$ is a viscosity solution of the state constrained HJ equation $SC(F,\Omega\setminus\mathcal{K}_{\mathrm{dst}},\partial\mathcal{K}_{\mathrm{dst}})$.

The uniqueness of the solution of Equation $SC(F,\Omega\setminus\mathcal{K}_{\mathrm{dst}},\partial\mathcal{K}_{\mathrm{dst}})$ in the viscosity sense and the equality of this solution with the value function need not hold if the boundary condition is not well defined. When the target set is empty, Soner [Son86a; Son86b] introduced sufficient conditions for the uniqueness of the viscosity solution of $(SC(F_0,\mathcal{O},\partial_{\mathrm{t}}\mathcal{O}))$ and the equality with the corresponding value function. One of these conditions involves the dynamics of the controlled process on $\partial\mathcal{O}$, see [Son86a, (A3)], which is automatically satisfied when F is as in (4.10), and f satisfies Assumption (A3) with $\mathcal{O}$ instead of Ω. Similar conditions are proposed in [CL90]. We state below the result of [CL90] with the remaining conditions, and for a general open bounded domain $\mathcal{O}$, instead of Ω, as we shall need such a result in the sequel.

Theorem 4.2.2 (Corollary of [CL90, Th. IX.1, IX.3 and X.2], see also [Son86a])**.** *Let $\mathcal{O}$ be an open domain of $\mathbb{R}^d$, let $\partial_{\mathrm{t}}\mathcal{O}\subset\partial\mathcal{O}$ be compact, and assume that $\partial\mathcal{O}\setminus\partial_{\mathrm{t}}\mathcal{O}$ is of class $\mathcal{C}^1$. Let F be as in* (4.10) *with f satisfying Assumption (A3) with $\mathcal{O}$ instead of Ω. Then the comparison principle holds for $SC(F,\mathcal{O},\partial_{\mathrm{t}}\mathcal{O})$, i.e., any viscosity subsolution is upper bounded by any viscosity supersolution. In particular, the viscosity solution is unique. Moreover, it coincides with the value function $v_{\to \mathrm{d}}$ in* (4.9) *of the optimal control problem with dynamics* (4.2) *and criteria* (4.8)*, in which Ω and $\mathcal{K}_{dst}$ are replaced by $\mathcal{O}$ and $\partial_{\mathrm{t}}\mathcal{O}$, respectively.*

We should also mention the recent works of [BFZ10; BFZ11; HWZ18] which characterized the value function of the state constrained problems without any controllability assumptions.

Once $SC(F,\Omega\setminus\mathcal{K}_{\mathrm{dst}},\partial\mathcal{K}_{\mathrm{dst}})$ is solved, one can easily get the value of the original minimum time problem by computing the minimum of $v_{\to \mathrm{d}}(x)$ over $\mathcal{K}_{\mathrm{src}}$. We shall denote the set of minimum points by $\mathcal{X}_{\mathrm{src}}$, i.e.,

$$\mathcal{X}_{\mathrm{src}} = \mathrm{Argmin}_{x\in\mathcal{K}_{\mathrm{src}}} v_{\to \mathrm{d}}(x) \ . \tag{4.11}$$

Since $v_{\to \mathrm{d}}$ is continuous (by Theorem 4.2.2) and $\mathcal{K}_{\mathrm{src}}$ is compact, we get that $\mathcal{X}_{\mathrm{src}}$ is a nonempty compact set.

4.2.3 HJ Equation in Reverse Direction.

We shall also use another equivalent optimality condition characterization for the minimum time problem (4.1), obtained by applying the dynamic programming principle in a reverse direction.

Let us consider the following controlled dynamical system:

$$\begin{cases} \dot{\tilde{y}}(t) = -f(\tilde{y}(t),\tilde{\alpha}(t))\tilde{\alpha}(t), \ \forall t\geqslant 0 \ , \\ \tilde{y}(0) = x \ . \end{cases} \tag{4.12}$$

We denote by $\tilde{y}_{\tilde{\alpha}}(x;t)$ the solution of the above dynamical system (4.12) with $\tilde{\alpha}(t)=\alpha(\tau-t)\in\mathcal{A}$, for all $t\in[0,\tau]$. Then automatically $\tilde{y}(t)=y(\tau-t)$ with y as in (4.2). Let us denote the state constrained control trajectories for this new problem by $\tilde{\mathcal{A}}_{\Omega,x}$, i.e.,

$$\tilde{\mathcal{A}}_{\Omega,x} = \{\tilde{\alpha}\in\mathcal{A} \mid \tilde{y}_{\tilde{\alpha}}(x;s)\in\overline{\Omega}, \text{ for all } s\geqslant 0\} \ . \tag{4.13}$$

Consider the following cost functional:

$$J_{\mathrm{s}\to}(\tilde{\alpha}(\cdot),x) = \inf\{\tau\geqslant 0 \mid \tilde{y}_{\tilde{\alpha}}(x;\tau)\in\mathcal{K}_{\mathrm{src}}\} \ , \tag{4.14}$$

where "$_{\text{s}\to}$" means from source, and the value function is given by

$$T_{\text{s}\to}(x) = \inf_{\tilde{\alpha}\in\tilde{\mathcal{A}}_{\Omega,x}} J_{\text{s}\to}(\tilde{\alpha}(\cdot), x) \in \mathbb{R}\cup\{+\infty\} \ . \tag{4.15}$$

Then, the restriction of $T_{\text{s}\to}$ to $\overline{\Omega\setminus\mathcal{K}_{\text{src}}}$ is a viscosity solution of the following state constrained HJ equation:

$$\begin{cases} -\left(\min_{\alpha\in S_1}\{-(\nabla T_{\text{s}\to}(x)\cdot\tilde{\alpha})f(x,\tilde{\alpha})\}+1\right)=0, & x\in\Omega\setminus\mathcal{K}_{\text{src}} \ , \\ -\left(\min_{\alpha\in S_1}\{-(\nabla T_{\text{s}\to}(x)\cdot\tilde{\alpha})f(x,\tilde{\alpha})\}+1\right)\geqslant 0, & x\in\partial\Omega \ , \\ T_{\text{s}\to}(x)=0, & x\in\partial\mathcal{K}_{\text{src}} \ . \end{cases} \tag{4.16}$$

Using the same change of variable technique, we have $v_{\text{s}\to}(x) = 1-e^{-T_{\text{s}\to}(x)}$, and we transform the above system (4.16) into the state constrained HJ equation $SC(F^*, \Omega\setminus\mathcal{K}_{\text{src}}, \partial\mathcal{K}_{\text{src}})$, where $F^*(x,r,p) = F(x,r,-p)$. Notice that $SC(F^*, \Omega\setminus\mathcal{K}_{\text{src}}, \partial\mathcal{K}_{\text{src}})$ is also associated to a new optimal control problem, for which the dynamics is given by (4.12) and the value function is given by

$$v_{\text{s}\to}(x) = \inf_{\tilde{\alpha}\in\tilde{\mathcal{A}}_{\Omega,x}} \inf\left\{\int_0^\tau e^{-t}dt \mid \tau\geqslant 0,\ \tilde{y}_{\tilde{\alpha}}(x;\tau)\in\mathcal{K}_{\text{src}}\right\} \ . \tag{4.17}$$

By doing so, to solve the original minimum time problem (4.1), one can also solve the equation $SC(F^*, \Omega\setminus\mathcal{K}_{\text{src}}, \partial\mathcal{K}_{\text{src}})$ to get $v_{\text{s}\to}$, and then compute the minimum of $v_{\text{s}\to}(x)$ over $\mathcal{K}_{\text{dst}}$. We shall also denote by $\mathcal{X}_{\text{dst}}$ the set of minimum points, i.e.,

$$\mathcal{X}_{\text{dst}} = \text{Argmin}_{x\in\mathcal{K}_{\text{dst}}} v_{\text{s}\to}(x) \ . \tag{4.18}$$

Again, as for $\mathcal{X}_{\text{src}}$, we get that $\mathcal{X}_{\text{dst}}$ is a nonempty compact set.

4.3 Reducing the State Space of the Continuous Space Problem

The above two equivalent characterizations of the minimum time between $\mathcal{K}_{\text{src}}$ and $\mathcal{K}_{\text{dst}}$ give us an inspiration to formulate optimal trajectories between $\mathcal{K}_{\text{src}}$ and $\mathcal{K}_{\text{dst}}$ by using the value functions from the two directions. In this section, we shall show how to reduce the state space Ω of the original minimum time problem, using $v_{\text{s}\to}$ and $v_{\to\text{d}}$, while preserving optimal trajectories.

4.3.1 The Optimal Trajectory

We first give the definition of an optimal trajectory:

Definition 4.3.1. For every $x\in\overline{\Omega}$, We say that $y_{\alpha^*}(x;\cdot):[0,\tau]\mapsto\overline{\Omega}$ is an optimal trajectory with associated optimal control α^* for Problem (4.2,4.8,4.9), if the minimum in (4.9) is achieved in α^*. We denote by Γ_x^* the set of *geodesic points* starting from x, i.e.,

$$\Gamma_x^* = \{y_{\alpha^*}(x;t) \mid t\in[0,\tau],\ \alpha^* \text{ optimal }\} \ .$$

Remark 4.3.2. We can use the same method to define the optimal trajectory for the problem in reverse direction as defined in Section 4.2.3. Moreover, we denote $\tilde{\Gamma}_x^* = \{\tilde{y}_{\tilde{\alpha}^*}(x;t) \mid t\in[0,\tau],\ \tilde{\alpha}^* \text{ optimal }\}$ the set of geodesic points starting from x in the reverse direction.

Proposition 4.3.3. *We have*

$$\cup_{x\in\mathcal{X}_{src}}\{\Gamma_x^*\} = \cup_{x\in\mathcal{X}_{dst}}\{\tilde{\Gamma}_x^*\} \ ,$$

and if the latter set is nonempty, then

$$\inf_{x\in\mathcal{K}_{src}} v_{\to\text{d}}(x) = \inf_{x\in\mathcal{K}_{dst}} v_{\text{s}\to}(x) \ .$$

Proof. Let $y_{\alpha^*}(x_{\rm src};\cdot):[0,\tau^*]\mapsto\overline{\Omega}$ be an optimal trajectory for the problem (4.2,4.8,4.9), with $x_{\rm src}\in\mathcal{K}_{\rm src}$ and the optimal control α^*. Let us denote $x_{\rm dst}:=y_{\alpha^*}(x_{\rm src};\tau^*)\in\mathcal{K}_{\rm dst}$. Consider the problem in reverse direction starting at $x_{\rm dst}$, and the control $\tilde{\alpha}^*$ such that $\tilde{\alpha}^*(s)=\alpha^*(\tau^*-s)$, $\forall s\in[0,\tau^*]$. Then, the associated state at time s is $\tilde{y}_{\tilde{\alpha}^*}(x_{\rm dst};s)=y_{\alpha^*}(x_{\rm src};\tau^*-s)$. In particular $\tilde{y}_{\tilde{\alpha}^*}(x_{\rm dst};\tau^*)=x_{\rm src}\in\mathcal{K}_{\rm src}$, and the trajectory $\tilde{y}_{\tilde{\alpha}^*}(x_{\rm dst};\cdot)$ arrives in $\mathcal{K}_{\rm src}$ at time τ^*. By definition of the value function, we have:

$$v_{\rm s\to}(x_{\rm dst})\leqslant\int_0^{\tau^*}e^{-s}ds=v_{\to\rm d}(x_{\rm src})\ , \tag{4.19}$$

with equality if and only if $\tilde{\alpha}^*$ is optimal.

Let us assume that $\cup_{x\in\mathcal{X}_{\rm src}}\{\Gamma^*_x\}$ is nonempty, and take $x_{\rm src}\in\mathcal{X}_{\rm src}$, such that $\Gamma^*_{x_{\rm src}}$ is nonempty, we get

$$v_{\rm s\to}(x_{\rm dst})\leqslant v_{\to\rm d}(x_{\rm src})=\inf_{x_{\rm src}\in\mathcal{K}_{\rm src}}v_{\to\rm d}(x_{\rm src})\ . \tag{4.20}$$

If the above inequality is strict, then there exists a trajectory (not necessary optimal) $\tilde{y}_{\tilde{\alpha}'}(x_{\rm dst};\cdot)$ starting from $x_{\rm dst}$ and arriving in $x'_{\rm src}\in\mathcal{K}_{\rm src}$ at time $\tau'<\tau^*$. Then, the reverse trajectory $y_{\alpha'}(x'_{\rm src};\cdot)$ is starting from $x'_{\rm src}$ and arrives in $x_{\rm dst}$ at time τ', and we get $v_{\to\rm d}(x'_{\rm src})=1-e^{-\tau'}<v_{\to\rm d}(x_{\rm src})$ which is impossible. This shows the equality in (4.20) and that $\tilde{\alpha}^*$ is optimal, so $\tilde{\Gamma}^*_{x_{\rm dst}}$ is nonempty. Also, if $x_{\rm dst}\notin\mathcal{X}_{\rm dst}$, by the same construction applied to $x'_{\rm dst}\in\mathcal{X}_{\rm dst}$, we get a contradiction, showing that $x_{\rm dst}\in\mathcal{X}_{\rm dst}$. Hence $\cup_{x\in\mathcal{X}_{\rm dst}}\{\tilde{\Gamma}^*_x\}$ is nonempty, and $y_{\alpha^*}(x_{\rm src};s)=\tilde{y}_{\tilde{\alpha}^*}(x_{\rm dst};\tau^*-s)\in\cup_{x\in\mathcal{X}_{\rm dst}}\{\tilde{\Gamma}^*_x\}$. Since this holds for all optimal trajectories y_{α^*} starting in any $x_{\rm src}\in\mathcal{X}_{\rm src}$, we deduce that $\cup_{x\in\mathcal{X}_{\rm src}}\{\Gamma^*_x\}\subset\cup_{x\in\mathcal{X}_{\rm dst}}\{\tilde{\Gamma}^*_x\}$. By symmetry, we obtain the equality, so the first equality of the proposition. Moreover, by the equality in (4.20), we also get the second equality of the proposition. □

Remark 4.3.4. Note that in Proposition 4.3.3, all the sets Γ^*_x with $x\in\mathcal{X}_{\rm src}$ may be empty, which would imply that all the sets $\tilde{\Gamma}^*_x$ with $x\in\mathcal{X}_{\rm dst}$ are empty. In that case, we need to replace the sets Γ^*_x and $\tilde{\Gamma}^*_x$ by δ-geodesic sets.

From now on, we set $\Gamma^*=\cup_{x\in\mathcal{X}_{\rm src}}\{\Gamma^*_x\}=\cup_{x\in\mathcal{X}_{\rm dst}}\{\tilde{\Gamma}^*_x\}$, and call it *the set of geodesic points from $\mathcal{K}_{src}$ to $\mathcal{K}_{dst}$*. When Γ^* is nonempty, using Proposition 4.3.3, we shall denote by v^* the following value:

$$v^*:=\inf_{x\in\mathcal{K}_{\rm src}}v_{\to\rm d}(x)=\inf_{x\in\mathcal{K}_{\rm dst}}v_{\rm s\to}(x)\ .$$

Once v^* is obtained, we can directly get the minimum time by $\tau^*=-\log(1-v^*)$.

Lemma 4.3.5. *Assume Γ^* is non-empty. Then, we have*

$$v^*=\inf_{y\in\overline{\Omega}}\{v_{\rm s\to}(y)+v_{\to\rm d}(y)-v_{\rm s\to}(y)v_{\to\rm d}(y)\}\ , \tag{4.21}$$

and the infimum is attained for every $x\in\Gamma^*$. Moreover, if there exists an optimal trajectory between any two points of $\overline{\Omega}$, then x is optimal in (4.21)*, that is $(v_{\rm s\to}(x)+v_{\to\rm d}(x)-v_{\rm s\to}(x)v_{\to\rm d}(x))=v^*$, if and only if $x\in\Gamma^*$.*

Proof. Fix $x\in\overline{\Omega}$. By definition of the value function, we have:

$$v_{\rm s\to}(x)=\inf_{\substack{\tau>0,\tilde{\alpha}\in\tilde{\mathcal{A}}_{\Omega,x}\\ \tilde{y}_{\tilde{\alpha}}(x;\tau)\in\mathcal{K}_{\rm src}}}\left\{\int_0^{\tau}e^{-t}dt\right\},\ \ v_{\to\rm d}(x)=\inf_{\substack{\tau'>0,\alpha'\in\mathcal{A}_{\Omega,x}\\ y_{\alpha'}(x;\tau')\in\mathcal{K}_{\rm dst}}}\left\{\int_0^{\tau'}e^{-t}dt\right\}\ .$$

For any $\tau > 0$ and $\tilde{\alpha} \in \tilde{\mathcal{A}}_{\Omega,x}$ s.t. $\tilde{y}_{\tilde{\alpha}}(x;\tau) \in \mathcal{K}_{\rm src}$, denote $x_{\rm src} = \tilde{y}_{\tilde{\alpha}}(x;\tau) \in \mathcal{K}_{\rm src}$. Then, by the simple change of variable $s = \tau - t$ and $\alpha(s) = \tilde{\alpha}(\tau - t)$, we have $\alpha \in \mathcal{A}_{\Omega,x_{\rm src}}$ and $y_\alpha(x_{\rm src};\tau) = x$. This implies

$$v_{\rm s\to}(x) = \inf_{\substack{\tau>0, x_{\rm src}\in\mathcal{K}_{\rm src}, \alpha\in\mathcal{A}_{\Omega,x_{\rm src}} \\ y_\alpha(x_{\rm src};\tau)=x}} \left\{ \int_0^\tau e^{-s} ds \right\} \ . \tag{4.22}$$

Moreover, for $\tau' > 0$ and $\alpha' \in \mathcal{A}_{\Omega,x}$ s.t. $y_{\alpha'}(x;\tau') \in \mathcal{K}_{\rm dst}$, we denote $x_{\rm dst} = y_{\alpha'}(x;\tau') \in \mathcal{K}_{\rm dst}$, so that

$$v_{\to\rm d}(x) = \inf_{\substack{\tau'>0, x_{\rm dst}\in\mathcal{K}_{\rm dst}, \alpha'\in\mathcal{A}_{\Omega,x} \\ y_{\alpha'}(x;\tau')=x_{\rm dst}}} \left\{ \int_0^{\tau'} e^{-t} dt \right\} \ . \tag{4.23}$$

Let $\tau > 0, x_{\rm src} \in \mathcal{K}_{\rm src}, \alpha \in \mathcal{A}_{\Omega,x_{\rm src}}$ be such that $y_\alpha(x_{\rm src};\tau) = x$ and $\tau' > 0, x_{\rm dst} \in \mathcal{K}_{\rm dst}, \alpha' \in \mathcal{A}_{\Omega,x}$ be such that $y_{\alpha'}(x;\tau') = x_{\rm dst}$. Concatenating α stopped at time τ and $t \in [\tau,\infty) \mapsto \alpha'(t-\tau)$, we obtain $\alpha'' \in \mathcal{A}_{\Omega,x_{\rm src}}$ such that $y_{\alpha''}(x_{\rm src};\tau+\tau') = x_{\rm dst}$ and the trajectory from $x_{\rm src}$ to $x_{\rm dst}$ is going through x at time τ. Then,

$$\int_0^\tau e^{-s} ds + e^{-\tau} \int_0^{\tau'} e^{-t} dt = \int_0^{\tau+\tau'} e^{-s} ds \geqslant v^* \ . \tag{4.24}$$

where the last inequality comes from the second equality in Proposition 4.3.3.

For any $\lambda, \mu \in [0,1]$, rewriting $\lambda + \mu - \lambda\mu = \lambda + (1-\lambda)\mu$ or $\mu + (1-\mu)\lambda$, we see that the map $[0,1]^2 \to \mathbb{R},\ (\lambda,\mu) \mapsto \lambda + \mu - \lambda\mu$ is nondecreasing in each of its variables (separately), and thus commutes with the infimum operation in each variable. Using this property, and that $0 \leqslant \int_0^\tau e^{-s} ds \leqslant 1$ and $1 - \int_0^\tau e^{-s} ds = e^{-\tau}$, for all $\tau > 0$, and taking the infimum in (4.24), first over τ and then over τ', we deduce:

$$v_{\rm s\to}(x) + v_{\to\rm d}(x) - v_{\rm s\to}(x) v_{\to\rm d}(x) \geqslant v^* \ .$$

Since the above inequality is an equality for $x \in \mathcal{X}_{\rm dst}$ or $x \in \mathcal{X}_{\rm src}$, we deduce (4.21).

If $x \in \Gamma^*$, there exist $x_{\rm src} \in \mathcal{K}_{\rm src}$, $x_{\rm dst} \in \mathcal{K}_{\rm dst}$ and $\alpha \in \mathcal{A}_{\Omega,x_{\rm src}}$ such that $y_\alpha(x_{\rm src};\tau^*) = x_{\rm dst}$ and $y_\alpha(x_{\rm src};\tau) = x$ for some $0 \leqslant \tau \leqslant \tau^*$. Taking $\tau' = \tau^* - \tau$, we get an equality in (4.24), and using the nondecreasing property with respect to τ and τ', we deduce the reverse inequality $v^* \geqslant v_{\rm s\to}(x) + v_{\to\rm d}(x) - v_{\rm s\to}(x) v_{\to\rm d}(x)$, so the equality.

Let now $x \in \Omega$ be optimal in (4.21), that is satisfy $(v_{\rm s\to}(x) + v_{\to\rm d}(x) - v_{\rm s\to}(x) v_{\to\rm d}(x)) = v^*$. Assuming that there exists an optimal trajectory for each of the two minimum time problems starting from any point, there exist $\tau > 0, x_{\rm src} \in \mathcal{K}_{\rm src}, \alpha \in \mathcal{A}_{\Omega,x_{\rm src}}$ such that $y_\alpha(x_{\rm src};\tau) = x$ and $\tau' > 0, x_{\rm dst} \in \mathcal{K}_{\rm dst}, \alpha' \in \mathcal{A}_{\Omega,x}$ such that $y_{\alpha'}(x;\tau') = x_{\rm dst}$, which are optimal in the above infimum (4.22) and (4.23). So again concatenating α stopped at time $\tau = T_{\rm s\to}(x)$ and $t \in [\tau,\infty) \mapsto \alpha'(t-\tau)$, we obtain $\alpha'' \in \mathcal{A}_{\Omega,x_{\rm src}}$ and a trajectory from $x_{\rm src}$ to $x_{\rm dst}$ going through x at time τ and arriving at $x_{\rm dst}$ at time $\tau+\tau'$. Using (4.24), we get $v^* = (v_{\rm s\to}(x) + v_{\to\rm d}(x) - v_{\rm s\to}(x) v_{\to\rm d}(x)) = \int_0^{\tau+\tau'} e^{-s} ds$, so α'' is optimal, which shows $x \in \Gamma^*$. □

For easy expression, for every $x \in \overline{\Omega}$ and $v = (v_{\rm s\to}, v_{\to\rm d})$, we denote

$$\mathcal{F}_v(x) = v_{\rm s\to}(x) + v_{\to\rm d}(x) - v_{\rm s\to}(x) v_{\to\rm d}(x) \ . \tag{4.25}$$

4.3.2 Reduction of The State Space

Let us now consider the open subdomain $\mathcal{O}_\eta$ of Ω, determined by a parameter $\eta > 0$, and defined as follows:

$$\mathcal{O}_\eta = \{ x \in (\Omega \setminus (\mathcal{K}_{\rm src} \cup \mathcal{K}_{\rm dst})) \mid \mathcal{F}_v(x) < \inf_{y\in\Omega} \{\mathcal{F}_v(y) + \eta\} \ \} \ . \tag{4.26}$$

By Assumption (A3), we observe that $v_{s\to}$, $v_{\to d}$ are continuous in $\overline{\Omega}$, so does $\mathcal{F}_v$, thus the infimum in (4.26) is achieved by an element $y \in \overline{\Omega}$, and by Lemma 4.3.5, it is equal to $v^* + \eta$. This also implies that $\mathcal{O}_\eta$ is an open set. Since $\eta > 0$, we also have that $\mathcal{O}_\eta$ is always nonempty.

Proposition 4.3.6. *Under Assumption (A3), and assuming that* Γ^* *is nonempty, we have*

$$\mathcal{X}_{src} \subseteq (\partial\mathcal{O}_\eta) \cap (\partial\mathcal{K}_{src}),\ \mathcal{X}_{dst} \subseteq (\partial\mathcal{O}_\eta) \cap (\partial\mathcal{K}_{dst}),\ \textit{and}\ \Gamma^* \subset \overline{\mathcal{O}}_\eta \quad \forall \eta > 0\ .$$

Proof. Let us first notice that, since $\mathcal{F}_v$ is continuous, we have $\overline{\mathcal{O}}_\eta \supseteq \{x \in \overline{(\Omega \setminus (\mathcal{K}_{\rm src} \cup \mathcal{K}_{\rm dst}))} \mid \mathcal{F}_v(x) < \inf_{y\in\Omega}\{\mathcal{F}_v(y) + \eta\}$. Moreover, since $\mathcal{K}_{\rm src}$ and $\mathcal{K}_{\rm dst}$ are disjoint compact subsets of Ω, then $\partial\mathcal{O}_\eta \supseteq \{x \in \partial\Omega \cup \partial\mathcal{K}_{\rm src} \cup \partial\mathcal{K}_{\rm dst} \mid \mathcal{F}_v(x) < \inf_{y\in\Omega}\{\mathcal{F}_v(y) + \eta\}$.

By the dynamic programming principle, and since the cost in (4.14) is 1 (so positive), and $\mathcal{K}_{\rm src}$ and $\mathcal{K}_{\rm dst}$ are disjoint, we have

$$v_{\to d}(x) > \inf_{y\in\partial\mathcal{K}_{\rm src}} v_{\to d}(y)\,,$$

for all x in the interior of $\mathcal{K}_{\rm src}$ (any trajectory starting in x need to go through $\partial\mathcal{K}_{\rm src}$). This implies that Γ^* does not intersects the interior of $\mathcal{K}_{\rm src}$, and similarly Γ^* does not intersects the interior of $\mathcal{K}_{\rm dst}$, so $\Gamma^* \subseteq \overline{(\Omega \setminus (\mathcal{K}_{\rm src} \cup \mathcal{K}_{\rm dst}))}$. Moreover, $\mathcal{X}_{\rm src} \subseteq \mathcal{K}_{\rm src} \cap \Gamma^* \subseteq \partial\mathcal{K}_{\rm src}$ and $\mathcal{X}_{\rm dst} \subseteq \mathcal{K}_{\rm dst} \cap \Gamma^* \subseteq \partial\mathcal{K}_{\rm dst}$.

Now for all $x \in \Gamma^*$, we have $\mathcal{F}_v(x) = v^* = \inf_{y\in\overline{\Omega}} \mathcal{F}_v(y)$, so $x \in \overline{\mathcal{O}}_\eta$, showing that $\Gamma^* \subseteq \overline{\mathcal{O}}_\eta$. Let us now take $x \in \mathcal{X}_{\rm src}$. We already shown that $\mathcal{X}_{\rm src} \subseteq \partial\mathcal{K}_{\rm src}$, and we also have $\mathcal{X}_{\rm src} \subset \Gamma^*$, so $\mathcal{F}_v(x) = v^*$. All together, this implies that The same argument holds for $\mathcal{X}_{\rm dst}$. □

To apply the comparison principle (Theorem 4.2.2), we need to work with a domain with a $\mathcal{C}^1$ boundary. To this end, we assume the following assumption

Assumption (A4) Assume that Γ^* is nonempty and that $\Gamma^* \subset \Omega$.

Then, for every $\mu > 0$, we select a function $\mathcal{F}_v^\mu : \overline{\Omega} \to \mathbb{R}$, that is $\mathcal{C}^d$, and that approximates $\mathcal{F}$, i.e.,

$$\|\mathcal{F}_v^\mu - \mathcal{F}_v\|_\infty < \mu\ . \tag{4.27}$$

Let us also consider a domain $\mathcal{O}_\eta^\mu$, deduced from $\mathcal{F}_v^\mu$ and defined as follows:

$$\mathcal{O}_\eta^\mu = \{x \in (\Omega \setminus (\mathcal{K}_{\rm src} \cup \mathcal{K}_{\rm dst})) \mid \mathcal{F}_v^\mu(x) < \inf_{y\in\Omega}\{\mathcal{F}_v(y) + \eta\}\ \}\ , \tag{4.28}$$

with $\mu < \eta$. We notice that $\mathcal{O}_{\eta-\mu}^\mu \subseteq \mathcal{O}_\eta \subseteq \mathcal{O}_{\eta+\mu}^\mu$ with arbitrary small μ. Moreover $\overline{\mathcal{O}_\eta^\mu} \subset \Omega$, for η and μ small enough. Then, $\mathcal{O}_\eta^\mu$ can be compared with $\mathcal{O}_\eta$, and it is the strict sublevel set of the $\mathcal{C}^d$ function. It can then be seen as a regularization of $\mathcal{O}_\eta$. So, for almost all η (small enough) and μ small enough, $(\partial\mathcal{O}_\eta^\mu) \setminus (\mathcal{K}_{\rm dst} \cup \mathcal{K}_{\rm src})$ is $\mathcal{C}^1$ (Corollary of Morse-Sard Theorem, see [Mor39; Sar42]).

We shall see that $\mathcal{O}_\eta^\mu$ is a smooth neighborhood of optimal trajectories, and we intend to reduce the state space of our optimal control problem from $\overline{\Omega}$ to the closure $\overline{\mathcal{O}_\eta^\mu}$ of $\mathcal{O}_\eta^\mu$. More precisely, starting with the problem in direction "to destination", we consider a new optimal control problem with the same dynamics and cost functional as in Problem (4.2,4.8,4.9), but we restrict the controls so that the state $y(s)$ stays inside the domain $\overline{\mathcal{O}_\eta^\mu}$, $\forall s \geqslant 0$.

The reduction of the state space leads to a new set of controls:

$$\mathcal{A}_{\eta,x} := \{\alpha \in \mathcal{A} \mid y_\alpha(x;s) \in \overline{\mathcal{O}_\eta^\mu},\ \text{for all } s \geqslant 0\}\ . \tag{4.29}$$

Let $v_{\to d}^\eta(x)$ denote the value function of the optimal control problem when the set of controls is $\mathcal{A}_{\eta,x}$. Consider a new state constrained HJ equation: $SC(F, \mathcal{O}_\eta^\mu, (\partial\mathcal{O}_\eta^\mu) \cap (\partial\mathcal{K}_{\rm dst}))$, we have the following result:

Proposition 4.3.7 (Corollary of Theorem 4.2.2)**.** *The value function $v^{\eta}_{\to \mathrm{d}}$ of the control problem in $\overline{\mathcal{O}^{\mu}_{\eta}}$ is the unique viscosity solution of $SC(F, \mathcal{O}^{\mu}_{\eta}, (\partial\mathcal{O}^{\mu}_{\eta}) \cap (\partial\mathcal{K}_{dst}))$.*

Remark 4.3.8. The same reduction works for the problem in the reverse direction, which is the problem in the direction "from source". We denote $v^{\eta}_{\mathrm{s}\to}$ the value function of this problem with the set of controls be $\tilde{\mathcal{A}}_{\eta,x} := \{\tilde{\alpha} \in \mathcal{A} \mid \tilde{y}_{\tilde{\alpha}}(x; s) \in \overline{\mathcal{O}^{\mu}_{\eta}}, \text{ for all } s \geqslant 0\}$.

By the above construction of $\mathcal{O}^{\mu}_{\eta}$, we have the following relation between the value function of the original problem and the value function of the reduced problem:

Proposition 4.3.9. *If Γ^* is not empty, then $\Gamma^* \subseteq \overline{\mathcal{O}^{\mu}_{\eta}}$, and for all $x \in \Gamma^*$, we have:*

$$v_{\mathrm{s}\to}(x) = v^{\eta}_{\mathrm{s}\to}(x), \ v_{\to \mathrm{d}}(x) = v^{\eta}_{\to \mathrm{d}}(x) \ .$$

Proof. $\Gamma^* \subseteq \overline{\mathcal{O}^{\mu}_{\eta}}$ is a straightforward result of Proposition 4.3.6. Then, we have $v_{\mathrm{s}\to}(x) \leqslant v^{\eta}_{\mathrm{s}\to}(x), v_{\to \mathrm{d}}(x) \leqslant v^{\eta}_{\to \mathrm{d}}(x)$ for all $x \in \mathcal{O}^{\mu}_{\eta}$, since $\mathcal{O}^{\mu}_{\eta} \subseteq \Omega$. Then, we also have $v_{\mathrm{s}\to}(x) \geqslant v^{\eta}_{\mathrm{s}\to}(x), v_{\to \mathrm{d}}(x) \geqslant v^{\eta}_{\to \mathrm{d}}(x)$ for all $x \in \Gamma^*$, since there exists optimal trajectories from $x \in \Gamma^*$ staying in Γ^* and $\Gamma^* \subseteq \overline{\mathcal{O}^{\mu}_{\eta}}$. □

4.3.3 δ-optimal trajectories and the value function

The above results express properties of exact optimal trajectories. We will also consider approximate, $\delta-$optimal, trajectories. We first give the definition of the $\delta-$optimal trajectory.

Definition 4.3.10. For every $x \in \Omega$, we say $y_{\alpha^\delta}(x; \cdot) : [0, \tau] \to \overline{\Omega}$ is a $\delta-$optimal trajectory with associated δ-optimal control $\alpha^\delta : [0, \tau] \to S^1$ for the problem (4.2,4.8,4.9) if :

$$y_{\alpha^\delta}(x; \tau) \in \mathcal{K}_{\mathrm{dst}} \quad \text{and} \quad \int_0^\tau e^{-t} dt \leqslant v_{\to \mathrm{d}}(x) + \delta \ .$$

We denote by Γ^δ_x the set of $\delta-$geodesic points starting from x, i.e.,

$$\Gamma^\delta_x = \{y_{\alpha^\delta}(x; t) \mid t \in [0, \tau], \ \alpha^\delta : [0, \tau] \to S^1 \ \delta\text{-optimal} \ \} \ .$$

We define analogously δ-optimal trajectories for the problem in reverse direction, and denote by $\tilde{\Gamma}^\delta_x$ the set of δ-geodesic points starting from x in the reverse direction.

Following the same argument as in Proposition 4.3.3, we have the following result:

Proposition 4.3.11. *Let us denote*

$$\mathcal{X}^\delta_{src} = \{x \in \partial\mathcal{K}_{src} \mid v_{\to \mathrm{d}}(x) \leqslant v^* + \delta\}, \ \mathcal{X}^\delta_{dst} = \{x \in \partial\mathcal{K}_{dst} \mid v_{\mathrm{s}\to}(x) \leqslant v^* + \delta\} \ ,$$

then we have:

$$\cup_{\delta' \in [0,\delta]} \cup_{x \in \mathcal{X}^{\delta-\delta'}_{src}} \{\Gamma^{\delta'}_x\} = \cup_{\delta' \in [0,\delta]} \cup_{x \in \mathcal{X}^{\delta-\delta'}_{dst}} \{\tilde{\Gamma}^{\delta'}_x\}. \tag{4.30}$$

□

Let us denote the set in (4.30) by Γ^δ, and call it *the set of $\delta-$geodesic points from $\mathcal{K}_{src}$ to $\mathcal{K}_{dst}$*. In what follows, we intend to deduce the relationship between Γ^δ and our $\eta-$neighborhood, $\mathcal{O}_\eta$. Let us start with a property of the $\delta-$optimal trajectories.

Lemma 4.3.12. *Let $y_{\alpha^\delta}(x;\cdot) : [0, \tau_x^\delta] \to \overline{\Omega}$ be a $\delta-$optimal trajectory of Problem* (4.2,4.8,4.9) *with associated δ-optimal control α^δ and δ-optimal time τ_x^δ. Assume $v_{\to \mathrm{d}}(x) < 1$, i.e., the minimum time from x to $\mathcal{K}_{dst}$: $\tau_x < +\infty$. For every $z = y_{\alpha^\delta}(x; t_z)$, let us define a control $\alpha' : [0, \tau_x^\delta - t_z] \to S_1$ such that $\alpha'(s) = \alpha^\delta(s + t_z), \forall s \in [0, \tau_x^\delta - t_z]$. Then, the associated trajectory starting in z with control α', $y_{\alpha'}(z;\cdot) : [0, \tau_x^\delta - t_z] \to \overline{\Omega}$, is at least $(e^{t_z}\delta)$-optimal for the problem* (4.2,4.8,4.9) *with initial state z.*

Proof. By definition, we have $y_{\alpha^\delta}(x; \tau_x^\delta) \in \mathcal{K}_{\mathrm{dst}}$, and $\int_0^{\tau_x^\delta} e^{-t} dt \leqslant v_{\to \mathrm{d}}(x) + \delta$. Then, considering the control α' defined above, we have $y_{\alpha'}(z; \tau_x^\delta - t_z) \in \mathcal{K}_{\mathrm{dst}}$ and

$$e^{-t_z} \int_0^{\tau_x^\delta - t_z} e^{-s} ds \leqslant v_{\to \mathrm{d}}(x) - \int_0^{t_z} e^{-s} ds + \delta \ .$$

By dynamic programming equation, we have

$$v_{\to \mathrm{d}}(x) \leqslant \int_0^{t_z} e^{-s} ds + e^{-t_z} v_{\to \mathrm{d}}(z) \ ,$$

which implies

$$\int_0^{\tau_x^\delta - t_z} e^{-s} ds \leqslant v_{\to \mathrm{d}}(z) + e^{t_z}\delta \ .$$

We deduce the result from the definition of $(e^{t_z}\delta)$-optimal trajectories. □

Remark 4.3.13. One can deduce the same result for the $\delta-$optimal trajectory of the problem in reverse direction. In fact, for the minimum time problem, our definition of the $\delta-$optimal trajectory implies $\tau_x^\delta - \tau_x^* \leqslant e^{\tau_x^*}\delta$, where τ_x^δ and τ_x^* denote the $\delta-$optimal time and the true optimal time respectively.

Lemma 4.3.14. *For every $\eta > \delta > 0$, we have $\Gamma^\delta \subseteq \overline{\mathcal{O}}_\eta$.*

Proof. Let $y_{\alpha^{\delta'}}(x_{\mathrm{src}};\cdot) : [0, \tau] \to \overline{\Omega}$ denote a $\delta'-$optimal trajectory for the problem (4.2,4.8,4.9) with $x_{\mathrm{src}} \in \mathcal{X}_{\mathrm{src}}^{\delta-\delta'}$ and $\delta' \leqslant \delta$, then we have $x_{\mathrm{dst}} := y_{\alpha^\delta}(x_{\mathrm{src}}; \tau) \in \mathcal{K}_{\mathrm{dst}}$, and

$$\int_0^{\tau} e^{-s} ds \leqslant v_{\to \mathrm{d}}(x_{\mathrm{src}}) + \delta' \leqslant v^* + \delta \ .$$

It is sufficient to show that $y_{\alpha^{\delta'}}(x_{\mathrm{src}}; t_x) \in \overline{\mathcal{O}}_\eta$ for every $t_x \in [0, \tau]$.

For an arbitrary $t_x \in [0, \tau]$, let us denote $x := y_{\alpha^{\delta'}}(x_{\mathrm{src}}; t_x)$. Let $\alpha' : [0, \tau - t_x] \to S_1$ be a control such that $\alpha'(s) = \alpha^{\delta'}(s + t_x), \forall s \in [0, \tau - t_x]$. Then, we have that the associated trajectory starting at x with control α', $y_{\alpha'}(x;\cdot) : [0, \tau - t_x] \to \overline{\Omega}$ satisfies $y_{\alpha'}(x; s) = y_{\alpha^{\delta'}}(x_{\mathrm{src}}; s + t_x)$, for every $s \in [0, \tau - t_x]$. Then,

$$\int_0^{\tau} e^{-s} ds = 1 - \Big(1 - \int_0^{t_x} e^{-s} ds\Big)\Big(1 - \int_0^{\tau - t_x} e^{-s} ds\Big) \ .$$

By the definition of $v_{\to \mathrm{d}}$, and since $y_{\alpha'}(x; \tau - t_x) = x_{\mathrm{dst}} \in \mathcal{K}_{\mathrm{dst}}$, we have $v_{\to \mathrm{d}}(x) \leqslant \int_0^{\tau - t_x} e^{-s} ds$. Similarly, using the simple change of variable $s = t_x - s'$, we have $v_{\mathrm{s}\to}(x) \leqslant \int_0^{t_x} e^{-s} ds$. Then we deduce:

$$\begin{aligned} &1 - \Big(1 - \int_0^{t_x} e^{-s} ds\Big)\Big(1 - \int_0^{\tau - t_x} e^{-s} ds\Big) \\ &\quad \geqslant 1 - (1 - v_{\mathrm{s}\to}(x))(1 - v_{\to \mathrm{d}}(x)) = v_{\mathrm{s}\to}(x) + v_{\to \mathrm{d}}(x) - v_{\mathrm{s}\to}(x) v_{\to \mathrm{d}}(x) \ , \end{aligned}$$

and so

$$v_{\mathrm{s}\to}(x) + v_{\to \mathrm{d}}(x) - v_{\mathrm{s}\to}(x) v_{\to \mathrm{d}}(x) \leqslant v^* + \delta \ .$$

Using Lemma 4.3.5, and $\eta > \delta$, we obtain that $x = y_{\alpha^\delta}(x_{\mathrm{src}}; t_x) \in \overline{\mathcal{O}}_\eta$ for all $t_x \in [0, \tau]$. Since this is true for all $0 \leqslant \delta' \leqslant \delta$, we obtain $\Gamma^\delta \subseteq \overline{\mathcal{O}}_\eta$. □

Lemma 4.3.15. *For every $\delta^{'} > 0$, we have $\mathcal{O}_\eta \subset \Gamma^{\eta+\delta'}$.*

Proof. Take a $x \in \mathcal{O}_\eta$, it is sufficient to show that there exists at least one $(\eta + \delta^{'})-$optimal trajectory from $\mathcal{K}_{\text{src}}$ to $\mathcal{K}_{\text{dst}}$ that passes through x.

Suppose there exist an optimal trajectory from x to $\mathcal{K}_{\text{src}}$, and an optimal trajectory from x to $\mathcal{K}_{\text{dst}}$, then, concatenating the reverse trajectory of the optimal trajectory from x to $\mathcal{K}_{\text{src}}$, with the optimal trajectory from x to $\mathcal{K}_{\text{dst}}$, we obtain an $\eta-$optimal trajectory from one point of $\mathcal{K}_{\text{src}}$ to $\mathcal{K}_{\text{dst}}$ (by definition).

Otherwise, one can consider a $\frac{\delta'}{2}-$optimal trajectory from x to $\mathcal{K}_{\text{src}}$, $y_{\alpha_1}(x;\cdot) : [0, \tau_1] \to \Omega$, and a $\frac{\delta'}{2}-$optimal trajectory from x to $\mathcal{K}_{\text{dst}}$, $\tilde{y}_{\tilde{\alpha}_2}(x;\cdot) : [0, \tau_2] \to \Omega$. Then we have:

$$\begin{aligned}\int_0^{\tau_1+\tau_2} e^{-s}ds &= \int_0^{\tau_1} e^{-s}ds + e^{-\tau_1}\int_0^{\tau_2} e^{-s}ds \\ &\leqslant v_{\text{s}\to}(x) + \frac{\delta'}{2} + (1 - v_{\text{s}\to}(x))(v_{\to\text{d}}(x) + \frac{\delta'}{2}) \\ &\leqslant v_{\text{s}\to}(x) + v_{\to\text{d}}(x) - v_{\text{s}\to}(x)v_{\to\text{d}}(x) + (\frac{\delta'}{2} + \frac{\delta'}{2}) \leqslant v^* + (\eta + \delta') \ .\end{aligned}$$

Thus, concatenating as above the two optimal trajectories, we obtain a $(\eta + \delta')-$optimal trajectory from $y_{\alpha_1}(x;\tau_1) \in \mathcal{K}_{\text{src}}$ to $\mathcal{K}_{\text{dst}}$. □

The above two lemmas entail that the sets of $\delta-$geodesic points Γ^δ and $\mathcal{O}_\eta$ constitute equivalent families of neighborhoods of the optimal trajectory, and, in particular, $\overline{\mathcal{O}}_\eta$ contains at least all $\delta-$optimal trajectories for every $\delta < \eta$. Moreover, the sets $\mathcal{O}_\eta$, and $\mathcal{O}^\mu_\eta$ are also equivalent families of neighborhoods of the optimal trajectory, since $\mathcal{O}^\mu_{\eta-\mu} \subseteq \mathcal{O}_\eta \subseteq \mathcal{O}^\mu_{\eta+\mu}$ for arbitrary small μ. Based on these properties, we have the following result regarding the value functions.

Theorem 4.3.16. *For every $\delta < \eta$, for every $x \in \Gamma^\delta$, we have:*

$$v^\eta_{\text{s}\to}(x) = v_{\text{s}\to}(x), \ v^\eta_{\to\text{d}}(x) = v_{\to\text{d}}(x) \ .$$

Proof. We have $v_{\text{s}\to}(x) \leqslant v^\eta_{\text{s}\to}(x), v_{\to\text{d}}(x) \leqslant v^\eta_{\to\text{d}}(x)$ for all $x \in \mathcal{O}^\mu_\eta$, since $\mathcal{O}^\mu_\eta \subseteq \Omega$.

Now, let $\delta < \eta$. We have $\Gamma^\delta \subset \overline{\mathcal{O}^\mu_\eta}$ for μ small enough, using Lemma 4.3.14. Then, to show the reverse inequalities $v_{\text{s}\to}(x) \geqslant v^\eta_{\text{s}\to}(x), v_{\to\text{d}}(x) \geqslant v^\eta_{\to\text{d}}(x)$ for $x \in \Gamma^\delta$, it is sufficient to show that for all $\epsilon > 0$, there exist ϵ-optimal trajectories from $x \in \Gamma^\delta$ staying in $\overline{\mathcal{O}^\mu_\eta}$.

Let $y_{\alpha^{\delta'}}(x_{\text{src}};\cdot) : [0, \tau^{\delta'}] \to \overline{\Omega}$, be a δ'-optimal trajectory from x_{src} to $\mathcal{K}_{\text{dst}}$, with $\delta' \in [0, \delta]$, $x_{\text{src}} \in \mathcal{X}^{\delta-\delta'}_{\text{src}}$ and $y_{\alpha^{\delta'}}(x_{\text{src}};\tau^{\delta'}) = x_{\text{dst}} \in \mathcal{K}_{\text{dst}}$, and the associated $\delta'-$optimal control $\alpha^{\delta'}$. Let $x = y_{\alpha^{\delta'}}(x_{\text{src}};t_x)$, for $t_x \in [0, \tau^{\delta'}]$. Let $\epsilon > 0$ and consider a ϵ-optimal trajectory from x to $\mathcal{K}_{\text{dst}}$ with time length τ'. So we have $v_{\to\text{d}}(x) \leqslant (1 - e^{-\tau'}) \leqslant v_{\to\text{d}}(x) + \epsilon$. Replacing the trajectory $y_{\alpha^{\delta'}}(x;\cdot) : [0, \tau^{\delta'} - t_x] \to \overline{\Omega}$ by the ϵ-optimal trajectory from x to $\mathcal{K}_{\text{dst}}$, we have $(1 - e^{-\tau'}) \leqslant v_{\to\text{d}}(x) + \epsilon \leqslant 1 - e^{-(\tau^{\delta'} - t_x)} + \epsilon$. Then, we obtain a trajectory from x_{src} to $\mathcal{K}_{\text{dst}}$ with time $t_x + \tau'$ such that $v_{\to\text{d}}(x_{\text{src}}) \leqslant (1 - e^{-(t_x+\tau')}) \leqslant 1 - e^{-\tau^{\delta'}} + e^{-t_x}\epsilon \leqslant v_{\to\text{d}}(x_{\text{src}}) + \delta + \epsilon$. Then, this trajectory is in $\Gamma^{\delta'+\epsilon}_{x_{\text{src}}} \subseteq \Gamma^{\delta+\epsilon}$. For ϵ small enough we have $\delta + \epsilon < \eta$, so $\Gamma^{\delta+\epsilon} \subset \overline{\mathcal{O}^\mu_\eta}$, for μ small enough. We deduce that the ϵ-optimal trajectory from x to $\mathcal{K}_{\text{dst}}$ is included in $\overline{\mathcal{O}^\mu_\eta}$, which implies that $v^\eta_{\to\text{d}}(x) \leqslant v_{\to\text{d}}(x) + \epsilon$. Since it is true for all ϵ small enough, we deduce $v^\eta_{\to\text{d}}(x) \leqslant v_{\to\text{d}}(x)$ and so the equality.

By same arguments, we have $v^\eta_{\text{s}\to}(x) = v_{\text{s}\to}(x)$. □

Based on the above results, if we are only interested to find v^* and optimal trajectories between $\mathcal{K}_{\text{src}}$ and $\mathcal{K}_{\text{dst}}$, we can focus on solving the reduced problem in the subdomain $\mathcal{O}^\mu_\eta$, i.e., solving the system $SC(F, \mathcal{O}^\mu_\eta, (\partial\mathcal{O}^\mu_\eta) \cap (\partial\mathcal{K}_{\text{dst}}))$.

4.4 The Multi-level Fast-Marching Algorithm

We now introduce the multi-level fast-marching algorithm, to solve the initial minimum time problem proposed in (4.1), and, in particular, to find a good approximation of v^*.

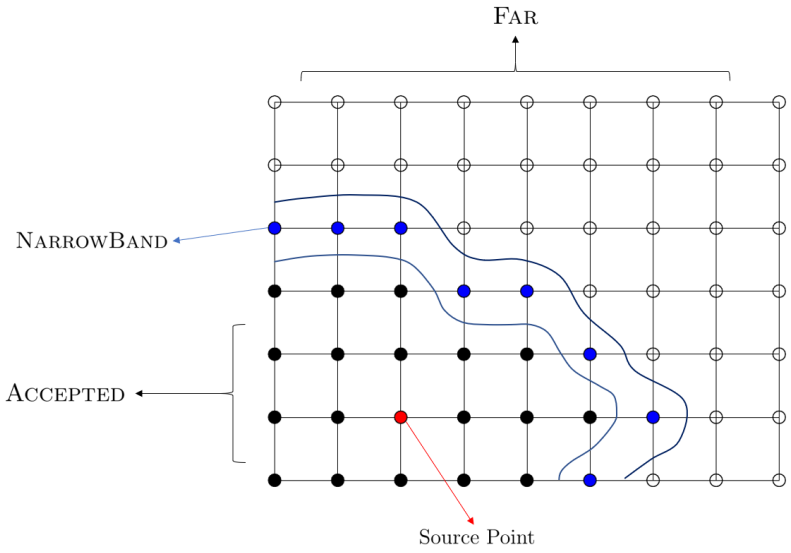
4.4.1 Classical Fast Marching Method

We first briefly recall the classical fast marching method introduced by Sethian [Set96] and Tsitsiklis [Tsi95], and which is one of the most effective numerical methods to solve the eikonal equation. It was first introduced to deal with the front propagation problem, then extended to general static HJ equations. Its initial idea takes advantage of the property that the evolution of the domain encircled by the front is monotone non-decreasing, thus one is allowed to only focus on the computation around the front at each iteration. Then, it is a single-pass method which is faster than standard iterative algorithms. Generally, it has computational complexity (number of arithmetic operations) in the order of $K_d M \log(M)$ in a d-dimensional grid with M points (see for instance [Set96; CF07]). The constant K_d is the maximal number of nodes of the discrete neighborhoods that are considered, so it depends on d and satisfies $K_d \in [2d, L^d]$ where L is the maximal diameter of discrete neighborhoods. For instance $K_d = 3^d$ for a local semilagrangian discretization, whereas $K_d = 2d$ for a first order finite difference discretization.

To be more precise, assume that we discretize the whole domain $\overline{\Omega}$ using a mesh grid X, and approximate the value function by the solution of a discrete equation of the form

$$V(x) = \mathcal{U}(V)(x), \ \forall \ x \in X \ . \tag{4.31}$$

Classical operators $\mathcal{U}$ in (4.31) are based on finite difference (for instance [KD01]) or semi-lagrangian discretizations (for instance [FF14]) of the system $SC(F, \Omega \setminus \mathcal{K}_{\text{dst}}, \partial\mathcal{K}_{\text{dst}})$. Note that (4.31) includes the boundary conditions. For our discounted problem (4.2,4.8,4.9), the operator $\mathcal{U}$ is monotone and contracting. The usual iterative methods compute the unique fixed point of $\mathcal{U}$. In fast marching method, $\mathcal{U}$ is also called the update operator. The fast marching algorithm visits the nodes of X in a special ordering and computes the approximate value function in just one iteration. The special ordering is such that the value function is monotone non-decreasing in the direction of propagation. This construction is done by dividing the nodes into three groups (see figure below): Far, which contains the nodes that have not been searched yet; Accepted, which contains the nodes at which the value function has been already computed and settled (by the monotone non-decreasing property of the front propagation, in the subsequent search, we do not need to update the value function of those nodes, see for instance [SV03]); and NarrowBand, which contains the nodes "around" the front (we only need to update the value function at these nodes).



At each step, the node in NarrowBand with the smallest value is added into the set of Accepted nodes, and then the NarrowBand and the value function over NarrowBand are updated, using the value of the last accepted node. The computation is done by appying an update operator $\mathcal{U}$. Sufficient conditions on the update operator $\mathcal{U}$ for the convergence of the fast marching algorithm are that $\mathcal{U}$ is not only monotone, but also causal (see for instance [Set96; CF07]).

A generic partial fast marching algorithm is given in Algorithm 4.1 (compare with [Set96; CF07]). We call it *partial* because the search stops when all the nodes of the ending set End are accepted. Then, the approximate value function may only be computed in End. The usual fast marching algorithm is obtained with End equal to the mesh grid X. Moreover, for an eikonal equation, the starting set Start plays the role of the target (intersected with X). If we only need to solve Problem (4.1), then we can apply Algorithm 4.1 with an update operator adapted to $SC(F, \Omega \setminus \mathcal{K}_{\text{dst}}, \partial\mathcal{K}_{\text{dst}})$ with F as in (4.10) and the sets Start and End equal to $\mathcal{K}_{\text{dst}} \cap X$ and $\mathcal{K}_{\text{src}} \cap X$ respectively. Similarly, we can apply Algorithm 4.1 with an update operator adapted to the reverse HJ equation $SC(F^*, \Omega \setminus \mathcal{K}_{\text{src}}, \partial\mathcal{K}_{\text{src}})$ (with F^* as in Section 4.2.3), which implies that the sets Start and End are equal to $\mathcal{K}_{\text{src}} \cap X$ and $\mathcal{K}_{\text{dst}} \cap X$ respectively.

4.4.2 Two Level Fast Marching Method

Our method combines coarse and fine grids discretizations, in order to obtain at a low cost, the value function on a subdomain of Ω around optimal trajectories. We start by describing our algorithm with only two levels of grid.

4.4.2.1 Computation in the Coarse Grid

We denote by X^H a coarse grid with constant mesh step H on $\overline{\Omega}$, and by x^H a node in this grid. We perform the two following steps, in the coarse grid:

(i) Do the partial fast marching search in the coarse grid X^H in both forward and backward directions, to solve Problem (4.1) as above.

(ii) Select and store the *active* nodes (see Definition 4.4.1) based on the two approximate value functions, as follows.

Algorithm 4.1 Partial Fast Marching Method.

Input: A mesh grid X; An update operator $\mathcal{U}$. Two sets of nodes: START and END.
Output: Approximate value function V and ACCEPTED set.
Initialization: Set $V(x) = +\infty, \forall x \in X$. Set all nodes as FAR.

Add START to ACCEPTED, add all neighborhood nodes to NARROWBAND.
Compute the initial value $V(x)$ of the nodes in NARROWBAND.
while (NARROWBAND is not empty and END is not accepted) **do**
  Select x^* having the minimum value $V(x^*)$ among the NARROWBAND nodes.
  Move x^* from NARROWBAND to ACCEPTED.
  for All nodes y not in ACCEPTED, such that $\mathcal{U}(V)(y)$ depends on x^* **do**
    $V(y) = \mathcal{U}(V)(y)$
    if y is not in NARROWBAND **then**
      Move y from FAR to NARROWBAND.
    end if
  end for
end while

The first step in coarse grid consists in applying Algorithm 4.1 to each direction, that is a partial fast marching search, with mesh grid X^H, and an update operator adapted to HJ equation $SC(F, \Omega \setminus \mathcal{K}_{\rm dst}, \partial\mathcal{K}_{\rm dst})$ or $SC(\tilde{F}, \Omega \setminus \mathcal{K}_{\rm src}, \partial\mathcal{K}_{\rm src})$ and the appropriate sets START and END. In particular, for a given parameter η, let us denote $\mathcal{K}^\eta_{\rm src} = \mathcal{K}_{\rm src} + B(0,\eta)$, $\mathcal{K}^\eta_{\rm dst} = \mathcal{K}_{\rm dst} + B(0,\eta)$. For the direction "from source", that is to solve the equation $SC(\tilde{F}, \Omega \setminus \mathcal{K}_{\rm src}, \partial\mathcal{K}_{\rm src})$, the set START is defined as $\mathcal{K}_{\rm src} \cap X^H$, and the set END is defined as $\mathcal{K}^\eta_{\rm dst} \cap X^H$. For the other direction, the set START is defined as $\mathcal{K}_{\rm dst} \cap X^H$ and the set END is defined as $\mathcal{K}^\eta_{\rm src} \cap X^H$.

This yields to the functions $V^{H,1}_{\rm s\to}$ and $V^{H,1}_{\to \rm d}$ that are numerical approximations of the value functions $v_{\rm s\to}$ and $v_{\to \rm d}$ on the sets of accepted nodes $A^H_{\rm s\to}$ and $A^H_{\to \rm d}$, respectively. Indeed, $V^{H,1}_{\rm s\to}$ (resp. $V^{H,1}_{\to \rm d}$) is a function defined on all X^H, which coincides on the set of accepted nodes $A^H_{\rm s\to}$ (resp. $A^H_{\to \rm d}$) with the unique fixed point of the update operator, that is the solution $V^H_{\rm s\to}$ (resp. $V^H_{\to \rm d}$) of the discretized equation; elsewhere it may be lower bounded by $V^H_{\rm s\to}$ (resp. $V^H_{\to \rm d}$), or $+\infty$. Under some regularity conditions on Problem (4.1), we have the following error bounds up to a certain order γ in H:

$$\begin{aligned} \varepsilon^H_{\rm s\to} &= \sup_{x \in X^H} \|V^H_{\rm s\to}(x) - v_{\rm s\to}(x)\| \leqslant C_{\rm s\to} H^\gamma, \\ \varepsilon^H_{\to \rm d} &= \sup_{x \in X^H} \|V^H_{\to \rm d}(x) - v_{\to \rm d}(x)\| \leqslant C_{\to \rm d} H^\gamma, \end{aligned} \tag{4.32}$$

which lead to the same bounds for $V^{H,1}_{\rm s\to}$ and $V^{H,1}_{\to \rm d}$ for the sup-norms restricted to $A^H_{\rm s\to}$ and to $A^H_{\to \rm d}$ respectively. When (4.32) holds, we get that $\mathcal{F}_{V^{H,1}}$ is an approximation of $\mathcal{F}_v$ on the set $A^H_{\rm s\to} \cap A^H_{\to \rm d}$ of accepted nodes for both directions, which should be an approximation of the set of geodesic points. We thus construct an approximation of $\mathcal{O}_\eta$ as follows.

Definition 4.4.1. For a given parameter $\eta_H > 0$, we say that a node $x^H \in X^H$ is *active* if $x^H \in A^H_{\rm s\to} \cap A^H_{\to \rm d}$ and

$$\mathcal{F}_{V^{H,1}}(x^H) \leqslant \min_{y^H \in X^H} \mathcal{F}_{V^{H,1}}(y^H) + \eta_H \ . \tag{4.33}$$

We denote by O^H_η the set of all active nodes for the parameter η_H.

The selection of active nodes in step (ii) above is based on the criterion (4.33).

4.4.2.2 Computation in the Fine Grid

Let us denote by X^h a grid discretizing $\overline{\Omega}$ with a constant mesh step $h < H$. For the computation in the fine grid, we again have two steps:

(i) Construct the fine grid, by keeping only the nodes of X^h that are in a neighborhood of the set of *active* nodes of the coarse grid.

(ii) Do a fast marching search in one direction in this fine grid.

More precisely, we select the fine grid nodes as follows:

$$G_\eta^h = \{x^h \in X^h \mid \exists x^H \in O_\eta^H : \|x^h - x^H\|_\infty \leqslant \max(H-h, h)\} \ . \tag{4.34}$$

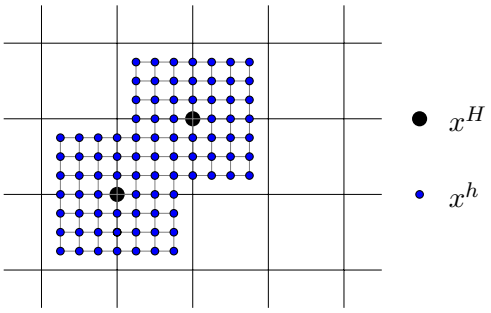


Figure 4.1: Constructing the fine neighborhood G_η^h given two active nodes x^H in the coarse grid.

Remark 4.4.2. As we shall see in Section 4.5, we may need to consider mesh steps H and h with h close to H, in particular such that $h > \frac{1}{2}H$. In this case, the bound in (4.34) is equal to h. In general, this bound is more efficient numerically, although any bound in $[H/2, H]$ would work theoretically.

To solve the original minimum time problem, the computation will only be done in the selected fine grid nodes, which means that a full fast marching algorithm Algorithm 4.1 is applied in the restricted fine grid G_η^h, with the update operator of one direction HJ equation (for instance with target set $\mathcal{K}_{\text{dst}}$). We will denote by $V_{\to \text{d}}^{h,2}$ the approximation of the value function $v_{\to \text{d}}$ generated by the above 2-level algorithm on G_η^h.

The complete algorithm is shown in Algorithm 4.2.

4.4.2.3 Convergence of Algorithm 4.2

In order to show the convergence of Algorithm 4.2, we first show that the computation in the fine grid is equivalent to the approximation of the value function of a new optimal control problem, with a restricted state space.

For this purpose, one shall first construct a continuous extension $O_\eta^{H,I}$ of O_η^H and G_η^h. Let us extend the approximate value function $V_{\text{s}\to}^{H,1}$ and $V_{\to \text{d}}^{H,1}$ from the nodes of X^H to the whole domain $\overline{\Omega}$ by a linear interpolation, and denote them by $V_{\text{s}\to}^{H.I}, V_{\to \text{d}}^{H,I}$ respectively. Then, $O_\eta^{H,I}$ is defined as follows

$$O_\eta^{H,I} = \{x \in (\Omega \setminus (\mathcal{K}_{\text{src}} \cup \mathcal{K}_{\text{dst}})) \mid \mathcal{F}_{V^{H,I}}(x) < \min_{x^H \in X^H} \mathcal{F}_{V^{H,1}}(x^H) + \eta_H\} \ . \tag{4.35}$$

Note however that another method may consists in constructing the region $O_\eta^{H,I}$ in the same way as G_η^h, but without the constraint $x \in X^h$. Nevertheless, $O_\eta^{H,I}$ can be thought of as a continuous

Algorithm 4.2 Two-Level Fast-Marching Method (2LFMM)

Input: Two grids X^h and X^H with mesh steps $h < H$ respectively. The parameter $\eta_H > 0$.
Input: Two update operators $\mathcal{U}_{\to \mathrm{d}}$ and $\mathcal{U}_{\mathrm{s}\to}$ adapted to both directions HJ equations.
Input: Target sets: $\mathcal{K}_{\mathrm{src}}, \mathcal{K}_{\mathrm{dst}}$.
Output: The fine grid FINE and approximate value function $V^{h,2}_{\to \mathrm{d}}$ on FINE.

Apply Algorithm 4.1 with Input grid X^H, update operator $\mathcal{U}_{\to \mathrm{d}}$, START $= \mathcal{K}_{\mathrm{dst}} \cap X^H$ and END $= \mathcal{K}^{\eta^H}_{\mathrm{src}} \cap X^H$, and output $V^{H,1}_{\to \mathrm{d}}$ and $A^H_{\to \mathrm{d}}$.
Apply Algorithm 4.1 with Input grid X^H, update operator $\mathcal{U}_{\mathrm{s}\to}$, START $= \mathcal{K}_{\mathrm{src}} \cap X^H$ and END $= \mathcal{K}^{\eta^H}_{\mathrm{dst}} \cap X^H$, and output $V^{H,1}_{\mathrm{s}\to}$ and $A^H_{\mathrm{s}\to}$.
for Every node x^H in $A^H_{\mathrm{s}\to} \cap A^H_{\to \mathrm{d}}$ **do**
    if $\mathcal{F}_{V^{H,1}(x)} \leqslant \min_{x^H \in X^H} \mathcal{F}_{V^{H,1}}(x^H) + \eta_H$ **then**
        Set x^H as ACTIVE.
    end if
end for
Set FINE to emptyset.
for Every node x^H in the ACTIVE set **do**
    for Every $x^h \in X^h$ satisfying $\|x^h - x^H\|_\infty \leqslant \max\{H - h, h\}$ **do**
        if x^h does not exist in set FINE **then**
            Add x^h in the set FINE.
        end if
    end for
end for
Apply Algorithm 4.1 with Input grid FINE, update operator $\mathcal{U}_{\to \mathrm{d}}$, START $= \mathcal{K}_{\mathrm{dst}} \cap$ FINE and END $= \mathcal{K}_{\mathrm{src}} \cap$ FINE, and output $V^{h,2}_{\to \mathrm{d}}$.

version of the set O_η^H of active nodes in coarse grid. We shall relate $O_\eta^{H,I}$ to the domain $\mathcal{O}_\eta$ defined in (4.26) – the notation "I" stands for "interpolation". Notice that one can also do a regularization of $O_\eta^{H,I}$ as in (4.28). Thus, in the following we shall do as if $\partial O_\eta^{H,I} \setminus (\mathcal{K}_{\rm src} \cup \mathcal{K}_{\rm dst})$ is of class $\mathcal{C}^1$. We then consider the continuous optimal control problem (4.2,4.8,4.9) with new state space $\overline{O_\eta^{H,I}}$, and the following new set of controls which is adapted to the new state constraint:

$$\mathcal{A}_{\eta_H,x} = \{\alpha \in \mathcal{A} \mid y_\alpha(x;s) \in \overline{O_\eta^{H,I}}, \text{ for all } s \geqslant 0 \} \ . \tag{4.36}$$

Denote by $v_{\to \rm d}^{\eta_H,I}$ the value function of this new state constrained problem. By Theorem 4.2.2, it is the unique solution of the new state constrained HJ equation $SC(F, O_\eta^{H,I}, (\partial O_\eta^{H,I}) \cap (\partial \mathcal{K}_{\rm dst}))$. In our two level fast marching algorithm, we indeed use the grid G_η^h to discretize $\overline{O_\eta^{H,I}}$, then $V_{\to \rm d}^{h,2}$ is an approximation of $v_{\to \rm d}^{\eta_H,I}$. Then, if $\overline{O_\eta^{H,I}}$ is big enough to contain the true optimal trajectories, by the results of Section 4.3, $v_{\to \rm d}^{\eta_H,I}$ coincides with $v_{\to \rm d}$ on the optimal trajectories. Then, $V_{\to \rm d}^{h,2}$ is an approximation of $v_{\to \rm d}$ on optimal trajectories. In the following result, we denote (as for $h = H$) by $V_{\to \rm d}^h$ the solution of the discretization of the HJ equation $SC(F, \Omega, \partial \mathcal{K}_{\rm dst})$ (associated to Problem (4.1)) on the grid X^h, or equivalently the unique fixed point of $\mathcal{U}_{\to \rm d}$, that is the output of Algorithm 4.1 with input grid X^h, update operator $\mathcal{U}_{\to \rm d}$, START $= \mathcal{K}_{\rm dst} \cap X^h$ and END $= X^h$.

Theorem 4.4.3 (Convergence of the Two-Level Fast-Marching Method)**.**

(i) *Assume* (4.32) *holds with* $\gamma \leqslant 1$*, and denote* $C_\gamma := C_{\rm s\to} + C_{\to \rm d}$ *and* $L_v := L_{v_{\rm s\to}} + L_{v_{\to \rm d}}$*, where* $L_{v_{\rm s\to}}$ *and* $L_{v_{\to \rm d}}$ *are the Lipschitz constants of* $v_{\rm s\to}$ *and* $v_{\to \rm d}$*, respectively. Then, there exists a constant* $C_\eta > 0$ *depending on* C_γ *and* L_v*, such that for every* $\delta' > \delta > 0$*, for every* $\eta_H \geqslant C_\eta H^\gamma + \delta'$*,* $\overline{O_\eta^{H,I}}$ *contains the set* $\overline{\mathcal{O}_{\delta'}} \supset \Gamma^\delta$*, that is the set of* $\delta-$*geodesic points for the continuous problem* (4.2,4.8,4.9)*. In particular, taking* $\eta_H \geqslant 2C_\eta H^\gamma$ *and* $\delta' = \frac{\eta_H}{2}$*, we have*

$$\Gamma^\delta \subset \overline{\mathcal{O}_{\frac{\eta_H}{2}}} \subset \overline{O_\eta^{H,I}} \ .$$

(ii) *Assume that the constants* $C_{\rm s\to}, C_{\to \rm d}$ *in* (4.32) *are uniform w.r.t. the state constraint (that is of* $\overline{\Omega}$*). Taking* η_H *and* $\delta' = \eta_H/2$ *as in* (i)*, we have, for every* $\delta < \eta_H/2$ *and* $x \in X^h \cap \Gamma^\delta$*,*

$$|V_{\to \rm d}^{h,2}(x) - v_{\to \rm d}(x)| \leqslant C_{\to \rm d} h^\gamma \ .$$

Thus, $V_{\to \rm d}^{h,2}(x)$ *converges towards* $v_{\to \rm d}(x)$ *as* $h \to 0$*.*

Proof. Let us prove Point (i). Using (4.32), for any $x^H \in X^H$, we have:

$$|\mathcal{F}_{V^H}(x^H) - \mathcal{F}_v(x^H)| \leqslant C_\gamma H^\gamma \ , \tag{4.37}$$

where $C_\gamma = C_{\rm s\to} + C_{\to \rm d}$. Moreover, using the Lipschitz continuity of $v_{\rm s\to}$ and $v_{\to \rm d}$, and denoting $L_v = L_{v_{\rm s\to}} + L_{v_{\to \rm d}}$, we obtain, for any $x \in \overline{\Omega}$ and $x^H \in X^H$ such that $\|x^H - x\| \leqslant H$,

$$|\mathcal{F}_{V^H}(x^H) - \mathcal{F}_v(x)| \leqslant C_\gamma H^\gamma + L_v H \ . \tag{4.38}$$

Applying (4.37) and $X^H \subset \overline{\Omega}$, we get

$$\min_{x^H \in X^H} \mathcal{F}_{V^H}(x^H) + C_\gamma H^\gamma \geqslant \min_{x \in \Omega} \mathcal{F}_v(x) \ . \tag{4.39}$$

Assume that x is in a d-dimensional polytope with vertices in X^H and that $V_{\to \rm d}^{H,I}$ and $V_{\rm s\to}^{H,I}$ are linear or affine on this polytope. One can show, using that both functions $V_{\to \rm d}^{H,I}$ and $V_{\rm s\to}^{H,I}$ take

their values in $[0,1]$, that the maximum (and minimum) of $\mathcal{F}_{V^{H,I}}$ on this polytope is attained on the vertices, so on points of X^H (although in some cases the function is concave). Using this property with (4.38), we obtain

$$\mathcal{F}_v(x) \geqslant \mathcal{F}_{V^{H,I}}(x) - (C_\gamma H^\gamma + L_v H) \ .$$

Let us assume now that $x \in \overline{\Omega \setminus (\mathcal{K}_{\rm src} \cup \mathcal{K}_{\rm dst})} \setminus \overline{O_\eta^{H,I}}$. Then, we deduce from the previous inequalities:

$$\begin{aligned}\mathcal{F}_v(x) &\geqslant \min_{x^H \in X^H} \mathcal{F}_{V^H}(x^H) + \eta_H - (C_\gamma H^\gamma + L_v H) \\ &\geqslant \min_{x\in\Omega} \mathcal{F}_v(x) + \eta_H - (2C_\gamma H^\gamma + L_v H) \ .\end{aligned}$$

Thus, if we take $\eta_H \geqslant 2C_\gamma H^\gamma + L_v H + \delta'$, we obtain $x \notin \mathcal{O}_{\delta'}$. This shows that $\mathcal{O}_{\delta'} \subset \overline{O_\eta^{H,I}}$ and thus $\overline{\mathcal{O}_{\delta'}} \subset \overline{O_\eta^{H,I}}$ Since $\gamma \leqslant 1$, we can take $\eta_H \geqslant C_\eta H^\gamma + \delta'$, with an appropriate constant C_η and also $\eta_H \geqslant 2C_\eta H^\gamma$ with $\delta' = \eta_H/2$, that is the result of Point (i).

As for Point (ii), taking η_H and $\delta' = \eta_H/2$ as in (i), we first notice that for every $x \in \overline{O_\eta^{H,I}} \cap X^h$, we have

$$|V^{h,2}_{\to \rm d}(x) - v^{\eta_H,I}_{\to \rm d}(x)| \leqslant C_{\to \rm d} h^\gamma \ . \tag{4.40}$$

Indeed this is (4.32), when the state is constrained to stay in $\overline{O_\eta^{H,I}}$ and the grid has mesh step h. Since $\overline{\mathcal{O}_{\frac{\eta_H}{2}}} \subset \overline{O_\eta^{H,I}} \subset \overline{\Omega}$, we have for every $x \in \mathcal{O}_{\frac{\eta_H}{2}}$,

$$v_{\to \rm d}(x) \leqslant v^{\eta_H,I}_{\to \rm d}(x) \leqslant v^{\frac{\eta_H}{2}}_{\to \rm d}(x) \ . \tag{4.41}$$

By Theorem 4.3.16, we have that for every $\delta < \eta_H/2$ and $x \in \Gamma^\delta \subset \overline{\mathcal{O}_{\frac{\eta_H}{2}}}$, $v_{\to \rm d}(x) = v^{\frac{\eta_H}{2}}_{\to \rm d}(x)$. Thus we get an equality in (4.41). Replacing $v^{\eta_H,I}_{\to \rm d}$ by $v_{\to \rm d}$ in (4.40), we obtain the result of Point (ii). □

4.4.3 Multi-level Fast Marching Method

The computation in two level coarse fine grid can be extended to the multi-level case. In a nutshell, we construct finer and finer grids, considering the fine grid of the previous step as the coarse grid of the current step, and defining the next fine grid by selecting the actives nodes of this coarse grid.

4.4.3.1 Computation in Multi-level Grids

Consider a N-level family of grids with successive mesh steps: $H_1 \geqslant H_2 \geqslant \ldots \geqslant H_{N-1} \geqslant H_N = h$, denoted X^{H_i}, for $i = 1, \ldots, N$. Given a family of real positive parameters $\{\eta_1, \eta_2, \ldots, \eta_{N-1}\}$, the computation works as follows:

Level-1: In first level, the computations are the same as in coarse grid of the two level method (Section 4.4.2.1), with mesh step H equal to H_1 and active nodes selected using the parameter η equal to η_1. At the end of level-1, we get a set of active nodes: $O^{H_1}_{\eta_1}$.

Level-l with $1 < l < N$: In level-l, we already know the set of active nodes in level-$(l-1)$, denoted $O^{H_{l-1}}_{\eta_{l-1}}$. We first construct the "fine grid" set of level-l as in (4.34), that is:

$$G^{H_l}_{\eta_{l-1}} = \{x^{H_l} \in X^{H_l} \mid \exists\, x^{H_{l-1}} \in O^{H_{l-1}}_{\eta_{l-1}} : \|x^{H_{l-1}} - x^{H_l}\|_\infty \leqslant \max(H_{l-1} - H_l, H_l)\} \ . \tag{4.42}$$

Then, we perform the fast marching in both directions in the grid $G^{H_l}_{\eta_{l-1}}$. This leads to the approximations $V^{H_l,l}_{\mathrm{s}\rightarrow}$, $V^{H_l,l}_{\rightarrow\mathrm{d}}$ and $\mathcal{F}_{V^{H_l,l}}$ of $v_{\mathrm{s}\rightarrow}$, $v_{\rightarrow\mathrm{d}}$ and $\mathcal{F}_v$ on grid $G^{H_l}_{\eta_{l-1}}$. We then select the active nodes in level-l, by using the parameter η_l, as follows:

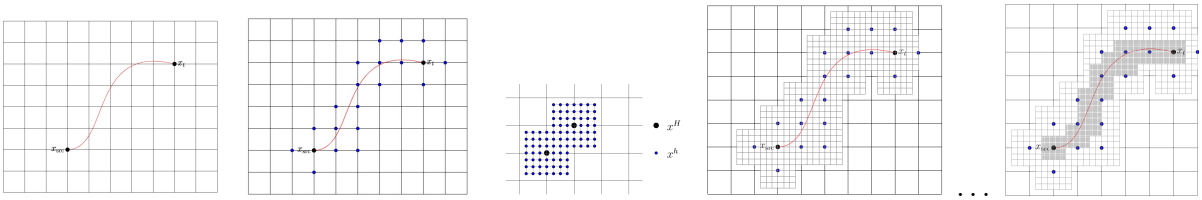
$$O^{H_l}_{\eta_l} = \{x^{H_l} \in G^{H_l}_{\eta_{l-1}} \mid \mathcal{F}_{V^{H_l,l}}(x^{H_l}) \leqslant \min_{x^{H_l} \in G^{H_l}_{\eta_{l-1}}} \mathcal{F}_{V^{H_l,l}}(x^{H_l}) + \eta_l \} \ . \tag{4.43}$$

Level-N: In the last level, we only construct the final fine grid:

$$G^{h}_{\eta_{N-1}} = \{x^h \in X^h \mid \exists\, x^{H_{N-1}} \in O^{H_{N-1}}_{\eta_{N-1}} : \|x^h - x^{H_{N-1}}\|_\infty \leqslant \max(H_{N-1} - h, h)\} \ . \tag{4.44}$$

Then, we only do the fast marching search in one direction in the grid $G^{h}_{\eta_{N-1}}$ and obtain the approximation $V^{h,N}_{\rightarrow\mathrm{d}}$ of $v_{\rightarrow\mathrm{d}}$ on grid $G^{h}_{\eta_{N-1}}$.

This is detailed in Algorithm 4.3. Some possible grids generated by our algorithm are shown in the following Figure 4.2.



(a) Level-0 (b) Active Nodes (c) Fine grid (d) Level-1 ... (e) Level-2

Figure 4.2: Sketch of MLFMM.

Algorithm 4.3 Multi-Level Fast-Marching Method (MLFMM)

Input: The mesh steps, grids, and selection parameters: H_l, X^{H_l}, η_l, for $l \in \{1, 2, \ldots, N\}$.
Input: Update operators $\mathcal{U}_{\rightarrow\mathrm{d}}$ and $\mathcal{U}_{\mathrm{s}\rightarrow}$ adapted to both directions HJ equations and levels.
Input: Target sets: $\mathcal{K}_{\mathrm{src}}, \mathcal{K}_{\mathrm{dst}}$.
Output: The final fine grid FINE and approximate value function $V^{h,N}_{\rightarrow\mathrm{d}}$ on FINE.

1: Set COARSE-GRID to X^{H_1}.
2: **for** $l = 1$ to $N - 1$ **do**
3: Do the partial fast marching search in COARSE-GRID in both directions.
4: Select the ACTIVE nodes from the ACCEPTED nodes using η_l.
5: Select the FINE nodes based on the ACTIVE nodes, and mesh step H_{l+1}.
6: Let FINE in current level be the new COARSE-GRID.
7: **end for**
8: Do the partial fast marching search in only one direction in FINE.

In Algorithm 4.3, Line-3 of Algorithm 4.3 corresponds to lines-1 and 2 in Algorithm 4.2, line-4 of Algorithm 4.3 corresponds to lines-3 to 7 in Algorithm 4.2, line-5 of Algorithm 4.3 corresponds to line-8 to line-15 in Algorithm 4.2.

4.4.3.2 Convergence of Algorithm 4.3

In each level-l with $l < N$, we have the approximate value functions $V^{H_l,l}_{\mathrm{s}\rightarrow}$ and $V^{H_l,l}_{\rightarrow\mathrm{d}}$ of $v_{\mathrm{s}\rightarrow}$ and $v_{\rightarrow\mathrm{d}}$ on the grid of level l. Then, we can apply the same constructions as in Section 4.4.2.3 for

the coarse grid X^H. This leads to the following continuous version of the set of active nodes in level-l:

$$O_{\eta_l}^{H_l,I} = \{x \in (\Omega \setminus (\mathcal{K}_{\rm src} \cup \mathcal{K}_{\rm dst})) \mid \mathcal{F}_{V^{H_l,I}}(x) < \min_{x^H \in G_{\eta_{l-1}}^{H_l}} \mathcal{F}_{V^{H_l,l}}(x^H) + \eta_l\} \ .$$

We also have sets $\mathcal{A}_{\eta_l,x}$ of controls adapted to the optimal control problems of the form (4.2,4.8,4.9) with state space equal to the set $O_{\eta_l}^{H_l,I}$, and the corresponding value function $v_{\to \rm d}^{\eta_l,I}$, which, by Theorem 4.2.2, is the unique solution of the new state constrained HJ equation $SC(F, O_{\eta_l}^{H_l,I}, (\partial O_{\eta_l}^{H_l,I}) \cap (\partial \mathcal{K}_{\rm dst}))$. We can also consider the HJ equations in the direction "from source", $SC(\tilde{F}, O_{\eta_l}^{H_l,I}, (\partial O_{\eta_l}^{H_l,I}) \cap (\partial \mathcal{K}_{\rm src}))$, and the corresponding value function $v_{\rm s\to}^{\eta_l,I}$. We obtain the following convergence result for Algorithm 4.3.

Theorem 4.4.4 (Convergence of the Multi-level Fast-Marching Method)**.** *Assume* (4.32) *holds for all $H > 0$, with $\gamma \leqslant 1$ and some constants $C_{\rm s\to}, C_{\to \rm d}$ that are uniform w.r.t. the state constraint (that is $\overline{\Omega}$).*

(i) *There exists constants $C_\eta > 0$ and $\kappa > 0$ such that, if for every $l \in \{1, \dots, N-1\}$, $\eta_l = C_\eta (H_l)^\gamma$ and $H_l/H_{l+1} \geqslant \kappa$, then for all $\delta < \frac{\eta_l}{2}$, the set $\overline{O_{\eta_l}^{H_l,I}}$ contains $\overline{\mathcal{O}_{\frac{\eta_l}{2}}} \supset \Gamma^\delta$, that is the set of δ-geodesic points for the continuous minimum time problem*(4.2,4.8,4.9).

(ii) *Taking η_l as proposed in* (i)*, then for every $l \in \{1, 2, \dots, N-1\}$, $\delta < \frac{\eta_l}{2}$, and $x \in X^{H_{l+1}} \cap \Gamma^\delta$, we have*

$$|V_{\to \rm d}^{H_{l+1},l+1}(x) - v_{\to \rm d}(x)| \leqslant C_{\to \rm d}(H_{l+1})^\gamma, \ |V_{\rm s\to}^{H_{l+1},l+1}(x) - v_{\rm s\to}(x)| \leqslant C_{\rm s\to}(H_{l+1})^\gamma \ .$$

Thus, $V_{\to \rm d}^{h,N}(x)$ converges towards $v_{\to \rm d}(x)$ as $h \to 0$.

Proof. When $l = 1$, Points (i) and (ii) of Theorem 4.4.4 follow from the corresponding points in Theorem 4.4.3: $\overline{\mathcal{O}_{\frac{\eta_1}{2}}} \subset \overline{O_{\eta_1}^{H_1,I}}$, and for every $x \in H^{H_2} \cap \Gamma^\delta$ with $\delta < \frac{\eta_1}{2}$:

$$|V_{\to \rm d}^{H_2,2}(x) - v_{\to \rm d}(x)| \leqslant C_{\to \rm d} H_2^\gamma, \quad |V_{\rm s\to}^{H_2,2}(x) - v_{\rm s\to}(x)| \leqslant C_{\rm s\to} H_2^\gamma \ . \tag{4.45}$$

Assume now that Points (i) and (ii) of Theorem 4.4.4 hold for $l = k-1$, with an arbitrary $k \geqslant 2$. We first prove Point (i) for $l = k$, that is $\mathcal{O}_{\frac{\eta_k}{2}} \subset \overline{O_{\eta_k}^{H_k,I}}$. Let us denote as before by $V_{\rm s\to}^{H_k}$, $V_{\to \rm d}^{H_k}$ the solutions in the grid X^{H_k} of the discretized equation in two directions, respectively. From (4.32), we obtain (4.37), which in the case $H = H_k$ writes

$$\sup_{x \in X^{H_k}} \|\mathcal{F}_{V^{H_k}}(x) - \mathcal{F}_v(x)\| \leqslant C_\gamma (H_k)^\gamma. \ . \tag{4.46}$$

Moreover, we notice that, for every $x \in G_{\eta_{k-1}}^{H_k} \subset X^{H_k}$, we have

$$V_{\rm s\to}^{H_k,k}(x) \geqslant V_{\rm s\to}^{H_k}(x), \quad V_{\to \rm d}^{H_k,k}(x) \geqslant V_{\to \rm d}^{H_k}(x) \ . \tag{4.47}$$

Hence,

$$\mathcal{F}_{V^{H_k,k}}(x) \geqslant \mathcal{F}_{V^{H_k}}(x) \geqslant \mathcal{F}_v(x) - C_\gamma (H_k)^\gamma \ , \tag{4.48}$$

which is similar to (4.39) for $H = H_k$. Consider a point $x \in \overline{\Omega \setminus (\mathcal{K}_{\rm src} \cup \mathcal{K}_{\rm dst})} \setminus \overline{O_{\eta_k}^{H_k,I}}$, then, by definition of $\overline{O_{\eta_k}^{H_k,I}}$, we have

$$\begin{aligned} \mathcal{F}_{V^{H_k,I}}(x) &\geqslant \min_{x^{H_k} \in G_{\eta_{k-1}}^{H_k}} \mathcal{F}_{V^{H_k,k}}(x^{H_k}) + \eta_k \\ &\geqslant \min_{x^{H_k} \in X^{H_k}} \mathcal{F}_v(x^{H_k}) + \eta_k - C_\gamma (H_k)^\gamma \\ &\geqslant \min_{x \in \Omega} \mathcal{F}_v(x) + \eta_k - C_\gamma (H_k)^\gamma \ . \end{aligned} \tag{4.49}$$

Assume also that $x \in \mathcal{O}_{\frac{\eta_k}{2}}$. Let us denote

$$B^{H_k}(x) = \{x^{H_k} \in X^{H_k} \mid \|x - x^{H_k}\|_\infty \leqslant H_k \} , \tag{4.50}$$

and assume further that $B^{H_k}(x) \subset \Gamma^\delta$ for some $\delta < \frac{\eta_{k-1}}{2}$. Using Lemma 4.3.15, and the Lipschitz continuity of $\mathcal{F}_v$, we get this property as soon as $\eta_k + 2L_v H_k < \eta_{k-1}$. Since we assumed that Point (i) holds for $l = k-1$, and $B^{H_k}(x) \subset X^{H_k} \cap \Gamma^\delta$, we have

$$\|\mathcal{F}_{V^{H_k,k}}(x^{H_k}) - \mathcal{F}_v(x^{H_k})\| \leqslant C_\gamma (H_k)^\gamma \ ,$$

for all $x^{H_k} \in B^{H_k}(x)$. Then, by the same arguments as in the proof of Theorem 4.4.3, using (4.49), we obtain

$$\begin{aligned}
\mathcal{F}_{V^{H_k,I}}(x) &\leqslant \max_{x^{H_k} \in B^{H_k}(x)} \mathcal{F}_{V^{H_k,k}}(x^{H_k}) \\
&\leqslant \mathcal{F}_v(x) + C_\gamma (H_k)^\gamma + L_v H_k \\
&\leqslant \min_{y \in \Omega} \mathcal{F}_v(y) + \frac{\eta_k}{2} + C_\gamma (H_k)^\gamma + L_v H_k \\
&\leqslant \mathcal{F}_{V^{H_k,I}}(x) - \frac{\eta_k}{2} + (2C_\gamma (H_k)^\gamma + L_v H_k)
\end{aligned} \tag{4.51}$$

If η_k satisfies also $\eta_k \geqslant 4C_\gamma (H_k)^\gamma + 2L_v H_k$, we get a contradiction. Since $\gamma \leqslant 1$, this condition is satisfied as soon as $\eta_k \geqslant C_\eta (H_k)^\gamma$ with some appropriate constants C_η. If η_k also satisfies $\eta_k \leqslant C'_\eta (H_k)^\gamma$ for some constant $C'_\eta > C_\eta$, we get that the above condition $\eta_k + 2L_v H_k < \eta_{k-1}$ holds as soon as $H_k \leqslant 1$ and $\frac{C'_\eta + 2L_v}{C_\eta} \leqslant (\frac{H_{k-1}}{H_k})^\gamma$. Under these conditions, we deduce that for all $\delta < \frac{\eta_k}{2}$, we have $\Gamma^\delta \subset \mathcal{O}_{\frac{\eta_k}{2}} \subset \overline{O^{H_k,I}_{\eta_k}}$.

By the same argument as in the proof of Point (ii) of Theorem 4.4.3, we deduce Point (ii) of Theorem 4.4.4 for $l = k$ from Point (ii) of Theorem 4.4.4 for the same $l = k$. The result of Theorem 4.4.4 follows from the induction on l. □

4.4.4 The Data Structure

In this section, we describe a dedicated data structure, which will allow us to store the successive neighborhoods, and implement the algorithm, in an efficient way.

Recall that for the classical fast-marching method, the data are normally stored using two types of structures [BCZ10]: a full d-dimensional table (or tensor), which contains all the values of the current approximate value function on the whole discretization grid (the values are updated at each step); a dynamical linked list, which contains the information on the narrow band nodes with the current approximate value function.

To implement efficiently our algorithms, we need to store the successive (constrained) grids $G^{H_l}_{\eta_{l-1}}$, for every $l \in \{2, \ldots, N\}$, in an efficient way. A d-dimensional full table would be too expensive for the storage, since the complexity would be in the order of $(\frac{1}{h})^d$, which is impossible to implement for a small mesh step h in high dimension. Moreover, the aim of our algorithm is to reduce the number of nodes in order to reduce the computational complexity, but this gain would be lost if we used a full table storage. We propose here a different storage of the grids $G^{H_l}_{\eta_{l-1}}$, in order to get a storage complexity in the order of the cardinality of these grids.

To implement our algorithm, we need to perform three type of operations, when constructing the fine grid $G^{H_l}_{\eta_{l-1}}$ from the active nodes, that is the elements of $O^{H_{l-1}}_{\eta_{l-1}}$:

1. Check if one node x^{H_l} already exists in the grid;

2. Add one node x^{H_l} into the existing grid;

3. Check the neighborhood information of one node $x^{H_{l-1}}$ or x^{H_l}.

We need to fulfil two goals. On the one hand, we want to keep the computational complexity for the operations "search" (the above steps 1 and 3) and "insert" (the above step 2) to be as low as possible, ideally in $O(1)$ time, since we need to do these operations at least once for every node of a grid. On the other hand, we want the memory used to store the grid at depth l not to exceed the size of the neighborhood of the "small" set $O^{H_{l-1}}_{\eta_{l-1}}$, interpolated in the fine grid of step H_l, avoiding to store nodes outside this neighborhood.

We used a "hash-table", to efficiently implement our algorithm. Suppose we are in the d-dimensional case. For the level-l grid we have approximately M_l nodes to store. For each node $x^l \in G^{H_l}_{\eta_{l-1}}$, we store three types of data in the hash table:

1. A d-dimensional vector of "int" type, which corresponds to its position in $\mathbb{R}^d$ or equivalently its corresponding indices in the full d-dimensional table.

2. A "double' type data, which corresponds to its value function.

3. Two "boolean" type data, for the fast matching search and selection of the active nodes.

We then use the position of a node, $x^l \in \mathbb{R}^d$, as the "key" for the hash table to compute the corresponding slot by a hash function $h(x^l)$. If several nodes have the same slot, i.e., a "collision" occurs, we need to attach to this slot the above data for each of these nodes. So we attach to a slot a vector, the entries of which are the above data for each node associated to this slot, see Figure 4.3. The simple hash function we used is as follows:

$$h(x^l) = (\sum_{k=1}^{d} x^l_k M'_i) \mod 2\overline{M_l} \ . \tag{4.52}$$

where $M'_i, i \in \{1, 2, \ldots, d\}$ is a random integer in $[1, 2\overline{M_l}]$, and $\overline{M_l}$ is the predicted number of nodes in level-l (which will be detailed later). In fact, the hash function intends to reduce the "collisions", and numerical experiments show that this function could handle most of the cases well. In some particular case, it can be optimized using other hashing methods, for example the multiplication method or the universal hashing method [Cor+09].

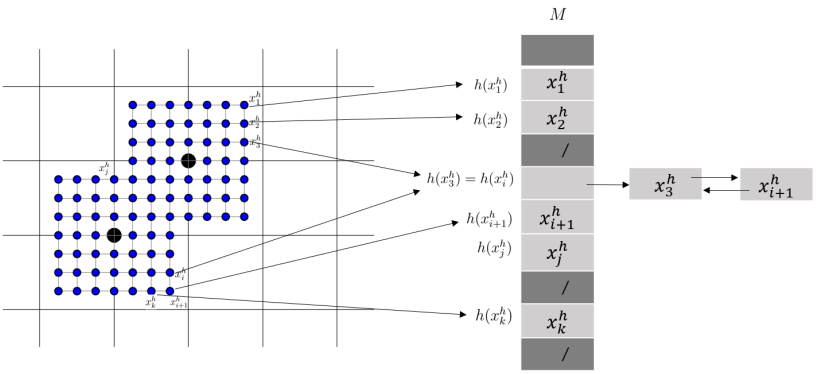


Figure 4.3: The hash table to store fine grid nodes.

4.5 Computational Complexity

In previous section, we already proved that our algorithm computes an approximation of the value of Problem (4.1), and of the set of its geodesic points, with an error depending on the mesh step of the finest grid. In this section, we analyze the space complexity and the computational complexity of our algorithm, and characterize the optimal parameters to tune the algorithm. In this section, we always use the following assumption:

Assumption (A5) The domain $\overline{\Omega}$ is convex and there exist constants $\underline{f}, \overline{f}$ such that:

$$0 < \underline{f} \leqslant f(x,\alpha) \leqslant \overline{f} < \infty, \text{ for all } x \in \Omega \text{ and } \alpha \in \mathcal{A} \ .$$

We will also assume that (4.32) holds for some $\gamma > 0$ and for all mesh sizes $H > 0$, so that the conclusions of Theorem 4.4.3 and Theorem 4.4.4 hold. Consider first the two-level case, and suppose we want to get a numerical approximation of the value of Problem (4.1) with an error bounded by some given $\varepsilon > 0$, and a minimal total complexity. Three parameters should be fixed before the computation:

(i) The mesh step of the fine grid h;

(ii) The mesh step of the coarse grid H;

(iii) The parameter η_H to select the active nodes in coarse grid.

The parameter h need to be small enough so that $C_{\to \mathrm{d}} h^\gamma \leqslant \varepsilon$ (using (4.32)). The parameter η_H is used to ensure that the subdomain $O_\eta^{H,I}$ does contain the true optimal trajectories, so it should be large enough as a function of H, see Theorem 4.4.3, but we also want it to be as small as possible to reduce the complexity. Using the optimal values of h and η_H, the total complexity becomes a function of H, when ε (or h) is fixed. Then, one need to choose the optimal value of the parameter H regarding this total complexity.

In order to be able to estimate the total complexity, we shall also use the following assumption on the neighborhood of optimal trajectories:

Assumption (A6) The set of geodesic points Γ^* consists of a finite number of optimal paths between $\mathcal{K}_{\mathrm{src}}$ and $\mathcal{K}_{\mathrm{dst}}$. Moreover, for every $x \in \mathcal{O}_\eta$, there exists $x^* \in \Gamma^*$ such that :

$$\|x - x^*\| \leqslant C_\beta \eta^\beta \ ,$$

where $0 < \beta \leqslant 1$, and C_β is a positive constant.

Remark 4.5.1. The above constant β depends on the geometry of the level sets of the value function. In typical situations in which the value function is smooth with a nondegenerate Hessian in the neighborhood of an optimal trajectory, one has $\beta = 1/2$. However, in general situations, it can take all the possible values in $(0, 1]$. Indeed, consider for instance the minimum time between two discs in a 2-dimensional space with $f(x,\alpha) = 1/\|\alpha\|_p$, where $\|\cdot\|_p$ is the L^p norm, with $p \in [1, \infty]$. This is equivalent to a problem with a dynamics independent of state and a direction chosen from the unit L^p sphere instead of the L^2 sphere. Then, one can show that for $p \neq \infty$, the straight line is the unique optimal path, and that the value function satisfies Assumption (A6) with $\beta = 1/p \in (0, 1]$. However, if $p = \infty$, the number of optimal paths is infinite, so Assumption (A6) is not satisfied.

Let us denote by D the maximum Euclidean distance between the points in $\mathcal{K}_{\mathrm{src}}$ and $\mathcal{K}_{\mathrm{dst}}$, i.e. $D = \sup\{\|x-y\| \mid x \in \mathcal{K}_{\mathrm{src}}, y \in \mathcal{K}_{\mathrm{dst}}\}$, and by D_Ω the diameter of $\overline{\Omega}$, i.e. $D = \sup\{\|x-y\| \mid x, y \in \overline{\Omega}\}$. For any positive functions $f, g : \mathbb{R}^p \to \mathbb{R}_{>0}$ of p real parameters, the notation $g(x) = \widetilde{O}(f(x))$ will mean $g(x) = O(f(x)(\log(f(x)))^q)$ for some integer q, that is $|g(x)| \leqslant C f(x) |\log(f(x))|^q$ for some constant $C > 0$. We have the folowing estimate of the space complexity.

Proposition 4.5.2. *Assume that $H \leqslant D$, and that η_H satisfies the condition of Theorem 4.4.3 and $\eta_H \leqslant D$. There exists a constant $C > 0$ depending on D_Ω, D, $\overline{f}$ and $\underline{f}$, β, γ, C_β, C_γ and L_v (see Theorem 4.4.3), such that the space complexity $\mathcal{C}_{spa}(H,h)$ of the two level fast marching algorithm with coarse grid mesh step H, fine grid mesh step h and parameter η_H is as follows:*

$$\mathcal{C}_{spa}(H,h) = \widetilde{O}\Big(C^d\Big(\frac{1}{H^d} + \frac{(\eta_H)^{\beta(d-1)}}{h^d}\Big)\Big) \ . \tag{4.53}$$

Proof. Up to a multiplicative factor, in the order of d (so which enters in the $\widetilde{O}$ part) the space complexity is equal to the total number of nodes of the coarse and fine grids.

We first show that up to a multiplicative factor, the first term in (4.53), $(\frac{C}{H})^d$, is the number of accepted nodes in the coarse-grid. To do so, we exploit the monotone property of the fast-marching update operator. Recall that in the coarse grid, we incorporate dynamically new nodes by partial fast-marching, starting from $\mathcal{K}_{\rm src} \cap X^H$ (resp. $\mathcal{K}_{\rm dst} \cap X^H$) until $\mathcal{K}_{\rm dst} \cap X^H$ (resp. $\mathcal{K}_{\rm src} \cap X^H$) is accepted.

Let us first consider the algorithm starting from $\mathcal{K}_{\rm src}$. In this step, let us denote $x^f_{\rm dst}$ the last accepted node in $\mathcal{K}_{\rm dst}$, then we have for all the nodes $x^H \in A^H_{\rm s\to}$ that have been accepted, $V^H_{\rm s\to}(x^H) \leqslant V^H_{\rm s\to}(x^f_{\rm dst})$. Then, using (4.32)), we obtain $v_{\rm s\to}(x^H) \leqslant v_{\rm s\to}(x^f_{\rm dst}) + 2C_{\rm s\to}H^\gamma \leqslant v_{\rm s\to}(x^f_{\rm dst}) + 2C_{\rm s\to}D^\gamma$, which gives the following inclusion, when D and $T_{\rm s\to}(x^f_{\rm dst})$ are small enough:

$$\begin{aligned} A^H_{\rm s\to} &\subseteq \{x \in \overline{\Omega} \mid v_{\rm s\to}(x) \leqslant v_{\rm s\to}(x^f_{\rm dst}) + 2C_{\rm s\to}H^\gamma\} \\ &\subseteq \{x \in \overline{\Omega} \mid T_{\rm s\to}(x) \leqslant T_{\rm s\to}(x^f_{\rm dst}) - \log(1 - 2C_{\rm s\to}D^\gamma e^{T_{\rm s\to}(x^f_{\rm dst})})\} \ . \end{aligned}$$

Let $x \in \overline{\Omega}$, recall that $T_{\rm s\to}(x)$ is the minimum time traveling from x to $\mathcal{K}_{\rm src}$, then we have for some $x^i_{\rm src} \in \mathcal{K}_{\rm src}$,

$$\|x - x^i_{\rm src}\| \leqslant \int_0^{T_{\rm s\to}(x)} \|\dot{x}(t)\| dt \leqslant \overline{f} T_{\rm s\to}(x) \ . \tag{4.54}$$

Moreover, we have for some $x^j_{\rm src} \in \mathcal{K}_{\rm src}$,

$$T_{\rm s\to}(x^f_{\rm dst}) \leqslant \frac{\|x^f_{\rm dst} - x^j_{\rm src}\|}{\underline{f}} \leqslant \frac{D}{\underline{f}} \ , \tag{4.55}$$

since we can take a control α proportional to $x^f_{\rm dst} - x^j_{\rm src}$, so that the trajectory given by (4.12) follows the straight line from $x^j_{\rm src}$ to $x^f_{\rm dst}$ (with variable speed). Combine (4.54) and (4.55), we have for the set of accepted nodes:

$$A^H_{\rm s\to} \subseteq \{x \mid \|x - x^i_{\rm src}\| \leqslant (\overline{f}D/\underline{f}) - \overline{f}\log(1 - 2C_{\rm s\to}D^\gamma e^{\overline{f}D/\underline{f}}), \text{ for some } x^i_{\rm src} \in \mathcal{K}_{\rm src}\} \ .$$

Thus, all the nodes we visit are included in a d-dimensional ball with radius R, where R is a constant depending on D, $\overline{f}$, $\underline{f}$, $C_{\rm s\to}$ and γ, when D is small enough. Otherwise, since $\overline{\Omega}$ has a diameter equal to D_Ω, one can take $R = D_\Omega$. Then, the total number of nodes that are accepted in the coarse grid is bounded by $(\frac{C}{H})^d$, in which C/R is a positive constant in the order of $(v_d)^{1/d}$, where v_d is the volume of unit ball in $\mathbb{R}^d$, and satisfying $C/R \leqslant 2$, so we can take $C/R = 2$. The same result can be obtained for the search starting from $\mathcal{K}_{\rm dst}$.

We now show that, still up to a multiplicative factor, the second term in (4.53), $D\frac{(\eta_H)^{\beta(d-1)}}{h^d}$, is the number of nodes of the fine-grid. Consider a node $x^h \in G^h_\eta$. By definition, there exists $x^H \in O^H_\eta$ such that $\|x^h - x^H\| \leqslant H$. Denote $C_\gamma = C_{\rm s\to} + C_{\to \rm d}$, and $L_v = L_{v_{\rm s\to}} + L_{v_{\to \rm d}}$ (the sum

of the Lipschitz constants of $v_{\to \mathrm{d}}$ and $v_{\mathrm{s}\to}$). Then, using similar arguments as in the proof of Theorem 4.4.3, we obtain:

$$\begin{aligned}\mathcal{F}_v(x^h) &\leqslant \mathcal{F}_v(x^H) + L_v\|x^h - x^H\| \\ &\leqslant \mathcal{F}_{V^H}(x^H) + C_\gamma H^\gamma + L_v H \\ &\leqslant \min_{y^H \in X^H} \mathcal{F}_v(y^H) + \eta_H + C_\gamma H^\gamma + L_v H \\ &\leqslant \min_{y\in\Omega} \mathcal{F}_v(y) + \eta_H + C_\gamma H^\gamma + 2L_v H \ .\end{aligned}$$

This entails that $x^h \in \mathcal{O}_{\eta_H+\varepsilon_H}$, with $\varepsilon_H = C_\gamma H^\gamma + 2L_v H$. Since $\gamma \leqslant 1$, so ε_H is of order H^γ, and $\eta_H \geqslant C_\eta H^\gamma$ by assumption, then using Assumption (A6), we deduce that for some positive constant C', depending on β, γ, C_β, C_γ, and L_v, there exists $x^* \in \Gamma^*$ such that

$$\|x^h - x^*\| \leqslant C'(\eta_H)^\beta \ . \tag{4.56}$$

We assume now that Γ^* consists of a single optimal path from $x_{\mathrm{src}} \in \mathcal{K}_{\mathrm{src}}$ to $x_{\mathrm{dst}} \in \mathcal{K}_{\mathrm{dst}}$. Indeed, the proof for the case of a finite number of paths is similar and leads to a constant factor in the complexity, which is equal to the number of paths. Up to a change of variables, we get a parametrization of Γ^* as the image of a one-to-one map $\phi : t \in [0, D_\Gamma] \mapsto \phi(t) \in \Gamma^*$, with unit speed $\|\phi'(t)\| = 1$, where D_Γ is a positive constant. Let us denote in the following by $d_{\Gamma^*}(x, y)$ the distance between two points $x, y \in \Gamma^*$ along this path, that is $d_{\Gamma^*}(x, y) = |t - s|$ if $x = \phi(t)$ and $y = \phi(s)$, with $t, s \in [0, D_\Gamma]$. Since the speed of ϕ is one, we have $\|x - y\| \leqslant d_{\Gamma^*}(x, y)$, and so $D \leqslant D_\Gamma$. Moreover, by the same arguments as above, we have $D_\Gamma \leqslant \overline{f} D/\underline{f}$.

Let us divide Γ^*, taking equidistant points $x_0, x_1, x_2, ..., x_N, x_{N+1} \in \Gamma^*$ between $x_0 = x_{\mathrm{src}}$ and $x_{N+1} = x_{\mathrm{dst}}$, with $N = \lfloor D_\Gamma/(C'(\eta_H)^\beta)\rfloor$, so that $d_{\Gamma^*}(x_k, x_{k+1}) = D_\Gamma/(N+1) \leqslant C'(\eta_H)^\beta \ \forall k \in \{0, 1, 2, ..., N\}$. Set $\Gamma^*_{\mathrm{dis}} := \{x_{\mathrm{src}}, x_1, x_2, ..., x_N, x_{\mathrm{dst}}\}$. Then, by (4.56), we have for every $x^h \in G^h_\eta$, there exists a point $x \in \Gamma^*_{\mathrm{dis}}$ such that :

$$\|x^h - x\| \leqslant \frac{3}{2} C'(\eta_H)^\beta \ .$$

Let us denote $B^d(x, r)$ the d-dimensional open ball with center x and radius r (for the Euclidian norm). Taking, $\Delta := \frac{3}{2}C'(\eta_H)^\beta + \frac{h}{2}$, we deduce:

$$\bigcup_{x\in G^h_\eta} B^d(x, \frac{h}{2}) \subset \bigcup_{x\in\Gamma^*_{\mathrm{dis}}} B^d(x, \Delta) \ .$$

Moreover, since the mesh step of G^h_η is h, all balls centered in $x \in G^h_\eta$ with radius $\frac{h}{2}$ are disjoint, which entails:

$$\mathrm{Vol}\Big(\bigcup_{x\in O^h_\eta} B^d(x, \frac{h}{2})\Big) = |G^h_\eta|(\frac{h}{2})^d \upsilon_d \ ,$$

where υ_d denotes the volume of the unit ball in dimension d, and $|G^h_\eta|$ denotes the cardinality of G^h_η, which is also the number of nodes in the fine grid. Thus, we have:

$$|G^h_\eta| = \frac{\mathrm{Vol}\left(\bigcup_{x\in O^h_\eta} B^d(x, \frac{h}{2})\right)}{(\frac{h}{2})^d \upsilon_d} \leqslant \frac{\mathrm{Vol}\left(\bigcup_{x\in\Gamma^*_{\mathrm{dis}}} B^d(x, \Delta)\right)}{(\frac{h}{2})^d \upsilon_d} \leqslant |\Gamma^*_{\mathrm{dis}}| 2^d \frac{\Delta^d}{h^d} \ .$$

Since $\eta_H \geqslant C_\eta H^\gamma$, $h \leqslant H$ and $\beta, \gamma \leqslant 1$, we get that $\Delta \leqslant C''(\eta_H)^\beta$ for some constant C'' depending on C_η, C', β and γ. Then,

$$|G^h_\eta| \leqslant D' \frac{(2C''(\eta_H)^\beta)^{d-1}}{h^d} \ ,$$

where D' depends on D_Γ, D and C'. This leads to the bound of the proposition. □

Remark 4.5.3. For the fast marching method with semi-lagrangian scheme, the computational complexity $\mathcal{C}_{comp}$ satisfies $\mathcal{C}_{comp} = \widetilde{O}(3^d \mathcal{C}_{spa})$. Then, the same holds for the two-level or multi-level fast marching methods. In particular for the two-level fast marching method $\mathcal{C}_{comp}$ has same estimation as $\mathcal{C}_{spa}$ in Proposition 4.5.2.

The same analysis as in two level case also works for the $N-$level case, for which we have the following result:

Proposition 4.5.4. *Assume that $H_1 \leqslant D$, and that, for $l = 1, \dots, N-1$, η_l satisfies the condition of Theorem 4.4.4, and $\eta_l \leqslant D$. Then, the total computational complexity $\mathcal{C}_{comp}(H_1, H_2, \dots, H_N)$ of the multi-level fast marching algorithm with N-levels, with grid mesh steps $H_1 \geqslant H_2 \geqslant, \dots, \geqslant H_{N-1} \geqslant H_N = h$ is*

$$\mathcal{C}_{comp}(\{H_l\}_{1\leqslant l\leqslant N}) = \widetilde{O}\Big(C^d\Big(\frac{1}{(H_1)^d} + \frac{(\eta_1)^{\beta(d-1)}}{(H_2)^d} + \frac{(\eta_2)^{\beta(d-1)}}{(H_3)^d} + \cdots + \frac{(\eta_{N-1})^{\beta(d-1)}}{h^d}\Big)\Big) \ , \quad (4.57)$$

with C as in Proposition 4.5.2. Moreover, the space complexity has a similar formula. □

Minimizing the formula in Proposition 4.5.2 and Proposition 4.5.4, we obtain the following result of the computational complexity.

Theorem 4.5.5. *Assume $d \geqslant 2$, and let $\nu := \gamma\beta(1-\frac{1}{d}) < 1$. Let $\varepsilon > 0$, and choose $h = (C_\gamma^{-1}\varepsilon)^{\frac{1}{\gamma}}$. Then, there exist some constant C_m depending on the same parameters as in Proposition 4.5.2, such that, in order to obtain an error bound on the value of Problem (4.1) less or equal to ε small enough, one can use one of the following methods:*

(i) *The two-level fast marching method with $\eta_H = C_\eta H^\gamma$, and $H = h^{\frac{1}{\nu+1}}$. In this case, the total computational complexity is $\mathcal{C}_{comp}(H, h) = \widetilde{O}((C_m)^d(\frac{1}{\varepsilon})^{\frac{d}{\gamma(\nu+1)}})$.*

(ii) *The $N-$level fast marching method with $\eta_l = C_\eta H_l^\gamma$ and $H_l = h^{\frac{1-\nu^l}{1-\nu^N}}$, for $l = 1, \dots, N-1$. In this case, the total computational complexity is $\widetilde{O}(N(C_m)^d(\frac{1}{\varepsilon})^{\frac{1-\nu}{1-\nu^N}\frac{d}{\gamma}})$.*

(iii) *The $N-$level fast marching method with $N = \lfloor \frac{1}{\gamma}\log(\frac{1}{\varepsilon})\rfloor$, and $\eta_l = C_\eta H_l^\gamma$ and $H_l = h^{\frac{l}{N}}$, for $l = 1, \dots, N-1$. Then, the total computational complexity reduces to $\widetilde{O}((C_m)^d(\frac{1}{\varepsilon})^{(1-\nu)\frac{d}{\gamma}}) = \widetilde{O}((C_m)^d(\frac{1}{\varepsilon})^{\frac{1+(d-1)(1-\gamma\beta)}{\gamma}})$. When $\gamma = \beta = 1$, it reduces to $\widetilde{O}((C_m)^d\frac{1}{\varepsilon})$.*

Proof. For (i), using (4.32) together with Theorem 4.4.3, which applies since $\eta_H = C_\eta H^\gamma$, we get that the error on the value obtained by the two-level fast marching method is less or equal to $C_\gamma h^\gamma = \varepsilon$. Note that in order to apply Theorem 4.4.3, η_H needs to satisfy $\eta_H \geqslant C_\eta H^\gamma$. Then, to get a minimal computational complexity, one need to take $\eta_H = C_\eta H^\gamma$ as in the theorem. We obtain the following total computational complexity:

$$\mathcal{C}_{comp}(H, h) = \widetilde{O}((C')^d(H^{-d} + h^{-d}H^{\gamma\beta(d-1)})) \ , \quad (4.58)$$

for some new constant $C' = C\max(1, C_\eta)^\beta$. When h is fixed, this is a function of H which gets its minimum value for

$$H = C_1 h^{\frac{d}{(\gamma\beta+1)d-\gamma\beta}} \quad \text{with} \quad C_1 = \Big(\frac{d}{\gamma\beta(d-1)}\Big)^{\frac{1}{(\gamma\beta+1)d-\gamma\beta}} \ , \quad (4.59)$$

since it is decreasing before this point and then increasing. Then, the minimal computational complexity bound is obtained by substituting the value of H of (4.59) in (4.58). The formula of

H in (4.59) is as in (i), up to the multiplicative factor $C_1 > 0$. One can show that $1 \leqslant C_1 \leqslant C_2$ with C_2 depending only on $\gamma\beta$. Hence, substituing this value of H instead of the one of (4.59) in (4.58), gives the same bound up to the multiplicative factor C_1^d. So in both cases, using $h = (C_\gamma^{-1}\varepsilon)^{\frac{1}{\gamma}}$, we obtain a computational complexity bound as in (i), for some constant C_m depending on the same parameters as in Proposition 4.5.2.

For (ii), using this time (4.32) together with Theorem 4.4.4, and that $\eta_l = C_l H_l^\gamma$, we get that the error on the value obtained by the multi-level fast marching method is less or equal to $C_\gamma h^\gamma = \varepsilon$. As in (i), we apply the formula of Proposition 4.5.4 with $\eta_l = C_l H_l^\gamma$, which gives

$$\mathcal{C}_{comp}(\{H_l\}_{1\leqslant l\leqslant N}) = \widetilde{O}\Big((C')^d((H_1)^{-d} + (H_1)^{\gamma\beta(d-1)}(H_2)^{-d} + \cdots + (H_{N-1})^{\gamma\beta(d-1)}h^{-d}\Big)\ , \quad (4.60)$$

for the same constant C' as above. This is again a function of $H_1, H_2, \ldots, H_{N-1}$ when h is fixed. We deduce the optimal mesh steps $\{H_1, H_2, \ldots, H_{N-1}\}$ by taking the minimum of $\mathcal{C}_{comp}(\{H_l\}_{1\leqslant l\leqslant N})$ with respect to $H_1, H_2, \ldots, H_{N-1}$, and then simplifying the formula by eliminating the constants. We can indeed proceed by induction on N, and use iterative formula similar to (4.59). We then obtain the formula for H_l as in (ii). Substituting these values of the H_l into (4.60), for a general $d \geqslant 2$, we obtain the following bound on the total computational complexity

$$\mathcal{C}_{comp}(\{H_l\}_{1\leqslant l\leqslant N}) = \widetilde{O}(N(C')^d(\frac{1}{h})^{\frac{1-\nu}{1-\nu^N}d})\ . \quad (4.61)$$

Now using $h = (C_\gamma^{-1}\varepsilon)^{\frac{1}{\gamma}}$, we get the formula of (ii).

For (iii), let us first do as if $\nu = 1$. In that case, passing to the limit when ν goes to 1 in previous formula, we obtain the new formula for the H_l given in (iii). We also obtain a new formula for the total complexity in (4.61) of the form $\widetilde{O}(N(C')^d(\frac{1}{h})^{\frac{d}{N}})$. The minimum of this formula with respect to N is obtained for $N = d\log(\frac{1}{h})$ which with $h = (C_\gamma^{-1}\varepsilon)^{\frac{1}{\gamma}}$, leads to a formula of N in the order of the one of (iii). Let us now substitute the values of H_l into the complexity formula (4.60), we obtain the total computational complexity (for h small enough)

$$\mathcal{C}_{comp}(\{H_l\}_{1\leqslant l\leqslant N}) = \widetilde{O}\Big((C')^d(\frac{1}{h})^{\frac{d}{N}}\sum_{l=0}^{N-1}(\frac{1}{h})^{\frac{l}{N}(1-\nu)d}\Big) = \widetilde{O}\Big(N(C')^d(\frac{1}{h})^{\frac{d}{N}}(\frac{1}{h})^{(1-\nu)d}\Big)\ . \quad (4.62)$$

Now taking $N = \lfloor \frac{1}{\gamma}\log(\frac{1}{\varepsilon})\rfloor$ and using again $h = (C_\gamma^{-1}\varepsilon)^{\frac{1}{\gamma}}$, we get the formula of (iii). □

Remark 4.5.6. In Point (iii) of Theorem 4.5.5, we can replace N by any formula of the form $N = \lfloor \kappa\log(\frac{1}{h})\rfloor$, with a constant $\kappa > 0$. In that case, the number of levels and the parameters H_l (the intermediate mesh steps) only depend on the final mesh step h and the dimension. However, the conditions on the parameters η_l depend on the parameters γ and C_η of the problem to be solved and thus are difficult to estimate in practice. Moreover, it may happen that the upper bound C_η is too large, making the theoretical complexity too large in practice even when $\gamma\beta = 1$.

Remark 4.5.7. In Theorem 4.5.5, the theoretical complexity bound highly depends on the value of $\gamma\beta$. For the first constant γ, which is the convergence rate of the fast marching method, the usual finite differences or semilagrangian schemes satisfy $\gamma = \frac{1}{2}$. However, it may be equal to 1, which typically occurs under a semiconcavity assumption (see for instance [CF96; FF14]). Higher order schemes (in time step) may also be used, under some additional regularity on the value function, see for instance [FF98; Bok+15] and lead to $\gamma \geqslant 1$.

Recall that the second constant β, defined in Assumption (A6), determines the growth of the neighborhood $\mathcal{O}_\eta$ of the optimal trajectories, as a function of η. This exponent depends on the geometry of the level sets of the value function. We provide some examples for which $\beta = 1$ in Section 4.B. In particular, one can find examples such that $\gamma = \beta = 1$.

4.6 Numerical Experiments

In this section, we present numerical tests, showing the improvement of our algorithm, compared with the original fast-marching method of Sethian et al. [Set96; SV01], using the same update operator. Both algorithms were implemented in C++, and executed on a single core of a Quad Core IntelCore I7 at 2.3Gh with 16Gb of RAM.

Note that, as said in Remark 4.5.6, the constant C_η in the formula of the parameters η_l of the multi-level fast marching method is difficult to estimate. Then, for a given problem, first several tests of the algorithm are done for large values of the mesh steps (or on the first levels of the multi-level method) with some initial guess of the constant C_η, assuming that $\gamma = 1$. This is not too expensive, so we do not count it in the total CPU time of the multi-level fast marching method.

4.6.1 The tested problems

In the numerical tests, we shall consider the following particular problems in several dimensions d.

Problem 1 (Euclidean distance in a box). We start with the easiest case: a constant speed $f(x,\alpha) \equiv 1$ in $\Omega = (0,1)^d$. The sets $\mathcal{K}_{\text{src}}$ and $\mathcal{K}_{\text{dst}}$ are the Euclidean balls with radius 0.1, centered at $(0.2,\dots,0.2)$ and $(0.8,\dots,0.8)$ respectively.

Problem 2 (Discontinuous speed field). The domain Ω and the sets $\mathcal{K}_{\text{src}}$, $\mathcal{K}_{\text{dst}}$ are the same as in Problem 1. The speed function is discontinuous, with the form:

$$f(x,\alpha) = \begin{cases} 0.3, & x \in (0.4,0.6)^d, \\ 1, & \text{elsewhere.} \end{cases}$$

Thus, the speed is reduced in a box centered in the domain. In this case, optimal trajectories must "avoid" the box, so there are $2\binom{d}{2} = d(d-1)$ optimal trajectories from $\mathcal{K}_{\text{src}}$ and $\mathcal{K}_{\text{dst}}$, which are obtained by symmetry arguments.

Problem 3 (The Poincaré Model). Consider the minimum time problem in the open unit Euclidian ball $\Omega = B^d(0,1)$. The sets $\mathcal{K}_{\text{src}}$ and $\mathcal{K}_{\text{dst}}$ are the Euclidean balls with radius $\frac{0.1}{\sqrt{d}}$, centered at $(\frac{-0.8}{\sqrt{d}},\dots,\frac{-0.8}{\sqrt{d}})$ and $(\frac{0.8}{\sqrt{d}},\dots,\frac{0.8}{\sqrt{d}})$ respectively. This speed function f is given as follows:

$$f(x,\alpha) = 1 - \|x\|^2 \ .$$

This particular choice of vector field corresponds to the Poincaré model of the hyperbolic geometry, that is, the optimal trajectories of our minimum time problem between two points are geodesics or "straight lines" in the hyperbolic sense.

Problem 4. Consider $\Omega = B^d(0,1)$, the open unit Euclidian ball, and let $\mathcal{K}_{\text{src}}$ and $\mathcal{K}_{\text{dst}}$ be the closed balls with radius $r/2$ and centers $(1-r,0,\dots,0)$ and $(-1+r,0,\dots,0)$, respectively, with $0 < r < \frac{1}{2}$. For every $x = (x_1,x_2,\dots,x_d) \in \Omega$ and $\alpha \in \mathcal{A}$, the speed is given as follows:

$$f(x,\alpha) = 1 + x_1^2 - \sum_{i=2}^{d} x_i^2 \ . \tag{4.63}$$

We prove in Example 2 of Section 4.B that this problem satisfies $\beta = 1$.

Problem 5. We address the problem with same domain and speed as in Problem 4, but with the same source and destination sets as in Problem 3.

4.6.2 Comparison between ordinary and multi-level fast-marching methods

We next provide detailed results comparing the performances of the classical and multi-level fast marching methods in the special case of Problem 1. Detailed results concerning Problems 2–5 are given in Section 4.C, showing a similar gain in performance. In the higher dimension cases, the classical fast marching method cannot be executed in a reasonable time. So we fix a time budget of 1 hour, that is we show the results of these algorithms when they finish in less than 1 hour only. Figure 4.4 shows the CPU time and the memory allocation for Problem 1 in dimensions range from 2 to 6, with grid meshes equal to $\frac{1}{50}$ and $\frac{1}{100}$. In all the dimensions in which both methods can be executed in less than one hour, we observe that the multi-level fast marching method with finest mesh step h yields the same relative error as the classical fast marching method with mesh step h, but with considerably reduced CPU times and memory requirements. Moreover, when the dimension is greater than 4, the classical fast marching method could not be executed in a time budget of 1 hour. In all, the multi-level method appears to be much less sensitive to the *curse-of-dimensionality*.

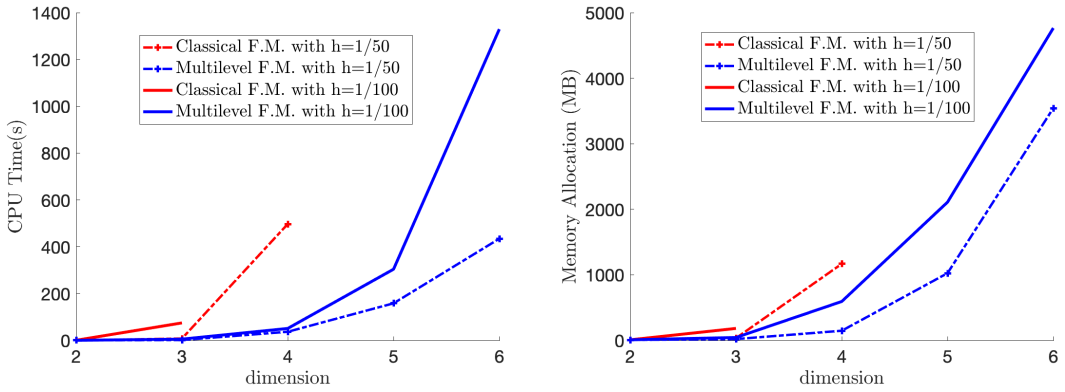


Figure 4.4: Problem 1. CPU time (left) and memory allocation (right) as a function of the dimension, for a fixed finest mesh step h.

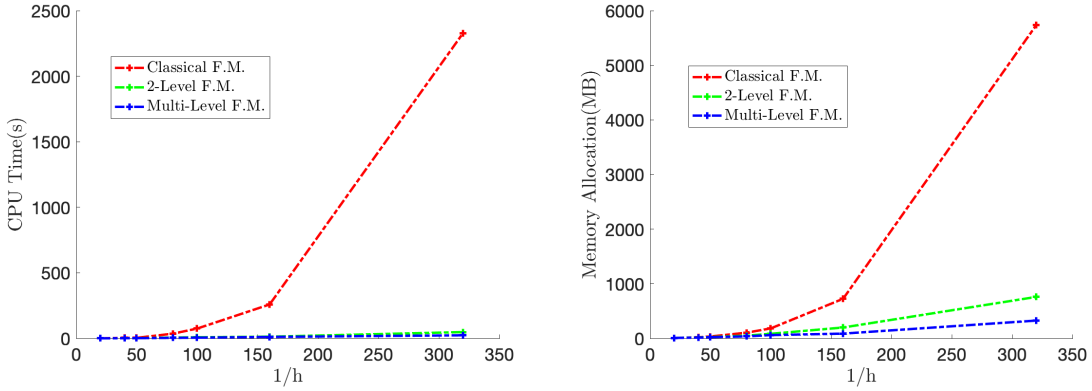
To better compare the multi-level method with the classical method, we test the algorithms with several (finest) mesh steps (which are proportional to the predictable error up to some exponent $1/\gamma$), when the dimension is fixed to be 3, see Figure 4.5. We consider both the 2-level algorithm and the multi-level algorithm. In the multi-level case, the number of levels is adjusted to be (almost) optimal for different mesh steps and dimensions.

4.6.3 Effective complexity of the multi-level fast-marching method

We next analyse the experimental complexity of the multi-level fast marching method in the light of the theoretical estimates of Theorem 4.5.5. For this purpose, we tested the multi-level fast marching method on Problems 1–5, with an almost optimal number of levels, and several dimensions and final mesh steps. For all these cases, we compute the (logarithm of) CPU time and shall plot it as a function of the dimension, or of the final mesh step.

If the number of levels is choosen optimal as in Theorem 4.5.5, we can expect a complexity in the order of $\widetilde{O}(C^d(\frac{1}{\varepsilon})^{\frac{1+(d-1)(1-\gamma\beta)}{\gamma}})$, depending on the model characteristics, to be compared with $(\frac{1}{\varepsilon})^{\frac{d}{\gamma}}$ for the usual fast marching method. This means that the logarithm of CPU time should be of the form

$$\log(\text{CPU time}) \simeq s_0 + s_2 \log(\frac{1}{h}) + (s_1 + s_3 \log(\frac{1}{h}))d = s_0 + s_1 d + (s_2 + s_3 d)\log(\frac{1}{h}) \ , \quad (4.64)$$

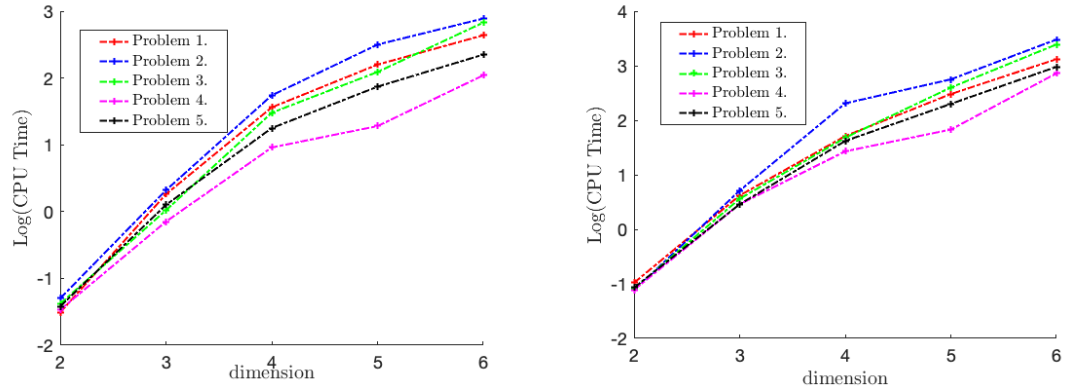


(a) CPU time as a function of $\frac{1}{h}$ (b) Memory allocation as a function of $\frac{1}{h}$

Figure 4.5: Problem 1. CPU time and memory allocation for several values of the finest mesh step h, in dimension 3.

where $s_1 = \log(C)$, $s_2 = \gamma\beta \in (0, 1]$ and $s_3 = 1 - \gamma\beta$. To check if such an estimation of complexity holds, we shall execute the multi-level fastmarching method on all the problems for several values of h and d, and compute the logarithm of the CPU time as a function of the dimension and then as a function of $\log(1/h)$. However, choosing an optimal number of levels may be difficult to implement due to the small differences between the mesh steps. So, the results will not always fit with the above theoretical prediction.

We first present tests done for dimension 2 to 6 and finest mesh step $\frac{1}{50}$ and $\frac{1}{100}$, for which we compute the (logarithm of) CPU time. We show in Figure 4.6, the graph of the logarithm of CPU time as a function of the dimension (when finest mesh step is fixed), for which the form (4.64) suggests a slope of the form $s_1 + s_3 \log(\frac{1}{h})$. We also give the precise values of these functions in Table 4.1, where we compute the slope by linear regression. If the slope satisfies this formula, then one can get an estimation of s_3 and then of $\gamma\beta$ using several values of h. Here, we have only two values of h, which gives a rough estimation of s_3, also given in Table 4.1. This gives an estimation of $\gamma\beta \in [0.71, 0.88]$, so close to 1. Another possibility is that $\gamma\beta = 1$ and that the number of levels is not optimal, which implies that the logarithm of the CPU time is not affine in the dimension.



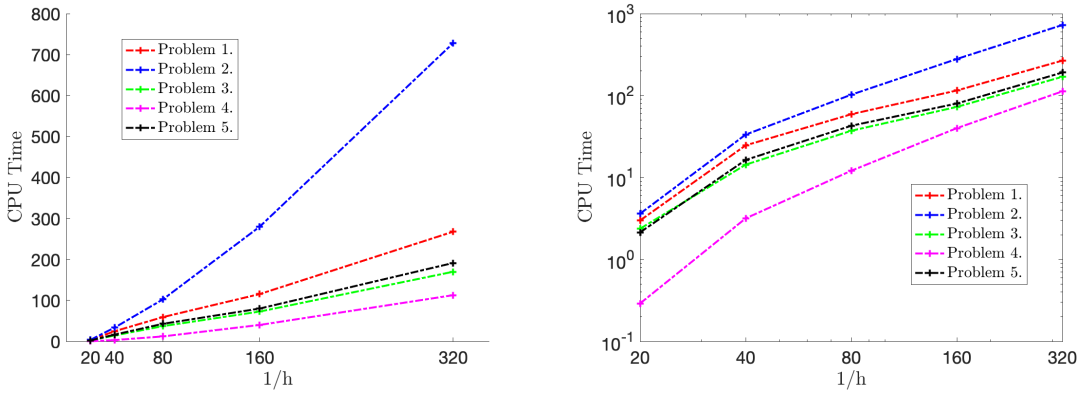
(a) log(CPU time) w.r.t. dimension when $h = \frac{1}{50}$. (b) log(CPU time) w.r.t. dimension when $h = \frac{1}{100}$.

Figure 4.6: Growth of CPU time w.r.t. dimensions.

Table 4.1: Values and slope of log(CPU time) w.r.t. dimension.

	log(CPU Time) w.r.t. dimension, h= $\frac{1}{50}$						log(CPU Time) w.r.t. dimension, h= $\frac{1}{100}$						
Dimension:	2	3	4	5	6	slope	2	3	4	5	6	slope	s_3
Problem 1	-1.44	0.26	1.56	2.20	2.64	0.778	-0.98	0.62	1.71	2.48	3.12	0.827	0.16
Problem 2	-1.30	0.32	1.74	2.50	2.89	0.847	-1.12	0.71	2.31	2.75	3.48	0.905	0.20
Problem 3	-1.38	0.02	1.48	2.09	2.83	0.904	-1.09	0.56	1.68	2.60	3.39	0.941	0.12
Problem 4	-1.47	-0.05	0.96	1.28	2.14	0.719	-1.11	0.46	1.43	1.83	2.86	0.760	0.13
Problem 5	-1.42	0.10	1.25	1.87	2.35	0.737	-1.07	0.46	1.62	2.30	2.98	0.824	0.29

We next present tests done for dimension 4 and several values of finest mesh step going from 1/20 to 1/320, for which we compute the (logarithm of) CPU time. We present in Figure 4.7, the graphs of the CPU time as a function of $\frac{1}{h}$, in both linear scales and log-log scales. We also compute the precise values of the logarithm of the CPU time as a function of the logarithm of $\frac{1}{h}$ in Table 4.2, where we compute the slope by linear regression. The form (4.64) of the CPU time suggests a slope $s_2 + s_3 d$ with $s_3 = 1 - s_2$ and $s_2 = \gamma\beta$. So the value of this slope for $d = 1$ is 1 and the value of the slope for $d = 4$ allows one to compute a second rough estimation of s_3, that we also give in Table 4.2. The results match with the ones of Table 4.1 and so again the estimation of $\gamma\beta \in [0.76, 0.86]$ is close to 1, or suggest that $\gamma\beta = 1$ but that the number of levels is not optimal.



(a) CPU time w.r.t. mesh step, linear scale.

(b) CPU time w.r.t. mesh step, log-log scale.

Figure 4.7: Growth of CPU time w.r.t. mesh steps in dimension 4.

Table 4.2: Values of log(CPU) time w.r.t. log($\frac{1}{h}$).

	log(CPU time) w.r.t. log($\frac{1}{h}$), d = 4						
log($\frac{1}{h}$):	1.3	1.6	1.9	2.2	2.5	slope	s_3
Problem 1	0.48	1.39	1.77	2.06	2.43	1.52	0.17
Problem 2	0.56	1.52	2.01	2.45	2.86	1.73	0.24
Problem 3	0.37	1.16	1.57	1.86	2.23	1.46	0.15
Problem 4	-0.54	0.50	1.08	1.60	2.05	1.41	0.14
Problem 5	0.33	1.21	1.63	1.90	2.28	1.52	0.17

4.A Update Operator for Fast Marching Method

We consider the problem with direction "to destination", and a semi-lagrangian discretization of the HJ equation associated to our first optimal control problem (4.2,4.8,4.9), and describe the associated update operator of the fast marching method. The first update operator in

Section 4.A.1 is based on the work of [CF07], which is shown to be efficient for isotropic case. We also provide a further update operator in Section 4.A.2 to treat certain amount of anisotropicity, which can be seen as a variant of the ordered upwind method proposed in [SV01] adapted to our case. We should mention that the anisotropicity is a major difficulty for the generalization of fast marching method, but it is beyond the scope of this work. There are many other schemes which are more efficient for certain type of anisotropy (see for instance [FLG08; Cri09; CFM11; Mir14; Mir19]) and, in principle, could be adapted to our algorithm by simply changing the update operator.

Consider the following semi-lagrangian type discretization of (4.2,4.8,4.9):

$$\begin{cases} v^h_{\to \mathrm{d}}(x) = \min\limits_{\alpha \in S_1} \left\{ (1 - \dfrac{h}{f(x,\alpha)}) v^h_{\to \mathrm{d}}(x + h\alpha) + \dfrac{h}{f(x,\alpha)} \right\}, & x \in \overline{\Omega} \setminus \mathcal{K}_{\mathrm{dst}} \ , \\ v^h_{\to \mathrm{d}}(x) = 1, & x \notin \overline{\Omega} \ , \\ v^h_{\to \mathrm{d}}(x) = 0, & x \in \mathcal{K}_{\mathrm{dst}} \ . \end{cases} \tag{4.65}$$

This is a time discretization of the optimal control problem, in which the time step is $h/f(x,\alpha)$, so depends on state and control. Note that the second equation can be restricted to the elements x of a h-neighborhood of $\overline{\Omega}$ (that is $\overline{\Omega} + B^d(0,h)$), since the first equation involves only points at a distance h of $x \in \overline{\Omega} \setminus \mathcal{K}_{\mathrm{dst}}$. Assume now given a discrete subset X of $\mathbb{R}^d$ (or of $\overline{\Omega} + B^d(0,h)$), for instance the nodes of a regular grid. Denote $V_{\to \mathrm{d}}$ the approximate value function for $v_{\to \mathrm{d}}$ on X, and apply the above equations (4.65) to all $x \in X$. When $x \in X \cap (\overline{\Omega} \setminus \mathcal{K}_{\mathrm{dst}})$, the points $x + h\alpha$ are not necessarily in X, so we need to get the value of $v^h_{\to \mathrm{d}}(x + h\alpha)$ by an interpolation of the value of its neighborhood nodes. We assume given an interpolation operator to be used in (4.65) when $x \in X \cap (\overline{\Omega} \setminus \mathcal{K}_{\mathrm{dst}})$. This interpolation may depend on x (that is on the equation to approximate), and will be denoted by $I^x[\cdot]$. However the value $I^x[V_{\to \mathrm{d}}](x')$ depends only on the values of $V_{\to \mathrm{d}}(y)$ with $y \in X$ in a neighborhood of x' (which does not depend on x). We then consider the following fully discretized semi-lagrangian scheme:

$$\begin{cases} V_{\to \mathrm{d}}(x_i) = \min\limits_{\alpha \in S_1} \left\{ (1 - \dfrac{h}{f(x_i,\alpha)}) I^{x_i}[V_{\to \mathrm{d}}](x_i + h\alpha) + \dfrac{h}{f(x_i,\alpha)} \right\} & x_i \in X \cap (\overline{\Omega} \setminus \mathcal{K}_{\mathrm{dst}}) \ , \\ V_{\to \mathrm{d}}(x_i) = 1 & x_i \notin X \cap \overline{\Omega} \ , \\ V_{\to \mathrm{d}}(x_i) = 0 & x \in X \cap \mathcal{K}_{\mathrm{dst}} \ . \end{cases} \tag{4.66}$$

4.A.1 Isotropic Case

Computing the minimum of the right hand side of the first equation in (4.66) is not trivial, especially when the dimension is high. Moreover, generally, in the d dimensional case, we need at least the value in $d+1$ nodes of the grid, in order to compute the interpolation in one node. We describe here one possible way to define an interpolation operator and to compute the minimum of the right hand side of the first equation in (4.66), within a regular grid with space mesh step equal to time step i.e., $\Delta x_i = h, \forall i \in \{1, 2, \ldots, d\}$.

Let $x = (x_1, x_2, \ldots, x_d)$ denote a point of X. Roughly speaking, the $d-$dimensional space is "partitioned" into 2^d orthants. We consider only the open orthants, since their boundaries are negligible. The values of the interpolation $I^x[V_{\to \mathrm{d}}](x + h\alpha)$ with $\alpha \in S_1$ are defined (differently) for α in each orthant, and the minimum value in each orthant is first computed. Then, the minimum will be obtained by further taking the minimum among the values in all orthants. Denote by $e_1, \ldots, e_d$ the vectors of the canonical base of $\mathbb{R}^d$. We compute the minimum in the positive orthant using $d+1$ nodes: $x^l := x + he_l, l \in \{1, \ldots, d\}$, and $x_{+1} := x + h(e_1 + e_2 + \cdots + e_d)$. The minimum in other orthants will be computed using the same method.

The interpolated value function in $x + h\alpha$ with α in the positive orthant of the sphere S_1, denoted by $v^{s,1}_{\to \mathrm{d}}$, will be given by the linear interpolation of $V_{\to \mathrm{d}}(x^1), V_{\to \mathrm{d}}(x^2), \ldots, V_{\to \mathrm{d}}(x^d)$ and $V(x_{+1})$, which is equal to

$$v^{s,1}_{\to \mathrm{d}}(x + h\alpha) = \sum_{k=1}^{d} \alpha_k V_{\to \mathrm{d}}(x^k) + \frac{V_{\to \mathrm{d}}(x_{+1}) - \sum_{l=1}^{d} V_{\to \mathrm{d}}(x^l)}{d-1} \Big((\sum_{\ell=1}^{d} \alpha_\ell) - 1 \Big) \ . \tag{4.67}$$

We then use $(\theta_1, \theta_2, \ldots, \theta_{d-1})$, $\theta_k \in (0, \frac{\pi}{2})$, to represent a vector $\alpha \in S_1$ belonging to the positive orthant, that is

$$\alpha_1 = \cos(\theta_1), \alpha_2 = \sin(\theta_1)\cos(\theta_2), \ldots, \alpha_d = \sin(\theta_1)\sin(\theta_2)\cdots\sin(\theta_{d-1}) \ . \tag{4.68}$$

This allows one to rewrite (4.67) as a function of $(\theta_1, \theta_2, \ldots, \theta_{d-1})$. By doing so, one can consider the result of the optimization in the first equation of (4.66), restricted to the positive orthant, as an approximate value of $V_{\to \mathrm{d}}(x)$, denoted by $V^1_{\to \mathrm{d}}$, and given by:

$$V^1_{\to \mathrm{d}}(x) = \min_{\theta_1, \ldots, \theta_{d-1}} \left\{ (1 - \frac{h}{f(x,\alpha)}) v^{s,1}_{\to \mathrm{d}}(x + h\alpha) + \frac{h}{f(x,\alpha)} \right\} \ . \tag{4.69}$$

Notice that the minimum in equation (4.69) is easier to compute by taking the minimum first on θ_{d-1}, then θ_{d-2}, until θ_1. Indeed, we notice in (4.68), that only the last two entries of α contain θ_{d-1}. Thus, the minimum of (4.69) over θ_{d-1} can be computed separately. Moreover, in the isotropic case, meaning $f(x,\alpha) \equiv f(x), \forall \alpha \in S_1$, the minimal θ_{d-1} is independent of $\theta_1, \ldots \theta_{d-2}$, due to the special form of (4.68) and (4.67). The iterative computation over θ_{d-2} to θ_1 will be the same.

Then, the update operator is as follows:

$$\mathcal{U}(V_{\to \mathrm{d}})(x) := \min_{k \in \{1,2,\ldots,2^d\}} V^k_{\to \mathrm{d}}(x) \ . \tag{4.70}$$

Proposition 4.A.1. *One step update using the above operator needs $O(d \times 2^d)$ arithmetic operations.*

4.A.2 Anisotropic Case: Order Upwind Method

Now we describe an update operator to treat a certain amount of anistropicity by adapting the method in [SV01], which essentially increases the size of neighborhood in each step of computation.

Let now X be a triangular mesh of $\overline{\Omega} + B^d(0,h)$ (that is the vertices of a simplicial complex covering this set) with diameter h. We denote $\Upsilon := \frac{\overline{f}}{\underline{f}} \geqslant 1$, and observe that this constant can be interpreted as a measure of anisotropicity. Let $x \in X$. For any given I-uple $(x_1, x_2, \ldots, x_I)$ of nodes of X with $I \leqslant d+1$, we define

$$x_\rho = \sum_{i=1}^{I} \rho_i x_i, \text{ for } \rho \in \Delta^I := \{\rho_i \geqslant 0, \sum_{i=1}^{I} \rho_i = 1\} \ .$$

Let us denote $d(\rho) := \|x_\rho - x\|$ and $\alpha_\rho := \frac{x_\rho - x}{\|x_\rho - x\|}$, which are the distance and the direction from x to x_ρ. Denote $[I] = \{1, 2, \ldots, I\}$ and $(x_i)_{i \in [I]}$ the I-uple. Let $V_{\to \mathrm{d}}(x; (x_i)_{i \in [I]})$ denote the approximate value of $V_{\to \mathrm{d}}(x)$ given by

$$V_{\to \mathrm{d}}(x; (x_i)_{i \in [I]}) = \min_{\rho \in \Delta^I} \left\{ \left(1 - \frac{d(\rho)}{f(x, \alpha_\rho)}\right) \left(\sum_i \rho_i V_{\to \mathrm{d}}(x_i)\right) + \frac{d(\rho)}{f(x, \alpha_\rho)} \right\} \ . \tag{4.71}$$

This is similar to the minimization in the first equation of (4.66), restricted to the elements α of S_1 that are of form α_ρ.

Let us consider the set $N(x)$ of neighborhood nodes of x, defined as follows:

$$N(x) := \{x_j \in X \mid \exists x_k \in X, \exists \tilde{x} \in [x_j, x_k], \text{ such that } x_j \sim x_k \text{ and } \|\tilde{x} - x\| \leqslant \Upsilon h\} \ ,$$

where, for $x, y \in X$, $x \sim y$ means that x and y are adjacent. Then, we can consider a new update operator for the value function defined by

$$\mathcal{U}'(V_{\to \mathrm{d}})(x) := \min V_{\to \mathrm{d}}(x; (x_i)_{i \in [I]}) \ , \tag{4.72}$$

where the minimization holds over all I-uples $(x_i)_{i \in [I]}$ of elements of $N(x)$ such that $x_1 \sim x_2$, $\ldots$, $x_{I-1} \sim x_I$ and $|I| \leqslant d + 1$.

4.B Examples with $\beta = 1$

For simplicity, we describe our examples in $\mathbb{R}^2$ and denote $B(0, 1)$ the open Euclidian ball with center 0 and radius 1.

Example 1. Consider the minimum time problem with $\Omega = B(0, 1)$, $\mathcal{K}_{\mathrm{src}}$ and $\mathcal{K}_{\mathrm{dst}}$ are the closed balls with radius $r/2$ and centers $(1 - r, 0)$ and $(-1 + r, 0)$, respectively, with $0 < r < \frac{1}{2}$. The speed function is as follows:

$$f((x, y), \alpha) = (\|\alpha\|_{L^1})^{-1} \ . \tag{4.73}$$

For this example, there exists a unique optimal trajectory follows a straight line with $y = 0$. Moreover, the dynamics with a speed as in (4.73) is equivalent to a dynamics with constant speed and a direction α chosen in the unit L^1 sphere instead of the L^2 sphere. One can also deduce that the distance between the optimal trajectory and a $\delta-$optimal trajectory is $\Delta = O(\delta)$. Indeed, in this case, the original minimum time problem can be approximated by a true shortest path problem.

Example 2. Consider the example of Problem 4 in dimension2. Recall that $\Omega = B(0, 1)$, and that $\mathcal{K}_{\mathrm{src}}$ and $\mathcal{K}_{\mathrm{dst}}$ are the closed balls with radius $r/2$ and centers $(1 - r, 0)$ and $(-1 + r, 0)$, respectively, with $0 < r < \frac{1}{2}$, and that the speed is

$$f((x, y), \alpha) = 1 + x^2 - y^2 \ . \tag{4.74}$$

One can easily notice that the optimal trajectory between $\mathcal{K}_{\mathrm{src}}$ and $\mathcal{K}_{\mathrm{dst}}$ follows again a straight line with $y = 0$. Moreover, if a $\delta-$optimal trajectory is at a distance Δ of the optimal trajectory, then it goes through a point of the form $(x, \pm\Delta)$, by symmetry. Without loss of generality, let us consider a δ-optimal trajectory which is optimal from $\mathcal{K}_{\mathrm{src}}$ to (x, Δ), and from (x, Δ) to $\mathcal{K}_{\mathrm{dst}}$. We take the optimal trajectory from $\mathcal{K}_{\mathrm{src}}$ to (x, Δ) and from (x, Δ) to $\mathcal{K}_{\mathrm{dst}}$, because it gives the maximum Δ, which is then the maximum distance from the optimal trajectory, for which the δ-optimal trajectory of the minimum time problem can reach. One can then characterize the optimal trajectory from $\mathcal{K}_{\mathrm{src}}$ to (x, Δ) and from (x, Δ) to $\mathcal{K}_{\mathrm{dst}}$, for instance by Pontryagin Maximum Principle, then deduce the travel time of this trajectory which is the sum of the minimum time from $\mathcal{K}_{\mathrm{src}}$ to (x, Δ) and from (x, Δ) to $\mathcal{K}_{\mathrm{dst}}$. Using the fact that this trajectory is δ-optimal, we deduce $\Delta = O(\delta)$, showing that $\beta = 1$.

4.C Detailed Numerical Data

4.C.1 Detailed Numerical Data for Problem 1

The exact solution is the Euclidean distance, denoted by v^*. The relative error, CPU time and memory allocation of our data structure for the classical fast-marching method and for the

multi-level method are shown in Table 4.3. The relative error is defined as error := $\frac{|v^h - v^*|}{v^*}$, where v^h is the approximate value computed by a numerical scheme with (finest) mesh step h. We also give the CPU time and the memory allocation needed for the fine grid mesh changes

Table 4.3: Problem 1. CPU times and Memory Allocation as a function of the relative error and of the dimension.

		Classical F.M.	Multi-level F.M. (Algorithm 4.3)				
			2-level	3-level	4-level	5-level	6-level
dimension-2							
$h = \frac{1}{50}$	CPU Time(s)	0.113	0.0343	0.031	–	–	–
error: 2.71%	Memory(MB)	6.4	6.4	6.0	–	–	–
$h = \frac{1}{100}$	CPU Time(s)	0.381	0.122	0.103	–	–	–
error: 1.42%	Memory(MB)	8.1	8.3	8.3	–	–	–
dimension-3							
$h = \frac{1}{50}$	CPU Time(s)	7.78	1.79	1.53	1.45	–	–
error : 3.85%	Memory(MB)	34.1	21.7	20.9	19.7	–	–
$h = \frac{1}{100}$	CPU Time(s)	74.69	8.41	6.17	5.21	–	–
error: 2.12%	Memory(MB)	182.2	57.8	47.9	43.6	–	–
dimension-4							
$h = \frac{1}{50}$	CPU Time(s)	497.03	53.6	39.8	36.73	–	–
error: 4.49%	Memory(MB)	1167.4	194.2	131.7	146.7	–	–
$h = \frac{1}{100}$	CPU Time(s)	–	364.84	220.32	121.45	77.32	50.91
error: 2.34%	Memory(MB)	–	1198.1	673.3	644.5	663.2	593.1
dimension-5							
$h = \frac{1}{50}$	CPU Time(s)	–	–	473.59	235.32	180.48	158.27
error: 5.49%	Memory(MB)	–	–	1105.9	947.9	781.4	1025
$h = \frac{1}{100}$	CPU Time(s)	–	–	–	1031.18	465.96	303.92
error: 2.89%	Memory(MB)	–	–	–	1679.4	1863.68	2109.2
dimension-6							
$h = \frac{1}{50}$	CPU Time(s)	–	–	–	1260.37	683.36	434.36
error: 7.12%	Memory(MB)	–	–	–	2764.8	3686.4	3543.3
$h = \frac{1}{100}$	CPU Time(s)	–	–	–	–	2485.79	1329.56
error: 3.94%	Memory(MB)	–	–	–	–	5939.2	4765.4

from 20 to 320 in Table 4.4, in which the relative error are exactly the same for both algorithms with same mesh step.

4.C.2 Detailed Numerical Data for Problem 2

We give in Figure 4.8 the numerical results for the classical fast marching method and our multi-level method to compare with the Problem 2. This shows the gain in performance persists in the presence of multiple optimal trajectories.

We also test our algorithm with different error estimation in dimension-3 and dimension-4 cases, with the level be different for different dimension and error bound. We give the result of CPU time and memory allocation needed for the fine grid mesh changes from 20 to 320 in Figure 4.9.

The detailed numerical results for Problem 2, with respect to different dimensions, are in Table 4.5:

We also compared, with the dimension to be fixed at 3 and 4, the CPU time and Memory Allocation for different relative error in Table 4.6.

4.C.3 Detailed Numerical Data for Problem 3

The numerical results with respect to different dimensions are shown in Figure 4.10.

For different error estimation in dimension-3 and dimension-4, the results are shown in Figure 4.11.

Table 4.4: Problem 1. CPU times and Memory Allocation for different precisions.

	dimension-3		dimension-4	
	CPU Time(s)	Memory(MB)	CPU Time(s)	Memory(MB)
$h = \frac{1}{20}$				
Classical F.M.	0.465	7.3	11.486	56.1
2-level F.M.	0.225	7.7	4.81	25.9
3-level F.M.	0.185	7.8	2.99	28.2
$h = \frac{1}{40}$				
Classical F.M.	3.869	20.4	225.502	695.2
3-level F.M.	1.168	14.2	28.94	104.9
4-level F.M.	0.913	16.8	24.68	107.9
$h = \frac{1}{80}$				
Classical F.M.	35.776	104.3	4241.85	9431.1
4-level F.M.	4.358	34.7	102.42	409.1
5-level F.M.	3.961	26.4	59.23	491.5
$h = \frac{1}{160}$				
Classical F.M.	258.67	724.8	–	
5-level F.M.	14.22	62.2	203.82	1116.2
6-level F.F.	10.8	73.8	115.41	846.5
$h = \frac{1}{320}$				
Classical F.M.	2328.57	5734.4	–	
5-level F.M.	55.17	194.4	1095.63	2068.5
6-level F.M.	31.84	227.0	438.37	1658.9
7-level F.M.	23.31	316.6	267.61	1470.1

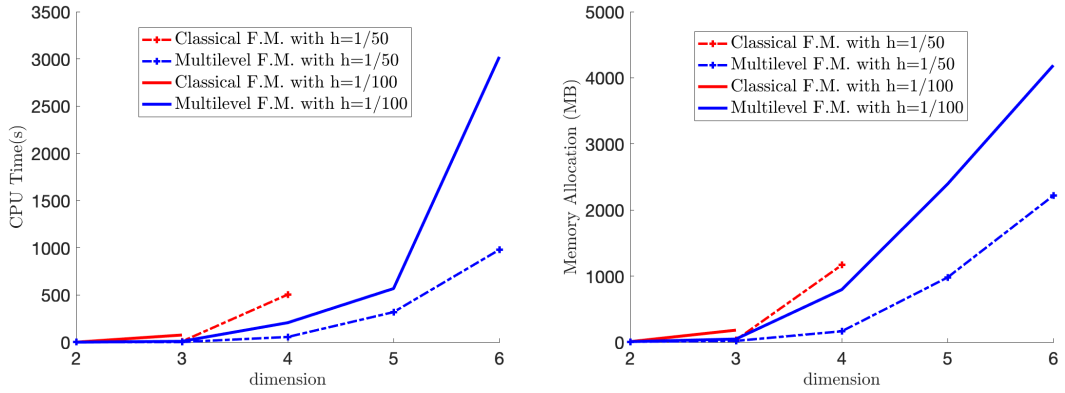
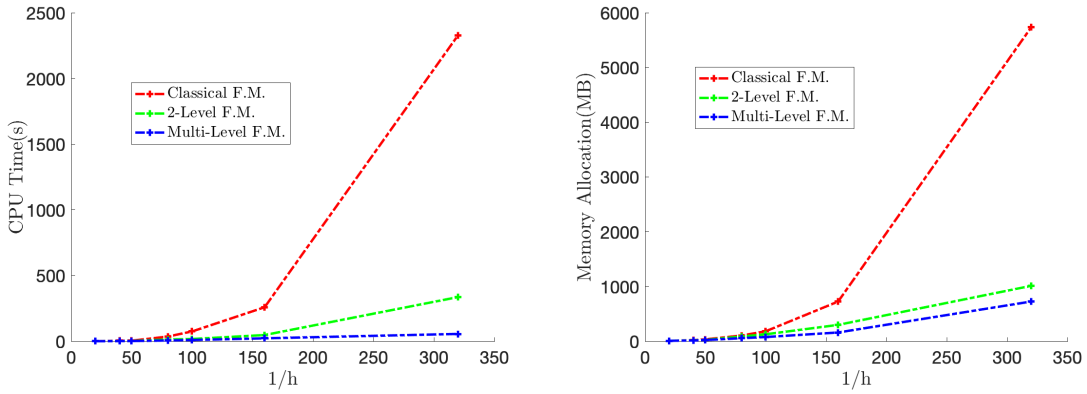


Figure 4.8: Problem 2. CPU time (left) and memory allocation (right) as a function of the dimension, for a fixed precision.



(a) CPU time for different $\frac{1}{h}$

(b) Memory allocation for different $\frac{1}{h}$

Figure 4.9: Problem 2. CPU time and memory needed to get certain error bound.

Table 4.5: Problem 2. Varying the dimension.

		Classical F.M.	Multi-level F.M.				
			2-level	3-level	4-level	5-level	6-level
dimension-3							
$h = \frac{1}{50}$	CPU Time(s)	8.06	2.20	1.88	1.66	–	–
error: 4.19%	Memory(MB)	34.1	20.2	19.7	19.8	–	–
$h = \frac{1}{100}$	CPU Time(s)	75.89	17.08	13.14	9.36	–	–
error: 2.22%	Memory(MB)	182.2	79.9	73.7	47.6	–	–
dimension-4							
$h = \frac{1}{50}$	CPU Time(s)	504.5	115.84	84.44	55.29	–	–
error: 5.13%	Memory(MB)	1167.4	230.2	191.9	165.1	–	–
$h = \frac{1}{100}$	CPU Time(s)	–	854.8	585.26	305.55	238.26	206.04
error: 2.67%	Memory(MB)	–	1351.7	1321.9	892.1	823.1	795.3
dimension-5							
$h = \frac{1}{50}$	CPU Time(s)	–	–	1731.27	974.32	478.48	318.66
error: 5.87%	Memory(MB)	–	–	2107.1	1291.7	1149.8	979.6
$h = \frac{1}{100}$	CPU Time(s)	–	–	–	3254.27	1245.71	567.98
error: 3.31%	Memory(MB)	–	–	–	3771.89	2487.32	2392.48
dimension-6							
$h = \frac{1}{50}$	CPU Time(s)	–	–	–	3951.47	2135.35	979.37
error: 7.21%	Memory(MB)	–	–	–	3771.89	2487.32	2217.39
$h = \frac{1}{50}$	CPU Time(s)	–	–	–	–	–	3021.91
error: 4.10%	Memory(MB)	–	–	–	–	–	4189.6

Table 4.6: Problem 2. Varying the step size.

	dimension-3		dimension-4	
	CPU Time(s)	Memory(MB)	CPU Time(s)	Memory(MB)
$h = \frac{1}{20}$				
Classical F.M.	0.453	7.3	11.374	56.1
2-level F.M.	0.279	8.1	7.63	26.1
3-level F.M.	0.237	8.9	3.62	28.9
$h = \frac{1}{40}$				
Classical F.M.	3.974	20.4	223.707	695.2
3-level F.M.	1.655	17.6	41.546	172.5
4-level F.M.	1.197	17.8	33.405	175.6
$h = \frac{1}{80}$				
Classical F.M.	36.897	104.3	4238.29	9431.1
4-level F.M.	6.371	41.3	245.169	956.5
5-level F.M.	5.631	38.4	102.52	899.7
$h = \frac{1}{160}$				
Classical F.M.	261.34	724.7	–	
5-level F.M.	29.36	170.4	551.19	1812.5
6-level F.F.	22.30	124.5	279.85	1516.29
$h = \frac{1}{320}$				
Classical F.M.	2335.71	5734.4	–	
5-level F.M.	131.42	493.7	–	
6-level F.M.	70.85	437.4	1426.5	3983.4
7-level F.M.	55.64	445.6	726.9	2765.1

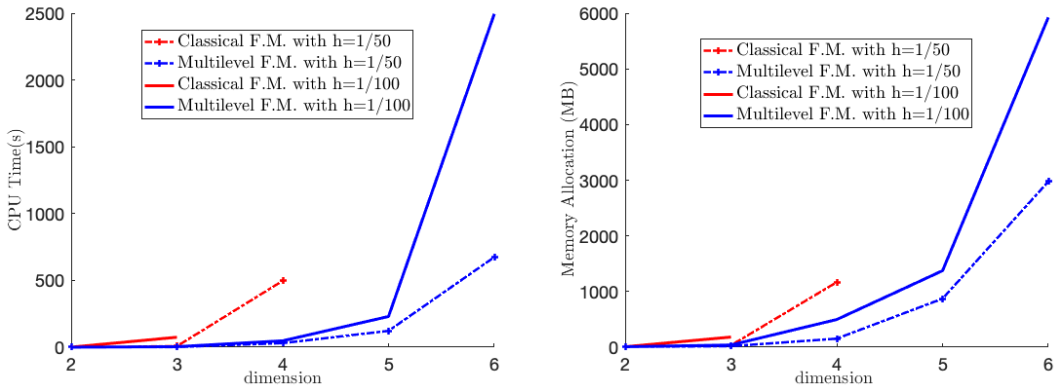
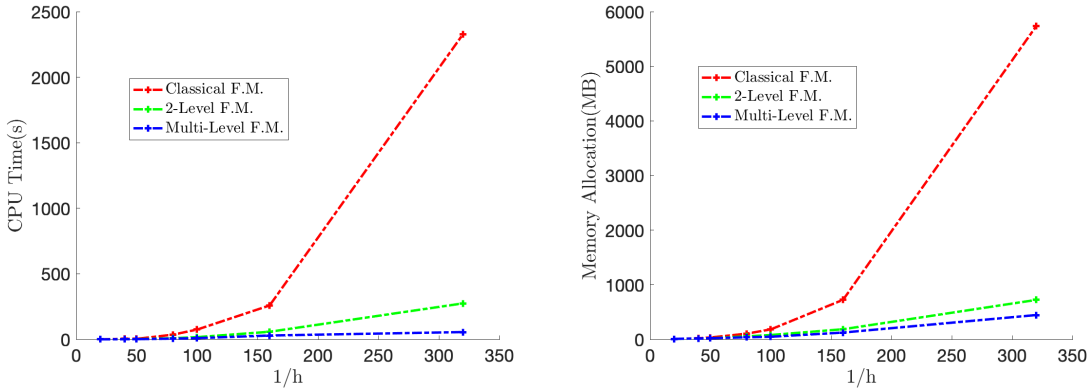


Figure 4.10: Problem 3. CPU time (left) and memory allocation (right) as a function of the dimension, for a fixed precision.



(a) CPU time for different $\frac{1}{h}$

(b) Memory allocation for different $\log(\frac{1}{h})$

Figure 4.11: Problem 3. CPU time and memory needed to get certain error bound.

The detailed numerical data for Problem 3, with respect to different dimensions, are in Table 4.7.

Table 4.7: Problem 3. Varying the dimension.

		Classical F.M.	Multi-level F.M.				
			2-level	3-level	4-level	5-level	6-level
dimension-2							
$h = \frac{1}{50}$	CPU Time(s)	0.109	0.041	–	–	–	–
	Memory(MB)	6.4	6.4	–	–	–	–
$h = \frac{1}{100}$	CPU Time(s)	0.394	0.081	–	–	–	–
	Memory(MB)	8.1	8.1	–	–		–
dimension-3							
$h = \frac{1}{50}$	CPU Time(s)	8.14	1.25	1.04	–	–	–
	Memory(MB)	34.1	23.1	18.7	–	–	–
$h = \frac{1}{100}$	CPU Time(s)	74.79	8.00	5.83	3.94	–	–
	Memory(MB)	182.1	84.5	62.0	39.9	–	–
dimension-4							
$h = \frac{1}{50}$	CPU Time(s)	507.2	48.77	32.29	30.18	–	–
	Memory(MB)	1276.3	177.4	148.6	153.8	–	–
$h = \frac{1}{100}$	CPU Time(s)	–	307.8	175.2	117.4	69.81	47.85
	Memory(MB)	–	1217.4	670.1	521.6	540.9	497.6
dimension-5							
$h = \frac{1}{50}$	CPU Time(s)	–	–	427.9	220.8	159.3	122.1
	Memory(MB)	–	–	1447.2	1102.4	925.9	872.4
$h = \frac{1}{100}$	CPU Time(s)	–	–	–	1171.4	495.7	397.8
	Memory(MB)	–	–	–	2115.7	1414.3	1374.9
dimension-6							
$h = \frac{1}{50}$	CPU Time(s)	–	–	–	1974.9	924.6	674.9
	Memory(MB)	–	–	–	2713.4	2249.5	2074.9
$h = \frac{1}{100}$	CPU Time(s)	–	–	–	–	–	2494.2
	Memory(MB)	–	–	–	–	–	5924.2

The result with fixed dimensions and various mesh steps are given in Table 4.8

4.C.4 Detailed Numerical Data for Problem 4

The following Figure 4.12 shows the CPU time and memory needed to get certain accuracy with respect to different dimensions.

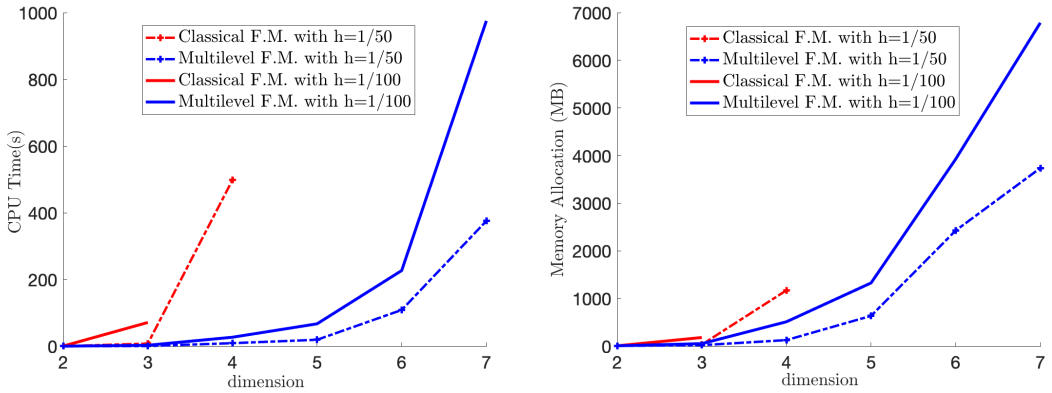


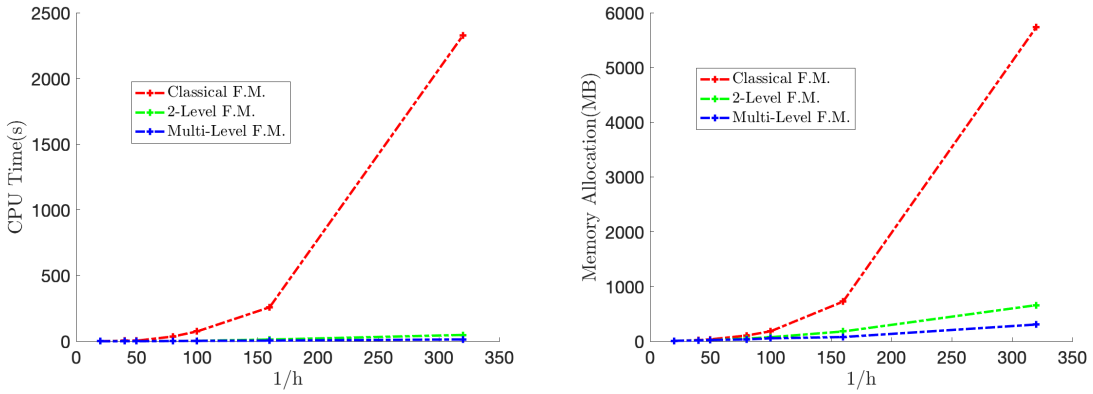
Figure 4.12: Problem 4. CPU time (left) and memory allocation (right) as a function of the dimension, for a fixed precision.

We also fix the dimension as 3 and 4, and vary the step size, the results are shown in Fig. 4.13.

The detailed numerical data for Problem 4 with respect to different dimensions are in Table 4.9.

Table 4.8: Problem 3. Varying the step size.

	dimension-3		dimension-4	
	CPU Time(s)	Memory(MB)	CPU Time(s)	Memory(MB)
$h = \frac{1}{20}$				
Classical F.M.	0.571	7.1	11.26	55.9
2-level F.M.	0.281	8.6	2.77	28.4
3-level F.M.	0.209	8.2	2.37	23.6
$h = \frac{1}{40}$				
Classical F.M.	3.96	20.2	221.3	695.4
3-level F.M.	0.974	15.5	17.31	82.3
4-level F.M.	0.875	15.2	14.34	23.6
$h = \frac{1}{80}$				
Classical F.M.	37.81	105.9	4298.77	9496.5
4-level F.M.	3.63	27.7	66.43	421.9
5-level F.M.	2.72	24.6	37.21	404.9
$h = \frac{1}{160}$				
Classical F.M.	262.78	729.7	–	
5-level F.M.	12.71	71.4	122.71	776.4
6-level F.F.	9.26	62.9	72.72	721.5
$h = \frac{1}{320}$				
Classical F.M.	2479.76	5757.4	–	
5-level F.M.	27.51	169.5	906.43	2776.4
6-level F.M.	20.31	165.4	309.44	1257.4
7-level F.M.	16.34	176.8	169.92	1201.5



(a) CPU time for different $\frac{1}{h}$

(b) Memory allocation for different $\frac{1}{h}$

Figure 4.13: Problem 4. CPU time and memory needed to get certain error bound.

Table 4.9: Problem 4. Varying the dimension.

		Classical F.M.	Multi-level F.M.				
			2-level	3-level	4-level	5-level	6-level
dimension-3							
$h = \frac{1}{50}$	CPU Time(s)	7.82	0.47	–	–	–	–
	Memory(MB)	33.9	21.7	–	–	–	–
$h = \frac{1}{100}$	CPU Time(s)	71.12	3.24	2.87	–	–	–
	Memory(MB)	182.2	73.8	53.9	–	–	–
dimension-4							
$h = \frac{1}{50}$	CPU Time(s)	498.2	15.56	13.61	9.08	–	–
	Memory(MB)	1172.6	134.7	148.8	128.3	–	–
$h = \frac{1}{100}$	CPU Time(s)	–	89.06	55.73	40.27	26.89	–
	Memory(MB)	–	1021.4	1121.9	706.8	512.7	–
dimension-5							
$h = \frac{1}{50}$	CPU Time(s)	–	43.79	29.07	24.27	19.27	–
	Memory(MB)	–	737.5	596.7	494.9	636.2	–
$h = \frac{1}{100}$	CPU Time(s)	–	–	629.61	223.77	113.04	67.1
	Memory(MB)	–	–	3429.8	1785.7	1527.6	1327.9
dimension-6							
$h = \frac{1}{50}$	CPU Time(s)	–	–	–	388.94	173.45	108.6
	Memory(MB)	–	–	–	2084.6	2072.8	2427.9
$h = \frac{1}{100}$	CPU Time(s)	–	–	–	–	396.37	226.85
	Memory(MB)	–	–	–	–	4241.5	3927.6
dimension-7							
$h = \frac{1}{50}$	CPU Time(s)	–	–	–	–	725.83	375.45
	Memory(MB)	–	–	–	–	3628.6	3738.2
$h = \frac{1}{100}$	CPU Time(s)	–	–	–	–	1426.22	975.38
	Memory(MB)	–	–	–	–	5228.9	6787.8

The results with fixed dimensions and various mesh steps are in Table 4.10

4.C.5 Detailed Numerical Data for Problem 5

The results with respect to different dimensions are shown in Figure 4.14.

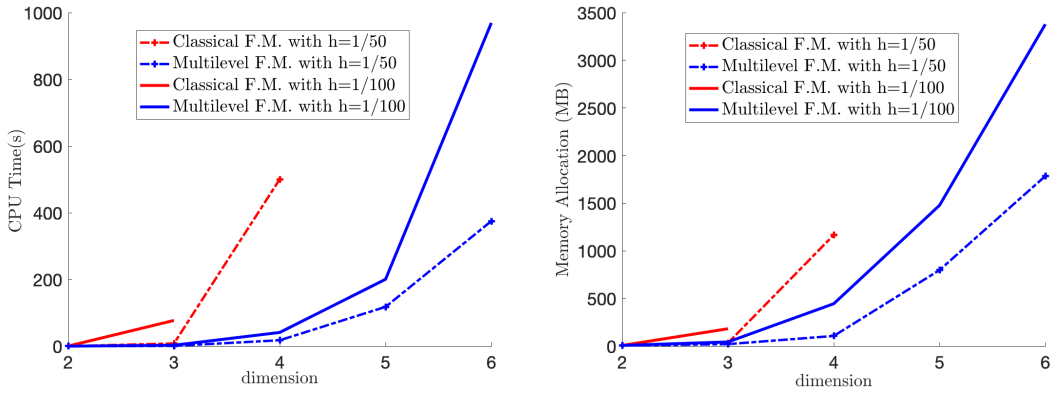


Figure 4.14: Problem 5. CPU time (left) and memory allocation (right) as a function of the dimension, for a fixed precision.

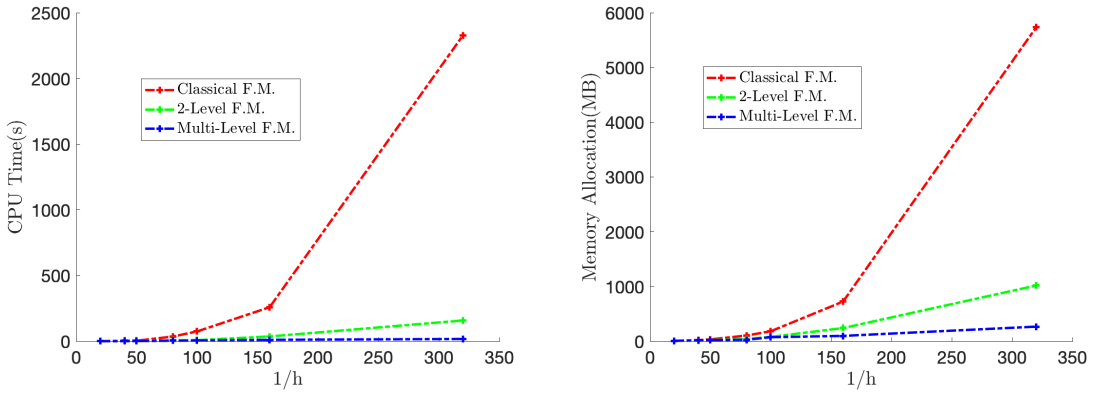
We again fix the dimension as 3 and 4, and vary the step size. The results are shown in Figure 4.15.

The detailed numerical data for Problem 5 with respect to different dimensions are in Table 4.11.

The result with fixed dimension and various step size is given in Table 4.12

Table 4.10: Problem 4. Varying the step size.

	dimension-3		dimension-4	
	CPU Time(s)	Memory(MB)	CPU Time(s)	Memory(MB)
$h = \frac{1}{20}$				
Classical F.M.	0.432	7.1	11.354	55.9
2-level F.M.	0.148	8.3	0.512	24.5
3-level F.M.	0.062	8.2	0.294	22.5
$h = \frac{1}{40}$				
Classical F.M.	3.978	20.2	219.076	694.9
3-level F.M.	0.395	15.8	4.72	70.2
4-level F.M.	0.289	14.7	3.17	70.9
$h = \frac{1}{80}$				
Classical F.M.	36.492	104.1	4236.27	9487.2
4-level F.M.	1.684	28.1	27.94	792.2
5-level F.M.	1.372	34.2	21.09	473.9
$h = \frac{1}{160}$				
Classical F.M.	260.36	724.7	–	
5-level F.M.	5.081	99.8	72.63	1392.5
6-level F.F.	4.781	76.9	47.05	935.1
$h = \frac{1}{320}$				
Classical F.M.	2443.21	5754.2	–	
5-level F.M.	25.15	337.9	749.85	3376.2
6-level F.M.	16.83	365.3	201.30	2072.8
7-level F.M.	13.77	305.5	101.69	1527.7



(a) CPU time for different $\frac{1}{h}$

(b) Memory allocation for different $\frac{1}{h}$

Figure 4.15: Problem 5. CPU time and memory for certain error bound.

Table 4.11: Problem 5. Varying the dimension.

		Classical F.M.	Multi-level F.M.				
			2-level	3-level	4-level	5-level	6-level
dimension-3							
$h = \frac{1}{50}$	CPU Time(s)	8.16	1.27	–	–	–	–
	Memory(MB)	34.3	21.8	–	–	–	–
$h = \frac{1}{100}$	CPU Time(s)	77.21	3.49	2.92	–	–	–
	Memory(MB)	183.2	57.8	44.6	–	–	–
dimension-4							
$h = \frac{1}{50}$	CPU Time(s)	500.7	27.39	21.49	17.85	–	–
	Memory(MB)	1167.9	137.2	112.7	107.4	–	–
$h = \frac{1}{100}$	CPU Time(s)	–	201.11	140.71	98.75	60.21	41.25
	Memory(MB)	–	1091.2	560.8	498.4	490.7	445.7
dimension-5							
$h = \frac{1}{50}$	CPU Time(s)	–	–	356.46	192.45	140.25	117.89
	Memory(MB)	–	–	1205.8	902.4	820.9	798.8
$h = \frac{1}{100}$	CPU Time(s)	–	–	–	627.85	299.37	200.59
	Memory(MB)	–	–	–	1508.9	1489.2.32	1478.3
dimension-6							
$h = \frac{1}{50}$	CPU Time(s)	–	–	–	792.85	407.85	225.09
	Memory(MB)	–	–	–	1927.6	1972.4	1785.4
$h = \frac{1}{50}$	CPU Time(s)	–	–	–	–	2049.75	969.28
	Memory(MB)	–	–	–	–	3921.6	3378.5

Table 4.12: Problem 5. Varying the step size.

	dimension-3		dimension-4	
	CPU Time(s)	Memory(MB)	CPU Time(s)	Memory(MB)
$h = \frac{1}{20}$				
Classical F.M.	0.483	7.3	11.984	57.1
2-level F.M.	0.215	7.8	2.95	20.3
3-level F.M.	0.177	7.8	2.14	24.5
$h = \frac{1}{40}$				
Classical F.M.	3.874	20.8	227.371	699.4
3-level F.M.	1.109	18.7	19.68	89.2
4-level F.M.	0.925	15.8	16.38	87.9
$h = \frac{1}{80}$				
Classical F.M.	37.771	105.6	4276.81	9550.8
4-level F.M.	3.605	27.8	67.41	399.2
5-level F.M.	2.619	22.4	42.94	371.4
$h = \frac{1}{160}$				
Classical F.M.	264.37	725.8	–	
5-level F.M.	10.65	67.2	130.88	879.24
6-level F.F.	8.34	57.9	80.03	665.21
$h = \frac{1}{320}$				
Classical F.M.	2435.91	5734.4	–	
5-level F.M.	29.38	178.4	974.85	2215.6
6-level F.M.	19.39	175.3	327.5	1486.7
7-level F.M.	17.23	154.8	191.3	1241.6

Chapter 5

Convergence and Error Estimates of A Semi-Lagrangian scheme for the Eikonal Equation

***

Abstract. We present a modification of the semi-lagrangian scheme focusing on solving the general eikonal equation that arises from the front propagation and minimum time problem with a given target. We show that the discrete-time value function associated with our discretization scheme is semiconcave under certain regularity assumptions on the dynamics and the target set. We show a convergence rate of order 1 for both the semi-lagrangian scheme and a fully

discretized semi-lagrangian scheme, in terms of the time step for the first scheme and of the mesh step for the latter one. We also establish convergence results under a particular state constraint. We apply our results to analyze the convergence rate and computational complexity of the fast-marching method, and of the multi-level fast-marching method recently introduced by the authors.

5.1 Introduction

5.1.1 Motivation and Context

This chapter discusses a numerical approach for solving a general eikonal equation, which is also a static first order Hamilton-Jacobi(HJ) Partial Differential Equation(PDE) arising in front propagation and minimum time problems. The value function for such problems is characterized as the solution of the associated HJ equation in the viscosity sense [CL83; CEL84; FS06]. Problems with state constraints can be addressed with the notion of constrained viscosity solution [Son86a; Son86b].

One class of numerical methods for solving HJ equations involves *Semi-Lagrangian schemes*, as in [Fal87; FF14], which arise by applying the Bellman dynamic programming principle to the discrete time optimal control problem obtained after an Euler time-discretization of the dynamics. The convergence of such a scheme should be understood in viscosity sense. Moreover, a convergence rate of order 1/2 of the time step is established under mild condition on the problems, and of order 1 is established typically under a semiconcavity condition [DI84], or a bounded variational condition [Fal87]. The Semi-Lagrangian scheme originally involves a semi-discretization in time. For practical computation, a further discretization in the state space is needed, which is often referred to the fully discretized scheme. After a space discretization (using a grid), the usual system of equations can be interpreted as the dynamic programming equation of a stochastic optimal control problem [KD01] with discrete time and state space. One can solve the discretized system by applying value iteration until convergence. Each iteration consists in updating the value function at nodes in a given grid by solving the corresponding discrete HJ equation. The convergence result of fully discretized scheme is often obtained as both the time step and the space step over time step tend to 0. Several works intend to show the convergence and the convergence rate of such schemes are proposed, for which we mention the works of Bardi and Falcone [BF90], of Falcone and Ferretti [FF94], of Grüne [Grü97], of Bokanowski, Megdich and Zidani[BMZ10].

The fast-marching method was originally introduced in [Tsi95] and [Set96] as an acceleration method in the case of monotone and causal discretizations of the eikonal equation. The method takes advantage of the property that the evolution of the region behind a "propagation front" is monotonically non-decreasing, allowing one to focus only on the computation around the front at each iteration. Specifically, the value function is computed by visiting the grid nodes in a special order, which is chosen so that the value function is monotone non-decreasing in the direction of propagation. Owing to this property, the fast-marching method is known as a "single pass" method. At every point of the discretization grid, the value function is updated at most k times, where k is a constant not related to the discretization mesh. The computational complexity of the fast-marching method is shown to be $O(K_d M \log(M))$ in terms of arithmetic operations, where M is the number of grid nodes and K_d is a constant that depends on the type of discretization neighborhood. Thus, considering for instance a d-dimensional grid with mesh step h, the computational complexity is $\widetilde{O}((\frac{1}{h})^d)$, where $\widetilde{O}$ ignores the logarithm factors. Moreover, under a "causality" condition, the fast-marching method computes the same solution as the one obtained by the standard value iteration method. In some previous workson the

fast marching method, by Sethian and Vladimirsky [SV03], Cristiani and Falcone [CF07], Carlini, Falcone, Forcadel and Monneau [Car+08] and Mirebeau [Mir14], the authors proved the convergence of their methods when the mesh step h goes to 0 without an explicit convergence rate. More recently, Shum, Morris, and Khajepour [SMK16], and Mirebeau [Mir19], established a convergence rate of order $h^{\frac{1}{2}}$. Though, most of numerical experiments in the above works reveal an actual convergence rate of order h. One of the purposes of the present chapter is to establish sufficient conditions for achieving a convergence rate of order 1.

One major difficulty which occurs with nonnegative cost functions for general problems comes from the "anisotropy". Indeed the "causality" property naturally holds for usual discretizations of isotropic equations, but it is hardly extended to anisotropic cases. Several studies intended to overcome this difficulty. In particular, in [SV03; Vla06], the authors extended the fast-marching method to handle some amount of anisotropicity by increasing the sets of neighborhood points for every node in the grid. However their methods could only deal with a certain class of equations, and the larger neighborhood increases the computational complexity. In[Cri09], the authors proposed a method called *buffered fast marching* method which is suitable for both general Hamilton-Jacobi-Bellman equations and Hamilton-Jacobi-Isaacs equations. They introduce an iterative step in the set "buffer", which contains the nodes pre-selected from the set "narrow bound" before adding to the set "accepted". The authors showed that this method is as effective as fast marching method in most experiments. But this method is technically not a single pass one, because the iterative procedure depends on the mesh step, and in fact in the worst case the computational complexity is even greater than the one of value iterative method. In [Mir14; Mir18; Mir19], Mirebeau extended the fast marching method to some 2-D and 3-D elliptic anisotropic cases, and other types of degenerate anelliptic cases related to curvature penalization. His method is based on discretization using adaptive stencil adapted to the Hamiltonian and associated Voronoi's first reduction of quadratic forms. The computational complexity of the algorithm in [Mir14; Mir18; Mir19] is $O(M \ln M + M \ln k)$, where k is the maximum anisotropic ratio. Other works intending to generalize the fast marching method include [CFM11; For09; FLG08].

As we can see from the computational complexity, the fast-marching method still suffers from the "curse of dimensionality". Indeed, the size of nonlinear systems to be solved is exponential in the dimension d, making the numerical computation untractable even on modern computers. Several types of discretizations or representations have been developed recently to overcome the curse of dimensionality for HJ equations. One may cite the sparse grids approximations in [Bok+13; KW17], the tensor decompositions in [DKK21; OSS22] the deep learning techniques in [DM21].In the case of structured problems, one may also cite the max-plus or tropical numerical method in [McE07; Qu13; Dow18], and the Hopf formula in [DO16].

Another way to overcome the curse of dimensionality is to focus on finding one (or several) particular optimal trajectories. The latter problem can be solved, under some convexity assumptions, by the Pontryagin Maximum Principle approach [RZ98; RZ99; BZ99], or the stochastic dual dynamic programming (SDDP) [PP91; Sha11; GLP15]. In the absence of convexity assumptions, these methods may only lead to a local minimum. In that case, more recent methods consist in exploiting the structure of the problem, in order to reduce the set of possible trajectories among which the optimization is done, see for instance [AFS19; AFS20] and [BGZ22]. In [AGL23a], we introduced a multi-level fast-marching method, which focuses on the neighborhood of optimal trajectories. Under some assumptions on the convergence rate of the fast-marching method (without or with a particular state constraint) and on the stiffness of the value function, we obtained that this method has a complexity of the same order as for one-dimensional problems. So one of the aims of the present work is to give sufficient conditions on the problem allowing to apply the results of [AGL23a].

5.1.2 Contribution

We present a semi-lagrangian scheme for solving the eikonal equation arising from the minimum time problem of reaching a target set $\mathcal{K}$, in which the time step varies depending on the state. We show that the solution of the discretized system corresponds to the value function of a discrete-time deterministic control problem. Moreover, we show that, under certain regularity assumptions on the dynamics and the target set $\mathcal{K}$, the discrete value function is semiconcave. This result leads to a convergence rate of order 1 concerning the time step for the semi-Lagrangian scheme, which involves a semi-discretization in time. Furthermore, we consider a fully discretized scheme, involving discretizing the state space using a mesh grid. We show that the solution of the fully discretized system is the value function of a controlled Markov problem. We show that, under further regularity assumptions on the dynamics and the target set $\mathcal{K}$, the error between the solution of the fully discretized system and the solution of the semi-discretized system is of the order of 1 with respect to the mesh step, using particular interpolation operators. This result yields a convergence rate of order 1 for the fully discretized scheme, in terms of both time step and mesh step. We also establish that the convergence results hold under a particular state constraint introduced in [AGL23a], which forces trajectories to stay in a tubular neighborhood of optimal trajectories.

As a consequence of the above results, we show a convergence rate of order 1 for the fast-marching method, which uses update operators derived from the (fully discretized) semi-lagrangian type discretization of the corresponding eikonal equation, for instance in the works of [SV03; CF07; SMK16; Mir19]. As a result, the computational complexity of the fast-marching method, as a function of the error bound ε, is in the order of $\widetilde{O}((C\frac{1}{\varepsilon})^d)$, where $C > 0$ is a constant that will be detailed. Moreover, we also show the computational complexity of the multi-level fast-marching method introduced in [AGL23a] is in the order of $\widetilde{O}(C'^d(\frac{1}{\varepsilon})^{1+(d-1)(1-\beta)})$, where $C' > 0$ is a constant, and $0 < \beta \leqslant 1$ measures the "stiffness" of the value function around the optimal trajectory. Thus, for the problems with $\beta = 1$, the ideal complexity bound in [AGL23a] is achieved, meaning that the complexity becomes $\widetilde{O}(\frac{1}{\varepsilon})$ and is of the same order as for one dimensional problems.

This chapter is organized as follows: In Section 5.2, we provide preliminary results on the HJ equation and the minimum time optimal control problem. In Section 5.3, we then establish a discrete time optimal control problem associated with the discretization scheme. An improved convergence result is obtained using the semiconcavity property of the discrete time value function, which is obtained under semiconcave assumptions on the dynamics and target set. In Section 5.4, we present a fully discretized scheme. We represent the solution of the fully discretized system as the value function of a controlled Markov problem. We then show the convergence rate under particular interpolation operators. As an application we analyze the computational complexity of a fast-marching method with update operator derived from the fully discretized scheme. In Section 5.5, we demonstrate convergence results within a particular state constraint, and then apply the results to analyze the computational complexity of the multi-level fast-marching method.

5.2 Preliminaries

5.2.1 The Eikonal Equation

Let $\mathcal{K}$ be a compact set in $\mathbb{R}^d$. Let S_1 be the unit sphere in $\mathbb{R}^d$, i.e., $S_1 = \{x \in \mathbb{R}^d, \|x\| = 1\}$ where $\|\cdot\|$ denotes the Euclidean norm. We consider an eikonal equation of the form:

$$\begin{cases} -(\min\limits_{\alpha \in S_1}\{(\nabla T(x)\cdot\alpha) f(x,\alpha)\} + 1) = 0, & x \in \mathbb{R}^d \setminus \mathcal{K}, \\ T(x) = 0, & x \in \partial\mathcal{K}\ , \end{cases} \tag{5.1}$$

where f is the speed function, and we assume the following basic regularity properties:

Assumption (A7)

(i) $f : \mathbb{R}^d \times S_1 \mapsto \mathbb{R}_{>0}$ is continuous.

(ii) f is bounded, i.e., $\exists M_f > 0$ s.t. $|f(x,\alpha)| \leqslant M_f$, $\forall x \in \mathbb{R}^d, \forall \alpha \in S_1$.

(iii) There exists constants $L_f, L_{f,\alpha} > 0$ such that $|f(x,\alpha) - f(x^{'},\alpha)| \leqslant L_f|x - x^{'}|, \forall \alpha \in S_1, \forall x, x^{'} \in \mathbb{R}^d$ and $|f(x,\alpha) - f(x,\alpha^{'})| \leqslant L_{f,\alpha}|\alpha - \alpha^{'}|, \forall x \in \mathbb{R}^d, \forall \alpha, \alpha^{'} \in S_1$.

The function $T : \mathbb{R}^d \to \mathbb{R}$ represents the minimum time required for a point $x \in \mathbb{R}^d \setminus \mathcal{K}$ to reach $\mathcal{K}$, while traveling with a state-dependent speed given by the function f. Such an eikonal equation is typically associated with the front propagation problem, which involves the evolution of the boundary of a domain, denoted by Γ_t, as described by T. In particular, the boundary of the domain Ω_t can be defined as $\Gamma_t = \partial\Omega_t = \{x \in \mathbb{R}^d \mid T(x) = t\}$, where the initial condition is $\Omega_0 = \mathcal{K}$. Notice that, given Assumption (A7), we have $\Omega_t \subset \Omega_{t+s}$ for all $t, s > 0$.

5.2.2 Minimum Time Optimal Control Problem

The above equation (5.1) also arises from the minimum time problem. A basic technique in the study of this problem (see for instance [Vla06], [Bar89], [BC08, Chapter-IV]) is the change of variable:

$$v(x) = 1 - e^{-T(x)}\ , \tag{5.2}$$

which was first used by Kruzkov [Kru75]. By doing so, $v(x)$ is automatically bounded and Lipschitz continuous. Once v is computed, we can directly get the value of $T(x)$ by $T(x) = -\log(1 - v(x))$.

Let us consider a control problem associated to the dynamical system:

$$\begin{cases} \dot{y}(t) = f(y(t),\alpha(t))\alpha(t),\ \forall t \geqslant 0\ , \\ y(0) = x\ , \end{cases} \tag{5.3}$$

where $\alpha \in \mathcal{A} := \{\alpha : \mathbb{R}_{\geqslant 0} \mapsto S_1,\ \alpha(\cdot) \text{ is measurable}\}$. Every $\alpha \in \mathcal{A}$ is then the unit vector determining the direction of motion. We denote by $y_\alpha(x;t)$ the solution of the dynamical system (5.3), and define the discounted cost functional by:

$$J(\alpha(\cdot), x) = \inf\left\{\int_0^\tau e^{-t}dt \mid \tau \geqslant 0,\ y_\alpha(x;\tau) \in \mathcal{K}\right\}\ , \tag{5.4}$$

for $\alpha \in \mathcal{A}$. Then, the value function v of the control problem given by

$$v(x) = \inf_{\alpha \in \mathcal{A}} J(\alpha(\cdot), x) \tag{5.5}$$

coincides with v in (5.2). Let now

$$F(x,r,p) = -\min_{\alpha\in\mathcal{A}}\{p\cdot f(x,\alpha)\alpha + 1 - r\}\ . \tag{5.6}$$

This Hamiltonian corresponds to the control problem (5.3,5.4,5.5). Then, under Assumption (A7), restricted to $\overline{\mathbb{R}^d\setminus\mathcal{K}}$, v is the unique viscosity solution of the following Hamilton-Jacobi-Bellman equation (see for instance [FS06]):

$$\begin{cases} F(x,v(x),Dv(x)) = 0, & x\in\mathbb{R}^d\setminus\mathcal{K},\\ v(x) = 0, & x\in\partial\mathcal{K}\ . \end{cases} \tag{5.7}$$

In the following, we will focus on the numerical approximation of system (5.7).

5.3 The Semi-Lagrangian Scheme: Convexity Properties And Convergence Analysis.

In this section, we propose a semi-lagrangian type discretization of the system (5.7). We analyze the convergence of the discretized value function to the viscosity solution of (5.7), and we give the convergence rate. An improved convergence rate is also obtained by exploiting the semiconcavity property of the discretized value function, which occurs under further regularity assumptions of the dynamics and the target set.

5.3.1 The Semi-lagrangian Scheme for the Minimum Time Problem

Consider the following semi-lagrangian type discretization of the system (5.7):

$$\begin{cases} v^h(x) = \min\limits_{\alpha\in S_1}\left\{(1-\dfrac{h}{f(x,\alpha)})v^h(x+h\alpha) + \dfrac{h}{f(x,\alpha)}\right\}, & x\in\mathbb{R}^d\setminus\mathcal{K},\\ v^h(x) = 0, & x\in\mathcal{K}\ , \end{cases} \tag{5.8}$$

where $h>0$ is a fixed parameter. This is a direct discretization in time of system (5.7), in which the time step is $h/f(x,\alpha)$, depending on state and control. The convergence of a similar discretization system, for which the time step is constant, has been studied for instance in [BF90; FGL94], and the method of proof can be straighforwardly adapted to our system (5.8), keeping in mind that (5.7) has a unique viscosity solution v.

Proposition 5.3.1. *Let us denote* $\underline{v}(x) = \liminf\limits_{h\to 0,\ y\to x} v^h(y)$, *and* $\overline{v}(x) = \limsup\limits_{h\to 0,\ y\to x} v^h(y)$. *Make Assumption (A7), then* $\overline{v}$ *(*$\underline{v}$ *resp.) is a viscosity subsolution (supersolution resp.) of* (5.7). *Thus,* $\{v^h\}$ *converge uniformly to* v *on any compact subset of* $\mathbb{R}^d$ *as* $h\to 0$. □

In the following, we denote $\overline{f}$ and $\underline{f}$ the upper and lower bounds for f, respectively, i.e.,

$$0 < \underline{f} \leqslant f(x,\alpha) \leqslant \overline{f} < \infty, \text{ for all } x\in\mathbb{R}^d \text{ and } \alpha\in\mathcal{A}\ .$$

Then, we have the following result for the convergence rate.

Proposition 5.3.2. *Suppose that Assumption (A7) holds. Then, for every* $0 < h < \frac{1}{\overline{f}}$, *there exists a constant* $C_{1/2} > 0$ *depending on* $L_f, L_v, \overline{f}$ *such that*

$$\|v^h - v\|_\infty \leqslant C_{1/2}h^{\frac{1}{2}}\ . \tag{5.9}$$

Proof. The proof for Proposition 5.3.2 is a slight modification of the original method in [CL84]. Therefore, we will only provide a brief sketch of the proof here, with the purpose of facilitating further analysis.

Let us denote by $\Omega = \mathbb{R}^d \setminus \mathcal{K}$, and define the following series of auxiliary functions. For every $1 > \varepsilon > 0$, for every $x \in \Omega$, we set $\theta_\varepsilon(x) = -|\frac{x}{\varepsilon}|^2$. For every $0 < h < \frac{1}{\overline{f}}$, for every $(x, y) \in \Omega \times \Omega$, we set $\varphi(x, y) = v^h(x) - v(y) + \theta_\varepsilon(x - y)$. As both v^h and v are bounded, for every $\zeta > 0$, there exists a point $(x_1, y_1) \in \Omega \times \Omega$ which is an approximate maximizer of φ up to a margin ζ, i.e.,

$$\varphi(x_1, y_1) > \sup_{(x,y)\in\Omega\times\Omega} (\varphi(x, y) - \zeta) \ .$$

Let us choose a function $\xi \in C_0^\infty(\Omega \times \Omega)$, such that $\xi(x_1, y_1) = 1$, and $\xi \in [0, 1]$, $|D\xi| \leqslant 1$. For every $1 > \zeta > 0$, for every $(x, y) \in \Omega \times \Omega$, let $\psi(x, y) = \varphi(x, y) + \zeta\xi(x, y)$. Let (x_0, y_0) be the point where ψ reaches its maximum, i.e,

$$\psi(x_0, y_0) \geqslant \psi(x, y), \text{ for all } (x, y) \in \Omega \times \Omega \ . \tag{5.10}$$

Then, automatically $y \to -\psi(x_0, y) = v(y) - (v^h(x_0) + \theta_\varepsilon(x_0 - y) + \zeta\xi(x_0, y))$ reaches its minimum at y_0. By definition of viscosity solution, letting $y \to (v^h_{s\to}(x_0) + \theta_\varepsilon(x_0 - y) + \zeta\xi(x_0, y))$ be a test function, we have:

$$v(y_0) - ((D\theta_\varepsilon(x_0 - y_0) \cdot \alpha^* - \zeta D_y\xi(x_0, y_0)) \cdot \alpha^*) f(y_0, \alpha^*) - 1 \geqslant 0 \ , \tag{5.11}$$

for some $\alpha^* \in S_1$. Since v^h is the solution of system (5.8), we have

$$v^h(x_0) \leqslant \left\{ (1 - \frac{1}{f(x_0, \alpha^*)}) v^h(x_0 + h\alpha^*) + \frac{h}{f(x_0, \alpha^*)} \right\} \ . \tag{5.12}$$

Take $x = x_0 + h\alpha^*, y = y_0$ in (5.10), we get

$$v^h(x_0 + h\alpha^*) \leqslant v^h(x_0) + (D\theta_\varepsilon(x_0 - y_0) \cdot \alpha^*)h + \frac{2}{\varepsilon^2}\alpha^* h^2 + \zeta\alpha^* h \ . \tag{5.13}$$

Combining (5.12) and (5.13), we get

$$v^h(x_0) \leqslant (1 - \frac{h}{f(x_0, \alpha^*)})((D\theta_\varepsilon(x_0 - y_0) \cdot \alpha^*) + \frac{2}{\varepsilon^2}\alpha^* h + \zeta\alpha^*) f(x_0, \alpha^*) + 1 \ . \tag{5.14}$$

Combining (5.11) and (5.14), we have

$$v^h(x_0) - v(y_0) \leqslant \frac{2L_f|x_0 - y_0|^2}{\varepsilon^2} + \frac{2|x_0 - y_0|h}{\varepsilon^2} + \frac{2\overline{f}h}{\varepsilon^2} + 2\zeta\overline{f} \ . \tag{5.15}$$

Let us choose $x = y = x_0$ in (5.10), then we obtain

$$|x_0 - y_0| \leqslant (L_v + \zeta)\varepsilon^2 \ , \tag{5.16}$$

where L_v is the Lipschitz constant for v. Substituting (5.16) into (5.15), we have

$$v^h(x_0) - v(y_0) \leqslant 2L_f(L_v + \zeta)^2\varepsilon^2 + 2(L_v + \zeta)h + \frac{2\overline{f}h}{\varepsilon^2} + 2\zeta\overline{f} \ . \tag{5.17}$$

Take $\varepsilon = h^{\frac{1}{4}}$, we get

$$v^h(x_0) - v(y_0) \leqslant (2L_f(L_v + \zeta)^2 + 2\overline{f})h^{\frac{1}{2}} + 2(L_v + \zeta)h + 2\zeta\overline{f} \ . \tag{5.18}$$

Let us now choose $x = y$ in (5.10), we obtain that

$$v^h(x) - v(x) \leqslant v^h(x_0) - v(y_0) + \zeta(\xi(x_0, y_0) - \xi(x, x)) - \frac{|x_0 - y_0|^2}{\varepsilon^2} \ . \tag{5.19}$$

Thus, combining (5.18) and (5.19), and take $\zeta \to 0$, we obtain that

$$v^h(x) - v(x) \leqslant (2L_f L_v^2 + 2\overline{f})h^{\frac{1}{2}} \ . \tag{5.20}$$

To show $v(x) - v^h(x) \leqslant (2L_f L_v^2 + 2\overline{f})h^{\frac{1}{2}}$, it is enough to take $\varphi(x, y) = v(x) - v^h(y) + \theta_\varepsilon(x - y)$. We conclude the estimate in Proposition 5.3.2 with $C_{1/2} = 2L_f L_v^2 + 2\overline{f}$. □

5.3.2 Discrete Time Control Problem and Its Value Function

We first represent the solution of the discretized system (5.8) as the value function of a discretized version of the control problem (5.3,5.4,5.5).

Consider the following discrete dynamical system,

$$\begin{cases} y^h(k+1) = y^h(k) + h\alpha_k, \ \forall k = 0, 1, 2, \dots \ , \\ y^h(0) = x \ , \end{cases} \tag{5.21}$$

where $\alpha_k \in S_1$, for every $k = 0, 1, 2, \dots$. Let us simply denote α^h the sequence of controls $\{\alpha_k\}_{k=0,1,2,\dots}$, and denote $y^h_{\alpha^h}(x; k)$, $k = 0, 1, 2, \dots$, the solution of the above system (5.21) with control α^h. Moreover, let

$$N(x, \alpha^h) = \inf\{N \in \mathbb{N}_+ \mid y^h_{\alpha^h}(x; N) \in \mathcal{K}\} \ . \tag{5.22}$$

Consider the following discrete cost functional:

$$J^h(\alpha^h, x) = \sum_{k=0}^{N(x,\alpha^h)} \Big(\frac{h}{f(y^h_{\alpha^h}(x; k), \alpha_k)} \big(\prod_{l=0}^{k-1} (1 - \frac{h}{f(y^h_{\alpha^h}(x; l), \alpha_l)}) \big) \Big) \ . \tag{5.23}$$

The associated value function is given by

$$v^h(x) = \inf_{\alpha^h \in \mathcal{A}^h} J^h(\alpha^h, x) \ , \tag{5.24}$$

where $\mathcal{A}^h$ is a subset of $\mathcal{A}$ containing the controls which take constant values in the interval $[k, k+1[$, for every $k = 0, 1, 2, \dots$, i.e. $\mathcal{A}^h = \{\{\alpha_k\}_{k \geqslant 0} \mid \alpha_k \in S_1, \ \forall k = 0, 1, 2, \dots \}$. Then, the value function of this discrete optimal control problem is the solution of system (5.8) (See for instance [FS06; BC08]).

Note that an equivalent formulation of the discrete cost functional in (5.23) is given by

$$J^h(\alpha^h, x) = 1 - \prod_{k=0}^{N(x,\alpha^h)} \left(1 - \frac{h}{f(y^h_{\alpha^h}(x; k), \alpha_k)} \right) \ . \tag{5.25}$$

The equality follows from an elementary computation. We will use this formulation of v^h in the following.

5.3.3 Improved Convergence Rate Under A Semiconcavity Assumption

Let us denote $d_{\mathcal{K}}(x) := \inf_{y\in\mathcal{K}} \|y - x\|$, for every $x \in \mathbb{R}^d$, the distance function from x to the target set $\mathcal{K}$. We shall make the following further assumptions on the target set $\mathcal{K}$ and speed function f.

Assumption (A8)

(i) There exists a constant $M_f > 0$ such that

$$\frac{1}{f(x+z,\alpha)} - 2\frac{1}{f(x,\alpha)} + \frac{1}{f(x-z,\alpha)} \leqslant M_f|z|^2,\ \forall x, z \in \mathbb{R}^d,\ \forall \alpha \in S_1. \tag{5.26}$$

(ii) There exists a constant $M_t > 0$ such that

$$d_{\mathcal{K}}(x+z) + d_{\mathcal{K}}(x-z) - 2d_{\mathcal{K}}(x) \leqslant M_t|z|^2,\ \forall x, z \in \mathbb{R}^d \setminus \mathring{\mathcal{K}}\ . \tag{5.27}$$

The assumption stated in (i) in Assumption (A8) is a semiconcavity property of the inverse of the speed function. In the following, we provide some criteria to check (ii) in Assumption (A8). This condition appeared in [CS95] and [CS04], it is a semiconcavity condition for the distance function $d_{\mathcal{K}}$ in $\mathbb{R}^d \setminus \mathring{\mathcal{K}}$. The authors of [CS95] provided the following sufficient condition to check (5.27):

Lemma 5.3.3 (Corollary of [CS95, Prop. 3.2])**.** *If there exists $r_t > 0$ such that*

$$\forall x \in \mathcal{K},\ \exists x_0 \in \mathcal{K} : x \in \overline{B^d(x_0, r_t)} \subset \mathcal{K}\ , \tag{5.28}$$

then (5.27) *holds. In particular, if $\partial\mathcal{K}$ is of class $\mathcal{C}^{1,1}$, then* (5.27) *holds.* □

Proposition 5.3.4 (Semiconcavity of discrete value function)**.** *Suppose that Assumption (A7) and Assumption (A8) hold. Then, we have that*

$$v^h(x+z) - 2v^h(x) + v^h(x-z) \leqslant C_v|z|^2,\ \textit{for every } x, z \in \mathbb{R}^d \setminus \mathcal{K}\ , \tag{5.29}$$

where C_v is a constant depends on $M_f, M_t, \overline{f}, \underline{f}$.

Proof. Let us denote $\alpha^*_x = \{\alpha^*_{x,0}, \alpha^*_{x,1}, \alpha^*_{x,2}, \ldots, \alpha^*_{x,N_x}\}$ the discrete optimal control for which the infimum in (5.24) is obtained, and let us simply denote $N_x = N(x, \alpha^*_x)$. For the problem starting from $(x+z)$ ($x-z$ resp.), let us consider a control α'_{x+z} (α'_{x-z} resp.) as following: α'_{x+z} (α'_{x-z} resp.) takes the same control as α^*_x until one of the three trajectories $y^h_{\alpha^*}(x;\cdot)$, $y^h_{\alpha'_{x+z}}(x+z;\cdot)$ and $y^h_{\alpha'_{x-z}}(x-z;\cdot)$ reaches $\mathcal{K}$. Then, we will show (5.29) by discussing two cases.

Case 1. $\forall N \leqslant N_x,\ y^h_{\alpha'_{x+z}}(x+z;N) \notin \mathcal{K}$ and $y^h_{\alpha'_{x-z}}(x-z;N) \notin \mathcal{K}$. In this case, the optimal trajectory for the problem starting from x will first reach $\mathcal{K}$. Then, for the problem starting from $x+z$ ($x-z$ resp.), we take the control following the shortest distance path, in euclidean sense, from $y^h_{\alpha'_{x+z}}(x+z;N_x)$ ($y^h_{\alpha'_{x-z}}(x-z;N_x)$ resp.) to $\mathcal{K}$. Let N_+, N_- denote the steps for which $y^h_{\alpha'_{x+z}}(x+z;N_+), y^h_{\alpha'_{x-z}}(x-z;N_-) \in \mathcal{K}$. For easy expression, we simply denote $y^h_{x,m} := y^h_{\alpha^*_x}(x;m)$, $y^h_{+,m} := y^h_{\alpha'_{x+z}}(x+z;m)$, $y^h_{-,m} := y^h_{\alpha'_{x-z}}(x-z;m)$ for $m = 0, 1, 2, \ldots$ and $\alpha_+ = \alpha'_{x+z}$, $\alpha_- = \alpha'_{x-z}$.

Following (5.25), we have:

$$
\begin{aligned}
&v^h(x+z) - 2v^h(x) + v^h(x-z) \\
&\leqslant \{J^h(\alpha_+, x+z) + J^h(\alpha_-, x-z) - 2J^h(\alpha_x^*, x)\} \\
&\leqslant \left\{ \Big(1 - \prod_{k=0}^{N_+}(1 - \frac{h}{f(y^h_{+,k}, \alpha_+)})\Big) + \Big(1 - \prod_{k=0}^{N_-}(1 - \frac{h}{f(y^h_{-,k}, \alpha_-)})\Big) - 2\Big(1 - \prod_{k=0}^{N_x}(1 - \frac{h}{f(y^h_{x,k}, \alpha_x^*)})\Big) \right\} \\
&\leqslant \prod_{k=0}^{N_x}(1 - \frac{h}{f(y^h_{x,k}, \alpha_x^*)}) \left\{ \Big(1 - \frac{\prod_{k=0}^{N_+}(1 - \frac{h}{f(y^h_{+,k}, \alpha_+)})}{\prod_{k=0}^{N_x}(1 - \frac{h}{f(y^h_{x,k}, \alpha_x^*)})}\Big) + \Big(1 - \frac{\prod_{k=0}^{N_-}(1 - \frac{h}{f(y^h_{-,k}, \alpha_-)})}{\prod_{k=0}^{N_x}(1 - \frac{h}{f(y^h_{x,k}, \alpha_x^*)})}\Big) \right\} \\
&\leqslant \left[(1 - \frac{h}{\overline{f}})^{N_x} \left\{ \sum_{k=0}^{N_x} \left(\frac{h}{f(y^h_{+,k}, \alpha_x^*)} + \frac{h}{f(y^h_{-,k}, \alpha_x^*)} - 2\frac{h}{f(y^h_{x,k}, \alpha_x^*)} \right) \right\} \right] \\
&\quad + \left[(1 - \frac{h}{\overline{f}})^{N_x} \left\{ \sum_{k=N_x}^{N_{x_+}} \frac{h}{f(y^h_{+,k}, \alpha_+)} + \sum_{k=N_x}^{N_{x_-}} \frac{h}{f(y^h_{-,k}, \alpha_-)} \right\} \right] .
\end{aligned}
\tag{5.30}
$$

We use the fact that in first N_x steps, three dynamics take the same control α_x^*. Let us first focus on the first part inside of [ ], denoted by Δ_1. Let $A_k := \frac{1}{f(y^h_{+,k}, \alpha_x^*)} + \frac{1}{f(y^h_{-,k}, \alpha_x^*)} - 2\frac{1}{f(y^h_{x,k}, \alpha_x^*)}$, then:

$$
\begin{aligned}
A_k &= \frac{1}{f(y^h_{x,k} + (y^h_{+,k} - y^h_{x,k}), \alpha_x^*)} - 2\frac{1}{f(y^h_{x,k}, \alpha_x^*)} + \frac{1}{f(y^h_{x,k} - (y^h_{+,k} - y^h_{x,k}), \alpha_x^*)} \\
&\quad + \frac{1}{f(y^h_{-,k}, \alpha_x^*)} - \frac{1}{f(y^h_{x,k} - (y^h_{+,k} - y^h_{x,k}), \alpha_x^*)} \\
&\leqslant M_f |y^h_{+,k} - y^h_{x,k}|^2 + \frac{L_f}{\underline{f}^2} |y^h_{+,k} - 2y^h_{x,k} + y^h_{-,k}| .
\end{aligned}
\tag{5.31}
$$

By the above construction of α_+ and α_-, we notice that $|y^h_{+,k} - y^h_{x,k}| = z$ and $|y^h_{+,k} - 2y^h_{x,k} + y^h_{-,k}| = 0$, for all $k \in \{0, 1, 2, \ldots, N_x\}$. Thus $A_k \leqslant M_f z^2$. Then we have:

$$
\Delta_1 \leqslant M_f |z|^2 \sum_{k=0}^{N_x} (1 - \frac{h}{\overline{f}})^{N_x} h \leqslant M_f \overline{f} |z|^2 . \tag{5.32}
$$

For the second part inside of [ ] in (5.30), denoted by Δ_2, we notice that at the end of N_x step, $y^h_{x,N_x} \in \mathcal{K}$, $y^h_{+,N_x} = y^h_{x,N_x} + z$, $y^h_{-,N_x} = y^h_{x,N_x} - z$. Then, by (5.27), we have:

$$
d_{\mathcal{K}}(y^h_{+,N_x}) + d_{\mathcal{K}}(y^h_{-,N_x}) \leqslant M_t |z|^2 . \tag{5.33}
$$

Thus, by the above construction of α_+ and α_-, we have:

$$
\Delta_2 \leqslant \frac{M_t}{\underline{f}} |z|^2 . \tag{5.34}
$$

Combine (5.32) and (5.34), we deduce (5.29) with $C_1 = M_f \overline{f} + \frac{M_t}{\underline{f}}$.

Case 2. $\exists N_- \leqslant N_x$ such that $y^h_{\alpha'_{x-z}}(x - z; N_-) \in \mathcal{K}$. In this case, the optimal trajectory for the discrete problem starting from $x - z$ will first reach $\mathcal{K}$. Then, for the problem starting from $x + z$, let us consider a control α'_{x+z} as follows:

$$
\alpha'_{x+z,k} = \begin{cases}
\alpha^*_{x,k} , & k \in \{0, 1, 2, \ldots, N_-\} ; \\
\alpha^*_{x,(N_- + \lfloor \frac{k - N_-}{2} \rfloor)} , & k \in \{N_-, (N_- + 1), \ldots, (2N_x - N_-)\} ; \\
\text{following Euclidean shortest path,} & (2N_x - N_-) < k \leqslant N_+ ,
\end{cases}
\tag{5.35}
$$

with $y^h_{\alpha'_{x+z}}(x+z;N_+) \in \mathcal{K}$. We first assume $N_+ \geqslant (2N_x - N_-)$, then the result will automatically holds as it is in a weaker situation when $N_+ < (2N_x - N_-)$. We also take the same simplified notations as in Case 1., and we omit the same computations. Then, following (5.25), we have:

$$\begin{aligned}
&v^h(x+z) - 2v^h(x) + v^h(x-z) \\
&\leqslant \{J^h(\alpha_+, x+z) + J^h(\alpha_-, x-z) - 2J^h(\alpha^*_x, x)\} \\
&\leqslant \left[(1-\frac{h}{\overline{f}})^{N_x}\left\{\sum_{k=0}^{N_-}\left(\frac{h}{f(y^h_{+,k},\alpha^*_x)} + \frac{h}{f(y^h_{-,k},\alpha^*_x)} - 2\frac{h}{f(y^h_{x,k},\alpha^*_x)}\right)\right\}\right] \\
&\quad + \left[\sum_{k=(N_-+1)}^{(2N_x-N_-)}\Big(\frac{h}{f(y^h_{+,k},\alpha_+)}\Big) - 2\sum_{k=(N_-+1)}^{N_x}\Big(\frac{h}{f(y^h_{x,k},\alpha^*_x)}\Big)\right] + \left[\sum_{2N_x-N_-+1}^{N_+}\Big(\frac{h}{f(y^h_{+,k},\alpha_+)}\Big)\right] .
\end{aligned} \tag{5.36}$$

The first part inside of [ ] follows the same computation as in Case 1. Let us now focus on the second part inside of [ ], denoted by Δ'_2. Based on the above construction of α_-, we have $y^h_{-,N_-} = y^h_{x,N_-} - z$. Since $y^h_{-,N_-} \in \mathcal{K}$, we have $d_{\mathcal{K}}(y^h_{x,N_-}) \leqslant z$. Since α^*_x is the optimal control for the problem starting from x, then we have:

$$\sum_{k=(N_-+1)}^{N_x} \frac{|y^h_{x,k} - y^h_{x,k-1}|}{f(y^h_{x,k-1},\alpha^*_x)} \leqslant \frac{d_{\mathcal{K}}(y^h_{x,N_-})}{\underline{f}} \leqslant \frac{|z|}{\underline{f}} , \tag{5.37}$$

which implies

$$\sum_{k=(N_-+1)}^{N_x} |y^h_{x,k} - y^h_{x,k-1}| \leqslant \frac{\overline{f}}{\underline{f}}|z| . \tag{5.38}$$

Based on the above construction of α_+, we have $y^h_{+,N_-} = y^h_{x,N_-} + z$. Moreover, for every $j \in \{1,2,\dots,(N_x - N_-)\}$, we have:

$$\max\{|y^h_{+,(N_-+2j-1)} - y^h_{x,(N_-+j)}|,\ |y^h_{+,(N_-+2j)} - y^h_{x,(N_-+j)}|\} \leqslant (\frac{\overline{f}}{\underline{f}} + 1)|z| . \tag{5.39}$$

Then, by the Lipschitz continuity of f, and the fact that $\alpha_{+,(N_-+2j-1)} = \alpha_{+,(N_-+2j)} = \alpha^*_{x,N_-+j}$, we have:

$$\begin{aligned}
\Delta'_2 &\leqslant \frac{h}{\underline{f}^2}\sum_{j=1}^{N_x-N_-}\Big(|f(y^h_{+,(N_-+2j-1)},\alpha_+) - f(y^h_{x,(N_-+j)},\alpha^*_x)| + |f(y^h_{+,(N_-+2j)},\alpha_+) - f(y^h_{x,(N_-+j)},\alpha^*_x)|\Big) \\
&\leqslant \frac{2L_f\overline{f}}{\underline{f}^3}(\frac{\overline{f}}{\underline{f}} + 1)|z|^2 .
\end{aligned} \tag{5.40}$$

For the third part inside of [ ], denoted by Δ'_3, we first notice that

$$y^h_{+,(2N_x-N_-)} - y^h_{x,N_x} = y^h_{x,N_x} - y^h_{-,N_-} \leqslant (\frac{\overline{f}}{\underline{f}} + 1)|z| . \tag{5.41}$$

Then, by (5.27), and the fact that $y^h_{x,N_x} \in \mathcal{K}$, $y^h_{-,N_-} \in \mathcal{K}$, we have:

$$d_{\mathcal{K}}(y^h_{+,(2N_x-N_-)}) \leqslant M_t(\frac{\overline{f}}{\underline{f}} + 1)^2|z|^2 . \tag{5.42}$$

Then,

$$\Delta'_3 \leqslant \frac{M_t}{\underline{f}}(\frac{\overline{f}}{\underline{f}} + 1)^2|z|^2 . \tag{5.43}$$

Combine (5.40) and (5.43), we deduce (5.29) with $C_2 = M_f\overline{f} + \frac{2L_f\overline{f}}{\underline{f}^3}(\frac{\overline{f}}{\underline{f}}+1) + \frac{M_t}{\underline{f}}(\frac{\overline{f}}{\underline{f}}+1)$.

Since another possible case, that is $\exists N_+ \leqslant N_x$ such that $y^h_{\alpha'_{x+z}}(x+z;N_+) \in \mathcal{K}$, is symmetric as Case 2., we conclude (5.29) with $C = \max\{C_1, C_2\}$. □

The semiconcavity property of the discrete value function leads to an improved convergence rate, which we state as the main result of this section below.

Theorem 5.3.5. *Suppose that Assumption (A7) and Assumption (A8) hold. Then, for every $0 < h < \frac{1}{\overline{f}}$, there exists a constant C_1 depends on $M_f, M_t, L_v, L_f, \overline{f}, \underline{f}$ such that*

$$\|v^h - v\|_\infty \leqslant C_1 h \ . \tag{5.44}$$

Proof. Let us first show that $v - v^h \leqslant Ch$. Since $\mathcal{A}^h \subseteq \mathcal{A}$, we always have

$$\begin{aligned} v(x) - v^h(x) &= \inf_{\alpha\in\mathcal{A}} J(x,\alpha) - \inf_{\alpha^h\in\mathcal{A}^h} J^h(x,\alpha^h) \\ &\leqslant \inf_{\alpha^h\in\mathcal{A}^h} J(x,\alpha^h) - \inf_{\alpha^h\in\mathcal{A}^h} J^h(x,\alpha^h) \\ &\leqslant \sup_{\alpha^h\in\mathcal{A}^h} (J(x,\alpha^h) - J^h(x,\alpha^h)) \ . \end{aligned}$$

For a given α^h, let us denote $\tau(x,\alpha^h)$ the travel time of the continuous control problem starting from x, i.e, $y_{\alpha^h}(x,\tau(x,\alpha^h)) \in \mathcal{K}$, then we have:

$$v(x) - v^h(x) \leqslant (J(x,\alpha) - J^h(x,\alpha)) \leqslant |e^{-\tau(x,\alpha^h)} - e^{-\sum_{k=0}^{N_x}\frac{h}{f(x,\alpha_k)}}| \leqslant \frac{L_f}{2\underline{f}^2}h \ . \tag{5.45}$$

To show $v^h - v \leqslant Ch$, we use the same definition of auxiliary functions as the in proof of Proposition 5.3.2. Then, similarly as in the proof of Proposition 5.3.2, let $y \to (v^h(x_0) + \theta_\varepsilon(x_0 - y) + \zeta\xi(x_0,y))$ be the test function, we have:

$$v(y_0) - ((D\theta_\varepsilon(x_0-y_0)\cdot\alpha^* - \zeta D_y\xi(x_0,y_0))\cdot\alpha^*)f(y_0,\alpha^*) - 1 \geqslant 0 \ , \tag{5.46}$$

for some $\alpha^* \in S_1$. Moreover, let us consider a function

$$\vartheta(x) = v^h(x_0+x) - v^h(x_0) + (D\theta_\varepsilon(x_0-y_0) + \zeta D_x\xi(x_0,y_0))\cdot x \ .$$

Then we have $\vartheta(0) = 0$, and

$$\vartheta(x+z) - 2\vartheta(x) + \vartheta(x-z) = v^h(x_0+x+z) - 2v^h(x_0+x) + v^h(x_0+x-z), \leqslant C_v|z|^2,$$

by Proposition 5.3.4. Moreover, by the definition of ψ, we have

$$\begin{aligned} \vartheta(x) = \psi(x_0+x,y_0) - \psi(x_0,y_0) + \theta_\varepsilon(x_0-y_0) - \theta_\varepsilon(x_0+x-y_0) \\ + D\theta_\varepsilon(x_0-y_0)x + \zeta(\xi(x_0,y_0) - \xi(x_0+x,y_0) + D_x\xi(x_0,y_0)\cdot x) \ . \end{aligned}$$

Since ψ get it's maximum at (x_0,y_0), we then have $\limsup_{|x|\to0}\frac{\vartheta(x)}{|x|} = \limsup_{|x|\to0}(\psi(x_0+x,y_0) - \psi(x_0,y_0) \leqslant 0$, which implies $\vartheta(x) \leqslant \frac{C_v}{2}|x|^2$.

Let us now take $x = h\alpha^*$, then

$$v^h(x_0+h\alpha^*) \leqslant v^h(x_0) + (D\theta_\varepsilon(x_0-y_0) + \zeta D_x\xi(x_0,y_0))\cdot h\alpha^* + \frac{C_v}{2}h^2 \ . \tag{5.47}$$

Since v^h is the solution of system (5.8), we have:

$$v^h(x_0) \leqslant \left\{(1 - \frac{h}{f(x_0,\alpha^*)})v^h(x_0 + h\alpha^*) + \frac{h}{f(x_0,\alpha^*)}\right\} \ . \tag{5.48}$$

Combining (5.47) and (5.48), we have

$$v^h(x_0) \leqslant (1-\frac{h}{f(x_0,\alpha^*)})(v^h(x_0)+(D\theta_\varepsilon(x_0-y_0)+\zeta D_x\xi(x_0,y_0))\cdot h\alpha^*+\frac{C_v}{2}h^2)+\frac{h}{f(x_0,\alpha^*)} \ , \tag{5.49}$$

which implies

$$v^h(x_0) \leqslant (1 - \frac{h}{f(x_0,\alpha^*)})(D\theta_\varepsilon(x_0 - y_0) + \zeta D_x\xi(x_0,y_0)) \cdot \alpha^* + \frac{C_v}{2}h)f(x_0,\alpha^*) + 1 \ . \tag{5.50}$$

Combining (5.46) and (5.50), we have:

$$\begin{aligned} v^h(x_0) - v(y_0) &\leqslant (1 - \frac{h}{f(x_0,\alpha^*)})(D\theta_\varepsilon(x_0 - y_0) + \zeta D_x\xi(x_0,y_0)) \cdot \alpha^* + \frac{C_v}{2}h)f(x_0,\alpha^*) \\ &\quad - ((D\theta_\varepsilon(x_0 - y_0) \cdot \alpha^* - \zeta D_y\xi(x_0,y_0)) \cdot \alpha^*)f(y_0,\alpha^*) \\ &\leqslant (f(x_0,\alpha^*) - f(y_0,\alpha^*))\frac{2|x_0 - y_0|}{\varepsilon^2} + \frac{2|x_0 - y_0|h}{\varepsilon^2} + \frac{C_v}{2}\overline{f}h + 2\zeta\overline{f} \\ &\leqslant \frac{2L_f|x_0 - y_0|^2}{\varepsilon^2} + \frac{2|x_0 - y_0|h}{\varepsilon^2} + \frac{C_v}{2}\overline{f}h + 2\zeta\overline{f} \ . \end{aligned} \tag{5.51}$$

Take $x = y = x_0$ for $\varphi(x, y)$, we obtain $|x_0 - y_0| \leqslant (L_v + \zeta)\varepsilon^2$. Thus, for (5.51) we have

$$v^h(x_0) - v(y_0) \leqslant (2L_f(L_v + \zeta)^2\varepsilon^2 + 2(L_v + \zeta)h + \frac{C_v}{2}\overline{f}h) + 2\overline{f}\zeta \ . \tag{5.52}$$

Take now $\varepsilon = h^{\frac{1}{2}}$ in (5.52), we then have

$$v^h(x_0) - v(y_0) \leqslant (2L_f(L_v + \zeta)^2 + 2(L_v + \zeta) + \frac{C_v}{2}\overline{f})h + 2\overline{f}\zeta \ . \tag{5.53}$$

Take $x = y$ for $\psi(x, y)$ and use the fact that $\psi(x_0, y_0) \geqslant \psi(x, x)$, we have

$$v^h(x) - v(x) \leqslant (v^h(x_0) - v(y_0)) + \zeta(\xi(x_0,y_0) - \xi(x,x)) - \frac{|x_0 - y_0|^2}{\varepsilon^2} \ . \tag{5.54}$$

Thus, combining (5.53) and (5.54), we have

$$v^h(x) - v(x) \leqslant (2L_f(L_v + \zeta)^2 + 2(L_v + \zeta) + \frac{C_v}{2}\overline{f})h + (2\overline{f} - 1)\zeta \ . \tag{5.55}$$

Then, taking $\zeta \to 0$ in (5.55), we have:

$$v^h(x) - v(x) \leqslant (2L_fL_v^2 + 2L_v + \frac{C_v}{2}\overline{f})h \ . \tag{5.56}$$

Combining (5.45) and (5.56), we conclude that (5.44) holds with $C_1 = \max\{\frac{L_f}{2\underline{f}^2}, 2L_fL_v^2 + 2L_v + \frac{C_v}{2}\overline{f}\}$. □

5.4 Convergence of A Fully Discretized Scheme, Application to Convergence Rate Analysis of Fast-Marching Method

In this section, we first present a fully discretized semi-lagrangian scheme of the system (5.7). We demonstrate the convergence rate of the scheme using particular interpolation operators. We then apply this result to show the convergence rate of a fast-marching method, for which the update operator is obtained by the presented scheme.

We should note that in previous works of the fast-marching method, in [SV03; CF07; Car+08; Mir14], the authors prove the convergence of their methods when the mesh step goes to 0 without an explicit convergence rate, wheares in [SMK16; Mir19] the authors establish a convergence rate of order $\frac{1}{2}$. Though, most of numerical experiments in the aforementioned works demonstrate an actual convergence rate of order 1.

5.4.1 A Fully Discretized Scheme and A First Convergence Analysis

To get the numerical approximation of (5.7), we also need to discretize the space. Assume now given a grid X^h discretizing Ω with mesh step h, that is the minimum distance between two distinct points is h. Let us denote w^h the approximate value function for v obtained by applying the semi-lagrangian scheme (5.8) to all grid nodes $x \in X^h$, while when the points $x + h\alpha$ are not in the grid X^h, we compute the value of $w^h(x + h\alpha)$ by an interpolation of the value of it's neighborhood nodes. We assume given an interpolation operator to be used in (5.8) when $x \in X^h$. This interpolation may depend on x (this is the index of the equation), and will be denoted by $I^x[\cdot]$. However the value $I^x[w^h](x')$ depends only on the values $w^h(y)$ with $y \in X^h$ in a neighborhood of x'. We then consider the following fully discretization semi-lagrangian scheme, define $z^h : X^h \to \mathbb{R}$ by

$$\begin{cases} z^h(x) = \min\limits_{\alpha \in S_1} \left\{ (1 - \dfrac{h}{f(x,\alpha)}) I^x[z^h](x + h\alpha) + \dfrac{h}{f(x,\alpha)} \right\}, & x \in X^h \setminus \mathcal{K} \ , \\ z^h(x) = 0, & x \in X^h \cap \mathcal{K} \ . \end{cases} \tag{5.57}$$

We begin by considering a regular triangular mesh, and a simple interpolation operator, denoted by I_1, which is the P_1 (piecewise linear) interpolation operator on the simplices of the triangular grid. More precisely, for a fixed $x \in \mathbb{R}^d$, let $Y^h(x) = \{y_k\}_{k=1\ldots d+1}$ denote the set of vertices of the simplex that contains x. We then define

$$I_1[z^h](x) = \sum_{y_k \in Y^h(x)} \lambda(x; y_k) z^h(y_k) \ , \tag{5.58a}$$

where the coefficients $\lambda(x; y_k)$ depend on x and y_k, and are uniquely determined by the following condition

$$\begin{cases} 0 \leqslant \lambda(x; y_k) \leqslant 1, \text{ for every } y_k \ , \\ \sum\limits_{y_k \in Y^h(x)} \lambda(x; y_k) = 1 \text{ and } \sum\limits_{y_k \in Y^h(x)} \lambda(x; y_k) y_k = x \ . \end{cases} \tag{5.58b}$$

Let S_1^h denote the space of continuous and piecewise linear functions on the triangulation X^h. We then have $w^h = I_1[z^h] \in S_1^h$. Denote $T^h : \mathbb{R}^{\mathbb{R}^d} \to \mathbb{R}^{\mathbb{R}^d}$ the operator defined as follows, for every $x \in \mathbb{R}^d$,

$$\begin{cases} T^h[v^h](x) := \min\limits_{\alpha \in S_1} \left\{ (1 - \dfrac{h}{f(x,\alpha)}) v^h(x + h\alpha) + \dfrac{h}{f(x,\alpha)} \right\}, & x \in \mathbb{R}^d \setminus \mathcal{K} \ , \\ T^h[v^h](x) := 0, & x \in \mathcal{K} \ . \end{cases} \tag{5.59}$$

The solution v^h of the semi-lagrangian scheme (5.8) is a fixed point of T^h. Denote $R_1 : \mathbb{R}^{\mathbb{R}^d} \to \mathbb{R}^{X^h}$, then $w^h = I_1[z^h]$ is indeed a fixed point of $(I_1 \circ R_1 \circ T^h)$. Moreover, the following property holds:

Lemma 5.4.1. *For every $w^1, w^2 \in S_1^h$,*

$$\|(I_1 \circ R_1 \circ T^h)[w^1] - (I_1 \circ R_1 \circ T^h)[w^2]\|_\infty \leqslant (1 - \frac{h}{\overline{f}})\|w^1 - w^2\|_\infty \ . \tag{5.60}$$

Moreover, denote $t[w] = \sup_x w(x)$, we have

$$t[(I_1 \circ R_1 \circ T^h)[w^1] - (I_1 \circ R_1 \circ T^h)[w^2]] \leqslant (1 - \frac{h}{\overline{f}})t[w_1 - w^2] \ . \tag{5.61}$$

Sketch of Proof. For (5.60), by (5.59), we first notice that T^h is a contraction mapping with contracting rate $(1-\frac{h}{\overline{f}})$. Moreover, $I_1 \circ R_1$ is monotone and additively homogeneous (it commutes with the addition of a constant function), so it is nonexpansive, see [CT80]. Thus, $I_1 \circ R_1 \circ T^h$ is a contraction mapping with contracting rate $(1 - \frac{h}{\overline{f}})$. The similar analysis also applies to (5.61). □

This result is extended to ϵ-monotone interpolation operators in [Bok+15].

In the following, we intend to bound the sup-norm between w^h and v. We begin by bounding $w^h - v$ in one direction.

Proposition 5.4.2. *Assume Assumption (A7) and Assumption (A8), taking I^x be I_1, for every $0 < h < \frac{1}{\overline{f}}$, there exists a constant depending on $L_f, L_v, \overline{f}$ and C_v in Proposition 5.3.4 such that*

$$\sup_{x \in \mathbb{R}^d} (w^h - v)(x) \leqslant C_{w_1} h \ . \tag{5.62}$$

Proof. We first notice that the semiconcavity of v^h in Proposition 5.3.4 implies

$$\sup_{x \in \mathbb{R}^d} ((I_1 \circ R_1)[v^h] - v^h)(x) \leqslant \frac{C_v}{2} h^2 \ . \tag{5.63}$$

Moreover, let w^h and v^h be the fixed points of $(I_1 \circ R_1 \circ T^h)$ and T^h, respectively. Using Lemma 5.4.1, we have

$$\begin{aligned}
\sup_{x \in \mathbb{R}^d} (w^h - v^h)(x) &= \sup_{x \in \mathbb{R}^d} (w^h - (I_1 \circ R_1)[v^h] + (I_1 \circ R_1)[v^h] - v^h)(x) \\
&= \sup_{x \in \mathbb{R}^d} ((I_1 \circ R_1 \circ T^h)[w^h] - (I_1 \circ R_1 \circ T^h)[v^h] + (I_1 \circ R_1)[v^h] - v^h)(x) \\
&\leqslant (1 - \frac{h}{\overline{f}}) \sup_{x \in \mathbb{R}^d} (w^h - v^h)(x) + \sup_{x \in \mathbb{R}^d} ((I_1 \circ R_1)[v^h] - v^h)(x) \ .
\end{aligned} \tag{5.64}$$

Combining (5.63) and (5.64), we have

$$\sup_{x \in \mathbb{R}^d} (w^h - v^h)(x) \leqslant \frac{C_v \overline{f}}{2} h \ . \tag{5.65}$$

Moreover, based on the proof of Theorem 5.3.5, we have

$$\sup_{x \in \mathbb{R}^d} (v^h - v)(x) \leqslant (2L_f L_v^2 + 2L_v + \frac{C_v}{2}\overline{f})h \ . \tag{5.66}$$

Combining (5.65) and (5.66), we conclude the result in (5.62) with $C_{w_1} = (2L_f L_v^2 + 2L_v + C_v\overline{f})$. □

5.4.2 Controlled Markov Problem and Its Value Function

In order to show the error bound in the other direction, we will first reformulate the fully discretized system (5.57) as a dynamic programming equation of a stochastic optimal control problem. We notice that, in the formulation (5.58b), the coefficients $\{\lambda_k(x;y_k)\}$ can be interpreted as the transition probabilities of a controlled Markov chain, for which the state space is the set of nodes in X^h. More precisely, we first rewrite the system (5.57) as follows:

$$\begin{cases} z^h(x_i) = \min\limits_{\alpha\in S_1}\left\{(1-\dfrac{h}{f(x_i,\alpha)})\displaystyle\sum_{y_k\in Y^h(x_i+\alpha h)}\lambda(x_i+\alpha h;y_k)z^h(y_k)+\dfrac{h}{f(x_i,\alpha)}\right\}, & x_i\in X^h\setminus\mathcal{K}\ , \\ z^h(x_i)=0, & x_i\in X^h\cap\mathcal{K}\ . \end{cases} \tag{5.67}$$

Let us consider a Markov decision process on the state space X^h, with controls in S_1 and transition probability given by

$$\mathbb{P}(\xi_{k+1}=y\mid\xi_k=x_i,\alpha_k=\alpha)=\begin{cases}\lambda(x_i+\alpha h;y), & \text{if } y\in Y^h(x_i+\alpha h)\ ,\\ 0, & \text{otherwise}\ .\end{cases} \tag{5.68}$$

Here, ξ_k denotes the state at time step k and α_k denotes the control at time step k. Given a pure strategy, that is a map σ^h which to any history $H_k=(\xi_0,\alpha_0,\ldots,\xi_{k-1},\alpha_{k-1},\xi_k)$ associates a control α_k, we can define a probability space $(\Omega,\mathcal{F},\mathbb{P})$ and processes $(\xi_k)_{k\geqslant 0}$ of states and $(\alpha_k)_{k\geqslant 0}$ of controls satisfying (5.68), and $\alpha_k=\sigma^h(H_k)$. Let us denote $N^h(\sigma^h)=\min\{n\in\mathbb{N}_+\mid\xi_n\in\mathcal{K}\}$, that is a stopping time adapted to the Markov decision process with strategy σ^h. By this formulation, we have the following property for the process $(\xi_k)_{k\geqslant 0}$:

$$\mathrm{E}[\xi_{k+1}-\xi_k\mid\xi_k=x,\alpha_k=\alpha]=h\alpha,\ \forall x\in X^h,\alpha\in S_1 \text{ and } k<N^h(\sigma^h) \tag{5.69a}$$

and

$$\mathrm{Tr}(\mathrm{Var}[\xi_{k+1}-\xi_k\mid\xi_k=x,\alpha_k=\alpha])\leqslant h^2,\ \forall x\in X^h,\alpha\in S_1 \text{ and } k<N^h(\sigma^h)\ . \tag{5.69b}$$

Let $E_x^{\sigma^h}$ denote the expectation given a initial condition x and a strategy σ^h. Consider the following cost functional:

$$W(\sigma^h,x)=\mathrm{E}_x^{\sigma^h}\left[1-\prod_{k=0}^{N^h(\sigma^h)}\left(1-\frac{h}{f(\xi_k,\alpha_k)}\right)\right]\ . \tag{5.70}$$

Then the solution of (5.67) is indeed the value function of the above controlled Markov problem (see for instance [KD01]), that is,

$$z^h(x)=\inf\ W(\sigma^h,x)\ . \tag{5.71}$$

5.4.3 Convergence Rate Analysis Under A Semiconvexity Assumption

To get the convergence rate for the fully discretized scheme, we shall make the following further assumptions for the target set $\mathcal{K}$ and the speed function f.

Assumption (A9)

(i) There exists a constant $-M_f'>0$ such that

$$\frac{1}{f(x+z,\alpha)}-2\frac{1}{f(x,\alpha)}+\frac{1}{f(x-z,\alpha)}\geqslant M_f'|z|^2,\ \forall x,z\in\mathbb{R}^d,\ \forall\alpha\in S_1. \tag{5.72}$$

(ii) There exists a constant $-M_t' > 0$ such that

$$d_{\mathcal{K}}(x+z) + d_{\mathcal{K}}(x-z) - 2d_{\mathcal{K}}(x) \geqslant M_t'|z|^2, \ \forall x, z \in \mathbb{R}^d \setminus \mathring{\mathcal{K}} \ . \tag{5.73}$$

The assumptions stated in (i) and (ii) in Assumption (A9) can be thought of the semiconvexity properties of the speed function and of the distance function $d_{\mathcal{K}}$ in $\mathbb{R}^d \setminus \mathring{\mathcal{K}}$, respectively. In particular, if f is of class $\mathcal{C}^2$ and $\partial\mathcal{K}$ is of class $\mathcal{C}^2$, one can check that both Assumption (A8) and Assumption (A9) hold.

We first state the following technical lemma, which is needed to prove our main result

Lemma 5.4.3. *Assume $g : \mathbb{R}^d \to \mathbb{R}$ is $\tilde{\alpha}$-semiconvex and let X be a $d-$dimensional random variable, then we have*

$$g(E[X]) - E[g(X)] \leqslant \tilde{\alpha}\, Tr(\mathrm{Var}[X]) \ . \tag{5.74}$$

Proof. Since $g(x)$ is $\tilde{\alpha}-$semiconvex, we have $g(x) + \tilde{\alpha}\|x\|^2$ is convex, then

$$\mathrm{E}[g(X) + \alpha\|X\|^2] \geqslant g(\mathrm{E}[X]) + \alpha\|\mathrm{E}[X]\|^2 \ . \tag{5.75}$$

(5.74) is then deduced using $E\|X\|^2 - \|E[X]\|^2 = \mathrm{Tr}(\mathrm{Var}[X])$. □

Proposition 5.4.4. *Suppose Assumption (A7) and Assumption (A9) hold, taking I^x be I_1, there exists a constant C_{w_2} depending on $L_f, \underline{f}, \overline{f}, M_f', M_t'$ such that, for every $0 < h < \frac{1}{\overline{f}}$,*

$$\sup_{x\in\mathbb{R}^d} (v - w^h)(x) \leqslant C_{w_2} h \ . \tag{5.76}$$

Proof. Let us denote σ^h a strategy for the stochastic control problem. Let $(\Omega, \mathcal{F}, \mathbb{P})$, (ξ_k), (α_k) and $N^h(\sigma^h)$ be defined as above. Now, when $\omega \in \Omega$ is fixed together with $(\alpha_k(\omega))_{0\leqslant k\leqslant N^h(\sigma^h)}$ the associated control, consider a deterministic trajectory $\{\bar{\xi}_k\}_{k=1,2,\dots}$ such that:

$$\bar{\xi}_0 = x, \quad \bar{\xi}_{k+1} = \bar{\xi}_k + h\alpha_k(\omega) \ , \tag{5.77}$$

and if $\bar{\xi}_n \notin \mathcal{K}$ for all $n \leqslant N^h(\sigma^h)$, we then take the controls following the straight line from $\bar{\xi}_{N^h(\sigma^h)}$ to $\mathcal{K}$. Let us denote $\bar{N}^h(\omega, \sigma^h) = \min\{n \in \mathbb{N}_+ \mid \bar{\xi}_n \in \mathcal{K}\}$. By this construction, $\{\bar{\xi}_k\}_{0\leqslant k\leqslant \bar{N}^h(w,\sigma^h)}$ is indeed a solution of the discrete system (5.21), i.e., $\bar{\xi}_k$ satisfies $\bar{\xi}_k = y^h_{\alpha^h}(x,k)$ for every $k \in \{0, 1\dots, \bar{N}^h(\omega,\sigma^h)\}$. Thus, we have

$$v^h(x) \leqslant J^h(\alpha^h, x) \ . \tag{5.78}$$

Since this holds for almost all $\omega \in \Omega$, we have

$$v^h(x) \leqslant \mathrm{E}_x^{\sigma^h}[J^h(\alpha^h, x)] \ . \tag{5.79}$$

Let us simply denote $N^h(\sigma^h)$ by N and $\bar{N}^h(\omega,\sigma^h)$ by $\bar{N}$ in the following. We have

$$\begin{aligned}
v^h(x) - w^h(x) &\leqslant \mathrm{E}_x^{\sigma^h}\left[\Big(1 - \prod_{k=0}^{\bar{N}}(1 - \frac{h}{f(\bar{\xi}_k,\alpha_k)})\Big) - \Big(1 - \prod_{k=0}^{N}(1 - \frac{h}{f(\xi_k,\alpha_k)})\Big)\right] \\
&\leqslant \mathrm{E}_x^{\sigma^h}\left[\mathbb{1}_{\bar{N}\leqslant N}\left\{\prod_{k=0}^{\bar{N}}(1 - \frac{h}{f(\xi_k,\alpha_k)}) - \prod_{k=0}^{\bar{N}}(1 - \frac{h}{f(\bar{\xi}_k,\alpha_k)})\right\}\right. \\
&\quad + \mathbb{1}_{N<\bar{N}}\left\{\prod_{k=0}^{N}(1 - \frac{h}{f(\xi_k,\alpha_k)}) - \prod_{k=0}^{N}(1 - \frac{h}{f(\bar{\xi}_k,\alpha_k)})\right. \\
&\quad \left.\left. + \Big(\prod_{k=0}^{N}(1 - \frac{h}{f(\bar{\xi}_k,\alpha_k)})\Big)\Big(1 - \prod_{k=N+1}^{\bar{N}}(1 - \frac{h}{f(\bar{\xi}_k,\alpha_k)})\Big)\right\}\right] \ .
\end{aligned} \tag{5.80}$$

First notice that

$$\begin{aligned}
&\mathrm{E}_x^{\sigma^h}\left[\frac{1}{f(\bar{\xi}_k,\alpha_k)}-\frac{1}{f(\xi_k,\alpha_k)}\mid k\leqslant(N\wedge\bar{N})\right]\\
&=\mathrm{E}_x^{\sigma^h}\left[\mathrm{E}[\frac{1}{f(\bar{\xi}_{k-1}+h\alpha_{k-1},\alpha_k)}-\frac{1}{f(\xi_{k-1}+\xi_k-\xi_{k-1},\alpha_k)}\mid\xi_{k-1},k-1\leqslant(N\wedge\bar{N})]\mid k\leqslant(N\wedge\bar{N})\right]\\
&\leqslant\mathrm{E}_x^{\sigma^h}\left[\frac{1}{f(\bar{\xi}_{k-1}+h\alpha_{k-1},\alpha_k)}-\frac{1}{f(\xi_{k-1}+h\alpha_{k-1},\alpha_k)}\mid k\leqslant(N\wedge\bar{N})\right]+M_f'\|\mathrm{cov}[\xi_k-\xi_{k-1}]\|\ ,
\end{aligned}\tag{5.81}$$

where the last inequality is deduced by (5.69a) and Lemma 5.4.3. Thus, by induction and by (5.69b), we have

$$\mathrm{E}_x^{\sigma^h}\left[\frac{1}{f(\bar{\xi}_k,\alpha_k)}-\frac{1}{f(\xi_k,\alpha_k)}\mid k\leqslant(N\wedge\bar{N})\right]\leqslant kM_f'h^2\ .\tag{5.82}$$

Let us focus on the first part of the sum in (5.80), for which we have

$$\begin{aligned}
&\mathrm{E}_x^{\sigma^h}\left[\mathbb{1}_{\bar{N}\leqslant N}\left\{\prod_{k=0}^{\bar{N}}(1-\frac{h}{f(\xi_k,\alpha_k)})-\prod_{k=0}^{\bar{N}}(1-\frac{h}{f(\bar{\xi}_k,\alpha_k)})\right\}\right]\\
&\leqslant\mathrm{E}_x^{\sigma^h}\left[\mathbb{1}_{\bar{N}\leqslant N}\left\{\Phi_k(\xi_k)\sum_{k=1}^{\bar{N}}\Big(\frac{h}{f(\bar{\xi}_k,\alpha_k)}-\frac{h}{f(\xi_k,\alpha_k)}\Big)\right\}\right]\\
&\leqslant h\mathrm{E}_x^{\sigma^h}\left[\mathbb{1}_{\bar{N}\leqslant N}(1-\frac{h}{\overline{f}})^{\bar{N}}\sum_{k=1}^{\bar{N}}kM_f'h^2\right]\\
&\leqslant\mathbb{P}(\mathbb{1}_{\bar{N}\leqslant N})2M_f'\overline{f}^2h\ ,
\end{aligned}\tag{5.83}$$

where $\Phi_k(\xi_k)$ is a random variable and $\Phi_k(\xi_k)\leqslant(1-\frac{h}{\overline{f}})^{\bar{N}}$. For the second part in (5.80), the first part of the sum is bounded by the same form as computed in (5.83). As for the remaining part, we notice that, by a similar computation as in (5.81),

$$\mathrm{E}_x^{\sigma^h}\left[d_{\mathcal{K}}(\bar{\xi}_k)-d_{\mathcal{K}}(\xi_k)\mid k\leqslant(N\wedge\bar{N})\right]\leqslant kM_t'h^2.\tag{5.84}$$

Then, we have

$$\begin{aligned}
&\mathrm{E}_x^{\sigma^h}\left[\mathbb{1}_{N<\bar{N}}\left\{\Big(\prod_{k=0}^{N}(1-\frac{h}{f(\bar{\xi}_k,\alpha_k)})\Big)\Big(1-\prod_{k=N+1}^{\bar{N}}(1-\frac{h}{f(\bar{\xi}_k,\alpha_k)})\Big)\right\}\right]\\
&\leqslant\mathrm{E}_x^{\sigma^h}\left[\mathbb{1}_{N<\bar{N}}\left\{(1-\frac{h}{\overline{f}})^N\sum_{k=N}^{\bar{N}}\Big(\frac{h}{f(\bar{\xi}_k,\alpha_k)}\Big)\right\}\right]\\
&\leqslant\mathrm{E}_x^{\sigma^h}\left[\mathbb{1}_{N<\bar{N}}(1-\frac{h}{\overline{f}})^N\frac{1}{\underline{f}}\Big(d_{\mathcal{K}}(\bar{\xi}_N)-d_{\mathcal{K}}(\xi_N)\Big)\right]\\
&\leqslant\mathrm{E}_x^{\sigma^h}\left[\mathbb{1}_{N<\bar{N}}(1-\frac{h}{\overline{f}})^N\frac{1}{\underline{f}}NM_t'h^2\right]\\
&\leqslant\mathbb{P}(\mathbb{1}_{N<\bar{N}})\frac{M_t'}{\underline{f}}h\ .
\end{aligned}\tag{5.85}$$

Combing (5.83) and (5.85), we have

$$v^h - w^h \leqslant \mathbb{P}(\mathbb{1}_{\bar{N}\leqslant N})2M'_f\overline{f}^2 h + \mathbb{P}(\mathbb{1}_{N<\bar{N}})(2M'_f\overline{f}^2 + \frac{M'_t}{\underline{f}})h \leqslant (2M'_f\overline{f}^2 + \frac{M'_t}{\underline{f}})h \ . \tag{5.86}$$

Combining with the result in Theorem 5.3.5, we have $v(x) - w^h(x) \leqslant (\frac{L_f}{2\underline{f}^2} + 2M'_f\overline{f}^2 + \frac{M'_t}{\underline{f}})h$. □

The following theorem is then a direct consequence of Proposition 5.4.2 and Proposition 5.4.4, which we state as the main result of this subsection.

Theorem 5.4.5. *Suppose that Assumption (A7), Assumption (A8) and Assumption (A9) hold, taking I^x be linear interpolation operator I_1, there exists a constant C_w depends on $L_f, L_v, \overline{f}, \underline{f}$, C_v, M'_f, M'_t such that, for every $0 < h < \frac{1}{\overline{f}}$,*

$$\sup_{x\in\mathbb{R}^d} \|w^h(x) - v(x)\| \leqslant C_w h \ . \tag{5.87}$$

5.4.4 A Particular Piecewise Linear Interpolation Operator

In this section, we will give a specific piecewise linear interpolation operator that leads to an efficient implementation, particularly for the isotropic eikonal equation. Notice that computing the minimum in (5.57) is not trivial, especially when the dimension is high. Moreover, generally, in the d dimensional case, we need at least the value in $d+1$ nodes of the grid, in order to compute the interpolation in one node. We describe here one possible way to define an interpolation operator and to compute the minimum in (5.57), within a regular grid with space mesh step equal to time step i.e., $\Delta x_i = h, \forall i \in \{1, 2, \ldots, d\}$. This interpolation operator is based on the work of [CF07], in which the convergence is shown in the isotropic case.

Let $x = (x_1, x_2, \ldots, x_d)$ denote a point of X. Roughly speaking, the $d-$dimensional space is "partitioned" into 2^d orthants. We consider only the open orthants, since their boundaries are negligible. Let us denote by V the approximate value functionin the grid point $x \in X$. The values of the interpolation $I^x[V](x + h\alpha)$ with $\alpha \in S_1$ are defined (differently) for α in each orthant, and the minimum value in each orthant is first computed. Then, the minimum will be obtained by further taking the minimum among the values in all orthants.

Denote by $e_1, \ldots, e_d$ the vectors of the canonical basis of $\mathbb{R}^d$. We compute the minimum in the positive orthant using $d+1$ nodes: $x^l := x + he_l, l \in \{1, \ldots, d\}$, and $x_{+1} := x + h(e_1 + e_2 + \cdots + e_d)$. The minimum in other orthants will be computed using the same method.

The interpolated value function in $x + h\alpha$ with α in the positive orthant of the sphere S_1, denoted by $v^{s,1}$, will be given by the linear interpolation of $V(x^1)$, $V(x^2)$, $\ldots$, $V(x^d)$ and $V(x_{+1})$, which is equal to

$$v^{s,1}(x + h\alpha) = \sum_{k=1}^{d} \alpha_k V(x^k) + \frac{V(x_{+1}) - \sum_{l=1}^{d} V(x^l)}{d-1}\Big((\sum_{\ell=1}^{d} \alpha_\ell) - 1\Big) \ . \tag{5.88}$$

We then use $(\theta_1, \theta_2, \ldots, \theta_{d-1})$, $\theta_k \in (0, \frac{\pi}{2})$, to represent a vector $\alpha \in S_1$ belonging to the positive orthant, that is

$$\alpha_1 = \cos(\theta_1), \alpha_2 = \sin(\theta_1)\cos(\theta_2), \ldots, \alpha_d = \sin(\theta_1)\sin(\theta_2)\cdots\sin(\theta_{d-1}) \ . \tag{5.89}$$

This allows one to rewrite (5.88) as a function of $(\theta_1, \theta_2, \ldots, \theta_{d-1})$. By doing so, one can consider the result of the optimization in the first equation of (5.57), with w^h replaced by V and I replaced

by I^x, restricted to the positive orthant, as an approximate value of $V(x)$, denoted by V^1, and given by:

$$V^1(x) = \min_{\theta_1,\dots,\theta_{d-1}} \left\{ (1 - \frac{h}{f(x,\alpha)}) v^{s,1}(x + h\alpha) + \frac{h}{f(x,\alpha)} \right\} \ . \tag{5.90}$$

Notice that the minimum in equation (5.90) is easier to compute by taking the minimum first on θ_{d-1}, then θ_{d-2}, until θ_1. Indeed, we notice in (5.89), that only the last two entries of α contain θ_{d-1}. Thus, the minimum of (5.90) over θ_{d-1} can be computed separately. Moreover, in the isotropic case, meaning $f(x,\alpha) \equiv f(x), \forall \alpha \in S_1$, the minimal θ_{d-1} is independent of $\theta_1, \dots \theta_{d-2}$, due to the special form of (5.89) and (5.88). The iteratively computation over θ_{d-2} to θ_1 will be the same.

Then, the fully discretized scheme, using the interpolation operator described as above, is as follows:

$$\begin{cases} V(x_i) = \min\limits_{k \in \{1,2,\dots,2^d\}} V^k(x_i), & x_i \in X^h \cap (\mathbb{R}^d \setminus \mathcal{K}) \ , \\ V(x_i) = 0, & x_i \in X^h \cap \mathcal{K} \ . \end{cases} \tag{5.91}$$

Proposition 5.4.6. *Suppose that Assumption (A7), Assumption (A8) and Assumption (A9) hold, taking I^x as in* (5.91)*, there exists a constant C_V depends on $L_f, L_v, \overline{f}, \underline{f}, C_v, M_f', M_t'$ such that, for every $0 < h < \frac{1}{\overline{f}}$*

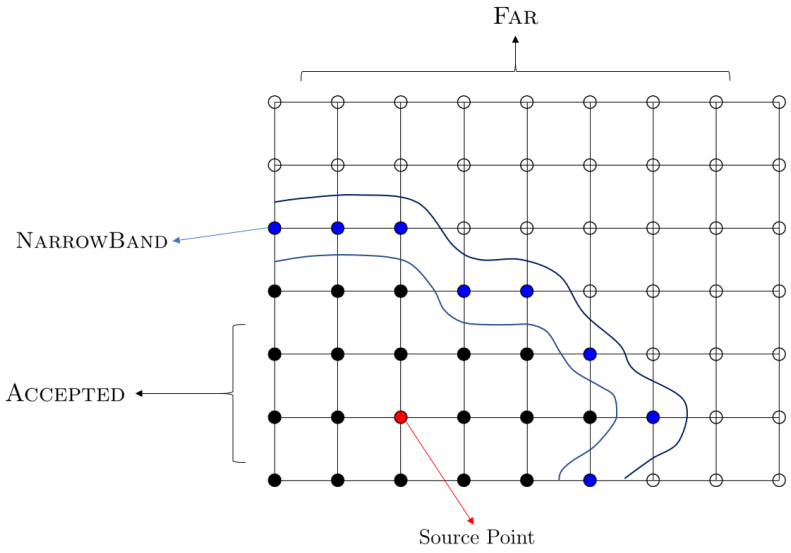
$$\sup_{x \in X^h} \|V(x) - v(x)\| \leqslant C_V h \ . \tag{5.92}$$

Sketch of Proof. We first observe that by replacing I_1 with the interpolation operator defined in (5.91), the property in (5.61) holds. Then, following a similar analysis, Proposition 5.4.2 holds with w^h replaced by V. Moreover, by the definition (4.67), the value at $x + h\alpha$ is a convex combination of the values at the points of the simplex that contains $x + h\alpha$. Thus, this interpolation can also be explained as a stochastic control problem, similar to (5.67) and (5.68), but with $\lambda(\cdot)$ depending also on x_i. The property (5.69) also holds for this controlled process, and as a result, Proposition 5.4.4 holds. The conclusion of the result follows. □

5.4.5 The Fast-Marching Method and Its Convergence Analysis

We briefly recall the fast marching method introduced by Sethian [Set96] and Tsitsiklis [Tsi95], which is one of the most effective numerical methods to solve the eikonal equation. Its initial idea takes advantage of the property that the evolution of the domain encircled by the front is monotone non-decreasing, thus one is allowed to only focus on the computation around the front at each iteration. Generally, it has computational complexity (number of arithmetic operations) in the order of $K_d M \log(M)$ in a d-dimensional grid with M points (see for instance [Set96; CF07]), where the constant K_d depends on the discretization scheme.

The fast marching method is searching the nodes of X according to a special ordering and computes the approximate value function in just one iteration. The special ordering is constructed in such a way that the value function is monotone non-decreasing in the direction of propagation. This construction is done by dividing the nodes into three groups (see below figure): Far, which contains the nodes that have not been searched yet; Accepted, which contains the nodes at which the value function has been already computed and settled – by the monotone property, in the subsequent search, we do not need to update the value function of those nodes; and NarrowBand, which contains the nodes "around" the front – at each step, the value function is updated only at these nodes.



At each step, the node in NarrowBand with the smallest value is added into the set of Accepted nodes, and then the NarrowBand and the value function over NarrowBand are updated, using the value of the last accepted node. The computation is done by appying an update operator $\mathcal{U} : (\mathbb{R} \cup \{+\infty\})^X \to (\mathbb{R} \cup \{+\infty\})^X$, which is based on the discretization scheme. The classical update operators are based on finite-difference (see for instance [Set96]) or semi-lagrangian discretizations (see for instance [CF07]). Sufficient conditions on the update operator $\mathcal{U}$ for the convergence of the fast marching algorithm are that the approximate value function on X is the unique fixed point of $\mathcal{U}$ satisfying the boundary conditions, and that $\mathcal{U}$ is monotone and causal [Set96].

A generic partial fast marching algorithm is given in Algorithm 5.1. We call it *partial* because the search stops when all the nodes of the ending set End are accepted. The usual fast marching algorithm is obtained with End equal to the mesh grid X and Start equal to the nodes in target set.

Algorithm 5.1 Partial Fast Marching Method (compare with [Set96; CF07]).

Input: Mesh grid X; Update operator $\mathcal{U}$. Two set of nodes: Start and End.
Output: Approximate value function V and Accepted set.
Initialization: Set $V(x) = +\infty, \forall x \in X$. Set all nodes as Far.

Add Start to Accepted, add all neighborhood nodes to NarrowBand.
Compute the initial value $V(x)$ of the nodes in NarrowBand.
while (NarrowBand is not empty and End is not accepted) **do**
    Select x^* having the minimum value $V(x^*)$ among the NarrowBand nodes.
    Move x^* from NarrowBand to Accepted.
    for All nodes y not in Accepted, such that $\mathcal{U}(V)(y)$ depends on x^* **do**
        $V(y) = \mathcal{U}(V)(y)$
        if y **then** is not in NarrowBand
            Move y from Far to NarrowBand.
        end if
    end for
end while

Let us now consider the fast-marching method with a particular update operator as described

in (5.91). I.e., we set the input update operator in Algorithm 5.1 as follows:

$$\mathcal{U}(V)(x) := \min_{k\in\{1,2,\dots,2^d\}} V^k(x) \ , \tag{5.93}$$

where V^k is defined similarly as in (5.90). Then, we have the following result:

Theorem 5.4.7. *Under Assumption (A7) and Assumption (A8), we have*

$$\sup_{x\in X} \|V(x) - v(x)\| \leqslant C_V h \ , \tag{5.94}$$

where C_V is a constant depends on C_v, C_1 and the diameter of the grid X.

Sketch of Proof. It is enough to notice that the interpolation operator is the same as (5.91), as the result is concluded by Proposition 5.4.6. □

Corollary 5.4.8. In order to get an error bound on the value of the problem (5.1) less of equal ε, we shall take the mesh grid $h = (C_V)^{-1}\varepsilon$. Then, the total computational complexity of the fast-marching method is $\tilde{O}\Big((2C_V\varepsilon^{-1})^d\Big)$.

Proof. The choice of the mesh step h is determined based on the error estimates in (5.94). This results in the presence of $O((C_v\varepsilon^{-1})^d)$ nodes in the grid. Moreover, one step update using the update operator (5.93) needs $O(d \times 2^d)$ arithmetic operations, and the fast-marching method needs a number of update steps in the order of $O(M\log(M))$ to operate on a grid with M nodes. Then result is then concluded. □

5.5 Convergence Under a Particular Sate Constraint, Application to Computational Complexity of The Multilevel Fast-Marching Method

In the recent work [AGL23a], the authors introduced a multilevel fast-marching method. We determine the computational complexity of this method as a function of the convergence rate of the original fast-marching method, and of the "stiffness" of the value function. In this section, we demonstrate that the convergence rate is equal to one for both the original problem and the problem with a particular state constraint. As a result, we can achieve the ideal complexity bound stated in [AGL23a] when the "stiffness", β, is equal to 1.

5.5.1 A Particular State Constraint of the Minimum Time Problem

We first briefly describe a particular state constraint problem introduced in [AGL23a]. To simplify the explanation, we begin by considering the problem without any state constraints. We aim to solve the following minimum time problem:

$$\inf\ \tau \geqslant 0 \qquad s.t. \begin{cases} \dot{y}(t) = f(y(t),\alpha(t))\alpha(t),\ \forall t\in[0,\tau] \ , \\ y(0)\in\mathcal{K}_{\text{src}},\ y(\tau)\in\mathcal{K}_{\text{dst}} \ , \\ \alpha(t)\in\mathcal{A},\ \forall t\in[0,\tau] \ , \end{cases} \tag{5.95}$$

where $\mathcal{K}_{\text{src}}$ and $\mathcal{K}_{\text{dst}}$ are two disjoint compact subsets of $\mathbb{R}^d$ (called the *source* and the *destination* resp.). One way to solve the above problem (5.95) is to consider the set $\mathcal{K}_{\text{dst}}$ as the "target" set $\mathcal{K}$ in Section 5.2.2, and using the same change of variable technique to get the new control

problem. Denote $v_{\to \mathrm{d}}$ the value function of such a control problem, the associated HJB equation is as follows:

$$\begin{cases} F(x, v_{\to \mathrm{d}}(x), Dv_{\to \mathrm{d}}(x)) = 0, & x \in \mathbb{R}^d \setminus \mathcal{K}_{\mathrm{dst}} \ , \\ v_{\to \mathrm{d}}(x) = 0, & x \in \partial \mathcal{K}_{\mathrm{dst}} \ . \end{cases} \tag{5.96}$$

Once (5.96) is solved, one can easily get the value of the problem (5.95) by computing the minimum of $v_{\to \mathrm{d}}(x)$ over $\mathcal{K}_{\mathrm{src}}$. We shall denote the set of minimum points by $\mathcal{X}_{\mathrm{src}} := \mathrm{Argmin}_{x \in \mathcal{K}_{\mathrm{src}}} v_{\to \mathrm{d}}(x)$.

An alternative approach to solving problem (5.95) is to treat the set $\mathcal{K}_{\mathrm{src}}$ as the "target" set $\mathcal{K}$ in Section 5.2.2, while replacing the dynamics (5.3) by:

$$\begin{cases} \dot{y}(t) = -f(y(t), \alpha(t))\alpha(t), \ \forall t \geqslant 0 \ , \\ y(0) = x \ , \end{cases} \tag{5.97}$$

Denote $v_{\mathrm{s}\to}$ the value function of this control problem, the associated HJB equation is then as follows:

$$\begin{cases} F^*(x, v_{\mathrm{s}\to}(x), Dv_{\mathrm{s}\to}(x)) = 0, & x \in \mathbb{R}^d \setminus \mathcal{K}_{\mathrm{src}}, \\ v_{\mathrm{s}\to}(x) = 0, & x \in \partial \mathcal{K}_{\mathrm{src}} \ , \end{cases} \tag{5.98}$$

where $F^*(x, r, p) = F(x, r, -p)$. By doing so, to solve the problem (5.95), one can also solve the equation (5.98) to get $v_{\mathrm{s}\to}$, and then compute the minimum of $v_{\mathrm{s}\to}(s)$ over $\mathcal{K}_{\mathrm{dst}}$. We shall also denote the set of minimum points by $\mathcal{X}_{\mathrm{dst}} := \mathrm{Argmin}_{x \in \mathcal{K}_{\mathrm{dst}}} v_{\mathrm{s}\to}(x)$. Then, we have the following result:

Lemma 5.5.1 (Corollary of [AGL23a, Prop. 3.3])**.** $v^* := \inf_{x \in \mathcal{K}_{dst}} v_{\to \mathrm{d}}(x) = \inf_{x \in \mathcal{K}_{src}} v_{\mathrm{s}\to}(x)$.

Let us now describe the new state constraint problem presented in [AGL23a] that aims to solve problem (5.95). For every $x \in \mathbb{R}^d$ and $v = (v_{\mathrm{s}\to}, v_{\to \mathrm{d}})$, we denote

$$\mathcal{F}_v(x) = v_{\mathrm{s}\to}(x) + v_{\to \mathrm{d}}(x) - v_{\mathrm{s}\to}(x) v_{\to \mathrm{d}}(x) \ . \tag{5.99}$$

Moreover, for every $\mu > 0$, we select a function $\mathcal{F}_v^\mu : \mathbb{R}^d \to \mathbb{R}$, that is $\mathcal{C}^d$, and that approximates $\mathcal{F}$, i.e.,

$$\|\mathcal{F}_v^\mu - \mathcal{F}_v\|_\infty < \mu \ . \tag{5.100}$$

Then, we consider a domain, determined by a parameter $\eta > 0$, and defined as follows

$$\mathcal{O}_\eta^\mu = \{x \in (\mathbb{R}^d \setminus (\mathcal{K}_{\mathrm{src}} \cup \mathcal{K}_{\mathrm{dst}})) \mid \mathcal{F}_v^\mu(x) < \inf_{y \in \mathbb{R}^d} \{\mathcal{F}_v(y) + \eta\} \ \} \ , \tag{5.101}$$

with $\mu < \eta$. We intend to reduce the state space of the original problem (5.95) to the closure $\overline{\mathcal{O}_\eta^\mu}$ of $\mathcal{O}_\eta^\mu$. More precisely, let us first consider the problem with target $\mathcal{K}_{\mathrm{dst}}$. We shall consider a new state constraint problem so that the state $y(s)$ stays within the domain $\overline{\mathcal{O}_\eta^\mu}$, for every $s \geqslant 0$. This leads to a new set of controls:

$$\mathcal{A}_{\eta,x} := \{\alpha \in \mathcal{A} \mid y_\alpha(x; s) \in \overline{\mathcal{O}_\eta^\mu}, \ \text{for all } s \geqslant 0\} \ . \tag{5.102}$$

Let us denote $v_{\to \mathrm{d}}^\eta(x)$ the value function of this state constraint problem. The associated HJB equation has the following form:

$$\begin{cases} F(x, v_{\to \mathrm{d}}^\eta(x), Dv_{\to \mathrm{d}}^\eta(x)) = 0, & x \in \mathcal{O}_\eta^\mu \ , \\ F(x, v_{\to \mathrm{d}}^\eta(x), Dv_{\to \mathrm{d}}^\eta(x)) \geqslant 0, & x \in \partial \mathcal{O}_\eta^\mu \setminus \partial \mathcal{K}_{\mathrm{dst}} \ , \\ v_{\to \mathrm{d}}^\eta(x) = 0, & x \in \partial \mathcal{K}_{\mathrm{dst}} \ . \end{cases} \tag{5.103}$$

5.5.2 Convergence Rate of The Semi-Lagrangian Scheme Under State Constraint

In this section, we will show the convergence rate of the semi-lagrangian type discretization of the system (5.103), that is,

$$\begin{cases} v^h_{\to \mathrm{d}}(x) = \min\limits_{\alpha\in S_1}\left\{(1-\dfrac{h}{f(x,\alpha)})v^h_{\to \mathrm{d}}(x+h\alpha)+\dfrac{h}{f(x,\alpha)}\right\}, & x\in \overline{\mathcal{O}^\mu_\eta}\setminus \mathcal{K}_{\mathrm{dst}}\ , \\ v^h_{\to \mathrm{d}}(x) = 1, & x\notin (\overline{\mathcal{O}^\mu_\eta}\cup \mathcal{K}_{\mathrm{dst}})\ , \\ v^\eta_{\to \mathrm{d}}(x) = 0, & x\in \mathcal{K}_{\mathrm{dst}}\ . \end{cases} \tag{5.104}$$

We start by considering the $\delta-$optimal trajectory.

Definition 5.5.2. For every $x\in\mathbb{R}^d$, we say that $y_{\alpha^\delta}(x;\cdot):[0,\tau]\to\mathbb{R}^d$ is a δ-optimal trajectory with associated δ-optimal control $\alpha^\delta:[0,\tau]\to S_1$ for the problem with target $\mathcal{K}_{\mathrm{dst}}$ if :

$$y_{\alpha^\delta}(x;\tau)\in\mathcal{K}_{\mathrm{dst}} \quad\text{and}\quad \int_0^\tau e^{-t}dt \leqslant v_{\to \mathrm{d}}(x)+\delta\ .$$

We denote by Γ^δ_x the set of $\delta-$geodesic points starting from x, i.e., $\Gamma^\delta_x=\{y_{\alpha^\delta}(x;t)\mid t\in[0,\tau],\ \alpha^\delta:[0,\tau]\to S^1\ \delta\text{-optimal }\}$. We define analogously δ-optimal trajectories for the problem in reverse direction, and denote by $\tilde\Gamma^\delta_x$ the set of δ-geodesic points starting from x in the reverse direction.

Then, for the trajectories that are $\delta-$optimal in the both alternative directions, we have the following result.

Lemma 5.5.3 (Corollary of [AGL23a, Prop. 3.11])**.** *Denote*

$$\mathcal{X}^\delta_{src}=\{x\in\partial\mathcal{K}_{src}\mid v_{\to \mathrm{d}}(x)\leqslant v^*+\delta\},\ \mathcal{X}^\delta_{dst}=\{x\in\partial\mathcal{K}_{dst}\mid v_{\mathrm{s}\to}(x)\leqslant v^*+\delta\}\ ,$$

we have:

$$\cup_{\delta'\in[0,\delta]}\cup_{x\in\mathcal{X}^{\delta-\delta'}_{src}}\{\Gamma^{\delta'}_x\}=\cup_{\delta'\in[0,\delta]}\cup_{x\in\mathcal{X}^{\delta-\delta'}_{dst}}\{\tilde\Gamma^{\delta'}_x\}. \tag{5.105}$$

Let us denote Γ^δ the set in (5.105), and call it *the set of $\delta-$geodesic points from $\mathcal{K}_{src}$ to $\mathcal{K}_{dst}$*. Indeed, the set Γ^δ and $\mathcal{O}^\mu_\eta$ defined in (5.101) constitute equivalent families of neighborhoods of the optimal trajectory. In particular, we have the following result

Lemma 5.5.4 (Corollary of [AGL23a, Lemma 3.14, Lemma 3.15])**.** *For every $\eta>\delta>0$, for every $\delta'>0$, we have $\Gamma^\delta\subseteq\overline{\mathcal{O}^\mu_\eta}\subseteq\Gamma^{\eta+\delta'}$, for μ small enough.*

Based on the above property, we have the convergence result of the semi-lagrangian scheme (5.104) as follows.

Theorem 5.5.5. *Suppose Assumption (A7), Assumption (A8) and Assumption (A9) hold (with $\mathcal{K}$ replace by $\mathcal{K}_{dst}$).*

(i) *There exists a constant C'_v depends on M_f, M_t, $\overline{f}$, $\underline{f}$ such that, for every $0\leqslant\delta\leqslant(\eta-h)$, for every $x\in\Gamma^\delta$ and $z\in\mathbb{R}^d$ such that $[x-z,x+z]\subseteq\Gamma^\delta$,*

$$v^h_{\to \mathrm{d}}(x+z)-2v^h_{\to \mathrm{d}}(x)+v^h_{\to \mathrm{d}}(x-z)\leqslant C'_v\|z\|^2\ . \tag{5.106}$$

(ii) *There exists a constant C'_1 depends on M_f, M_t, M'_f, M'_t, L_v, $\overline{f}$, $\underline{f}$ such that, for every $0\leqslant\delta\leqslant(\eta-h)$ and $x\in\Gamma^\delta$, for every $0<h<\frac{1}{\overline{f}}$,*

$$\sup_{x\in\Gamma^\delta}\|v^h_{\to \mathrm{d}}(x)-v_{\to \mathrm{d}}(x)\|\leqslant C'_1h\ . \tag{5.107}$$

5.5.3 The Multilevel Fast-Marching Method and Its Computational Complexity

We begin by describing the algorithm with two levels of grids. Firstly, a coarse grid is used to approximate $\mathcal{O}_\eta^\mu$ by applying the fast-marching search in both directions, which solve equations (5.96) and (5.98) with relaxed accuracy and error requirements. Then, a finer grid is used to discretize the approximation of $\mathcal{O}_\eta^\mu$, and to solve the corresponding state constraint equation (5.103).

Let X^H denote a grid with a constant mesh step of H. The first step consists in applying Algorithm 5.1 to solve equations (5.96) and (5.98). This involves performing a partial fast-marching method with mesh grid X^H, an update operator that is adapted to the HJ equation (5.96) or (5.98), as well as the appropriate sets START and END. This yields numerical approximations $V_{\mathrm{s}\to}^H$ and $V_{\to\mathrm{d}}^H$ of the value functions $v_{\mathrm{s}\to}$ and $v_{\to\mathrm{d}}$, on the sets of accepted nodes $A_{\mathrm{s}\to}^H$ and $A_{\to\mathrm{d}}^H$, respectively. Then we select and denote by O_η^H the set of *active* nodes, which is determined by a parameter η_H, as follows:

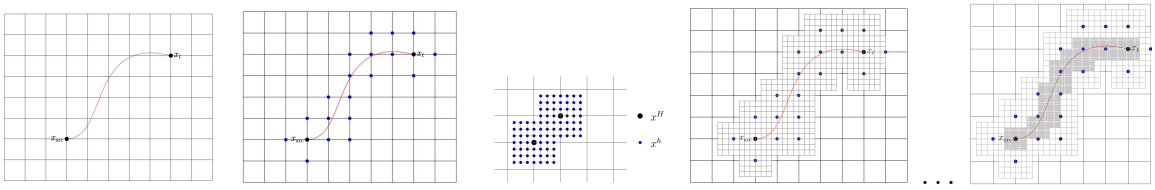
$$O_\eta^H = \left\{ x \in X^H \mid \mathcal{F}_{V^H}(x^H) \leqslant \min_{y^H \in X^H} \mathcal{F}_{V^H}(y^H) + \eta_H \right\} . \tag{5.108}$$

As for the computation in the fine grid, we denote by X^h a grid with a constant mesh step of h. We select and denote the fine grid nodes as follows:

$$G_\eta^h = \left\{ x^h \in X^h \mid \exists x^H \in O_\eta^H : \|x^h - x^H\|_\infty \leqslant \max(H-h, h) \right\} . \tag{5.109}$$

Then, the computation will only be done in the selected fine grid nodes, which means that a full fast marching algorithm Algorithm 5.1 is applied in the restricted fine grid G_η^h, with the update operator of one direction HJ equation (for instance with target set $\mathcal{K}_{\mathrm{dst}}$).

The computation in two levels of grids can be extended to the multilevel case. We construct finer and finer grids, considering the fine grid of the previous step as the coarse grid of the current step, and defining the next fine grid by selecting the actives nodes of this coarse grid. The algorithm is detailed in Algorithm 5.2, and some possible grids generated by our algorithm are shown in the following Figure 5.1.



(a) Level-0 (b) Active Nodes (c) Fine grid (d) Level-1 ... (e) Level-2

Figure 5.1: Sketch of MLFM.

In [AGL23a], the computational complexity of Algorithm 5.2 is shown to be dependent on two factors: γ, which is the convergence rate of the fast-marching method, and β, which measures the "stiffness" of the value function around the optimal trajectories. In Section 5.4.5 and Section 5.5.2, we demonstrate that $\gamma = 1$ for both the original problem and the problem with a particular state constraint $\mathcal{O}_\eta^\mu$. As a consequence, the computational complexity is solely a function of β. In some particular problems, we can find that $\beta = 1$, leading to the ideal computational complexity.

We state in the following the improved computational complexity for Algorithm 5.2 as the main result of this section.

Algorithm 5.2 Two-Level Fast-Marching Method (2LFM)

Input: The mesh steps, grids and selection parameters: H_l, X^{H_l}, η_l, for $l \in \{1, 2, \ldots, N\}$.
Input: Two update operators $\mathcal{U}_{\to \mathrm{d}}$ and $\mathcal{U}_{\mathrm{s}\to}$ adapted to both directions HJ equations.
Input: Target sets: $\mathcal{K}_{\mathrm{src}}, \mathcal{K}_{\mathrm{dst}}$.
Output: The final fine grid FINE and approximate value function $V_{\to \mathrm{d}}^{h,N}$ on FINE.

Set COARSE-GRID to X^{H_1}.
for $(l = 1$ to $N - 1)$ **do**
    Apply Algorithm 5.1 with Input grid COARSE-GRID, update operator $\mathcal{U}_{\to \mathrm{d}}$, START $= \mathcal{K}_{\mathrm{dst}} \cap X^{H_l}$ and END $= \mathcal{K}_{\mathrm{src}} \cap X^{H_l}$, and output $V_{\to \mathrm{d}}^{H_l}$ and $A_{\to \mathrm{d}}^{H_l}$.
    Apply Algorithm 5.1 with Input grid COARSE-GRID, update operator $\mathcal{U}_{\mathrm{s}\to}$, START $= \mathcal{K}_{\mathrm{src}} \cap X^{H_l}$ and END $= \mathcal{K}_{\mathrm{dst}} \cap X^{H_l}$, and output $V_{\mathrm{s}\to}^{H_l}$ and $A_{\mathrm{s}\to}^{H_l}$.
    for (Every node x^{H_l} in $A_{\mathrm{s}\to}^{H_l} \cap A_{\to \mathrm{d}}^{H_l}$) **do**
        if $(\mathcal{F}_{V^{H_l}(x^{H_l})} \leqslant \min_{x^{H_l} \in X^{H_l}} \{\mathcal{F}_{V^{H_l}}(x^{H_l}) + \eta_l\})$ **then**
            Set x^{H_l} as ACTIVE.
        end if
    end for
    Set FINE to be emptyset.
    for (Every node x^{H_l} in the ACTIVE set) **do**
        for (Every $x^{H_{l+1}} \in X^{H_{l+1}}$ satisfying $\|x^{H_{l+1}} - x^{H_l}\|_\infty \leqslant \max\{(H_l - H_{l+1}), H_{l+1}\}$) **do**
            if $x^{H_{l+1}}$ does not exist in set FINE **then**
                Add $x^{H_{l+1}}$ in the set FINE.
            end if
        end for
    end for
    Set the new COARSE-GRID to be the current FINE.
end for
Apply Algorithm 5.1 with Input grid FINE, update operator $\mathcal{U}_{\to \mathrm{d}}$, START $= \mathcal{K}_{\mathrm{dst}} \cap$ FINE and END $= \mathcal{K}_{\mathrm{src}} \cap$ FINE, and output $V_{\to \mathrm{d}}^{h,N}$.

Theorem 5.5.6 (Corollary of [AGL23a, Th. 4.4, Th. 5.4])**.** *Suppose Assumption (A7), Assumption (A8), Assumption (A9) hold (with both $\mathcal{K}$ replaced by $\mathcal{K}_{src}$ and $\mathcal{K}$ replaced by $\mathcal{K}_{dst}$), and $d \geqslant 2$, then we have*

(i) *There exists a constant $C_\eta \geqslant 0$ such that, by setting $\eta_l = C_\eta H_l$ for every $l \in \{1, 2, \dots, N-1\}$, we have $V^{h,N}_{\to \mathrm{d}}(x) = V^h_{\to \mathrm{d}}(x)$, for every $x \in G^h_{\eta_{N-1}} \cap \Gamma^\delta$ and $\delta < \eta_{N-1}$. Consequently, $V^{h,N}_{\to \mathrm{d}}$ converges towards $v_{\to \mathrm{d}}(x)$ as $h \to 0$.*

(ii) *In order to obtain an error bound on the value of the problem* (5.95) *less or equal ε, we shall take $h = C_\gamma^{-1}\varepsilon$, $N = \lfloor d \log(\frac{1}{\varepsilon}) \rfloor$, η_l as in* (i) *and $H_l = h^{\frac{l}{N}}$, for every $l \in \{1, 2, \dots, N\}$. Then, the total computational complexity of Algorithm 5.2 is $\widetilde{O}((C_m)^d (\frac{1}{\varepsilon})^{1+(d-1)(1-\beta)})$. When $\beta = 1$, it reduces to $\widetilde{O}((C_m)^d \frac{1}{\varepsilon})$.*

Chapter 6

An Adaptive Multi-Level Max-Plus Method for Deterministic Optimal Control Problems

***

A shorter version of this chapter, showing the first idea, has been published in the proceedings of the IFAC World Congress 2023 [AGL23b].

Abstract. We introduce a new numerical method to approximate the solution of a finite horizon deterministic optimal control problem. We exploit two Hamilton-Jacobi-Bellman PDE, arising by considering the dynamics in forward and backward time. This allows us to compute a neighborhood of the set of optimal trajectories, in order to reduce the search space. The solutions

of both PDE are successively approximated by max-plus linear combinations of appropriate basis functions, using a hierarchy of finer and finer grids. We show that the sequence of approximate value functions obtained in this way does converge to the viscosity solution of the HJB equation in a neighborhood of optimal trajectories. Then, under certain regularity assumptions, we show that the number of arithmetic operations needed to compute an approximate optimal solution of a d-dimensional problem, up to a precision ε, is bounded by $O(C^d(1/\varepsilon))$, for some constant $C > 1$, whereas ordinary grid-based methods have a complexity in $O(1/\varepsilon^{ad})$ for some constant $a > 0$.

6.1 Introduction

6.1.1 Motivation and Context

We are interested here in the numerical solution of finite horizon deterministic optimal control problems. Such problems are associated to Hamilton-Jacobi-Bellman (HJB) equations via the Bellman dynamic programming principle (see for instance [FS06]). The value function, for this class of optimal control problems, has been characterized as the solution of a HJB PDE, in the viscosity sense ([CL83]). Several classes of numerical methods have been proposed to solve such PDEs. Among them, we mention the finite difference schemes introduced in [CL84], which involve a direct discretization of the HJB equation, and the semi-lagrangian schemes, studied in particular in ([Fal87], [FF14]), which rely on applying the dynamic programming principle to a discrete time control problem derived from an Euler discretization in time of the dynamics. In both cases, the discretized system can be interpreted as the dynamic programming equation of a stochastic optimal control problem.

More recently, max-plus based discretization schemes have been developed to solve the first order HJB equations. In a broad sense, these methods take advantage of the max-plus linearity of the evolution semigroup of the HJB PDE, the so called *Lax-Oleinik semigroup*. After a time discretization, this allows one to approximate the value function for a given time horizon, by a supremum of appropriate basis functions, for instance quadratic forms. Such suprema are propagated by the action of the Lax-Oleinik semigroup, between two successive time steps. In particular, in [FM00], the authors approximated the value function at a given time horizon by the max-plus linear combination of "basis functions" together with scalars. The computation of the scalars was carried out inductively by applying a max-plus linear operator at each time step. This scheme can be interpreted as a dynamic programming equation of a discrete deterministic optimal control problem. Alternatively in [AGL08], a similar form of approximation for the value function was proposed. To derive the recursive equations for the scalars, the authors introduced a family of "test" functions. The inductive computation of the scalars in this case involved applying a nonlinear operator, which can be though of a projection on the space of basis functions and then on the space of tests functions. This scheme can be interpreted as a dynamic programming equation of a deterministic zero-sum two player game.

Both of the aforementioned methods exhibit advantages in solving various classes of control problems and the associated HJB equations under specific regularity conditions. Nevertheless, their computational complexities remain comparable to those of classical grid-based methods (see also the further development in [Lak07; GMQ11]), and thus suffer from the *curse-of-dimensionality*, that is the size of nonlinear systems to be solved is exponential in the dimension d. In [McE07], McEneaney introduced a curse-of-dimensionality free method to solve infinite horizon switched optimal control problem, for which the Hamiltonian is expressed as a maximum of finite many "simpler" Hamiltonians. Each of the Hamiltonians is a linear/quadratic form originating from a linear quadratic optimal control problem. The author demonstrates that

the complexity exhibits cubic growth in dimension (of the state)(see also [McE09]). This complexity, however, is bounded by a number that is exponential in the number of iterations, which is referred to as the "curse of complexity". Several "pruning" methods are proposed to improve such complexity bound, for instance, in [MDG08b; Sri+10; GMQ11; Qu14b]. Other approaches that use the max-plus linearity of HJB PDEs also demonstrate an advantage on mitigating the curse-of-dimensionality, among them we cite the work of Dower and McEneaney [DM11; DM15], of Darbon, Dower and Meng [DDM23], of Yegorov and Dower [YD21b].

Other approaches to reduce *curse-of-dimensionality* is to focus on finding one (or several) optimal trajectories. We mention, for instance, the Pontryagin Maximum Principle approach [RZ99; BT13], the stochastic dual dynamic programming (SDDP) [Sha11; GLP15]. More recent developments, using the property (or structure) of the optimal trajectories, include the computation of the value function at one given point by constructing the grid from the possible trajectories and reducing the set of trajectories using Lipschitz continuity properties, together with the low dimensionality of the control set, like in [AFS19], [AFS20], and [BGZ22]. In [AGL23a], we introduced a multi-level fast-marching method, which focus on the neighborhood of optimal trajectories, and such neighborhood is approximated by a hierarchy of levels of grids. The present chapter is adapting somehow similar ideas and analysis as in [AGL23a].

6.1.2 Contribution

Here, we address the curse-of-dimensionality issue with another approach. The main idea is to consider a hierarchy of finer and finer irregular grids, concentrated around optimal trajectories, thus allowing us to dynamically reduce the search space, while increasing the precision. This is achieved by considering a pair of HJB PDE, associated to two optimal control problems: one with a forward dynamics, fixed initial state and free final state, and a dual one, with a backward dynamics, fixed final state and free initial state. The value functions of these two PDE allow us to compute a family of nested neighborhoods of optimal trajectories. Then, we adaptively add new basis functions, from one grid level to the next one, to refine the approximation. These new basis functions are chosen to be concentrated near the optimal trajectories of the control problem, and the refined neighborhood of optimal trajectories is computed from the solutions of the two HJB PDE in the coarser grid.

We show that using our algorithm, the number of basis functions needed to get a certain error ε is considerably reduced. Indeed, for a d-dimensional problem, under certain regularity assumptions, we get a complexity bound of $C^d(1/\varepsilon)^{\frac{1}{2}}$ arithmetic operations, for some constant $C > 1$. This should be compared with methods based on regular grids, which yield complexity bounds of order $\mathcal{O}(1/\varepsilon^{ad})$ in which $a > 0$ depends on regularity assumptions and on the order of the scheme (see for instance [BC08]). With our adaptative method, the curse of dimensionality remains only present in the term C^d.

The present work extends the idea of dynamic grid refinement around tubular neighborhood of optimal trajectories, originally introduced in [AGL23a] to solve semi-Lagrangian discretizations of special, minimal time, problems. By comparison, the main novelty here is the use of max-plus approximations combined with direct methods, which leads to a higher degree of accuracy. Indeed, under appropriate regularity assumptions, the method of [AGL23a] has a computional complexity of order $\mathcal{O}(\varepsilon^{-1-(d-1)(1-\beta)})$, in which the parameter $0 < \beta \leqslant 1$ measures the "stiffness" of the value functions near optimal trajectories. Typical instances are moderately stiff, and have a parameter $\beta = 1/2$, leading to a complexity of order $\mathcal{O}(\varepsilon^{-1-(d-1)/2})$. In contrast, we get here a complexity of order $\mathcal{O}(\varepsilon^{-\frac{1}{2}})$, with less demanding assumptions. The present method also allows one to address finite horizon problems with more general cost and dynamics structure.

This chapter is organized as follows: In Section 6.2, we give some preliminary results concerning the HJB equations and the finite horizon deterministic optimal control problems. We introduce the main technique of restricting the state space to a neighborhood of the optimal trajectory. In Section 6.3, we present the Max-Plus finite element method, and propose a novel approach that involves combining this method with the direct method to achieve a higher degree of accuracy. In Section 6.4.2, we give our new algorithm and demonstrate its convergence. The complexity estimates of the algorithm, under certain regularity conditions, is given in Section 6.6. Finally in Section 6.7, we present some numerical tests, confirming the theoretical estimation of the complexity. A shorter version of the present work, showing the first idea, has been presented in the proceedings of the IFAC World Congress 2023 [AGL23b].

6.2 Optimal control problem, hjb equation, characterization of optimal trajectories

6.2.1 The Optimal Control Problem.

We consider the finite horizon deterministic optimal control problem

$$\max \left\{ \int_0^T \ell(x(s), u(s))ds + \phi_0(x(0)) + \phi_T(x(T)) \right\} , \tag{6.1a}$$

where the maximum is taken over the set of trajectories $(x(s), u(s))$ satisfying:

$$\begin{cases} \dot{x}(s) = f(x(s), u(s)) , \\ x(s) \in X, \ u(s) \in U , \end{cases} \tag{6.1b}$$

for all $s \in [0, T]$. Let us denote v^* the maximum in (6.1). Here, $X \subset \mathbb{R}^d$, assumed to be bounded, is the state space and $U \subset \mathbb{R}^m$ is the control space. The functions $\phi_0, \phi_T : X \mapsto \mathbb{R}$ are the initial and final cost respectively. The function ℓ yields the running cost and f denotes the dynamics. We make the following regularity assumptions:

Assumption (A10)

i. $f : X \times U \to \mathbb{R}^d$ is bounded and Lipschitz continuous with respect to x, i.e.,

$$\begin{aligned} &\exists M_f > 0, s.t. \ \|f(x, u)\| \leqslant M_f, \ \forall x \in X, u \in U , \\ &\exists L_f > 0, s.t. \ \|f(x, u) - f(x', u)\| \leqslant L_f \|x - x'\|, \ \forall x, x' \in X, u \in U . \end{aligned}$$

ii. $\ell : X \times U \to \mathbb{R}$ is bounded and Lipschitz continuous with respect to x, i.e.,

$$\begin{aligned} &\exists M_\ell > 0, s.t. \ |\ell(x, u)| \leqslant M_\ell, \ \forall x \in X, u \in U , \\ &\exists L_\ell > 0, s.t. \ |\ell(x, u) - \ell(x', u)| \leqslant L_\ell |x - x'|, \ \forall x, x' \in X, u \in U . \end{aligned}$$

6.2.2 Optimality Conditions in Terms of HJB Equations

A well known sufficient and necessary optimality condition for the above problem is given by the Hamilton-Jacobi-Bellman equation, which is deduced from the dynamic programming principle. Indeed, we consider the value function $v_{\to \mathrm{d}}$, defined as follows, for any $(x, t) \in X \times [0, T]$:

$$v_{\to \mathrm{d}}(x, t) = \sup \left\{ \int_t^T \ell(x(s), u(s))ds + \phi_T(x(T)) \right\} , \tag{6.2}$$

under the constraint (6.1b) with the initial state $x(t) = x$. Here, the symbol "$_{\to \mathrm{d}}$" indicates that (x, t) is the *source*, so that the corresponding HJB PDE is of a backward nature. Indeed, $v_{\to \mathrm{d}}$ is known to be the viscosity solution of the following HJB equation (see for instance [FS06]):

$$\begin{cases} -\dfrac{\partial v_{\to \mathrm{d}}}{\partial t} - H(x, \nabla v_{\to \mathrm{d}}) = 0, & (x,t) \in X \times [0,T] \ , \\ v_{\to \mathrm{d}}(x,T) = \phi_T(x), & x \in X \ , \end{cases} \tag{6.3}$$

where $H(x,p) = \sup_{u \in U}\{p \cdot f(x,u) + \ell(x,u)\}$ is the Hamiltonian of the problem. Once (6.3) is solved, one can easily obtain the value of the original problem (6.1a) by further taking the maximum over X, i.e.,

$$v^* = \max_{x \in X}\{\phi_0(x) + v_{\to \mathrm{d}}(x,0)\} \ . \tag{6.4}$$

We shall also use another, equivalent, optimality condition for problem (6.1a), obtained by applying the dynamic programming principle in the reverse direction. This leads us to consider the value function $v_{\mathrm{s}\to}$, for any $(x,t) \in X \times [0,T]$, such that

$$v_{\mathrm{s}\to}(x,t) = \sup\left\{ \int_0^t \ell(x(s),u(s))ds + \phi_0(x(0)) \right\} \ , \tag{6.5}$$

under the same constraint (6.1b), but with the final state $x(t) = x$. The notation "$_{\mathrm{s}\to}$" indicates that (x,t) is now the destination. Then, $v_{\mathrm{s}\to}$ is known to be the viscosity solution of the following HJB equation, in forward time:

$$\begin{cases} \dfrac{\partial v_{\mathrm{s}\to}}{\partial t} - H(x, -\nabla v_{\mathrm{s}\to}) = 0, & (x,t) \in X \times [0,T] \ , \\ v_{\mathrm{s}\to}(x,0) = \phi_0(x), & x \in X \ . \end{cases} \tag{6.6}$$

Once (6.6) is solved, we can then get the maximum in (6.1) by

$$v^* = \max_{x \in X}\{\phi_T(x) + v_{\mathrm{s}\to}(x,T)\} \ . \tag{6.7}$$

6.3 Propagation by Lax-Oleinik Semi-Groups and Max-Plus Approximation

In this section, we first briefly recall the "max-plus finite element method" introduced in [AGL08] to solve the optimal control problem above, which is based on an approximation of the value function as a supremum of elementary "basis functions". Next, we will introduce a new method to solve the small time propagation problem of the basis functions used in the scheme. This method will lead to a higher degree of accuracy and will be a crucial step for our new algorithm.

In this section, we aim to approximate the value function $v_{\to \mathrm{d}}$ defined in (6.2), and to solve the associated HJB PDE (6.3) in a backward nature. Notice that the results and properties presented in this section hold, mutatis mutandis, for value function $v_{\mathrm{s}\to}$ defined in (6.5) and the evolution operator of the dual equation (6.6).

6.3.1 Max-Plus Variational Formulation

We denote by $S^t_{\to \mathrm{d}}$ the *Lax Oleinik semigroup* of (6.3), i.e., the evolution semigroup of this PDE, meaning that, for all $0 \leqslant t \leqslant T$, $S^t_{\to \mathrm{d}}$ is the map sending the final cost function $\phi_T(\cdot)$ to the value function $v_{\to \mathrm{d}}(\cdot, T-t)$, so that the semi-group property $S^{t_1+t_2} = S^{t_1} \circ S^{t_2}$ is satisfied. In

addition, the map $S^t_{\to \mathrm{d}}$ is *max-plus linear*, meaning that for all $\lambda \in \mathbb{R}$ and for all functions ϕ^1_T and $\phi^j_T : X \to \mathbb{R}$, we have:

$$\begin{aligned} S^t_{\to \mathrm{d}}[\sup(\phi^1_T, \phi^2_T)] &= \sup(S^t_{\to \mathrm{d}}[\phi^1_T], S^t_{\to \mathrm{d}}[\phi^2_T]) \ , \\ S^t_{\to \mathrm{d}}[\lambda + \phi^1_T] &= \lambda + S^t_{\to \mathrm{d}}[\phi^1_T] \ , \end{aligned} \tag{6.8}$$

where for any function ϕ on X, $\lambda + \phi$ is the function $x \in X \mapsto \lambda + \phi(x)$. Indeed, the property (6.8) can be interpreted as the linearity in the sense of the max-plus semifield, which is the set $\mathbb{R}_{\max} := \mathbb{R} \cup \{-\infty\}$ equipped with the addition $a \oplus b := \max(a, b)$ and the multiplication $a \odot b := a + b$, with $-\infty$ as the zero and 0 as the unit. We refer the reader to [FM00], [AGL08], and [YD21b] for more information.

6.3.2 Max-Plus Approximation Method

We will briefly describe the approximation method based on the max-plus linearity introduced in [AGL08], which may be thought of as a max-plus analogue of the finite element methods.

Let us discretize the time horizon by $N = \frac{T}{\delta}$ steps. Denote $v^t_{\to \mathrm{d}} = v_{\to \mathrm{d}}(\cdot, t)$. By the semigroup property we have:

$$v^{t-\delta}_{\to \mathrm{d}} = S^\delta_{\to \mathrm{d}}[v^t_{\to \mathrm{d}}], \ \forall\ t = \delta, 2\delta, \dots, T \ , \quad v^T = \phi_T \ . \tag{6.9}$$

Denote $\overline{\mathbb{R}}_{\max} := \mathbb{R}_{\max} \cup \{+\infty\}$ the complete semiring extending $\mathbb{R}_{\max}$, and let $\mathcal{W}$ be a complete $\mathbb{R}_{\max}$-semimodule of functions $w : X \to \overline{\mathbb{R}}_{\max}$, meaning that $\mathcal{W}$ is stable under taking the supremum of an arbitrary family of functions, and by the addition of a constant, see [McE06; CGQ04] for background. We choose this semimodule $\mathcal{W}$ in such a way that $v^t_{\to \mathrm{d}} \in \mathcal{W}$ for all $t \geqslant 0$. In many applications, the value function v^t is known to be c-semiconcave for all $t \in [0, T]$, and then $\mathcal{W}$ can be taken to be the set of c-semiconcave functions, which is a complete module, see [McE06; AGL08]. We also choose $\mathcal{Z}$, a complete $\mathbb{R}_{\max}$-semimodule of test functions $z : X \mapsto \overline{\mathbb{R}}_{\max}$. If the space of test functions $\mathcal{Z}$ is large enough, (6.9) is equivalent to:

$$\langle z, v^{t-\delta}_{\to \mathrm{d}} \rangle = \langle z, S^\delta_{\to \mathrm{d}}[v^t_{\to \mathrm{d}}] \rangle \ \forall t, \ \langle z, v^T \rangle = \langle z, \phi_T \rangle \ \forall z \in \mathcal{Z} \ , \tag{6.10}$$

where the max-plus scalar product of $u \in \mathcal{W}$ and $v \in \mathcal{Z}$ is defined by $\langle u, v \rangle = \sup_{x \in X}(u(x) + v(x)) \in \overline{\mathbb{R}}_{\max}$.

Note that in the system (6.10), the unknown value functions are elements of $\mathcal{W}$, therefore having an infinite number of degrees of freedom, and that there are infinitely many equations (one for each element $z \in \mathcal{Z}$). Hence, we need to discretize this system. To do so, we consider $\mathcal{W}^h \subset \mathcal{W}$, a semimodule generated by a finite family of basis functions $\{w_i\}_{1 \leqslant i \leqslant p}$. The value function $v^t_{\to \mathrm{d}}$ at time t is approximated by $v^{t,h}_{\to \mathrm{d}} \in \mathcal{W}^h$, that is:

$$v^{t,h}_{\to \mathrm{d}} := \sup_{1 \leqslant i \leqslant p} \{\lambda^t_i + w_i\} \ : \ x \mapsto \max_{1 \leqslant i \leqslant p} \{\lambda^t_i + w_i(x)\} \ , \tag{6.11}$$

where $\{\lambda^t_i\}_{1 \leqslant i \leqslant p}$ is a family of scalars. We then consider $\mathcal{Z}^h \subset \mathcal{Z}$, a semimodule generated by a finite family of test functions $\{z_j\}_{1 \leqslant j \leqslant q}$, and, instead of requiring (6.10) to hold for all $z \in \mathcal{Z}$, we only require that it holds for generators, leading to a finite system of equations. Therefore, the approximation $v^{t-\delta,h}_{\to \mathrm{d}}$ and $v^T_{\to \mathrm{d}}$ should satisfy:

$$\langle z_j, v^{t-\delta,h}_{\to \mathrm{d}} \rangle = \langle z_j, S^\delta_{\to \mathrm{d}}[v^{t,h}_{\to \mathrm{d}}] \rangle, \ \langle z_j, v^T_{\to \mathrm{d}} \rangle = \langle z_j, \phi_T \rangle, \ \forall j \ . \tag{6.12}$$

It is a key property of max-plus algebra that a system of linear equations, even when the number of equations coincides with the number of degrees of freedom, and when the system is "nonsingular", may have no solution, so that the notion of solution must be replaced by a

notion of maximal subsolution, which is always well posed. In particular, (6.12) may not have a solution. Hence, we define $v_{\to \mathrm{d}}^{t-\delta,h}$ and $v_{\to \mathrm{d}}^{T}$ to be the maximal solution of the following system of inequalities:

$$\langle z_j, v_{\to \mathrm{d}}^{t-\delta,h}\rangle \leqslant \langle z_j, S_{\to \mathrm{d}}^{\delta}[v_{\to \mathrm{d}}^{t,h}]\rangle,\ \langle z_j, v_{\to \mathrm{d}}^{T}\rangle \leqslant \langle z_j, \phi_T\rangle,\ \forall j\ . \tag{6.13}$$

Let us denote $W_h : \mathbb{R}_{\max}^p \mapsto \mathcal{W}$ the max-plus linear operator such that $W_h(\lambda) = \oplus_{1\leqslant i\leqslant p}\{\lambda_i \odot w_i\}$, and $Z_h^* : \mathcal{W} \mapsto \mathbb{R}_{\max}^q$ with $(Z_h^*(w))_j = \langle z_j, w\rangle, \forall 1 \leqslant j \leqslant q$. Recall that, for every ordered sets S, T and order preserving map $g : S \mapsto T$, the residuated map $g^{\#}$ is defined as $g^{\#}(t) = \max\{s \in S \mid g(s) \leqslant t\}$, when it exists. Max-plus linear operators have a residuated map. Moreover, by [CGQ96, Th. 1], for all max-plus linear operators $B : \mathcal{U} \mapsto \mathcal{X}$, $C : \mathcal{X} \mapsto \mathcal{Y}$ over complete semimodules $\mathcal{X}, \mathcal{Y}, \mathcal{U}$, the operator $\Pi_B^C := B \circ (C \circ B)^{\#} \circ C$ is a projector, and we have, for all $x \in \mathcal{X}$:

$$\Pi_B^C(x) = \max\{y \in \mathrm{im}B \mid Cy \leqslant Cx\}\ . \tag{6.14}$$

Then, the approximations $v_{\to \mathrm{d}}^{t,h}$ can be expressed as follows.

Proposition 6.3.1 ([AGL08])**.** *Consider the maximal $\lambda^t \in \mathbb{R}_{\max}^p$ and $v_{\to \mathrm{d}}^{t,h} \in \mathcal{W}_h$, $t = 0, \delta, \ldots, T$, such that $v_{\to \mathrm{d}}^{t,h} = W_h\lambda^t$, with $v_{\to \mathrm{d}}^{t-\delta,h}$, $t \geqslant \delta$, and v^T solutions of* (6.13)*. We have,*

$$v_{\to \mathrm{d}}^{t-\delta,h} = S_{\to \mathrm{d}}^{\delta,h}[v_{\to \mathrm{d}}^{t,h}],\ \textit{where}\ S_{\to \mathrm{d}}^{\delta,h} = \Pi_{W_h}^{Z_h^*} \circ S_{\to \mathrm{d}}^{\delta}\ , \tag{6.15a}$$

and

$$\begin{cases} \lambda^{t-\delta} = (Z_h^*W_h)^{\#}(Z_h^*S_{\to \mathrm{d}}^{\delta}W_h\lambda^t),\ \forall t = \delta, 2\delta, \ldots, T\ , \\ \lambda^T = W_h^{\#}\phi_T\ . \end{cases} \tag{6.15b}$$

The above formula can be expressed using the linear operators $M_h := Z_h^*W_h$ and $K_h := Z_h^*S_{\to \mathrm{d}}^{\delta}W_h$, with entries:

$$(M_h)_{j,i} = \langle z_j, w_i\rangle\ , \quad (K_h)_{j,i} = \langle z_j, S_{\to \mathrm{d}}^{\delta}w_i\rangle\ . \tag{6.16}$$

The matrices M_h and K_h may be thought of as max-plus analogues of the *mass* and *stiffness* matrices arising in the finite element method, see [AGL08]. Computing $(M_h)_{j,i}$ is a convex programming problem, which can be solved by standard optimization methods (sometimes the solution can even be computed analytically). The main difficulty here is to compute $(K_h)_{j,i}$. An approximation method proposed in [AGL08] is to use the Hamiltonian of the problem, that is, when w_i is differentiable,

$$(K_h)_{j,i} \approx (K_{H,h})(j,i) := \sup_{x\in X}\Big(z_j(x) + w_i(x) + \delta H(x, \nabla w_i(x))\Big)\ , \tag{6.17a}$$

or, when w_i is nondifferentiable but z_j is differentiable,

$$(K_h)_{j,i} \approx (K_{H,h})(j,i) := \sup_{x\in X}\Big(z_j(x) + w_i(x) + \delta H(x, -\nabla z_j(x))\Big)\ . \tag{6.17b}$$

Both the approximation in (6.17a) and (6.17b) introduce an error $\mathcal{O}(\delta^2)$ or $\mathcal{O}(\delta^{\frac{3}{2}})$, depending the properties of z_i and w_i. In the following, we will propose a new approximation method for a small enough time horizon δ to get a high degree of accuracy, and thus avoid this approximation error

6.3.3 Small Time Propagation of Basis Functions

As mentioned above, a key point to get an effective max-plus method is to compute K_h, which is equivalent to evaluate every scalar product $\langle z_j, S^\delta_{\to \mathrm{d}} w_i \rangle$. This small time propagation of the basis functions leads to a new optimal control problem:

$$\begin{aligned} &\langle z_j, S^\delta[w_i] \rangle = \\ &\max \left\{ z_j(x(0)) + \int_0^\delta \ell(x(s), u(s)) ds + w_i(x(\delta)) \right\} , \end{aligned} \tag{6.18}$$

over the set of trajectories $(x(s), u(s))$ satisfying (6.1b). This problem is similar to the original one, but with two new essential properties: first, the time horizon δ is *small*, and second, the initial and final costs, z_j and w_i, are "nice" concave functions, e.g., strongly concave quadratic forms. Then, the strong convexity of the initial or terminal cost "propagates" over a small horizon, which entails that (6.18) is actually a *convex* infinite dimensional optimization problem, which, after an appropriate discretization, using a so-called *direct method* in optimal control (see e.g. [Bon+17] for background on direct methods in optimal control.), can be reduced to a convex finite dimensional optimization problem, which can be solved *globally* by convex optimization methods. This is explained in [AGL05], in a simple case, for which (6.18) is approximated by one step semi-lagrangian type discretization. The idea of applying direct methods was recently used in [BB20], in which the authors used a gradient descent to do one step computation of a discrete MDP with a finite set of controls. Based on those observations, we have the following result:

Proposition 6.3.2. *Assume the functions z_i, w_i are strongly concave, then there exists a $\bar\delta > 0$ such that, for every $\delta \leqslant \bar\delta$, $\langle z_i, S^\delta[w_i] \rangle$ can be computed exactly, or with an error negligible compared with the projection error, by a direct method.*

To implement efficiently our algorithm, we need to know in advance the lower bound of $\bar\delta$, which depends on the properties of f and ℓ. Thus, inspired by [AGL05], we shall make the following assumption:

Assumption (A11)

i. $X \subset \mathbb{R}^d$, $U \subset \mathbb{R}^m$ are convex sets.

ii. f is affine w.r.t. x and u.

iii. $\ell \in \mathcal{C}^2(X \times U, \mathbb{R})$ and ℓ is strongly concave w.r.t. u, i.e., there exists a positive constant $\alpha > 0$ such that $\|\frac{\partial^2 \ell}{\partial u^2}\| \geqslant \alpha$. Moreover, there exist positive constants $C_{xx} > 0$ and $C_{xu} > 0$ such that $\|\frac{\partial^2 \ell}{\partial x^2}\| \leqslant C_{xx}$ and $\|\frac{\partial^2 \ell}{\partial x \partial u}\| \leqslant C_{xu}$, for every $(x, u) \in X \times U$.

Lemma 6.3.3. *Assume Assumption (A11), take $w_i, z_j \in \mathcal{C}^2(\mathbb{R}^d, \mathbb{R})$ be strongly concave functions with $\|\frac{\partial^2 w_i}{\partial x^2}\|, \|\frac{\partial^2 z_j}{\partial x^2}\| \geqslant \beta > 0$, denote $A = \frac{\partial f}{\partial x}$ and $B = \frac{\partial f}{\partial u}$. Assume further*

$$\alpha \|A\|^2 - 2C_{xu}\|A\|\|B\| - C_{xx}\|B\|^2 \geqslant 0 , \tag{6.19}$$

then Proposition 6.3.2 holds with $\bar\delta = \bar\delta(\alpha, \beta, C_{xx}, \|A\|, \|B\|)$, with

$$\bar\delta(\alpha, \beta, C_{xx}, \|A\|, \|B\|) = \frac{8\alpha\beta\lambda\|B\|^2}{(\alpha\|A\|^2 + C_{xx}\|B\|^2)(\alpha\|A\|^2 + C_{xx}\|B\|^2 + 2\beta\|A\|\|B\|^2)} . \tag{6.20}$$

Proof. Let us denote

$$J^t_{s\to}(x(\cdot),u(\cdot)) = z(x(0)) + \int_0^t \ell(x(s),u(s))ds \ ,$$

over the set of trajectories $(x(s),u(s))$ satisfying (6.1b). Then, it is sufficient to show that up to $t=\bar{\delta}$, $J^t_{s\to}$ is concave w.r.t. $(x(\cdot),u(\cdot))$. Let us consider two trajectories $(x_1(\cdot),u_1(\cdot))$ and $(x_2(\cdot),u_2(\cdot))$, we need to show that

$$J^t_{s\to}(\frac{x_1+x_2}{2}(\cdot),\frac{u_1+u_2}{2}(\cdot)) - \frac{1}{2}\Big(J^t_{s\to}(x_1(\cdot),u_1(\cdot)) + J^t_{s\to}(x_2(\cdot),u_2(\cdot))\Big) \geqslant 0 \ . \tag{6.21}$$

We first notice that, under the Assumption (A11), we have

$$\dot{x}_1(s) - \dot{x}_2(s) = A(x_1(s)-x_2(s)) + B(u_1(s)-u_2(s)) \ , \tag{6.22}$$

for every $s \geqslant 0$. Moreover, we have

$$\begin{aligned}
&J^t_{s\to}(\frac{x_1+x_2}{2}(\cdot),\frac{u_1+u_2}{2}(\cdot)) - \frac{1}{2}\Big(J^t_{s\to}(x_1(\cdot),u_1(\cdot)) + J^t_{s\to}(x_2(\cdot),u_2(\cdot))\Big)\\
&= \Big[z(\frac{x_1+x_2}{2}(0)) - \frac{1}{2}(z(x_1(0))+z(x_2(0)))\Big]\\
&\quad + \Big[\int_0^t \Big(\ell(\frac{x_1+x_2}{2}(s),\frac{u_1+u_2}{2}(s)) - \frac{1}{2}(\ell(x_1(s),u_1(s))ds + \ell(x_2(s),u_2(s)))\Big)ds\Big] \ .
\end{aligned} \tag{6.23}$$

For the first part inside of [ ], denoted by Δ_1, by the strong concavity of w we have

$$\Delta_1 \geqslant \frac{\beta}{4}(\|x_1(0)-x_2(0)\|^2) \tag{6.24}$$

For the second part inside of [ ], denoted by Δ_2, for easy expression let us denote $\Delta u(s) = u_1(s)-u_2(s)$, $\Delta x(s) = x_1(s)-x_2(s)$. We first notice that

$$\begin{aligned}
&\ell(\frac{x_1+x_2}{2}(s),\frac{u_1+u_2}{2}(s)) - \frac{1}{2}(\ell(x_1(s),u_1(s)) + \ell(x_2(s),u_2(s)))\\
&= \frac{1}{2}\Big(2\ell(\frac{x_1+x_2}{2}(s),\frac{u_1+u_2}{2}(s)) - \ell(\frac{x_1+x_2}{2}(s),u_1(s)) - \ell(\frac{x_1+x_2}{2}(s),u_2(s))\\
&\quad + \ell(\frac{x_1+x_2}{2}(s),u_1(s)) - \ell(x_1(s),u_1(s)) + \ell(\frac{x_1+x_2}{2}(s),u_2(s)) - \ell(x_2(s),u_2(s))\Big)\\
&\geqslant \alpha\|\frac{\Delta u(s)}{2}\|^2 + \frac{1}{2}\Big((\frac{\Delta x(s)}{2})^T\frac{\partial \ell}{\partial x}(\frac{x_1+x_2}{2}(s),u_1(s)) - C_{xx}\|\frac{\Delta x(s)}{2}\|^2\\
&\quad + (\frac{-\Delta x(s)}{2})^T\frac{\partial \ell}{\partial x}(\frac{x_1+x_2}{2}(s),u_2(s)) - C_{xx}\|\frac{\Delta x(s)}{2}\|^2\Big)\\
&\geqslant \frac{\alpha}{4}\|\Delta u(s)\|^2 - \frac{C_{xx}}{4}\|\Delta x(s)\|^2 - \frac{C_{xu}}{4}\|\Delta x(s)\|\|\Delta u(s)\| \ .
\end{aligned} \tag{6.25}$$

Moreover, for arbitrary $0<\theta_1<\alpha$ and $0<\theta_2$, we have

$$\begin{aligned}
&\frac{\alpha}{4}\|\Delta u(s)\|^2 - \frac{C_{xx}}{4}\|\Delta x(s)\|^2 - \frac{C_{xu}}{4}\|\Delta x(s)\|\|\Delta u(s)\|\\
&\geqslant \frac{1}{4}\Big(\frac{(\alpha-\theta_1)}{\|B\|^2}\|A\Delta x(s)+B\Delta u(s)\|^2\\
&\quad - \frac{1}{2\|A\|}(\frac{(\alpha-\theta_1)\|A\|^2}{\|B\|^2} + C_{xx} + \theta_2)2\Delta x(s)^T(A\Delta x(s)+B\Delta u(s))\\
&\quad + \Big[\theta_1\|\Delta u(s)\|^2 + \theta_2\|\Delta x(s)\|^2\\
&\quad - (\frac{(\alpha-\theta_1)\|A\|}{\|B\|} + (C_{xx}+\theta_2)\|B\|) + C_{xu})\|\Delta x(s)\|\|\Delta u(s)\|\Big]\Big) \ .
\end{aligned} \tag{6.26}$$

For the part inside of [ ] of (6.26), we recognize a quadratic form in the variables $\|\Delta x(s)\|$ and $\|\Delta u(s)\|$. This quadratic form will keep a constant sign if its discriminant is negative, and this is the case in particular if we take $\theta_1 = \theta_2 = \frac{1}{2}\alpha$. Moreover, we observe that

$$\|\Delta\dot{x}(s)\|^2 = \|A\Delta x(s) + B\Delta u(s)\|^2,\ (\|\Delta x(s)\|^2)' = 2\Delta x(s)^T(A\Delta x(s) + B\Delta u(s))\ . \tag{6.27}$$

Thus, taking θ_1 and θ_2 as proposed above, combining with (6.26), we have

$$\begin{aligned}
\Delta_2 &\geqslant \int_0^t (\frac{\alpha}{4}\|\Delta u(s)\|^2 - \frac{C_{xx}}{4}\|\Delta x(s)\|^2 - \frac{C_{xu}}{2}\|\Delta x(s)\|\|\Delta u(s)\|)ds \\
&\geqslant \int_0^t \Big(\frac{(\alpha-\theta_1)}{4\|B\|^2}\|\Delta\dot{x}(s)\|^2 - \frac{(\alpha-\theta_1)\|A\|^2 + (C_{xx}+\theta_2)\|B\|^2}{8\|A\|\|B\|^2}(\|\Delta x(s)\|^2)' \\
&\geqslant \frac{(\alpha-\theta_1)}{4\|B\|^2 t}\|\int_0^t \Delta\dot{x}(s)ds\|^2 - \frac{(\alpha-\theta_1)\|A\|^2 + (C_{xx}+\theta_2)\|B\|^2}{8\|A\|\|B\|^2}(\|\Delta x(t)\|^2 - \|\Delta x(0)\|^2) \\
&\geqslant \frac{(\alpha-\theta_1)}{4\|B\|^2 t}\|\Delta x(t) - \Delta x(0)\|^2 - \frac{(\alpha-\theta_1)\|A\|^2 + (C_{xx}+\theta_2)\|B\|^2}{8\|A\|\|B\|^2}(\|\Delta x(t)\|^2 - \|\Delta x(0)\|^2)\ .
\end{aligned} \tag{6.28}$$

Combining (6.24) and (6.28), we have

$$\begin{aligned}
&J_{s\to}(\frac{x_1+x_2}{2}(\cdot), \frac{u_1+u_2}{2}(\cdot)) - \frac{1}{2}\Big(J_{s\to}(x_1(\cdot),u_1(\cdot)) + J_{s\to}(x_2(\cdot),u_2(\cdot))\Big) \\
&\geqslant \frac{1}{4}\Big((\beta + \frac{(\alpha-\theta_1)}{\|B\|^2 t} + \frac{(\alpha-\theta_1)\|A\|^2 + (C_{xx}+\theta_2)\|B\|^2}{2\|A\|\|B\|^2})\|\Delta x(0)\|^2 \\
&\quad + (\frac{(\alpha-\theta_1)}{\|B\|^2 t} - \frac{(\alpha-\theta_1)\|A\|^2 + (C_{xx}+\theta_2)\|B\|^2}{2\|A\|\|B\|^2})\|\Delta x(t)\|^2 \\
&\quad - \frac{2(\alpha-\theta_1)}{\|B\|^2 t}\|\Delta x(t)\|\|\Delta x(0)\|\Big)\ .
\end{aligned} \tag{6.29}$$

We again recognize a quadratic form in the variables $\Delta x(0)$ and $\Delta x(t)$. This quadratic form will keep positive, for every non zero $\Delta x(0)$ and $\Delta x(t)$, if we have:

$$\frac{(\alpha-\theta_1)}{\|B\|^2 t} - \frac{(\alpha-\theta_1)\|A\|^2 + (C_{xx}+\theta_2)\|B\|^2}{2\|A\|\|B\|^2} > 0\ , \tag{6.30}$$

and

$$\begin{aligned}
(\frac{2(\alpha-\theta_1)}{\|B\|^2 t})^2 - 4(\beta + \frac{(\alpha-\theta_1)}{\|B\|^2 t} + \frac{(\alpha-\theta_1)\|A\|^2 + (C_{xx}+\theta_2)\|B\|^2}{2\|A\|\|B\|^2}) \\
(\frac{(\alpha-\theta_1)}{\|B\|^2 t} - \frac{(\alpha-\theta_1)\|A\|^2 + (C_{xx}+\theta_2)\|B\|^2}{2\|A\|\|B\|^2}) < 0\ .
\end{aligned} \tag{6.31}$$

Combing (6.30) and (6.31), we have

$$t \leqslant \frac{4(\alpha-\theta_1)\beta\|A\|\|B\|^2}{((\alpha-\theta_1)\|A\|^2 + (C_{xx}+\theta_2)\|B\|^2)((\alpha-\theta_1)\|A\|^2 + (C_{xx}+\theta_2)\|B\|^2 + 2\beta\|A\|\|B\|^2)}\ . \tag{6.32}$$

Since the propagation of basis functions in two directions are symmetric, take θ_1 and θ_2 in particular equal to $\frac{\alpha}{2}$, we deduce

$$\bar{\delta}(\alpha,\beta,C_{xx},\|A\|,\|B\|) = \frac{8\alpha\beta\lambda\|B\|^2}{(\alpha\|A\|^2 + (2C_{xx}+\alpha)\|B\|^2)(\alpha\|A\|^2 + (2C_{xx}+\alpha)\|B\|^2 + 4\beta\|A\|\|B\|^2)}\ . \tag{6.33}$$

□

We shall take the time step $\delta \leqslant \bar{\delta}$, then each element $(K_h)_{j,i}$ can be approximated by using direct methods computing the scalar product $\langle z_j, S^\delta[w_i]\rangle$.

6.3.4 Improved Max-Plus Finite Element Method and Error Estimation

In this section, we will always make the following assumption on the value functions $v_{\mathrm{s}\to}$ and $v_{\to\mathrm{d}}$.

Assumption (A12) The functions $v^t_{\mathrm{s}\to}, v^t_{\to\mathrm{d}}$ are L_v-Lipschitz continuous, α_1-semiconvex and α_2-semiconcave w.r.t. x for every $t \in [0,T]$.

Let G, $\hat{G}$ be two finite subsets of $\mathbb{R}^d$. After a time discretization of $N = \frac{T}{\delta}$ steps, in order to compute the approximation of value function, we shall take the basis functions generated by the points of $\hat{G} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_p\}$, and take the tests functions generated by the points of $G = \{x_1, x_2, \dots, x_q\}$. Under Assumption (A12), natural choices for basis functions and test functions to approximate the value functions are the quadratic functions of the form $w_{\hat{x}_i}(x) := -\frac{c}{2}\|x - \hat{x}_i\|_2^2$, for every $\hat{x}_i \in \hat{G}$, and $z_{x_i} = -\frac{c}{2}\|x - x_i\|_2^2$, for every $x_i \in G$.

The matrix M_h is computed by solving a convex programming problem. The entries of matrix K_h are approximated, under Assumption (A11), by using direct methods. This means that we replace the infinite dimensional convex programming problem (6.18) by a finite dimensional one, in particular using a further discretization in time. An approximation can then be obtained up to a precision $\epsilon \ll \delta$. The complete algorithm works as in Algorithm 6.1.

Algorithm 6.1 Max-Plus Approximation Method

Input: Mesh grid: $\hat{G}$ and G, parameter for quadratic basis/test functions: c, time step: $\delta \leqslant \bar{\delta}$, precision of direct method $\epsilon \ll \delta$.

Output: $\forall t \in \{0, \delta, \dots, T\}$, set of scalars $\{\lambda_i^t\}_{1 \leqslant i \leqslant p}$.

1: Discretize time horizon by $N = \frac{T}{\delta}$ steps.
2: Choose quadratic basis functions $\{w_i\}_{\hat{x}_i \in \hat{G}}$ generate by $\hat{G}$.
3: Choose quadratic test functions $\{z_j\}_{x_j \in G}$ generated by G.
4: Compute $(M_h)_{j,i} = \langle z_j, w_i \rangle$, $\forall\ (j,i) \in \{1,\dots,q\} \times \{1,\dots,p\}$.
5: Approximate $(K_h)_{j,i} = \langle z_j, S^\delta[w_i] \rangle$ by direct method up to an error ϵ, $\forall\ (j,i) \in \{1,\dots,q\} \times \{1,\dots,p\}$.
6: Initialize $\lambda^T = W_h^{\#} \phi_T$.
7: **for** t = T, T-δ,...,δ **do**
8: $\quad \lambda^{t-\delta} = M_h^{\#} K_h \lambda^t$.
9: **end for**

Let us denote by $\tilde{v}^h_{\to\mathrm{d}}$ the approximate value function obtained using quadratic basis functions together with scalars computed in Algorithm 6.1. We then have the following result.

Theorem 6.3.4. *Make Assumption (A10), Assumption (A11), Assumption (A12). Denote $\hat{X} = X + B(0, \frac{L_v}{c})$. Assume $\alpha_1 < c$, $\alpha_2 < c$ and $\epsilon = C_0 h^2$, $\delta \leqslant \bar{\delta}$. Take $G, \hat{G}$ such that $\hat{X} \subset \mathrm{Conv}(\hat{G})$ and $\hat{X} \subset \mathrm{Conv}(\hat{G})$, where $\mathrm{Conv}(\cdot)$ denotes the convex hull of a subset of $\mathbb{R}^d$. Then, there exists a constant $C > 0$ depending on $c - \alpha_1$, $c - \alpha_2$ and C_0 such that:*

$$\|\tilde{v}^h_{\to\mathrm{d}}(t,\cdot) - v_{\to\mathrm{d}}(t,\cdot)\|_\infty \leqslant \frac{C}{\delta} h^2 \ , \tag{6.34}$$

where h is the maximum radius of the Voronoi cells of the space $\hat{X}$ divided by the points of $\hat{G}$ or G.

Sketch of Proof. Under Assumption (A11), we are allowed to use a direct method with $\delta \leqslant \bar{\delta}$ and get a propagation error less than ϵ. By [Lak07, Coro. 66], the projection error is bounded

by Ch^2, for some constant C_1 depending on $c-\alpha_1$ and $c-\alpha_2$. Then, by [AGL08] (or [Lak07]), the total error is bounded as in (6.34) with $C = C_1 + C_2$. □

Remark 6.3.5. When the value function is only known to be semiconvex (or semiconcave, respectively), we shall employ quadratic basis functions and Lipschitz test functions (Lipschitz basis functions and quadratic tests functions, respectively). In such cases, the maximal time step to use in the direct method is $\frac{\bar{\delta}}{2}$ (by the proof of (6.3.3)), and the error is then in the order of h in (6.34). We refer to [Lak07] for a more detailed analysis of the error estimation of the max-plus finite element method.

6.4 Characterization and Max-plus approximation of optimal trajectories

In Section 6.2, the optimality conditions of problem (6.1) are characterized by the HJB equation in two directions. In this section, we will show that the optimal trajectories of problem (6.1) can be characterized by two value functions, which are the solutions of the associated HJB equations. Then, combining with the max-plus approximation method in (6.3), the optimal trajectories will be approximated using the scalars computed at every time step for the value functions.

6.4.1 Optimal and $\delta-$optimal Trajectories

The two value functions $v_{\mathrm{s}\to}$ and $v_{\to\mathrm{d}}$ allow us to determine the points belonging to optimal trajectories:

Definition 6.4.1. We say that $x^*(\cdot)$ is an optimal trajectory of the optimal control problem (6.1) if there exists a control $u^*(\cdot)$ such that $(x^*(\cdot), u^*(\cdot))$ achieves the maximum in (6.1a), under the constraint (6.1b). Moreover, we denote, for all $t \in [0,T]$,

$$\Gamma_t^* = \{x^*(t) \mid \ x^*(\cdot) \text{ is an optimal trajectory } \} \ , \tag{6.35}$$

and $\Gamma^* = \cup_{t\in[0,T]}\Gamma_t^*$.

Then, we have the following result:

Proposition 6.4.2. *Assume Γ^* is non-empty, then*

$$v^* = \sup_{x\in X}\{v_{\mathrm{s}\to}(x,t) + v_{\to\mathrm{d}}(x,t)\}, \ \forall t \in [0,T] \ . \tag{6.36}$$

Moreover, for all $t \in [0,T]$, the above supremum is achieved for some $x \in \Gamma_t^*$. Conversely, for all $t \in [0,T]$ and $x \in \Gamma_t^*$, the above supremum is achieved at point x.

Proof. The equality (6.36) follows in a straightforward way from the definition of the value functions $v_{\to\mathrm{d}}, v_{\mathrm{s}\to}$ in (6.2) and (6.5). Moreover, since there exists an optimal trajectory $x^*(\cdot)$, then the supremum in (6.36) is achieved at $x^*(t) \in \Gamma_t^*$, for all $t \in [0,T]$. Conversely, for all $x \in \Gamma_t^*$, there exists an optimal trajectory x^* such that $x^*(t) = x$, and the supremum in (6.36) is achieved at $x = x^*(t)$. □

For all $t \in [0,T]$, let us define the map $\mathcal{F}_v^t : X \mapsto \mathbb{R}$ by

$$\mathcal{F}_v^t(x) := v_{\mathrm{s}\to}(x,t) + v_{\to\mathrm{d}}(x,t) \ . \tag{6.37}$$

Consider for every $t \in [0,T]$, the subdomain $\mathcal{O}_\eta^t \subset X$, depending on a parameter $\eta > 0$, and defined as follows

$$\mathcal{O}_\eta^t = \{x \in X \mid \mathcal{F}_v^t(x) > \sup_{y\in X} \mathcal{F}_v^t(y) - \eta\} \ . \tag{6.38}$$

In fact, $\mathcal{O}_\eta^t$ can be thought of as a $\eta-$neighborhood of Γ_t^*, which is the optimal trajectory at time t. We intend to reduce the (state,time)-space $X \times [0,T]$ of our optimal control problem to such an $\eta-$neighborhood $\{(x,t) \mid x \in \mathcal{O}_\eta^t\}$. I.e., for all $s \in [0,T]$, we replace the constraint (6.1b) by

$$\begin{cases} \dot{x}(s) = f(x(s), u(s)) \ , \\ x(s) \in \mathcal{O}_\eta^s, \ u(s) \in U \ . \end{cases} \tag{6.39}$$

For any $(x,t) \in X \times [0,T]$, let us denote $v_{\to \mathrm{d}}^\eta$, $v_{\mathrm{s}\to}^\eta$ the value functions of the optimal control problems in backward and forward directions under the new constraint (6.39) respectively, i.e., the value of (6.2) and (6.5) under the constraint (6.39) respectively. Moreover, we denote $S_{\mathrm{s}\to}^{t,\eta}$ and $S_{\to \mathrm{d}}^{t,\eta}$ the propagation semi-group for the two functions, respectively. Denote v_η^* the maximum of (6.1a) under the new constraint (6.39). Then we have

Proposition 6.4.3. $v_\eta^* = v^*$.

Proof. The inequality $v^* \geqslant v_\eta^*$ is straightforward since $\mathcal{O}_\eta^s \subset X$ for all $s \in [0,T]$. To show the reverse inequality, let us take an optimal trajectory $x^*(\cdot)$ for the original problem. Then, by the result of Proposition 6.4.2, we have $x^*(s) \in \mathcal{O}_\eta^s, \forall s \in [0,T]$. Thus $v_\eta^* \geqslant v^*$ since v^* is exactly the value of the integral in (6.1a) following the optimal trajectory $x^*(\cdot)$. □

We shall also consider approximate, $\delta-$optimal, trajectories.

Definition 6.4.4. We say that $x^\delta(\cdot)$ is a δ-optimal trajectory of the optimal control problem (6.1a) if there exists a control $u^\delta(\cdot)$ such that

$$\int_0^T \ell(x^\delta(s), u^\delta(s))ds + \phi_0(x^\delta(0)) + \phi_T(x^\delta(T)) \geqslant v^* - \delta \ , \tag{6.40}$$

under the constraint (6.1b). Moreover, we denote, for all $t \in [0,T]$,

$$\Gamma_\delta^t = \{x^\delta(t) \mid x^\delta(\cdot) \text{ is a } \delta\text{-optimal trajectory}\} \ , \tag{6.41}$$

and $\Gamma_\delta = \cup_{t\in[0,T]} \Gamma_\delta^t$.

We begin by showing the connection between the $\delta-$optimal trajectories, Γ_δ^t, and our $\eta-$neighborhood $\mathcal{O}_\eta^t$.

Proposition 6.4.5. *The set of δ-optimal trajectories and $\mathcal{O}_\eta^t$ constitute two equivalent families of neighborhoods of optimal trajectories, in the sense that for every $\eta > 0$ and $\delta' > 0$, for every $t \in [0,T]$,*

$$\mathcal{O}_\eta^t \subset \Gamma_{\eta+\delta'}^t, \quad \Gamma_\eta^t \subset \mathcal{O}_{\eta+\delta'}^t \ . \tag{6.42}$$

Proof. For the first inclusion, let us consider a $x \in \mathcal{O}_\eta$. It is sufficient to show that there exists at least a $\eta+\delta-$optimal trajectory for the problem (6.1) passing through x at time t. We can consider a $\frac{\delta'}{2}-$optimal trajectory $x_{\mathrm{s}\to}(\cdot) : [0,t] \to X$ for the problem in direction "from source", with final state $x_{\mathrm{s}\to}(t) = x$, together with a $\frac{\delta'}{2}-$optimal trajectory $x_{\to \mathrm{d}}(\cdot) : [t,T] \to X$ for the problem in direction "to destination", with initial state $x_{\to \mathrm{d}}(t) = x$. Thus, concatenating the two trajectories, we obtain a $\eta+\delta'-$optimal trajectories for the problem (6.1).

For the second inclusion, consider an arbitrary $\eta-$optimal trajectory $x^\eta(\cdot) : [0,T] \to X$ together with the control $u^\eta(\cdot) : [0,T] \to U$. For every $t \in [0,T]$, let us denote $x_t^\eta = x^\eta(t) \in \Gamma_\eta^t$. By the definition of the value function, we have

$$v_{\mathrm{s}\to}(x_t^\eta, t) \geqslant \int_0^t \ell(x^\eta(s), u^\eta(s))ds + \phi_0(x^\eta(0)), \quad v_{\to \mathrm{d}}(x^\eta, t) \geqslant \int_t^T \ell(x^\eta(s), u^\eta(s))ds + \phi_T(x^\eta(T)).$$

Thus, we have

$$\mathcal{F}_v(x_t^\eta) = v_{\mathrm{s}\rightarrow}(x_t^\eta, t) + v_{\rightarrow\mathrm{d}}(x_t^\eta, t) \geqslant \int_0^T \ell(x^\eta(s), u^\eta(s))ds + \phi_0(x^\eta(0)) + \phi_T(x^\eta(T)) \geqslant v^* - \eta \ .$$

The result in (6.42) is then concluded. □

Based on this property, we have the following result regarding the value function

Theorem 6.4.6. *For every $t \in [0,T]$ and $\delta < \eta$, for every $x \in \Gamma_t^\delta$, we have*

$$v_{\mathrm{s}\rightarrow}^\eta(x,t) = v_{\mathrm{s}\rightarrow}(x,t), \ v_{\rightarrow\mathrm{d}}^\eta(x,t) = v_{\rightarrow\mathrm{d}}(x,t) \ .$$

Proof. Let us first show the equality of $v_{\mathrm{s}\rightarrow}^\eta$ and $v_{\mathrm{s}\rightarrow}$. For an arbitrary $t \in [0,T]$, $v_{\mathrm{s}\rightarrow}^\eta(x,t) \leqslant v_{\mathrm{s}\rightarrow}(x,t)$ is straightforward since $\mathcal{O}_\eta^s \subseteq X$, for every $s \in [0,t]$. To show the reverse inequality, let us consider a δ-optimal trajectory $x^\delta(\cdot) : [0,T] \to X$ together with the control $u^\delta(\cdot) : [0,T] \to U$ of problem (6.1), and denote $x_t^\delta = x^\delta(t)$. One can deduce (argue by contradiction) that $x^\delta(\cdot)$, restricted in $[t,T]$, is a δ-optimal trajectory for the problem in backward direction, for which the value function is given by $v_{\rightarrow\mathrm{d}}$ with initial state (x_t^δ, t), i.e.,

$$\int_t^T \ell(x^\delta(s), u^\delta(s))ds + \phi_T(x^\delta(T)) \geqslant v_{\rightarrow\mathrm{d}}(x_t^\delta, t) - \delta \ . \tag{6.43}$$

Then, replacing $x^\delta(\cdot)$ restricted in $[0,t]$ by a ϵ-optimal trajectory for the forward problem with final state (x_t, t), we get a $(\eta+\epsilon)$-optimal trajectory for the problem (6.1). Since this holds for every $t \in [0,T]$, and by (6.42), for ϵ small enough, we deduce that $\Gamma_{\delta+\epsilon}^t \subset \mathcal{O}_\eta^t$, for every $t \in [0,T]$. Thus

$$v_{\mathrm{s}\rightarrow}^\eta(x,t) \geqslant v_{\mathrm{s}\rightarrow}(x,t) - \epsilon, \quad \forall t \in [0,T] \ . \tag{6.44}$$

Since (6.44) holds for arbitrary small ϵ, we deduce $v_{\mathrm{s}\rightarrow}^\eta(x,t) \geqslant v_{\mathrm{s}\rightarrow}(x,t)$ for all $t \in [0,T]$ and so the equality.

By the same arguments we have $v_{\rightarrow\mathrm{d}}^\eta(x,t) = v_{\rightarrow\mathrm{d}}(x,t)$. □

Theorem 6.4.6 indeed tells us that, to solve the problem (6.1), only the $\eta-$neighborhood at every time t, $\mathcal{O}_\eta^t$, around the optimal trajectory is relevant. In the following, we will focus on solving the problem (6.1) using an approximation of such a neighborhood.

6.4.2 Max-Plus Approximation of the Optimal Trajectories

Using the max-plus formulation of $v_{\rightarrow\mathrm{d}}$, we notice that (6.4) can be written as $v^* = \langle \phi_0, S_{\rightarrow\mathrm{d}}^T[\phi_T] \rangle$. Conversely, (6.7) can be written as $v^* = \langle S_{\mathrm{s}\rightarrow}^T[\phi_0], \phi_T \rangle$. Based on this observation, the optimality condition characterized in Proposition 6.4.2 can be written as

$$v^* = \sup_{x \in X} \mathcal{F}_v^t(x) = \langle S_{\mathrm{s}\rightarrow}^t[\phi_0], S_{\rightarrow\mathrm{d}}^t[\phi_T] \rangle \ , \ \forall t \in [0,T] \ . \tag{6.45}$$

Let us assume that we have a set of basis functions $\{w_i\}_{1 \leqslant i \leqslant p}$ together with a set of scalar $\{\lambda_i^{\mathrm{s}\rightarrow,t}\}_{1 \leqslant i \leqslant p}$ to approximate $v_{\mathrm{s}\rightarrow}^t$, and we use the same set of basis functions together with a set of scalar $\{\lambda_i^{\rightarrow\mathrm{d},i}\}_{1 \leqslant i \leqslant p}$ to approximate $v_{\rightarrow\mathrm{d}}^t$, i.e., for every $t \in \{0, \delta, \ldots, T\}$ and for every $x \in X$:

$$\begin{aligned} v_{\mathrm{s}\rightarrow}^t(x) &\approx v_{\mathrm{s}\rightarrow}^{t,h}(x) = \max_{1 \leqslant i \leqslant p} \{\lambda_i^{\mathrm{s}\rightarrow,t} + w_i(x)\} \ , \\ v_{\rightarrow\mathrm{d}}^t(x) &\approx v_{\rightarrow\mathrm{d}}^{t,h}(x) = \max_{1 \leqslant i \leqslant p} \{\lambda_i^{\rightarrow\mathrm{d},t} + w_i(x)\} \ . \end{aligned} \tag{6.46}$$

Then, an approximation for $\mathcal{F}_v^t$ can be obtained by:

$$\mathcal{F}_v^t \approx \mathcal{F}_{v^h}^t := \sup_{1\leqslant i,j\leqslant p} \{\lambda_i^{\mathrm{s}\rightarrow,t} + \lambda_{i'}^{\rightarrow\mathrm{d},t} + w_i + w_j\} \ . \tag{6.47}$$

Consequently, an approximation of (6.45) is achieved by

$$v^* \approx \sup_{x\in X} \mathcal{F}_{v^h}^t(x) = \sup_{1\leqslant i,j\leqslant p} \{\lambda_i^{\mathrm{s}\rightarrow,t} + \lambda_i^{\rightarrow\mathrm{d},t} + \langle w_i, w_j\rangle\} \ , \ \forall t \in [0,T] \ . \tag{6.48}$$

For a given η, let us denote by $\mathcal{O}_{\eta,h}^t \subseteq X$ the approximation of $\mathcal{O}_\eta^t$ defined as follows:

$$\mathcal{O}_{\eta,h}^t = \{x \in X \mid \mathcal{F}_{v^h}^t(x) > \max_{y\in X}\{\mathcal{F}_{v^h}^t(y) - \eta\} \}, \ \forall t \in [0,T] \ . \tag{6.49}$$

To efficiently find $\mathcal{O}_{\eta,h}^t$, we first notice that the r.h.s. in (6.49) can be computed using (6.48) as a function of the scalars $\mathcal{M}_{i,j}^* := \max_{y\in X} \mathcal{M}_{i,j}(y) = \langle w_i, w_j\rangle$, which can be computed easily (even analytically) when the basis functions w_i and w_j are chosen. Let us denote $\mathcal{N}_{v^h}^t(i,j) = \lambda_i^{\mathrm{s}\rightarrow,t} + \lambda_j^{\rightarrow\mathrm{d},t} + \mathcal{M}_{i,j}^*$ and let $\mathcal{N}_{v^h}^{t,*} = \max_{i,j\in I} \mathcal{N}_{v^H}^t(i,j)$, for every $t \in [0,T]$. Then, we first select the couples (i,j) as follows:

$$\mathcal{I}_{\eta,h}^t := \{(i,j) \in I^2 \mid \mathcal{N}_{v^h}^t(i,j) > \mathcal{N}_{v^h}^{t,*} - \eta\} \ . \tag{6.50}$$

Based on $\mathcal{I}_{\eta,h}^t$, we select $\mathcal{A}_{\eta,h}^t \subset X$ as follows:

$$\mathcal{A}_{\eta,h}^t = \{x \in X \mid \exists (i,j) \in \mathcal{I}_{\eta,h}^t, \ \mathcal{M}_{i,j}(x) > \mathcal{M}_{i,j}^* - \eta\} \ . \tag{6.51}$$

The set $\mathcal{A}_{\eta,h}^t$ can be compared with $\mathcal{O}_{\eta,h}^t$, and regarded as an approximation of $\mathcal{O}_\eta^t$. Moreover, we have the following result:

Theorem 6.4.7. *There exists an $\bar\eta$ depending on h and δ such that, for all $\eta \geqslant \bar\eta$ and $t \in \{0,\delta,\dots,T\}$, $\mathcal{A}_{\eta,h}^t$ contains $\Gamma_t^{\frac{\eta}{2}}$, that is the set of $\frac{\eta}{2}-$geodesic points for problem* (6.1a) *at time t.*

Proof. Fix a time step δ, and a time $t \in \{0,\delta,\dots,T\}$. We first notice that $\mathcal{O}_{\eta,h}^t \subset \mathcal{A}_{\eta,h}^t$. As shown in Proposition 6.4.2, the value function in a $\tilde\delta$-geodesic point $x \in \Gamma_t^{\tilde\delta}$ satisfies $\mathcal{F}_v^t(x) \geqslant \sup_{y\in X} \mathcal{F}_v^t(y) - \tilde\delta$. We know that the approximations $v_{\mathrm{s}\rightarrow}^{t,h}$ and $v_{\rightarrow\mathrm{d}}^{t,h}$ have certain error bounds (for the sup-norm) $\varepsilon_{\mathrm{s}\rightarrow}^h, \varepsilon_{\rightarrow\mathrm{d}}^h$ resp., depending on h and δ, but not on t:

$$\|v_{\mathrm{s}\rightarrow}^{t,h} - v_{\mathrm{s}\rightarrow}^t\|_\infty \leqslant \varepsilon_{\mathrm{s}\rightarrow}^h, \ \|v_{\rightarrow\mathrm{d}}^{t,h} - v_{\rightarrow\mathrm{d}}^t\|_\infty \leqslant \varepsilon_{\rightarrow\mathrm{d}}^h \ .$$

Denote $\varepsilon^h = \varepsilon_{\mathrm{s}\rightarrow}^h + \varepsilon_{\rightarrow\mathrm{d}}^h$, we have for every $y \in X$:

$$(\mathcal{F}_v^t(y) - \varepsilon^h) \leqslant \mathcal{F}_{v^h}^t(y) \leqslant (\mathcal{F}_v^t(y) + \varepsilon^h) \ .$$

Consider now $x' \notin \mathcal{A}_{\eta,h}^t$, so that $x' \notin \mathcal{O}_{\eta,h}^t$. Then

$$\begin{aligned}\mathcal{F}_v^t(x') &\leqslant \mathcal{F}_{v^h}^t(x') + \varepsilon^h \leqslant \sup_{y\in X}\{(\mathcal{F}_{v^h}^t(y) - \eta^h) + \varepsilon^h\} \\ &\leqslant \sup_{y\in X}\{\mathcal{F}_v^t(y) + (2\varepsilon^h - \eta^h)\} \ .\end{aligned}$$

Thus, if we take η^h big enough such that $(2\varepsilon^h + \tilde\delta - \eta^h) < 0$, we have $x' \notin \Gamma_t^{\tilde\delta}$. In particular, taking $\tilde\delta = \frac{\eta^h}{2}$, the result of Theorem 6.4.7 follows. □

6.5 Adaptive Max-Plus Approximation Method

In Section 6.2, we observed that to solve the optimal control problem (6.1), we only need to focus on a neighborhood of the optimal trajectory if we could approximately know it in advance. This also works when we intend to find an approximation of the value function. In this section, we will propose an adaptive max-plus approximation method to solve problem (6.1). The general idea is to begin with a small set of basis functions and test functions, then adaptively add more basis functions and test functions to improve the approximation of the two value functions $v_{\mathrm{s}\to}$ and $v_{\to\mathrm{d}}$, within a suitable neighborhood of the optimal trajectories derived from the approximate value functions.

6.5.1 Adaptive Two-level Max-Plus Method

In Section 6.4.2, we show how to approximate the optimal trajectories using the approximation of a pair of HJB equations. In this section, we propose a two-level approximation method, considering a coarse approximation for the optimal trajectories and a fine approximation in the domain containing the optimal trajectories, which is deduced from the coarse approximation.

Let us start with discretizing the time horizon by $N = \frac{T}{\delta}$ steps. Our algorithm consists of three main steps:

Step 1. Coarse Approximation. Denote $\hat{G}^H := \{\hat{x}_1^H, \hat{x}_2^H, \dots, \hat{x}_{p^H}^H\}$ and $G^H := \{x_1^H, x_2^H, \dots, x_{q^H}^H\}$ be two finite subsets of $\mathbb{R}^d$. Let $[I]^H := \{1, \dots, p^H\}$ be the index set of $\hat{G}^H$ and $I^H := \{1, \dots, q^H\}$ be the index set of G^H. We fix some sets of basis functions $\{w_{\hat{x}_i^H}\}_{\hat{x}_i^H \in \hat{G}^H}$ and test functions $\{z_{x_i^H}\}_{x_i^H \in G^H}$, and apply Algorithm 6.1 to approximate the value functions in two directions. This leads to an approximation of the two value functions $v_{\mathrm{s}\to}$ and $v_{\to\mathrm{d}}$, by the maps $v_{\mathrm{s}\to}^{t,H}$ and $v_{\to\mathrm{d}}^{t,H}$, which are tropical linear combination of the basis functions $\{w_{\hat{x}_i^H}\}_{\hat{x}_i^H \in \hat{G}^H}$ together with the scalars $\{\lambda_i^{\mathrm{s}\to,t}\}_{i\in[I]^H}$ and $\{\lambda_i^{\to\mathrm{d},t}\}_{i\in[I]^H}$, for every $t \in \{0, \delta, \dots, T\}$, respectively.

Step 2. Optimal Trajectory Approximation. We admit the same notation as in Section 6.4.2 for the approximation of optimal trajectories. For a given parameter η^H (will be detailed later), we first select the couples (i, i') as follows,

$$\mathcal{I}_{\eta^H,H}^t := \{(i,i') \in ([I]^H)^2 \mid \mathcal{N}_{v^H}^t(i,i') > \mathcal{N}_{v^H}^{t,*} - \eta^H\} \ . \tag{6.52}$$

Let us also denote

$$\begin{aligned} [I]_{\mathrm{s}\to,H}^{t,A} &= \{i \in [I]^H \mid \exists i' \in [I]^H \text{ such that } \mathcal{N}_{v^H}^t(i,i') > \mathcal{N}_{v^H}^{t,*} - \eta^H\} \ , \\ [I]_{\to\mathrm{d},H}^{t,A} &= \{i' \in [I]^H \mid \exists i \in [I]^H \text{ such that } \mathcal{N}_{v^H}^t(i,i') > \mathcal{N}_{v^H}^{t,*} - \eta^H\} \ , \end{aligned} \tag{6.53}$$

where the notation "A" stands for *active*. Then, based on $\mathcal{I}_{\eta^H,H}^t$, we can approximate the optimal trajectories using $\mathcal{A}_{\eta,H}^t \subset X$ defined as follows,

$$\mathcal{A}_{\eta^H,H}^t = \{x \in X \mid \exists (i,j) \in \mathcal{I}_{\eta^H,H}^t,\ \mathcal{M}_{i,j}(x) > \mathcal{M}_{i,j}^* - \eta^H\} \ . \tag{6.54}$$

By doing so, at the end of this step, we obtain an approximation for $\mathcal{O}_\eta^t$, where the restriction of the value functions $v_{\mathrm{s}\to}$, $v_{\to\mathrm{d}}$ in the domain $\mathcal{O}_\eta^t$ are approximated using the max-plus linear combination of the scalars $\{\lambda_i^{\mathrm{s}\to,t}\}_{i\in[I]_{\mathrm{s}\to,H}^{t,A}}$, and $\{\lambda_i^{\to\mathrm{d},t}\}_{i\in[I]_{\to\mathrm{d},H}^{t,A}}$, together with the basis functions $\{w_{\hat{x}_i^H}\}_{\hat{x}_i^H\in\hat{G}^H, i\in[I]_{\mathrm{s}\to,H}^{t,A}}$ and $\{w_{\hat{x}_i^H}\}_{\hat{x}_i^H\in\hat{G}^H, i\in[I]_{\to\mathrm{d},H}^{t,A}}$, respectively. Let us denote

$$X_f = \underset{t\in\{0,\delta,\dots,T\}}{\cup} \{\mathcal{A}_{\eta^H,H}^t\} \ , \tag{6.55}$$

and call it the *fine* region in coarse approximation.

Step 3. Fine Approximation. In this section, we first consider again two finite subsets of $\mathbb{R}^d$, $\hat{G}^h = \{\hat{x}_1^h, \hat{x}_2^h, \dots, \hat{x}_{p^h}^h\}$ and $G = \{x_1^h, x_2^h, \dots, x_{q^h}^h\}$, with $p^h > p^H$ and $q^h > q^H$. Then, we select the active points, denoted by $\hat{G}_A^h$ and G_A^h respectively, in $\hat{G}^h$ and G^h as following,

$$\begin{aligned} \hat{G}_A^h &:= \{\hat{x}^h \in \hat{G}^h \mid \exists\, i \in \bigcup_{t\in\{0,\dots,T\}} [I]^{t,A}_{\mathrm{s}\to,H} \text{ such that } \|\hat{x}^h - \hat{x}_i^H\|_\infty \leqslant \max\{H, H-h\} \ \}\ , \\ G_A^h &:= \{x^h \in G^h \mid \exists\, i \in \bigcup_{t\in\{0,\dots,T\}} [I]^{t,A}_{\to\mathrm{d},H} \text{ such that } \|x^h - \hat{x}_i^H\|_\infty \leqslant \max\{H, H-h\} \ \}\ , \end{aligned} \tag{6.56}$$

where H, h are the maximum radius of the Voronoi cells of the space $\hat{X}$ divided by the points of $\hat{G}^H$ and G^h respectively.

We then add new set of basis functions: $\{w_{\hat{x}_i^h}\}_{x_i^h \in \hat{G}_A^h}$ and the new set of test functions $\{z_{x_i^h}\}_{x_i^h \in G_A^h}$ to approximate the two value functions. In this step, the approximation of the value function will be only done in the fine region in coarse approximation. I.e., we apply Algorithm 6.1 with the new set of basis functions and test functions, and when compute the small time propagation, we restriction the trajectory in $\mathcal{A}^t_{\eta^H,H}$ for every time t. The complete algorithm works as in Algorithm 6.2.

Algorithm 6.2 Adaptive Two Level Max-Plus Approximation Method

Input: Mesh grids for coarse approximation $\hat{G}^H$ and G^H, mesh grids for fine approximation $\hat{G}^h$ and G^h. Parameter for quadratic basis/test functions: c, time step: $\delta \leqslant \bar{\delta}$.
Output: $\forall t \in \{0, \delta, \dots, T\}$, set of scalars $\{\lambda_i^t\}_{1\leqslant i\leqslant p}$.

Discretize time horizon by $N = \frac{T}{\delta}$ steps.
Choose quadratic basis functions *Base* generated by $\hat{G}^H$.
Choose quadratic test functions *Test* generated by G^H.
Approximate $v_{\mathrm{s}\to}$ and $v_{\to\mathrm{d}}$ using Algorithm 6.1 with *Base* and *Test*.
Set $[I]^H$ as an index set for *Base*;
Compute $\mathcal{M}^*_{i,i'}$ for all $i, i' \in ([I]^H)^2$.
for $t = 0, \delta, \dots, T$ **do**
    Compute $\mathcal{N}^t_{v^H}(i, i')$, for all $(i, i') \in ([I]^H)^2$.
    Compute $\mathcal{N}^{t,*}_{v^H}$.
    Select $[I]^{t,A}_{\mathrm{s}\to,H}$ and $[I]^{t,A}_{\to\mathrm{d},H}$ as in (6.53).
end for
Select active points $\hat{G}_A^h$ and G_A^h as in (6.56).
Base = *Base* $\cup \{w_{\hat{x}_i}\}_{\hat{x}_i \in \hat{G}_A^h}$;
Test = *Test* $\cup \{z_{x_i}\}_{x_i \in G_A^h}$;
Approximate $v_{\mathrm{s}\to}$ or $v_{\to\mathrm{d}}$ using Algorithm 6.1 with *Base* and *Test*;

For every $t \in [0, \delta, \dots, T]$, let us denote $v^{t,2}_{\mathrm{s}\to}$ and $v^{t,2}_{\to\mathrm{d}}$ the approximation using Algorithm 6.2, and denote $\tilde{v}^{t,h}_{\mathrm{s}\to}$, $\tilde{v}^{t,h}_{\to\mathrm{d}}$ the approximation of $v^t_{\mathrm{s}\to}$, $v^t_{\to\mathrm{d}}$ using Algorithm 6.1 with the sets of basis functions and test functions obtained from $\hat{G}^h$ and G^h respectively.

6.5.2 Adaptive Multi-Level Max-Plus Method.

The above approximation steps can be repeated, for instance, m times. To define the repeated steps, we need a family of parameters $\{\eta_l\}_{l=1,2,\dots,m-1}$ to select the *active* scalars for both directions, and to select the approximation of optimal trajectories, based on the previous two

directions' approximations. We also need a family of pairs of mesh grids $\{\hat{G}^{H_l}, G^{H_l}\}_{1\leqslant l\leqslant m}$ to generate the basis functions and test functions for both directions. We assume these parameters are fixed in advance, the computation works as follows

Level-1. In the first level, we do the same as in coarse approximation of two-level case, with $\hat{G}^H$ and G^H replaced by $\hat{G}^{H_1}$ and G^{H_1} respectively.

Level$-(l+1)$ *with* $1 \leqslant l < m$. In every level$-(l+1)$, we shall do iteratively the optimal trajectory approximation and fine approximation as in the two-level case. Let us denote $\{\lambda_i^{\mathrm{s}\rightarrow,t}\}_{i\in[I]_l}$ and $\{\lambda_i^{\rightarrow\mathrm{d},t}\}_{i\in[I]_l}$ the two set of scalars obtained at level$-l$ for the approximate value functions in two directions at time $t \in \{0,\delta,\dots,T\}$. Notice that $[I]_l$ is also the index set of $\hat{G}_A^{H_l} = \{x_1^{H_l}, x_2^{H_l}, \dots x_{|[I]_l|}^{H_l}\}$, where the basis functions at level l is generated from. When $l=1$, $\hat{G}_A^{H_1} = \hat{G}^{H_1}$.

Denote $\mathcal{M}_{i,j}^{*,l} := \max_{y\in X}\mathcal{M}_{i,j}^l(y) = \langle w_{x_i^{H_l}}, w_{x_j^{H_l}},\rangle$, for every $i,j \in [I]_l$. For every $t \in \{0,\delta,\dots,T\}$, denote $\mathcal{N}_l^t(i,j) = \lambda_i^{\mathrm{s}\rightarrow,t} + \lambda_j^{\rightarrow\mathrm{d},t} + \mathcal{M}_{i,j}^{*,l}$, for every $i,j \in [I]_l$, and let $\mathcal{N}_l^{t,*} = \max_{i,j\in[I]_l^t}\mathcal{N}_l^t(i,j)$. Then, given η_l, the *active* scalars in this level is selected as follow,

$$\begin{aligned}
[I]_{\mathrm{s}\rightarrow,l}^{t,A} &= \{i \in [I]_l^t \mid \exists i' \in [I]_l^t \text{ such that } \mathcal{N}_l^t(i,i') > \mathcal{N}_l^{t,*} - \eta_l\}\ ,\\
[I]_{\rightarrow\mathrm{d},l}^{t,A} &= \{i' \in [I]_l^t \mid \exists i \in [I]_l^t \text{ such that } \mathcal{N}_l^t(i,i') > \mathcal{N}_l^{t,*} - \eta_l\}\ .
\end{aligned} \tag{6.57}$$

The grids to generate basis functions and test functions in the level $l+1$ are selected as follows, denote $\Delta_{l+1} = \max\{H_l, H_l - H_{l+1}\}$, where H_l, H_{l+1} are the maximum radius of the Voronoi cells of the space $\hat{X}$ divided by the points of $\hat{G}^{H_l}$ and $\hat{G}^{H_{l+1}}$ respectively,

$$\begin{aligned}
\hat{G}_A^{H_{l+1}} &:= \{\hat{x}^{H_{l+1}} \in \hat{G}^{H_{l+1}} \mid \underset{t\in[0,\dots,T]}{\cup}[I]_{\mathrm{s}\rightarrow,l}^{t,A} \text{ s.t. } \|\hat{x}^{H_{l+1}} - \hat{x}_i^{H_l}\|_\infty \leqslant \Delta_{l+1}\}\ ,\\
G_A^{H_{l+1}} &:= \{\hat{x}^{H_{l+1}} \in G^{H_{l+1}} \mid \underset{t\in[0,\dots,T]}{\cup}[I]_{\rightarrow\mathrm{d},l}^{t,A} \text{ s.t. } \|x^{H_{l+1}} - \hat{x}_i^{H_l}\|_\infty \leqslant \Delta_{l+1}\}\ .
\end{aligned} \tag{6.58}$$

The basis functions and test functions for the approximation in level$-(l+1)$ are generated by $\{w_{\hat{x}_i^{H_{l+1}}}\}_{\hat{x}_i^{H_{l+1}}\in\hat{G}_A^{H_{l+1}}}$ and $\{z_{x_i^{H_{l+1}}}\}_{x_i^{H_{l+1}}\in G_A^{H_{l+1}}}$. Indeed, for every $1\leqslant l<m$, given the parameter η_l, we can also define the approximation of the optimal trajectory in level$-l$. Namely, for every $t\in\{0,\delta,\dots,T\}$, we select the couples (i,i') as follows,

$$\mathcal{I}_{\eta_l,l}^t := \{(i,i') \in ([I]_l)^2 \mid \mathcal{N}_l^t(i,i') > \mathcal{N}_l^{t,*} - \eta_l\}\ . \tag{6.59}$$

Then, based on $\mathcal{I}_{\eta_l,l}^t$, we can approximate the optimal trajectories using $\mathcal{A}_{\eta_l,l}^t \subset X$ defined as follows,

$$\mathcal{A}_{\eta_l,l}^t := \{x \in X \mid \exists (i,j) \in \mathcal{I}_{\eta_l,l}^t,\ \mathcal{M}_{i,j}(x) > \mathcal{M}_{i,j}^* - \eta_l\}\ . \tag{6.60}$$

By doing so, at the end of each level$-l$'s computation, we obtain an approximation for $\mathcal{O}_{\eta_l}^t$, where the restriction of the value function $v_{\mathrm{s}\rightarrow}$, $v_{\rightarrow\mathrm{d}}$ in the domain $\mathcal{O}_{\eta_l}^t$ are approximated using max-plus linear combination of the scalars $\{\lambda_i^{\mathrm{s}\rightarrow,t}\}_{i\in[I]_{\mathrm{s}\rightarrow,l}^{t,A}}$ and $\{\lambda_i^{\rightarrow\mathrm{d},t}\}_{i\in[I]_{\rightarrow\mathrm{d},l}^{t,A}}$, together with the basis functions $\{w_{\hat{x}_i^{H_l}}\}_{\hat{x}_i^{H_l}\in\hat{G}_A^{H_l}, i\in[I]_{\mathrm{s}\rightarrow,l}^{t,A}}$ and $\{w_{\hat{x}_i^{H_l}}\}_{\hat{x}_i^{H_l}\in\hat{G}_A^{H_l}, i\in[I]_{\rightarrow\mathrm{d},l}^{t,A}}$, respectively.

We then apply Algorithm 6.1 with the new basis test functions and test functions, in both directions. The computation in every time step t are also restricted at $\mathcal{A}_{\eta_l,l}^t$. The complete $m-$level Max-plus approximation method is given in Algorithm 6.3:

We count, in Algorithm 6.3, each time's computation of one *level* l, that is the first two main steps in two-level case. For each level $l\in\{1,2,\dots,m\}$ of Algorithm 6.3, let $v_{\mathrm{s}\rightarrow}^{t,H_l}, v_{\rightarrow\mathrm{d}}^{t,H_l}$ with $t\in\{0,\delta,\dots,T\}$, be the approximations of $v_{\mathrm{s}\rightarrow}^t$ and $v_{\rightarrow\mathrm{d}}^t$ computed until l, i.e., in the pseudocode when $l=l$. For all $l\in\{1,2,\dots,m\}$, and $t\in\{0,\delta,\dots,T\}$, let us denote by $\tilde{v}_{\mathrm{s}\rightarrow}^{t,H_l}$ and $\tilde{v}_{\rightarrow\mathrm{d}}^{t,H_l}$,

Algorithm 6.3 Adaptive $m-$level Max-Plus Approximation Method

Input: Mesh grids $\{\hat{G}^{H_l}, G^{H_l}\}_{1\leqslant l\leqslant m}$. Set of parameters $\{\eta_l\}_{1\leqslant l\leqslant m-1}$. Parameter for basis/test functions: c. time step $\delta \leqslant \bar{\delta}$.

Output: For every time t, set of scalars $\{\lambda_i^{t,\mathrm{s}\to}\}_{i\in[I]_m}$, $\{\lambda_i^{t,\to\mathrm{d}}\}_{i\in[I]_m}$.

Discretize time horizon by $N = \frac{T}{\delta}$ steps.
Choose basis functions *Base* generated by $\hat{G}^{H_1}$.
Choose test functions *Test* generated by G^{H_1}.
Approximate $v_{\mathrm{s}\to}$, $v_{\to\mathrm{d}}$ using Algorithm 6.1 with *Base* and *Test*.
Set $[I]$ as an index set for *Base*.
for $l = 1$ to m **do**
    $Base = Base \cup \{w_{\hat{x}_i}\}_{x_i\in\hat{G}_A^{H_l}}$.
    $Test = Test \cup \{z_{x_i}\}_{x_i\in G_A^{H_l}}$.
    Approximate $v_{\mathrm{s}\to}$, $v_{\to\mathrm{d}}$ using Algorithm 6.1 with *Base* and *Test*.
    if $l < m$ **then**
        Set $[I]$ as an index set for *Base*;
        Compute $\mathcal{M}^*_{i,j}$, for all $(i,j) \in ([I])^2$.
        for $t = 0, \delta, \ldots, T$ **do**
            Compute $\mathcal{N}_l^t(i,j)$, for all $(i,j) \in ([I]^2)$.
            Compute $\mathcal{N}_l^{t,*}$.
            Select $[I]^{t,A}_{\mathrm{s}\to,l}$ and $[I]^{t,A}_{\to\mathrm{d},l}$ as in (6.57).
        end for
        Select active points $\hat{G}_A^{H_{l+1}}$ and $G_A^{H_{l+1}}$ as in (6.58).
    end if
end for

the approximations of $v^t_{\mathrm{s}\to}$ and $v^t_{\to\mathrm{d}}$ using Algorithm 6.1 with the sets of basis functions and test functions obtained from the discretization grids $\hat{G}^{H_l}$ and G^{HJ_l}, respectively. Due to the initialization, the functions $\tilde{v}^{t,H_l}_{\mathrm{s}\to}, \tilde{v}^{t,H_l}_{\to\mathrm{d}}$ coincide with $v^{t,H_l}_{\mathrm{s}\to}, v^{t,H_l}_{\to\mathrm{d}}$ for $l = 1$. In next section, we will show the convergence using the relation between v^{t,H_l} and $\tilde{v}^{t,H_l}$.

6.5.3 Convergence and error analysis.

Let us begin by showing the convergence results of the two-level max-plus method. We first give some technical tools to show the convergence.

Definition 6.5.1. If $v : X \to \mathbb{R}$ is a $c-$semiconvex function on X, we define it's dual $\hat{v} : \hat{X} \to \mathbb{R}$ as follows, for every $x \in X$,

$$\hat{v}(\hat{x}) = \inf_{x\in X}\left\{\frac{c}{2}\|x-\hat{x}\|^2 + v(x)\right\} := \mathcal{D}^\flat_c(v)(\hat{x}) \ . \tag{6.61}$$

If v is a $c-$semiconcave function on X, we define it's dual $\hat{\hat{v}} : \hat{\hat{X}} \to \mathbb{R}$ as follows, for every $x \in X$,

$$\hat{\hat{v}}(\hat{\hat{x}}) = \sup_{x\in X}\left\{-\frac{c}{2}\|x-\hat{\hat{x}}\|^2 + v(x)\right\} := \mathcal{D}^\#_c(v)(\hat{\hat{x}}) \ . \tag{6.62}$$

This can be seen as a generalization of the Legendre-Fenchel transform. We refer to [FM00; CGQ04; McE06] the studies of such dualities.

Lemma 6.5.2. *(see [McE06])*

(i) *Assume v is $(c-\varepsilon)-$semiconvex on X, with $c > \varepsilon > 0$, then $\hat{v} = \mathcal{D}^\flat_c(v)$ is unique.*

(ii) *Assume v is $(c-\varepsilon)-$semiconcave on X, with $c > \varepsilon > 0$, then $\hat{\hat{v}} = \mathcal{D}^\#_c(v)$ is unique.*

Moreover, semiconcave and semiconvex functions have the following regularity property (see [CS04; Lak07]).

Lemma 6.5.3. *Let $f : X \to \mathbb{R}$ be $c-$semiconcave and $c-$semiconvex. Then $f \in C^{1,1}(X)$ and ∇f is Lipschitz continuous with Lipschitz constant c.*

For a given time horizon t, the approximation formula (6.48) for the optimal value of our problem can be interpreted using the duality defined in Definition 6.5.1. More precisely, we have

$$\begin{aligned} v^* &= \sup_{x\in X}\{v^t_{\mathrm{s}\to}(x) + v^t_{\to\mathrm{d}}(x)\} \\ &\approx \sup_{x\in X,\hat{x}\in\hat{X}}\{v^t_{\mathrm{s}\to}(x) - c\|x-\hat{x}\|^2 + \lambda^{\to\mathrm{d},t}_{\hat{x}}\} \\ &= \sup_{\hat{x}\in\hat{X}}\{\mathcal{D}^\#_c(v^t_{\mathrm{s}\to})(\hat{x}) + \mathcal{D}^\flat_c(v^t_{\to\mathrm{d}})(\hat{x})\} \ . \end{aligned} \tag{6.63}$$

Here, $\hat{x}$ corresponds to j in the expression (6.48), which is the center of basis function for approximating $v^t_{\to\mathrm{d}}$. Moreover, we have a similar formula for approximating the optimal value, that is

$$v^* \approx \sup_{\hat{x}\in\hat{X}}\{\mathcal{D}^\flat_c(v_{\mathrm{s}\to}(\hat{x})) + \mathcal{D}^\#_c(v_{\to\mathrm{d}})(\hat{x})\} \ . \tag{6.64}$$

In (6.64), $\hat{x}$ corresponds to i in the expression (6.48), which is the center of basis function for approximating $v^t_{\mathrm{s}\to}$. In our algorithm proposed in Section 6.5.1, the active nodes (6.53) are indeed selected in a neighborhood of $\hat{x}$ in which the maximum in (6.63) and (6.64) are achieved. Moreover, by a similar argument as in Theorem 6.4.7, we have the following result for the coarse approximation in the two-level method.

Lemma 6.5.4 (Corollary of Theorem 6.4.7)**.** *There exists an $\bar{\eta}^H > 0$ depending on H and δ such that, for all $\eta^H \geqslant \bar{\eta}^H$ and $t \in \{0, \delta, \dots, T\}$, $\mathcal{A}^t_{\eta^H, H}$ contains $\mathcal{O}^t_{\eta^H/2}$.*

Let us denote $v^{\eta^H}_{\mathrm{s}\rightarrow}$, $v^{\eta^H}_{\rightarrow\mathrm{d}}$ the value function for the control problems with the state constraint replaced by $\mathcal{O}^t_{\eta^H/2}$. We also assume that there is no propagation error, meaning that both $S^\delta_{\mathrm{s}\rightarrow}[w]$ and $S^\delta_{\rightarrow\mathrm{d}}[w]$ can be computed exactly for every basis function w. By the construction of the two-level method in Section 6.5.1, we have the following result, which represents the computation steps.

Proposition 6.5.5. *Under Assumption (A12), take quadratic basis functions and test functions centered at the points of $\hat{G}^h_A$ and G^h_A, respectively, with Hessian c. For every $t = 0, \delta, \dots, T - \delta$, we have*

$$v^{t+\delta,2}_{\mathrm{s}\rightarrow} = \mathcal{D}^\#_c \circ R_{\hat{G}^h_A} \circ \mathcal{D}^\flat_c \circ \mathcal{D}^\flat_c \circ R_{G^h_A} \circ \mathcal{D}^\#_c \circ S^\delta_{\mathrm{s}\rightarrow} \circ v^{t,2}_{\mathrm{s}\rightarrow} \ , \tag{6.65a}$$

$$v^{t,2}_{\rightarrow\mathrm{d}} = \mathcal{D}^\#_c \circ R_{\hat{G}^h_A} \circ \mathcal{D}^\flat_c \circ D^\flat_c \circ R_{G^h_A} \circ \mathcal{D}^\#_c \circ S^\delta_{\rightarrow\mathrm{d}} \circ v^{t+\delta,2}_{\rightarrow\mathrm{d}} \ , \tag{6.65b}$$

where $R_{G^h_A} : \mathbb{R}^{\hat{\hat{X}}} \to \mathbb{R}^{G^h_A}$, $R_{\hat{G}^h_A} : \mathbb{R}^{\hat{X}} \to \mathbb{R}^{\hat{G}^h_A}$ denote the restrictions.

Lets us simply denote $P^{h,2}_{\mathrm{s}\rightarrow}$ the operator in (6.65a) such that $v^{t+\delta,2}_{\mathrm{s}\rightarrow} = P^{h,2}_{\mathrm{s}\rightarrow} \circ S^\delta_{\mathrm{s}\rightarrow} \circ v^{t,2}_{\mathrm{s}\rightarrow}$, and $P^{h,2}_{\rightarrow\mathrm{d}}$ the operator in (6.65b) such that $v^{t,2}_{\rightarrow\mathrm{d}} = P^{h,2}_{\rightarrow\mathrm{d}} \circ S^\delta_{\rightarrow\mathrm{d}} \circ v^{t,2}_{\rightarrow\mathrm{d}}$.

Remark 6.5.6. Further refinement is required for the error estimates of the two-level and multi-level methods. Indeed, in [Lak07], error estimates are established as if we take the restriction in a grid discretizing a domain $\hat{X} = (X + B(0, \frac{L_v}{c}))$, where L_v is the Lipschitz constant for the value function. Here, we restrict further the grid such that it covers a neighborhood of the nodes in which the optimal of (6.63) and (6.64), which can be thought of the optimal trajectories for the dual problems corresponding to the dual value functions. We expect that the error estimates in [Lak07] still hold at certain neighborhood of the original problems. We present the following proposition, as a conjecture, and as a light to the computational complexity analysis.

Proposition 6.5.7.

(i) For every $l \in \{1, 2, \dots, m\}$, there exists an $\bar{\eta}_l$ depending on H_l and δ such that for all $\eta_l \geqslant \bar{\eta}_l$, and $t \in \{0, \delta, \dots, T\}$, X^{l+1}_f contains $\Gamma^{\frac{\eta_l}{2}}$, that is the $\frac{\eta_l}{2}-$geodesic points for problem (6.1a) *at time t.*

(ii) Taking η_l as proposed in (i)*, then for every $l \in \{2, \dots, m+1\}$, $t \in \{0, \delta, \dots, T\}$ and $x \in \Gamma^{\frac{\eta_l}{2}}_t$, we have*

$$|v^{t,H_l}_{\mathrm{s}\rightarrow}(x) - v^t_{\mathrm{s}\rightarrow}(x)| \leqslant C_{\mathrm{s}\rightarrow}(H_l)^2, \quad , |v^{t,H_l}_{\rightarrow\mathrm{d}}(x) - v^t_{\rightarrow\mathrm{d}}(x)| \leqslant C_{\rightarrow\mathrm{d}}(H_l)^2 \tag{6.66}$$

Thus, $\{v^{t,H_m}_{\mathrm{s}\rightarrow}\}$, $\{v^{t,H_m}_{\rightarrow\mathrm{d}}\}$ converge to $v^t_{\mathrm{s}\rightarrow}, v^t_{\rightarrow\mathrm{d}}$ respectively as $H_m \to 0$.

6.6 Computational Complexity

In this section, we analyze the computational complexity of our algorithm, and give the optimal parameters to turn the algorithm. The efficiency of our algorithm is highly dependent on the property of (the neighborhood of) optimal trajectories, for which we always make the following assumption in this section:

Assumption (A13) The set of geodesic points Γ^* is nonempty, and it consists of a finite number, K, of optimal trajectories for the problem (6.1). Moreover, there exists a positive constant $\bar{H} > 0$ such that

$$\Gamma^* + B^d(0, \bar{H}) \subset X \ . \tag{6.67}$$

Let us start with evaluating the neighborhood of optimal trajectories:

Proposition 6.6.1. *For every $t \in [0,T]$ and for every $x \in \mathcal{O}_\eta^t$, there exists a $x^* \in \Gamma_t^*$ such that:*

$$\|x - x^*\| \leqslant C_\beta(\eta)^\beta \ ,$$

where $C_\beta > 0$ and $\beta > 0$ are constants independent of x, t and η.

In Proposition 6.6.1, the exponent β determines the growth of the neighborhood $\mathcal{O}_\eta$ of the optimal trajectories, as a function of η. This exponent depends on the geometry of the value function. We shall see in Proposition 6.6.3 that for typical instances, taking $\beta = 1/2$ is admissible.

Based on Proposition 6.6.1, and the property that $\mathcal{O}^t_{\eta^H,H} \subset \mathcal{A}^t_{\eta^H,H} \subset \mathcal{O}^t_{2\eta^H,H}$ are approximations of $\mathcal{O}^t_{\eta^H}$, we obtain the following general space complexity result:

Proposition 6.6.2. *Given the sets of parameters $\{\eta_l\}_{l=1,2,\dots,m}$ and $\{H_l\}_{l=1,2,\dots,m+1}$, the number of discretization points generated by the adaptative max-plus approximation method can be bounded as follows:*

$$\mathcal{C}_{spa}(\{\eta_l, H_l\}) = \tilde{O}\Big(C^d(\frac{1}{H_1})^d + \sum_{l=2}^{m+1}(\frac{(\eta_{l-1})^{\beta(d-1)}}{(H_l)^d})\Big) \ . \tag{6.68}$$

Sketch of Proof. The summand $(\frac{1}{H_1})^d$ is the number of discretization points needed in the first level's grid, for which we discretized using mesh step H_1. Each summand $(\frac{(\eta_{l-1})^{\beta(d-1)}}{(H_l)^d})$ corresponds to the number of points in the level-l's grid, which is a "tubular" neighborhood around the optimal trajectory: at each time step, we only approximate the value functions using the points in a ball with radius $(\eta_{l-1})^\beta$ around the optimal trajectory. (This idea of using tubular neighborhoods of optimal paths to obtain complexity estimates originates from our recent work [AGL23a], dealing with a minimal time optimal control problem.) □

To obtain a complexity bound showing an attenuation of the curse of dimensionality, we certainly do not want the value function to be too "flat" near optimal trajectories. Indeed, this would result in a large neighborhood $\mathcal{O}_\eta$, and since this neighborhood is used to reduce the search space and define the new grid in Algorithm 6.3, the size of the new grid would not be so much reduced. Therefore, in addition to Assumption (A13), we shall also make the following convexity assumption, around the optimal trajectories.

Assumption (A14) There exists a constant $\bar{\eta}$ such that, for every optimal trajectory $x_k^*(\cdot)$, $k \in \{1,\dots,K\}$, and every $t \in [0,T]$, the value functions $v^t_{\mathrm{s}\to}$ and $v^t_{\to\mathrm{d}}$ are $\mu-$strongly concave in $B^d(x_k^*(t), \bar{\eta}) \cap X$.

Proposition 6.6.3. *Under Assumption (A13) and Assumption (A14), for every $\eta \leqslant (\frac{\bar{\eta}}{2\mu})^{\frac{1}{2}}$, we can take $\beta = \frac{1}{2}$ in Proposition 6.6.1.*

Proof. Without loss of generality, we assume first there exists an unique optimal trajectory $x^*(\cdot)$. For all $t \in [0,T]$ and $\eta \leqslant (\frac{\bar{\eta}}{2\mu})^{\frac{1}{2}}$, by Assumption (A14) we have

$$\mathcal{O}_\eta^t \subseteq B^d(x^*(t), \bar{\eta}) \cap X \ .$$

Moreover, the function $\mathcal{F}_v^t$ is $2\mu-$strongly concave on $B^d(x_k^*(t), \bar{\eta}) \cap X$. Let us now consider a $x \in \mathcal{O}_\eta^t$, and denote $x^*(t)$ by x_t^*. For all $s \in [0,1]$, the point $sx + (1-s)x_t^* \in B^d(x_t^*, \bar{\eta}) \cap X$. Then, by the strong concavity property, we have

$$\begin{aligned}&\mathcal{F}_v^t(sx + (1-s)x_t^*) + \mu(sx + (1-s)x_t^*)^2 \\ &\quad \geqslant \frac{1}{2}\Big\{s(\mathcal{F}_v^t(x) + \mu x^2) + (1-s)(\mathcal{F}_v^t(x_t^*) + \mu(x_t^*)^2)\Big\} \ .\end{aligned}$$

By a simple computation we obtain that if $s > 0$, then $\|x - x_t^*\| \leqslant (\frac{\eta}{2\mu(1-s)})^{\frac{1}{2}}$, and passing to the limit in s, we deduce that $\|x - x_t^*\| \leqslant (\frac{\eta}{2\mu})^{\frac{1}{2}}$. □

To make sure our *active region* X_f does contain all Γ_t^*, with $t = 0, \delta, \ldots, T$, we need to take η_l big enough, as discussed in Proposition 6.5.7. This result also holds for each level of the grids. Equation (6.66) indeed give us an upper bound for choosing the parameters η_l, depending on the parameters H_l. Let us plug this relationship between η_l and H_l into (6.68), and use the result of Proposition 6.6.3 under the Assumption (A13) and Assumption (A14), we have

$$\begin{aligned}&\mathcal{C}_{spa}(\{H_l\}_{l=1,\ldots,m+1}) \\ &\leqslant O\Big((H_1)^{-d} + C^{d-1}\sum_{l=2}^{m+1}\big((H_{l-1})^{d-1}(H_l)^{-d}\big)\Big) \ .\end{aligned} \tag{6.69}$$

As for the computational complexity, our aim is to establish an *ideal* complexity bound within an *oracle* Turing machine model. In this model, the time to solve a convex optimal control problem, in a small horizon, by calling a direct method (calling the oracle), is counted as one unit. This ideal complexity bound can be subsequently refined to get an effective bound in the ordinary Turing model of computation, recalling that ϵ-approximate solutions of *well conditioned* convex programming problems can be obtained in polynomial time by the ellipsoid or interior point methods. Using such an ideal model of computation is justified, since the only source of curse of dimensionality is the growth of the grid size, and since the execution time in this model is essentially the size of the largest grid.

Suppose now we want to have a final error in the order of ε, then we need to take $H_{m+1} = \mathcal{O}(\varepsilon^{\frac{1}{2}})$. Once H_{m+1} is fixed, $\mathcal{C}_{spa}$ is a convex function w.r.t. $\{H_l\}_{l=1,\ldots,m}$. We also notice that, up to a multiplicative factor, the computational complexity, in our oracle model, is the same as space complexity. Then, we have the following main result for the computational complexity of our algorithm:

Theorem 6.6.4. *Under Assumption (A11), Assumption (A12), Assumption (A13), Assumption (A14), assume further $d \geqslant 2$ and let $\nu := (1 - \frac{1}{d}) < 1$. Take $\eta_l = C(H_l)^2$ for every $l = 1, 2 \ldots, m$. In order to get a error $\mathcal{O}(\varepsilon)$:*

(i) *We shall take $H_{m+1} = C(\varepsilon)^{\frac{1}{2}}$, and $H_l = (H_{m+1})^{\frac{1-\nu^l}{1-\nu^{m+1}}}$ for all $l \in \{1, 2, \ldots, m\}$. In this case, the total computational complexity of our m-level method, expressed in the oracle model, is bounded by $O((C')^d(m+1)(\frac{1}{\varepsilon})^{\frac{1-\nu}{1-\nu^{m+1}}\frac{d}{2}})$, for some constant C'.*

(ii) *Set $m = \lceil \frac{1}{2}|d\log(\varepsilon)| - 1\rceil$, and take $H_l = (H_{m+1})^{\frac{l}{m+1}}$, then the total computational complexity reduces to $O((C')^d(\frac{1}{\varepsilon})^{\frac{1}{2}})$, for some constant C'.*

Proof. For (i), using the result of Proposition 6.5.7, the error of the algorithm is less or equal to $C(H_{m+1})^2 = \varepsilon$. Let us start with only two level of the grids, that is $m = 1$. In that case, the computational complexity, deduced from (6.69), is

$$\mathcal{C}_{comp}(H_1, H_2) = O((C_2)^2(H_1)^{-d} + (H_1)^{-d}(H_2)^{d-1}) \ , \tag{6.70}$$

for some constant C_2. When H_2 is fixed, this is a function of H_1 which gets its minimum for

$$H_1 = (C_2')(H_2)^{\frac{d}{2d-1}} \quad \text{with} \quad C_2' = (\frac{d}{d-1})^{\frac{1}{2d-1}} . \tag{6.71}$$

Then, for the $(m+1)$ level case, the total computational complexity derived from (6.69) remains dependent on $H_1, H_2, \ldots, H_m$, when H_{m+1} is fixed. To deduce the optimal values of $\{H_l\}_{l\in\{1,\ldots,m\}}$, we proceed by an induction on m, and use the iterative formula similar to (6.71). We then obtain the formula for every H_l as in (i). By substituting these values of H_l and $H_{m+1} = C(\varepsilon)^{\frac{1}{2}}$ into (6.69), we obtain the computational complexity bound in (i).

For (ii), we begin by considering when $\nu = 1$. In that case, pass to the limit when ν approaches to 1 for the formula of H_l as given in (i), we obtain the formula of H_l presented in (ii). This process also yields a new computational complexity bound, attained by taking the limit of the formula presented in (i), that is,

$$\mathcal{C}_{comp}(\{H_l\}_{l=1,\ldots,m+1}) = O\Big((C')^d(m+1)(\frac{1}{\varepsilon})^{\frac{d}{2(m+1)}}\Big) . \tag{6.72}$$

This function is dependent on the value of m, and achieves its minimum concerning m when $m = \frac{1}{2}d\log(\frac{1}{\varepsilon}) - 1$, leads to the formula of m (in the order of) the one in (ii). Now, let us substitute the values of H_l and into (6.69), yielding the following expression

$$\begin{aligned}\mathcal{C}_{comp}(\{H_l\}_{l=1\ldots,m+1}) &= O\Big((C')^d(\frac{1}{H_{m+1}})^{\frac{d}{m+1}}\sum_{l=0}^{m}(\frac{1}{H_{m+1}})^{\frac{l(1-\nu)}{m+1}d}\Big)\\ &= O\Big((C')^d(m+1)(\frac{1}{H_{m+1}})^{\frac{d}{m+1}}(\frac{1}{H_{m+1}})^{(1-\nu)d}\Big) .\end{aligned} \tag{6.73}$$

Take m as in (ii) and using $H_{m+1} = C(\varepsilon)^{\frac{1}{2}}$, we obtain the formula of complexity in (ii). □

6.7 Implementation and Numerical Experiments

In this section, we present some numerical tests, showing the efficiency of our algorithm. We also compare the results with the Max-plus finite element method in [AGL08]. The algorithms are implemented in C++, and executed on a single core of Quad Core IntelCore I7 at 2.3GHz with 16Gb of RAM.

Notice that for the efficient implementation of our algorithm, it is essential to dynamically construct and store information of the successive grid nodes. To accomplish this, we use a "hash-table" date structure, which has space complexity in the same order as the grid nodes and computational complexity of $O(1)$ for both searching information and inserting new grid nodes. Detailed information on a similar hash-table technique for storing dynamically constructed grid nodes can be found in the recent work of authors in [AGL23a].

6.7.1 Effective complexity of the multi-level max-plus method.

We applied our algorithm to several simple examples, in which the value function is known, so that the final approximation error can be computed exactly: the linear-quadratic control problems.

Consider the problem (6.1a) with $U = \mathbb{R}^d$ and $X = [-5, 5]^d$, the initial and final cost functions are $\phi_0(x) = -5(x - x_0)^2$, $\phi_T(x) = -5(x - x_T)^2$ with $x_0 = (-3, \ldots, -3)$ and $x_T = (3, \ldots, 3)$. The time horizon is $T = 5$ and is discretized with the time step $\delta = 0.5$. We choose quadratic basis functions and test functions with $c = 10$, centered at the points of regular grids.

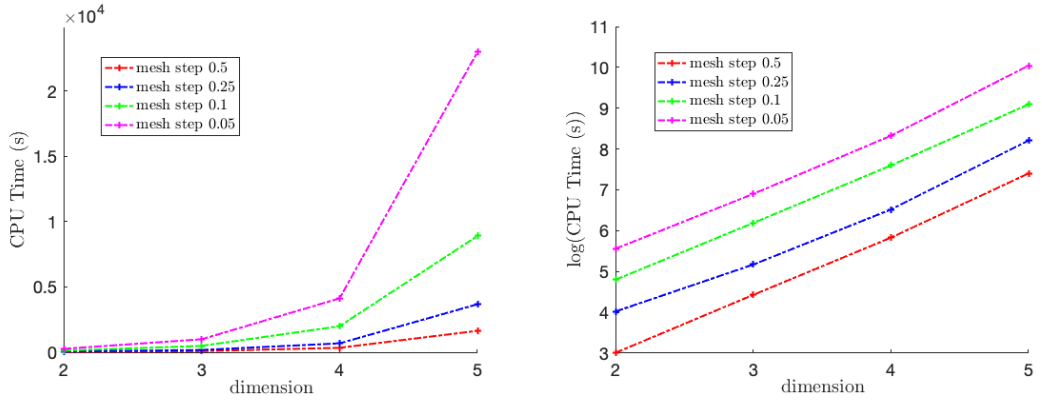
We vary the quadratic running costs and linear dynamics, and compare the results with the solution of Riccati equations. The computation times are approximately the same for different linear-quadratic problems. To summarize, within a time budget of 7 hours, we can reach dimension 6 with finial mesh step of $h = 0.2$ or dimension 5 with $h = 0.05$, whereas for classical grid based methods the computational complexity is $O(50^6)$ and $O(200^5)$ respectively. To compare, with the mesh step fixed at $h = 0.2$, the max-plus finite element method takes approximately 400 seconds for dimension 2, and approximately 3 hours for dimension 3.

To analysis the experimental complexity of the multi-level max-plus method in the light of the theoretical estimates of Theorem 6.6.4, we tested and compared the method on several dimensions and final mesh steps, with an almost optimal number of levels. Recall that if the number of levels is chosen optimal as in Theorem 6.6.4, we can expect a complexity in the order of $O(C^d(\frac{1}{h}))$. This means that the logarithm of CPU time should be of the form

$$\log(\text{CPU time}) \approxeq \log(C)d + \log(\frac{1}{h}) \ . \tag{6.74}$$

To verify the validity of this complexity estimation, we will run our algorithm in various dimensions and with different final mesh steps. We will then compute the logarithm of the CPU time as a function of the dimension and also as a function of $\log(\frac{1}{h})$, where h represents the mesh step. However, choosing an optimal number of levels may be difficult to implement due to small differences between the mesh steps. As a result, the obtained results may not always align perfectly with the theoretical predictions. Nevertheless, we have observed a nearly accurate match between the experimental complexity and the theoretical one.

We first present the tests for dimension ranging from 2 to 5, with several final mesh steps, for which we compute both the CPU time and logarithm of the CPU time as a function of dimension. We show in Figure 6.1 the graphs of these two functions (with several final mesh steps). The theoretical estimation suggests a slope of the $\log(C)$ for the logarithm of CPU time



(a) CPU time w.r.t. dimension

(b) log(CPU time) w.r.t. dimension

Figure 6.1: Growth of CPU time w.r.t. dimensions.

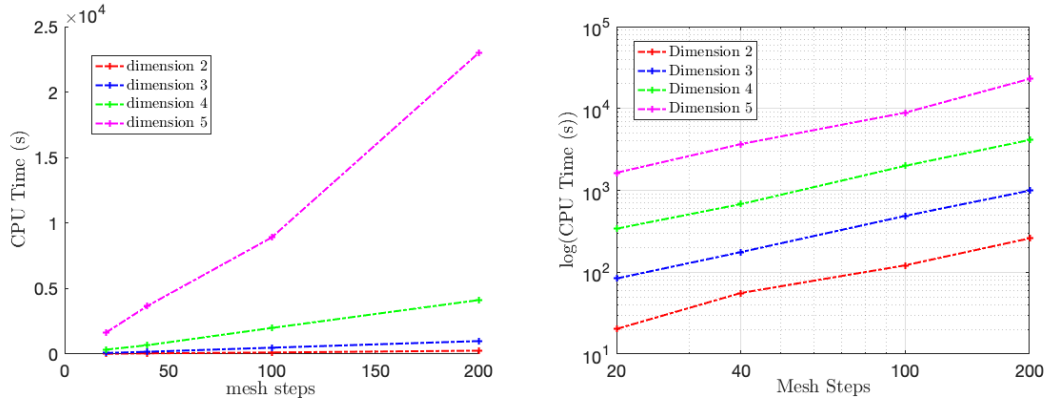
as a function of the dimension, which does not change for different mesh steps. We give the precise values of the logarithm of CPU time as a function of dimension in Table 6.1, where we compute the slope by linear regression. The estimated slopes are nearly the same for different mesh step, whereas the difference may be due to the number of levels not being exact optimal for different mesh steps.

Next, we present tests by varying the value of the final mesh steps, for which we compute the (logarithm of) CPU time. The graph of CPU time as a function of $\frac{1}{h}$ is shown in both linear

Table 6.1: Values and slope of log(CPU time) w.r.t. dimension.

	log(CPU Time) w.r.t. dimension				
Dimension	2	3	4	5	slope
Final mesh step 0.5	3.01	4.43	5.83	7.40	1.458
Final mesh step 0.25	4.02	5.17	6.52	8.21	1.391
Final mesh step 0.1	4.80	6.19	7.60	9.09	1.429
Final mesh step 0.05	5.56	6.90	8.32	10.04	1.488

and log-log scales in Figure 6.2. We also give the precise values of the logarithm of CPU time as a function of $\log(\frac{1}{h})$ in Table 6.2,with slopes computed by linear regression. The theoretical estimation indicates a slope of 1 for the logarithm of the CPU time as a function of $\log(\frac{1}{h})$ for all dimensions, whereas we observed again a nearly accurate match from the experimental estimated slope.



(a) CPU time w.r.t. mesh step, linear scale

(b) CPU time w.r.t. mesh step, log-log scale

Figure 6.2: Growth of CPU time w.r.t mesh steps

Table 6.2: Values and slope of log(CPU time) w.r.t. $\log(\frac{1}{h})$.

	log(CPU Time) w.r.t. $\log(\frac{1}{h})$				
$\log(\frac{1}{h})$	3.00	3.69	4.61	5.30	slope
Dimension 2	3.01	4.02	4.80	5.56	1.072
Dimension 3	4.43	5.17	6.19	6.90	1.080
Dimension 4	5.83	6.52	7.60	8.32	1.093
Dimension 5	7.40	8.21	9.09	10.04	1.122

Chapter 7

Semiconcave Dual Dynamic Programming and Its Application to N-body Systems

***

Abstract. We introduce a novel algorithm for approximating the value function, along with the optimal trajectory, for a class of Hamilton-Jacobi-Bellman equations arising from finite horizon deterministic optimal (maximization) control problems. In particular the reward function is only semiconcave with respect to the state. We rely on approximating the value function at a given time horizon by a min-plus linear combination of quadratic basis functions. These basis functions are then propagated by solving a dual problem. We show the convergence of our algorithm to the global optimum of the control problem. We apply our algorithm to obtain a min-plus low-rank tensor approximation of an N-body system.

7.1 Introduction

7.1.1 Motivation and Context

In this chapter, we consider the numerical approximation of the value function in a deterministic optimal control problem. One of the well-known optimality conditions for this type of control problem is provided by the *dynamic programming approach* (see for instance [FS06; BC08]). According to this approach, the value function of the optimal control problem is characterized as the *viscosity solution* of a first order *Hamilton-Jacobi-Bellman* (HJB) equation (see[CL83; CEL84]).

In general, whether in continuous time or discrete time cases, one significant advantage of the dynamic programming principle approach, compared to other approaches such as *Pontryagin's maximum principe* (PMP) [RZ98; RZ99], is its ability to find the global optimum of the problem. It achieves the global optimum with only mild regularity assumptions without necessitating stringent requirement of convexity. Moreover, it allows one to synthetize a feedback optimal control, leading to a solution robust against system perturbations. However, it entails solving either a nonlinear recurrence or fixed point equation, or the Hamilton-Jacobi-Bellman equation, which is a fully nonlinear partial differential equations over the state space. Consequently, as it is well known, it suffers from the *curse-of-dimensionality*. In essence, when the state space is continuous, classical grid based methods like finite difference scheme (for instance in [CL84]) and semi-lagrangian scheme (for instance in [FF14]) require constructing a discretized grid with the same dimension as the state space. Thus, the computational complexity of these schemes are exponential in dimension. Moreover, for a dimension $d \geqslant 5$, the memory allocation required to store information about the nodes, and the storage and retrieval of the value function become infeasible on modern computers due to its enormous size. On the other hand, however, it's important to note that even in low dimensions (in terms of state space), the numerical approximation of the continuous time control problem and HJB equation remains a challenging task. The regularity of the value function, which in most cases is only Lipschitz continuous, requires that the convergence of the numerical scheme should be understood in the viscosity sense. Additionally, each step of computation at the grid node involves an optimization problem over the control space, which is generally entailing a non-convex optimization problem. The aforementioned difficulties limit the practical applications of the dynamic programming principle approach, even through the theoretical studies are rather complete.

One way to overcome the curse-of-dimensionality is to replace the general problem of solving the HJ equation and approximating the value function in the entire state space with the computation of only one or several optimal trajectories with a fixed initial state. The latter problem can be solved, in particular in discrete time setting, using the stochastic dual dynamic programming (SDDP) method, which was first introduced in [PP91] (see also the further developments in [Sha11; GLP15; ZAS19; Gui20; Gui21]). It is designed to solve deterministic or stochastic control problems with a specific structure where the costs are jointly convex with respect to state and control, in the sense of minimization, and the dynamics are linear with respect to both state and control. Such a special structure guarantees that the value function is convex at every time horizon. Thus, the value function is approximated by a finite supremum of affine maps (that is, a piecewise affine convex map), and the approximated value function, together with the optimal trajectories starting from a fixed initial state, can then be computed efficiently using linear programming solvers. We refer to [Sha11; GLP15] for the convergence of SDDP. In cases where the assumptions on costs and dynamics are not satisfied, meaning there is a lack of convexity, the SDDP method typically only leads to a local optimum. In such situations, we mention more recent works that somehow involve exploiting the structure of optimal trajectories to approximate the value function. In [AFS19; AFS20], the authors introduced a tree-structured

discretization starting from a given initial state. They then prune the tree to a neighborhood around the optimal trajectory using the Lipschitz continuity of the value function. In [BGZ22], the authors introduced an adaptive discretization in the control space, which has been shown to be efficient when the dimension of the control space is low. In Chapter 4 and Chapter 6, or [AGL23a; AGL23b], a multi-level discretization method was introduced, using a coarse discretization to refine a tubular neighborhood around the optimal trajectory, then employing a finer discretization within this tubular neighborhood.

More recently, max-plus (or tropical) based methods have been developed to solve optimal control problems and HJB equations, for instance, in [FM00; McE06; AGL08; McE07; Qu14b; Dow18; ACT20; DDM23]. These methods take advantage of the max-plus linearity of the evolution semigroup of the HJB PDE [Mas87], the so called Lax-Oleinik semigroup. In a broad sense, following a discretization in time, the value function in a given time horizon of a maximization problem is approximated by a max-plus linear combination of basis functions. Then the basis functions are propagated over time steps using the max-plus linearity of the propagation semigroup. The max-plus based methods have shown advantages in solving classes of control problems and the associated HJB equations under specific regularity conditions.

7.1.2 Contribution

Here, our goal is to approximate the value function of a deterministic optimal control problem with a fixed initial state, together with the optimal trajectory. We consider a maximization problem, in the case that the reward function is known to be only semiconcave with respect to the state. Recall that a function defined on a convex set X of $\mathbb{R}^d$, $\phi : X \to \mathbb{R}$, is $c-$semiconcave with $c \geqslant 0$ if the map $x \mapsto \phi(x) - \frac{c}{2}\|x\|_2^2$ is concave on X. Seminconcavity is a useful generalization of concavity, especially for the value function of an optimal control problem (see for instance in [CF91; CS04]). Our method combines the concept of approximating semiconcave functions with tropical linear combination of quadratic basis functions, and the idea of dual dynamic programming method to propagate the basis functions, leading to a new algorithm.

In more details, instead of approximating the value function of a maximization problem by a max-plus linear combination of basis functions, which appears naturally in max-plus based methods, we employ a min-plus linear combination of basis functions to approximate from above the value function at a certain time horizon. Moreover, we only look for a tight approximation around the optimal trajectories. More precisely, we start with a (arbitrary) feasible trajectory for the control problem, and construct an initial upper approximate for the value function. In every iteration, we start with a rank k approximation for the value function, that is a min-plus linear combination of k basis functions. Since the evolutionary semigroup associated with maximization problem is only max-plus linear not min-plus linear, we then propagate the basis functions by solving a dual problem of the propagation, which gives us a new upper approximation. The trajectory is then updated to the optimal trajectory of the current approximate value function. This method is inspired by, and can be thought of as a generalization of the (Stochastic) Dual Dynamic Programming algorithm. We show that under certain regularity assumptions, in particular the reward function is only required to be semiconcave with respect to the state, our method converges towards the global maximum. We then apply our algorithm to find a tropical low-rank approximation of a N-body system.

The chapter is organized as follows. In Section 7.2, we give preliminary results on optimal control problems, in both continuous and discrete cases. We also give results on HJB equation, max-plus approximation method and the (deterministic) DDP method. In Section 7.3, we describe our algorithm, and give a comparison with the DDP method. In Section 7.4, we show the convergence of our algorithm to the global optimum. In Section 7.5, we apply our algorithm to obtain a tropical low-rank tensor approximation of a N-body system, and we give

the numerical results finding the ground state of this system.

7.2 Preliminaries

7.2.1 Optimal Control Problem, Hamilton-Jacobi-Bellman Equation

We are interested in solving the following finite horizon deterministic optimal control problem,

$$\max\left\{\int_0^T \ell(x(s),u(s))ds + \phi(x(T))\right\} \tag{7.1a}$$

over the set of trajectories $(x(s),u(s))$ satisfying

$$\begin{cases}\dot{x}(s) = f(x(s),u(s))\ ,\\ x(s)\in X,\ u(s)\in U\ ,\end{cases} \tag{7.1b}$$

for all $s\in[0,T]$, with the initial condition

$$x(0) = x\ . \tag{7.1c}$$

Here, $X\subset\mathbb{R}^d$, assumed to be bounded, is the state space and $U\subset\mathbb{R}^m$ is the control space. We assume that both X and U are convex sets. The final reward $\phi: X\to\mathbb{R}$ is concave. The Lagrangian (or running reward) $\ell: X\times U\to\mathbb{R}$, the dynamics $f: X\times U\to\mathbb{R}^d$ are given functions, and we assume the following basic regularity properties.

Assumption (A15)

i. $f: X\times U\to\mathbb{R}^d$ is bounded and Lipschitz continuous with respect to x, i.e.,

$$\begin{aligned}&\exists M_f>0, s.t.\ \|f(x,u)\|\leqslant M_f,\ \forall x\in X, u\in U\ ,\\ &\exists L_f>0, s.t.\ \|f(x,u)-f(x',u)\|\leqslant L_f(\|x-x'\|),\ \forall x,x'\in X, \forall u\in U\ .\end{aligned}$$

ii. $\ell: X\times U\to\mathbb{R}$ is bounded and Lipschitz continuous with respect to x, i.e.,

$$\begin{aligned}&\exists M_\ell>0, s.t.\ \|\ell(x,u)\|\leqslant M_\ell,\ \forall x\in X, u\in U\ ,\\ &\exists L_\ell>0, s.t.\ \|\ell(x,u)-\ell(x',u)\|\leqslant L_\ell(\|x-x'\|),\ \forall x,x'\in X, \forall u\in U\ .\end{aligned}$$

A well known sufficient and necessary optimality condition for the above problem is given by the Hamilton-Jacobi-Bellman (HJB) equation, which is derived from the dynamic programming principle. Indeed, let us consider the value function v associated to any $(x,t)\in X\times[0,T]$, where $v(x,t)$ is the supremum of $\int_t^T \ell(x(s),u(s))ds+\phi(x(T))$ under the constraint (7.1b), for every $s\in[t,T]$, and with the initial condition $x(t)=x$. Then, v is known to be the viscosity solution of the following HJB equation (see for instance [FS06]),

$$\begin{cases}-\dfrac{\partial v}{\partial t} - H(x,\nabla v) = 0, \quad (x,t)\in X\times[0,T]\ ,\\ v(x,T)=\phi(x), \qquad\qquad\qquad\qquad x\in X\ ,\end{cases} \tag{7.2a}$$

where

$$H(x,p) = \sup_{u\in U}\{p\cdot f(x,u)+\ell(x,u)\} \tag{7.2b}$$

is the Hamiltonian of the problem.

7.2.2 Propagation by Lax-Oleinik Semi-group and Max-plus Approximation

Recall that the *max-plus semifield* is the set $\mathbb{R}_{\max} := \mathbb{R} \cup \{-\infty\}$ equipped with the addition $a \oplus b := \max(a, b)$ and the multiplication $a \odot b := a + b$, with $-\infty$ as the zero and 0 as the unit. In the following, we denote by $v^t = v(\cdot, t)$ the value function of the optimal control problem (7.1) at time $t \in [0, T]$, and S^t the *Lax Oleinik semigroup* (or the evolution semigroup) of equation (7.2), that is, for all $0 \leqslant t \leqslant T$, S^{T-t} is the map sending the final cost function $\phi(\cdot)$ to the value function v^t:

$$v^t = S^{T-t}[\phi], \quad \forall t \in [0, T] \ , \tag{7.3}$$

such that the semi-group property $S^{t_1+t_2} = S^{t_1} \circ S^{t_2}$ is satisfied. In addition, the map S^t is *max-plus linear* (see [Mas87]), meaning that for all scalars $\lambda \in \mathbb{R}_{\max}$ and for all functions ϕ^1, $\phi^2 : X \to \mathbb{R}_{\max}$, we have:

$$\begin{aligned} S^t[\phi^1 \oplus \phi^2] &= S^t[\phi^1] \oplus S^t[\phi^2] \ , \\ S^t[\lambda \odot \phi^1] &= \lambda \odot S^t[\phi^1] \ , \end{aligned} \tag{7.4}$$

where for any functions ϕ^1 and ϕ^2, $\lambda \odot \phi^1$ is the function $x \in X \mapsto \lambda + \phi^1(x)$ and $\phi^1 \oplus \phi^2$ is the function $x \in X \mapsto \sup(\phi^1(x), \phi^2(x))$, in the usual sense (see for instance [FM00], [AGL08], [YD21b]). Indeed, the property (6.8) can be interpreted as the linearity in the sense of the max-plus semifield, and linear operators over max-plus semifield have been widely studied, for instance in [McE06; KM97].

The max-plus based approximation methods are recently developed to solve the problem (7.1). This kind of methods takes advantage of the max-plus linearity of S^t. For a given time horizon $t \in [0, T]$, the value function v^t is approximated by a max-plus linear combination of a family of "basic functions", $\{w_i\}_{1 \leqslant i \leqslant p}$, together with a set of scalars, $\{\lambda_i^t\}_{1 \leqslant i \leqslant p}$, that is

$$v^t \approx \bigoplus_{1 \leqslant i \leqslant p} w_i \lambda_i^t \ : \ x \mapsto \max_{1 \leqslant i \leqslant p} \{\lambda_i^t + w_i(x)\} \ . \tag{7.5}$$

Natural choices of the family of basis functions are the Lipschitz functions of the form $w_i(x) := -c\|x - x_i\|_1$, and the quadratic functions of the form $w_i(x) = -\frac{c}{2}\|x - a\|_2^2$. Indeed, denote $\overline{\mathbb{R}}_{\max} := \mathbb{R}_{\max} \cup \{+\infty\}$ the complete semiring extending $\mathbb{R}_{\max}$, and let $\mathcal{W}$ be a complete $\mathbb{R}_{\max}$-semimodule of functions $w : X \to \overline{\mathbb{R}}_{\max}$, meaning that $\mathcal{W}$ is stable under taking the supremum of an arbitrary family of functions, and by the addition of a constant, see [McE06; CGQ04] for background. The semimodule $\mathcal{W}$ is chosen in such a way that $v^t \in \mathcal{W}$ for all $t \in [0, T]$. The family of quadratic functions with Hessian c generates, in the sense of max-plus, the semimodule of lower-semicontinuous $c-$semiconconvex functions. In many applications, the value function v^t is known to be semiconcave for all $t \in [0, T]$, and then $\mathcal{W}$ can be taken as the set of semiconcave functions, which is a complete module (see for instance [McE06; AGL08]). The computation of the scalars is achieved through an iterative process.

7.2.3 (Deterministic) Markov Decision Process

A notable property of the max-plus representation of the value function in (7.2) is that, it leads to a discrete time deterministic optimal control problem, or a deterministic Markov decision processes. After a time discretization by, for instance, $N = \frac{T}{\delta}$ steps, the system (7.2) is represented as follows:

$$\begin{cases} v^t = S^\delta[v^{t+\delta}], \quad \forall \ t = T - \delta, T - 2\delta, \ldots, 0 \ , \\ v^T = \phi \ . \end{cases} \tag{7.6}$$

To numerically solve the system (7.6), we need also to approximate the small time propagation S^δ. This is indeed a similar optimal control problem as in (7.1), only here the time horizon δ is

small. Let us consider, for every function $\varphi : X \to \mathbb{R}$, a semi-lagrangian type approximation for S^δ, that is

$$S^\delta[\varphi] \approx S_h^\delta[\varphi] \ : \ x \in X \mapsto \sup_{u \in U} \{\delta \ell(x,u) + \varphi(x + \delta f(x,u))\} \ . \tag{7.7}$$

S_h^δ is indeed the Bellman operator of the following discrete-time finite-horizon deterministic optimal control problem:

$$\max \sum_{k=0}^{N-1} \delta \ell(x_k, u_k) + \phi(x_N)$$
$$s.t. \begin{cases} x_{k+1} = x_k + \delta f(x_k, u_k), \ \forall \ k \ , \\ x_k \in X, u_k \in U, \ \forall \ k \ , \\ x_0 \in X \text{ is given.} \end{cases} \tag{7.8}$$

The problem (7.8) is also referred to as a *multistage optimization problem* or a *deterministic Markov decision problem (MDP)*. In this context, given a state x_k and decision u_k at step k, a deterministic state x_{k+1} in the next step is reached. One intends to maximize the sum of the rewards $\delta \ell(x_k, u_k)$ induced by the controls, starting from a given state x_0 during a time horizon N, together with the final reward $\phi(x_N)$. The problem (7.8) can be solved by iteratively solving the following Bellman equation,

$$\begin{cases} V_N = \phi \ , \\ V_k = S_h^\delta[V^{k+1}], \quad \forall \ k = N-1, N-2, \ldots, 0 \ . \end{cases} \tag{7.9}$$

In (7.9), V_k, the value function at step k, can be thought of an approximation of $v^{k\delta}$ in (7.2). Moreover, the value of the problem (7.8) is equal to $V_0(x_0)$.

Combing the approximation method of the value function in (7.5), and the discrete approximation system in (7.9), we have the following recursive equation of the scalars:

$$\bigoplus_{1 \leqslant i \leqslant p} w_i \lambda_i^t = \bigoplus_{1 \leqslant i \leqslant p} S_h^\delta[w_i] \lambda_i^{t+\delta}, \quad \forall \ t = T - \delta, T - 2\delta \ldots, 0 \ . \tag{7.10}$$

Techniques to solve system (7.10) include applying a max-plus linear operator to λ^t at every time step (see in [FM00]), or applying a nonlinear operator, obtained by introducing a new family of "test" functions (see in [AGL08]).

7.2.4 (Deterministic) Dual Dynamic Programming

In the aforementioned max-plus based approximation method, the basis functions (or test functions) are typically generated by a grid, which is obtained through a discretization of the state space. Another class of algorithms, known as the Stochatic Dual Dynamic Programming (SDDP), solve the problems of the form (7.8) that do not involve a discretization in space. These algorithms are designed to identify one or several optimal trajectories of the problem. For the scope of our discussion, we are only interested in a deterministic version (DDP in the following).

The DDP is originally presented to solve the problems of the form (7.8), in the particular cases in which (see in [GLP15]),

(i) the running reward ℓ is jointly concave w.r.t. both x and u;

(ii) the dynamic f is linear w.r.t. both x and u;

(iii) the final reward ϕ is concave w.r.t. x.

The algorithm is initialized by drawing a trial (or arbitrary) trajectory x_k^0 of the problem (7.8), and an upper approximation V_k^0 for the value function V_k, obtained by a linear cut in x_k^0, for every $k = 0, 1, \ldots, N$. Then, at each iteration step m, the DDP performs a loop to first update the approximation of value function, then update the trajectory.

In every iteration step m, one first sets $x_0^m = x_0$. Then, for every $k = 1, 2, \ldots, N$, one solves the sub-problem which have the following form:

$$\begin{aligned} &\max_{x,u} \ \delta\ell(x,u) + V_{k+1}^{m-1}(x + \delta f(x,u)) \ . \\ &\text{s.t. } x = x_k^m \end{aligned} \tag{7.11}$$

This is indeed a convex programming problem. Denote θ_k^m the value of the problem (7.11), β_k^m the Lagrangian multiplier of the constraint $x = x_k^m$ and u_m^k the maximizer. Then, the updated approximate value function at step m is obtained as follows

$$V_k^m(x) := \min\{V_{m-1}^k(x),\ \theta_k^m + \langle \beta_k^m, x - x_k^m \rangle\} \ . \tag{7.12a}$$

The new trajectory is obtained as follows:

$$x_{k+1}^m = x_k^m + \delta f(x_k^m, u_m^k) \ . \tag{7.12b}$$

Notice that the DDP algorithm presented above involves only one loop in time to update both the value function and trajectory. Alternatively, a different approach involves two separate loops: a backward-in-time loop for value function updates, followed by a forward-in-time loop to update the trajectory. Then, the trajectory can be computed using the updated value function, i.e., the computation of u_m^k, used in (7.12b), employs V_{k+1}^m instead of V_{k+1}^{m-1} in (7.11).

7.3 Semiconcave Dual Dynamic Programming

In this section, we introduce a numerical approximation method addressing the problems of the form (7.1), or of the form (7.8), which is essentially a discretized form of (7.1). This method is inspired by the Dual Dynamic Programming method, which is only known to converge when ℓ is jointly concave w.r.t x and u, and f is linear w.r.t x and u (see for instance [GLP15]). Here, we address certain amount of non-convexity w.r.t x by approximating the value function from above using quadratic basis functions instead of affine basis functions.

In this section, we always make the following assumption:

Assumption (A16)

(i) f is affine with respect to both x and u.

(ii) $\ell \in \mathbb{C}^2(X \times U, \mathbb{R})$ and ℓ is strongly concave with respect to u, i.e., there exists a constant $\alpha_\ell > 0$ such that $\frac{\partial^2 \ell}{\partial u^2} \leqslant -\alpha_\ell I_d$, for the Loewner order of symmetric matrices, where I_d is the $d \times d$ identity matrix.

(iii) There exist constants $\beta_\ell, C > 0$ such that $-\beta_\ell I_d \leqslant \frac{\partial^2 \ell}{\partial x^2} \leqslant \beta_\ell I_d$, and $\| \frac{\partial^2 \ell}{\partial x \partial u} \| \leqslant C$.

(iv) There exist a constants L_T and $\beta_T > 0$ such that the final reward ϕ is a β_T-semiconcave and L_T-Lipschitz continuous function.

Moreover, we consider the problem (7.1) with a fixed initial condition $x(0) = x_0$ only.

7.3.1 Min-Plus Upper Approximation

Let us start by discretizing the time horizon by $N = \frac{T}{\delta}$ steps. Let S^t and v^t be defined as in Section 7.2. Recall that, by the semigroup property we have

$$\begin{aligned} &v^t = S^\delta[v^{t+\delta}],\ \forall\ t = T-\delta, T-2\delta, \dots, 0\ , \\ &v^T = \phi\ . \end{aligned} \tag{7.13}$$

Recall that the *min-plus semifield* is the set $\mathbb{R}_{\min} := \mathbb{R} \cup \{+\infty\}$ equipped with the addition $a \oplus b := \min(a, b)$ and the multiplication $a \odot b := a + b$, with $+\infty$ as the zero and 0 as the unit. We intend to solve the system (7.13) by approximating the value function from above using a min-plus linear combination of quadratic basis functions. More precisely, given a sequence of positive constant $\{c^t\}_{t=0,\delta,\dots,T}$, we define, for every $t \in \{0, , \delta, \dots, T\}$, the set of basis functions W_t as follows

$$W_t := \left\{ \frac{c^t}{2}\|x - a\|^2 + b \mid (a, b) \in X \times \mathbb{R} \right\}\ . \tag{7.14}$$

Our objective is to find an upper approximation $v^{t,h}$ of the value function v^t, for every $t \in \{0, \delta, \dots, T\}$, by a min-plus linear combination of a family of finitely many quadratic basis functions $\{w_i^t\}_{1 \leqslant i \leqslant r}$ such that $w_i^t \in W_t$, for every $1 \leqslant i \leqslant r$. The approximation $v^{t,h}$ takes the form

$$v^t \lessapprox v^{t,h} := \inf_{1 \leqslant i \leqslant r} \{w_i^t\},\ \forall t = 0, \delta, \dots, T\ . \tag{7.15}$$

However, for the maximization problem (7.1), S^t is only max-plus linear but generally not min-plus linear. Therefore the basis functions cannot be directly propagated using (7.13). In the following, we propose a recursive propagation method for the basis function, intending to provide at least a tight approximation in the set pf geodesic points $\Gamma^* = \{(x, t) \mid x = x^*(t), t = 0, \delta, 2\delta, \dots, T\}$, where $x^*(\cdot) : [0, T] \to X$ is an optimal trajectory of the control problem (7.1) with the fixed initial condition $x(0) = x_0$.

7.3.2 Propagation of Basis Functions By Dual Dynamic Programming

We first adapt the same approximation method for the small time propagation $S^\delta[v^{t+\delta}]$ as in (7.7), namely, for every $x \in X$ and $t = T, T-\delta, \dots, \delta$:

$$S^\delta[v^{t+\delta}](x) \approx S_h^\delta[v^{t+\delta}](x) := \sup_{u \in U}\{\delta\ell(x, u) + v^{t+\delta}(x + \delta f(x, u))\}\ . \tag{7.16}$$

By approximating $v^{t+\delta}$ in (7.16) using $v^{t+\delta,h}$, which has the formula (7.15), we obtain

$$S_h^\delta[v^{t+\delta}](x) \lessapprox \sup_{u \in U}\{\delta\ell(x, u) + \inf_{1 \leqslant i \leqslant r}\{w_i^{t+\delta}(x + \delta f(x, u))\}\}\ . \tag{7.17}$$

For every $x \in X$, (7.17) is indeed itself a constrained optimization problem, for which the supremum is taken over $u \in U$. Let $J(x)$ denote the supremum of the right hand side of (7.17). An equivalent formulation of this sub-problem is expressed as follows:

$$\begin{aligned} &J(x) = \max_{u,s}\ s, \\ &\text{s.t. } s \leqslant \delta\ell(x, u) + w_i^{t+\delta}(x + \delta f(x, u)),\ \forall 1 \leqslant i \leqslant r\ . \end{aligned} \tag{7.18}$$

The dual problem of (7.18) can be formulated as:

$$\begin{aligned} &J^*(x) = \min_\lambda \max_u \left\{ \sum_{i=1}^r \lambda_i^t \big(\delta\ell(x, u) + w_i^{t+\delta}(x + \delta f(x, u))\big) \right\}, \\ &\text{s.t. } \lambda_i^t \geqslant 0,\ \forall 1 \leqslant i \leqslant r,\ \text{and } \sum_{i=1}^r \lambda_i^t = 1\ . \end{aligned} \tag{7.19}$$

Remark 7.3.1. It always holds, according to the weak duality theorem, that $J(x) \leqslant J^*(x)$.

The formulation (7.19) gives again an upper approximation of the value function at the time step t. Moreover, one can notice that if ℓ is approximated by a quadratic function from above, then $J^*(x)$ will remain as an upper quadratic approximation of the value function at the time step t. An effective way to select such an upper approximation from the family of our basis functions (7.14) is to compute the optimal control u and the multiplier λ w.r.t. the constraint in (7.18) using a fixed x, and then construct a function that is a tight approximation in (or around) x.

This concept motivates us to use a recursive approach to select the basis functions: at each iteration, we add one new basis function at each time step, which is constructed by solving and approximating (7.19) at the "previous" optimal trajectory, and then updating the optimal trajectory.

7.3.3 The Semiconcave Dual Dynamic Programming Method

In this section, we present our methods, for both initialization and iterative steps.

7.3.3.1 Initialization Step

Assuming Assumption (A16), we can find a "simple", not necessary tight, quadratic upper approximation for ℓ and ϕ. Then, replacing ℓ by such an approximation, we can easily solve (7.1) along with the optimal trajectory, for instance, by integrating the Riccati differential equation. Let $v^{0,t}$, for every $t = 0, \delta, \dots, T$, denote the approximate value function, and $x^0(\cdot)$ denote the optimal trajectory for this system. Then, for every $t = 0, \delta, \dots, T$, we construct the initial basis functions, denoted by $w_1^t \in W_t$, as follows, for all $x \in X$:

$$\begin{aligned} w_1^t(x) =& \frac{c^t}{2}\|x - x^0(t)\|_2^2 \\ &+ \frac{\partial v^{0,t}}{\partial x}(x^0(t))\Big(x - x^0(t)\Big) \\ &+ v^{0,t}(x^0(t)) \ , \end{aligned} \tag{7.20a}$$

and we set the initial optimal trajectory as

$$x_1^*(t) = x^0(t) \ . \tag{7.20b}$$

By doing so, we have an upper approximation for v^t, such that

$$v^t(x) \leqslant v_1^{t,h}(x) := w_1^t(x), \ \forall t = 0, \delta, \dots, T \ . \tag{7.20c}$$

7.3.3.2 Iterative Step

At the iterative step $m + 1$, for every $t = 0, \delta, \dots, T$, we start with a rank m approximation of the value function at $v^{t+\delta}$, obtained in step m, that is

$$v^{t+\delta} \lessapprox v_m^{t+\delta,h} := \inf_{1 \leqslant i \leqslant m} \{w_i^{t+\delta}\}, \ \forall t = 0, \delta, \dots, T \ , \tag{7.21}$$

with $w_i^{t+\delta} \in W_{t+\delta}$, for every $i = 1, 2, \dots, m$. We first set $x_{m+1}^*(0) = x_0$. For every $t = 0, \delta, \dots, T$, we solve the sub-problem (7.19) with $r = m$, where we fix x as $x_{m+1}^*(t)$, and denote the optimal

multiplier λ and optimal control u by $\lambda^{t,*}_{m+1} = (\lambda^{t,*}_{m+1,i})_{i=1,\dots,m}$ and $u^{t,*}_{m+1}$, respectively. Then, we fix λ and u as $\lambda^{t,*}_{m+1}$ and $u^{t,*}_{m+1}$ respectively, and approximate $\ell(x,\cdot)$ as follows

$$\begin{aligned}\ell(x,u^{t,*}_{m+1}) \lessapprox \ell^{t,h}_{m+1}(x,u^{t,*}_{m+1}) =& \frac{c_\ell}{2}\|x - x^*_{m+1}(t)\|_2^2 \\ &+ \frac{\partial \ell}{\partial x}(x^*_{m+1}(t),u^{t,*}_{m+1})\Big(x - x^*_{m+1}(t)\Big) \\ &+ \ell(x^*_{m+1}(t),u^{t,*}_{m+1})\ .\end{aligned} \tag{7.22}$$

By doing so, for $u = u^{t,*}_{m+1}$ fixed, we have an upper quadratic approximation for ℓ, which is tight at $x^*_{m+1}(t)$, that is

$$\ell(x,u^{t,*}_{m+1}) \leqslant \ell^{t,h}_{m+1}(x,u^{t,*}_{m+1}),\ \forall x \in X\ , \tag{7.23a}$$

and

$$\ell(x^*_{m+1}(t),u^{t,*}_{m+1}) = \ell^{t,h}_{m+1}(x^*_{m+1}(t),u^{t,*}_{m+1})\ . \tag{7.23b}$$

With this approximation, we construct the new basis function, at the iteration step $m+1$ and at time step t, as follows:

$$w^t_{m+1} = \sum_{i=1}^m \lambda^{t,*}_{m+1,i}\Big(\delta\ell^{t,h}_{m+1}(\cdot,u^{t,*}_{m+1}) + w^{t+\delta}_i(\cdot + \delta f(\cdot,u^{t,*}_{m+1}))\Big)\ . \tag{7.24}$$

Remark 7.3.2. One can notice that, by our construction, we shall need to set $c_t = \delta c_\ell + c_{t+\delta}$. This implies a consistent increase in the Hessian of the basis functions by a constant value with each successive time step. We will give the details in the next section of fixing such parameters.

Finally, the new upper approximation of v^t at iterative step $m+1$ and time step t is updated as follows

$$v^{t,h}_{m+1} = \inf\{v^{t,h}_m, w^t_{m+1}\} = \inf_{1\leqslant i\leqslant m+1}\{w^t_i\}\ . \tag{7.25a}$$

The new optimal trajectory is updated as follows

$$x^*_{m+1}(t+\delta) = x^*_{m+1}(t) + \delta f(x^*_{m+1}(t),u^{t,*}_{m+1})\ . \tag{7.25b}$$

We repeat the processes until reaching the final time step T. The complete algorithm is presented in Algorithm 7.1. Here we use a fixed number r of iteration steps.

Algorithm 7.1 Semiconcave Dual Dynamic Programming (1)

1: Discretize time horizon by $N = \frac{T}{\delta}$ steps.
2: Find an upper quadratic approximation for ℓ and ϕ, solve the approximate system.
3: Set the initial trajectory, $x^*_t(0)$, and the initial approximation, $v^{t,h}_1$, as in (7.20).
4: **for** $m = 1,\dots,r-1$ **do**
5: Set $x^*_{m+1}(0) = x_0$.
6: **for** $t = 0,\delta,\dots,T$ **do**
7: Fix x as $x^*_{m+1}(t)$, solve the subproblem (7.19) to get the optimal $u^{t,*}_{m+1}$ and $\lambda^{t,*}_{m+1}$.
8: Fix u and λ as $u^{t,*}_{m+1}$ $\lambda^{t,*}_{m+1}$, approximate ℓ by (7.22).
9: Construct the new basis function by (7.24).
10: Construct the new approximation $v^{t,h}_{m+1}$ of v^t by (7.25a).
11: Update the trajectory by (7.25b).
12: **end for**
13: **end for**

Remark 7.3.3. In Algorithm 7.1, The construction of our approximation indicates that it is only tight for the points in optimal trajectories. Moreover, we indeed update the optimal trajectories using the value function at previous iteration step. Thus, it requires only one loop in time.

7.3.3.3 A variant of the Semiconcave DDP

In the following, we introduce a slightly variant of Algorithm 7.1, which involves two loops in time: a backward-in-time loop for updating the approximate value function, and a forward-in-time loop for updating the trajectory. This can be compared with most of the variants of the DDP, where the trajectory is updated using the freshly updated approximate value function at every iteration step.

We adapt the same initialization step as in Section 7.3.3.1. For the iterative step, to obtain the basis function as well as the approximate value function at iteration step $m+1$, we do a backward-in-time loop. At every time step, we apply the same computation as in Section 7.3.3.2, while using the optimal trajectory obtained in the previous step m, denoted by $\tilde{x}^*_m(\cdot)$.

More precisely, for every $t = T, T-\delta, \ldots, \delta$, we solve the sup-problem (7.19) with $r = m$, where we fix x as $x^*_m(t)$. Denote as before the optimal multiplier by $\tilde{\lambda}^{t,*}_{m+1}$ and the optimal control u by $\tilde{u}^{t,*}_{m+1}$. Then we fix $\tilde{\lambda}^{t,*}_{m+1}$ and $\tilde{u}^{t,*}_{m+1}$, and approximate $\ell(x,\cdot)$ by

$$\begin{aligned}\ell(x,\tilde{u}^{t,*}_{m+1}) \lessapprox \tilde{\ell}^{t,h}_{m+1}(x,\tilde{u}^{t,*}_{m+1}) =& \frac{c_\ell}{2}\|x-\tilde{x}^*_m(t)\|_2^2 \\ &+ \frac{\partial \ell}{\partial x}(\tilde{x}^*_m(t),\tilde{u}^{t,*}_{m+1})\Big(x-\tilde{x}^*_m(t)\Big) \\ &+ \ell(\tilde{x}^*_m(t),\tilde{u}^{t,*}_{m+1})\ .\end{aligned} \tag{7.26}$$

The construction of the new basis function is then as follows:

$$\tilde{w}^t_{m+1} = \sum_{i=1}^{m} \tilde{\lambda}^{t,*}_{m+1,i}\Big(\delta\tilde{\ell}^{t,h}_{m+1}(\cdot,\tilde{u}^{t,*}_{m+1}) + \tilde{w}^{t+\delta}_i(\cdot+\delta f(\cdot,\tilde{u}^{t,*}_{m+1}))\Big)\ , \tag{7.27}$$

and the new approximate value function at time t is

$$\tilde{v}^{t,h}_{m+1} = \inf\{\tilde{v}^{t,h}_m, \tilde{w}^t_{m+1}\} = \inf_{1\leqslant i\leqslant m+1}\{\tilde{w}^t_i\}\ . \tag{7.28}$$

Then, to find the new approximate optimal trajectory at step $m+1$, we do a forward-in-time loop. I.e., we begin with $\tilde{x}^*_{m+1}(0) = x_0$, for every $t = 0, \delta, \ldots, T-\delta$, we recursively get the new optimal control in step $m+1$ by

$$\begin{aligned}u^*_{m+1}(t) = \arg\max_{u\in U}\{&\ell(\tilde{x}^*_{m+1}(t),u) \\ &+ \tilde{v}^{t+\delta,h}_{k+1}(\tilde{x}^*_{m+1}(t)+\delta f(\tilde{x}^*_{m+1}(t),u))\}\ .\end{aligned} \tag{7.29a}$$

The optimal trajectory in step $m+1$ is computed by

$$\tilde{x}^*_{m+1}(t+\delta) = \tilde{x}^*_{m+1}(t) + \delta f(\tilde{x}^*_{m+1}(t), u^*_{m+1}(t)), \tag{7.29b}$$

The complete algorithm is presented in Algorithm 7.2. In this implementation, a fixed number of iteration steps r is used in order to obtain a rank r approximation of v^t. Another reasonable approach to determining the iteration step is to stop when the (relative) difference between the current approximation and the approximation in the previous step is small enough.

Remark 7.3.4. As mentioned previously, computing the optimal $u^{t,*}_p$ in the iteration of Algorithm 7.2 is equivalent to solving a maximization problem. Assuming Assumption (A16) and when δ is small, the objective function is concave and the feasible set U is also concave. Thus, this problem can be solved using standard optimization method, and sometimes can even be computed analytically. Computing the optimal $\lambda^{t,*}_p$ is equivalent to solving a convex programming problem, which can be efficiently approached using a solver, for instance CPLEX.

Algorithm 7.2 Semiconcave Dual Dynamic Programming (2)

Discretize time horizon by $N = \frac{T}{\delta}$ steps.
Find an upper quadratic approximation for ℓ and ϕ, solve the approximate system.
Set the initial trajectory, $x_t^*(0)$, and the initial approximation, $v_1^{t,h}$, as in (7.20).
for $m = 1, \dots, r-1$ **do**
    for $t = T, T-\delta, \dots, \delta$ **do**
        Fix x as $x_m^*(t)$, solve the subproblem (7.19) to get the optimal $\tilde{u}_{m+1}^{t,*}$ and $\tilde{\lambda}_{m+1}^{t,*}$.
        Fix u and λ as $u_{m+1}^{t,*}$ and $\lambda_{m+1}^{t,*}$, approximate ℓ by (7.26).
        Construct the new basis function by (7.27).
        Construct the new approximate $\tilde{v}_{m+1}^{t,h}$ of v^t by (7.28).
    end for
    Set $x_{k+1}^*(0) = x_0$.
    for $t = 0, \delta, \dots, T$ **do**
        Compute new optimal control $u_{m+1}^*(t)$ by (7.29a).
        Update the trajectory $x_{m+1}^*(t+\delta)$ by (7.29b).
    end for
end for

7.3.4 Comparison with Deterministic DDP

In this section, we compare our algorithm with the conventional DDP method. We aim to show that our approach can be thought of as an extension of DDP, in particular to handle situations involving the semiconcavity condition on the running reward.

We adapt the same initialization step as in Section 7.3.3.1. As for the iterative step, let us follow the lines of Section 7.2.4 to solve the problem of the form (7.8), where ℓ is only known to be semiconcave w.r.t. x. At the iteration step m, the major difficulty involves to solve the following subproblem for every $k = 0, 1, 2, \dots, N-1$:

$$\begin{aligned} &\max_{x,u} \left\{ \delta\ell(x,u) + V_{k+1}^{m-1}(x + \delta f(x,u)) \right\} \\ &\text{s.t. } x = x_k^m \ . \end{aligned} \tag{7.30}$$

In this case, both $\ell(\cdot, u)$ and the value function are only known to be semiconcave. Thus, instead of solving the problem (7.30), we consider a new subproblem as follows

$$\begin{aligned} &\max_{x,u} \left\{ \delta\ell(x,u) + V_{k+1}^{m-1}(x + \delta f(x,u)) - \frac{c_t}{2}x^2 \right\} \\ &\text{s.t. } x = x_k^m \ , \end{aligned} \tag{7.31}$$

with $t = k\delta$. Let us denote θ_k^m the value of problem (7.31), $\lambda_{k,x}^m$ the Lagrangian multiplier of the constraint $x = x_k^m$, that is

$$\lambda_{k,x}^m = \partial\varphi(x_k^m) \ , \tag{7.32a}$$

where $\partial\varphi$ denotes the supdifferential of φ and

$$\varphi(x) = \max_{u \in U} \left\{ \delta\ell(x,u) + V_{k+1}^{m-1}(x + \delta f(x,u)) - \frac{c_t}{2}x^2 \right\} \ . \tag{7.32b}$$

Denote also by u_m^k the maximizer. Then, we construct the new approximate value function as follows

$$V_k^m(x) := \min\{V_k^{m-1}(x), \frac{c_t}{2}x^2 + \langle \lambda_{k,x}^m, x - x_k^m \rangle + \theta_k^m\} \ . \tag{7.33}$$

It's worth noting that the term $-\frac{c_t}{2}x^2$ in (7.31) takes a role similar to the "regularization" for the concavity of the original maximization problem. Moreover, by construction, the map $x \to \frac{c_t}{2}x^2 + \langle \lambda_{k,x}^m, x - x_k^m \rangle + \theta_k^m$ belongs to the set of basis functions defined in (7.14), for every $m = 1, 2, \ldots, r$. Thus, we are indeed using the basis functions belonging to W_t to approximate the value function at time $t = k\delta$.

In the following, we will show that the construction of the approximate value function in (7.33) is identically to (7.25a). Let us denote

$$\varphi_k^m(x, u) = \delta\ell(x, u) + V_{k+1}^m(x + \delta f(x, u)) - \frac{c_t}{2}x^2 \ . \tag{7.34}$$

Then we have the following result.

Proposition 7.3.5. *Under Assumption (A16), denote $B = \frac{\partial f}{\partial u}$, there exist $\bar{\delta} > 0$ depending on α_ℓ and B, such that, for every $\delta \leqslant \bar{\delta}$, for every $k \in \{1, 2, \ldots, N\}$ and for every $m \in \{1, 2, \ldots, r\}$, φ_k^m is concave w.r.t. u.*

Proof. By construction in (7.33), we have that for every $m \in \{1, 2, \ldots, r\}$ and $k \in \{1, 2, \ldots, N\}$,

$$V_k^m(x) = \min_{1 \leqslant i \leqslant m} \{\frac{c_t}{2}x^2 + \langle \lambda_{k,x}^i, x - x_k^i \rangle + \theta_k^i\} \ .$$

Thus, it is enough to show that, for every $i \in \{1, 2, \ldots m\}$

$$\begin{aligned} \varphi_{k,i}^m(x, u) :=& \delta\ell(x, u) + \frac{c_{t+\delta}}{2}\|x + \delta f(x, u)\|^2 \\ &+ \langle \lambda_{k+1,x}^i, x + \delta f(x, u) - x_{k+1}^i \rangle + \theta_{k+1}^i - \frac{c_t}{2}x^2 \end{aligned} \tag{7.35}$$

is concave w.r.t. u, since minimization preserves the concavity. Since $\varphi_{k,i}^m(x, \cdot)$ is $\mathbb{C}^2$, and U is a convex set, by Assumption (A16), we have

$$\begin{aligned} \frac{\partial^2 \varphi_i^m(x, u)}{\partial u^2} &= \delta \frac{\partial^2 \ell}{\partial u^2}(x, u) + \delta^2 c_{t+\delta} B^T B \\ &\leqslant \delta(-\alpha_\ell I_m + \delta c_{t+\delta} B^T B) \ . \end{aligned} \tag{7.36}$$

Thus, taking $\bar{\delta} = \frac{\alpha_\ell}{c_{t+\delta}\|B\|}$, the result follows. □

Proposition 7.3.6. *Under assumption Assumption (A16), taking $\delta \leqslant \bar{\delta}$ as in Proposition 7.3.5, for every $m \in \{1, \ldots, r\}$, for every $k \in \{0, 1, \ldots, N\}$ such that $t = k\delta$, we have*

$$w_m^t(x) = \frac{c_t}{2}x^2 + \langle \lambda_{k,x}^m, x - x_k^m \rangle + \theta_k^m \ . \tag{7.37}$$

Proof. For every $k \in \{0, 1, \ldots, N\}$, denote $\hat{w}_m^t(x) = \frac{c_t}{2}x^2 + \langle \lambda_{k,x}^m, x - x_k^m \rangle + \theta_k^m$ with $t = k\delta$. Since both w_m^t and $\hat{w}_m^t$ are quadratic functions with Hessian c_t, i.e., $w_m^t, \hat{w}_m^t \in W_t$, it is enough to show that, for one particular $x = x_k^m$

$$w_m^t(x_k^m) = \hat{w}_m^t(x_k^m) \text{ and } \frac{\partial w_m^t}{\partial x}(x_k^m) = \frac{\partial \hat{w}_m^t}{\partial x}(x_k^m) \ . \tag{7.38}$$

When $m = 1$, (7.38) holds by our initialization step. Assume that for every $m \leqslant \bar{m}$ and for every $t \in \{0, \delta, \ldots, T\}$, (7.38) holds. First, by the construction in (7.31) and (7.32), for $t \in \{0, \delta, \ldots, T - \delta\}$ we have

$$\begin{aligned} \hat{w}_{\bar{m}+1}^t(x_k^{\bar{m}+1}) &= \frac{c_t}{2}(x_k^{\bar{m}+1})^2 + \theta_k^{\bar{m}+1} \\ &= \max_{x \in U} \left\{ \delta\ell(x_k^{\bar{m}+1}, u) + \min_{1 \leqslant i \leqslant \bar{m}} \left\{ \hat{w}_i^{t+\delta}(x_k^{\bar{m}+1} + \delta f(x_k^{\bar{m}+1}, u)) \right\} \right\} . \end{aligned} \tag{7.39}$$

By Proposition 7.3.5, the maximum over $u \in U$ is uniquely achieved in (7.39). Moreover, following the same formulations as in (7.18) and (7.19), we obtain that $w_m^t(x_k^m) = \hat{w}_m^t(x_k^m)$, since $\hat{w}_i^{t+\delta} = w_i^{t+\delta}$, for every $1 \leqslant i \leqslant \bar{m}$, and $\ell_{\bar{m}+1}^{t,h}(x_k^{\bar{m}+1}, \cdot) = \ell(x_k^{\bar{m}+1}, \cdot)$, as indicated in (7.23). Let us denote

$$\hat{\varphi}_k^{\bar{m}}(x) = \max_{x \in U} \left\{ \delta \ell(x,u) + \min_{1 \leqslant i \leqslant \bar{m}} \left\{ \hat{w}_i^{t+\delta}(x + \delta f(x_k, u) \right\} \right\} .$$

We have

$$\frac{\partial \hat{w}_{\bar{m}+1}^t}{\partial x}(x_k^{\bar{m}+1}) = c_t x_k^{\bar{m}+1} + \lambda_{k,x}^{\bar{m}} . \tag{7.40}$$

Combining the construction of $\lambda_{k,x}^{\bar{m}}$ in (7.32), we then have

$$\frac{\partial \hat{w}_{\bar{m}+1}^t}{\partial x}(x_k^{\bar{m}+1}) = \frac{\partial \hat{\varphi}_k^{\bar{m}}}{\partial x}(x_k^{\bar{m}+1}) . \tag{7.41}$$

Notice again that $\hat{w}_i^{t+\delta} = w_i^{t+\delta}$, for every $1 \leqslant i \leqslant \bar{m}$ and that by our approximation in (7.22), we have $\frac{\partial \ell_{\bar{m}+1}^{t,h}}{\partial x}(x_k^{\bar{m}+1}, \cdot) = \frac{\partial \ell}{\partial x}(x_k^{\bar{m}+1}, \cdot)$. Thus $\frac{\partial \hat{w}_{\bar{m}+1}^t}{\partial x}(x_k^{\bar{m}+1}) = \frac{\partial w_{\bar{m}+1}^t}{\partial x}(x_k^{\bar{m}+1})$.

The result of Proposition 7.3.6 is then concluded by an induction on m. □

7.4 Convergence Analysis

In this section, we will show the convergence of our algorithms. In particular, we will show the convergence to the global maximum of Problem (7.1) or (7.8). We start by showing some regularity and monotone properties of our approximation.

To simplify the analysis of the convergence, we shall add the following assumption.

Assumption (A17) The domain is invariant by the discretized dynamics in time $\delta > 0$ small enough, that is, for all $u \in U$ and for all $x \in X$, $x + \delta f(x,u) \in X$.

Assumption (A17) can be though of as a controllability assumption on the discrete system. Such assumption appears for the convergence analysis of numerical schemes for state constrained problems (see [FF14; AGL08]).

For every $V : X \to \mathbb{R}$, and δ as in Assumption (A17), let us denote $J : X \times U \to \mathbb{R}$ such that

$$J(x,u) = \delta \ell(x,u) + V(x + \delta f(x,u)) . \tag{7.42}$$

Moreover, we denote $A = \frac{\partial f}{\partial x}$ and $B = \frac{\partial f}{\partial u}$ in the following.

Proposition 7.4.1. *Let $V : X \to \mathbb{R}_{\max}$ be a β_V-semiconvave function. Under Assumption (A16), there exists $\bar{\delta} > 0$ depending on α_ℓ and B such that, for every $\delta < \bar{\delta}$, for every $x \in X$, the function $J(x,\cdot) : U \to \mathbb{R}_{\max}$ is strongly concave w.r.t. u.*

Proof. For every $u_1, u_2 \in U$, for an arbitrary $\lambda \in [0,1]$, for we have

$$\begin{aligned}
&J(x, \lambda u_1 + (1-\lambda)u_2) - (\lambda J(x,u_1) + (1-\lambda) J(x,u_2)) \\
&= \delta(\ell(x, \lambda u_1 + (1-\lambda)u_2) - (\lambda \ell(x,u_1) + (1-\lambda)\ell(x,u_2))) \\
&\quad + V(x + \delta f(x, \lambda u_1 + (1-\lambda)u_2)) - (\lambda V(x + \delta f(x,u_1)) + (1-\lambda) V(x + \delta f(x,u_2))) \\
&\geqslant \delta \frac{\alpha_\ell}{2} \lambda(1-\lambda)\|u_1 - u_2\|^2 - \frac{\beta_V}{2}\lambda(1-\lambda)\|\delta(f(x,u_1) - f(x,u_2))\|^2 \\
&\geqslant \lambda(1-\lambda)\frac{\delta(\alpha_\ell - \delta\beta_V\|B\|^2)}{2}\|u_1 - u_2\|^2 .
\end{aligned} \tag{7.43}$$

Thus, the result is concluded by taking $\bar{\delta} < \frac{\alpha_\ell}{\beta_V \|B\|^2}$. □

Corollary 7.4.2. For every $x \in X$, the maximum of $J(x,u)$ over $u \in U$ is uniquely achieved at some $u_x \in U$.

Proposition 7.4.3. *Let $V : X \to \mathbb{R}$ be a β_V-semiconvave function, assume Assumption (A16), Assumption (A17) and $\delta \leqslant \bar{\delta}$ as in Proposition 7.4.1 and Assumption (A17). Let us define the map $\mathcal{U} : X \to U$ such that, for every $x \in X$:*

$$\mathcal{U}(x) = \mathrm{Argmax}_{u \in U} J(x,u) \ . \tag{7.44}$$

Then, there exists a constant $L_{\mathcal{U}}$ such that $\mathcal{U}$ is $L_{\mathcal{U}}$ Lipschitz continuous, that is,

$$\|\mathcal{U}(x_1) - \mathcal{U}(x_2)\| \leqslant L_{\mathcal{U}} \|x_1 - x_2\| \ . \tag{7.45}$$

Proof. For simplicity, we do as if $V \in \mathcal{C}^2$ on X. Then, J is $\mathcal{C}^2$ on $X \times U$, by Assumption (A16) and Assumption (A17). By Corollary 7.4.2, for every $x \in X$, $J(x,u)$ achieves it's maximum at $u_x \in U$ such that

$$\frac{\partial J}{\partial u}(x, u_x) = 0 \ . \tag{7.46}$$

Then by implicit function theorem, the map $\mathcal{U}$ is continuously differentiable ($\mathcal{C}^1$) and thus on any compact subset of X, it is Lipschitz continuous. Moreover, we have

$$\frac{\partial \mathcal{U}}{\partial x} = -\Big(\frac{\partial^2 J}{\partial u^2}\Big)^{-1} \frac{\partial^2 J}{\partial u \partial x} \ . \tag{7.47}$$

By Assumption (A16), we have, if V is β_V-semiconcave, then

$$\frac{\partial^2 J}{\partial u^2} \leqslant -\delta \alpha_\ell I_d + \delta^2 \left(\frac{\partial f}{u}\right)^T \frac{\partial^2 V}{\partial x^2} \frac{\partial f}{u} \leqslant (-\alpha_\ell \delta + \delta^2 \beta_V \|\|B\|^2) I_d \ . \tag{7.48}$$

Moreover, $\frac{\partial^2 J}{\partial u \partial x} \leqslant C\delta + C\delta^2$. Thus, we have $\|\frac{\partial \mathcal{U}}{\partial x}\| \leqslant \frac{C(1+\delta)}{\alpha_\ell - \delta\beta_V \|B\|^2}$, where the denominator is positive since $\delta \leqslant \bar{\delta}$ as in Proposition 7.4.1.

We then conclude that $\mathcal{U}$ is Lipschitz continuous with the Lipschitz constant

$$L_{\mathcal{U}} = \frac{C(1+\delta)}{\alpha_\ell - \delta\beta_V \|B\|^2} \ . \tag{7.49}$$

□

Proposition 7.4.4. *Let $V : X \to \mathbb{R}$ be β_V-semiconvave. Denote $A = \frac{\partial f}{\partial x}$, $B = \frac{\partial f}{\partial u}$. Under Assumption (A16), Assumption (A17) and with $\delta \leqslant \bar{\delta}$ as in Proposition 7.4.1 and Assumption (A17), there exist a constant β'_V depending on $\beta_V, \beta_\ell, \delta$,A,B, C and $L_{\mathcal{U}}$, such that $S_h^\delta[V]$ is β'_V-semiconcave.*

Proof. For the semiconcavity of $S_h^\delta[V]$, it is enough to show that

$$S_h^\delta[V](x+h) + S_h^\delta[V](x-h) - 2S_h^\delta[V](x) \leqslant \beta'_V \|h\|^2 \ , \tag{7.50}$$

for every $x \in X$ and $h \in \mathbb{R}^d$ such that $[x-h, x+h] \subset X$. Let us denote by u_+^* an optimal control at $x+h$, and u_-^* an optimal control at $x-h$, that is, u_+^* and u_-^* satisfy

$$\begin{aligned} S_h^\delta[V](x+h) &= \delta \ell(x+h, u_+^*) + V(x+h+\delta f(x+h, u_+^*)) \ , \\ S_h^\delta[V](x-h) &= \delta \ell(x-h, u_-^*) + V(x-h+\delta f(x-h, u_-^*)) \ . \end{aligned} \tag{7.51}$$

We first notice that, for some $x'_+ \in [x, x+h]$, $x'_- \in [x-h, x]$, we have

$$\begin{aligned}
&\ell(x+h, u^*_+) + \ell(x-h, u^*_-) - (\ell(x, u^*_+) + \ell(x, u^*_-)) \\
&\leqslant h^T \frac{\partial \ell}{\partial x}(x, u^*_+) + \frac{1}{2} h^T \frac{\partial^2 \ell}{\partial x^2}(x'_+, u^*_+) h - h^T \frac{\partial \ell}{\partial x}(x, u^*_-) + \frac{1}{2} h^T \frac{\partial \ell^2}{\partial x^2}(x'_-, u^*_-) h \\
&\leqslant C\|h\|\|u^*_+ - u^*_-\| + \beta_\ell \|h\|^2 \\
&\leqslant (2CL_{\mathcal{U}} + \beta_\ell)\|h\|^2 \ ,
\end{aligned} \tag{7.52}$$

where C is a bound on the norm of $\frac{\partial^2 \ell}{\partial x \partial u}$. By a similar computation, we also have

$$\begin{aligned}
&V(x+h+\delta f(x+h, u^*_+)) + V(x-h+\delta f(x-h, u^*_-)) \\
&\quad - (V(x+\delta f(x, u^*_+)) + V(x+\delta f(x, u^*_-))) \\
&\leqslant \beta_V \delta \|B\|\|u^*_+ - u^*_-\|(1+\delta\|A\|)\|h\| + \beta_V (1+\delta\|A\|)^2 \|h\|^2 \\
&\leqslant \beta_V (1+\delta\|A\|)(1+\delta\|A\| + 2\delta L_{\mathcal{U}}\|B\|)\|h\|^2 \ .
\end{aligned} \tag{7.53}$$

Combining (7.52) and (7.53), and using the fact that u^*_+, u^*_- are two admissible controls for the problem at x, we have

$$\begin{aligned}
&S^\delta_h[V](x+h) + S^\delta_h[V](x-h) - 2S^\delta_h[V](x) \\
&= \delta \ell(x+h, u^*_+) + V(x+h+\delta f(x+h, u^*_+)) \\
&\ + \delta \ell(x-h, u^*_-) + V(x-h+\delta f(x-h, u^*_-)) \\
&\ - 2 \max_{u \in U} \{\delta \ell(x, u) + V(x+\delta(x, u))\} \\
&\leqslant \delta(\ell(x+h, u^*_+) + \ell(x-h, u^*_-) - (\ell(x, u^*_+) + \ell(x, u^*_-))) \\
&\ + V(x+h+\delta f(x+h, u^*_+)) + V(x-h+\delta f(x-h, u^*_-)) \\
&\ - (V(x+\delta f(x, u^*_+)) + V(x+\delta f(x, u^*_-))) \\
&\leqslant (\delta(2CL_{\mathcal{U}} + \beta_\ell) + \beta_V (1+\delta\|A\|)(1+\delta\|A\| + 2\delta L_{\mathcal{U}}\|B\|))\|h\|^2 \ .
\end{aligned} \tag{7.54}$$

Thus, the result is concluded with $\beta'_V = \delta(2CL_{\mathcal{U}} + \beta_\ell) + \beta_V(1+\delta\|A\|)(1+\delta\|A\| + 2\delta L_{\mathcal{U}}\|B\|)$. □

Using similar arguments as for Proposition 7.4.4, we can prove the following result.

Proposition 7.4.5. *Let $V : X \to \mathbb{R}$ be L_V-Lipschitz continuous. Denote $A = \frac{\partial f}{\partial x}$, $B = \frac{\partial f}{\partial u}$. Under Assumption (A16), Assumption (A17) and with $\delta \leqslant \bar{\delta}$ as in Proposition 7.4.1 and Assumption (A17), there exist a constant L'_V depending on L_V, δ, A, B, and $L_{\mathcal{U}}$, such that $S^\delta_h[V]$ is L'_V-Liptschitz continuous.*

For every $t = \{0, \delta, \dots, T\}$, let us denote v^t_δ the approximate value function of the problem (7.1) with S^δ approximated by S^δ_h, that is indeed the value function of (7.8) at step $k = \frac{t}{\delta}$. Then we have the following regularity properties for v^t_δ.

Corollary 7.4.6. Under Assumption (A16), Assumption (A17) and with $\delta \leqslant \bar{\delta}$ as in Proposition 7.4.1 and Assumption (A17), there exist constants $L^t_v, \beta^t_v > 0$ for $t = 0, \delta, \dots, T$, such that, $L^T_v = L_T$ and $\beta^T_v = \beta_T$ and for every $t = 0, \delta, \dots, T$, v^t_δ is β^t_v-semiconcave and L^t_v Lipschitz continuous on X.

For every $t \in \{0, \delta, \dots, T\}$, recall that the approximate value function $v^{t,h}_m$, for every $m \in \{1, 2, \dots, r\}$, is constructed using (7.25a). We have the following regularity and monotone properties regarding the approximate value functions.

Proposition 7.4.7. *For every $t = 0, \delta, \dots, T$, the sequence $\{v_m^{t,h}\}_{m\in\{1,2\dots,r\}}$ is monotone non-decreasing and lower bounded by v_δ^t, in the sense that for every $x \in X$,*

$$v_\delta^t(x) \leqslant v_r^{t,h}(x) \leqslant v_{r-1}^{t,h}(x) \leqslant \cdots \leqslant v_1^{t,h}(x) \ . \tag{7.55}$$

Moreover, we have, for all $t = 0, \delta, \dots, T$ and $m = 1, \dots, r$,

$$v_m^{t,h}(x) \geqslant S_h^\delta[v_{m-1}^{t,h}](x) \quad \forall x \in X \ , \tag{7.56}$$

and $v_m^{t,h}$ is L_v^t Lipschitz continuous on X.

Proof. The result can be deduced from the equivalence with the usual deterministic DDP algorithm shown in Section 7.3.4.

By the construction of the algorithm in (7.25a), we have that the sequence of approximations $\{v_m^{t,h}\}_{m\in\{1,2\dots,r\}}$ is non-increasing.

To show that $v^t \leqslant v_m^{t,h}$, we proceed by induction forward in m. First, we notice that for every $t = 0, \delta, \dots, T$, $v^t \leqslant v_1^{t,h}$. Assume that for $m > 1$, for every $t = 0, \delta, \dots, T$, we have $v^t \leqslant v_m^{t,h}$, that is,

$$w_i^t \geqslant v^t, \quad \forall\ 1 \leqslant i \leqslant m \ . \tag{7.57}$$

Then, following the construction in (7.22) and by the property (7.23), we have $w_{m+1}^t \geqslant v^t$, which is deduced by the construction in (7.24). Thus, the result is deduced further by (7.25a). □

Proposition 7.4.8. *Under Assumption (A16), for every $t \in \{0, \delta, \dots, T\}$, Let $\widetilde{\mathcal{W}}$ denote the set of c_T-semiconvex functions in X, then for every $\varphi \in \widetilde{W}$, we have*

$$\|S_h^t[\varphi] - S^t[\varphi]\|_\infty \leqslant C_h\delta, \quad \forall t \in [0, T] \ , \tag{7.58}$$

Sketch of Proof. It is enough to notice that the approximation in (7.16) is a semi-Lagrangian type discretization in time for our control problem. Thus the convergence follows from the convergence of the semi-lagrangian scheme. □

The following theorem shows that our algorithm converges to the true value function in the final generated trajectory.

Theorem 7.4.9. *Under Assumption (A16), Assumption (A17) and assuming $\delta \leqslant \bar{\delta}$, and X compact, we have, for every $t \in \{0, \delta, \dots, T\}$,*

(i) *The sequence of functions $v_k^{t,h}$ converges uniformly to a function $v_*^{t,h}$ on X as $k \to \infty$, with $v_*^{t,h} \geqslant v_\delta^t$.*

(ii) *The sequence $x_k^*(t)$ converges to a point $x^{t,*} \in X$ as $k \to \infty$, such that $v_*^{t,h}(x^{t,*}) = v_\delta^t(x^{t,*})$. Moreover, $(x^{t,*})_{t=0,\delta,\dots,T}$ is an optimal trajectory of the discrete time control problem (7.8).*

Sketch of Proof. The proof follows some of the arguments as in [GLP15; ACT20]. By Proposition 7.4.7, the sequence of functions $\{v_k^{t,h}, k \geqslant 0\}$ is equicontinuous, bounded and monotone on the compact set X. Therefore, the convergence of $\{v_k^{t,h}\}$ is deduced from the Arzelà-Ascoli theorem. Moreover, by Proposition 7.4.7 again, the limit satisfies $v_*^{t,h} \geqslant v_\delta^t$. Moreover, since S_h^δ is continuous for the uniform convergence, we also get that $v_*^{t,h}(x) \geqslant S_h^\delta[v_*^{t+\delta,h}](x)$, for all $x \in X$.

For (ii), since X is compact, the sequence of trajectories $(x_k^*(t))_{t=0,\delta,\dots,T}$ admits limit points when k goes to infinity. Let $(x^{t,*})_{t=0,\delta,\dots,T}$ be such a limit point. Since U is also compact, one can assume also that the corresponding subsequences of controls $(u_k^{t,*})_{t=0,\delta,\dots,T}$ converges towards $(u^{t,*})_{t=0,\delta,\dots,T}$. Then, by continuity of the dynamics f, the discrete trajectory $\{x^{t,*}\}_{t\in\{0,\delta,\dots,T\}}$ is

a feasible trajectory for the discrete control problem obtained by taking the control $u^{t,*}$ at time t.

By definition of $\{v_k^{t,h}\}$, we have $v_{k+1}^{t,h}(x_{k+1}^*(t)) \leqslant w_{k+1}^t(x_{k+1}^*(t)) = S_h^\delta[v_k^{t+\delta,h}](x_{k+1}^*(t)) = \delta\ell(x_{k+1}^*(t), u_{k+1}^{t,*}) + v_k^{t+\delta,h}(x_{k+1}^*(t) + \delta f(x_{k+1}^*(t), u_{k+1}^{t,*}))$. Passing to the limit in the subsequence when k goes to infinity, we obtain that $v_*^{t,h}(x^{t,*}) \leqslant S_h^\delta[v_*^{t+\delta,h}](x^{t,*})$, and that $u^{t,*}$ is the unique optimal control in the expression (7.16) of $S_h^\delta[v_*^{t+\delta,h}](x^{t,*})$. Recall that the functional inside the max is strongly concave w.r.t u when $\delta \leqslant \bar{\delta}$, as shown in Proposition 7.4.4, consequently, the optimal control is uniquely achieved. Since $v_*^{T,h}(x^{T,*}) = \varphi(x^{T,*})$, we deduce by backward induction on t that $v_*^{t,h}(x^{t,*}) \leqslant v_\delta^t(x^{t,*})$ for all $t = 0, \dots, T$. Since the other inequality holds, we get the equality. In particular $v_*^{0,h}(x_0) = v_\delta^0(x_0)$. Now going forward and using the uniqueness of the optimal control, we deduce that $x^{t,*}$ is the unique optimal trajectory of the discrete optimal control problem starting at x_0. This shows in particular that the limit point $(x^{t,*})_{y=0,\dots,T}$ is unique and thus that the sequence $(x_k^*(t))_{t=0,\delta,\dots,T}$ converges towards this unique trajectory. □

Remark 7.4.10. We proved the convergence of our algorithm towards a value function which coincides with the true value function on some limit of the generated trajectory. We also show that this limit of the generated trajectory is the optimal trajectory of discrete time control problem. Then, using Proposition 7.4.8, the convergence to the global maximum is automatic.

7.5 Application to Tropical Low-Rank Approximation of a N-Body System

In this section, we apply the approximation method introduced in Section 7.3 to solve N-body optimal control problems. In particular, we show that the approximation of the value function provided by semiconcave dual dynamic programming can be interpreted as a tropical (min-plus) analogue of a *low rank* approximation of a tensor. We illustrate the method by an application to a collision avoidance problem.

7.5.1 Min-Plus Low-Rank Approximation

Consider a function $F : \mathbb{R}^n \to \mathbb{R}_{\min}$. We intend to approximate F using a family of functions $\{\tilde{F}^k\}_{k=1,2,\dots,r}$, where $\tilde{F}^k : \mathbb{R}^n \to \mathbb{R}_{\min}$ for every k, by a min-plus addition such that, for every $x = (x_1, x_2, \dots, x_d) \in \mathbb{R}^n$

$$F(x) \approx \min_{1 \leqslant k \leqslant r} \tilde{F}^k(x) \ , \tag{7.59a}$$

Each function $\tilde{F}^k$ consists of a family of univariate functions $\{\tilde{F}_i^k\}_{i=1,2,\dots,n}$, where $\tilde{F}_i^k : \mathbb{R} \to \mathbb{R}_{\min}$ for every i, by min-plus multiplication such that:

$$\tilde{F}^k(x_1, \dots, x_n) = \sum_{i=1}^n \tilde{F}_i^k(x_i), \ \forall k = 1, 2, \dots, r \ . \tag{7.59b}$$

The function F^k, being the sum of functions in each variable, is analogous to a *rank one* tensor. Then, (7.59) provides an approximation of F by a tropical analogue of a "rank r tensor decomposition", see e.g. [OR20] for background. More generally, we will consider a function F defined on a Cartesian product $(\mathbb{R}^d)^N$, and look for an approximation of the same form, where now every x_i belongs to $\mathbb{R}^d$.

7.5.2 Optimal Control of A N-Body System

Consider a system consisting of N elementary dynamical subsystems in interaction, and denote $\xi_i(s) \in \mathbb{R}^d$ the position of the state at time t for every elementary subsystem $i \in \{1, 2, \dots, N\}$.

Denote by $V_i : \mathbb{R}^d \to \mathbb{R}$ an individual potential energy function for subsystem i, and $T_i : \mathbb{R}^d \to \mathbb{R}$ such that $T_i(\dot{\xi}_i(s)) := \frac{1}{2}\dot{\xi}_i^t(s) M_i \dot{\xi}_i(s)$ an individual kinetic energy at time s. Moreover, we denote by $W : \mathbb{R}^{N\times d} \to \mathbb{R}_{\max}$ a potential energy representing the interaction betweeen all the elementary subsystems. We look for a trajectory $\xi(s)$, for $s \in [0,t]$ minimizing the following action functional, under suitable initial and terminal conditions:

$$\mathcal{F}(\xi(\cdot)) = \int_0^t \Big(\sum_{i=1}^N \Big(V_i(\xi_i(s)) + T_i(\dot{\xi}_i(s)) \Big) + W(\xi(s)) \Big) ds \ . \tag{7.60}$$

Observe that the Lagrangian appearing in the action is the sum of the kinetic and potential energy, instead of their difference, as in classical mechanics. Lagrangians of the form (7.60), in which the potential is typically coercive (tending to ∞ as $\|\xi\| \to \infty$), are the most natural ones in applications to optimal control. In particular, thanks to coercivity of the potential, the minimization problem is well defined over an arbitrary horizon t. Indeed, we shall give a concrete illustration, solving a collision avoidance problem for N-bodies, below. In contrast, in classical mechanics, we recall that the trajectory of a conservative dynamical system is a minimizer of the action only for a *sufficiently small* time horizon [GT07]; so, the methods we present here apply only to mechanical problems over a small time horizon. We refer however the reader to [MD15] for the application of tropical methods to mechanical systems, in situations in which the action is not least.

Since our control problem evolves to minimize the action functional (7.60). This can be interpreted using the framework of optimal control problem as in (7.1), and the corresponding HJ equation. Since our control problem (7.1) is formulated as a maximization problem, we consider the lagrangian as the opposite of the action functional. More precisely, in the formulation of our control problem (7.1), we take

$$\begin{cases} x = \xi = (\xi_1, \dots, \xi_N), \quad u = \dot{\xi} \ , \\ \ell(x,u) = -\Big(\sum_{i=1}^N \Big(V_i(\xi_i) + T_i(\dot{\xi}_i) \Big) + W(\xi) \Big) \ , \\ f(x,u) = u \ , \\ \phi = 0 \ . \end{cases} \tag{7.61}$$

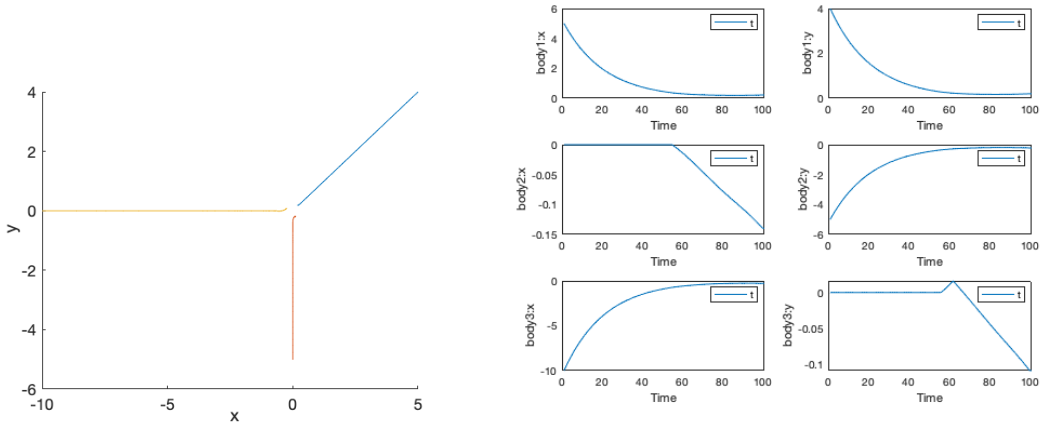
We characterize the minimum of the action functional at time t by v^t, which is the viscosity solution of the HJ equation (7.2).

7.5.3 Low-Rank Approximation of The N-Body System

We consider a particular interaction energy, the Coulomb potential, that is

$$W(x) = \sum_{1 \leqslant i < j \leqslant N} \frac{w}{|x_i - x_j|} \ , \tag{7.62}$$

where $w \geqslant 0$ is a constant. The physical interpretation of (7.62) is that each element must maintain a certain distance from one another. When two elements are sufficiently far apart, the interaction energy between them will have negligible effect on the individual system. We observe that the Lagrangian of the N-body system has naturally a low rank structure, as it involves a sum of local kinetic and potential energies, $\sum_i (V_i(\xi_i) + T_i(\dot{\xi}_i))$. However, the interaction term $W(x)$ coupling the different elementary subsystems, violates the low rank structure. We circumvent this issue with the present semiconcave SDDP method, which generates an approximation of the value function as an infimum of decomposable quadratic functions (given by sums of quadratic terms in each variable), thus ultimately generating a low-rank tensor approximation of the value function.



(a) Trajectories

(b) Evolution of individual trajectory over time

Figure 7.1: Trajectory of 3-body system with initial states $(5, 4), (0, -5), (-10, 0)$.

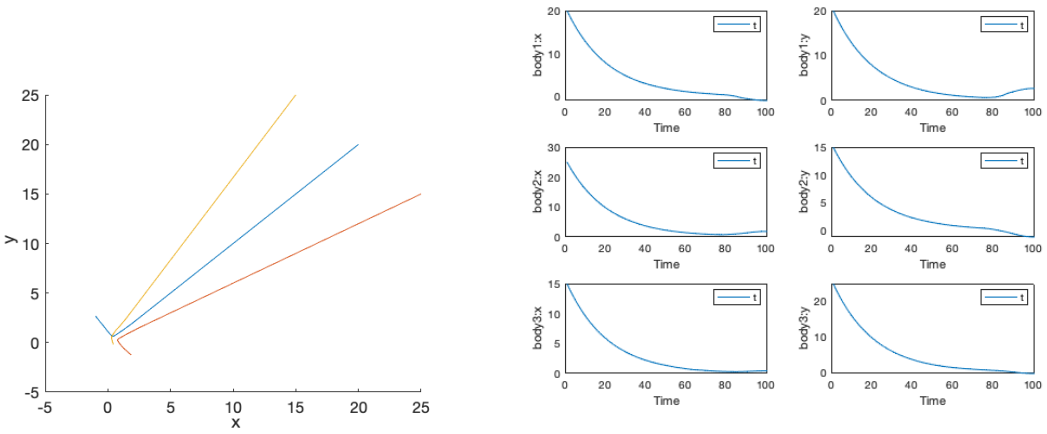
7.5.4 Numerical Results

In this section, we apply our algorithm to an N-body system described in Section (7.5), solving a collision avoidance problem. Our objective is to approximate the value function, identify the optimal trajectory, and determine the ground state for each element, with the initial positions of the elements are fixed, and the final cost $\phi = 0$.

Three Body System. We first test the algorithm on on three-body systems in dimension 2, resulting in optimal control problems in dimension 6. For each elementary dynamical subsystem, we take the same individual potential energy $V_i(\xi_i) = \frac{1}{2}\|\xi_i\|^2$ and individual kinetic energy $T_i(\xi_i) = \frac{1}{2}\|\dot{\xi}_i\|^2$. The interaction energy is taken to be the Coulomb potential as in (7.62) with $w = 1$. The state space for each body is taken to be $\mathbb{R}^2$ and the control space is also $\mathbb{R}^2$. The time horizon is 5 and is discretized by $\delta = 0.05$. The iteration step is fixed to be 50. Below, we present the trajectories of the three bodies and the evolution of individual trajectory over time, with different fixed initial states. The algorithms are implemented in MATLAB, and executed on a single core of Quad Core IntelCore I7 at 2.3GHz with 16Gb of RAM.

Notice that if $W = 0$, meaning that there is no interaction, the trajectory of each elementary subsystem converges to $(0, 0)$ in a straight line. With the Coulomb potential, we expect that the trajectory of each elementary subsystem still follows the straight line, as the distance between each element is large. However, the trajectory will not convergence to $(0, 0)$, but a new ground state that maximizes (in the framework of our optimal control problem) the total energy of the system. Below we present the pictures of trajectories of the system, and the evolution of individual trajectory of each elementary subsystem with respect to time, showing that the evolution of trajectories when the elements are close.

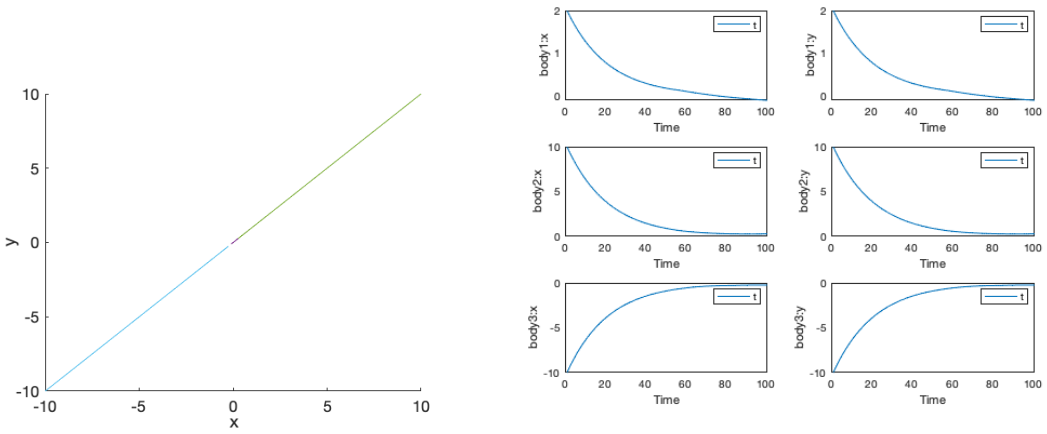
Four Body System. We then test our algorithm on the systems with four bodies, keeping the data the same as in the three-body system, resulting in an optimal control problem in dimension 8. In this case, we fixed the iteration step to be 100, ensuring numerical convergence for arbitrary initial states. Below, we present the trajectories of the four bodies and the evolution of individual trajectory over time, with different fixed initial states.



(a) Trajectories

(b) Evolution of individual trajectory over time

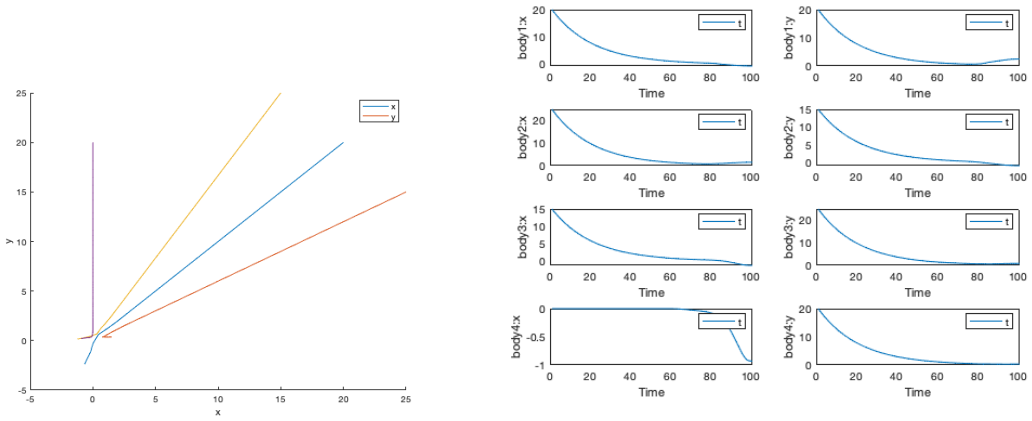
Figure 7.2: Trajectory of 3-body system with initial states $(20, 20), (25, 15), (15, 25)$.



(a) Trajectories

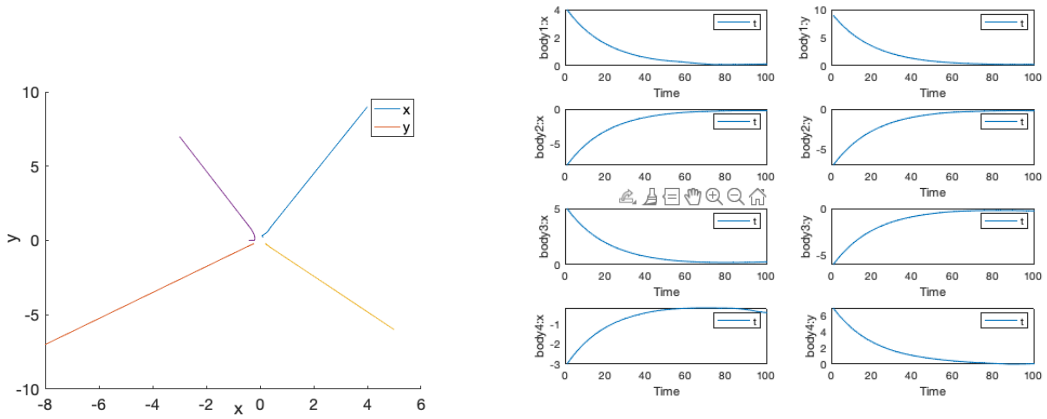
(b) Evolution of individual trajectory over time

Figure 7.3: Trajectory of 3-body system with initial states $(2, 2,), (10, 10), (-10, -10)$.



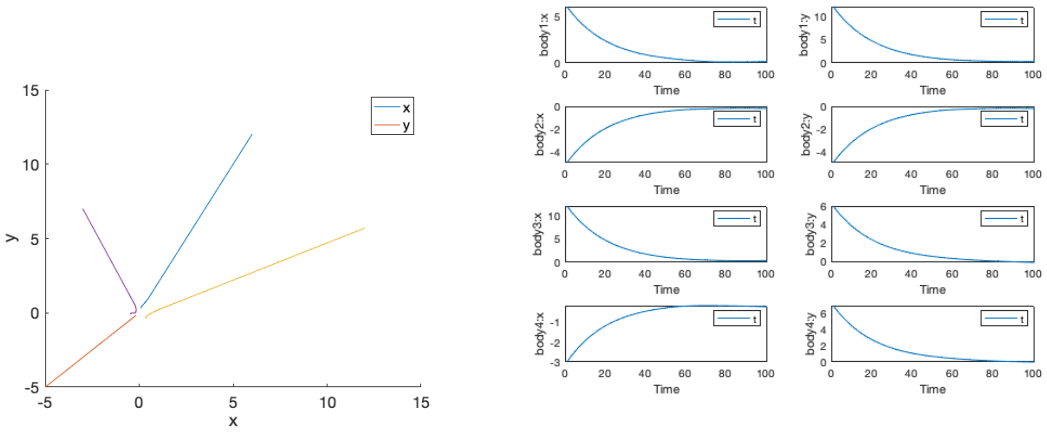
(a) Trajectories (b) Evolution of individual trajectory over time

Figure 7.4: Trajectory of 4-body system with initial states $(20, 20), (20, 15), (15, 20), (0, 20)$.



(a) Trajectories (b) Evolution of individual trajectory over time

Figure 7.5: Trajectory of 4-body system with initial states $(4, 10), (-8, -7), (5, -5), (-3, 6)$.



(a) Trajectories

(b) Evolution of individual trajectory over time

Figure 7.6: Trajectory of 4-body system with initial states $(6, 12), (-5, -5), (12, 6), (-3, 8)$.

Conclusion and Perspectives

***

We tackle here some perspectives and further developments of the work presented in this PhD dissertation.

In Chapter 4, we showed that our algorithm involves dynamically constructing the hierarchy of finer and finer grids in tubular neighborhoods around optimal trajectories. We established a computational complexity bound with respect to certain error bound by counting the number of grid nodes in such a tubular neighborhood. However, the general computational complexity depends on the set of parameters $\{\eta_l\}_{1\leqslant l\leqslant N}$. The conditions on the parameters η_l depend on the parameter γ and C_γ of the problem to be solved which are difficult to estimate in practice. Moreover, it may happen that the lower bound C_η is too large, making the theoretical complexity too large in practice even when $\gamma\beta = 1$. Indeed, in our numerical experiments, for a given problem, first several tests of the algorithm are done for large values of the mesh steps (or on the first levels of the multi-level method) with some initial guess of the constant C_γ, assuming $\gamma = 1$. In practice we observe that a slight adjustment of this constant can have a substantial impact on the computation time, while the outcomes of the approximation remain relatively consistent. One interesting question is to identify the conditions on the parameter η_l in an implementation efficient way.

Another open problem is to investigate the adaptability of our algorithm in Chapter 4 to anisotropic front propagation problem. As previously mentioned, fast marching techniques work as long as the "causality" property holds in the discretization, thus preventing the generalization of fast marching method to more general anisotropic front propagation problems. However, one can notice that the causality property always holds in the optimal trajectories. Since the initial concept of our algorithm is to identify the optimal trajectory and to restrict the search space within a tubular neighborhood of the optimal trajectory, we would like to analyze the causality property around the optimal trajectories, and to show the possibility to handle a certain amount of anisotropy within our algorithm.

In Chapter 5, we analyzed the convergence rate of a Semi-Lagrangian scheme for eikonal equation, in both semi-discretized case and fully discretized cases. We formulated the semi-discretized equation as the dynamic programming equation of a deterministic optimal control problem, and showed that the discrete value function is semiconcave. A convergence rate is obtained using this property. We then represented the fully-discretized equation as the dynamic programming equation of a stochastic optimal control problem, and showed the convergence rate using the properties of the corresponding Markov process. We intend to extend such techniques to encompass a broader range of optimal control problems with exit time, and to the associated Hamilton-Jacobi-Bellman equations. Indeed, in the proof of the minimum time problem case, we used the fact that the running cost is equal to 1, and the dynamics are upper and lower bounded, so that certain controllability assumptions are automatically satisfied. For more general exit time problems, we shall need controllability assumptions, in particular in the boundary of target sets.

In Chapter 6, we considered finite horizon deterministic optimal control problems. We

first showed that, after approximating the value function in a given time horizon by a max-plus linear combination of basis functions, the small time propagation of the coefficients of the basis function is indeed a convex programming problem. We then proposed further discretizing the time horizon, and solving this problem using direct methods. Alternative approaches can be used to tackle this small time propagation problem, for instance the Pontryagin Maximum Principle (PMP), given that we have already established the convexity property for this problem under certain regularity assumptions. This provides the possibility to combine the dynamic programming approach with PMP to address some control problems from a numerical point of view, as these two numerical approaches are typically considered belonging to two separate worlds. In the second part of Chapter 6, we extended the idea of dynamic grid refinement around the tubular neighborhood of optimal trajectories of Chapter 4. However, the error estimates under mild regularity assumptions need to be further refined. We intend to explore properties of the semiconcave duality of the value function. In our algorithm of Chapter 6, we indeed identify and construct the grid around the optimal trajectories of a semiconcave dual problem of the original one. We aim to identify a max-plus linear propagation operator associated with this dual problem, and subsequently do the convergence analysis and error estimates within the framework of the dual problem.

In Chapter 7, we introduced a novel algorithm for numerically finding the value function, along with the optimal trajectory, for a class of finite horizon deterministic optimal control problems with a fixed initial state. In particular, the reward function is only required to be semiconcave with respect to the state x. This method can be thought of as an extension of the (S)DDP method. We applied this method to solve a particular $N-$body system. We first notice that when the time horizon is long, the trajectory of the proposed $N-$body problem converges to the turnpike of the system, which gives the minimizer of the non-convex function, in that way the $N-$body problem appears as a dynamic version of non-convex global optimization problem. Thus, the work presented in Chapter 7 indeed proposes a novel framework to find the global maximum of semiconcave functions. We shall analyze the computational complexity and the convergence rate of the algorithm with respect to a certain error estimation. Moreover, a major motivation of this work is to find a tropical low-rank tensor approximation, which can be thought of as a tropical analogue of the classical low rank tensor decomposition used to approximate continuous functions. Consider generally a function $F : \mathbb{R}^d \to \mathbb{R}_{\min}$. In min-plus case, we intend to approximate F by a min-plus addition of a family of functions $\{\tilde{F}^k\}_{k=1,2,\dots,r}$, $\tilde{F}^k : \mathbb{R}^d \to \mathbb{R}_{\min}$ for every k, that is, for every $x = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$,

$$F(x) \approx \min_{1 \leqslant k \leqslant r} \tilde{F}^k(x) \ , \tag{7.63a}$$

where each function $\tilde{F}^k$ consists of min-plus multiplication of a family of univariate functions $\{\tilde{F}_i^k\}_{i=1,2,\dots,d}$, $\tilde{F}_i^k : \mathbb{R} \to \mathbb{R}_{\min}$ for every i:

$$\tilde{F}^k(x_1, \dots, x_d) = \sum_{i=1}^{d} \tilde{F}_i^k(x_i), \ \forall k = 1, 2, \dots, r \ . \tag{7.63b}$$

It should be interesting to employ the tropical low-rank tensor approximation method to directly approximate the value function, and then solve problems that exhibit specific structures.

Bibliography

***

[AGL05] M. Akian, S. Gaubert, and A. Lakhoua. "The max-plus finite element method for optimal control problems: further approximation results". In: *Proceedings of the joint 44th IEEE Conference on Decision and Control and European Control Conference ECC 2005 (CDC-ECC'05)*. Seville, Espagne, 2005, pp. 4505–4510. DOI: 10.1109/CDC.2005.1582872. eprint: math.OC/0509250.

[ACT20] M. Akian, J.-P. Chancelier, and B. Tran. "Tropical dynamic programming for lipschitz multistage stochastic programming". In: *arXiv preprint arXiv:2010.10619* (2020).

[AGL08] M. Akian, S. Gaubert, and A. Lakhoua. "The max-plus finite element method for solving deterministic optimal control problems: basic properties and convergence analysis". In: *SIAM Journal on Control and Optimization* 47.2 (2008), pp. 817–848.

[AGL23a] M. Akian, S. Gaubert, and S. Liu. *A Multi-Level Fast-Marching Method For The Minimum Time Problem*. Preprint arXiv:2303.10705. 2023.

[AGL23b] M. Akian, S. Gaubert, and S. Liu. "An adaptive multi-level max-plus method for deterministic optimal control problems". In: *Proceedings of the 22nd IFAC World Congress*. Also Preprint arXiv:2304.10342. Yokohama, Japan, 2023.

[AFS19] A. Alla, M. Falcone, and L. Saluzzi. "An efficient DP algorithm on a tree-structure for finite horizon optimal control problems". In: *SIAM Journal on Scientific Computing* 41.4 (2019), A2384–A2406.

[AFS20] A. Alla, M. Falcone, and L. Saluzzi. *A tree structure algorithm for optimal control problems with state constraints*. arXiv preprint arXiv:2009.12384. 2020.

[BC08] M. Bardi and I. Capuzzo-Dolcetta. *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Modern Birkhäuser Classics. Birkhäuser Boston, 2008. ISBN: 9780817647544.

[Bar89] M. Bardi. "A boundary value problem for the minimum-time function". In: *SIAM journal on control and optimization* 27.4 (1989), pp. 776–785.

[BF90] M. Bardi and M. Falcone. "An approximation scheme for the minimum time function". In: *SIAM Journal on Control and Optimization* 28.4 (1990), pp. 950–965.

[BS91] G. Barles and P. E. Souganidis. "Convergence of approximation schemes for fully nonlinear second order equations". In: *Asymptotic analysis* 4.3 (1991), pp. 271–283.

[BCC57] R. Bellman, R. Corporation, and K. M. R. Collection. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957. ISBN: 9780691079516.

[BZ99] M. Bergounioux and H. Zidani. "Pontryagin maximum principle for optimal control of variational inequalities". In: *SIAM Journal on Control and Optimization* 37.4 (1999), pp. 1273–1290.

[BB20] E. Berthier and F. Bach. "Max-plus linear approximations for deterministic continuous-state markov decision processes". In: *IEEE Control Systems Letters* 4.3 (2020), pp. 767–772.

[BCZ10] O. Bokanowski, E. Cristiani, and H. Zidani. "An Efficient Data Structure and Accurate Scheme to Solve Front Propagation Problems". In: *Journal of Scientific Computing* 42.2 (2010), pp. 251–273.

[Bok+15] O. Bokanowski, M. Falcone, R. Ferretti, L. Grüne, D. Kalise, and H. Zidani. "Value iteration convergence of ε-monotone schemes for stationary Hamilton-Jacobi equations". In: *Discrete and Continuous Dynamical Systems-Series A* 35.9 (2015), pp. 4041–4070.

[BFZ10] O. Bokanowski, N. Forcadel, and H. Zidani. "Reachability and minimal times for state constrained nonlinear problems without any controllability assumption". In: *SIAM Journal on Control and Optimization* 48.7 (2010), pp. 4292–4316.

[BFZ11] O. Bokanowski, N. Forcadel, and H. Zidani. "Deterministic state-constrained optimal control problems without controllability assumptions". In: *ESAIM: Control, Optimisation and Calculus of Variations* 17.4 (2011), pp. 995–1015.

[BGZ22] O. Bokanowski, N. Gammoudi, and H. Zidani. "Optimistic planning algorithms for state-constrained optimal control problems". In: *Computers & Mathematics with Applications* 109 (2022), pp. 158–179.

[Bok+13] O. Bokanowski, J. Garcke, M. Griebel, and I. Klompmaker. "An adaptive sparse grid semi-Lagrangian scheme for first order Hamilton-Jacobi Bellman equations". English. In: *J. Sci. Comput.* 55.3 (2013), pp. 575–605. ISSN: 0885-7474. DOI: 10.1007/s10915-012-9648-x.

[BMZ10] O. Bokanowski, N. Megdich, and H. Zidani. "Convergence of a non-monotone scheme for Hamilton–Jacobi–Bellman equations with discontinous initial data". In: *Numerische Mathematik* 115 (2010), pp. 1–44.

[BPW23] O. Bokanowski, A. Prost, and X. Warin. "Neural networks for first order HJB equations and application to front propagation with obstacle terms". In: *Partial Differential Equations and Applications* 4.5 (2023), p. 45.

[Bon+17] J. Bonnans Frederic, D. Giorgi, V. Grelard, B. Heymann, S. Maindrault, P. Martinon, O. Tissot, and J. Liu. *Bocop – A collection of examples*. Tech. rep. INRIA, 2017.

[Bon+06] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, 2006.

[BT13] L. Bourdin and E. Trélat. "Pontryagin maximum principle for finite dimensional nonlinear optimal control problems on time scales". In: *SIAM Journal on Control and Optimization* 51.5 (2013), pp. 3781–3813.

[CF96] F. Camilli and M. Falcone. "Approximation of optimal control problems with state constraints: estimates and applications". In: *Nonsmooth analysis and geometric methods in deterministic optimal control*. Springer, 1996, pp. 23–57.

[CF91] P. Cannarsa and H. Frankowska. "Some characterizations of optimal trajectories in control theory". In: *SIAM Journal on Control and optimization* 29.6 (1991), pp. 1322–1347.

[CS95] P. Cannarsa and C. Sinestrari. "Convexity properties of the minimum time function". In: *Calculus of Variations and Partial Differential Equations* 3.3 (1995), pp. 273–298.

[CS04] P. Cannarsa and C. Sinestrari. *Semiconcave functions, Hamilton-Jacobi equations, and optimal control*. Vol. 58. Springer Science & Business Media, 2004.

[Cap83] I. Capuzzo Dolcetta. "On a discrete approximation of the Hamilton-Jacobi equation of dynamic programming". In: *Appl. Math. Optim.* 10.4 (1983), pp. 367–377. ISSN: 0095-4616. DOI: 10.1007/BF01448394.

[CL90] I. Capuzzo-Dolcetta and P.-L. Lions. "Hamilton-Jacobi equations with state constraints". English. In: *Trans. Am. Math. Soc.* 318.2 (1990), pp. 643–683. ISSN: 0002-9947. DOI: 10.2307/2001324.

[Car+08] E. Carlini, M. Falcone, N. Forcadel, and R. Monneau. "Convergence of a generalized fast-marching method for an eikonal equation with a velocity-changing sign". In: *SIAM Journal on Numerical Analysis* 46.6 (2008), pp. 2920–2952.

[CFM11] E. Carlini, N. Forcadel, and R. Monneau. "A generalized fast marching method for dislocation dynamics". In: *SIAM journal on numerical analysis* 49.6 (2011), pp. 2470–2500.

[Cho+19] Y. T. Chow, J. Darbon, S. Osher, and W. Yin. "Algorithm for Overcoming the Curse of Dimensionality for State-Dependent Hamilton-Jacobi Equations". In: *Journal of Computational Physics* 387 (June 2019), pp. 376–409. ISSN: 00219991. DOI: 10.1016/j.jcp.2019.01.051.

[CCV14] Z. Clawson, A. Chacon, and A. Vladimirsky. "Causal domain restriction for Eikonal equations". In: *SIAM Journal on Scientific Computing* 36.5 (2014), A2478–A2505.

[CGQ04] G. Cohen, S. Gaubert, and J.-P. Quadrat. "Duality and Separation Theorems in Idempotent Semimodules". In: *Linear Algebra and Appl.* 379 (2004), pp. 395–422. DOI: 10.1016/j.laa.2003.08.010. eprint: math.FA/0212294.

[CGQ96] G. Cohen, S. Gaubert, and J.-P. Quadrat. "Kernels, images and projections in dioids". In: *Proceedings of WODES'96*. IEE Edinburgh. 1996, pp. 151–158.

[Cor+09] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009.

[CL84] M. Crandall and P. Lions. "Two approximations of solutions of Hamilton-Jacobi equations". In: *Mathematics of Computation* 43 (1984), pp. 1–19.

[CEL84] M. G. Crandall, L. C. Evans, and P.-L. Lions. "Some properties of viscosity solutions of Hamilton-Jacobi equations". In: *Transactions of the American Mathematical Society* 282.2 (1984), pp. 487–502.

[CL83] M. G. Crandall and P.-L. Lions. "Viscosity solutions of Hamilton-Jacobi equations". In: *Transactions of the American mathematical society* 277.1 (1983), pp. 1–42.

[CT80] M. G. Crandall and L. Tartar. "Some relations between nonexpansive and order preserving mappings". In: *Proceedings of the American Mathematical Society* 78.3 (1980), pp. 385–390.

[Cri09] E. Cristiani. “A fast marching method for Hamilton-Jacobi equations modeling monotone front propagations”. In: *Journal of Scientific Computing* 39.2 (2009), pp. 189–205.

[CF07] E. Cristiani and M. Falcone. “Fast semi-Lagrangian schemes for the Eikonal equation and applications”. In: *SIAM Journal on Numerical Analysis* 45.5 (2007), pp. 1979–2011.

[DDM23] J. Darbon, P. M. Dower, and T. Meng. “Neural network architectures using min-plus algebra for solving certain high-dimensional optimal control problems and Hamilton–Jacobi PDEs”. In: *Mathematics of Control, Signals, and Systems* 35.1 (2023), pp. 1–44.

[DM21] J. Darbon and T. Meng. “On some neural network architectures that can represent viscosity solutions of certain high dimensional Hamilton-Jacobi partial differential equations”. English. In: *J. Comput. Phys.* 425 (2021). Id/No 109907, p. 16. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2020.109907.

[DO16] J. Darbon and S. Osher. “Algorithms for overcoming the curse of dimensionality for certain Hamilton-Jacobi equations arising in control theory and elsewhere”. English. In: *Res. Math. Sci.* 3 (2016). Id/No 19, p. 26. ISSN: 2522-0144. DOI: 10.1186/s40687-016-0068-7.

[Del+06] D. Delling, P. Sanders, D. Schultes, and D. Wagner. “Highway Hierarchies Star.” In: *The Shortest Path Problem*. 2006, pp. 141–174.

[DI84] I. C. Dolcetta and H. Ishii. “Approximate solutions of the Bellman equation of deterministic control theory”. In: *Applied Mathematics and Optimization* 11.1 (1984), pp. 161–181.

[DKK21] S. Dolgov, D. Kalise, and K. K. Kunisch. “Tensor decomposition methods for high-dimensional Hamilton-Jacobi-Bellman equations”. English. In: *SIAM J. Sci. Comput.* 43.3 (2021), a1625–a1650. ISSN: 1064-8275. DOI: 10.1137/19M1305136.

[DM11] P. M. Dower and W. M. McEneaney. “A max-plus based fundamental solution for a class of infinite dimensional Riccati equations”. In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE. 2011, pp. 615–620.

[DM15] P. M. Dower and W. M. McEneaney. “A max-plus dual space fundamental solution for a class of operator differential Riccati equations”. In: *SIAM Journal on Control and Optimization* 53.2 (2015), pp. 969–1002.

[Dow18] P. M. Dower. “An Adaptive Max-Plus Eigenvector Method for Continuous Time Optimal Control Problems”. In: *Numerical Methods for Optimal Control Problems*. Ed. by M. Falcone, R. Ferretti, L. Grüne, and W. M. McEneaney. Vol. 29. Cham: Springer International Publishing, 2018, pp. 211–240. ISBN: 978-3-030-01958-7 978-3-030-01959-4. DOI: 10.1007/978-3-030-01959-4_10.

[FF94] M. Falcone and R. Ferretti. “Discrete time high-order schemes for viscosity solutions of Hamilton-Jacobi-Bellman equations”. In: *Numerische Mathematik* 67.3 (1994), pp. 315–344.

[Fal87] M. Falcone. “A numerical approach to the infinite horizon problem of deterministic control theory”. In: *Applied Mathematics and Optimization* 15.1 (1987), pp. 1–13.

[FF98] M. Falcone and R. Ferretti. “Convergence analysis for a class of high-order semi-Lagrangian advection schemes”. In: *SIAM Journal on Numerical Analysis* 35.3 (1998), pp. 909–940.

[FF14] M. Falcone and R. Ferretti. *Semi-Lagrangian approximation schemes for linear and Hamilton-Jacobi equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2014, pp. xii+319. ISBN: 978-1-611973-04-4.

[FGL94] M. Falcone, T. Giorgi, and P. Loreti. "Level sets of viscosity solutions: some applications to fronts and rendez-vous problems". In: *SIAM Journal on Applied Mathematics* 54.5 (1994), pp. 1335–1354.

[Fer02] R. Ferretti. "Convergence of semi-Lagrangian approximations to convex Hamilton–Jacobi equations under (very) large Courant numbers". In: *SIAM journal on numerical analysis* 40.6 (2002), pp. 2240–2253.

[FM00] W. H. Fleming and W. M. McEneaney. "A Max-Plus-Based Algorithm for a Hamilton–Jacobi–Bellman Equation of Nonlinear Filtering". In: *SIAM Journal on Control and Optimization* 38.3 (2000), pp. 683–710.

[FS06] W. H. Fleming and H. M. Soner. *Controlled Markov processes and viscosity solutions*. Vol. 25. Springer Science & Business Media, 2006.

[For09] N. Forcadel. "Comparison principle for a generalized fast marching method". In: *SIAM journal on numerical analysis* 47.3 (2009), pp. 1923–1951.

[FLG08] N. Forcadel, C. Le Guyader, and C. Gout. "Generalized fast marching method: applications to image segmentation". In: *Numerical Algorithms* 48.1 (2008), pp. 189–211.

[FV00] H. Frankowska and R. Vinter. "Existence of neighboring feasible trajectories: applications to dynamic programming for state-constrained optimal control problems". In: *Journal of Optimization Theory and Applications* 104 (2000), pp. 20–40.

[Fra93] H. Frankowska. "Lower semicontinuous solutions of Hamilton–Jacobi–Bellman equations". In: *SIAM Journal on Control and Optimization* 31.1 (1993), pp. 257–272.

[FP00] H. Frankowska and S. Plaskacz. "Semicontinuous solutions of Hamilton–Jacobi–Bellman equations with degenerate state constraints". In: *Journal of mathematical analysis and applications* 251.2 (2000), pp. 818–838.

[GMQ11] S. Gaubert, W. McEneaney, and Z. Qu. "Curse of dimensionality reduction in max-plus based approximation methods: Theoretical estimates and improved pruning algorithms". In: (2011), pp. 1054–1061.

[GLP15] P. Girardeau, V. Leclere, and A. B. Philpott. "On the convergence of decomposition methods for multistage stochastic convex programs". In: *Mathematics of Operations Research* 40.1 (2015), pp. 130–145.

[GO09] T. Goldstein and S. Osher. "The split Bregman method for L1-regularized problems". In: *SIAM journal on imaging sciences* 2.2 (2009), pp. 323–343.

[GT07] C. G. Gray and E. F. Taylor. "When action is not least". In: *American Journal of Physics* 75.5 (May 2007), pp. 434–458. DOI: 10.1119/1.2710480.

[Grü97] L. Grüne. "An adaptive grid scheme for the discrete Hamilton-Jacobi-Bellman equation". In: *Numerische Mathematik* 75 (1997), pp. 319–337.

[Gui20] V. Guigues. "Inexact cuts in stochastic dual dynamic programming". In: *SIAM Journal on Optimization* 30.1 (2020), pp. 407–438.

[Gui21] V. Guigues. "Inexact stochastic mirror descent for two-stage nonlinear stochastic programs". In: *Mathematical Programming* 187 (2021), pp. 533–577.

[HWZ18] C. Hermosilla, P. R. Wolenski, and H. Zidani. "The mayer and minimum time problems with stratified state constraints". In: *Set-Valued and Variational Analysis* 26.3 (2018), pp. 643–662.

[Kan+21] W. Kang, Q. Gong, T. Nakamura-Zimmerer, and F. Fahroo. "Algorithms of data generation for deep learning and feedback design: A survey". In: *Physica D: Nonlinear Phenomena* 425 (2021), p. 132955.

[KW17] W. Kang and L. C. Wilcox. "Mitigating the Curse of Dimensionality: Sparse Grid Characteristics Method for Optimal Feedback Control and HJB Equations". In: *Computational Optimization and Applications* 68.2 (Nov. 2017), pp. 289–315. ISSN: 0926-6003, 1573-2894. DOI: 10.1007/s10589-017-9910-0.

[Kir+18] M. R. Kirchner, R. Mar, G. Hewer, J. Darbon, S. Osher, and Y. T. Chow. "Time-Optimal Collaborative Guidance Using the Generalized Hopf Formula". In: *IEEE Control Systems Letters* 2.2 (Apr. 2018), pp. 201–206. ISSN: 2475-1456. DOI: 10.1109/LCSYS.2017.2785357.

[KM97] V. N. Kolokoltsov and V. P. Maslov. *Idempotent analysis and applications*. Kluwer Acad. Publisher, 1997.

[Kru75] S. Kružkov. "Generalized solutions of the Hamilton-Jacobi equations of eikonal type. I. Formulation of the problems; existence, uniqueness and stability theorems; some properties of the solutions". In: *Mathematics of the USSR-Sbornik* 27.3 (1975), p. 406.

[KD01] H. Kushner and P. Dupuis. *Numerical methods for stochastic control problems in continuous time*. Vol. 24. Springer Science & Business Media, 2001.

[Lak07] A. Lakhoua. "Méthode des éléments finis max-plus pour la résolution numérique de problèmes de commande optimale déterministe". PhD thesis. Université Paris 6, 2007.

[Mas87] V. P. Maslov. "Méthodes opératorielles". In: (1987).

[MDG08a] W. McEneaney, A. Deshpande, and S. Gaubert. "Curse-of-Complexity Attenuation in the Curse-of-Dimensionality-Free Method for HJB PDEs". In: *Proc. American Control Conf.* 2008.

[McE06] W. McEneaney. *Max-Plus Methods for Nonlinear Control and Estimation*. en. Systems & Control: Foundations & Applications. Boston: Birkhäuser-Verlag, 2006. ISBN: 978-0-8176-3534-3. DOI: 10.1007/0-8176-4453-9.

[McE09] W. M. McEneaney. "Convergence rate for a curse-of-dimensionality-free method for Hamilton–Jacobi–Bellman PDEs represented as maxima of quadratic forms". In: *SIAM Journal on Control and Optimization* 48.4 (2009), pp. 2651–2685.

[MDG08b] W. M. McEneaney, A. Deshpande, and S. Gaubert. "Curse-of-complexity attenuation in the curse-of-dimensionality-free method for HJB PDEs". In: *2008 American Control Conference*. IEEE. 2008, pp. 4684–4690.

[McE07] W. M. McEneaney. "A Curse-of-Dimensionality-Free Numerical Method for Solution of Certain HJB PDEs". en. In: *SIAM Journal on Control and Optimization* 46.4 (Jan. 2007), pp. 1239–1276. ISSN: 0363-0129, 1095-7138. DOI: 10.1137/040610830.

[MD15] W. M. McEneaney and P. M. Dower. "The Principle of Least Action and Fundamental Solutions of Mass-Spring and N-Body Two-Point Boundary Value Problems". In: *SIAM Journal on Control and Optimization* 53.5 (Jan. 2015), pp. 2898–2933. DOI: 10.1137/130921908.

[Mir14] J.-M. Mirebeau. "Anisotropic Fast-Marching on Cartesian Grids Using Lattice Basis Reduction". In: *SIAM Journal on Numerical Analysis* 52.4 (2014), pp. 1573–1599.

[Mir18] J.-M. Mirebeau. "Fast-marching methods for curvature penalized shortest paths". In: *Journal of Mathematical Imaging and Vision* 60.6 (2018), pp. 784–815.

[Mir19] J.-M. Mirebeau. "Riemannian Fast-Marching on Cartesian Grids, Using Voronoi's First Reduction of Quadratic Forms". In: *SIAM Journal on Numerical Analysis* 57.6 (2019), pp. 2608–2655.

[Mor39] A. P. Morse. "The behavior of a function on its critical set". In: *Annals of Mathematics* 40 (1939), pp. 62–70.

[NGK21] T. Nakamura-Zimmerer, Q. Gong, and W. Kang. "Adaptive deep learning for high-dimensional Hamilton-Jacobi-Bellman equations". English. In: *SIAM J. Sci. Comput.* 43.2 (2021), a1221–a1247. ISSN: 1064-8275. DOI: 10.1137/19M1288802.

[OSS22] M. Oster, L. Sallandt, and R. Schneider. "Approximating optimal feedback controllers of finite horizon control problems using hierarchical tensor formats". English. In: *SIAM J. Sci. Comput.* 44.3 (2022), b746–b770. ISSN: 1064-8275. DOI: 10.1137/21M1412190.

[OR20] G. Ottaviani and P. Reichenbach. *Tensor Rank and Complexity.* arXiv:2004.01492. 2020.

[PP91] M. V. Pereira and L. M. Pinto. "Multi-stage stochastic optimization applied to energy planning". In: *Mathematical programming* 52.1 (1991), pp. 359–375.

[PC08] G. Peyré and L. D. Cohen. "Heuristically driven front propagation for fast geodesic extraction". In: *International Journal for Computational Vision and Biomechanics* 1.1 (2008), pp. 55–67.

[Qu13] Z. Qu. "Nonlinear Perron-Frobenius Theory and Max-plus Numerical Methods for Hamilton-Jacobi Equations". en. PhD thesis. Ecole Polytechnique X, Oct. 2013.

[Qu14a] Z. Qu. "A Max-plus Based Randomized Algorithm for Solving a Class of HJB PDEs". In: *53rd IEEE Conference on Decision and Control.* Dec. 2014, pp. 1575–1580. DOI: 10.1109/CDC.2014.7039624.

[Qu14b] Z. Qu. "Contraction of Riccati Flows Applied to the Convergence Analysis of a Max-Plus Curse-of-Dimensionality–Free Method". In: *SIAM Journal on Control and Optimization* 52.5 (2014), pp. 2677–2706.

[RZ99] J.-P. Raymond and H. Zidani. "Pontryagin's principle for time-optimal problems". In: *Journal of Optimization Theory and Applications* 101.2 (1999), pp. 375–402.

[RZ98] J.-P. Raymond and H. Zidani. "Pontryagin's principle for state-constrained control problems governed by parabolic equations with unbounded controls". In: *SIAM Journal on Control and Optimization* 36.6 (1998), pp. 1853–1879.

[SS12] P. Sanders and D. Schultes. "Engineering highway hierarchies". In: *Journal of Experimental Algorithmics (JEA)* 17 (2012), pp. 1–1.

[Sar42] A. Sard. "The measure of the critical values of differentiable maps". In: *Bulletin of the American Mathematical Society* 48 (1942), pp. 883–890.

[Set96] J. A. Sethian. "A fast marching level set method for monotonically advancing fronts". In: *Proceedings of the National Academy of Sciences* 93.4 (1996), pp. 1591–1595.

[SV01] J. A. Sethian and A. Vladimirsky. "Ordered upwind methods for static Hamilton–Jacobi equations". In: *Proceedings of the National Academy of Sciences* 98.20 (2001), pp. 11069–11074.

[SV03] J. A. Sethian and A. Vladimirsky. "Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms". In: *SIAM Journal on Numerical Analysis* 41.1 (2003), pp. 325–363.

[Sha11] A. Shapiro. "Analysis of stochastic dual dynamic programming method". In: *European Journal of Operational Research* 209.1 (2011), pp. 63–72.

[SMK16] A. Shum, K. Morris, and A. Khajepour. "Convergence Rate for the Ordered Upwind Method". In: *Journal of Scientific Computing* 68.3 (2016), pp. 889–913. ISSN: 0885-7474.

[Son86a] H. M. Soner. "Optimal control with state-space constraint I". In: *SIAM Journal on Control and Optimization* 24.3 (1986), pp. 552–561.

[Son86b] H. M. Soner. "Optimal control with state-space constraint. II". In: *SIAM journal on control and optimization* 24.6 (1986), pp. 1110–1122.

[Sri+10] S. Sridharan, M. Gu, M. R. James, and W. M. McEneaney. "Reduced-complexity numerical method for optimal gate synthesis". In: *Phys. Rev. A* 82 (4 Oct. 2010), p. 042319.

[Tré05] E. Trélat. *Contrôle optimal: théorie & applications*. Vol. 36. Vuibert Paris, 2005.

[Tsi95] J. N. Tsitsiklis. "Efficient algorithms for globally optimal trajectories". In: *IEEE Transactions on Automatic Control* 40.9 (1995), pp. 1528–1538.

[Vla06] A. Vladimirsky. "Static PDEs for time-dependent control problems". In: *Interfaces and Free Boundaries* 8.3 (2006), pp. 281–300.

[YD21a] I. Yegorov and P. M. Dower. "Perspectives on characteristics based curse-of-dimensionality-free numerical approaches for solving Hamilton-Jacobi equations". English. In: *Appl. Math. Optim.* 83.1 (2021), pp. 1–49. ISSN: 0095-4616. DOI: 10.1007/s00245-018-9509-6.

[YD21b] I. Yegorov and P. M. Dower. "Perspectives on characteristics based curse-of-dimensionality-free numerical approaches for solving Hamilton-Jacobi equations". English. In: *Appl. Math. Optim.* 83.1 (2021), pp. 1–49. ISSN: 0095-4616. DOI: 10.1007/s00245-018-9509-6.

[ZAS19] J. Zou, S. Ahmed, and X. A. Sun. "Stochastic dual dynamic integer programming". In: *Mathematical Programming* 175 (2019), pp. 461–502.

List of Figures

***

List of Tables

***

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Hierarchies d'autoroutes pour les équations d'Hamilton-Jacobi-Bellman (HJB)

Mots clés : contrôle optimal, Équations aux dérivées partielles d'Hamilton-Jacobi-Bellman, Méthodes numériques, Plus court chemin, Algorithme Fast-marching, Méthode numérique tropicale (max-plus)

Résumé : Dans cette thèse, nous développons de nouvelles méthodes numériques pour résoudre des problèmes de contrôle optimal déterministe et les équations d'Hamilton-Jacobi-Bellman (HJB) du premier ordre associées. L'objectif principal est d'atténuer la malédiction de la dimension. Une idée commune à tous nos travaux est de se concentrer sur le calcul d'une ou plusieurs trajectoires optimales avec des conditions initiales et/ou finales fixées.
Dans la première partie, nous abordons les problèmes de temps minimum et la résolution des équations eikonales. Nous introduisons une méthode "fast marching" multi-niveaux, qui s'appuie sur des grilles imbriquées, permettant de rechercher les trajectoires optimales dans un voisinage tubulaire obtenu au moyen d'approximations dans des grilles grossières. Nous établissons la convergence et la complexité de notre algorithme. En outre, nous analysons un schéma Semi-Lagrangien pour les équations eikonales. Nous montrons la semiconcavité de la solution sous certaines conditions. Nous en déduisons un taux de convergence d'ordre 1 pour les schémas semi-discrétisés et entièrement discrétisés, le taux étant mesuré en terme de pas de temps pour le premier schéma et de pas en espace pour le second. Nous appliquons ces résultats pour obtenir le taux de convergence de la méthode "fast marching" et la complexité de notre méthode "fast marching" multi-niveaux.
Dans la deuxième partie, nous explorons les méthodes numériques tropicales pour les problèmes en horizon fini. Dans un premier travail, nous combinons les méthodes directes avec la méthode des éléments finis max-plus, ce qui conduit à une plus grande précision. Ensuite, nous combinons les concepts de la première partie avec la méthode numérique tropicale afin d'obtenir la meilleure borne de complexité sous des conditions plus générales. Dans un deuxième travail, nous introduisons un nouvel algorithme pour approximer numériquement la valeur en un état initial fixé, ainsi que la trajectoire optimale correspondante. Cet algorithme peut être vu comme une combinaison de la méthode numérique tropicale et de la méthode de programmation dynamique stochastique duale. Nous montrons que notre algorithme converge vers l'optimum global, dans le cas de problèmes semi-concaves.

Title : Highway hierarchies for Hamilton-Jacobi-Bellman (HJB) PDEs

Keywords : Optimal control, Hamilton-Jacobi-Bellman Partial Differential Equations, Numerical methods, Shortest path problem, Fast-marching method, Tropical (max-plus) numerical method

Abstract : In this thesis, we develop new numerical methods to solve deterministic optimal control problems and the associated first order Hamilton-Jacobi-Bellman (HJB) equations. Our primary aim is to mitigate the curse of dimensionality. One common idea in all our work is to focus on identifying one or several optimal trajectories with fixed initial and/or final conditions.
In the first part, we address minimum time problems and solve eikonal equations. We introduce a multilevel fast marching method, relying on nested grid approximations to search for optimal trajectories within a tubular neighborhood obtained from coarse grid approximations. We establish the convergence and computational complexity of our algorithm. Furthermore, we analyze a Semi-Lagrangian scheme for eikonal equations. We demonstrate semiconcavity of the solution under certain conditions. This allows us to establish a convergence rate of order 1 for both the semi-discretized and fully discretized schemes, with the rate being in terms of the time step for the former and the mesh step for the latter. We apply these results to get convergence rate of the fast marching method and complexity of our multilevel fast marching method.
In the second part, we explore tropical numerical methods for finite horizon problems. In a first work, we combine direct methods with the max-plus finite element method to achieve higher accuracy. Then, we combine the concepts of the first part with the tropical method to derive the best complexity bound under broader conditions. In a second work, we introduce a novel algorithm to numerically approximate the value at a fixed initial state, along with the optimal trajectory. This can be thought of as a combination of tropical numerical method and stochastic dual dynamic programming (SDDP) method. We show that our algorithm converges to the global optimum, in the case of semiconcave problems.

Institut Polytechnique de Paris
91120 Palaiseau, France