

Technische Universität München
Zentrum Mathematik

Fast Marching Methods for Indefinite and Anisotropic Metrics

Thomas Satzger

Vollständiger Abdruck der von der Fakultät für Mathematik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Martin Brokate
Prüfer der Dissertation: 1. Univ.-Prof. Dr. Folkmar Bornemann
2. Univ.-Prof. Dr. Daniel Matthes
3. o. Univ.-Prof. Dr. Karl Kunisch,
Technische Universität Graz, Österreich
(schriftliche Beurteilung)

Die Dissertation wurde am 31.08.2011 bei der Technischen Universität München eingereicht und durch die Fakultät für Mathematik am 29.11.2011 angenommen.

Acknowledgment

At this point I would like to thank all the people who supported me during the last years while writing this thesis. Without them this would have not been accomplished. First I express my deep gratitude to my supervisor Prof. Dr. F. Bornemann for the support, the freedom to research and the faith he put in me.

Further I must thank the Hanns Seidel Foundation for their support during the first two years.

Many thanks to all current and former members of the Chair of Scientific Computing for their help and suggestions and the fun during the applied game theory breaks. In particular I would like to thank Dr. Christian Ludwig for his support in \LaTeX , Java, Matlab, C, Linux and proof reading. Further thanks to Georg Wechsberger for the answering many questions about Mathematica, Matlab and \LaTeX .

Further I thank Fethallah Benmansour for fruitful discussions, the support of raw data and code used in example 3.4.6.

Finally I would like to thank Sarah Bluhme for proof reading.

Writing such a thesis is more than ordinary work; it also is an exiting period life and I thank my family, all the people I met and the friends I found for coming along with me and the many things I have experienced with them.

To all I just say "Vergelt's Gott!"

Contents

Acknowledgment	iii
Preface	ix
1 Hamilton-Jacobi equations	1
1.1 Introductory example	1
1.2 Continuous viscosity solutions	2
1.3 Static Hamilton-Jacobi equations	7
1.3.1 Definitions and basic properties	7
1.3.2 Eikonal equation	9
1.3.3 Optimal control	11
1.4 Discontinuous viscosity solutions	12
1.5 Front propagation	13
1.5.1 Level-Set approach	13
1.5.2 Basic properties	15
1.5.3 Non-empty interior	16
2 Generalized fast marching method	21
2.1 Fast marching method	21
2.1.1 FMM for structured grids	22
2.1.2 FMM for acute triangulations	23
2.1.3 FMM for general triangulations	25
2.1.4 FMM for triangulations and the anisotropic eikonal equation	25
2.2 GFMM on structured grids	26
2.2.1 GFMM	27
2.2.1.1 Preliminary definitions	27
2.2.1.2 Description of the algorithm	28
2.2.1.3 Properties	32
2.2.2 Modified GFMM	34
2.2.2.1 Preliminary definitions	34
2.2.2.2 Description of the algorithm	35
2.2.2.3 Properties	38
2.3 Generalized fast marching method for unstructured triangulations	40
2.3.1 First version of GFT	40
2.3.1.1 Definitions for GFT	40
2.3.1.2 Algorithm GFT	42
2.3.1.3 Comparison principle for GFT	43
2.3.1.4 Remarks	46

2.3.2	Modified generalized fast marching method for unstructured triangulations (mGFT)	47
2.3.2.1	Definition mGFT	47
2.3.2.2	Algorithm mGFT	48
2.3.2.3	Comments	50
2.3.3	Computational complexity	50
2.4	Comparison principle for mGFT	51
2.4.1	Comparison principle for θ^ϵ	54
2.4.2	Basic properties for mGFT	55
2.4.3	Comparison principle for the t_n time	59
2.4.4	Comparison principle for the \tilde{t}_n times: proof of proposition 2.4.12	69
2.5	Numerical Tests	76
2.5.1	Comments on the implementation and numerical tests	76
2.5.2	Numerical examples	78
2.5.2.1	Rotating line	78
2.5.2.2	Propagation of a circle	82
2.5.2.3	Comparison to FMM	84
2.5.2.4	Collapsing circles	85
2.5.2.5	Virtual updates	88
2.6	GFT for anisotropic speed metrics	89
2.6.1	Front propagation for anisotropic speed.	90
2.6.2	Generalized fast marching for anisotropic speed functions	91
2.6.3	Numerical example for GFA	93
2.7	Conclusion and further problems	95
2.7.1	Conclusion	95
2.7.2	Further variants of (m)GFMM	96
2.7.3	Theoretic results for anisotropic speed	96
2.7.4	Adaptivity	97
2.7.5	Applications for (m)GFT	97
2.7.6	Convergence of (m)GFT	97
3	Anisotropic fast marching on Cartesian grids	101
3.1	Introduction	101
3.2	The AFM algorithm	102
3.2.1	Motivation	102
3.2.2	Description of the AFM algorithm	102
3.2.3	Finding virtual triangles	103
3.2.4	Estimates for virtual triangles in the 4-point neighbourhood	105
3.2.5	Estimates for virtual triangles in the 8-point neighbourhood	107
3.2.6	Differences to [SKo4]	110
3.3	Restrictions for the plain fast marching method on cartesian grids	110
3.4	Numerical tests	111
3.4.1	Implementational issues	111
3.4.2	Constant speed	112
3.4.3	Monotonicity of AFM	112
3.4.4	Characteristics	114

3.4.5	Three regions	115
3.4.6	Spiral	116
3.5	Conclusion	117
Symbols and notation		121
Bibliography		123

Preface

In this thesis we discuss two different aspects of the fast marching method (FMM). The FMM was introduced in [Set96a] and it solves the static Hamilton-Jacobi-equation

$$\|\nabla u(x)\|c(x) = 1 \text{ on } \Omega, \quad u|_{\partial\Omega=0} \quad (0.1)$$

with a continuous speed function $c > 0$.

The first aspect is to use indefinite speed functions, that means we allow that the speed c changes its sign. Such indefinite speed functions makes only sense if we regard the corresponding initial value problem of (0.1), that is

$$\begin{cases} \theta_t(t, x) - c(t, x) |\theta_x(t, x)| = 0 & \text{on } (0, T) \times \mathbb{R}^d \\ \theta(0, \cdot) = 1_{\Omega_0} - 1_{\Omega_0^c}, \end{cases} \quad (0.2)$$

A first approach for the numerical solution of such an equation on a regular lattice in \mathbb{R}^d was presented in [CFFMo8] wherein convergence of the proposed algorithm was also proven. In [CFFMo8] this method has been refined such that we also have a discrete comparison principle. In chapter 2 we will, after introducing the underlying theory in chapter 1, present a generalization of these methods to unstructured grids.

The main idea behind this generalization to unstructured grid is that the (m)GFMM locally solves the eikonal equation (0.1) using the same upwind discretization as in the FMM. We transport the meaning of the grid and the discretization to the unstructured grid and then we modify the algorithm in the right way. This is all described in section 2.3. As the GFMM the GFT can be modified such that it copes with an discrete comparison principle – at least if the underlying triangulation is acute. Section 2.4 is dedicated to the proof of this result. Of course we carried out some numerical experiments to validate whether these methods work in practice, see section 2.5.

Using unstructured grids for the FMM one can not only use a scalar speed function but one can also solve the anisotropic eikonal equation as in [BR06, Section 7]. This extension to anisotropic metrics is transferred to the generalized fast marching method from above in section 2.6. On the one hand this works fine in a heuristic and also computational sense but there are arise open problems in theory.

This leads to the second aspect, namely a fast marching method for the anisotropic eikonal equation. Such a fast marching method for unstructured triangulation is already presented in [Ras07]. In chapter 3 we apply this method to the special case of a Cartesian grid. The key ingredient for such a method is the concept of virtual triangles as introduced in [KS98]. We give a detailed analysis for the virtual triangles and show that some new problems arise. Further we show the connection to other algorithms used to solve the anisotropic eikonal equation and compare the numerical results of these methods.

1 Hamilton-Jacobi equations

In this chapter we establish the theory of Hamilton-Jacobi equations (HJE). In particular we introduce the concept of viscosity solutions, which is one of the most important concepts in this thesis.

This chapter is organized as follows. In the first section we give an example which shows us that the concept of classical solutions for partial differential equations (PDEs) is not sufficient, and that we therefore have to look for a more appropriate theory of solutions. In section 1.2 we introduce the concept of viscosity solutions and also state some basic properties. In section 1.3 we take a closer look at the static HJE because we need this special case to understand the fast marching method (FMM) which is a key ingredient in the construction of the GFMM in chapter 2. In addition we will also use the concept of static HJE in chapter 3. Furthermore we also look at equations of eikonal type and show a connection to optimal control problems. In section 1.4 we generalize the definition of viscosity solutions for discontinuous functions because the generalized fast marching method (GFMM) works with them. In the last section 1.5 we illustrate the connection between the eikonal equation and front propagation with some examples.

1.1 Introductory example

We start with the scalar eikonal equation

$$\begin{aligned} |u'(x)| - 1 &= 0 \text{ in }]-1, 1[\\ u(-1) &= u(1) = 0 \end{aligned} \tag{1.1.1}$$

For this equation a classical solution $u \in \mathcal{C}^1([-1, 1])$ does not exist. Due to the mean value theorem there should exist a point $\xi \in]-1, 1[$ with $u'(\xi) = 0$ which contradicts the condition that $|u'(x)| = 1$ for all $x \in]-1, 1[$. Thus the notion of classical solutions is not useful here because this concept is too strict to obtain a solution. However, all the functions $u_n(x)$ defined on $[-1, 1]$ by

$$\begin{aligned} u_0(x) &= 1 - |x| \\ u_j(x) &= \frac{1}{2^j} - \left| \frac{1}{2^j} - u_{j-1}(x) \right| \quad \text{for } j \in \mathbb{N} \end{aligned} \tag{1.1.2}$$

are weak¹ solutions of (1.1.1).

Thus we see that this simple equation has on the one hand no classical solution but on the other hand infinitely many weak solutions, so this situation is not satisfactory.

¹Weak solutions means here that the function u is Lipschitz-continuous and therefore differentiable almost everywhere by Rademacher's theorem.

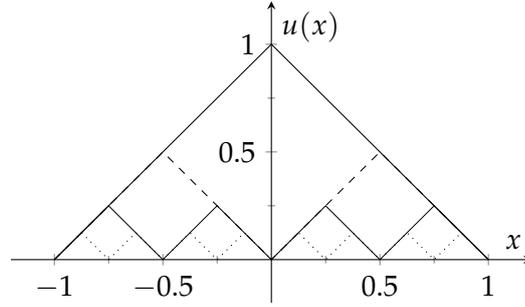


Figure 1.1.1: Different weak solutions for (1.1.1).

There exists a physical interpretation of (1.1.1) which has a unique solution. One can interpret u as the first arrival time of a wave which expands with constant speed 1 starting at the boundary. The function $u_0(x) = 1 - |x|$ is the unique solution for this interpretation. We will see later on, that u_0 is the so called viscosity solution.

1.2 Continuous viscosity solutions

We will now introduce the notion of viscosity solutions and state some basic properties. This notion was first introduced in [CL81] and developed and discussed further in [Lio82], [CL83], [CEL84] and [CL84]. A good overview of this topic can also be found in [CIL92].

We start here with the initial-value problem and we use this setting for the front propagation which is described in section 1.5. Let us regard the initial-value problem for the Hamilton-Jacobi equation

$$\begin{aligned} u_t(t, x) + H(t, x, Du(t, x)) &= 0 \text{ in } (0, \infty) \times \mathbb{R}^d \\ u &= g \text{ on } \{t = 0\} \times \mathbb{R}^d. \end{aligned} \tag{1.2.1}$$

Here the Hamiltonian $H : \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ and the initial function $g : \mathbb{R}^d \rightarrow \mathbb{R}$ is given. The unknown here is $u : [0, \infty) \times \mathbb{R}^d \rightarrow \mathbb{R}$, $u = u(t, x)$ with $Du = D_x u = (u_{x_1}, \dots, u_{x_n})$ as the spatial derivative. We will write $H = H(t, x, p)$, so that p is the placeholder for the spatial gradient.

In the next definitions and theorems we follow the ideas and methods of [Eva10], to generalize the results for Hamiltonians of the form $H(t, x, Du)$. Hamiltonians of the form $H(x, Du)$ are treated in [Eva10].

DEFINITION 1.2.1 (according to [Eva10, Chapter 10.1]):

We assume that $H(t, x, p)$ is continuous. A bounded, uniformly continuous function u is called a viscosity solution of the initial-value problem (1.2.1) for the Hamilton-Jacobi equation provided:

- (i) $u = g$ on $\{t = 0\} \times \mathbb{R}^d$,

(ii) for each $v \in C^\infty([0, \infty) \times \mathbb{R}^d)$,

$$\begin{cases} \text{if } u - v \text{ has a local maximum at a point } (t_0, x_0) \in (0, \infty) \times \mathbb{R}^d, \text{ then} \\ v_t(t_0, x_0) + H(t_0, x_0, Dv(t_0, x_0)) \leq 0, \end{cases} \quad (1.2.2)$$

and

$$\begin{cases} \text{if } u - v \text{ has a local minimum at a point } (t_0, x_0) \in (0, \infty) \times \mathbb{R}^d, \text{ then} \\ v_t(t_0, x_0) + H(t_0, x_0, Dv(t_0, x_0)) \geq 0, \end{cases} \quad (1.2.3)$$

A function u that satisfies the condition (1.2.2) it is called a subsolution and if it satisfies (1.2.3) it is called a supersolution.

REMARK 1.2.2:

In the literature there are different ways to define the viscosity solutions. Some people (e.g [BCD97, Definition II.1.1] for static HJE) assume that the test-function v only has to be differentiable. This leads to an equivalent formulation because, loosely speaking, the C^∞ -functions are dense in the set of C^1 -functions.

Some definitions use strict extrema which also leads to an equivalent definition because if $u - v$ attains a local extremum at (t_0, x_0) one can construct $\tilde{v} = v \pm (|x - x_0| + |t - t_0|)$ such that $u - \tilde{v}$ has a strict extremum.

A priori we do not know whether there is a connection between a classical solution, if it exists, and a viscosity solution of (1.2.1). This question will be answered in the following lemma and theorem.

LEMMA 1.2.3 (Touching by a C^1 function [Eva10, chapter 10.1]):

Assume $u : \mathbb{R}^d \rightarrow \mathbb{R}$ is continuous and is also differentiable at some point x_0 . Then there exists a function $v \in C^1(\mathbb{R}^d)$ such that

$$u(x_0) = v(x_0) \quad (1.2.4)$$

and

$$u - v \text{ has a strict local maximum at } x_0. \quad (1.2.5)$$

The proof can be found in [Eva10, Chapter 10.1].

THEOREM 1.2.4 (Consistency of viscosity solutions (according to [Eva10, chapter 10.1])): Let u be a viscosity solution of (1.2.1), and suppose u is differentiable at some point $(t_0, x_0) \in (0, \infty) \times \mathbb{R}^d$. Then

$$u_t(t_0, x_0) + H(t_0, x_0, Du(t_0, x_0)) = 0.$$

Proof. In the first step we apply the lemma 1.2.3 (in \mathbb{R}^{d+1}) to the function u . We know that there exists a function v in C^1 such that $u - v$ has a strict maximum at (t_0, x_0) .

We set $v^\varepsilon := \eta_\varepsilon * v$ with a mollifier² η_ε in the $1 + d$ variables (t, x) . Then we obtain

$$\begin{cases} v^\varepsilon \rightarrow v \\ Dv^\varepsilon \rightarrow Dv \\ v_t^\varepsilon \rightarrow v_t \end{cases} \quad \text{uniformly near } (t_0, x_0) \quad (1.2.6)$$

²A mollifier is a smooth function η on \mathbb{R}^d which is compactly supported with $\int_{\mathbb{R}^d} \eta(x) dx = 1$ and for $\varepsilon \rightarrow 0$ we have $|\eta_\varepsilon * f - f|_\infty \rightarrow 0$ for $f \in C^\infty$ where $\eta_\varepsilon(x) := \varepsilon^{-d} \eta(x/\varepsilon)$.

and therefore we conclude that $u - v_\varepsilon$ has a maximum at $(t_\varepsilon, x_\varepsilon)$ with

$$\lim_{\varepsilon \rightarrow 0} (t_\varepsilon, x_\varepsilon) = (t_0, x_0). \quad (1.2.7)$$

Due to the definition of viscosity solutions, we have

$$v_t^\varepsilon(t_\varepsilon, x_\varepsilon) + H(t_\varepsilon, x_\varepsilon, Dv^\varepsilon(t_\varepsilon, x_\varepsilon)) \leq 0.$$

Now let $\varepsilon \rightarrow 0$ and we obtain with (1.2.6) and (1.2.7)

$$v_t(t_0, x_0) + H(t_0, x_0, Dv(t_0, x_0)) \leq 0. \quad (1.2.8)$$

As $u - v$ has a strict local maximum, and u is differentiable at (t_0, x_0) we know

$$Du(t_0, x_0) = Dv(t_0, x_0) \text{ and } u_t(t_0, x_0) = v_t(t_0, x_0).$$

Using the above with (1.2.8) we get

$$u_t(t_0, x_0) + H(t_0, x_0, Du(t_0, x_0)) \leq 0.$$

The other inequality

$$u_t(t_0, x_0) + H(t_0, x_0, Du(t_0, x_0)) \geq 0.$$

can be proofed analogous by using the lemma 1.2.3 to $-u$ to find a C^1 function v such that $u - v$ has a strict minimum at (t_0, x_0) . \square

The other way round we will now show that a classical solution u is also a viscosity solution.

THEOREM 1.2.5 (Classical solutions are viscosity solutions (according to [Eva10, chapter 10.1])):

Let $u \in C^1([0, \infty) \times \mathbb{R}^d) \cup BUC([0, \infty) \times \mathbb{R}^d)$ be a classical solution of (1.2.1). Then u is also a viscosity solution.

Proof. Let $v \in C^\infty([0, \infty) \times \mathbb{R}^d)$ such that $u - v$ has a local maximum at (t_0, x_0) . Thus we have

$$\begin{cases} Du(t_0, x_0) = Dv(t_0, x_0) \text{ and} \\ u_t(t_0, x_0) = v_t(t_0, x_0). \end{cases}$$

Therefore we gain

$$0 = u_t(t_0, x_0) + H(t_0, x_0, Du(t_0, x_0)) = v_t(t_0, x_0) + H(t_0, x_0, Dv(t_0, x_0)).$$

The other case that $u - v$ attains a local minimum at (t_0, x_0) is treated analogously. Thus u is also a viscosity solution of (1.2.1). \square

The next step is to look whether there exists an unique solution of (1.2.1). Therefore we suppose that the Hamiltonian H satisfies the following conditions of Lipschitz continuity:

$$\begin{cases} |H(t, x, p) - H(t, x, q)| \leq C |p - q| \\ |H(t, x, p) - H(t, y, p)| \leq C |x - y| (1 + |p|) \\ |H(t, x, p) - H(s, x, p)| \leq C |t - s| (1 + |p|) \end{cases} \quad (1.2.9)$$

THEOREM 1.2.6 (Uniqueness of viscosity solutions (according to [Eva10, chapter 10.2])): *Under the assumptions (1.2.9) there exists at most one viscosity solution of (1.2.1).*

Proof. We split the proof into six steps.

1. Assume u and \tilde{u} are both viscosity solutions with the same initial conditions, but

$$\sup_{[0,\infty) \times \mathbb{R}^d} (u - \tilde{u}) =: \sigma > 0. \quad (1.2.10)$$

We choose $0 < \varepsilon, \lambda < 1$ and set

$$\Phi(x, y, t, s) := u(t, x) - \tilde{u}(s, y) - \lambda(t + s) - \frac{1}{\varepsilon^2} \left(|x - y|^2 + (t - s)^2 \right) - \varepsilon(|x|^2 + |y|^2) \quad (1.2.11)$$

for $x, y \in \mathbb{R}^d$ and $t, s \geq 0$. The functions u and \tilde{u} are bounded due to definition 1.2.1 and thus there exists a point $(x_0, y_0, t_0, s_0) \in [0, \infty)^2 \times \mathbb{R}^{2d}$ such that

$$\Phi(x_0, y_0, t_0, s_0) = \max_{[0,\infty)^2 \times \mathbb{R}^{2d}} \Phi(x, y, t, s). \quad (1.2.12)$$

2. Now fix $0 < \varepsilon, \lambda < 1$ so small such that (1.2.10) implies

$$\Phi(x_0, y_0, t_0, s_0) \geq \sup_{(t, xt) \in [0,\infty) \times \mathbb{R}^d} \Phi(x, x, t, t) \geq \frac{\sigma}{2}. \quad (1.2.13)$$

We have furthermore $\Phi(x_0, y_0, t_0, s_0) \geq \Phi(0, 0, 0, 0)$ and therefore

$$\begin{aligned} & u(t_0, x_0) - \tilde{u}(s_0, y_0) - u(0, 0) + \tilde{u}(0, 0) \\ & \geq \lambda(t_0 + s_0) + \frac{1}{\varepsilon^2} \left(|x_0 - y_0|^2 + (t_0 - s_0)^2 \right) + \varepsilon(|x_0|^2 + |y_0|^2). \end{aligned} \quad (1.2.14)$$

Due to the boundedness of u and \tilde{u} we deduce

$$|x_0 - y_0|, |t_0 - s_0| = \mathcal{O}(\varepsilon) \text{ as } \varepsilon \rightarrow 0. \quad (1.2.15)$$

and furthermore we get from (1.2.14) that $\varepsilon(|x_0|^2 + |y_0|^2) = \mathcal{O}(1)$. So we get with help of the AM-GM inequality

$$\begin{aligned} \varepsilon(|x_0| + |y_0|) &= \varepsilon^{1/4} \varepsilon^{3/4} (|x_0| + |y_0|) \\ &\leq \varepsilon^{1/2} + C\varepsilon^{3/2} (|x_0|^2 + |y_0|^2) \\ &\leq C\varepsilon^{1/2} \end{aligned}$$

and thus

$$\varepsilon(|x_0| + |y_0|) = \mathcal{O}(\varepsilon^{1/2}). \quad (1.2.16)$$

3. Since $\Phi(x_0, y_0, t_0, s_0) \geq \Phi(x_0, x_0, t_0, t_0)$ we get as in (1.2.14)

$$\begin{aligned} & u(t_0, x_0) - \tilde{u}(s_0, y_0) - \lambda(t_0 + s_0) - \frac{1}{\varepsilon^2} \left(|x_0 - y_0|^2 + (t_0 - s_0)^2 \right) - \varepsilon(|x_0|^2 + |y_0|^2) \\ & \geq u(t_0, x_0) - \tilde{u}(t_0, x_0) - 2\lambda t_0 - 2\varepsilon|x_0|^2. \end{aligned}$$

Therefore we have

$$\tilde{u}(t_0, x_0) - \tilde{u}(s_0, y_0) + \lambda(t_0 - s_0) + \varepsilon(x_0 + y_0)(x_0 - y_0) \geq \frac{1}{\varepsilon^2} \left(|x_0 - y_0|^2 + (t_0 - s_0)^2 \right).$$

Due to the uniform continuity of \tilde{u} given in definition 1.2.1 and (1.2.16) plus (1.2.15) we obtain that

$$|x_0 - y_0|, |t_0 - s_0| = o(\varepsilon). \quad (1.2.17)$$

which is stronger than (1.2.15).

4. We denote by $\omega(\cdot)$ the modulus of continuity of u . This is a function with

$$|u(t, x) - u(s, y)| \leq \omega(|x - y| + |t - s|)$$

for all $x, y \in \mathbb{R}^d$, $t, s \in [0, \infty)$ and $\lim_{r \rightarrow 0} \omega(r) = 0$. Similarly $\tilde{\omega}(\cdot)$ denotes the modulus of continuity of \tilde{u} .

Equation (1.2.13) with (1.2.17) and the initial condition implies

$$\begin{aligned} \frac{\sigma}{2} &\leq u(t_0, x_0) - \tilde{u}(s_0, y_0) \\ &= u(x_0, t_0) - u(x_0, 0) + u(x_0, 0) - \tilde{u}(x_0, 0) + \tilde{u}(x_0, 0) - \tilde{u}(x_0, t_0) + \tilde{u}(x_0, t_0) - \tilde{u}(s_0, y_0) \\ &\leq \omega(t_0) + \tilde{\omega}(t_0) + \tilde{\omega}(o(\varepsilon)). \end{aligned} \quad (1.2.18)$$

Now take $\varepsilon > 0$ to be so small that the above implies

$$\frac{\sigma}{4} \leq \omega(t_0) + \tilde{\omega}(t_0).$$

Thus we know $t_0 \geq \mu > 0$ for some constant μ . Likewise we can deduce that $s_0 \geq \mu > 0$.

5. In view of (1.2.12) we can see that the mapping $(t, x) \mapsto \Phi(x, y_0, t, s_0)$ has a maximum at (t_0, x_0) . Using the function

$$v(t, x) := \tilde{u}(s_0, y_0) + \lambda(t + s_0) + \frac{1}{\varepsilon^2} (|x - y_0|^2 + (t - s_0)^2) + \varepsilon(|x|^2 + |y_0|^2)$$

we conclude with (1.2.11) that $u - v$ has a maximum at (x_0, t_0) .

Since u is a viscosity solution of (1.2.1) we conclude that

$$v_t(t_0, x_0) + H(t_0, x_0, Dv(t_0, x_0)) \leq 0.$$

Therefore we get

$$\lambda + \frac{2(t_0 - s_0)}{\varepsilon^2} + H\left(t_0, x_0, \frac{2(x_0 - y_0)}{\varepsilon^2} + 2\varepsilon x_0\right) \leq 0 \quad (1.2.19)$$

On the other hand we regard now the mapping $(s, y) \mapsto -\Phi(x_0, y, t_0, s)$ with

$$\tilde{v}(s, y) := u(t_0, x_0) - \lambda(t_0 + s) - \frac{1}{\varepsilon^2} (|x_0 - y|^2 + (t_0 - s)^2) - \varepsilon(|x_0|^2 + |y|^2).$$

As above we see that $\tilde{u} - \tilde{v}$ has a minimum at (s_0, y_0) . Since \tilde{u} is also a viscosity solution of (1.2.1), we know that

$$\tilde{v}_t(s_0, y_0) + H(s_0, y_0, D\tilde{v}(s_0, y_0)) \geq 0.$$

So we get

$$-\lambda + \frac{2(t_0 - s_0)}{\varepsilon^2} + H\left(s_0, y_0, \frac{2(x_0 - y_0)}{\varepsilon^2} - 2\varepsilon y_0\right) \leq 0. \quad (1.2.20)$$

6. Subtracting (1.2.19) and (1.2.20) we get

$$2\lambda \leq H\left(s_0, y_0, \frac{2(x_0 - y_0)}{\varepsilon^2} - 2\varepsilon y_0\right) - H\left(t_0, x_0, \frac{2(x_0 - y_0)}{\varepsilon^2} + 2\varepsilon x_0\right). \quad (1.2.21)$$

Now we employ the assumption (1.2.9) on H to get³

$$\lambda \leq C \left((|t_0 - s_0| + |x_0 - y_0|) \left(1 + \left| \frac{2(x_0 - y_0)}{\varepsilon^2} - 2\varepsilon y_0 \right| \right) + 2\varepsilon |x_0 + y_0| \right) \quad (1.2.22)$$

Now we use the estimates (1.2.16) and (1.2.17) in the above inequality. Then we let $\varepsilon \rightarrow 0$ to discover $0 < \lambda \leq 0$, which leads to a contradiction. □

1.3 Static Hamilton-Jacobi equations

In this section we introduce the concept of Hamilton-Jacobi equations for Dirichlet problems and the corresponding notion of viscosity solutions. The definitions and basic properties are stated in subsection 1.3.1. In subsection 1.3.2 we look at the special case of equations of eikonal type, which will be the most important case for the following chapters. This section is then closed by subsection 1.3.3 where we briefly show the connection between the eikonal equation and problems of optimal control.

1.3.1 Definitions and basic properties

First we will take a close look at the static Hamilton-Jacobi equations.

Let $\Omega \subset \mathbb{R}^d$ denote an open set. Furthermore assume $H : \overline{\Omega} \times \mathbb{R}^d \rightarrow \mathbb{R}$ is at least continuous and $g : \partial\Omega \rightarrow \mathbb{R}$ is a known continuous function with the boundary data. We consider the Dirichlet problem of the form

$$\begin{aligned} H(x, Du(x)) &= 0 \text{ in } \Omega \\ u &= g \text{ on } \partial\Omega. \end{aligned} \quad (1.3.1)$$

The next definition follows definition 1.2.1.

³With the assumptions (1.2.9) we get

$$\begin{aligned} |H(t, x, p) - H(s, y, q)| &= |H(t, x, p) - H(s, x, p) + H(s, x, p) - H(s, y, p) + H(s, y, p) - H(s, y, q)| \\ &\leq C|t - s|(1 + |p|) + C|x - y|(1 + |p|) + C|p - q| \\ &= C(|t - s| + |x - y|)(1 + |p|) + |p - q| \end{aligned}$$

DEFINITION 1.3.1 ([CEL84]):

A bounded, uniformly continuous function u is called a viscosity solution of the Dirichlet problem (1.3.1) for the static Hamilton-Jacobi equation provided:

(i) $u = g$ on $\partial\Omega$,

(ii) for each $v \in C^\infty(\Omega)$,

$$\begin{cases} \text{if } u - v \text{ has a local maximum at a point } x_0 \in \Omega, \text{ then} \\ H(x_0, Dv(x_0)) \leq 0, \end{cases} \quad (1.3.2)$$

and

$$\begin{cases} \text{if } u - v \text{ has a local minimum at a point } x_0 \in \Omega, \text{ then} \\ H(x_0, Dv(x_0)) \geq 0. \end{cases} \quad (1.3.3)$$

A function u that fulfills only condition (1.3.2) is called a subsolution. On the other hand a function u that satisfies (1.3.3) is called a supersolution.

Remark 1.2.2 also applies to this definition. Furthermore there is also a general connection between the initial-value problem and the Dirichlet problem.

REMARK 1.3.2 ([BCD97, chapter II.1, p. 26]):

We consider the initial-value problem with the HJE

$$u_t + H(t, x, D_x u(t, x)) = 0 \text{ in } (0, \infty) \times \mathbb{R}^d.$$

We can write this equation as a Dirichlet problem with the HJE

$$\tilde{H}(y, Du(y)) = 0 \text{ in } \mathbb{R}^{d+1}.$$

by setting $y = (t, x)$ and $\tilde{H}(y, p) = p_{d+1} + H(t, x, (p_1, \dots, p_d))$ with $p = (p_1, \dots, p_{d+1}) \in \mathbb{R}^{d+1}$.

For the eikonal equation there is another kind of relation between the formulation as an initial-value problem and a Dirichlet problem. We will look at this in section 1.5. We want to mention that for the Dirichlet formulation of a HJE there also is a consistency result [BCD97, Proposition 1.9] like theorem 1.2.4. For the fast marching method (see section 2.1) we need some more assumptions on the Hamiltonian. To gain more valuable results we will from now on only consider Hamiltonians H with the following properties: (see [Lio82, Section 5.2])

(H1) (Continuity) $H \in C(\bar{\Omega} \times \mathbb{R}^d)$.

(H2) (Convexity) $p \mapsto H(x, p)$ is convex for all $x \in \bar{\Omega}$.

(H3) (Coercivity) $H(x, p) \rightarrow \infty$ as $|p| \rightarrow \infty$, uniformly in $x \in \bar{\Omega}$.

(H4) (Compatibility of the Hamiltonian) $H(x, 0) \leq 0$ for all $x \in \bar{\Omega}$.

The existence of a solution of (1.3.1) requires a further condition for the boundary data, namely

(H5) (Compatibility of Dirichlet data) $g(x) - g(y) \leq \delta(x, y)$ for all $x, y \in \partial\Omega$.

Here δ denotes the optical distance between the points x and y . It is defined under the assumptions (H1)-(H4) by

$$\delta(x, y) = \inf \left\{ \int_0^1 \rho(\xi(t), -\xi'(t)) dt : \xi \in C^{0,1}([0, 1], \overline{\Omega}), \xi(0) = x, \xi(1) = y \right\} \quad (1.3.4)$$

with $\rho(x, q) = \max_{H(x,p)=0} \langle p, q \rangle$

at which $C^{0,1}([0, 1], \overline{\Omega})$ denotes the set of Lipschitz-continuous functions from $[0, 1]$ to $\overline{\Omega}$.

The optical distance δ fulfills the properties

$$\delta(x, x) = 0, \quad 0 \leq \delta(x, z) \leq \delta(x, y) + \delta(y, z).$$

If in addition H is symmetric with respect to p , this means $H(x, p) = H(x, -p)$, then $\delta(x, y) = \delta(y, x)$ and therefore δ is a pseudo-metric. These properties are proven in [Ras07, Lemma 1.21] and a similar construction was already used in [Lio82, Theorem 5.1].

In the next theorem we see, that there is a solution of the Dirichlet problem and it can be written with the help of the optical distance.

THEOREM 1.3.3 ([Lio82, Theorem 5.3]):

Assume (H1)-(H4). The Dirichlet problem (1.3.1) has a viscosity solution u if and only if the boundary condition satisfies the compatibility condition (H5). A specific viscosity solution is then given by the Hopf-Lax formula

$$u(x) = \inf_{y \in \partial\Omega} (g(y) + \delta(x, y)) \quad (1.3.5)$$

To get a uniqueness result we must enforce a slightly stronger condition

(H4') $H(x, 0) < 0$ for all $x \in \Omega$.

With the following comparison result one can gain the uniqueness of the solution given in equation (1.3.5).

THEOREM 1.3.4 ([Ish87, Theorem 1]):

Assume⁴ (H1)-H(3) and H(4') and let u, v be viscosity sub- and supersolutions of (1.3.1), respectively. If $u \leq v$ on $\overline{\Omega}$ then $u \leq v$ in $\overline{\Omega}$.

1.3.2 Eikonal equation

In chapter 2 and 3 we deal with equations of eikonal type. Therefore we introduce this notion in this subsection. We will only consider from now on equations of eikonal type, this means we regard equations of the form

$$H(x, Du(x)) = 0, \quad x \in \Omega, \quad \text{with } H(x, p) = E(x, p) - 1 \quad (1.3.6)$$

where the function $E(x, p)$ fulfills the conditions

⁴The assumptions we use here are stricter than the one Ishii uses in his paper.

(E1) (Continuity) $E \in \mathcal{C}(\overline{\Omega} \times \mathbb{R}^d)$

(E2) (Convexity) $p \mapsto E(x, p)$ is convex for all $x \in \overline{\Omega}$

(E3) (Homogeneity) $E(x, tp) = tE(x, p)$ for all $x \in \overline{\Omega}$, $p \in \mathbb{R}^d$, $t > 0$

(E4) (Positivity) $E(x, p) > 0$ for all $x \in \overline{\Omega}$, $p \neq 0$.

It is easy to see that a function H which is defined by (1.3.6) also fulfills the conditions (H1)-(H4). Thus the uniqueness and existence theorem also holds for the equation of eikonal type.

We want to remark here that all functions H that suffice the conditions (H1)-(H3) and (H4'), can be written in the form (1.3.6) with a function E that fulfills the conditions (E1)-(E4). This result is formally stated and proved in [Ras07, Theorem 1.29].

Now we review the example from section 1.1. We show that the function u_0 in (1.1.2) is really the viscosity solution of the simple eikonal equation (1.1.1).

EXAMPLE 1.3.5:

In section 1.1 we defined $u_0(x) = 1 - |x|$ and we proposed that this is a viscosity solution of (1.1.1). To confirm that u_0 solves the equation we first check, whether it is a subsolution.

The function u_0 is differentiable in $] -1, 0[\cup] 0, 1[$ and there it fulfills the equation $|u_0'(x)| - 1 = 0$. Therefore u_0 is due to theorem 1.2.5 a viscosity solution there.

Now let us look at the critical point 0. Let $v \in \mathcal{C}^\infty(]-1, 1[)$ such that $u_0 - v$ attains at 0 a strict local maximum. This means that there is a $\delta > 0$, such that for all $|x| < \delta$ we have $(u_0 - v)(x) < (u_0 - v)(0)$. We conclude that for all $0 < x < \delta$ we gain $-\frac{v(x)-v(0)}{x} < 1$ and vice versa we get for all $-\delta < x < 0$ the inequality $\frac{v(x)-v(0)}{x} < 1$. Thus we know $|v'(0)| \leq 1$ and therefore u is a viscosity subsolution in $] -1, 1[$ by definition 1.3.1.

Now we show that u_0 is also a supersolution. For all points $x \in] -1, 0[\cup] 0, 1[$ we already know that u_0 is also a supersolution.

We now take a closer look at the point 0. We should take a function $v \in \mathcal{C}^\infty(]-1, 1[)$ such that $u_0 - v$ has a strict local minimum at 0. The point is that there is no such function v with this property because u_0 is not differentiable in the point 0 and u_0 is concave in a neighbourhood of 0. Thus u_0 is also a supersolution and therefore a viscosity solution.

In the last step we look at the boundary condition. u_0 satisfies the boundary condition $u_0(\pm 1) = 0$ and obviously u_0 is uniformly continuous. Thus u_0 is a viscosity solution of (1.1.1).

EXAMPLE 1.3.6:

Let Ω be a bounded and open set in \mathbb{R}^d and $c : \Omega \rightarrow \mathbb{R}$ a continuous function with $c > 0$. Then the equation

$$\begin{aligned} c(x) |Du(x)| - 1 &= 0 \quad \text{in } \Omega \\ u(x) &= 0 \quad \text{on } \partial\Omega \end{aligned} \tag{1.3.7}$$

is called the eikonal equation. The value $u(x)$ can be interpreted as the arrival time at x of a wave starting at $\partial\Omega$ and travelling with speed c . Of course this equation is of eikonal type and satisfies the conditions (E1)-(E4).

EXAMPLE 1.3.7:

The eikonal equation can be generalized by allowing the speed c to depend also on the direction of the gradient of u . We now assume $\Omega \subset \mathbb{R}^2$ to be open and bounded. Furthermore let $M(x) : \mathbb{R}^2 \rightarrow \mathcal{S}^2$ be a continuous mapping into the set of 2×2 symmetric positive definite matrices. We denote the scalar product induced by $M(x)$ with $\langle p, q \rangle_{M(x)} := \langle M(x)p, q \rangle$. The corresponding norm is defined by $|p|_{M(x)} := \langle p, p \rangle_{M(x)}$. The equation

$$\begin{aligned} |Du(x)|_{M(x)} - 1 &= 0 \quad \text{in } \Omega \\ u(x) &= 0 \quad \text{on } \partial\Omega \end{aligned} \tag{1.3.8}$$

is called the anisotropic eikonal equation. Sometimes this equation is also called generalized eikonal equation (cf. [BR06, Section 7]) but the notion of anisotropic is more specific and we use the word generalized in this thesis in another context (cf. chapter 2). As in the above example this equation is of eikonal type and fulfills the conditions (E1)-(E4).

1.3.3 Optimal control

In this subsection we show that the eikonal equation can be seen as a problem of optimal control.

Let us regard the eikonal equation (1.3.7). To interpret this equation in the sense of optimal control we look at a controlled dynamic system

$$\begin{aligned} y'(t) &= f(y(t), a(t)), \quad t > 0 \\ y(0) &= x \end{aligned} \tag{1.3.9}$$

with $a(\cdot) \in \mathcal{A} := \{a : [0, \infty) \rightarrow A \text{ measurable}\}$, A a given compact metric space, $x \in \mathbb{R}^d$ and the function $f : \mathbb{R}^d \times A \rightarrow \mathbb{R}^d$ is assumed to be Lipschitz-continuous in the first argument. y is called the state of the system and a the control. Further we write $y_x(\cdot, a) = y_x(\cdot)$ for the solution of (1.3.9).

Now we introduce the two cost functionals which depend on the initial state $x \in \mathbb{R}^d$ and the control $a(\cdot) \in \mathcal{A}$

- Finite horizon: given $g \in \mathcal{C}(\mathbb{R}^d)$, $t > 0$

$$J(x, t, a) := g(y_x(t, a)), \tag{1.3.10}$$

- Minimum time: given a closed target $\mathcal{T} \subset \mathbb{R}^d$

$$t_x(a) := \begin{cases} \min\{s : y_x(s, a) \in \mathcal{T}\} & \text{if } \{s : y_x(s, a) \in \mathcal{T}\} \neq \emptyset \\ +\infty & \text{otherwise.} \end{cases} \tag{1.3.11}$$

Loosely speaking $J(x, t, a)$ gives the value of g at point y which is reached by a path starting in x and travelling for time t with control a . On the other hand $t_x(a)$ gives the time of first contact with \mathcal{T} , starting at x and travelling with control a .

Now we define the value functions

- Finite horizon: $v(t, x) := \inf_{a(\cdot) \in \mathcal{A}} J(t, x, a)$,
- Minimum time: $T(x) := \inf_{a(\cdot) \in \mathcal{A}} t_x(a)$.

We can interpret $v(t, x)$ as the cost of the cheapest point we can reach if we start in x and travelling with the best control a for the time t . Likewise we can interpret $T(x)$ as the first time such that starting from x we reach the target \mathcal{T} by using an optimal control a .

In the next theorem we show that the value function is, under some assumptions, the solution of a certain PDE.

THEOREM 1.3.8 ([Bar97, Proposition 1.3]):

Assume the value function is C^1 in a neighbourhood of (t, x) (or x for the minimum time problem). Then

$$\begin{aligned} \text{Finite horizon:} \quad & v_t(t, x) + H(x, D_x v(t, x)) = 0 \\ \text{Minimum time:} \quad & H(x, DT(x)) = 1 \end{aligned}$$

Here the Hamiltonian is for both cases $H(x, p) = \sup_{a \in \mathcal{A}} \{-f(x, a) \cdot p\}$ (see [BCD97, Proposition III.3.5 and Proposition IV.2.3]). Now we must find a dynamic f and a set of controls such that $\sup_{a \in \mathcal{A}} \{-f(x, a) \cdot p\} = c(x) |p|$. It is easy to see that a suitable dynamic f and set \mathcal{A} of controls are

$$f(x, a) = c(x) \cdot a \quad \text{and} \quad \mathcal{A} := \{a : [0, \infty) \rightarrow \overline{B_1(0)}\}.$$

1.4 Discontinuous viscosity solutions

We must extend the notion of viscosity solutions to discontinuous functions since we deal with such problems in chapter 2. Before we can give a formal definition we introduce the upper and lower semi-continuous envelope.

DEFINITION 1.4.1 ([BCD97, Section V.2.1]):

Let $u : X \rightarrow \mathbb{R}$ be a bounded function. The upper semi-continuous envelope of u is defined by

$$u^*(x) = \limsup_{y \rightarrow x} u(y) := \lim_{r \rightarrow 0^+} \sup \{u(y) : y \in X, |y - x| \leq r\}. \quad (1.4.1)$$

The lower semi-continuous envelope of u is

$$u_*(x) = \liminf_{y \rightarrow x} u(y) := \lim_{r \rightarrow 0^+} \inf \{u(y) : y \in X, |y - x| \leq r\}. \quad (1.4.2)$$

We want to remark that u^* is the smallest upper semi-continuous function which is greater than or equal u and on the other hand u_* is the greatest lower semi-continuous function which is smaller than or equal to u .

Now we can define what we understand by a discontinuous viscosity solution.

DEFINITION 1.4.2 ([BP90, Section 4 and Proposition 5]):

Let $H : [0, \infty) \times \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ be a continuous function and $g : \mathbb{R}^d \rightarrow \mathbb{R}$ bounded. A function u is called a discontinuous viscosity solution of the initial-value problem (1.2.1) provided:

(i) $u^* \leq g$ and $u_* \geq g$ on $\{t = 0\} \times \mathbb{R}^d$,

(ii) for each $v \in C^\infty([0, \infty) \times \mathbb{R}^d)$,

$$\begin{cases} \text{if } u^* - v \text{ has a local maximum at a point } (t_0, x_0) \in (0, \infty) \times \mathbb{R}^d, \text{ then} \\ v_t(t_0, x_0) + H(t_0, x_0, Dv(t_0, x_0)) \leq 0, \end{cases} \quad (1.4.3)$$

and

$$\begin{cases} \text{if } u_* - v \text{ has a local minimum at a point } (t_0, x_0) \in (0, \infty) \times \mathbb{R}^d, \text{ then} \\ v_t(t_0, x_0) + H(t_0, x_0, Dv(t_0, x_0)) \geq 0. \end{cases} \quad (1.4.4)$$

We see that this definition is very similar to 1.2.1. The only difference is that we do not test the function u but instead its upper and lower semi-continuous envelope. As before we call u a subsolution if it satisfies (1.4.3) and a supersolution if it satisfies (1.4.4).

1.5 Front propagation

In chapter 2 we are interested in computing a discontinuous viscosity solution of the generalized eikonal equation (2.0.1). This equation give us the motion of a front in time. To get a better understanding of this topic we introduce the problem and give some basic results.

In the first subsection 1.5.1 we introduce the problem of front propagation and give all the definitions. In subsection 1.5.2 we list some basic results of existence and uniqueness and in the last subsection 1.5.3 we take a closer look at the non-empty interior condition.

1.5.1 Level-Set approach

In this subsection we give a definition for the problem of front propagation. A short introduction to this matter can be found in [BSS93, Section 1] or [Sou97, Section 1].

The front propagation is formally explained as follows. Γ_t denotes a smooth front at time $t > 0$. Let $\Gamma_t = \partial\Omega_t$ with $\Omega_t \subset \mathbb{R}^d$ open. The front is moving outward from Γ_t with normal velocity V which is formally given by

$$V = v(t, x, n, Dn). \quad (1.5.1)$$

We assume that v is a continuous function, n denotes here the unit normal vector of Γ_t which points outward and Dn is the derivative of vector n . Furthermore we assume that there is a smooth function u such that

$$\begin{aligned} \Omega_t &= \{x \in \mathbb{R}^d : u(t, x) > 0\} \\ \Gamma_t &= \{x \in \mathbb{R}^d : u(t, x) = 0\} \end{aligned}$$

and $Du(x) \neq 0$ for all $x \in \Gamma_t$. V , n and Dn depends on u and its derivatives in the following way

$$V = \frac{u_t}{|Du|}, \quad n = -\frac{Du}{|Du|} \quad \text{and} \quad Dn = -\frac{1}{|Du|} \left(I - \frac{Du \otimes Du}{|Du|^2} \right) D^2u. \quad (1.5.2)$$

With these relations we can rewrite equation (1.5.1) in the form

$$u_t + F(t, x, Du, D^2u) = 0$$

and owing to (1.5.2) we get

$$F(t, x, p, X) = -|p| v \left(t, x, -\frac{p}{|p|}, -\frac{1}{|p|} \left(I - \frac{p \otimes p}{|p|^2} \right) X \right)$$

with $p \in \mathbb{R}^d$ and $X \in \mathcal{S}^d$, the space of symmetric $d \times d$ matrices. We further see that F fulfills a certain functional equation, namely

$$F(t, x, \lambda p, \lambda X + \mu(p \otimes p)) = \lambda F(t, x, p, X) \quad \text{for all } \lambda > 0 \text{ and } \mu \in \mathbb{R}. \quad (1.5.3)$$

A function F is called geometric, if it satisfies this functional equation (1.5.3). To ensure that the problem is well-posed, we must further assume that F is degenerate elliptic, that means

$$F(t, x, p, X) \leq F(t, x, p, Y) \quad \text{if } X \geq Y \quad (1.5.4)$$

for all $(t, x) \in (0, \infty) \times \mathbb{R}^d$, $p \in \mathbb{R}^d$ and $X, Y \in \mathcal{S}^d$. Here $X \geq Y$ means that $X - Y$ has only non-negative eigenvalues.

Until now everything was quite formal. Let us choose a function F that satisfies (1.5.3) and (1.5.4). Further we choose a suitable initial function u_0 . We assume that we have a closed set $\Gamma_0 \subset \mathbb{R}^d$ such that

$$\Gamma_0 = \{x \in \mathbb{R}^d : u_0(x) = 0\}.$$

Thus Γ_0 represents the front at time $t = 0$. Now solve the equation

$$\begin{aligned} u_t + F(t, x, Du, D^2u) &= 0 && \text{in } (0, \infty) \times \mathbb{R}^d, \\ u(0, x) &= u_0(x) && \text{on } \mathbb{R}^d. \end{aligned} \quad (1.5.5)$$

in the viscosity sense⁵. Then we define the front Γ_t at time t by

$$\Gamma_t = \{x \in \mathbb{R}^d : u(t, x) = 0\}. \quad (1.5.6)$$

This formulation has the advantage that we do not need to assume that Γ_t is smooth and we can track the propagation of the front over singularities.

⁵Here we use a notion of viscosity solutions for second order equations. We will not use second order equations except in this section, so we omit a formal introduction. All results that are stated in this section hold also for the first order equations which we consider in the following chapters. An introduction to viscosity solutions for second order equations can be found in [CIL92].

1.5.2 Basic properties

In this subsection we recall a theorem which states that (1.5.3) really deals with the geometric properties of u . For the following theorems we need some technical assumptions on F . These are listed below.

(F1) The mapping $(t, x, p, X) \rightarrow F(t, x, p, X)$ is bounded for bounded (p, X) and continuous for $x \in \mathbb{R}^d$, $t \in [0, R] \in B_R(0) \setminus \{0\}$ and $|X| \leq R$ for all $R > 0$.

(F2) $F_*(t, x, \alpha(x - y), X) - F^*(t, x, \alpha(x - y), Y) \geq -\omega(|x - y|(1 + \alpha|x - y|))$, where $\omega(0^+) = 0$ and for all $x, y \in \mathbb{R}^d$, $t \in (0, \infty)$, $\alpha \geq 0$ and matrices $X, Y \in \mathcal{S}^n$ such that $\begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix} \leq K\alpha \begin{pmatrix} I & -I \\ -I & I \end{pmatrix}$ for some constant $K > 0$.

(F3) $F_*(t, x, 0, 0) = F^*(t, x, 0, 0)$

The following theorem shows the uniqueness of the solution and also gives a comparison principle.

THEOREM 1.5.1 ([BSS93, Theorem 1.1]):

Assume that F suffices the conditions (F1)-(F3) and (1.5.3) plus (1.5.4). Then, for any $u_0 \in UC(\mathbb{R}^d)$, there exists a unique solution $u \in UC([0, \infty) \times \mathbb{R}^d)$ of (1.5.5). Moreover, if u and v are, respectively, sub- and supersolutions of (1.5.5) in $UC([0, \infty) \times \mathbb{R}^d)$, then

$$u(0, \cdot) \leq v(0, \cdot) \text{ in } \mathbb{R}^d \Rightarrow u \leq v \text{ in } [0, \infty) \times \mathbb{R}^d. \quad (1.5.7)$$

THEOREM 1.5.2 ([BSS93, Theorem 1.2]):

Assume the hypotheses of theorem 1.5.1 hold and let $u, v \in UC([0, \infty) \times \mathbb{R}^d)$ be solutions of (1.5.5) such that

$$\begin{aligned} \{x \in \mathbb{R}^d : u(0, x) > 0\} &= \{x \in \mathbb{R}^d : v(0, x) > 0\}, \\ \{x \in \mathbb{R}^d : u(0, x) < 0\} &= \{x \in \mathbb{R}^d : v(0, x) < 0\}, \\ \{x \in \mathbb{R}^d : u(0, x) = 0\} &= \{x \in \mathbb{R}^d : v(0, x) = 0\} \end{aligned}$$

and

$$\lim_{|x| \rightarrow \infty} |u(0, x)|, \lim_{|x| \rightarrow \infty} |v(0, x)| > 0.$$

Then for all $t > 0$ we have

$$\begin{aligned} \{x \in \mathbb{R}^d : u(t, x) > 0\} &= \{x \in \mathbb{R}^d : v(t, x) > 0\}, \\ \{x \in \mathbb{R}^d : u(t, x) < 0\} &= \{x \in \mathbb{R}^d : v(t, x) < 0\}, \\ \{x \in \mathbb{R}^d : u(t, x) = 0\} &= \{x \in \mathbb{R}^d : v(t, x) = 0\}. \end{aligned}$$

This theorem justifies calling a function F geometric if it satisfies (1.5.3). So the evolution $\Gamma_0 \rightarrow \Gamma_t$ depends only on F and the right choice of signs of the initial datum u_0 inside and outside of Γ_0 , but it does not depend on the precise choice of u_0 .

1.5.3 Non-empty interior

In [BSS93] is a very interesting theorem which states that in general we do not get an unique solution of (1.5.5) if the initial data are discontinuous.

We start with a definition of a regular evolution Γ_t and then we give a theorem which connects the uniqueness and the regularity. After this we give an example of an evolution that has multiple solutions.

DEFINITION 1.5.3 ([BSS93, Definition 2.1]):

Let Γ_t be the evolution of Γ_0 by the level set approach. We say that $\{\Gamma_t\}_{t \geq 0}$ is regular if

$$\begin{aligned} \text{cl}\{(t, x) : u(t, x) > 0\} &= \{(t, x) : u(t, x) \geq 0\} \quad \text{and} \\ \text{int}\{(t, x) : u(t, x) \geq 0\} &= \{(t, x) : u(t, x) > 0\} \end{aligned}$$

For Γ_0 we define the set $\Omega_0 = \{x : u(0, x) > 0\}$. With this notation, we can state the following interesting result.

THEOREM 1.5.4 ([BSS93, Theorem 2.1]):

$\{\Gamma_t\}_{t \geq 0}$ is regular if and only if there exists a unique solution of (1.5.5) with initial datum $1_{\Omega_0} - 1_{\Omega_0^c}$.

For some special functions F we can be assured that the evolution has an empty interior and is therefore unique.

THEOREM 1.5.5 ([BSS93, Theorem 4.1]):

We assume that $u(0, x) = \text{dist}(x, \Gamma_0)$ in \mathbb{R}^d and $\Gamma_0 = \partial\{x \in \mathbb{R}^d : d(x, \Gamma_0) < 0\} = \partial\{x \in \mathbb{R}^d : d(x, \Gamma_0) > 0\}$. Let $\alpha \in W^{1, \infty}((0, T) \times \mathbb{R}^d)$ for all $T > 0$, with the condition that either (i) α does not change sign in $(0, \infty) \times \mathbb{R}^d$ or (ii) α is independent of t . Then $\Gamma_t = \{x : u(t, x) = 0\}$ is regular, where $u \in UC((0, \infty) \times \mathbb{R}^d)$ is the solution of $u_t + \alpha(t, x) |Du| = 0$ in $(0, \infty) \times \mathbb{R}^d$. In particular Γ_t has an empty interior.

Some numerical tests of the generalized fast marching method in in section 2.5 use functions that suffices the conditions of theorem 1.5.5. But we cannot always assume these conditions and in the following example we give an evolution that has a non-unique solution and therefore develops a non-empty interior.

The following example was first published in [Bar93, Section 4] and afterwards in [BSS93, Proposition 4.4]. Afterwards we extend this example which gives us an explicit evolution, which develops a non-empty interior.

EXAMPLE 1.5.6 (Non-empty interior):

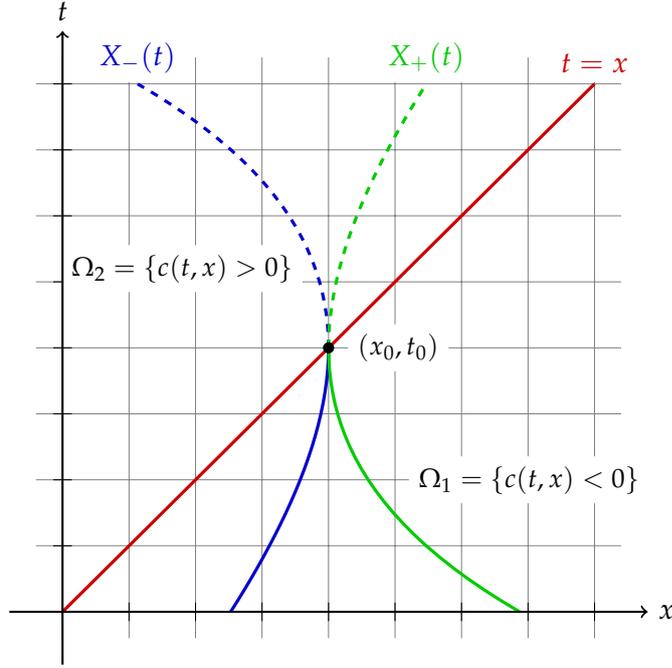
Consider the equation

$$\begin{aligned} u_t - (t - x) |u_x| &= 0 \text{ in } \mathbb{R} \times (0, \infty) \\ u(0, x) &= (1_I - 1_{I^c})(x) \text{ on } \mathbb{R} \end{aligned} \tag{1.5.8}$$

with some appropriate interval I . We construct a suitable interval such that this evolution develops a non-empty interior.

To get a solution we interpret (1.5.8) in the domain $\Omega_1 = \{(x, t) : c(t, x) < 0\}$ as a finite horizon problem, see also subsection 1.3.3. The dynamics are given by

$$\dot{y}_x(s) = c(s, y_x(s))a(s) \quad \text{with } y_x(0) = x \tag{1.5.9}$$

Figure 1.5.1: Plot of example 1.5.6 with $(x_0, t_0) = (1, 1)$.

and $a(\cdot) \in L^\infty((0, \infty), [-1, 1])$.

Now we determine the trajectories passing through (x_0, t_0) and driven by the forward and backward ODE

$$\dot{X}_\pm(t) = \pm c(t, X_\pm(t))$$

with $c(t, x) = t - x$. For the solution we get

$$X_\pm(t) = C_\pm e^{\mp t} + t \mp 1 \quad \text{with} \quad C_\pm = (x_0 - t_0 \pm 1)e^{\pm t_0}.$$

Let us fix $t_0 := x_0 := 1$ and define the open interval $I := I(0)$ with $I(t) := (X_-(t); X_+(t))$. Then we set for the initial state $u(0, x) = (1_I - 1_{I^c})(x)$. The evolution of this discontinuous function is displayed in figure 1.5.1.

By interpreting this as an optimal control problem the minimal supersolution and the maximal subsolution in Ω_1 are given by

$$u_*(t, x) = \inf_{a(\cdot) \in \mathcal{A}} u_*(0, y_x(t)) \quad \text{and} \quad u^*(t, x) = \inf_{a(\cdot) \in \mathcal{A}} u^*(0, y_x(t)) \quad (1.5.10)$$

which can also be written in the form $u_*(t, x) = (1_{I(t)} - 1_{I(t)^c})(x)$ and $u^*(t, x) = (1_{\overline{I(t)}} - 1_{\overline{I(t)}^c})(x)$.

In the set $\{(x, t) : x = t\}$ one gets $u_*(t, x) = -1$. On the other hand we have in $\{(x, t) : x = t\} \setminus \{(x_0, t_0)\}$ now $u^*(t, x) = -1$, but $u^*(x_0, t_0) = 1$.

We take now a closer look at the solution in the set $\Omega_2 = \{(x, t) : x < t\}$. In this domain we can also use a similar interpretation as an optimal control problem. We get

for the maximal and minimal solutions

$$\bar{u}(t, x) = \sup_{a(\cdot) \in \mathcal{A}} \{u^*(\tau, y_x(\tau))1_{\{\tau < t\}}\}$$

and

$$\underline{u}(t, x) = \sup_{a(\cdot) \in \mathcal{A}} \{u_*(\tau, y_x(\tau))1_{\{\tau < t\}}\}$$

where τ denotes the time the trajectory y_x is exiting from Ω_2 .

It follows that we have on the one hand $\underline{u} = -1$ in Ω_2 . On the other hand we have $\bar{u}(t, x) = 1$ for those points in Ω_2 for which the trajectory y_x can reach the point (x_0, t_0) and $\bar{u}(t, x) = -1$ otherwise. In other words, we gain $\bar{u}(t, x) = 1$ for all points in Ω_2 with $X_+(t) \leq x \leq X_-(t)$ and -1 otherwise. Thus the evolution has developed a non-empty interior.

Loosely speaking the problem of non-empty interiors emerges if a singularity of the evolution appears at a point at which the speed function changes its sign. Such a situation cannot be foreseen because in general we do not have enough information to decide on it.

We continue the preceding example and give a continuous evolution that develops a non-empty interior.

EXAMPLE 1.5.7:

Now regard the evolution governed by the equation

$$\begin{aligned} u_t - (t - x) |u_x| &= 0 \text{ in } \mathbb{R} \times (0, \infty) \\ u(0, x) &= u_0(x) \text{ on } \mathbb{R}. \end{aligned}$$

with a continuous function u_0 with $\{u_0 > 0\} = I$. We set $x_0^- = X_-(0) = 1 - 1/e$ and $x_0^+ = X_+(0) = e - 1$, using the trajectories X_{\pm} defined above.

Let us chose for u_0 the signed distance of the points x_0^-, x_0^+ namely $u_0(x) = \min\{x - x_0^-, -(x - x_0^+)\}$. This function is uniformly continuous and we expect from theorem 1.5.4 that the evolution should develop a non-empty interior.

In Ω_1 the solution u can be achieved by the trajectories X_- and X_+ starting in (t, x) and following them to the intersection with the axis $\{t = 0\}$. We get the two intersection points $x^- = X_-(0) = 1 + (x - t - 1)e^{-t}$ and $x^+ = (x - t + 1)e^t - 1$. Thus $u(t, x) = \min\{u_0(x^-), u_0(x^+)\}$ and on the set $\{t = x\}$ we get $u(t, t) = \min\{u_0(1 - 1/e^t), u_0(e^t - 1)\}$, especially $u(1, 1) = 0$.

In the set Ω_2 we gain by the interpretation as an optimal control problem the following solution:

$$u(t, x) = \begin{cases} u(1, 1) = 0 & \text{if } X_-^0(t) \leq x \leq X_+^0(t), \\ u(\tau, \tau) & \text{if } x \leq X_-^0(t) \text{ with } X_-(\tau) = \tau, \\ u(\tau, \tau) & \text{if } X_+^0(t) \leq x \text{ with } X_+(\tau) = \tau, \end{cases}$$

where the trajectories X_- and X_+ pass through (x, t) and the trajectories X_+^0 and X_-^0 pass through (x_0, t_0) . Thus we have for $X_-(\tau) = \tau$ that $\tau = (x - t - 1)e^{-t}e^{\tau} + \tau + 1$ which yields $\tau = t - \ln(1 + t - x)$. On the other hand we get for $X_+(\tau) = \tau$ then $\tau = (x - t + 1)e^te^{-\tau} + \tau - 1$ which gives us $\tau = t + \ln(1 - t + x)$. The function u is displayed for several values of t in figure 1.5.2.

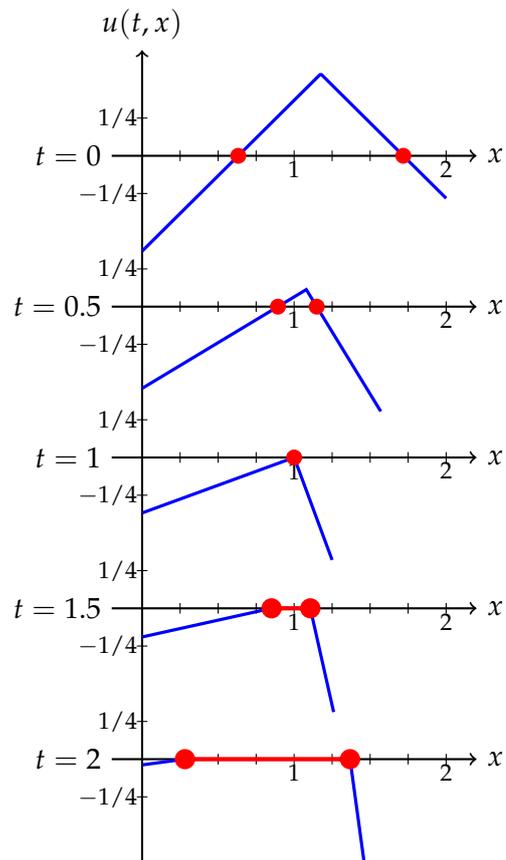


Figure 1.5.2: Plot of example 1.5.6 with u_0 as the signed distance. The evolution (blue) is plotted for the times 0, 0.5, 1, 1.5 and 2. The corresponding 0-level-set is marked red. One can see that the 0-level-set develops a non-empty interior for $t > 1$.

2 Generalized fast marching method

This chapter deals with the numerical solution of a generalized eikonal equation. For an open set $\Omega_0 \subset \mathbb{R}^d$ and a continuous function $c : \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}$ the generalized eikonal equation is defined as

$$\begin{cases} \theta_t(t, x) - c(t, x) |\theta_x(t, x)| = 0 & \text{on } (0, T) \times \mathbb{R}^d \\ \theta(0, \cdot) = 1_{\Omega_0} - 1_{\Omega_0^c}, \end{cases} \quad (2.0.1)$$

with $\theta : \mathbb{R} \times \mathbb{R}^d \rightarrow \mathbb{R}$. For a structured discretization of the space \mathbb{R}^d a first algorithm was proposed in [CCFo6] and after this in [Cri07, Section 3.4]. This first version was improved in [CFFMo8] to be convergent which is also proven in this article. The authors of this algorithm also gave some numerical examples in \mathbb{R}^2 . A year later Forcadel proposed in [For09] a slightly modified algorithm such that the algorithm fulfills also a discrete comparison principle.

The aim of this chapter is to propose a modification of both versions such that one can use an unstructured triangulation for the discretization of \mathbb{R}^2 .

This chapter is organized as follows. First we recall in section 2.1 the well known fast marching method (FMM) for structured and unstructured meshes. In the next section 2.2 we take a short review of the two versions of the generalized fast marching method (GFMM). There we also recall some properties of these methods. In section 2.3 we use the ideas of the two versions of GFMM to propose two new versions that work on unstructured triangulations (GFT). We also state some properties of these methods there. Section 2.4 is dedicated to the proof of a comparison principle. In section 2.5 we give some numerical examples that compare the different versions of GFMM with the new algorithms. In section 2.6 we extend the GFT to deal with anisotropic speed functions. In the last section 2.7 we sum up the results and briefly discuss some open problems.

If the reader is already familiar with the mGFMM he is encouraged to read remark 2.2.12 where we correct an error of the mGFMM.

Here a short comment on the notation: We follow [CFFMo8], thus the symbol θ is always used in a time-dependent context and refers to the solution of (2.0.1) and the symbol u is always regarded in a static context.

2.1 Fast marching method

In this section we explain the fast marching method (FMM) and its extension to triangulated domains. In the first subsection 2.1.1 we recall the fast marching method for structured grids. After this we see in subsection 2.1.2 how one can extend this method to acute triangulations. Then in subsection 2.1.3 we extend the FMM to deal

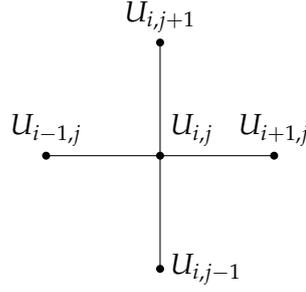


Figure 2.1.1: 4-point neighbourhood for the upwind discretization.

with non-acute triangulations by using virtual updates. In the last subsection 2.1.4 we see how FMM can cope with the anisotropic eikonal equation.

2.1.1 FMM for structured grids

We now introduce the fast marching method, which was first published in [Set96a] and [Set96b]. An overview of FMM, with an elaborate discussion of theory and applications can be found in [Set99b, Chapter 8]. For reasons of simplicity we stick here only to the two-dimensional case, but of course this method could also be used in d dimensions.

The fast marching method is an algorithm to solve the eikonal equation

$$\begin{aligned} |\nabla u(x)| \cdot c(x) - 1 &= 0 \quad \text{in } [0, 1]^2 \setminus \Gamma \\ u(x) &= 0 \quad \text{on } \partial\Gamma \end{aligned} \quad (2.1.1)$$

where Γ is a closed set and the speed function $c > 0$ is assumed to be continuous.

For the discretization of (2.1.1) we use the upwind scheme introduced in [RT92]. First, we endow $[0, 1]^2$ with a Cartesian grid (ih, jh) ($i, j = 0, \dots, n$) of grid-spacing $h = 1/n$. Furthermore we split the index set $\{0, \dots, n\}^2$ into the two disjoint sets Ω_D and Γ_D . Here Γ_D represents the set Γ in the discrete grid. The discrete equation is

$$\begin{aligned} \max(D_{ij}^{-x}U, -D_{ij}^{+x}U, 0)^2 + \max(D_{ij}^{-y}U, -D_{ij}^{+y}U, 0)^2 &= \frac{1}{C_{ij}^2}, \quad (i, j) \in \Omega_D \\ U_{ij} &= 0, \quad (i, j) \in \Gamma_D. \end{aligned} \quad (2.1.2)$$

Here we denote by $D_{ij}^{\pm x}U$ the backward- and forward finite-difference approximation of the partial derivative with respect to x , that is

$$D_{ij}^{-x}U = \frac{U_{ij} - U_{i-1,j}}{h}, \quad -D_{ij}^{+x}U = \frac{U_{ij} - U_{i+1,j}}{h}, \quad (2.1.3)$$

where $U_{ij} = u(ih, jh)$, $C_{ij} = c(ih, jh)$. The finite differences $D_{ij}^{\pm y}U$ in y -direction are defined in the same way. For notational convenience we set $U_{i,n+1} = U_{i,-1} = \infty$ and $U_{n+1,j} = U_{-1,j} = \infty$ ($i, j = 0, \dots, n$). This notation is also very useful for the implementation.

The FMM works as follows: In the first step we tag all points in Γ_D to be *known*. After this we tag all points in Ω_D that have a neighbour in Γ_D as *trial*. All other points are tagged as *far*. Now we compute trial values for all *trial* points, using (2.1.2), where we use the values of the points in Γ_D . Let NB be the set of trial points, the so called *narrow-band*. Now we perform the iteration:

- (FMM1) Let $(i, j) \in NB$ denote the point with the smallest trial value.
- (FMM2) Tag (i, j) as *known*, and remove it from NB .
- (FMM3) Tag all neighbours of (i, j) , which are not *known*, as *trial* and add them to NB .
- (FMM4) Recompute the values of U for all *trial* neighbours of (i, j) by using (2.1.2).
- (FMM5) If $NB \neq \emptyset$, goto (FMM1).

As the value of U_{ij} in (2.1.2) depends only on smaller values of U in the neighbouring points, the fast marching method computes a solution to (2.1.2). This dependency of U_{ij} only on neighbour points with smaller values of U is also denoted as the causality. This property is very essential for the discretization, because with this we have a notion of upwind and therefore we can solve the discrete scheme (2.1.2) in only one iteration.

The complexity of the FMM is $\mathcal{O}(N \log N)$, with N denoting the total number of grid points. The factor $\log N$ comes from step (FMM1). In this step one searches the point with the smallest trial value in the narrow-band. This is done by a heapsort which needs $\log N$ steps in every cycle of the algorithm.

Another method which is even faster, that means it has complexity $\mathcal{O}(N)$, was introduced by Tsitsiklis in [Tsi95]. But this method has not been as popular as the FMM. Further there exist variants of the FMM with complexity $\mathcal{O}(N)$. There one uses not an exact priority queue but a so called untidy priority queues, see also [YBS05].

2.1.2 FMM for acute triangulations

We want to apply the FMM to grids with acute triangulations. To do this, we need a suitable update scheme that works on triangulations. We regard a triangulation \mathcal{T} of Ω and we denote by \mathcal{N} the set of vertices and by τ we denote a single triangle in \mathcal{T} . We follow here the concept of Hopf-Lax updates introduced in [BR06]. Using other ideas identical update-schemes have been developed, i.e. [Set99b, Chapter 10] or [Fom97].

On structured grids we use the 4-point neighbourhood for the update scheme. In unstructured triangulations, we use the following sense of neighbourhood. The neighbourhood $\omega(C)$ of a node $C \in \mathcal{N}$ is defined as the collection of all triangles $\tau \in \mathcal{T}$ that have C as a common vertex (see figure 2.1.2).

The idea for the update scheme is the following: We fix the speed c on the neighbourhood $\omega(C)$ and we carry out a linear interpolation of u on the edges between the vertices where we already know the function values. Then we calculate the exact viscosity solution in $\omega(C)$ to get a update $(\Delta U)(C)$. Formally we can write the update with the optical distance (see (1.3.4)) as

$$(\Delta U)(C) := \inf_{y \in \partial\omega(C)} \{U(y) + \delta_C(y, C)\}$$

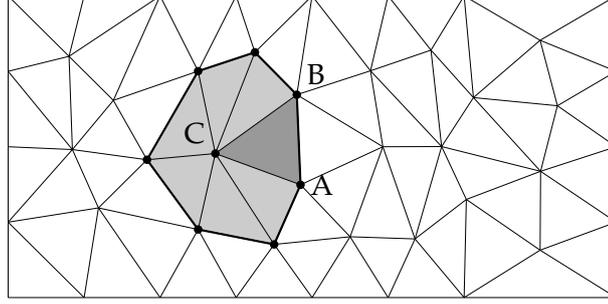


Figure 2.1.2: The neighbourhood $\omega(C)$ of the point C is the collection of the grey triangles joining the point C .

where δ_C denotes the optical distance if we fix the speed c in the neighbourhood $\omega(C)$ to the value at the point C .

Thus we get an update scheme given by

$$(\Lambda U)(C) = \min_{\tau \in \mathcal{T}: C \in \tau} \min \{U(y) + \delta_C(C, y) : y \in \{\text{edge opposite to } C\}\}.$$

Now assume that $\tau_1, \dots, \tau_m \in \omega(C)$ are the triangles which have C as a common vertex. Then we can rewrite the above formula by

$$(\Lambda U)(C) = \min_{1 \leq i \leq m} U_i \tag{2.1.4}$$

where U_i is given in the triangle $ABC = \tau_i \in \omega(C)$ by

$$U_i = \begin{cases} U(A) + |A - C| & \cos \alpha \leq \Delta, \\ U(A) + \cos(\delta - \alpha) |A - C| & \alpha \leq \delta\pi - \beta, \\ U(B) + |B - C| & \Delta \leq \cos \pi - \beta, \end{cases} \tag{2.1.5}$$

with $\Delta = \frac{U(B) - U(A)}{B - A}$, $\cos(\delta) = \Delta$ if $|\Delta| \leq 1$, and α and β are the angles at A and B .

The update scheme fulfills a monotonicity property [BR06, Lemma 5] which will be important in the proof of a comparison principle in proposition 2.3.5 and lemma 2.4.13. Let U and V be two discrete functions on $\partial\omega(C)$ with $U \leq V$ then we know that

$$(\Lambda U)(C) \leq (\Lambda V)(C). \tag{2.1.6}$$

We must assume that all triangles $\tau \in \mathcal{T}$ are acute if we want use the fast marching method. This condition is due to the causality, see also [Set99b, Section 10.3.2]. If we have no acute triangulation we still can use an iterative solver like Gauß-Seidel, but FMM does not converge to the right solution.

At the end of this section we want to mention that FMM works also in \mathbb{R}^d using simplexes instead of triangles but we regard here only examples in \mathbb{R}^2 .

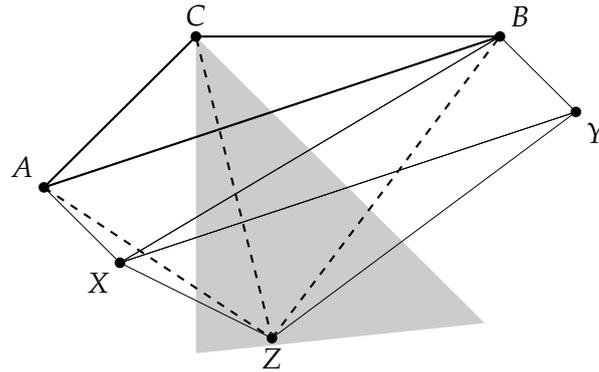


Figure 2.1.3: Construction of the splitting of a non-acute triangle ABC into two acute triangles AZC and ZBC . The splitting section is marked gray.

2.1.3 FMM for general triangulations

A crucial assumption for the FMM is the causality and therefore we are restricted to acute triangulations. In [KS98, Section 4.2] there is an extension of FMM to non-acute triangulations. The main idea is that we split an obtuse triangle into two acute triangles and afterwards we use the acute triangles for the update scheme. These two acute triangles are sometimes called virtual triangles and the corresponding update is called virtual update.

The idea for the construction of the virtual triangles is to search a node that lies in the so called splitting section. The splitting section of an obtuse angle $\angle(ACB)$ is defined as the set of points x such that $0 \leq \angle(ACx) \leq \pi/2$ and $0 \leq \angle(xCB) \leq \pi/2$, see also figure 2.1.3. In essence the virtual triangles have the effect that one locally coarsens the mesh such that the coarsened mesh is acute. Thus we have a lack of accuracy but we still can use the fast single pass method.

We want to emphasize that using virtual updates one does not solve the discrete system for the eikonal equation [Ras07, Section 3.3.3] anymore. Nevertheless this method has been widely used because it gives good results with low computational effort.

There is another issue on which I did not find any discussions in the literature: Using virtual updates, it can happen that the updates are not monotone because we use points which are not neighbours for the update. In Section 3.4.3 we give an example for this and show some effects.

2.1.4 FMM for triangulations and the anisotropic eikonal equation

Now we briefly recall how one can extend the fast marching method to handle the anisotropic eikonal equation (1.3.8), see also [BR06, Section 7]. This extension will be used in section 2.6 and chapter 3. As in the case of the isotropic eikonal equation we

get the update scheme

$$(\Delta U)(C) = \min_{1 \leq i \leq m} U_i \text{ with}$$

$$U_i = \min_{y \in J_i} \left(U(y) + |C - y|_{M(C)} \right).$$

where J_i denotes the edge of triangle τ_i opposite to C and the norm and scalar product induced by a symmetric positive definit matrix M are $|p|_M = \sqrt{\langle p, p \rangle_M}$ with $\langle p, p \rangle_M = \langle Mp, p \rangle$.

The idea to calculate this minimum is that we transform the triangle τ_i with the matrix $M(C)^{-1/2}$. On this transformed triangle we solve the isotropic eikonal equation and therefore we can apply the explicit update formula (2.1.5). Setting $\langle p, q \rangle_C = \langle p, q \rangle_{M(C)^{-1}}$, $|p|_C = |p|_{M(C)^{-1}}$, $c_\alpha = \cos(\alpha)$ and $c_\beta = \cos(\beta)$ we gain the concrete update formula

$$\begin{aligned} \Delta &= \frac{U(B) - U(A)}{|B - A|_C} \\ c_\alpha &= \frac{\langle C - A, B - A \rangle_C}{|C - A|_C \cdot |B - A|_C} \\ c_\beta &= \frac{\langle C - B, A - B \rangle_C}{|C - B|_C \cdot |A - B|_C} \\ U_i &= \begin{cases} U(A) + |C - A|_C & \text{if } c_\alpha \leq \Delta \\ U(B) + |C - B|_C & \text{if } \Delta \leq -c_\beta \\ U(A) + \left(c_\alpha \Delta + \sqrt{(1 - c_\alpha^2)(1 - \Delta^2)} \right) |C - A|_C & \text{else.} \end{cases} \end{aligned} \tag{2.1.7}$$

We must ensure that the transformed triangle $M^{-1/2}\tau_i$ is still acute to use fast marching. If this is not the case we have to search in the transformed triangulation for suitable virtual triangles.

As we can see in (2.1.7), we need only $M(C)^{-1}$ but not $M(C)^{-1/2}$ for our calculations, and $M(C)^{-1}$ is very easy to compute.

2.2 GFMM on structured grids

In this section we recall the two versions of the generalized fast marching method (GFMM) as they were proposed by Carilini et al. in [CFFMo8] and Forcadel in [For09]. In the first subsection 2.2.1, we write down the first version of GFMM and give also some results which have been proved in [CFFMo8]. In the second subsection 2.2.2 we recall the modified version of the GFMM (mGFMM) which was presented in [For09]. This method fulfills a comparison principle and we discuss some of the differences. We will refer to this section in section 2.3 where we use the ideas of the two versions of GFMM to extend these algorithms to unstructured grids.

There are two main ideas in the construction of the GFMM. The first idea is that one separates the information whether a point lies inside or outside from the information about the time of the last visit at a point of the front. The other idea is to use a double narrow-band to keep track of the front. One narrow-band deals with the part of the

front pointing outwards. The other narrow-band keeps track of the nodes pointing inwards. There are also some more minor subtle issues one has to keep in mind. We will take up these points later on.

2.2.1 GFMM

2.2.1.1 Preliminary definitions

We consider a lattice $Q := \{x_I = (x_{i_1}, \dots, x_{i_d}) = (i_1 \Delta x, \dots, i_d \Delta x) : I = (i_1, \dots, i_d) \in \mathbb{Z}^d\}$ with stepsize in space $\Delta x > 0$; further we need a time step $\Delta t > 0$.

DEFINITION 2.2.1 ([CFFMo8, Definition 2.1]):

We define the neighbourhood of the node $I \in \mathbb{Z}^d$ as the set

$$V(I) := \{K \in \mathbb{Z}^d : |K - I| \leq 1\}.$$

DEFINITION 2.2.2 ([CFFMo8, Definition 2.2]):

Given the speed $c_I^n := c(t_n, x_I)$, we define the regularized speed by the function

$$\tilde{c}_I^n := \begin{cases} 0 & \text{if there exists } K \in V(I) \text{ such that } (c_I^n c_K^n < 0) \text{ and } |c_I^n| \leq |c_K^n| \\ c_I^n & \text{otherwise} \end{cases}$$

DEFINITION 2.2.3 ([CFFMo8, Definition 2.3]):

Let $E \subset \mathbb{Z}^d$. We define the numerical boundary ∂E of E as the set

$$\partial E := V(E) \setminus E,$$

with

$$V(E) := \{K \in \mathbb{Z}^d : \exists I \in E \text{ such that } K \in V(I)\}.$$

DEFINITION 2.2.4 ([CFFMo8, Definition 2.4]):

Given a field θ_I^n with values $+1$ and -1 . We define the two phases

$$\Theta_{\pm}^n = \{I \in \mathbb{Z}^d : \theta_I^n = \pm 1\}$$

and the fronts

$$F_{\pm}^n := \partial \Theta_{\mp}^n \text{ and } F^n := F_+^n \cup F_-^n.$$

The definition of F_{\pm}^n can also be written as

$$F_{\pm}^n = \{I \in \mathbb{Z}^d : \theta_I^n = \pm 1 \text{ and } \exists K \in V(I) \text{ with } \theta_K^n = -\theta_I^n\}.$$

We use some further notations for the algorithm, namely

$$\pm g \geq 0 \text{ for } I \in F_{\pm}$$

means

$$+g \geq 0 \text{ for } I \in F_+ \text{ and } -g \geq 0 \text{ for } I \in F_-$$

and for the min and max operators we define

$$\min_{\pm} \{0, g_{\pm}\} := \min\{0, g_+, g_-\} \text{ and } \max_{\pm} \{0, g_{\pm}\} := \max\{0, g_+, g_-\}.$$

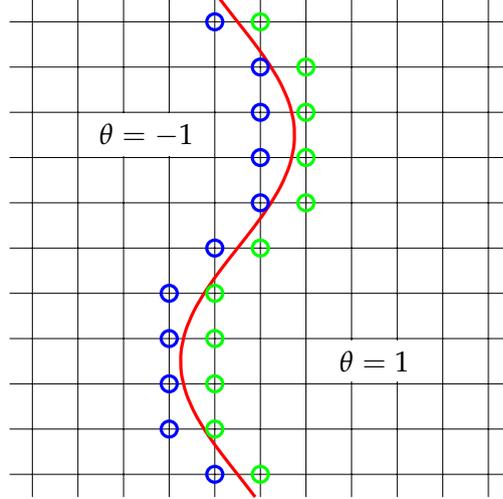


Figure 2.2.1: The front F_- (blue) and F_+ (green) in the GFMM.

2.2.1.2 Description of the algorithm

The algorithm generates an increasing sequence of times $(t_n)_{n \in \mathbb{N}}$ with $t_0 = 0$. The phase parameter θ_I^n takes the values $+1$ and -1 and indicates thereby whether the point x_I at time t_n is inside or outside the front. Thus θ_I^n should be viewed as an approximation to the solution θ of (2.0.1), see also theorem 2.2.9.

We have to introduce some more variables to state the algorithm. First, u_I^n is defined for all nodes I which the front has already passed. This value can be interpreted as the time when the front F^n reaches the node I . The variable \hat{u}_I^n is equal to u_I^n if $I \in F^n$. Thus it is just a restriction of u to F^n . \tilde{u}_I^n is the tentative time of the node I ; this is the time the front may reach the node I . Then \tilde{t}_n is the minimum of all \tilde{u}_I^n , so it give us the time when the front F^n reaches its next point. We have to restrict the value of \tilde{t}_n , to ensure convergence and this restricted value will be denoted by \hat{t}_n .

Now we list the algorithm GFMM:

Initialization

1. Set $n = 1$.
2. Initialize the field θ^0

$$\theta^0 = \begin{cases} 1 & \text{for } x_I \in \Omega_0, \\ -1 & \text{elsewhere.} \end{cases}$$

3. Initialize the time on F^0 by $u_I^0 = 0$ for all $I \in F^0$ and set $t_0 = 0$.

Main cycle

4. Initialize \hat{u}^{n-1} everywhere on the grid by

$$\hat{u}_{\pm, I}^{n-1} = \begin{cases} u_I^{n-1} & \text{for } I \in F_{\pm}^{n-1}, \\ \infty & \text{elsewhere.} \end{cases}$$

5. Compute \tilde{u}^{n-1} on F^{n-1} :

For $I \in F_{\pm}^{n-1}$ do

- a) if $\pm \widehat{c}_I^{n-1} \geq 0$ then $\tilde{u}_I^{n-1} = \infty$
- b) if $\pm \widehat{c}_I^{n-1} < 0$ then we compute \tilde{u}_I^{n-1} as the solution of the following second order equation:

$$\sum_{k=1}^d \left(\max_{\pm} \left(0, \tilde{u}_I^{n-1} - \widehat{u}_{+,I^{k,\pm}}^{n-1} \right) \right)^2 = \frac{(\Delta x)^2}{|\widehat{c}_I^{n-1}|^2} \quad \text{if } I \in F_-^{n-1},$$

$$\sum_{k=1}^d \left(\max_{\pm} \left(0, \tilde{u}_I^{n-1} - \widehat{u}_{-,I^{k,\pm}}^{n-1} \right) \right)^2 = \frac{(\Delta x)^2}{|\widehat{c}_I^{n-1}|^2} \quad \text{if } I \in F_+^{n-1},$$

where $I^{k,\pm} = (i_1, \dots, i_{k-1}, i_k \pm 1, i_{k+1}, \dots, i_d)$.

6. $\tilde{t}_n = \min \left\{ \tilde{u}_I^{n-1} : I \in F^{n-1} \right\}$.

7. $\widehat{t}_n = \min \{ \tilde{t}_n, t_{n-1} + \Delta t \}$.

8. $t_n = \max \{ t_{n-1}, \widehat{t}_n \}$.

9. If $t_n = t_{n-1} + \Delta t$ and $t_n < \tilde{t}_n$ go to step 4 with $n := n + 1$ and

$$\begin{cases} u_I^n = u_I^{n-1} & \text{for all } I \in F^n := F^{n-1}, \\ \theta_I^n = \theta_I^{n-1} & \text{for all } I \in \mathbb{Z}^d. \end{cases}$$

10. Initialize the new accepted points

$$NA_{\pm}^n = \left\{ I \in F_{\pm}^{n-1} : \tilde{u}_I^{n-1} = \tilde{t}_n \right\}, \quad NA^n = NA_+ \cup NA_-.$$

11. Reinitialize θ_I^n

$$\theta_I^n = \begin{cases} -1 & \text{for } I \in NA_+^n, \\ 1 & \text{for } I \in NA_-^n, \\ \theta_I^{n-1} & \text{elsewhere.} \end{cases}$$

12. Reinitialize u^n on F^n :

- a) If $I \in F^n \setminus V(NA^n)$, then $u_I^n = u_I^{n-1}$.
- b) If $I \in NA^n$, then $u_I^n = t_n$.
- c) If $I \in (F^{n-1} \cap V(NA^n)) \setminus NA^n$, then $u_I^n = u_I^{n-1}$.
- d) If $I \in V(NA^n) \setminus F^{n-1}$, then $u_I^n = t_n$.

13. Set $n := n + 1$ and go to 4.

2 Generalized fast marching method

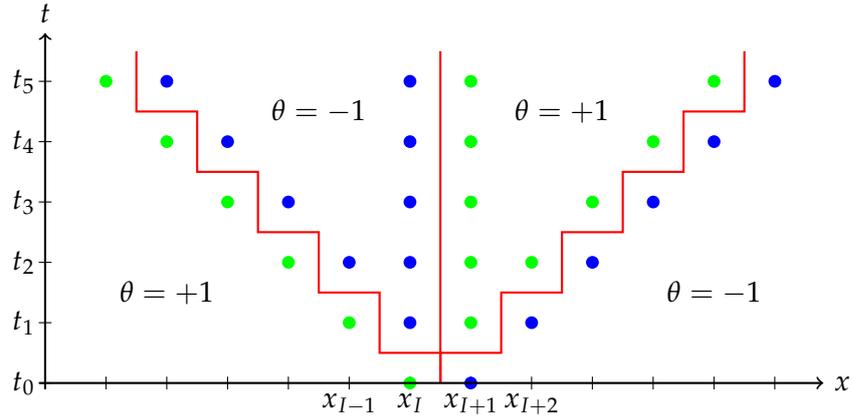


Figure 2.2.2: The discrete evolution of remark 2.2.5. The front F_+ (green) and F_- (blue) using the velocity c instead of \hat{c} .

We present now some remarks on the GFMM.

REMARK 2.2.5 (Using \hat{c} instead of c):

In this remark we want to explain why to use \hat{c} instead of c . To do so we adapt (and correct) an example published in a non-reviewed lecture note by Forcadel¹. Let us consider an 1-D example of the GFMM and we use the speed c instead of \hat{c} . Now assume that we use the space step Δx and that we pick up a certain point x_I . The time-independent speed² c and the evolution θ at the beginning should be

$$\theta(0, x) = \begin{cases} 1 & \text{for } x \leq x_I, \\ -1 & \text{for } x > x_I, \end{cases} \quad c(x) = \begin{cases} -\delta & \text{for } x \leq x_I, \\ \delta & \text{for } x > x_I. \end{cases}$$

Further we set $\Delta t = \Delta x / \delta$. The evolution can be viewed in figure 2.2.2. At time $t_1 = \Delta t$ both points x_I and x_{I+1} are accepted. In the next step at time $t_2 = t_1 + \Delta t$ the points x_{I-1} and x_{I+2} are accepted, and so on. One sees that the discrete evolution splits and develops a topological change that does not appear in the analytic evolution. If we use \hat{c} then we have $\hat{c}_I = 0$ and $\hat{c}_{I+1} = 0$. Then the front will not move as we can see in figure 2.2.3. Thus the regularization of c prevents a separation of the evolution that would not appear in the analytic evolution.

REMARK 2.2.6 (Using \tilde{t} , \hat{t} and t):

In Step 6 of the GFMM we define \tilde{t} as the minimum of all \tilde{u} . This step corresponds to the selection of points with minimal trial value in the FMM.

Step 7 assures that the progress in time does not use too large time steps. Just assume, that we reach in the algorithm a situation at which $\hat{c} \equiv 0$. Then \tilde{t} is computed to ∞ and

¹http://uma.ensta.fr/work/labo_work/files/zidani/NOTES_de_COURS_CEA-EDF-INRIA08/LectureNotes_Forcadel.pdf section 3 on page 6. Last access: 03. August 2011

²Forcadel defines the speed c at point x_I by $c(x_I) = \delta$. But then this example would not work any more. Probably this is just a typesetting error.

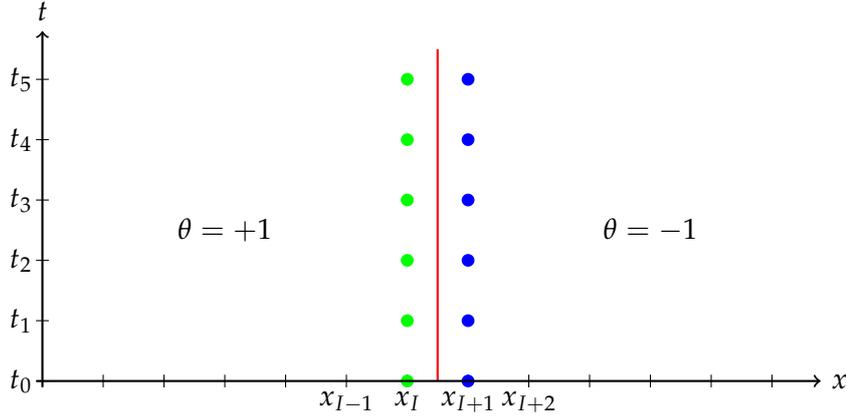


Figure 2.2.3: The discrete evolution of remark 2.2.5. The front F_+ (green) and F_- (blue) using the velocity \hat{c} .

the algorithm would terminate, though the analytic evolution does not necessarily stop for all time as the discrete evolution would do.

On the other hand step 8 enforces that the sequence of times $(t_n)_{n \in \mathbb{N}}$ increases and therefore this could be regarded as a sequence of physical time. Such a situation can happen if at a point x_I the speed grows rapidly, so $\hat{c}_I^{n-1} \ll \hat{c}_I^n$.

We give now some interpretations and comments on the algorithm. We use this interpretation in section 2.3 to construct the new algorithm.

- The aim of steps 1 to 3 is to initiate the data for the computation.
- The parameter Δx is responsible for the resolution in space. Δt together with step 7 ensures, that the speed c is sampled at least after the period of time Δt .
- In step 12 the variable u_I^n is updated with the physical time t_n and not with its tentative time \tilde{t}_n . Thus the sequence $(t_n)_{n \in \mathbb{N}}$ is monotone.
- The two variables $\hat{u}_{\pm, I}^n$ are equal to ∞ if the node I is not in the narrow-band F_{\pm}^n . This means that we restrict the information about u_I^n in a suitable way for computing and everything outside the front F^n is unimportant.
- The update scheme in step 5 is just the d -dimensional version of the upwind discretization of Rouy and Tourin [RT92].
- The steps 9 to 12 are meant to update the information about the narrow-band such that we keep track of the front.
- One has to update the variables F^n , Θ^n and \hat{u}^n in an efficient way to maintain optimal complexity.
- One can find in the preprint [CFFM, Section 3] of [CFFMo8] another motivation for the truncation of t .

2.2.1.3 Properties

We cite here two important properties of the GFMM. The first one is a weak comparison principle and the other one is the convergence theorem.

PROPOSITION 2.2.7 ([CFFMo8, Proposition 3.2]):

We denote by u_I^n (resp., v_I^n) the numerical solution at the point (t_n, x_I) of the GFMM algorithm with velocity c_u (resp., c_v). We assume that there exists $T > 0$ such that for all $(t, x) \in [0, T] \times \mathbb{R}^d$

$$\inf_{s \in [t-\Delta t], s \geq 0} c_v(s, x) \geq \sup_{s \in [t-\Delta t], s \geq 0} (c_u(s, x))^+$$

where f^+ is the positive part of f . We assume that

$$\{\theta_u^0 = 1\} \subset \{\theta_v^0 = 1\} \text{ and } v^0 = u^0 = 0.$$

We define \bar{m} and \bar{k} such that

$$\begin{cases} t_{\bar{m}} \leq T < t_{\bar{m}+1}, \\ s_{\bar{k}} \leq T < s_{\bar{k}+1}, \end{cases}$$

where $(t_m)_m$ and $(s_m)_m$ are the sequences of time constructed by the GFMM algorithm with velocity c_u and c_v , respectively. We then consider

$$v_I = \begin{cases} v_i^0 & \text{if } \theta_{v,I}^0 = 1, \\ v_I^k & \text{if } I \in NA^k \text{ for some } k \leq \bar{k} + 1 \\ s_{\bar{k}+1} & \text{if } \theta_{v,I}^{\bar{k}} = -1. \end{cases}$$

Then, $\forall l \leq \bar{m}$, $\forall I \in NA_u^l$, we have $v_I \leq u_I^l$.

In subsection 2.3.1.3 we will prove an analogous comparison principle for the GFT algorithm.

Now we will cite the convergence theorem for the GFMM. To do this we must introduce some notational conventions. The sequence of physical times $(t_n)_{n \in \mathbb{N}}$ defined in step 8 of the algorithm is non-decreasing. We can extract a strictly increasing sub-sequence $(t_{k_n})_{n \in \mathbb{N}}$ such that

$$t_{k_n} = t_{k_{n+1}} = \dots = t_{k_{n+1}-1} < t_{k_{n+1}}.$$

Till now we approximated the function θ only at the grid points and therefore we have to extend this definition. Denote by S_I^n the square cell $S_I^n := [t_{k_n}, t_{k_{n+1}}[\times [x_I, x_I + \Delta x[$ with the positive quadrant $[x_I, x_I + \Delta x[= \prod_{\alpha=1}^d [x_{i_\alpha}, x_{i_\alpha} + \Delta x[$. Furthermore we denote by ε the couple $\varepsilon := (\Delta t, \Delta x)$.

Now we can define the extension of θ by

$$\theta^\varepsilon(t, x) = \theta_I^{k_{n+1}-1} \text{ if } (t, x) \in S_I^n.$$

and further the half-relaxed limits

$$\bar{\theta}^0(t, x) = \limsup_{\varepsilon \rightarrow 0, (s, y) \rightarrow (t, x)} \theta^\varepsilon(s, y), \quad \underline{\theta}^0(t, x) = \liminf_{\varepsilon \rightarrow 0, (s, y) \rightarrow (t, x)} \theta^\varepsilon(s, y).$$

For this extended function a comparison principle also holds, namely

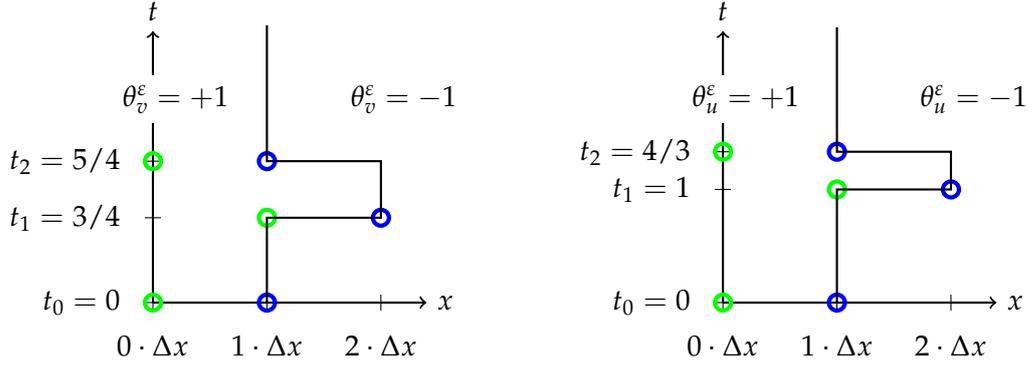


Figure 2.2.4: Evolution for c_v (left) and c_u (right) in remark 2.2.10. The front F_+ (green) and F_- (blue).

COROLLARY 2.2.8 ([CFFMo8, Corollary 3.5]):

Under the assumptions of proposition 2.2.7 we have for all $(t, x) \in [0, T] \times \mathbb{R}^d$

$$\theta_u^\varepsilon(t, x) \leq \theta_v^\varepsilon(t, x). \quad (2.2.1)$$

For the proof of convergence we need also an assumption about the velocity c and the domain Ω_0 :

- (A) The velocity $c \in W^{1,\infty}([0, T] \times \mathbb{R}^d)$ fulfills for some constant $L > 0$ the condition $|c(t, x) - c(s, y)| \leq L(|s - t| + |x - y|)$. Furthermore Ω_0 is a C^2 open set, with bounded boundary $\partial\Omega_0$.

THEOREM 2.2.9 ([CFFMo8, Theorem 2.5]):

Under assumption (A), $\bar{\theta}^0$ (resp., $\underline{\theta}^0$) is a viscosity subsolution (resp., supersolution) of (2.0.1). In particular, if (2.0.1) satisfies a comparison principle, then $\bar{\theta}^0 = (\underline{\theta}^0)^*$ and $(\bar{\theta}^0)_* = \underline{\theta}^0$.

We would like to comment that the assumption that (2.0.1) satisfies a comparison principle is equivalent to the non-empty interior condition. Therefore we have in general no possibility to ensure a priori the uniqueness of the solution. See also section 1.5 for this point.

REMARK 2.2.10:

In this remark we will give an explicit example of two evolutions which does not fulfill the comparison principle. This example is adapted from [For09, Section 5.2], and will be continued in remark 2.2.14.

Let us assume that Δt and Δx small enough. We look at the lattice $Q = \{x_i := i\Delta x, i \in \mathbb{Z}\}$. We choose the the speed c_v and c_u with

$$c_v(t, x) = \begin{cases} \max(0, \min(1, \frac{4}{3}x)) & \text{if } t \leq 1, \\ \min(0, \max(-1, -2x)) & \text{if } t \geq \frac{17}{16}, \end{cases}$$

and

$$c_u(t, x) = \begin{cases} \max(0, \min(1, x)) & \text{if } t \leq 1, \\ \min(0, \max(-1, -3x)) & \text{if } t \geq \frac{17}{16}. \end{cases}$$

We interpolate c_v and c_u linear for constant x between $1 \leq t \leq \frac{17}{16}$. Thus we gain³ $c_v, c_u \in C^{0,1}([0, T] \times \mathbb{R})$.

The initial condition is

$$\theta_{v,i}^0 = \theta_{u,i}^0 = \begin{cases} 1 & \text{if } i \leq 0 \\ -1 & \text{if } i \geq 1 \end{cases}$$

First we compute the evolution for c_v . We gain that point 1 is accepted at time $t_1 = \frac{3}{4}$. Then we set $v_1 = v_2 = t_1$ due to step 12 of GFMM. In the next iteration the tentative value of point 2 becomes $\tilde{v}_2 = t_1 + \frac{3}{8} = \frac{9}{8} > \frac{17}{16}$. Thus point 2 will not be accepted because the speed has previously changed its sign. If Δt is small enough point 1 will be accepted again at time $t_2 = t_1 + \frac{\Delta x}{2\Delta x} = \frac{5}{4}$.

Now let us take a look at the evolution governed by the speed c_u . Point 1 is accepted at time $t_1 = 1$ and we set $u_1 = u_2 = 1$. Point 2 will not be accepted because the speed has previously changed its sign. Now the front turns back and point 1 will be accepted again at time $t_2 = t_1 + \frac{\Delta x}{3\Delta x} = 1 + \frac{1}{3} = \frac{4}{3} > \frac{5}{4}$.

So we have in the interval $[\frac{5}{4}, \frac{4}{3}]$ that

$$-1 = \theta_v^\varepsilon(t, \Delta x) < \theta_u^\varepsilon(t, \Delta x) = 1,$$

which contradicts the comparison principle 2.2.8 for the GFMM.

2.2.2 Modified GFMM

Forcadel, one of the authors of the GFMM, published in a succeeding paper [For09] a modified version of the GFMM, which we call mGFMM, that fulfills a stronger comparison principle.

2.2.2.1 Preliminary definitions

Before we can present the mGFMM we give some preliminary definitions. We still work on the same lattice Q as for the GFMM and the definitions 2.2.1 to 2.2.4 can be reused.

We define the narrow-bands NB_\pm^n at step n via

$$NB_+^n := F_+^n \cap \{I \in \mathbb{Z}^d : \tilde{c}_I^n < 0\}, \quad NB_-^n := F_-^n \cap \{I \in \mathbb{Z}^d : \tilde{c}_I^n > 0\}$$

and

$$NB^n := NB_+^n \cup NB_-^n.$$

In NB^n we store the points which can be reached directly from the front. In all other points in $F^n \setminus NB^n$ the direction of the speed points to the wrong side and therefore the front cannot propagate.

³The restriction $-1 \leq c_v, c_u \leq 1$ is just to ensure that c_v and c_u are Lipschitzian.

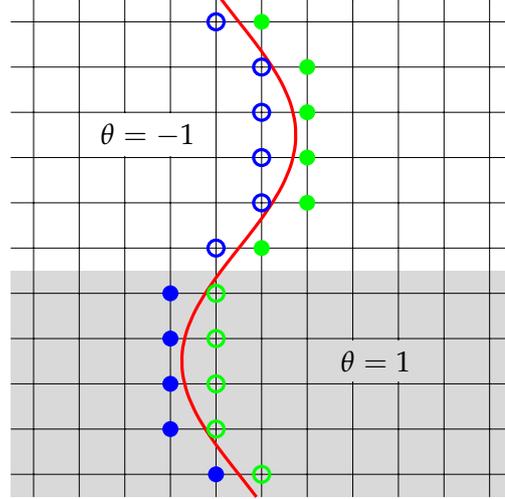


Figure 2.2.5: The front F_- (blue) and F_+ (green) in the mGFMM. The points of NB_- (blue filled circles) and NB_+ (green filled circles) are part of the front F_- and F_+ . In the grey region is the speed c positive, in the white region negative.

We further introduce the notion of useful points for some point K , that is

$$\mathcal{U}^n(K) = \{I \in V(K) : \theta_I^n = -\theta_K^n\}, \quad \mathcal{U}^n = \bigcup_{K \in NB^n} \mathcal{U}^n(K).$$

The meaning of the useful points $\mathcal{U}(K)$ is that one uses only the information of these neighbours of K to compute a tentative value for the arrival time of K . Furthermore we introduce the useful time $u_{I \rightarrow K}^n$. This time $u_{I \rightarrow K}^n$ can be interpreted as the time when the front F^n begin to move from I to K . This meaning is emphasized in remark 2.2.13. In an algorithmic view the useful times $u_{I \rightarrow K}$ replace the times \hat{u}_\pm used in the GFMM.

2.2.2.2 Description of the algorithm

The mGFMM is very similar to the GFMM. First we state the modified algorithm and afterwards we discuss the differences.

Initialization

1. Set $n = 1$.
2. Initialize the field θ^0 as

$$\theta^0 = \begin{cases} 1 & \text{for } x_I \in \Omega_0, \\ -1 & \text{elsewhere.} \end{cases}$$

3. Initialize the time $t_0 = 0$ and set for points I

$$u_{I \rightarrow K}^0 = \begin{cases} t_0 & \text{if } I \in \mathcal{U}^0(K) \text{ and } K \in NB^0 \\ +\infty & \text{elsewhere.} \end{cases}$$

Main cycle

4. Compute \tilde{u}^{n-1} on NB^{n-1}
 Let $I \in NB^{n-1}$, then we compute \tilde{u}_I^{n-1} as the solution of the following second order equation:

$$\sum_{k=1}^d \left(\max_{\pm} \left(0, \tilde{u}_I^{n-1} - u_{I^{k,\pm} \rightarrow I}^{n-1} \right) \right)^2 = \frac{(\Delta x)^2}{|\hat{c}_I^{n-1}|^2}$$

where $I^{k,\pm} = (i_1, \dots, i_{k-1}, i_k \pm 1, i_{k+1}, \dots, i_d)$.

5. $\tilde{t}_n = \min \left\{ \tilde{u}_I^{n-1} : I \in NB^{n-1} \right\}$.
6. Truncate \tilde{t}_n
 $t_n = \max \{ t_{n-1}, \min \{ \tilde{t}_n, t_{n-1} + \Delta t \} \}$.
7. If $t_n = t_{n-1} + \Delta t$ and $t_n < \tilde{t}_n$ go to step 10 with $\theta^n = \theta^{n-1}$.
8. Initialize the new accepted points

$$NA_{\pm}^n = \{ I \in NB_{\pm}^{n-1} : \tilde{u}_I^{n-1} = \tilde{t}_n \}, \quad NA^n = NA_+^n \cup NA_-^n.$$

9. Reinitialize θ_I^n

$$\theta_I^n = \begin{cases} -\theta_I^{n-1} & \text{for } I \in NA^n, \\ \theta_I^{n-1} & \text{otherwise.} \end{cases}$$

10. Reinitialize $u_{I \rightarrow K}^n$

$$u_{I \rightarrow K}^n = \begin{cases} \min \{ u_{I \rightarrow K}^{n-1}, t_n \} & \text{if } I \in \mathcal{U}^n(K) \text{ and } K \in NB^n \\ +\infty & \text{otherwise.} \end{cases}$$

11. Set $n := n + 1$ and go to 4.

In this modified GFMM there is no need to update u^n at the whole front F^n . It suffices to update only these points which are useful for the propagation in the next step. Further, it is an important new idea that the time $u_{I \rightarrow K}^n$ at a point I depends also on its neighbours. Of course, one has to update the information about F^n and NB^n ; this is not explicitly stated but should be done in an efficient way, otherwise one loses optimal complexity.

REMARK 2.2.11:

In step 5 of the GFMM we distinguish two cases depending on the sign of \hat{c} to ensure that the evolution moves upwind. In the mGFMM there is no distinction on the sign of \hat{c} in step 4. This is already implicitly done in the definition of NB . The narrow-band is chosen in such a way that only upwind points are regarded and therefore we do not need to treat two cases.

REMARK 2.2.12:

We changed step 7 of mGFMM to get a working algorithm. In the original work [For09], an application to image segmentation [FLGGo8] and a preprint for the usage of the mGFMM in dislocation dynamics [CFM] step 7 was formulated as follows:

7. If $t_n = t_{n-1} + \Delta t$ and $t_n < \tilde{t}_n$ go to step 4 with $n := n + 1$, $\theta^n = \theta^{n-1}$ and $u^n = u^{n-1}$.

With this definition of step 7 it is possible that a front cannot change its direction. An explicit example is given in the following remark 2.2.13

REMARK 2.2.13:

In this remark we want to emphasize the meaning of useful points and the correct choice of step 7. This is done by a small explicit calculation that is loosely inspired by 2.2.10, see also [For09, Section 5.2].

Assume Δt and Δx small enough and regard the lattice $Q = \{x_i := i\Delta x, i \in \mathbb{Z}\}$. Let us define the speed

$$c(t, x) = \begin{cases} \max(0, \min(1, \frac{4}{3}x)) & \text{if } t \leq 1, \\ -1 & \text{if } t \geq \frac{17}{16}, \end{cases}$$

and interpolate c linear for constant x between $1 \leq t \leq \frac{17}{16}$. Thus we gain⁴ $c \in \mathcal{C}^{0,1}(\mathbb{R} \times [0, T])$.

The initial condition is

$$\theta_i^0 = \begin{cases} 1 & \text{if } i \leq 0, \\ -1 & \text{if } i \geq 1. \end{cases}$$

Now we compute the evolution governed by c .

First, we have $F_+^0 = \{0\}$, $F_-^0 = \{1\}$. Furthermore it holds $NB_+^0 = \emptyset$ because $\hat{c}_0^0 = 0$ and $NB_-^0 = \{1\}$ because $\hat{c}_1^0 = \frac{4}{3}\Delta x > 0$. Thus we get $\mathcal{U}^0 = \{0\}$ as well as $u_{0 \rightarrow 1} = 0$, all other values are $u_{\cdot \rightarrow \cdot} = \infty$.

The first point that will be accepted is $i = 1$ at time $t_1 = \frac{3}{4}$. We can number the new state with $n = 1$ because before t_1 nothing else would change.

Now we have $F_+^1 = \{1\}$, $F_-^1 = \{2\}$. We get $NB_+^1 = \emptyset$ because $\hat{c}_1^1 = \frac{4}{3}\Delta x \geq 0$ and $NB_-^1 = \{2\}$ because $\hat{c}_2^1 = \frac{8}{3}\Delta x > 0$. Thus $\mathcal{U}^1 = \{1\}$ and this leads to $u_{1 \rightarrow 2} = \frac{3}{4}$, and all other values $u_{\cdot \rightarrow \cdot} = \infty$.

The tentative value of point 2 is

$$\tilde{u}_2^1 = \frac{3}{4} + \frac{2\Delta x}{\frac{4}{3} \cdot 2\Delta x} = \frac{9}{8} > \frac{17}{16}.$$

Thus point 2 cannot be accepted because the speed changes its sign at time t with $1 \leq t \leq \frac{17}{16}$.

We can now assume that for a time t_2 with $1 \leq t_2 \leq \frac{17}{16}$ we have that $c(x, t_2) = -1 < 0$ for all $x \in \mathbb{R}$. Due to the change of the sign of the velocity, the sets NB and \mathcal{U} changes also. Thus we have at state $n = 2$ now the following:

⁴The restriction of c is just to ensure that c is Lipschitzian.

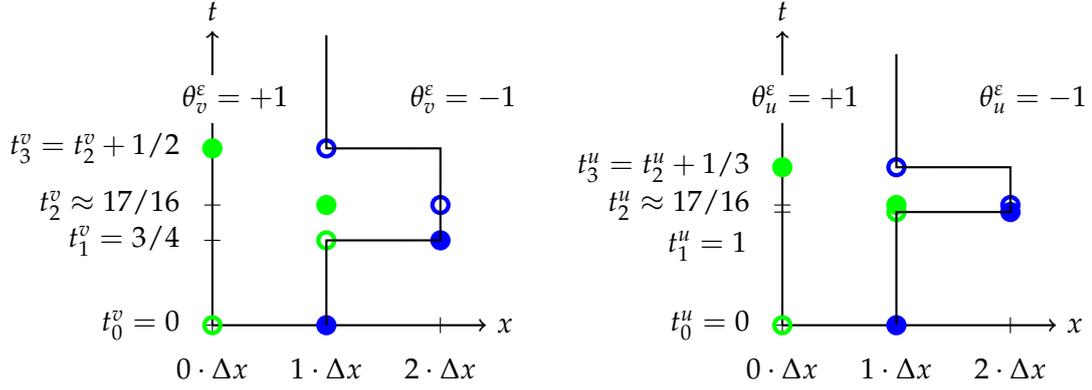


Figure 2.2.6: Evolution for c_v (left) and c_u (right) in remark 2.2.14. The front F_+ (green) and F_- (blue) are displayed as also the points of NB_+ (green filled circles) and NB_- (blue filled circles).

The front has not changed: $F_+^2 = \{1\}$, $F_-^2 = \{2\}$, but for the narrow-band we get $NB_+^2 = \{1\}$ because $\hat{c}_1^2 < 0$ and $NB_-^2 = \emptyset$ because $\hat{c}_2^2 < 0$. Thus $\mathcal{U}^1 = \{2\}$ and due to the update of the useful times at step 10 we get $u_{2 \rightarrow 1} = t_2$. All other values are $u_{\cdot \rightarrow \cdot} = \infty$.

Therefore the tentative value for point 1 is now $\tilde{u}_1^2 = t_2 + \frac{\Delta x}{1} < \infty$ and point 1 will be accepted again. So the algorithm keeps on working.

If we use the original description of the mGFMM we would have in state $n = 2$ the same front and narrow-band, but the useful times would not be updated and we would still have $u_{1 \rightarrow 2} = \frac{3}{4}$. All other useful times would still be $u_{\cdot \rightarrow \cdot} = \infty$ (i.e. $u_{2 \rightarrow 1} = \infty$). Thus we would get for the tentative value for point 1 just $\tilde{u}_1^2 = \infty$. The algorithm would terminate because nothing else could change anymore.

REMARK 2.2.14:

We continue here remark 2.2.10. Let us assume that we have the same speed c_v and c_u and also the same initial condition $\theta_v^0 = \theta_u^0$, but we use now the mGFMM.

The evolution of θ_v driven by c_v is as follows: We get that point 1 is accepted at $t_1^v = \frac{3}{4}$ and point 2 cannot be accepted as above. We can assume that the velocity c_v changes its sign at time $1 \leq t_2^v \leq \frac{17}{16}$. Thus point 1 will be accepted again at time $t_3^v = t_2^v + \frac{\Delta x}{2\Delta x} = t_2^v + \frac{1}{2}$.

For the evolution of u we get that point 1 is accepted at time $t_1^u = 1$ and point 2 cannot be accepted like above. Now we can assume that the speed changes its sign at time $1 \leq t_2^u \leq \frac{17}{16}$ with $|t_2^v - t_2^u| \leq \frac{1}{17}$. So point 1 is accepted again at time $t_3^u = t_2^u + \frac{\Delta x}{3\Delta x} = t_2^u + \frac{1}{3} < t_3^v$. Thus we see that this evolution copes with the comparison principle.

2.2.2.3 Properties

We cite here again two important properties of the mGFMM. The first one is the comparison principle. The other one is the convergence of this scheme. We use the

same definitions for θ^ε , $\bar{\theta}^0$ and $\underline{\theta}^0$ as for the GFMM.

Before we can state the comparison principle we first present a definition about comparable and compared times.

DEFINITION 2.2.15 ([For09, Definition 3.1]):

Let $T > 0$ and let us consider two velocities c_u and c_v . We denote by $(t_n)_n$ and $(u_{I \rightarrow K}^n)_n$ (resp. $(s_m)_m$ and $(v_{I \rightarrow K}^m)_m$) the sequences of times and of useful times associated to the velocity c_u (resp. c_v). We say that s_m and t_n are comparable if the property $C(t_n, s_m)$ holds true:

$$C(t_n, s_m) \text{ is true if } \begin{cases} t_n \leq s_m < t_{n+1} \text{ and } s_m < s_{m+1} \\ \text{or} \\ s_m \leq t_n < s_{m+1} \text{ and } t_n < t_{n+1} \end{cases}$$

We say that θ_v^m and θ_u^n are compared and we denote this by $\theta_v^m \succeq \theta_u^n$ if for all $I \in \mathbb{Z}^d$,

$$\begin{cases} \theta_{v,I}^m > \theta_{u,I}^n, \\ \text{or} \\ \begin{cases} \theta_{v,I}^m = \theta_{u,I}^n =: \sigma_I = \pm 1 \\ \text{and (with some obvious notation)} \\ \sigma_I u_{I \rightarrow K}^n \geq \sigma_I v_{I \rightarrow K}^m \text{ for all } K \in V(I) \setminus \{I\} \text{ such that } I \in \mathcal{U}_u^n(K) \cap \mathcal{U}_v^m(K). \end{cases} \end{cases}$$

If the times t_n and s_m are comparable then they are part of the strictly increasing sequence of times which are used in the definition of θ^ε .

We will use an analogous definition later on in subsection 2.4 for the proof of a comparison principle for the mGFT algorithm. Now we cite here the comparison principle:

THEOREM 2.2.16 ([For09, Theorem 3.4]):

Let $T > 0$ and consider the two velocities c_u and c_v . Given $\theta_{u,I}^0$ (resp. $\theta_{v,I}^0$) for all $I \in \mathbb{Z}^d$ and $u_{I \rightarrow K}^0$ for all $I \in \mathcal{U}_u^0(K), K \in NB_u^0$ (resp. $v_{I \rightarrow K}^0$ for all $I \in \mathcal{U}_v^0(K), K \in NB_v^0$), we assume that

$$\begin{aligned} t_0 &:= \sup_{I \in \mathcal{U}_u^0(K), K \in NB_u^0} u_{I \rightarrow K}^0 \leq T, \\ \left(\text{resp. } s_0 &:= \sup_{I \in \mathcal{U}_v^0(K), K \in NB_v^0} v_{I \rightarrow K}^0 \leq T \right) \end{aligned}$$

for a given $T \geq 0$.

We also assume that the two velocities satisfy for all $(t, x) \in [\min(t_0, s_0), T - \Delta t] \times \mathbb{R}^d$

$$\inf_{s \in [t, t + \Delta t]} c_v(s, x) \geq \sup_{s \in [t, t + \Delta t]} c_u(s, x).$$

If $C(t_0, s_0)$ and $\theta_v^0 \succeq \theta_u^0$, then

$$\theta_v^\varepsilon(t, x) \geq \theta_u^\varepsilon(t, x)$$

for all $(t, x) \in [\max(t_0, s_0), T] \times \mathbb{R}^d$.

This comparison principle is then used to prove the convergence result which we cite here. Here the same assumptions (A) are used as for theorem 2.2.9.

THEOREM 2.2.17 ([For09, Theorem 3.6]):

Under assumption (A), $\bar{\theta}^0$ (resp., $\underline{\theta}^0$) is a viscosity subsolution (resp., supersolution) of (2.0.1). In particular, if (2.0.1) satisfies a comparison principle, then $\bar{\theta}^0 = (\underline{\theta}^0)^*$ and $(\bar{\theta}^0)_* = \underline{\theta}^0$ is the unique discontinuous viscosity solution of (2.0.1).

2.3 Generalized fast marching method for unstructured triangulations

In this section I propose an adaption of the GFMM to unstructured triangulations (GFT). Taking a closer look at the GFMM (or the mGFMM) we see that this algorithm deals mainly with the concept of neighbourhood. So we discretize \mathbb{R}^2 not with a regular lattice Q but with an unstructured triangulation. Furthermore we do not use the 4-point neighbourhood any more but the concept of adjacent nodes in the triangulation. Another important point is the correct choice of an update formula like the one used in step 5 of GFMM and step 4 of mGFMM. Here we use the update scheme of [BR06]. See also subsection 2.1.2 for a short overview of the update formula for the eikonal equation for unstructured triangulations.

This section is organized as follows: In the first subsection 2.3.1 we propose the GFT algorithm. Therein, we give some preliminary definitions, propose the algorithm and then state and prove a weak comparison principle. In subsection 2.3.2 we present the modified GFT (mGFT) with the needed definitions and the algorithm. We propose also a comparison principle which we state and prove in section 2.4. The last subsection 2.3.3 is devoted to some comments on implementation and complexity of GFT and mGFT.

2.3.1 First version of GFT

The GFT is an adaption of the GFMM to unstructured grids. The definitions and the description of the GFT are analogous to the GFMM.

2.3.1.1 Definitions for GFT

Let us consider an acute triangulation \mathcal{T} of \mathbb{R}^2 . We denote by the \mathcal{N} the set of nodes of \mathcal{T} and by \mathcal{E} the set of edges of \mathcal{T} . We can interpret a triangulation of \mathbb{R}^2 as an undirected graph and therefore \mathcal{E} can be seen as a subset of $\{\{I, K\} : I, K \in \mathcal{N}\}$. Like in the GFMM we also use a time step $\Delta t > 0$. The following definitions are analogously to the definitions 2.2.1 to 2.2.4 for the GFMM.

DEFINITION 2.3.1:

We define the neighbourhood of the node $I \in \mathcal{N}$ as the set

$$V(I) := \{K \in \mathcal{N} : \{I, K\} \in \mathcal{E}\}.$$

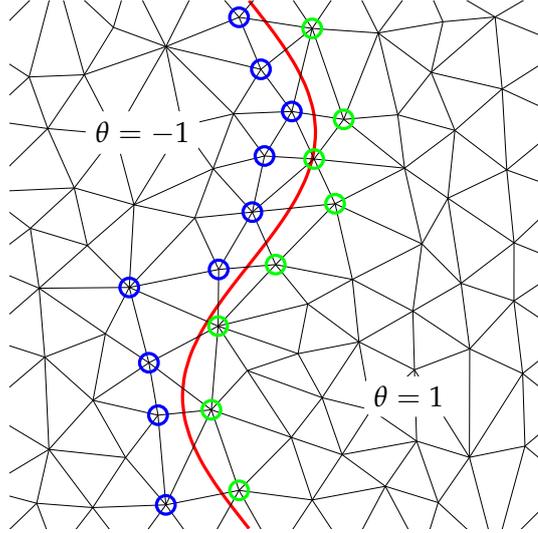


Figure 2.3.1: The front F_- (blue) and F_+ (green) in the GFT.

DEFINITION 2.3.2:

Given the speed $c_I^n := c(t_n, x_I)$, we define the regularized speed by the function

$$\tilde{c}_I^n := \begin{cases} 0 & \text{if there exists } K \in V(I) \text{ such that } (c_I^n c_K^n < 0) \text{ and } |c_I^n| \leq |c_K^n|, \\ c_I^n & \text{otherwise.} \end{cases}$$

DEFINITION 2.3.3:

Let $E \subset \mathcal{N}$. The numerical boundary ∂E of E is the set

$$\partial E := V(E) \setminus E,$$

with

$$V(E) := \{K \in \mathcal{N} : \exists I \in E \text{ such that } K \in V(I)\}.$$

DEFINITION 2.3.4:

Given a field θ_I^n with values $+1$ and -1 . We define the two phases

$$\Theta_{\pm}^n = \{I \in \mathcal{N} : \theta_I^n = \pm 1\}$$

and the fronts

$$F_{\pm}^n := \partial \Theta_{\mp}^n \text{ and } F^n := F_+^n \cup F_-^n.$$

We also use the same notions $\pm g \geq 0$ for $I \in F_{\pm}$, $\min_{\pm}\{0, g_{\pm}\}$ and $\max_{\pm}\{0, g_{\pm}\}$ as in the GFMM.

A typical situation of a front F is displayed in figure 2.3.1. Compare it also to figure 2.2.1 where we used the same situation but a regular lattice.

2.3.1.2 Algorithm GFT

Now the GFT can be written formally in exactly the same way as the GFMM except step 5 where we must use an appropriate extension of the update formula of Rouy and Tourin to unstructured grids. For this we use the Hopf-Lax updates described in subsection 2.1.2.

Initialization

1. Set $n = 1$.
2. Initialize the field θ^0 as

$$\theta^0 = \begin{cases} 1 & \text{for } x_I \in \Omega_0, \\ -1 & \text{elsewhere.} \end{cases}$$

3. Initialize the time on F^0 by $u_I^0 = 0$ for all $I \in F^0$.

Main cycle

4. Initialize \widehat{u}^{n-1} everywhere on the grid by

$$\widehat{u}_{\pm, I}^{n-1} = \begin{cases} u_I^{n-1} & \text{for } I \in F_{\pm}^{n-1}, \\ \infty & \text{elsewhere.} \end{cases}$$

5. Compute \tilde{u}^{n-1} on F^{n-1}
For $I \in F_{\pm}^{n-1}$ do

- a) if $\pm \widehat{c}_I^{n-1} \geq 0$ then $\tilde{u}_I^{n-1} = \infty$.
- b) if $\pm \widehat{c}_I^{n-1} < 0$ then we compute \tilde{u}_I^{n-1} via the Hopf-Lax update. We have to distinguish two cases:

- if $I \in F_-^{n-1}$, then $\tilde{u}_I^{n-1} = (\Lambda \widehat{u}_+^{n-1})(I)$ with the speed $|\widehat{c}_I^{n-1}|$.
- if $I \in F_+^{n-1}$, then $\tilde{u}_I^{n-1} = (\Lambda \widehat{u}_-^{n-1})(I)$ with the speed $|\widehat{c}_I^{n-1}|$.

6. $\tilde{t}_n = \min \{ \tilde{u}_I^{n-1}, I \in F^{n-1} \}$.

7. $\widehat{t}_n = \min \{ \tilde{t}_n, t_{n-1} + \Delta t \}$.

8. $t_n = \max \{ t_{n-1}, \widehat{t}_n \}$.

9. If $t_n = t_{n-1} + \Delta t$ and $t_n < \tilde{t}_n$ go to 4 with $n := n + 1$ and

$$\begin{cases} u_I^n = u_I^{n-1} & \text{for all } I \in F^n := F^{n-1}, \\ \theta_I^n = \theta_I^{n-1} & \text{for all } I \in \mathcal{N}. \end{cases}$$

10. Initialize new accepted points

$$NA_{\pm}^n = \left\{ I \in F_{\pm}^{n-1} : \tilde{u}_I^{n-1} = \tilde{t}_n \right\}, \quad NA^n = NA_+ \cup NA_-$$

11. Reinitialize θ_I^n

$$\theta_I^n = \begin{cases} -1 & \text{for } I \in NA_+^n, \\ 1 & \text{for } I \in NA_-^n, \\ \theta_I^{n-1} & \text{elsewhere.} \end{cases}$$

12. Reinitialize u^n on F^n

- a) If $I \in F^n \setminus V(NA^n)$ then $u_I^n = u_I^{n-1}$.
- b) If $I \in NA^n$ then $u_I^n = t_n$.
- c) If $I \in (F^{n-1} \cap V(NA^n)) \setminus NA^n$ then $u_I^n = u_I^{n-1}$.
- d) If $I \in V(NA^n) \setminus F^{n-1}$ then $u_I^n = t_n$.

13. Set $n := n + 1$ and go to step 4.

The interpretation for all variables are exactly the same as for the GFMM. Especially the usage of \hat{c} , \tilde{t} and \hat{t} is justified in the same way. The remarks 2.2.5, 2.2.6 and 2.2.10 work also for the GFT by embedding the one-dimensional situation in \mathbb{R}^2 .

2.3.1.3 Comparison principle for GFT

In this subsection we state and prove a weak comparison principle which is analogous to the one for the GFMM, see also proposition 2.2.7.

PROPOSITION 2.3.5 (Weak comparison principle for GFT):

We denote by u_I^n (resp., v_I^n) the numerical solution at the point (t_n, x_I) of the GFT algorithm with velocity c_u (resp., c_v). We assume that there exists $T > 0$ such that for all $(t, x) \in [0, T] \times \mathbb{R}^2$

$$\inf_{s \in [t-\Delta t], s \geq 0} c_v(s, x) \geq \sup_{s \in [t-\Delta t], s \geq 0} (c_u(s, x))^+$$

where f^+ is the positive part of f . Further we assume that

$$\{\theta_u^0 = 1\} \subset \{\theta_v^0 = 1\} \text{ and } v^0 = u^0 = 0.$$

We define \bar{m} and \bar{k} such that

$$\begin{cases} t_{\bar{m}} \leq T < t_{\bar{m}+1}, \\ s_{\bar{k}} \leq T < s_{\bar{k}+1}, \end{cases}$$

where $(t_m)_m$ and $(s_m)_m$ are the sequences of time constructed by the GFT algorithm with velocity c_u and c_v , respectively. We then consider

$$v_I = \begin{cases} v_i^0 & \text{if } \theta_{v,I}^0 = 1, \\ v_I^k & \text{if } I \in NA_v^k \text{ for some } k \leq \bar{k} + 1 \\ s_{\bar{k}+1} & \text{if } \theta_{v,I}^{\bar{k}} = -1. \end{cases}$$

Then, $\forall l \leq \bar{m}$, $\forall I \in NA_u^l$, we have $v_I \leq u_I^l$.

2 Generalized fast marching method

The proof of this proposition is motivated by the proof of the corresponding result for the GFMM, see also [CFFMo8, Proof of Proposition 3.2].

Proof. For a proof by contradiction we assume that there exists a minimal index μ with $I \in NA_{\mu}^{\mu}$ such that

$$u_I^{\mu} < v_I. \quad (2.3.1)$$

Now we define κ such that $I \in NA_{\bar{v},I}^{\kappa}$. In the case that $\theta_{\bar{v},I}^{\kappa} = -1$ we set $\kappa = \bar{k} + 1$, that means the point I has not been accepted till the time T . Thus we know that

$$t_{\mu} = u_I^{\mu} < v_I = s_{\kappa}. \quad (2.3.2)$$

We distinguish now two cases.

Case 1: $I \in NA_{+,u}^{\mu} \subset F_{+,u}^{\mu-1}$

Either we have $I \in \{\theta_v^0 = 1\}$, then we obtain $v_I = v_I^0 = 0 = u_I^0 \leq u_I^{\mu}$, which contradicts (2.3.1), or we have instead $I \in \{\theta_v^0 = -1\}$ then we have by the assumption $\{\theta_u^0 = 1\} \subset \{\theta_v^0 = 1\}$ that $\theta_u^0 = -1$. Therefore we know that there exists $n < \mu$ with

$$\theta_{u,I}^{n-1} = -1 \quad \text{and} \quad \theta_{u,I}^n = 1.$$

Thus we deduce $u_I^n \geq v_I > u_I^{\mu} \geq u_I^n$, which is absurd.

Case 2: $I \in NA_{-,u}^{\mu} \subset F_{-,u}^{\mu-1}$

We prove the second case in four steps.

(i) In the first step, we show that for all $K \in V(I) \setminus \{I\}$

$$\hat{u}_{+,K}^{\mu-1} \geq \hat{v}_{+,K}^{\kappa-1} \quad (2.3.3)$$

holds. We may assume that $\hat{u}_{+,K}^{\mu-1} < \infty$, because otherwise (2.3.3) is trivially true.

This means that $K \in F_{+,u}^{\mu-1}$, see step 4 of the algorithm. Thus we have

$$t_{\mu} \geq \hat{u}_{+,K}^{\mu-1} \geq v_K. \quad (2.3.4)$$

The first inequality comes from the fact that the sequence t_n is non-decreasing and the second inequality is due to the minimality of μ in (2.3.1).

Now we must show $v_K \geq \hat{v}_{+,K}^{\kappa-1}$. If we have $\hat{v}_{+,K}^{\kappa-1} < \infty$ then we know $K \in F_{+,v}^{\kappa-1}$. Thus K has been already accepted and therefore we get $v_K = \hat{v}_{+,K}^{\kappa-1}$, because v is a non-negative velocity and a point can only be passed once by the front. If instead $\hat{v}_{+,K}^{\kappa-1} = \infty$, then we know $\theta_{v,K}^{\kappa-1} = -1$. Thus we get $v_K \geq s_{\kappa}$ because point K has not been accepted, but this contradicts $v_K \leq t_{\mu} < s_{\kappa}$ by (2.3.2). So the case $\hat{v}_{+,K}^{\kappa-1} = \infty$ is not possible.

Thus we have showed (2.3.3) and the first step is completed.

(ii) In this step we show for all $K \in V(I) \cap F_{+,u}^{\mu-1}$

$$s_{k^*} \geq \widehat{v}_{+,K}^{\kappa-1} \quad (2.3.5)$$

with

$$k^* := \sup\{k, s_k \leq t_\mu\} < \kappa.$$

With (2.3.3) and (2.3.4) from step (i) we have

$$t_\mu \geq \widehat{u}_{+,K}^{\mu-1} \geq \widehat{v}_{+,K}^{\kappa-1}.$$

Now let us assume that $\infty > \widehat{v}_{+,K}^{\kappa-1} > s_{k^*}$, the case $\infty = \widehat{v}_{+,K}^{\kappa-1}$ could not happen, see step (i). Then this would imply that there exists $k' > k^*$ such that $t_\mu \geq \widehat{v}_{+,K}^{\kappa-1} = s_{k'}$. This contradicts the definition of k^* and therefore (2.3.5) holds.

(iii) Next we prove that for all $K \in V(I) \cap F_{+,u}^{\mu-1}$

$$\widehat{v}_{+,K}^{\kappa-1} = \widehat{v}_{+,K}^{k^*}. \quad (2.3.6)$$

holds. Due to $\widehat{v}_{+,K}^{\kappa-1} < \infty$, see step (i), we know that $\theta_{v,K}^{\kappa-1} = 1$. Now look at the iteration k in which point K was accepted. Either we have $k = 0$, that means $\theta_{v,K}^0 = 1$ and therefore $\widehat{v}_{+,K}^{\kappa-1} = v_K^0 = 0$, or in the case $k \geq 1$ we have $K \in NA_v^k$ and thus $\widehat{v}_{+,K}^{\kappa-1} = v_K^k = s_k$.

If we assume that $k > k^*$, then we get

$$\widehat{v}_{+,K}^{\kappa-1} = v_K^k = s_K \geq s_{k^*+1} > s_{k^*}$$

which contradicts (2.3.5) from step (ii), and thus we know $k \leq k^*$. Now we know that there exists a $m \leq \kappa - 1$ such that $\theta_{v,I}^m = -1$ and $\theta_{v,I}^\kappa = 1$, see also the definition of κ . Thus we get $K \in F_{+,v}^{k^*}$ and therefore

$$\widehat{v}_{+,K}^{\kappa-1} = v_K^k = \widehat{v}_{+,K}^{k^*}.$$

(iv) Now we construct a contradiction. With (2.3.3) from step (i) and (2.3.6) from step (iii) we get

$$\widehat{v}_{+,K}^{\kappa-1} = \widehat{v}_{+,K}^{k^*} \leq \widehat{u}_{+,K}^{\mu-1}.$$

We use the monotonicity of the update formula (2.1.6) to conclude $\widehat{v}_K^{k^*} \leq \widehat{u}_K^{\mu-1}$. Therefore we have $s_{k^*+1} \leq \widehat{v}_K^{k^*} \leq \widehat{u}_K^{\mu-1} \leq u_K^\mu = t_\mu$ which contradicts the definition of k^* . Thus proposition 2.3.5 is proven. \square

Let us use the function θ^ε as defined in definition 2.4.1. Then we have a comparison principle for θ^ε , namely

COROLLARY 2.3.6:

Under the assumptions of proposition 2.3.5, we have for all $(t, x) \in [0, T] \times \mathbb{R}^2$

$$\theta_u^\varepsilon(t, x_I) \leq \theta_v^\varepsilon(t, x_I).$$

2 Generalized fast marching method

The proof is exactly the same as in [CFFMo8, Corollary 3.5].

Of course using velocities which are negative instead of the positive velocities as in the preceding results we gain also a comparison principle like [CFFMo8, Corollary 3.6]. This is just a direct consequence of the symmetry of the GFT.

COROLLARY 2.3.7:

We denote by u_1^n (resp., v_1^n) the numerical solution at the point (t_n, x_1) of the GFT algorithm with velocity c_u (resp., c_v). We assume that there exists $T > 0$ such that for all $(t, x) \in [0, T] \times \mathbb{R}^2$

$$\sup_{s \in [t-\Delta t], s \geq 0} c_u(s, x) \leq \inf_{s \in [t-\Delta t], s \geq 0} -(c_v(s, x))^-$$

where $f^- \geq 0$ is the negative part of f . Further we assume that

$$\{\theta_v^0 = -1\} \subset \{\theta_u^0 = -1\} \text{ and } v^0 = u^0 = 0.$$

Then, for all $(t, x) \in [0, T] \times \mathbb{R}^2$, we have

$$\theta_u^\varepsilon(x, t) \leq \theta_v^\varepsilon(x, t).$$

2.3.1.4 Remarks

We want to give two remarks on the GFT. The aim of the first remark is to extend the algorithm to the d -dimensional case and in the second remark we adapt the method of virtual updates such that we can also use the GFT for a non-acute triangulation.

REMARK 2.3.8:

The first remark is that this algorithm can be formulated also in \mathbb{R}^d . To get a working algorithm we do not use a triangulation anymore, but a partition of \mathbb{R}^d into simplexes. The neighbourhood $V(I)$ is then similarly defined as the set of points I such that $\{I, K\}$ is an edge of a simplex. We also have to use an appropriate update-scheme. For this we use the Hopf-Lax update as in [BR06, Section 4]. This Hopf-Lax update also has a monotonicity property. So the comparison principle 2.3.5 and also its proof holds in the d -dimensional setting. The proof uses only the notion of neighbourhood and the monotonicity of the update scheme but no notion that is inherent 2-dimensional. I implement the GFT only in the 2-dimensional case; thus I restricted the presentation to $d = 2$, but the idea of the algorithm works for any finite dimension. Also the GFMM is presented in \mathbb{R}^d but the numerical examples are all in two dimensions.

REMARK 2.3.9:

Till now we have the restriction that the GFT can only deal with an acute triangulation. Also the FMM on unstructured grids has this restriction, but there one can use the method of virtual triangles to construct on two acute triangles the fly which replace a non-acute triangle.

It is no problem to construct virtual triangles in the GFT because this is a pure geometric problem and therefore can be carried out in exactly the same way as in the FMM. In the FMM the u -values we use for the update are defined on all points, especially on all points that are already accepted. But in the GFT the values of u

we use for the the updates are restricted to the front. So at first glance there are no usable values \hat{u} outside the front. This problem is only an issue due to the algorithmic description of the GFT. It is no difference for an acute triangulation if we define \hat{u}_\pm for all points. This can be done by replacing step 3 and 4 with the following steps 3' and 4':

3'. Initialize the time by

$$u_I^0 = \begin{cases} 0 & \text{if } I \in F^0, \\ \infty & \text{otherwise.} \end{cases}$$

4'. Initialize \hat{u}^{n-1} everywhere on the grid by

$$\hat{u}_{\pm, I}^{n-1} = \begin{cases} u_I^{n-1} & \text{for } I \in \Theta_\pm^{n-1}, \\ \infty & \text{elsewhere.} \end{cases}$$

With this definition we can access useful values of \hat{u} on the whole domain and thus we can use the strategy of virtual updates.

We want to emphasize that using virtual update we do not calculate the Hopf-Lax update on $\omega(C)$ for a point C exactly but only a good approximation, see also subsection 2.1.3. Viewing only numeric results we can deal with non-acute triangulations by using virtual updates. The example 2.5.2.5 in section 2.5 is devoted to the usage of non-acute triangulations and virtual updates. In theory we are still restricted to an acute triangulation, because using virtual updates we cannot guarantee the monotonicity of the update formula and with a non-acute triangulation the causality property is not fulfilled anymore.

2.3.2 Modified generalized fast marching method for unstructured triangulations (mGFT)

The GFT algorithm is based on the GFMM for unstructured grids. As with the GFMM the GFT fulfills only a weak comparison principle and the counterexample of Forcadel [For09, Section 5.2] applies here too. Therefore we adapt the idea of the mGFMM to build a version of the GFT that copes with a comparison principle. This comparison principle is stated and proven in section 2.4.

2.3.2.1 Definition mGFT

We need some preliminary definitions before we can state the mGFT. As in the definition of the mGFMM where we reused the definitions 2.2.1 to 2.2.4 we can reuse the definitions 2.3.1 to 2.3.4 in the case of unstructured triangulations.

We first introduce the notion of the narrow-band, which connects the information about θ with the speed \hat{c} .

DEFINITION 2.3.10:

Given a field θ^n and the regularized speed \hat{c}^n from definition 2.3.2. We define the narrow-band NB by the set of points which can be reached immediately by the front, that is

$$NB^n = \{I \in \mathcal{N} : \exists K \in V(I) \text{ with } \theta_I^n = -\theta_K^n \text{ and } \theta_I^n \hat{c}_I^n < 0\}.$$

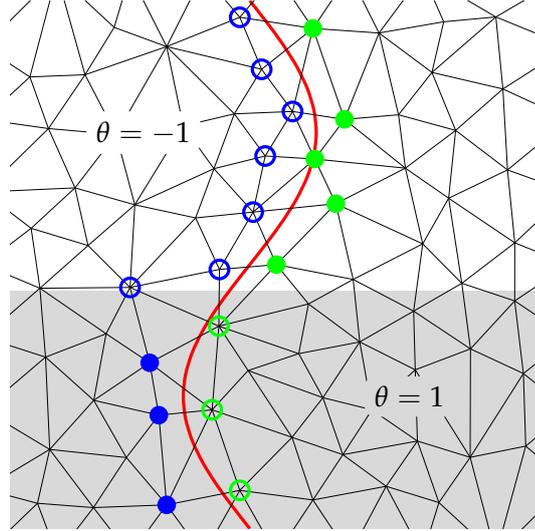


Figure 2.3.2: The front F_- (blue) and F_+ (green) in the mGFT. The points of NB_- (blue filled circles) and NB_+ (green filled circles) are part of the front F_- resp. F_+ . In the grey region is the speed c positive, in the white region negative. Compare this figure also with figure 2.3.1 where the same situation is displayed for the GFT.

Furthermore we set

$$NB_{\pm}^n = NB^n \cap \{I \in \mathcal{N} : \theta_I^n = \pm 1\}.$$

The narrow-band NB_{\pm}^n can also be written in the form

$$NB_+^n = F_+ \cap \{I \in \mathcal{N} : \hat{c}_I^n < 0\} \text{ and } NB_-^n = F_- \cap \{I \in \mathcal{N} : \hat{c}_I^n > 0\}.$$

As in the GFMM we introduce now the notion of useful points. For every $I \in NB^n$ one computes a tentative value \tilde{u}_I^n of the arrival time at this point I . So we define the set of useful points by

$$\mathcal{U}^n(K) = \{I \in V(K) : \theta_K^n = -\theta_I^n\}, \quad \mathcal{U}^n = \cup_{K \in NB^n} \mathcal{U}^n(K).$$

For each useful point $K \in NB^n$ we introduce further the useful time $u_{I \rightarrow K}^n$. This value can be seen as the time when the fronts F^n goes from node $I \in V(K)$ to K . We use this notion for the computation of the tentative arrival time at point K .

2.3.2.2 Algorithm mGFT

Now we can state the algorithm for the mGFT. As one can see the algorithm has the very same structure as the mGFMM.

Initialization

1. Set $n = 1$.

2. Initialize the field θ^0 as

$$\theta^0 = \begin{cases} 1 & \text{for } x_I \in \Omega_0, \\ -1 & \text{elsewhere.} \end{cases}$$

3. Initialize the time for points I

$$u_{I \rightarrow K}^0 = \begin{cases} t_0 & \text{it } I \in \mathcal{U}^0(K) \text{ and } K \in NB^0, \\ +\infty & \text{otherwise.} \end{cases}$$

Loop

4. Compute \tilde{u}^{n-1} on NB^{n-1}

Let $I \in NB^{n-1}$, then we compute \tilde{u}_I^{n-1} via the Hopf-Lax update by using the information about the useful times $u_{\cdot \rightarrow I}$ for point I .

$$\tilde{u}_I^{n-1} = (\Lambda u_{\cdot \rightarrow I}^{n-1})(I) \text{ with the speed } |\hat{c}_I^{n-1}|.$$

5. $\tilde{t}_n = \min \{ \tilde{u}_I^{n-1} : I \in NB^{n-1} \}$.

6. Truncate \tilde{t}_n

$$t_n = \max \{ t_{n-1}, \min \{ \tilde{t}_n, t_{n-1} + \Delta t \} \}.$$

7. if $t_n = t_{n-1} + \Delta t$ and $t_n < \tilde{t}_n$ go to step 10 with $\theta^n = \theta^{n-1}$.

8. Initialize new accepted points

$$NA_{\pm}^n = \left\{ I \in NB_{\pm}^{n-1} : \tilde{u}_I^{n-1} = \tilde{t}_n \right\}, \quad NA^n = NA_+ \cup NA_-.$$

9. Reinitialize θ_I^n

$$\theta_I^n = \begin{cases} -\theta_I^{n-1} & \text{for } I \in NA^n, \\ \theta_I^{n-1} & \text{elsewhere.} \end{cases}$$

10. Reinitialize $u_{I \rightarrow K}^n$

$$u_{I \rightarrow K}^n = \begin{cases} \min \left(u_{I \rightarrow K}^{n-1}, t_n \right) & \text{if } I \in \mathcal{U}(K) \text{ and } K \in NB^n, \\ +\infty & \text{otherwise.} \end{cases}$$

11. Set $n := n + 1$ and go to step 4.

2.3.2.3 Comments

Like the GFT the mGFT can also be extended to \mathbb{R}^d . This can be done in the exact same way as presented in remark 2.3.8. Thus also the comparison principle 2.4.5 holds also in \mathbb{R}^d .

Now we come to a point where GFT and mGFT are essentially different. In the GFT it is possible to use the method of virtual updates to deal with non-acute triangulations because the extension of \hat{u} to the whole domain is straight forward. In the mGFT we run into problems if we would extend the useful times. The useful times $u_{I \rightarrow K}$ for a point K depends also on its neighbours I . If we would use the virtual updates we should be able to define the useful time at the point K not only for its neighbours in $V(K)$, but for all points $I \in \mathcal{N}$ in the whole computational domain. So the effort to store and update the information about useful times would be at least quadratic in a naive implementation and thus the algorithm would not be efficient anymore. In a more sophisticated implementation one could reduce this effort because the complexity of calculating a single virtual triangle is bounded by a constant [KS98, Section 4] and one also has to traverse only a bounded number of triangles. But still the effort for the implementation would be high and using virtual updates we still would lose the comparison principle which was the motivation for constructing this method.

Therefore it seems to be reasonable to use the mGFT only on acute triangulations, but the GFT can be used with general triangulations.

2.3.3 Computational complexity

In this subsection we will give a short overview of the complexity of GFT and mGFT. In essence, we will obtain the same results as presented in [CFFMo8, Subsection 2.3] for the GFMM.

We assume that the speed c is constant for the time interval $[k\Delta t, (k+1)\Delta t)$ with some Δt . This assumption is not too strict, because due to the discretization every function can be seen as constant for a small period of time. Furthermore we use a mesh with M points that is contained in the unit-interval $[0, 1]^d$. Assume further that the speed is normalized to $|c| \simeq 1$ and that the time T which the front needs to traverse the grid is also normalized to $T \simeq 1$. The number of mesh points that are typically contained in the front F (or narrow-band NB for mGFT) is of size $M^{\frac{d-1}{d}}$. Further the typical length of an edge is about $M^{-1/d}$.

There are 3 different cases for Δt :

Case (a) $\Delta t = \infty$

In this case the speed c is independent in time. Therefore the algorithm behaves like a doubled FMM and we need only to update in step 5 the points $I \in V(NA^n)$, cause all other values \tilde{u} cannot change due the constant of the speed in time. Also in step 4, 11 and 12 we need only to update points that are accepted and their neighbours. Furthermore we can use the min-heap to get the point with minimal value of \tilde{u} . So we can recover the complexity of the FMM and gain for GFT and mGFT a complexity of $\mathcal{O}(M \log M)$. We have to modify the usual data structure for the min-heap because we have also to remove elements of the heap which are not at its root. This is done by adding

to every point a reference to its position in the heap; so we can directly access any point in the heap and it can be removed similar to an element which is removed from the heap by the extract-min operation.

Case (b) $\mathcal{O}(M^{-1/d}) \leq \Delta t \leq \infty$

This case is very similar to the case above. During the interval $[k\Delta t, (k+1)\Delta t)$ we have the same behavior as in case (a). So the effort in this period is $\mathcal{O}(M_k \log M)$ where M_k denotes the number of points that have been crossed by the front during this period. We normalized the time such that the front passes the whole domain in about $T \simeq 1$, so we can assume that we have about $\frac{T}{\Delta t} = \frac{1}{\Delta t}$ time intervals of the form $[k\Delta t, (k+1)\Delta t)$ and in total we cross $M = \sum_k M_k$ points. At time $k\Delta t$ we also recompute the tentative values on the whole front (or narrow-band for the mGFT), which requires an effort of $\mathcal{O}(M^{\frac{d-1}{d}} \log M)$ every time.

Thus we get

$$\sum_{k=1}^{1/\Delta t} \mathcal{O}(M_k \log M) + \sum_{k=1}^{1/\Delta t} \mathcal{O}(M^{\frac{d-1}{d}} \log M) = \mathcal{O}((M + M^{\frac{d-1}{d}} / \Delta t) \log M).$$

As long as $\Delta t > M^{-1/d}$ we get an overall complexity of $\mathcal{O}(M \log M)$ which is the same order as in case (a).

Case (c) $0 \leq \Delta t < \mathcal{O}(M^{-1/d})$

In this case the speed varies very strongly and therefore it will no longer be efficient to use a heap. Due to the changes of the speed we recompute the tentative values for every point in the front at each iteration n , and for the mGFT (and the mGFMM) we have to check which points are in the narrow-band and which are not. Thus we have in every iteration an effort of $\mathcal{O}(M^{\frac{d-1}{d}})$. Therefore we get for the overall complexity $\mathcal{O}(M^{\frac{2d-1}{d}})$. In this case a level-set-method may be faster than the generalized fast marching methods.

In section 2.5 we carry out a numerical experiment where we can see the effect of these different cases. In case (b) we used a heap for the extraction of the nodes with minimal \tilde{u} value. In principle it would be possible to use an untidy priority queue instead of the heap as proposed in [YBS05]. This could slightly decrease the computational effort but it would also introduce new errors as presented in [RS09]. Furthermore the reduction of complexity can probably be neglected because the relation of operations for the priority queue to the total number of operations is very low.

2.4 Comparison principle for mGFT

In this section we prove a comparison principle of the discrete evolution for the mGFT, see theorem 2.4.5. This comparison principle is analogous to the one stated and proved in [For09, Theorem 3.4], see also theorem 2.2.16.

We adapt here the proof which Forcadel presents in [For09]. In essence we can do the very same and nearly copy his proof. This may seem curious at first sight, because

we deal here with a triangulation (or with simplexes in \mathbb{R}^d) and Forcadel deals with a regular grid. And indeed a priori we do not know whether the proof of Forcadel can be conveyed to our situation.

In the proof of [For09, Theorem 3.4] (see also Theorem 2.2.16) Forcadel uses a comparison principle for the times t_n [For09, Theorem 5.1] and a method for extending the discrete function to $[0, \infty) \times \mathbb{R}^d$, see also the construction of θ^ε in 2.2.1.3. The proof of [For09, Theorem 5.1] uses the monotonicity of the update scheme and some auxiliary results on the sequences of time t_n .

To prove the comparison principle 2.4.5 for the mGFT, we first introduce how we can extend the discrete function θ_I^n to a function θ^ε on the whole domain. For this step we have only few restrictions, so we can easily define such an extension. In the next step we prove theorem 2.4.5 under the assumption that we have a comparison principle for our times t_n , see also theorem 2.4.9. In the proof of theorem 2.4.9 we use properties of the discrete evolution computed by the algorithm. These properties are collected in some lemmata and propositions and we can use all the methods which Forcadel used. The monotonicity of the update scheme is only used in lemma 2.4.13, all other results depends on the description of the algorithm and the notion of neighbourhood.

A posteriori we can see that this is possible because in the proof of these auxiliary results Forcadel used the concept of neighbourhood and the connection between the values at the grid points due to the algorithm. Forcadel used throughout the proof always the discrete point of view and especially he did not use any kind of analytical construction. Now I transferred these notions of neighbourhood and the algorithm in a suitable way to unstructured grids, so we can recycle the proof of Forcadel because on a discrete point of view the algorithms mGFMM and mGFT only differ in step 4 where the tentative values are computed. But the important point of step 4 is that the updates are monotone, and this property is preserved. Therefore one can say that the proof of Forcadel can essentially be used in a much more general setting than in his paper [For09].

Now we construct θ^ε . The sequence of time $\{t_n, n \in \mathbb{N}\}$ defined in step 6 of the mGFT is by construction non-decreasing and we can extract a subsequence $\{t_{n_k}, k \in \mathbb{N}\}$ which is strictly increasing such that

$$t_{n_k} = t_{n_{k+1}} = \dots = t_{n_{k+1}-1} < t_{n_{k+1}}.$$

Hence the sequence t_{n_k} is defined in exactly the same way as for the GFMM and the mGFMM. We cannot directly reuse the definition of the square cell S_I^k because we work on an unstructured grid. We replace the square cells by the patches \mathcal{P}_I^k which we define as follows:

We denote by \mathcal{P}_I^k the patch $[t_{n_k}, t_{n_{k+1}}[\times \mathcal{P}_I$. Thereby \mathcal{P}_I is a union of triangles which has the point I as a common corner. Furthermore we require that $\bigcup \mathcal{P}_I = \mathbb{R}^2$, that means the patches cover \mathbb{R}^2 . A second condition is, that we assume $\mathcal{P}_I \cap \mathcal{P}_K = \emptyset$ for $I \neq K$. To achieve this the edges of the triangulation can only belong to one patch and therefore a patch can be on one side open and on the other closed.

For the comparison principle it would even suffice that \mathcal{P}_I is just any subset of \mathbb{R}^2 with $I \in \mathcal{P}_I$ and that $(\mathcal{P}_I)_{I \in \mathcal{N}}$ is a partition of \mathbb{R}^2 . This is because we need for the comparison principle only a mapping that links every point in \mathbb{R}^2 to a node in \mathcal{N} .

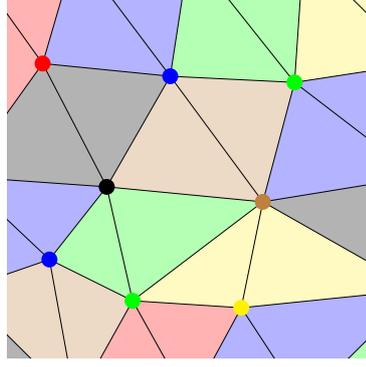


Figure 2.4.1: Example for the mapping of triangles to nodes for the construction of the patches.

DEFINITION 2.4.1:

Let h_{\max} the maximal diameter of all triangles in \mathcal{T} and $\mathcal{P}_I^k := [t_{n_k}, t_{n_{k+1}}[\times \mathcal{P}_I$ the patch \mathcal{P}_I with the times t_{n_k} from the strictly increasing sequence of times. We denote by ε the couple

$$\varepsilon = (h_{\max}, \Delta t).$$

Then we define θ^ε by

$$\theta^\varepsilon(t, x) = \theta_I^{n_{k+1}-1} \text{ if } (t, x) \in \mathcal{P}_I^k.$$

We remark that the construction of θ^ε is exactly the same for the GFT, so there is no difference between these two methods.

Now we introduce the notion of comparable and compared times. This is the same notion as used for the mGFMM, see also definition 2.2.15.

DEFINITION 2.4.2 (Comparable and compared times):

Let $T > 0$ and let us consider two velocities c_u and c_v . We denote by $(t_n)_n$ and $(u_{I \rightarrow K}^n)_n$ (resp. $(s_m)_m$ and $(v_{I \rightarrow K}^m)_m$) the sequences of times and of useful times associated to the velocity c_u (resp. c_v). We say that s_m and t_n are comparable if the property $C(t_n, s_m)$ holds true. This property is

$$C(t_n, s_m) \text{ is true if } \begin{cases} t_n \leq s_m < t_{n+1} \text{ and } s_m < s_{m+1} \\ \text{or} \\ s_m \leq t_n < s_{m+1} \text{ and } t_n < t_{n+1}. \end{cases}$$

We say that θ_v^m and θ_u^n are compared and we denote this by $\theta_v^m \succeq \theta_u^n$ if for all points $I \in \mathcal{N}$ holds

$$\begin{cases} \theta_{v,I}^m > \theta_{u,I}^n, \\ \text{or } \begin{cases} \theta_{v,I}^m = \theta_{u,I}^n =: \sigma_I = \pm 1 \\ \text{and} \\ \sigma_I u_{I \rightarrow K}^n \geq \sigma_I v_{I \rightarrow K}^m \text{ for all } K \in V(I) \text{ such that } I \in \mathcal{U}_u^n(K) \cap \mathcal{U}_v^m(K). \end{cases} \end{cases}$$

2 Generalized fast marching method

We want to emphasize that $C(t_n, s_m)$ can only be true if t_n and s_m are elements from the strictly increasing sequence of times which we used for the construction of the patch \mathcal{P}_I^k . This construction was made to ensure that algorithm cannot change the front without advancing in time.

REMARK 2.4.3:

The notation $\theta_v^m \succeq \theta_u^n$ may be misleading because \succeq does not only depend on the values of θ but the values of v^m and u^n are also regarded. We have further that $\theta_v^m \succeq \theta_u^n$ implies $\theta_v^m \geq \theta_u^n$.

Using the initial values $u_{I \rightarrow K}^0 = 0$ for all $I \in \mathcal{U}_u^0(K)$, $K \in NB_u^0$ and $v_{I \rightarrow K}^0$ for all $I \in \mathcal{U}_v^0(K)$, $K \in NB_v^0$ we get that $\theta_v^0 \succeq \theta_u^0$ is equivalent to $\theta_{v,I}^0 \geq \theta_{u,I}^0$ for all I . In all numerical examples in section 2.5 we use this zero initial condition.

REMARK 2.4.4:

In the definition of the comparable times we use the inequality $\sigma_I u_{I \rightarrow K}^n \geq \sigma_I v_{I \rightarrow K}^m$ which depends on the sign of $\theta_{v,I}^m$. The reason is that we look at which point will be accepted earlier. For $\sigma_I = 1$ the point K has to be accepted for c_v before for c_u , thus we ask for $u_{I \rightarrow K}^n \geq v_{I \rightarrow K}^m$. Vice versa, when $\sigma_I = -1$, the point K has to be accepted for c_u before for c_v and then we need the reversed inequality.

2.4.1 Comparison principle for θ^ε

THEOREM 2.4.5 (Comparison principle for the θ^ε):

Let $T > 0$ and consider the two velocities c_v and c_u . Given $\theta_{u,I}^0$ (resp. $\theta_{v,I}^0$) for all $I \in \mathcal{N}$ and $u_{I \rightarrow K}^0$ for all $I \in \mathcal{U}_u^0(K)$, $K \in NB_u^0$ (resp. $v_{I \rightarrow K}^0$ for all $I \in \mathcal{U}_v^0(K)$, $K \in NB_v^0$), we assume that

$$t_0 := \sup_{I \in \mathcal{U}_u^0(K), K \in NB_u^0} u_{I \rightarrow K}^0 \leq T$$

$$\left(\text{resp. } s_0 := \sup_{I \in \mathcal{U}_v^0(K), K \in NB_v^0} v_{I \rightarrow K}^0 \leq T \right)$$

for a given $T \geq 0$.

We also assume that the two velocities satisfy for all $(t, x) \in [\min(t_0, s_0), T - \Delta t] \times \mathbb{R}^2$

$$\inf_{s \in [t, t + \Delta t]} c_v(s, x) \geq \sup_{s \in [t, t + \Delta t]} c_u(s, x).$$

If $C(t_0, s_0)$ and $\theta_v^0 \succeq \theta_u^0$, then

$$\theta_v^\varepsilon(t, x) \geq \theta_u^\varepsilon(t, x)$$

for all $(t, x) \in [\max(t_0, s_0), T] \times \mathbb{R}^2$.

Proof. We prove this theorem by contradiction. For this we define

$$t^* = \inf\{t \geq \max(t_0, s_0) \text{ such that } \theta_v^\varepsilon(t, x) < \theta_u^\varepsilon(t, x) \text{ for some } x \in \mathbb{R}^2\}$$

and

$$x^* \text{ such that } \theta_v^\varepsilon(t^*, x^*) < \theta_u^\varepsilon(t^*, x^*).$$

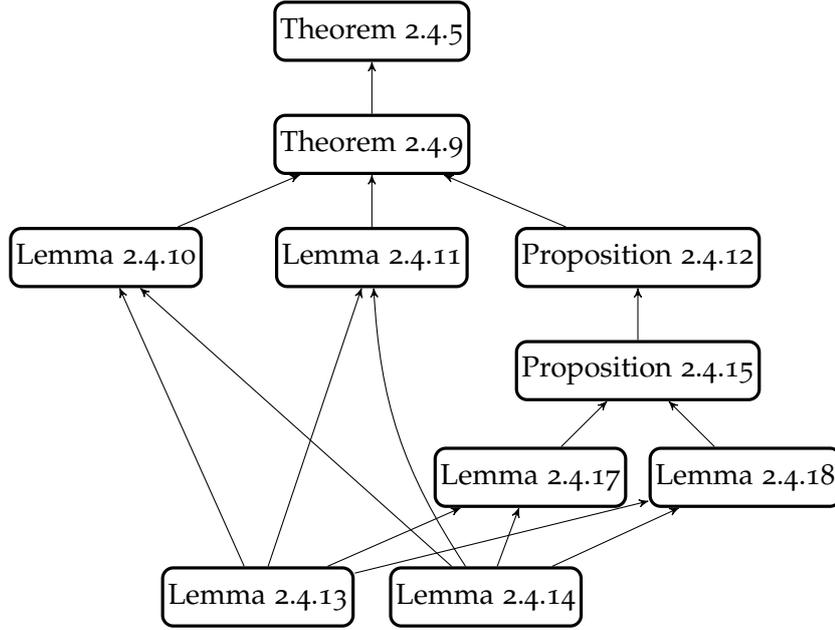


Figure 2.4.2: Structure of the proof of theorem 2.4.5.

Now we choose m , n and I such that

$$\begin{cases} s_m \leq t^* < s_{m+1}, \\ t_n \leq t^* < t_{n+1}, \\ x^* \in \mathcal{P}_I. \end{cases}$$

Then $\theta_v^\varepsilon(t, x^*) = \theta_{v,I}^m$ for $t \in [s_m, s_{m+1})$ and $\theta_u^\varepsilon(t, x^*) = \theta_{u,I}^n$ for $t \in [t_n, t_{n+1})$. Therefore $t^* = s_m$ or $t^* = t_n$ and hence $C(t_n, s_m)$ is true. Theorem 2.4.9 says that $\theta_v^m \succeq \theta_u^n$ and thus

$$\theta_{v,I}^m \geq \theta_{u,I}^n \text{ for all } I \in \mathcal{N}$$

which contradicts the construction of t^* and x^* . \square

We used in this proof theorem 2.4.9. The proof of this theorem depends also on some further results. The dependence is displayed in figure 2.4.2.

2.4.2 Basic properties for mGFT

We list in this subsection some basic properties of the mGFT. We will use these results in the following proofs.

PROPOSITION 2.4.6 (Basic properties of mGFT):

Using the mGFT we get the following properties:

1. $0 \leq t_n - t_{n-1} \leq \Delta t$.

2 Generalized fast marching method

2. For all points $K \in \mathcal{N}$ and $I \in \mathcal{U}^n(K)$, we have

$$u_{I \rightarrow K}^n \leq t_n.$$

3. $NB^n \cap \mathcal{U}^n = \emptyset$.

4. If $I \in NA^n$, then

$$u_{I \rightarrow K}^n = \begin{cases} t_n & \text{if } I \in \mathcal{U}^n(K), \\ +\infty & \text{otherwise.} \end{cases}$$

5. If $I \in NA^n$, then $\tilde{u}_I^{n-1} \leq t_n$.

6. If $I \in \mathcal{U}^{n-1}(K) \cap \mathcal{U}^n(K)$, then

$$u_{I \rightarrow K}^n = u_{I \rightarrow K}^{n-1}.$$

7. If $I \in \mathcal{U}^n(K) \setminus \mathcal{U}^{n-1}(K)$, then

$$u_{I \rightarrow K}^n = t_n.$$

8. If $t_n > t_{n-1}$, then $\tilde{t}_n \geq t_n$.

9. If $t_n = t_{n-1}$, then $\tilde{t}_n \leq t_n$.

Proof. 1. This follows directly from point 6 of the algorithm.

2. This is a straight forward consequence of point 10 of the algorithm.

3. By contradiction, we assume that there exists I and $K \in V(I)$ such that $I \in NB^n \cap \mathcal{U}^n(K)$. The fact that $I \in NB^n$ implies that

$$\theta_I^n \hat{c}_I^n < 0.$$

Furthermore we deduce from $I \in \mathcal{U}^n(K)$ that $K \in NB^n$ and therefore

$$\theta_I^n = -\theta_K^n \text{ and } \theta_I^n \hat{c}_K^n > 0.$$

Taking both formulae together we get

$$\hat{c}_I^n \hat{c}_K^n < 0,$$

which is a contradiction to the definition of \hat{c} , because $\hat{c}_I \hat{c}_K \geq 0$ for adjacent nodes I and K .

4. Assume that $I \in NA^n$. Then we know $I \in NB^{n-1}$ and due to property 3 we get $I \notin \mathcal{U}^{n-1}(K)$. Hence we get $u_{I \rightarrow K}^{n-1} = +\infty$. Due to step 10 of mGFT we then have $u_{I \rightarrow K}^n = t_n$ in the case $I \in \mathcal{U}^n(K)$ and $+\infty$ otherwise.

5. We prove by contradiction. We assume that $\tilde{u}^{n-1} > t_n$. Since $I \in NA^n$, we get that

$$\tilde{u}_I^{n-1} = \tilde{t}_n > t_n.$$

Step 6 of mGFT gives us that $t_n = t_{n-1} + \Delta t < \tilde{t}_n$ and so by step 7, no point would be accepted. Thus we have a contradiction to $I \in NA^n$.

6. Assume that $I \in \mathcal{U}^{n-1}(K) \cap \mathcal{U}^n(K)$. Due to step 10 of mGFT and property 2 we see

$$u_{I \rightarrow K}^n = \min(u_{I \rightarrow K}^{n-1}, t_n) = u_{I \rightarrow K}^{n-1}.$$

7. Assume $I \in \mathcal{U}^n(K) \setminus \mathcal{U}^{n-1}(K)$. Due to $I \notin \mathcal{U}^{n-1}(K)$ we know that $u_{I \rightarrow K}^{n-1} = +\infty$. Thus step 10 of mGFT gives us

$$u_{I \rightarrow K}^n = \min(u_{I \rightarrow K}^{n-1}, t_n) = t_n.$$

8. If $t_n > t_{n-1}$, then we see with step 6 of mGFT that

$$t_n = \min(\tilde{t}_n, t_{n-1} + \Delta t) \leq \tilde{t}_n.$$

9. If $t_n = t_{n-1}$, then we see with step 6 of mGFT that

$$t_{n-1} \geq \min(\tilde{t}_n, t_{n-1} + \Delta t)$$

and thus we gain $t_n = t_{n-1} \geq \tilde{t}_n$. □

For the properties 2, 4, 6 and 7 we use step 10 of the mGFT. These properties hold only if we use the right formulation of step 7 of the mGFT and not the one used in the original publication of the mGFMM [Forog].

PROPOSITION 2.4.7 (Monotonicity of $c \rightarrow \hat{c}$):

The mapping $c \rightarrow \hat{c}$ is monotone, that means if we have $c_v \geq c_u$ then we get $\hat{c}_v \geq \hat{c}_u$.

Proof. Let $c_v \geq c_u$. We will prove that we have for all nodes $I \in \mathcal{N}$

$$\hat{c}_{v,I} \geq \hat{c}_{u,I}. \tag{2.4.1}$$

To proof this, we distinguish four cases:

Case 1: $\hat{c}_{v,I} = c_{v,I}$ and $\hat{c}_{u,I} = c_{u,I}$.

This is the simplest case. We just get

$$\hat{c}_{v,I} = c_{v,I} \geq c_{u,I} = \hat{c}_{u,I}$$

and therefore (2.4.1) is true.

Case 2: $\hat{c}_{v,I} \neq c_{v,I}$ and $\hat{c}_{u,I} \neq c_{u,I}$.

In this case we have $\hat{c}_{v,I} = 0 = \hat{c}_{u,I}$ due to the definition 2.3.2 for \hat{c} and so (2.4.1) is true.

Case 3: $\hat{c}_{v,I} = c_{v,I}$ and $\hat{c}_{u,I} \neq c_{u,I}$.

We have $\hat{c}_{u,I} = 0$ like above. We assume by contradiction that

$$\hat{c}_{v,I} = c_{v,I} < 0 = \hat{c}_{u,I}. \tag{2.4.2}$$

Due to $c_v \geq c_u$ we get that $c_{u,I} < 0$. Since $\hat{c}_{u,I} \neq c_{u,I}$, we know that there exists $K \in V(I)$ with $c_{u,K} > 0$ and $|c_{u,I}| \leq |c_{u,K}|$. Therefore we get

$$|c_{v,I}| \leq |c_{u,I}| \leq |c_{u,K}| = c_{u,K} \leq c_{v,K}$$

and by the definition of \hat{c} we get $\hat{c}_{v,I} = 0$. This contradicts (2.4.2) and proves therefore (2.4.1).

2 Generalized fast marching method

Case 4: $\widehat{c}_{v,I} \neq c_{v,I}$ and $\widehat{c}_{u,I} = c_{u,I}$.

We use the same idea as in case 3. Now we have $\widehat{c}_{v,I} = 0$. By contradiction, assume that

$$\widehat{c}_{u,I} = c_{u,I} > 0 = \widehat{c}_{v,I}. \quad (2.4.3)$$

Since $c_v \geq c_u$, we know that $c_{v,I} > 0$. Since $\widehat{c}_{v,I} \neq c_{v,I}$, we deduce that there exists $K \in V(I)$ such that $c_{v,K} < 0$ and $|c_{v,I}| \leq |c_{v,K}|$. Therefore we get

$$|c_{u,I}| \leq |c_{v,I}| \leq |c_{v,K}| = -c_{v,K} \leq -c_{u,K} = |c_{u,K}|$$

and by the definition of \widehat{c} we get $\widehat{c}_{u,I} = 0$. This contradicts (2.4.3) and proves therefore (2.4.1). □

PROPOSITION 2.4.8 (Symmetry of the algorithm):

Let us consider the two velocities c_u and c_v . We denote by $(t_n^u)_n, (u_{I \rightarrow K}^n)_n, (\theta_u^n)_n$ (resp. $(s_m^v)_m, (v_{I \rightarrow K}^m)_m, (\theta_v^m)_m$) the sequences of times, of useful times and of field values associated with the velocity c_u (resp. c_v) and with the initial condition θ_u^0 (resp. θ_v^0).

We also define $(s_m^{\bar{u}})_m, (\bar{u}_{I \rightarrow K}^m)_m, (\theta_{\bar{u}}^m)_m$ (resp. $(t_n^{\bar{v}})_n, (\bar{v}_{I \rightarrow K}^n)_n, (\theta_{\bar{v}}^n)_n$) the sequences of times, of useful times and of field values associated to the velocity $c_{\bar{u}} = -c_v$ (resp. $c_{\bar{v}} = -c_u$) and with the initial condition $\theta_{\bar{u}}^0 = -\theta_v^0$ (resp. $\theta_{\bar{v}}^0 = -\theta_u^0$).

Then we have the following equivalence:

$$\theta_v^m \succeq \theta_u^n \Leftrightarrow \theta_{\bar{v}}^n \succeq \theta_{\bar{u}}^m.$$

Proof. Due to the construction of the algorithm we have symmetry in the sequences of times, the useful times and the field values associated to the velocity, i.e. it holds that

$$\theta_{\bar{u}}^m = -\theta_v^m, \quad \theta_{\bar{v}}^n = -\theta_u^n, \quad s_m^{\bar{u}} = s_m^v, \quad t_n^{\bar{v}} = t_n^u, \quad \bar{u}_{I \rightarrow K}^m = v_{I \rightarrow K}^m, \quad \bar{v}_{I \rightarrow K}^n = u_{I \rightarrow K}^n.$$

Now assume that $\theta_v^m \succeq \theta_u^n$ and let us prove that $\theta_{\bar{v}}^n \succeq \theta_{\bar{u}}^m$. We distinguish two cases:

Case 1: $\theta_v^m > \theta_u^n$

We get

$$\theta_{\bar{u}}^m = -\theta_v^m < -\theta_u^n = \theta_{\bar{v}}^n.$$

Case 2: $\theta_v^m = \theta_u^n = \sigma$

Here we have

$$\theta_{\bar{u}}^m = \theta_{\bar{v}}^n = \bar{\sigma} = -\sigma.$$

Let $I \in \mathcal{U}_{\bar{u}}^m(K) \cap \mathcal{U}_{\bar{v}}^n(K)$. Thus we know especially that $I \in \mathcal{U}_v^m(K) \cap \mathcal{U}_u^n(K)$.

Then we get

$$\bar{\sigma} \bar{v}_{I \rightarrow K}^n = -\sigma u_{I \rightarrow K}^n \leq -\sigma v_{I \rightarrow K}^m = \bar{\sigma} \bar{u}_{I \rightarrow K}^m.$$

In the last step we used that $\theta_v^m \succeq \theta_u^n$.

In total we get $\theta_{\bar{v}}^n \succeq \theta_{\bar{u}}^m$. □

2.4.3 Comparison principle for the t_n time

The main idea for the comparison principle is as follows. We take a point $K \in NB_u^n \cap NB_v^m$ at comparable times t_n and s_m , that means $C(t_n, s_m)$ is true. Then the points I with $I \in \mathcal{U}_u^n(K) \cap \mathcal{U}_v^m(K)$ become useful with velocity v before become useful with velocity u . With the definition of $u_{I \rightarrow K}$ and $v_{I \rightarrow K}$ we are able to show that $v_{I \rightarrow K}^m \leq u_{I \rightarrow K}^n$. Thus we can conclude with the monotonicity of the update scheme that $\tilde{v}_K^m \leq \tilde{u}_K^n$. Hence point K will be accepted with velocity v before with velocity u . This preserves the comparison of θ_u and θ_v .

The introduction of useful times is essential for the algorithm because with this notion and the right handling in step 10 we know that $v_{I \rightarrow K}^m \leq u_{I \rightarrow K}^n$ if a node I becomes useful for v before it becomes useful for u . Thus it is necessary to introduce a separate time for each neighbour of K and also to set this time only to a finite value if K is in upwind direction of I .

THEOREM 2.4.9 (Comparison principle for the t_n times):

Under the assumptions of theorem 2.4.5, we denote by $(t_n)_n$ and $(u_{I \rightarrow K}^n)_n$ (resp. $(s_m)_m$ and $(v_{I \rightarrow K}^m)_m$) the sequences of times and of useful times associated to the velocity c_u (resp. c_v). Then $\theta_v^m \succeq \theta_u^n$ for every time satisfying $C(t_n, s_m)$ and $t_n, s_m \leq T$.

Proof. We will do this proof by contradiction. Let us define

$$s^* = \inf\{\min(t_n, s_m) \text{ such that } \theta_v^m \not\succeq \theta_u^n, t_n, s_m \leq T \text{ and } C(t_n, s_m) \text{ is true}\}.$$

Due to the symmetry of the mGFT (proposition 2.4.8), we can assume that $s^* = s_{m^*}$ for a certain index m^* . Now we define

$$n^* = \inf\{n \geq 0 \text{ such that } \theta_v^{m^*} \not\succeq \theta_u^n, t_n \leq T \text{ and } C(t_n, s_{m^*}) \text{ is true}\}.$$

Thus we have (t_{n^*}, s_{m^*}) as the minimal couple with $\theta_v^{m^*} \succeq \theta_u^{n^*}$, $s_{m^*} \leq t_{n^*} < s_{m^*+1}$ and $t_{n^*} < t_{n^*+1}$. We write in the following n for n^* and m for m^* to simplify the notation.

We distinguish now two cases:

Case 1: $s_m < t_n$

We define $p \geq 0$ such that

$$t_{n-p-1} < t_{n-p} = \dots = t_n < t_{n+1}.$$

Step 1: $\theta_v^m \succeq \theta_u^{n-p-1}$

We will prove in this step that $\theta_v^m \succeq \theta_u^{n-p-1}$. The couple (n, m) is minimal one such that $\theta_v^m \not\succeq \theta_u^n$. Thus it suffices to show that $C(t_{n-p-1}, s_m)$ is satisfied. We recall that

$$s_m < t_{n-p} = \dots = t_n < s_{m+1}, \quad \text{and} \quad t_{n-p-1} < t_{n-p}.$$

Either we have $s_m \leq t_{n-p-1}$, then it holds $s_m \leq t_{n-p-1} < s_{m+1}$ and therefore $t_{n-p-1} < t_{n-p}$, or we have $t_{n-p-1} < s_m$, then we get $t_{n-p-1} < s_m < t_{n-p}$ which yields $s_m < s_{m+1}$.

Therefore $C(t_{n-p-1}, s_m)$ is true and thus it holds $\theta_v^m \succeq \theta_u^{n-p-1}$.

2 Generalized fast marching method

Step 2: Contradiction

We use lemma 2.4.10 to conclude that $\theta_v^m \succeq \theta_u^{n-p}, \dots, \theta_v^m \succeq \theta_u^n$. This is contradiction to the construction of n .

Case 2: $s_m = t_n$

We define analogously to case 1 now p and q such that

$$t_{n-p-1} < t_{n-p} = \dots = t_n < t_{n+1} \quad \text{and} \quad s_{m-q-1} < s_{m-q} = \dots = s_m < s_{m+1}.$$

Like above we have either $t_{n-p-1} \leq s_{m-q-1} < s_{m-q} = t_{n-p}$ or $s_{m-q-1} \leq t_{n-p-1} < t_{n-p} = s_{m-q}$. Thus $C(t_{n-p-1}, s_{m-q-1})$ is true. Due to the minimality of (m, n) we gain that $\theta_v^{m-q-1} \succeq \theta_u^{n-p-1}$. Now we use lemma 2.4.11 to deduce that $\theta_v^{m-q} \succeq \theta_u^{n-p}$. Proposition 2.4.12 gives us that $\theta_v^m \succeq \theta_u^n$, which is a contradiction to the definition of n . □

We used in the proof of theorem 2.4.9 the lemmata 2.4.10 and 2.4.11 as well as the proposition 2.4.12. We state these results and give also their proofs. The proof of proposition 2.4.12 is given in subsection 2.4.4.

LEMMA 2.4.10 (Two jumps, two arrivals):

Assume that $\theta_v^{m-1} \succeq \theta_u^{n-1}$. Then, if $s_{m-1} \leq t_n < s_m$ (resp. $t_{n-1} \leq s_m < t_n$) and $t_n, s_m \leq T$, we have $\theta_v^{m-1} \succeq \theta_u^n$ (resp. $\theta_v^m \succeq \theta_u^{n-1}$).

Proof. We show here the proof in the case $s_m > t_n$. The other case is equivalent due to the symmetry of the algorithm, see also proposition 2.4.8.

We do the proof by contradiction. Let us assume that $\theta_v^{m-1} \not\succeq \theta_u^n$. Thus there exists a point I with

$$\begin{cases} \theta_{v,I}^{m-1} < \theta_{u,I}^n \\ \text{or} \\ \theta_{v,I}^{m-1} = \theta_{u,I}^n = \sigma_I \text{ and } \exists K : I \in \mathcal{U}_v^{m-1}(K) \cap \mathcal{U}_u^n(K) \text{ with } \sigma_I u_{I \rightarrow K}^n < \sigma_I v_{I \rightarrow K}^{m-1}. \end{cases} \quad (2.4.4)$$

We distinguish two cases: in the first case the value of θ_u^n changes but not in the second case.

Case 1: $I \in NA_u^n$

Subcase 1.1: $\theta_{v,I}^{m-1} > \theta_{u,I}^{n-1}$

We know $\theta_{v,I}^{m-1} = 1$ and $\theta_{u,I}^{n-1} = -1$. Since $I \in NA_u^n$, we deduce that $\theta_{u,I}^n = 1 = \theta_{v,I}^{m-1} = \sigma_I$. Due to $I \in \mathcal{U}_v^{m-1}(K)$ and point 2 of proposition 2.4.6 we gain

$$v_{I \rightarrow K}^{m-1} \leq s_{m-1}.$$

Further we know by $I \in NA_u^n \cap \mathcal{U}_u^n(K)$ and point 4 of proposition 2.4.6 that

$$t_n = u_{I \rightarrow K}^n.$$

Together with $s_{m-1} \leq t_n$ from the assumption this yields

$$\tilde{v}_{I \rightarrow K}^{m-1} \leq s_{m-1} \leq t_n = u_{I \rightarrow K}^n.$$

which is a contradiction to (2.4.4).

Subcase 1.2: $\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1}$

We distinguish now the cases ± 1 .

Subcase 1.2.1: $\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1} = 1$

Since $I \in NA_u^n$ we know that $\theta_{u,I}^n = -1$, but this contradicts (2.4.4).

Subcase 1.2.2: $\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1} = -1$

Step 1: $\tilde{u}_I^{n-1} \geq \tilde{v}_I^{m-1}$ and $I \in NB_{v,-}^{m-1}$

Since $I \in NA_u^n$ we know that $I \in NB_{u,-}^{n-1}$. Due to $s_m - s_{m-1} \leq \Delta t$, $t_n - t_{n-1} \leq \Delta t$ and $s_m > t_n$ we have $|t_{n-1} - s_{m-1}| \leq \Delta t$. Further we recall that $\theta_v^{m-1} \succeq \theta_u^{n-1}$ and $\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1} = -1$. Thus the assumptions of lemma 2.4.13 are fulfilled and we conclude

$$\tilde{u}_I^{n-1} \geq \tilde{v}_I^{m-1} \text{ and } I \in NB_{v,-}^{m-1}.$$

Step 2: $\tilde{v}_I^{m-1} \geq s_m$

Due to step 6 of the mGFT we know $s_m = \max(s_{m-1}, \min(s_{m-1} + \Delta t, \tilde{s}_m))$. By the assumption of the lemma we have $s_m > s_{m-1}$ and so

$$s_m = \min(s_{m-1} + \Delta t, \tilde{s}_m) \leq \tilde{s}_m.$$

Using the definition of \tilde{s}_m , we conclude $\tilde{s}_m \leq \tilde{v}_I^{m-1}$ with $I \in NB_{v,-}^{m-1}$. Thus we know

$$\tilde{v}_I^{m-1} \geq s_m.$$

Step 3: Contradiction

Via step 1 and 2 we get

$$\tilde{u}_I^{n-1} \geq \tilde{v}_I^{m-1} \geq s_m > t_n$$

and then

$$\tilde{u}_I^{n-1} > t_n.$$

This is impossible because $I \in NA_u^n$, see also point 5 in proposition 2.4.6.

Subcase 1.3: $\theta_{v,I}^{m-1} < \theta_{u,I}^{n-1}$

This case is impossible since we assume $\theta_v^{m-1} \succeq \theta_u^{n-1}$.

Case 2: $I \notin NA_u^n$

2 Generalized fast marching method

Subcase 2.1: $\theta_{v,I}^{m-1} < \theta_{u,I}^n$

Since $I \notin NA_u^n$, we have $\theta_{u,I}^{n-1} = \theta_{u,I}^n$ and therefore $\theta_{v,I}^{m-1} < \theta_{u,I}^{n-1}$. This is a contradiction to $\theta_v^{m-1} \succeq \theta_u^{n-1}$.

Subcase 2.2: $\theta_{v,I}^{m-1} = \theta_{u,I}^n$

Due to $I \notin NA_u^n$, we have

$$\theta_{u,I}^{n-1} = \theta_{u,I}^n = \theta_{v,I}^{m-1} = \sigma_I = \pm 1. \quad (2.4.5)$$

From equation (2.4.4) we get that there exists a node $K \in \mathcal{E}$ such that

$$I \in \mathcal{U}_v^{m-1}(K) \cap \mathcal{U}_u^n(K) \text{ and } \sigma_I u_{I \rightarrow K}^n < \sigma_I v_{I \rightarrow K}^{m-1}. \quad (2.4.6)$$

We distinguish three subcases:

Subcase 2.2.1: $I \in \mathcal{U}_u^{n-1}(K)$

Equation (2.4.6) and the assumption $I \in \mathcal{U}_u^{n-1}(K)$ give us that $I \in \mathcal{U}_v^{m-1}(K) \cap \mathcal{U}_u^{n-1}(K)$. Together with the assumption $\theta_v^{m-1} \succeq \theta_u^{n-1}$ and (2.4.5) we conclude

$$\sigma_I u_{I \rightarrow K}^{n-1} \geq \sigma_I v_{I \rightarrow K}^{m-1}.$$

Furthermore it holds that $SI \in \mathcal{U}_u^{n-1}(K) \cap \mathcal{U}_u^n(K)$. Point 6 in proposition 2.4.6 gives us that $u_{I \rightarrow K}^n = u_{I \rightarrow K}^{n-1}$. Thus we have

$$\sigma_I u_{I \rightarrow K}^n \geq \sigma_I v_{I \rightarrow K}^{m-1}$$

which contradicts (2.4.6).

Subcase 2.2.2: $I \notin \mathcal{U}_u^{n-1}(K)$ and $\sigma_I = +1$

We have $I \in \mathcal{U}_u^n(K) \setminus \mathcal{U}_u^{n-1}(K)$. Using point 7 of proposition 2.4.6 we conclude that $u_{I \rightarrow K}^n = t_n$. In addition we know $I \in \mathcal{U}_v^{m-1}(K)$ and point 2 of proposition 2.4.6 leads to $s_{m-1} \leq v_{I \rightarrow K}^{m-1}$. In combination with the assumption of the lemma we get

$$u_{I \rightarrow K}^n = t_n \geq s_{m-1} \geq v_{I \rightarrow K}^{m-1}.$$

This is a contradiction to (2.4.6).

Subcase 2.2.3: $I \notin \mathcal{U}_u^{n-1}(K)$ and $\sigma_I = -1$

The equations (2.4.5) and (2.4.6) give us $I \in \mathcal{U}_v^{m-1}(K) \cap \mathcal{U}_u^n(K)$ and $\theta_{u,I}^{n-1} = \theta_{u,I}^n = \theta_{v,I}^{m-1} = -1$. Furthermore we have we can see like in subcase 1.2.2 that $|t_{n-1} - s_{m-1}| \leq \Delta t$. The assumptions of lemma 2.4.14 are fulfilled and thus we get $I \in \mathcal{U}_u^{n-1}(K)$, which leads to a contradiction.

Subcase 2.3: $\theta_{v,I}^{m-1} > \theta_{u,I}^n$

This case is impossible. See also equation (2.4.4).

□

LEMMA 2.4.11 (Two jumps, one arrival):

Assume that $\theta_v^{m-1} \succeq \theta_u^{n-1}$, $t_{n-1} < t_n$ and $s_{m-1} < s_m$. If $s_m = t_n \leq T$, then $\theta_v^m \succeq \theta_u^n$.

Proof. The proof of this lemma uses the same techniques as the proof of lemma 2.4.10. It is just more lengthy because we have to distinguish more cases.

We do the proof by contradiction. Let us assume that $\theta_v^{m-1} \not\geq \theta_u^n$. Thus there exists a point I with

$$\begin{cases} \theta_{v,I}^m < \theta_{u,I}^n \\ \text{or} \\ \theta_{v,I}^m = \theta_{u,I}^n = \sigma_I \text{ and } \exists K : I \in \mathcal{U}_v^m(K) \cap \mathcal{U}_u^n(K) \text{ and } \sigma_I u_{I \rightarrow K}^n < \sigma_I v_{I \rightarrow K}^m. \end{cases} \quad (2.4.7)$$

We consider the following four cases:

Case 1: $I \in NA_u^n \setminus NA_v^m$

Subcase 1.1: $\theta_{v,I}^{m-1} > \theta_{u,I}^{n-1}$

We know $\theta_{v,I}^{m-1} = 1$ and $\theta_{u,I}^{n-1} = -1$. Since $I \in NA_u^n \setminus NA_v^m$, we deduce that $\theta_{u,I}^n = 1 = \theta_{v,I}^m = \sigma_I$. Due to $I \in \mathcal{U}_v^m(K)$ and point 2 of proposition 2.4.6 we gain

$$v_{I \rightarrow K}^m \leq s_m.$$

Further we know by $I \in NA_u^n \cap \mathcal{U}_u^n(K)$ and point 4 of proposition 2.4.6 that

$$t_n = u_{I \rightarrow K}^n.$$

Together with $s_m = t_n$ from the assumption this yields

$$v_{I \rightarrow K}^{m-1} \leq s_{m-1} = t_n = u_{I \rightarrow K}^n.$$

which is a contradiction to (2.4.7).

Subcase 1.2: $\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1}$

We distinguish now the cases ± 1 .

Subcase 1.2.1: $\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1} = 1$

Due to $I \in NA_u^n$ we know that $\theta_{u,I}^n = -1$, but this contradicts (2.4.7).

Subcase 1.2.2: $\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1} = -1$

Step 1: $\tilde{u}_I^{n-1} \geq \tilde{v}_I^{m-1}$ and $I \in NB_{v,-}^{m-1}$

Since $I \in NA_u^n$ we know that $I \in NB_{u,-}^{n-1}$. We know $t_n - t_{n-1} \leq \Delta t$, $s_m - s_{m-1} \leq \Delta t$ and $s_m = t_n$. Therefore we get $|t_{n-1} - s_{m-1}| \leq \Delta t$. Further we recall that $\theta_v^{m-1} \geq \theta_u^{n-1}$ and $\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1} = -1$. Thus the assumptions of lemma 2.4.13 are fulfilled and we conclude

$$\tilde{u}_I^{n-1} \geq \tilde{v}_I^{m-1} \text{ and } I \in NB_{v,-}^{m-1}.$$

Step 2: $\tilde{v}_I^{m-1} \geq s_m$

Due to step 6 of the mGFT we know $s_m = \max(s_{m-1}, \min(s_{m-1} +$

2 Generalized fast marching method

$\Delta t, \tilde{s}_m$). By the assumption of the lemma we have $s_m > s_{m-1}$ and so

$$s_m = \min(s_{m-1} + \Delta t, \tilde{s}_m).$$

Using the definition of \tilde{s}_m , we conclude with $I \in NB_{v,-}^{m-1}$ that $\tilde{s}_m \leq \tilde{v}_I^{m-1}$.

Either we have $\tilde{v}_I^{m-1} \geq \tilde{s}_m$ and therefore

$$s_m \leq \tilde{s}_m < v_I^{m-1},$$

or we have $\tilde{v}_I^{m-1} \geq \tilde{s}_m$, but then we know with $J \notin NA_v^m$ and steps 7 and 8 of the mGFT that

$$s_m = s_{m-1} + \Delta t < \tilde{s}_m < v_I^{m-1}.$$

Thus we know

$$\tilde{v}_I^{m-1} \geq s_m.$$

Step 3: Contradiction

Via step 1 and 2 we get

$$\tilde{u}_I^{n-1} \geq \tilde{v}_I^{m-1} \geq s_m = t_n$$

and then

$$\tilde{u}_I^{n-1} > t_n.$$

This is impossible because $I \in NA_u^n$, see also point 5 in proposition 2.4.6.

Subcase 1.3: $\theta_{v,I}^{m-1} < \theta_{u,I}^{n-1}$

This case is impossible since we assume $\theta_v^{m-1} \succeq \theta_u^{n-1}$.

Case 2: $I \in NA_v^m \setminus NA_u^n$

Due to the symmetry of the mGFT in proposition 2.4.8, this case is done like case 1.

Case 3: $I \in NA_u^n \cap NA_v^m$

Subcase 3.1: $\theta_{v,I}^{m-1} > \theta_{u,I}^{n-1}$

We know $\theta_{v,I}^{m-1} = 1$ and $\theta_{u,I}^{n-1} = -1$. Due to $I \in NA_u^n \cap NA_v^m$ we conclude that $I \in NB_{u,-}^{n-1} \cap NB_{v,+}^{m-1}$. Therefore we have

$$\hat{c}_{u,I}^{n-1} > 0 \text{ and } \hat{c}_{v,I}^{m-1} < 0.$$

This is a contradiction because $|s_{m-1} - t_{n-1}| \leq \Delta t$ and thus $c_{v,I}^{m-1} \geq c_{u,I}^{n-1}$.

Subcase 3.2: $\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1}$

Since $I \in NA_u^n \cap NA_v^m$ we have $\theta_{v,I}^m = \theta_{u,I}^n$. From equation (2.4.7) we know $I \in \mathcal{U}_u^n(K) \cap \mathcal{U}_v^m(K)$. With point 4 of proposition 2.4.6 we gain

$$v_{I \rightarrow K}^m = s_m = t_n = u_{I \rightarrow K}^n.$$

This is a contradiction to (2.4.7).

Subcase 3.3: $\theta_{v,I}^{m-1} < \theta_{u,I}^{n-1}$

This is impossible since $\theta_v^{m-1} \succeq \theta_u^{n-1}$.

Case 4: $I \notin NA_u^n \cup NA_v^m$

Subcase 4.1: $\theta_{v,I}^m < \theta_{u,I}^n$

Due to $I \notin NA_u^n \cup NA_v^m$ we have $\theta_{u,I}^{n-1} = \theta_{u,I}^n$ and $\theta_{v,I}^m = \theta_{v,I}^{m-1}$. Thus we get $\theta_{v,I}^{m-1} < \theta_{u,I}^{n-1}$. This is a contradiction to $\theta_{v,I}^{m-1} \succeq \theta_{u,I}^{n-1}$.

Subcase 4.2: $\theta_{v,I}^m = \theta_{u,I}^n$

Since $I \notin NA_u^n \cup NA_v^m$, we have

$$\theta_{u,I}^{n-1} = \theta_{u,I}^n = \theta_{v,I}^{m-1} = \theta_{v,I}^m = \sigma_I = \pm 1. \quad (2.4.8)$$

Due to equation (2.4.7) we know that there exists a node K such that $I \in \mathcal{U}_v^m(K) \cap \mathcal{U}_u^n(K)$ and

$$\sigma_I u_{I \rightarrow K}^n < \sigma_I v_{I \rightarrow K}^m. \quad (2.4.9)$$

We have to distinguish subcases according to $\mathcal{U}(K)$:

Subcase 4.2.1: $I \in \mathcal{U}_u^{n-1}(K) \cap \mathcal{U}_v^{m-1}(K)$

Using (2.4.8) and the assumption $\theta_v^{m-1} \succeq \theta_u^{n-1}$ of the lemma we conclude

$$\sigma_I u_{I \rightarrow K}^{n-1} \geq \sigma_I v_{I \rightarrow K}^{m-1}.$$

Furthermore we have $I \in \mathcal{U}_u^{n-1}(K) \cap \mathcal{U}_u^n(K)$ and $I \in \mathcal{U}_v^{m-1}(K) \cap \mathcal{U}_v^m(K)$. Thus we get with point 6 of proposition 2.4.6 that $u_{I \rightarrow K}^n = u_{I \rightarrow K}^{n-1}$ and $v_{I \rightarrow K}^m = v_{I \rightarrow K}^{m-1}$. Therefore we have

$$\sigma_I u_{I \rightarrow K}^n \geq \sigma_I v_{I \rightarrow K}^m,$$

which contradicts (2.4.9).

Subcase 4.2.2: $I \in \mathcal{U}_v^{m-1}(K) \setminus \mathcal{U}_u^{n-1}(K)$

Subcase 4.2.2.1: $\sigma_I = +1$

We use $I \in \mathcal{U}_u^n(K) \setminus \mathcal{U}_u^{n-1}(K)$ together with point 7 of proposition 2.4.6 which give us $u_{I \rightarrow K}^n = t_n$. Furthermore we know that $I \in \mathcal{U}_v^m(K)$. Point 2 of proposition 2.4.6 gives us $s_m \geq v_{I \rightarrow K}^m$. Thus we see

$$u_{I \rightarrow K}^n = t_n = s_m \geq v_{I \rightarrow K}^m,$$

which contradicts (2.4.9).

Subcase 4.2.2.2: $\sigma_I = -1$

We know $I \in \mathcal{U}_v^{m-1}(K) \cap \mathcal{U}_u^n(K)$ and furthermore $\theta_{u,I}^{n-1} = \theta_{u,I}^n = \theta_{v,I}^{m-1} = -1$. Like in Subcase 1.2.2 we can see that $|t_{n-1} - s_{m-1}| \leq \Delta t$. The assumption of lemma 2.4.14 are satisfied and we get that $I \in \mathcal{U}_u^{n-1}(K)$. This is a contradiction.

2 Generalized fast marching method

Subcase 4.2.3: $I \in \mathcal{U}_u^{n-1}(K) \setminus \mathcal{U}_v^{m-1}(K)$

Due to the symmetry this case is done in the same way as the preceding subcase 4.2.2.

Subcase 4.2.4: $I \notin \mathcal{U}_u^{n-1}(K) \cup \mathcal{U}_v^{m-1}(K)$

We have $I \in \mathcal{U}_u^n(K) \setminus \mathcal{U}_u^{n-1}(K)$ and $I \in \mathcal{U}_v^m(K) \setminus \mathcal{U}_v^{m-1}(K)$. Via point 7 of proposition 2.4.6 we get $t_n = u_{I \rightarrow K}^n$ and $s_m = v_{I \rightarrow K}^m$. Thus we have

$$v_{I \rightarrow K}^m = s_m = t_n = u_{I \rightarrow K}^n.$$

This is a contradiction to (2.4.9).

Subcase 4.3: $\theta_{v,I}^m > \theta_{u,I}^n$

This case is impossible due to the assumption in equation (2.4.7). □

PROPOSITION 2.4.12 (Stationary case with the same arrival):

Let us assume that $t_{n-p-1} < t_{n-p} = \dots = t_n < t_{n+1}$ and $s_{m-q-1} < s_{m-q} = \dots = s_m < s_{m+1}$ with $p, q \geq 0$ and $t_n = s_m \leq T$. If $\theta_v^{m-q} \succeq \theta_u^{n-p}$, then $\theta_v^m \succeq \theta_u^n$.

Subsection 2.4.4 is devoted to the proof of the preceding proposition, but before we state this proof we will turn our attention to the lemmata 2.4.13 and 2.4.14.

The following lemma handles with the monotonicity of the update scheme and transfers this property to the discrete evolution. Thus we can with this lemma compare θ_u^{n-1} and θ_v^{m-1} by comparing the times \tilde{u}^{n-1} and \tilde{v}^{m-1} at the narrow-band. Therefore it can be seen as the heart of the proof of theorem 2.4.5.

LEMMA 2.4.13 (Comparison of the candidate time \tilde{u}):

Assume that $\theta_v^{m-1} \succeq \theta_u^{n-1}$, $|t_{n-1} - s_{m-1}| \leq \Delta t$, with $t_{n-1}, s_{m-1} \leq T$. If

$$\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1} = -1 \quad \text{and} \quad I \in NB_{u,-}^{n-1}$$

$$\left(\text{resp.} \quad \theta_{v,I}^{m-1} = \theta_{u,I}^{n-1} = 1 \quad \text{and} \quad I \in NB_{v,+}^{m-1} \right)$$

then

$$I \in NB_{v,-}^{m-1} \quad \text{and} \quad \tilde{u}_I^{n-1} \geq \tilde{v}_I^{m-1}$$

$$\left(\text{resp.} \quad I \in NB_{u,+}^{n-1} \quad \text{and} \quad \tilde{u}_I^{n-1} \leq \tilde{v}_I^{m-1} \right).$$

Proof. First, we prove this lemma in the case $\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1} = -1$ and $I \in NB_{u,-}^{n-1}$. The other case will be handled afterwards.

Due to $I \in NB_{u,-}^{n-1}$, we know that

$$\begin{cases} \exists K \in V(I) \text{ such that } \theta_{u,K}^{n-1} = 1, \\ \tilde{c}_{u,I}^{n-1} > 0. \end{cases} \quad (2.4.10)$$

Step 1: $I \in NB_{v,-}^{m-1}$

Using the assumption $\theta_v^{m-1} \succeq \theta_u^{n-1}$, we know that $\theta_{v,K}^{m-1} \geq \theta_{u,K}^{n-1} = 1$ and thus we have $\theta_{v,K}^{m-1} = 1$. Hence we conclude with $\theta_{v,I}^{m-1} = -1$ that

$$I \in F_{v,-}^{m-1}. \quad (2.4.11)$$

Furthermore due to $|t_{n-1} - s_{m-1}| \leq \Delta t$ it holds

$$c_v(\cdot, s_{m-1}) \geq c_u(\cdot, t_{n-1}).$$

The monotonicity of $c \rightarrow \hat{c}$ in proposition 2.4.7 together with equation (2.4.10) give us

$$\hat{c}_{v,I}^{m-1} \geq \hat{c}_{u,I}^{n-1} > 0. \quad (2.4.12)$$

Now we see that due to equations (2.4.11) and (2.4.12) that

$$I \in NB_{v,-}^{m-1}. \quad (2.4.13)$$

Step 2: Ordering $u_{\cdot \rightarrow I}^{n-1} \geq v_{\cdot \rightarrow I}^{m-1}$

Let $K \in V(I)$. We distinguish two cases:

Case 1: $\theta_{v,K}^{m-1} = -1$

Since $\theta_v^{m-1} \succeq \theta_u^{n-1}$, we get that $\theta_{v,K}^{m-1} \geq \theta_{u,K}^{n-1}$ and therefore $\theta_{u,K}^{n-1} = -1$. Thus K is not useful for I and therefore we have

$$u_{K \rightarrow I}^{n-1} = +\infty = v_{K \rightarrow I}^{m-1}.$$

Case 2: $\theta_{v,K}^{m-1} = 1$

First we assume $\theta_{u,K}^{n-1} = 1$. Together with $I \in NB_{u,-}^{n-1} \cap NB_{v,-}^{m-1}$ we know that $K \in \mathcal{U}_u^{n-1}(I) \cap \mathcal{U}_v^{m-1}(I)$. Then using the assumption $\theta_v^{m-1} \succeq \theta_u^{n-1}$ yields that $u_{K \rightarrow I}^{n-1} \geq v_{K \rightarrow I}^{m-1}$ with $\sigma_I = 1$.

In the other case we assume $\theta_{u,K}^{n-1} = -1$. Thus point K is not useful for I and we get that $u_{K \rightarrow I}^{n-1} = +\infty$ which leads to $u_{K \rightarrow I}^{n-1} \geq v_{K \rightarrow I}^{m-1}$.

In total this yields

$$u_{K \rightarrow I}^{n-1} \geq v_{K \rightarrow I}^{m-1}$$

for all $K \in V(I)$.

Step 3: $\tilde{u}_I^{n-1} \geq \tilde{v}_I^{m-1}$

In step 1 we showed in equation (2.4.12) that $\hat{c}_{v,I}^{m-1} \geq \hat{c}_{u,I}^{n-1} > 0$ and step 2 yield $u_{\cdot \rightarrow I}^{n-1} \geq v_{\cdot \rightarrow I}^{m-1}$. The claim is followed now by the monotonicity in the Hopf-Lax-Formula (2.1.6) which we use to compute the update in step 4 of the mGFT.

Now we will do the proof for the case $\theta_{v,I}^{m-1} = \theta_{u,I}^{n-1} = 1$ and $I \in NB_{v,+}^{m-1}$. This done by using the same methods as above.

Due to $I \in NB_{v,+}^{m-1}$, we know that

$$\begin{cases} \exists K \in V(I) \text{ such that } \theta_{v,K}^{m-1} = -1, \\ \hat{c}_{v,I}^{m-1} < 0. \end{cases} \quad (2.4.14)$$

2 Generalized fast marching method

Step 1: $I \in NB_{u,+}^{n-1}$

Using the assumption $\theta_v^{m-1} \succeq \theta_u^{n-1}$, we know that $-1 = \theta_{v,K}^{m-1} \geq \theta_{u,K}^{n-1}$ and thus we have $\theta_{u,K}^{n-1} = -1$. Hence we conclude with $\theta_{v,I}^{m-1} = 1$ that

$$I \in F_{u,+}^{n-1}. \quad (2.4.15)$$

Furthermore due to $|t_{n-1} - s_{m-1}| \leq \Delta t$ it holds that

$$c_v(\cdot, s_{m-1}) \geq c_u(\cdot, t_{n-1}).$$

The monotonicity of $c \rightarrow \hat{c}$ in proposition 2.4.7 together with equation (2.4.14) give us

$$0 > \hat{c}_{v,I}^{m-1} \geq \hat{c}_{u,I}^{n-1}. \quad (2.4.16)$$

Now we see that due to equations (2.4.15) and (2.4.16) that

$$I \in NB_{u,+}^{n-1}. \quad (2.4.17)$$

Step 2: Ordering $u_{\cdot \rightarrow I}^{n-1} \leq v_{\cdot \rightarrow I}^{m-1}$

Let $K \in V(I)$. We distinguish two cases:

Case 1: $\theta_{u,K}^{n-1} = +1$

Since $\theta_v^{m-1} \succeq \theta_u^{n-1}$, we get that $\theta_{v,K}^{m-1} \geq \theta_{u,K}^{n-1}$ and therefore $\theta_{v,K}^{m-1} = +1$. Thus K is not useful for I and therefore we have

$$u_{K \rightarrow I}^{n-1} = +\infty = v_{K \rightarrow I}^{m-1}.$$

Case 2: $\theta_{u,K}^{n-1} = -1$

First we assume $\theta_{v,K}^{m-1} = -1$. Together with $I \in NB_{u,+}^{n-1} \cap NB_{v,+}^{m-1}$ we know that $K \in \mathcal{U}_u^{n-1}(I) \cap \mathcal{U}_v^{m-1}(I)$. Then using the assumption $\theta_v^{m-1} \succeq \theta_u^{n-1}$ yields that $u_{K \rightarrow I}^{n-1} \leq v_{K \rightarrow I}^{m-1}$ with $\sigma_I = -1$.

In the other case we assume $\theta_{v,K}^{m-1} = +1$. Thus point K is not useful for I and we get that $v_{K \rightarrow I}^{m-1} = +\infty$ which leads to $u_{K \rightarrow I}^{n-1} \leq v_{K \rightarrow I}^{m-1}$.

Thus we have for all $K \in V(I)$

$$u_{K \rightarrow I}^{n-1} \leq v_{K \rightarrow I}^{m-1}.$$

Step 3: $\tilde{u}_I^{n-1} \leq \tilde{v}_I^{m-1}$

In step 1 we showed in equation (2.4.16) that $0 > \hat{c}_{v,I}^{m-1} \geq \hat{c}_{u,I}^{n-1}$ and in step 2 yields $u_{\cdot \rightarrow I}^{n-1} \leq v_{\cdot \rightarrow I}^{m-1}$. The claim is followed now by the monotonicity in the Hopf-Lax-Formula (2.1.6) which we use to compute the update in step 4 of the mGFT.

□

The lemma 2.4.14 give us an useful condition to decide whether a point as already been useful or not. In this lemma the ordering of the velocity is essential for the proof.

LEMMA 2.4.14 (Property of useful points):

Assume that $I \in \mathcal{U}_v^m(K) \cap \mathcal{U}_u^n(K)$, $\theta_{v,I}^m = \theta_{u,I}^n = \theta_{u,I}^{n-1} = -1$ and $|t_{n-1} - s_m| \leq \Delta t$ with $t_n, s_m \leq T$. Then it holds that $I \in \mathcal{U}_u^{n-1}(K)$.

Proof. We do the proof by contradiction. We assume that $I \notin \mathcal{U}_u^{n-1}(K)$.

Due to $I \in \mathcal{U}_v^m(K)$ and $\theta_{v,I}^m = -1$ we know that

$$\theta_{v,K}^m = 1 \text{ and } \widehat{c}_{v,K}^m < 0.$$

Likewise we have $I \in \mathcal{U}_u^n(K)$ and $\theta_{v,I}^n = -1$ and thus

$$\theta_{u,K}^n = 1 \text{ and } \widehat{c}_{u,K}^n < 0.$$

Due to the assumption $|t_{n-1} - s_m| \leq \Delta t$ and the ordering of the velocity c_u and c_v we gain

$$\widehat{c}_{u,K}^{n-1} \leq \widehat{c}_{v,K}^m < 0.$$

We know $\theta_{u,I}^{n-1} = -1$, $\widehat{c}_{u,K}^{n-1} < 0$ and $I \notin \mathcal{U}_u^{n-1}(K)$, thus we get

$$\theta_{u,I}^{n-1} = \theta_{u,K}^{n-1} = -1.$$

This is a contradiction, because we cannot get from $\theta_{u,K}^{n-1} = -1$ to $\theta_{u,K}^n = +1$ with a negative velocity $\widehat{c}_{u,K}^{n-1} < 0$. \square

2.4.4 Comparison principle for the \tilde{t}_n times: proof of proposition 2.4.12

In this subsection we prove proposition 2.4.12. To do so we state and prove a slightly more general result in proposition 2.4.15.

First we recall that we consider in this subsection two sequences of times with

$$t_{n-p-1} < t_{n-p} = \dots = t_n < t_{n+1} \text{ and } s_{m-q-1} < s_{m-q} = \dots = s_m < s_{m+1}$$

such that $t_n = s_m \leq T$. To simplify the notation, we set for this subsection

$$\tilde{t}_{n-p} = -\infty \text{ and } \tilde{s}_{m-q} = -\infty. \quad (2.4.18)$$

With this notational convention the definition of comparable times 2.4.2 for times \tilde{t}_{n-k} and \tilde{s}_{m-l} with $0 \leq k \leq p$ and $0 \leq l \leq q$ can be read as

$$C(\tilde{t}_{n-k}, \tilde{s}_{m-l}) \text{ is true if } \begin{cases} \tilde{t}_{n-k} \leq \tilde{s}_{m-l} < \tilde{t}_{n-k+1}, \\ \text{or} \\ \tilde{s}_{m-l} \leq \tilde{t}_{n-k} < \tilde{s}_{m-l+1}. \end{cases}$$

This is because the velocities c_u^{n-k} and c_v^{m-l} are independent of $k = 0, \dots, p$ and $l = 0, \dots, q$ respectively and the notational convention (2.4.18). Thus we have for $0 \leq k \leq p$ and $0 \leq l \leq q$ that

$$\tilde{t}_{n-k} < \tilde{t}_{n-k+1} \quad \text{ans} \quad \tilde{s}_{m-l} < \tilde{s}_{m-l+1}.$$

2 Generalized fast marching method

PROPOSITION 2.4.15 (Comparison principle for the \tilde{t}_n times):

Let $p, q \geq 0$. If it holds that $\theta_v^{m-q} \succeq \theta_u^{n-p}$ with $p + q \geq 1$, then we have $\theta_v^{m-l} \succeq \theta_u^{n-k}$ for $0 \leq k \leq p$, $0 \leq l \leq q$ satisfying $C(\tilde{t}_{n-k}, \tilde{s}_{m-l})$.

Proof. We do the proof by contradiction. Let us define

$$\tilde{s}^* = \inf \left\{ \min(\tilde{t}_{n-k}, \tilde{s}_{m-l}) \mid \theta_v^{m-l} \not\succeq \theta_u^{n-k}, 0 \leq k \leq p, 0 \leq l \leq q \text{ and } C(\tilde{t}_{n-k}, \tilde{s}_{m-l}) \text{ is true} \right\}.$$

Due to the symmetry of the mGFT in proposition 2.4.8, we can assume that $\tilde{s}^* = \tilde{s}_{m-l^*}$ for a certain index l^* . Now we define

$$k^* = \sup \left\{ 0 \leq k \leq p \text{ such that } \theta_v^{m-l^*} \not\succeq \theta_u^{n-k} \text{ and } C(\tilde{t}_{n-k}, \tilde{s}_{m-l^*}) \text{ is true} \right\}.$$

Thus $(n - k^*, m - l^*)$ is the minimal couple with $\theta_v^{m-l^*} \not\succeq \theta_u^{n-k^*}$ and

$$\tilde{s}_{m-l^*} \leq \tilde{t}_{n-k^*} < \tilde{s}_{m-l^*+1}.$$

Due to our convention $C(\tilde{t}_{n-p}, \tilde{s}_{m-q})$ is true. Therefore we know that $k^* < p$ or $l^* < q$. Together with $\tilde{s}_{m-l^*} \leq \tilde{t}_{n-k^*}$ we conclude $k^* < p$.

In the following we will write k instead of k^* and l instead of l^* to simplify the notation. We distinguish for the proof now two cases:

Case 1: $\tilde{s}_{m-l} < \tilde{t}_{n-k}$

Step 1: $\theta_v^{m-l} \succeq \theta_u^{n-k-1}$

Due to the minimality of the couple $(n - k, m - l)$ it suffices to show that $C(\tilde{t}_{n-k-1}, \tilde{s}_{m-l})$ is true. This implies then $\theta_v^{m-l} \succeq \theta_u^{n-k-1}$. The definition of k and l gives us

$$\tilde{s}_{m-l} < \tilde{t}_{n-k} < \tilde{s}_{m-l+1}.$$

Either we have $\tilde{s}_{m-l} \leq \tilde{t}_{n-k-1}$, which implies $\tilde{s}_{m-l} \leq \tilde{t}_{n-k-1} < \tilde{t}_{n-k} < \tilde{s}_{m-l+1}$, or we have $\tilde{t}_{n-k-1} < \tilde{s}_{m-l}$, which yields $\tilde{t}_{n-k-1} < \tilde{s}_{m-l} < \tilde{t}_{n-k}$.

Thus $C(\tilde{t}_{n-k-1}, \tilde{s}_{m-l})$ is true anyway and we get that $\theta_v^{m-l} \succeq \theta_u^{n-k-1}$.

Step 2: Contradiction

Now we can use lemma 2.4.17 to conclude that $\theta_v^{m-l} \succeq \theta_u^{n-k}$, but this leads to a contradiction of the construction of k and l .

Case 2: $\tilde{s}_{m-l} = \tilde{t}_{n-k}$

We have by construction that $k < p$. Thus we know that $\tilde{s}_{m-l} = \tilde{t}_{n-k} > -\infty$ and therefore is $l < q$. Furthermore we have $\tilde{t}_{n-k-1} \leq \tilde{s}_{m-l-1} < \tilde{s}_{m-l} = \tilde{t}_{n-k}$ or $\tilde{s}_{m-l-1} \leq \tilde{t}_{n-k-1} < \tilde{t}_{n-k} = \tilde{s}_{m-l}$. Thus $C(\tilde{t}_{n-k-1}, \tilde{s}_{m-l-1})$ is satisfied. This yields by the construction of (k, l) that $\theta_v^{m-l-1} \succeq \theta_u^{n-k-1}$. Now we use lemma 2.4.18 to deduce that $\theta_v^{m-l} \succeq \theta_u^{n-k}$. This is a contradiction to the definition of k and l .

□

REMARK 2.4.16:

In this remark we show the connection to proposition 2.4.12. Due to the assumptions of this proposition we have with point 8 and 9 of proposition 2.4.6 either $\tilde{s}_m \leq \tilde{t}_n \leq t_n = s_m < s_{m+1} \leq \tilde{s}_{m+1}$ or $\tilde{t}_n \leq \tilde{s}_m \leq s_m = t_n < t_{n+1} \leq \tilde{t}_{n+1}$. Thus $C(\tilde{t}_n, \tilde{s}_m)$ is true and we can apply proposition 2.4.15 which gives us that $\theta_v^n \succeq \theta_u^n$. Therefore proposition 2.4.12 true.

We used in the proof of Proposition 2.4.15 the two lemmata 2.4.17 and 2.4.18. We state and prove now these two lemmata. The proofs are similar to the proofs of lemmata 2.4.10 and 2.4.11.

LEMMA 2.4.17 (Two jumps, two arrivals):

Assume that $\theta_v^{m-l} \succeq \theta_u^{n-k}$ for $0 \leq k \leq p$, $0 \leq l \leq q$. Then, if $\tilde{s}_{m-l} \leq \tilde{t}_{n-k+1} < \tilde{s}_{m-l+1}$ (resp. $\tilde{t}_{n-k} \leq \tilde{s}_{m-l+1} < \tilde{t}_{n-k+1}$), we have $\theta_v^{m-l} \succeq \theta_u^{n-k+1}$ (resp. $\theta_v^{m-l+1} \succeq \theta_u^{n-k+1}$).

Proof. We show here the proof in the case $\tilde{s}_{m-l+1} > \tilde{t}_{n-k+1}$. The other case is equivalent due to the symmetry of the algorithm, see also proposition 2.4.8.

We use point 8 and 9 of proposition 2.4.6 to conclude that $\tilde{t}_{n+1} \geq t_{n+1} > t_n = s_m \geq \tilde{s}_m \geq \tilde{s}_{m-l+1}$ for $1 \leq l \leq q+1$. Thus we know that $1 \leq k \leq p$ and $t_{n-k+1} = t_n$.

We do the proof by contradiction. We assume that $\theta_v^{m-l} \not\succeq \theta_u^{n-k+1}$. Thus there exists a point I with

$$\begin{cases} \theta_{v,I}^{m-l} < \theta_{u,I}^{n-k+1}, \\ \text{or} \\ \theta_{v,I}^{m-l} = \theta_{u,I}^{n-k+1} = \sigma_I \text{ and } \exists K : I \in \mathcal{U}_v^{m-l}(K) \cap \mathcal{U}_u^{n-k+1}(K) \text{ and } \sigma_I u_{I \rightarrow K}^{n-k+1} < \sigma_I v_{I \rightarrow K}^{m-l}. \end{cases} \quad (2.4.19)$$

We distinguish two cases: in the first case the value of θ_u^{n-k+1} changes but not in the second case. This is analogous to the cases in the proof of lemma 2.4.10.

Case 1: $I \in NA_u^{n-k+1}$

Subcase 1.1: $\theta_{v,I}^{m-l} > \theta_{u,I}^{n-k}$

We know $\theta_{v,I}^{m-l} = 1$ and $\theta_{u,I}^{n-k} = -1$. Since $I \in NA_u^{n-k+1}$, we deduce that $\theta_{u,I}^{n-k+1} = 1 = \theta_{v,I}^{m-l} = \sigma_I$. Due to $I \in \mathcal{U}_v^{m-l}(K)$ and point 2 of proposition 2.4.6 we gain

$$v_{I \rightarrow K}^{m-l} \leq s_{m-l}.$$

Further we know by $I \in NA_u^{n-k+1} \cap \mathcal{U}_u^{n-k+1}(K)$ and point 4 of proposition 2.4.6 that

$$t_{n-k+1} = u_{I \rightarrow K}^{n-k+1}.$$

Together with $s_{m-l} = t_{n-k+1}$ from the assumption this yields

$$v_{I \rightarrow K}^{m-l} \leq s_{m-l} \leq t_{n-k+1} = u_{I \rightarrow K}^{n-k+1}$$

which is a contradiction to (2.4.19).

Subcase 1.2: $\theta_{v,I}^{m-l} = \theta_{u,I}^{n-k}$

We distinguish now the cases ± 1 .

2 Generalized fast marching method

Subcase 1.2.1: $\theta_{v,I}^{m-l} = \theta_{u,I}^{n-k} = 1$

Since $I \in NA_u^{n-k+1}$ we know that $\theta_{u,I}^{n-k+1} = -1$, but this contradicts (2.4.19).

Subcase 1.2.2: $\theta_{v,I}^{m-l} = \theta_{u,I}^{n-k} = -1$

Step 1: $\tilde{u}_I^{n-k} \geq \tilde{v}_I^{m-l}$

Since $I \in NA_u^{n-k-1}$, we know that $I \in NB_{u,-}^{n-k}$. Further we recall that $\theta_v^{m-l} \succeq \theta_u^{n-k}$ as well as $\theta_{v,I}^{m-l} = \theta_{u,I}^{n-k} = -1$. In addition we have $t_{n-k} = s_{m-l}$. Thus the assumptions of lemma 2.4.13 are fulfilled and we conclude

$$\tilde{u}_I^{n-k} \geq \tilde{v}_I^{m-l}.$$

Step 2: Contradiction

Due to $I \in NA_u^{n-k+1}$, we know that $\tilde{t}_{n-k+1} = \tilde{u}_I^{n-k}$. Furthermore we know due the definition that $\tilde{v}_I^{m-l} \geq \tilde{s}_{m-l+1}$. Together with step 1 we conclude

$$\tilde{t}_{n-k+1} = \tilde{u}_I^{n-k} \geq \tilde{v}_I^{m-l} \geq \tilde{s}_{m-l+1}.$$

This is a contradiction to $\tilde{s}_{m-l+1} > \tilde{t}_{n-k+1}$.

Subcase 1.3: $\theta_{v,I}^{m-l} < \theta_{u,I}^{n-k}$

This case is impossible since we assume $\theta_v^{m-1} \succeq \theta_u^{n-1}$.

Case 2: $I \notin NA_u^{n-k+1}$

Subcase 2.1: $\theta_{v,I}^{m-l} < \theta_{u,I}^{n-k+1}$

Since $I \notin NA_u^{n-k+1}$, we have $\theta_{u,I}^{n-k+1} = \theta_{u,I}^{n-k}$ and therefore $\theta_{v,I}^{m-l} < \theta_{u,I}^{n-k}$. This is a contradiction to $\theta_v^{m-l} \succeq \theta_u^{n-k}$.

Subcase 2.2: $\theta_{v,I}^{m-l} = \theta_{u,I}^{n-k+1}$

Due to $I \notin NA_u^{n-k+1}$, we have

$$\theta_{u,I}^{n-k} = \theta_{u,I}^{n-k+1} = \theta_{v,I}^{m-l} = \sigma_I = \pm 1. \quad (2.4.20)$$

From equation (2.4.19) we get, that there exists a node K such that

$$I \in \mathcal{U}_v^{m-l}(K) \cap \mathcal{U}_u^{n-k+1}(K) \text{ and } \sigma_I u_{I \rightarrow K}^{n-k+1} < \sigma_I v_{I \rightarrow K}^{m-l} \quad (2.4.21)$$

We distinguish three subcases:

Subcase 2.2.1: $I \in \mathcal{U}_u^{n-k}(K)$

Equation (2.4.21) and the assumption $I \in \mathcal{U}_u^{n-k}(K)$ gives us that $I \in \mathcal{U}_v^{m-l}(K) \cap \mathcal{U}_u^{n-k}(K)$. Together with the assumption $\theta_v^{m-l} \succeq \theta_u^{n-k}$ and (2.4.20) we conclude that

$$\sigma_I u_{I \rightarrow K}^{n-k} \geq \sigma_I v_{I \rightarrow K}^{m-l}.$$

Furthermore it holds $I \in \mathcal{U}_u^{n-k}(K) \cap \mathcal{U}_u^{n-k+1}(K)$. Point 6 in proposition 2.4.6 gives us that $u_{I \rightarrow K}^{n-k+1} = u_{I \rightarrow K}^{n-k}$. Thus we have.

$$\sigma_I u_{I \rightarrow K}^{n-k+1} \geq \sigma_I v_{I \rightarrow K}^{m-l}$$

which contradicts (2.4.21).

Subcase 2.2.2: $I \notin \mathcal{U}_u^{n-k}(K)$ and $\sigma_I = +1$

We have $I \in \mathcal{U}_u^{n-k+1}(K) \setminus \mathcal{U}_u^{n-k}(K)$. Using point 7 of proposition 2.4.6 we conclude that $u_{I \rightarrow K}^{n-k+1} = t_{n-k+1}$. In addition we know $I \in \mathcal{U}_v^{m-l}(K)$ and point 2 of proposition 2.4.6 leads to $s_{m-l} \leq v_{I \rightarrow K}^{m-l}$. In combination with the assumption of the lemma we conclude

$$u_{I \rightarrow K}^{n-k+1} = t_{n-k+1} = s_{m-l} \geq v_{I \rightarrow K}^{m-l}.$$

This is a contradiction to (2.4.21).

Subcase 2.2.3: $I \notin \mathcal{U}_u^{n-k}(K)$ and $\sigma_I = -1$

The equations (2.4.20) and (2.4.21) give us $I \in \mathcal{U}_v^{m-l}(K) \cap \mathcal{U}_u^{n-k+1}(K)$ and $\theta_{u,I}^{n-k} = \theta_{u,I}^{n-k+1} = \theta_{v,I}^{m-l} = -1$. Furthermore we have $t_{n-k} = s_{m-l}$. The assumptions of lemma 2.4.14 are fulfilled and thus we get $I \in \mathcal{U}_u^{n-k}(K)$, which leads to a contradiction.

Subcase 2.3: $\theta_{v,I}^{m-l} > \theta_{u,I}^{n-k}$

This case is impossible. See also equation (2.4.19).

□

LEMMA 2.4.18 (Two jumps, one arrival):

Assume that $\theta_v^{m-l} \geq \theta_u^{n-k+1}$ for $1 \leq k \leq p$, $1 \leq l \leq q$. If $\tilde{s}_{m-l+1} = \tilde{t}_{n-k+1}$, then $\theta_v^{m-l+1} \geq \theta_u^{n-k+1}$.

Proof. We do the proof by contradiction. Let us assume that $\theta_v^{m-l+1} \not\geq \theta_u^{n-k+1}$. Then there exists a point I with

$$\left\{ \begin{array}{l} \theta_{v,I}^{m-l+1} < \theta_{u,I}^{n-k+1} \\ \text{or} \\ \theta_{v,I}^{m-l+1} = \theta_{u,I}^{n-k+1} = \sigma_I \text{ and } \exists K : I \in \mathcal{U}_v^{m-l+1}(K) \cap \mathcal{U}_u^{n-k+1}(K) \text{ and } \sigma_I u_{I \rightarrow K}^{n-k+1} < \sigma_I v_{I \rightarrow K}^{m-l+1} \end{array} \right. \quad (2.4.22)$$

We consider the following four cases:

Case 1: $I \in NA_u^{n-k+1} \setminus NA_v^{m-l+1}$

Subcase 1.1: $\theta_{v,I}^{m-l} > \theta_{u,I}^{n-k}$

We know $\theta_{v,I}^{m-l} = 1$ and $\theta_{u,I}^{n-k} = -1$. Since $I \in NA_u^{n-k+1} \setminus NA_v^{m-l+1}$, we deduce that $\theta_{u,I}^{n-k+1} = 1 = \theta_{v,I}^{m-l+1} = \sigma_I$. Due to $I \in \mathcal{U}_v^{m-l+1}(K)$ and point 2 of proposition 2.4.6 we gain

$$v_{I \rightarrow K}^{m-l+1} \leq s_{m-l+1}.$$

2 Generalized fast marching method

Further we know by $I \in NA_u^{n-k+1} \cap \mathcal{U}_u^{n-k+1}(K)$ and point 4 of proposition 2.4.6 that

$$t_{n-k+1} = u_{I \rightarrow K}^{n-k+1}.$$

Together with $s_{m-l+1} = t_{n-k+1}$ from the assumption this yields

$$v_{I \rightarrow K}^{m-l+1} \leq s_{m-l+1} = t_{n-k+1} = u_{I \rightarrow K}^{n-k+1}$$

which is a contradiction to (2.4.22).

Subcase 1.2: $\theta_{v,I}^{m-l} = \theta_{u,I}^{n-k}$

We distinguish now the cases ± 1 .

Subcase 1.2.1: $\theta_{v,I}^{m-l} = \theta_{u,I}^{n-k} = +1$

Due to $I \in NA_u^{n-k+1} \setminus NA_v^{m-l+1}$ we know that $\theta_{u,I}^{n-k+1} = -1$ and $\theta_{v,I}^{m-l+1} = 1$, which contradicts (2.4.22).

Subcase 1.2.2: $\theta_{v,I}^{m-l} = \theta_{u,I}^{n-k} = -1$

Step 1: $\tilde{u}_I^{n-k} \geq \tilde{v}_I^{m-l}$

Since $I \in NA_u^{n-k+1}$ and $\theta_{u,I}^{n-k} = -1$ we know that $I \in NB_{u,-}^{n-k}$. Moreover it holds $t_{n-k} = s_{m-l}$. Further we recall that $\theta_v^{m-l} \succeq \theta_u^{n-k}$ and $\theta_{v,I}^{m-l} = \theta_{u,I}^{n-k+1} = -1$. Thus the assumptions of lemma 2.4.13 are fulfilled and we conclude

$$\tilde{u}_I^{n-k} \geq \tilde{v}_I^{m-l}.$$

Step 2: Contradiction

Due to $s_{m-l+1} = s_{m-l}$ and $I \notin NA_v^{m-l+1}$ we conclude using step 8 of the mGFT that $\tilde{s}_{m-l+1} < \tilde{v}_I^{m-l}$. Together with step 1 we get

$$\tilde{u}_I^{n-k} \geq \tilde{v}_I^{m-l} > \tilde{s}_{m-l+1} = \tilde{t}_{n-k+1}.$$

This is impossible because $I \in NA_u^{n-k+1}$ with step 8 of the mGFT yields $\tilde{u}_I^{n-k} = \tilde{t}_{n-k+1}$.

Subcase 1.3: $\theta_{v,I}^{m-l} < \theta_{u,I}^{n-k}$

This case is impossible since we assume $\theta_v^{m-l} \succeq \theta_u^{n-k}$.

Case 2: $I \in NA_v^{m-l+1} \setminus NA_u^{n-k+1}$

Due to the symmetry of the mGFT in proposition 2.4.8, this case can be handled like case 1.

Case 3: $I \in NA_u^{n-k+1} \cap NA_v^{m-l+1}$

Subcase 3.1: $\theta_{v,I}^{m-l} > \theta_{u,I}^{n-k}$

We know $\theta_{v,I}^{m-l} = 1$ and $\theta_{u,I}^{n-k} = -1$. Due to $I \in NA_u^{n-k+1} \cap NA_v^{m-l+1}$ we conclude that $I \in NB_{u,-}^{n-k} \cap NB_{v,+}^{m-l}$. Therefore we have

$$\hat{c}_{u,I}^{n-k} > 0 \text{ and } \hat{c}_{v,I}^{m-l} < 0.$$

This is a contradiction because $s_{m-l} = t_{n-k}$ and thus $c_{v,I}^{m-l} \geq c_{u,I}^{n-k}$.

Subcase 3.2: $\theta_{v,I}^{m-l} = \theta_{u,I}^{n-k}$

Since $I \in NA_u^{n-k+1} \cup NA_v^{m-l+1}$ we have $\theta_{v,I}^{m-l+1} = \theta_{u,I}^{n-k+1}$. From equation (2.4.22) we know $I \in \mathcal{U}_u^{n-k+1}(K) \cap \mathcal{U}_v^{m-l+1}(K)$. With point 4 of proposition 2.4.6 we gain

$$v_{I \rightarrow K}^{m-l+1} = s_{m-l+1} = t_{n-k+1} = u_{I \rightarrow K}^{n-k+1}.$$

This is a contradiction to (2.4.22).

Subcase 3.3: $\theta_{v,I}^{m-l} < \theta_{u,I}^{n-k}$

This is impossible since $\theta_v^{m-l} \succeq \theta_u^{n-k}$.

Case 4: $I \notin NA_u^{n-k+1} \cup NA_v^{m-l+1}$

Subcase 4.1: $\theta_{v,I}^{m-l+1} < \theta_{u,I}^{n-k+1}$

Due to $I \notin NA_u^{n-k+1} \cup NA_v^{m-l+1}$ we have $\theta_{u,I}^{n-k} = \theta_{u,I}^{n-k+1}$ and $\theta_{v,I}^{m-l} = \theta_{v,I}^{m-l+1}$. Thus we get $\theta_{v,I}^{m-l} < \theta_{u,I}^{n-k}$. This is a contradiction to $\theta_{v,I}^{m-l} \succeq \theta_{u,I}^{n-k}$.

Subcase 4.2: $\theta_{v,I}^{m-l+1} = \theta_{u,I}^{n-k+1}$

Since $I \notin NA_u^{n-k+1} \cup NA_v^{m-l+1}$, we have

$$\theta_{u,I}^{n-k} = \theta_{u,I}^{n-k+1} = \theta_{v,I}^{m-l} = \theta_{v,I}^{m-l+1} = \sigma_I = \pm 1. \quad (2.4.23)$$

Due to equation (2.4.22) we know, that there exists a node K such that $I \in \mathcal{U}_v^{m-l+1}(K) \cap \mathcal{U}_u^{n-k+1}(K)$ and

$$\sigma_I u_{I \rightarrow K}^{n-k+1} < \sigma_I v_{I \rightarrow K}^{m-l+1} \quad (2.4.24)$$

We distinguish several subcases according to $\mathcal{U}(K)$:

Subcase 4.2.1: $I \in \mathcal{U}_u^{n-k}(K) \cap \mathcal{U}_v^{m-l}(K)$

Using (2.4.23) and the assumption $\theta_v^{m-l} \succeq \theta_u^{n-k}$ of the lemma we conclude

$$\sigma_I u_{I \rightarrow K}^{n-k} \geq \sigma_I v_{I \rightarrow K}^{m-l}.$$

Furthermore we have $I \in \mathcal{U}_u^{n-k}(K) \cap \mathcal{U}_u^{n-k+1}(K)$ and $I \in \mathcal{U}_v^{m-l}(K) \cap \mathcal{U}_v^{m-l+1}(K)$. Thus we get with point 6 of proposition 2.4.6 that $u_{I \rightarrow K}^{n-k+1} = u_{I \rightarrow K}^{n-k}$ and $v_{I \rightarrow K}^{m-l+1} = v_{I \rightarrow K}^{m-l}$. Therefore we have

$$\sigma_I u_{I \rightarrow K}^{n-k+1} \geq \sigma_I v_{I \rightarrow K}^{m-l+1},$$

which contradicts (2.4.24).

Subcase 4.2.2: $I \in \mathcal{U}_v^{m-l}(K) \setminus \mathcal{U}_u^{n-k}(K)$

Subcase 4.2.2.1: $\sigma_I = +1$

We use $I \in \mathcal{U}_u^{n-k+1}(K) \setminus \mathcal{U}_u^{n-k}(K)$ together with point 7 of proposition 2.4.6 which gives us $u_{I \rightarrow K}^{n-k+1} = t_{n-k+1}$. Furthermore we know that $I \in \mathcal{U}_v^{m-l+1}(K)$. Point 2 of proposition 2.4.6 gives us $s_{m-l+1} \geq v_{I \rightarrow K}^{m-l+1}$. Thus we see

$$u_{I \rightarrow K}^{n-k+1} = t_{n-k+1} = s_{m-l+1} \geq v_{I \rightarrow K}^{m-l+1},$$

which contradicts (2.4.24).

2 Generalized fast marching method

Subcase 4.2.2.2: $\sigma_I = -1$

We know $I \in \mathcal{U}_v^{m-l}(K) \cap \mathcal{U}_u^{n-k+1}(K)$ and furthermore $\theta_{u,I}^{n-k} = \theta_{u,I}^{n-k+1} = \theta_{v,I}^{m-l} = -1$. Using $t_{n-k} = s_{m-l}$ the assumptions of lemma 2.4.14 are satisfied and we get that $I \in \mathcal{U}_u^{n-k}(K)$. This is a contradiction.

Subcase 4.2.3: $I \in \mathcal{U}_u^{n-k}(K) \setminus \mathcal{U}_v^{m-l}(K)$

Due to the symmetry this case is done in the same way as the preceding subcase 4.2.2.

Subcase 4.2.4: $I \notin \mathcal{U}_u^{n-k}(K) \cup \mathcal{U}_v^{m-l}(K)$

We have $I \in \mathcal{U}_u^{n-k+1}(K) \setminus \mathcal{U}_u^{n-k}(K)$ and $I \in \mathcal{U}_v^{m-l+1}(K) \setminus \mathcal{U}_v^{m-l}(K)$. Via point 7 of proposition 2.4.6 we get $t_{n-k+1} = u_{I \rightarrow K}^{n-k+1}$ and $s_{m-l+1} = v_{I \rightarrow K}^{m-l+1}$. Thus we have

$$v_{I \rightarrow K}^{m-l+1} = s_{m-l+1} = t_{n-k+1} = u_{I \rightarrow K}^{n-k+1}.$$

This is a contradiction to (2.4.24).

Subcase 4.3: $\theta_{v,I}^{m-l+1} > \theta_{u,I}^{n-k+1}$

This case is impossible due to the assumption in equation (2.4.22).

□

2.5 Numerical Tests

In this section we present some numerical examples for the GFT and the mGFT. We adapt the examples presented in [CFFMo8, Section 6], where I also implemented the GFMM and the mGFMM. Therefore we can directly compare the results of the different versions of the generalized fast marching method. The numerical results for the mGFMM are also new because in [For09] and also in [CFM] no numerical examples for the mGFMM are given.

2.5.1 Comments on the implementation and numerical tests

The different versions of the generalized fast marching are implemented in Java using as far as possible the same code to have a direct comparison. The choice of Java has some pragmatical reasons. We have a direct interface to Matlab, the implementation is independent of the platform and Java supports a large API. The set operations are made with the `LinkedHashSet` class of Java which gives us at least theoretically constant complexity for the remove-, add- and contain- operation per element, and the complexity for an iterator over the set is linear in the number of elements stored in this structure. Thus we can reach at least theoretically optimal computational complexity. Further I modified the `ArrayHeap` class from the package `edu.stanford.nlp.util` of the NLP Project⁵ such that one can remove arbitrary objects from the heap, see also case (a) and (b) in subsection 2.3.3. The triangulations are calculated using the Triangle library

⁵<http://nlp.stanford.edu/software/lex-parser.shtml>

by Jonathan Richard Shewchuk⁶ and an interface to Matlab. In addition I had access to the code used by Rasch in [Ras07], which shows how to handle virtual updates.

In [CFFMo8] two methods to measure the error are used. The one method is to compute the difference between the area $\mathcal{A}(\cdot)$ of the set $\Omega_T^\pm = \{x \mid \theta(T, x) = \pm 1\}$ with the approximated set $\Omega_m^\pm = \{x \mid \theta_m^\pm(x) = \pm 1\}$, where m is the number of iterations to reach the final time T . We do not use this method because with this measure we cannot really detect convergence. This is already mentioned in [CFFMo8, p. 2946].

The other method used in [CFFMo8] is to compute the Hausdorff distance between the exact front and the approximated front. I do not use this kind of error measure because the Hausdorff distance may be too sensitive to topological changes of the front. I use instead of the Hausdorff distance the L^1 -norm. The L^1 -norm of $(\theta - \theta^\varepsilon)(T, \cdot)$ is in our case (up to the constant 2) equal to the Lebesgue measure of the set $\{\theta(T, \cdot) \neq \theta^\varepsilon(T, \cdot)\}$. This is because θ and θ^ε are characteristic functions except for scaling.

Now we want to compute $\int_D |(\theta - \theta^\varepsilon)(T, x)| dx$ where D denotes the computational domain. We approximate this error by replacing the difference $\theta - \theta^\varepsilon$ with $I[R[(\theta - \theta^\varepsilon)(T, \cdot)]]$, where I denotes the linear interpolation on a triangle (resp. the bilinear interpolation using a regular grid) and R denotes the restriction of a function to the grid points of the mesh. The approximated error $\int_D |I[R[(\theta - \theta^\varepsilon)(T, \cdot)]]| dx$ can easily be computed exactly which is roughly speaking a weighted summation of the points which lie in the false region. For the estimation of the error we use the triangle inequality

$$\|(\theta - \theta^\varepsilon)(T, \cdot)\|_1 \leq \|(\theta - I[R[\theta]])(T, \cdot)\|_1 + \|I[R[(\theta - \theta^\varepsilon)]](T, \cdot)\|_1 + \|(\theta^\varepsilon - I[R[\theta^\varepsilon]])(T, \cdot)\|_1.$$

Assume that the computational domain is the cube $[0, 1]^d$ with M grid points and that we have a typical distance between two adjacent nodes of $\mathcal{O}(h) = \mathcal{O}(M^{-1/d})$. Then the typical front crosses $\mathcal{O}(M^{1-(1/d)})$ cells, thus we know that the discretization error $\|(\theta - I[R[\theta]])(T, \cdot)\|_1$ and $\|(\theta^\varepsilon - I[R[\theta^\varepsilon]])(T, \cdot)\|_1$ can be estimated by

$$\begin{aligned} \|(\theta^\varepsilon - I[R[\theta^\varepsilon]])(T, \cdot)\|_1 &= \mathcal{O}(M^{-1/d}) = \mathcal{O}(h) \quad \text{and} \\ \|(\theta - I[R[\theta]])(T, \cdot)\|_1 &= \mathcal{O}(M^{-1/d}) = \mathcal{O}(h), \end{aligned}$$

because the volume of a grid cell is about $\mathcal{O}(M^{-1}) = \mathcal{O}(h^d)$ and θ differs from $I[R[\theta]]$ only in cells of the front. Therefore we get

$$\|(\theta - \theta^\varepsilon)(T, \cdot)\|_1 \leq \|I[R[(\theta - \theta^\varepsilon)]](T, \cdot)\|_1 + \mathcal{O}(h).$$

Thus we know that the error measure can only detect linear convergence. This is no restriction for our application because due to the construction of the generalized fast marching method we can only archive linear convergence.

We use in some examples the level set of a polynomial to calculate the exact solution. This method was already used in [CFFMo8, Section 6] and we briefly recall it here.

Assume that we have a linear speed function $c(t, x) = a(t)x + b(t)$ and an initial front $\Gamma_0 = \{x : P(0, x) = 0\}$, where P is a polynomial in x whose coefficients depends on t . Then the front Γ_t at time t is equal to $\{x : P(t, x) = 0\}$ if P suffices the equation

$$P_t(t, x) - |DP(t, x)| c(t, x) = 0, \quad (2.5.1)$$

⁶<http://www.cs.cmu.edu/~quake/triangle.html>

see also section 1.5.

- (a) Hyperplanes: We assume that $P(t, x) = A(t)x + B(t)$. If the coefficients are solutions of the ODE

$$\begin{aligned}\dot{A}(t) &= |A(t)| a(t) \\ \dot{B}(t) &= |A(t)| b(t)\end{aligned}$$

then P is a solution of (2.5.1) and therefore we can use this formulation to calculate the exact solution of the evolution.

- (b) Spheres: Let $P(t, x) = R(t)^2 - |x - x_0(t)|^2$. Then P suffices (2.5.1) if the coefficients solves the ODE

$$\begin{aligned}\dot{x}_0(t) &= a(t)R(t) \\ \dot{R}(t) &= x_0(t)a(t) + b(t).\end{aligned}$$

Before we start with the concrete numerical examples, we give here the default setting for our computation which holds if nothing else is reported. We use the update procedure of case (b) in 2.3.3 with time step $\Delta t = 4h$. That means that we recompute the speed and the tentative values of all points at the front at least after the time interval Δt . We further use always a square with sides of length L as the computational domain, and we set for unstructured grids with M nodes the grid spacing h to the number $L/(M^{1/d} - 1)$. Thus a regular grid with grid spacing h would have the same number of nodes as the unstructured grid. In the Triangle library which we use to calculate the triangulation we used a setting such that every angle in the triangulation is at least 30° and nevertheless we used for the GFT virtual updates in the case of a non-acute triangle. For the computation of the order of the error and the time we used the results of the eight finest meshes in the computation. The examples are computed on a virtual Linux server with Matlab R2010b and the host server has Intel X5680 processors. The algorithm is not parallelized, so all computation work on one CPU core.

2.5.2 Numerical examples

In the first three examples 2.5.2.1 to 2.5.2.3 we use the same settings as in [CFFMo8, Section 6, Test 1, 2 and 3]. Example 2.5.2.4 is motivated by [CFFMo8, Section 6, Test 4] but no quantitative results have been given therein. The last example 2.5.2.5 is dedicated to the effect of degenerated meshes and the usage of virtual updates.

2.5.2.1 Rotating line

In this example we use the initial function $P(0, x) = 1.5x_1 + x_2$ which gives us the initial state for θ by

$$\theta(0, x) = \begin{cases} 1 & \text{if } 1.5x_1 + x_2 > 0, \\ -1 & \text{else.} \end{cases}$$

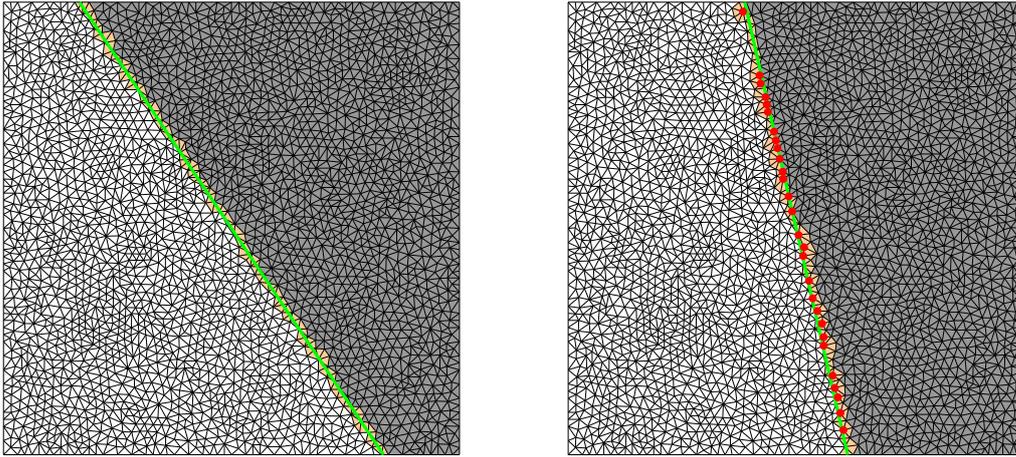


Figure 2.5.1: Solution for the example 2.5.2.1 with constant speed at $t = 0$ (left) and $t = 1$ (right). The green line indicates the location of the analytical front. The nodes at which analytic and computed solution differ are marked as red dots. The orange filled triangles indicate the numerical front.

We test here with two speed functions. The first one is $c(t, x) = x_1$. The case of the other speed function will be handled later. With the remarks above we know that the coefficients of $P(t, x) = A(t)x + B(t)$ are solutions of the ODE

$$\begin{cases} \dot{A}_1(t) = \sqrt{1 + A_1(t)^2}, & \begin{cases} \dot{A}_2(t) = 0, \\ A_2(0) = 1, \end{cases} & \begin{cases} \dot{B}(t) = 0, \\ B(0) = 0. \end{cases} \\ A_1(0) = 1.5, \end{cases}$$

One gets that

$$P(t, x) = \sinh(t + \operatorname{asinh}(A_1(0)))x_1 + x_2$$

is the corresponding solution for P . We compute the solution on the domain $D = [-1, 1]^2$ till the time $T = 1$. In figure 2.5.1 the initial data and the computed solution at time $T = 1$ are displayed. The mesh consists of 2785 nodes which corresponds to a grid spacing of $h = 0.0386$. We can see that the error is located at a small band around the analytical front Γ_t . This behavior can also be observed in the other examples.

In figure 2.5.2 we can see that all four variants of the generalized fast marching converge linear to the solution. Further we see that in this case GFMM and mGFMM resp. GFT and mGFT compute the same solution. The reason for this is that the speed in every point is constant in time. Thus both versions are equivalent.

The dependence of the number of nodes to the computational time is displayed in figure 2.5.3.

We can see that the time is heavily reduced if we use the efficient update of the speed as indicated in subsection 2.3.3. Furthermore it makes practically no difference whether we use case (a) or (b). Thus it seems to be a reasonable choice to use case (b) as long we have no indication to use case (c). Due to the analysis in 2.3.3 we would predict an order of 1.5 for case (c) and an order of ≈ 1 for cases (a) and (b), but in our experiments we detect a significantly higher complexity. The reason for this behavior is not clear and may be connected to the internals of the virtual machine of Java. One can further

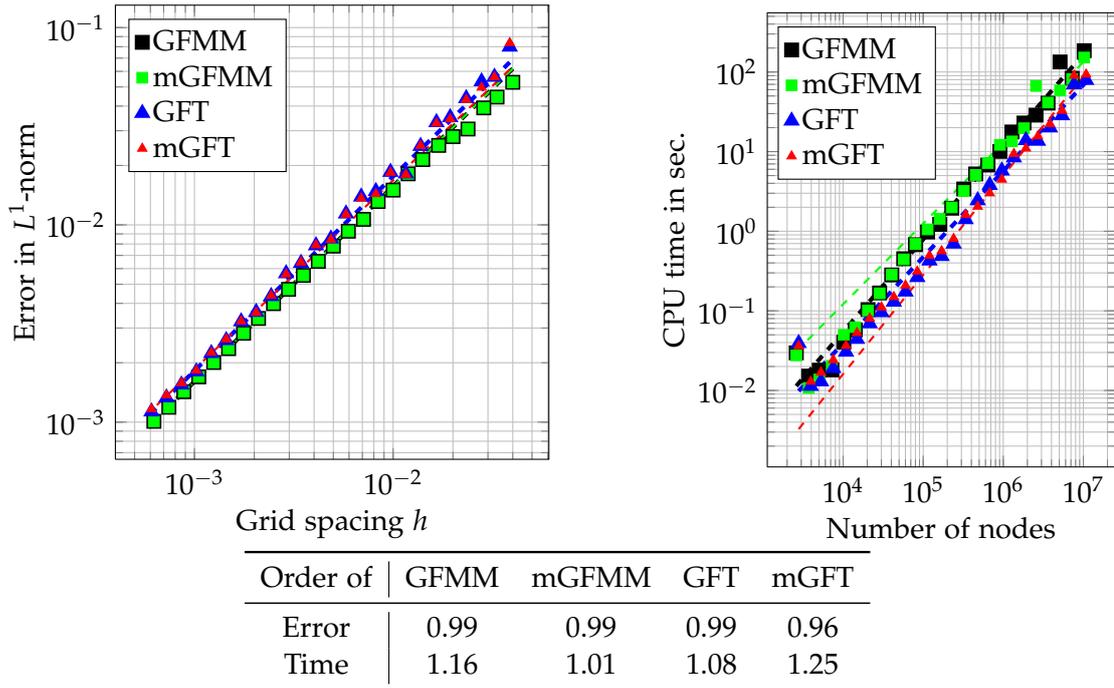


Figure 2.5.2: The approximated error in the L^1 -norm for the example in 2.5.2.1 with constant speed $c(t, x) = x_1$.

observe that the computation times differ about 30%, thus the results for the order of the time may not be accurate, but overall we can say that the observed complexity is higher than the complexity we predict.

On a notebook with an Intel P8400 processor we need with a grid spacing $h = 0.005$ about 1.2 sec. for the GFMM, 1.1 sec. for the mGFMM and the GFT and 1.3 sec. for the mGFT. This is about 60-times faster then the times presented in [CFFM08].

Now we change the speed function to $c(t, x) = \sin(2\pi t)x_1$. The front rotates clockwise for $0 \leq t \leq 1/2$, then counterclockwise for $1/2 \leq t \leq 1$ and so on. As in the previous calculation we get in this case for the polynomial

$$P(t, x) = \sinh(\operatorname{asinh}(3/2) + (1 + \cos(2\pi t))/(2\pi))x_1 + x_2. \quad (2.5.2)$$

The initial data and the computed solution for this choice of c are presented in figure 2.5.4.

In this setting we see that GFMM and mGFMM and likewise GFT and mGFT return slightly different results. Further we can recognize that the error of the modified versions is smaller. One can estimate such an behavior, because the mGFMM and mGFT cope also with the comparison principle and therefore they share more properties of the analytical evolution.

For the computation of the error we have to restrict the set of points we consider for this. The point is that in this case the solution of the evolution in \mathbb{R}^2 restricted to the Domain $D := [-1, 1]^2$ is not the same as when we solve the evolution on the domain D .

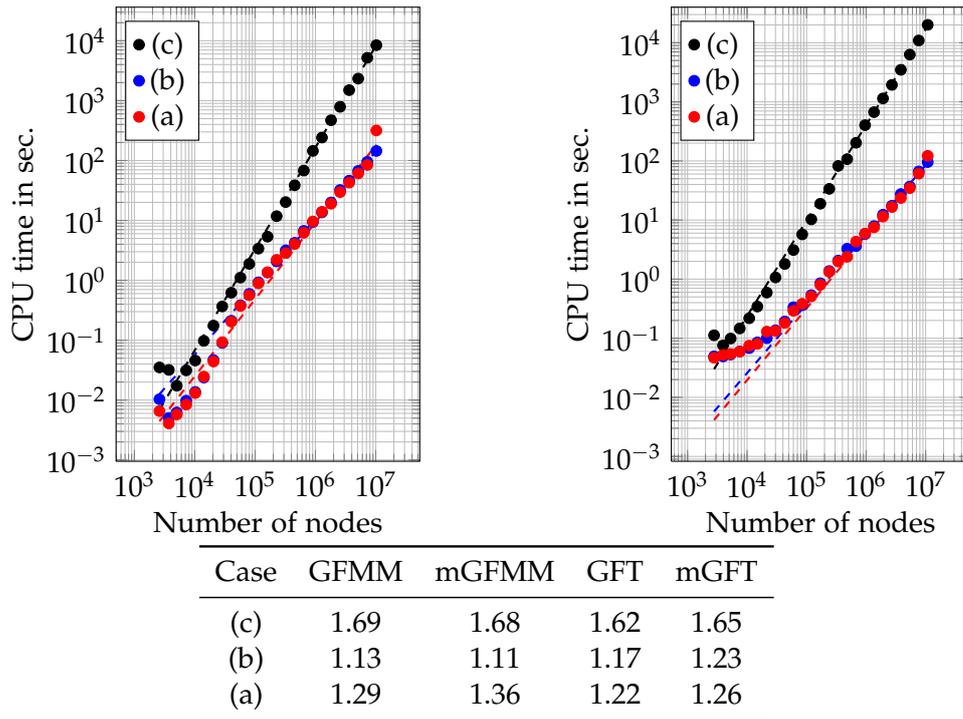


Figure 2.5.3: CPU time in seconds dependent on the number of grid points for the GFMM (left) and the GFT (right). We distinguish the cases (a), (b) and (c) as in subsection 2.3.3.

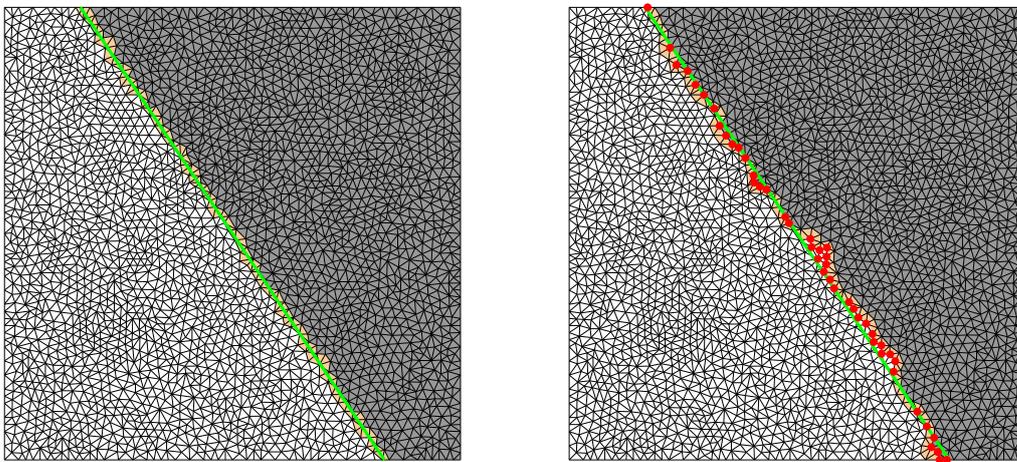


Figure 2.5.4: Initial data (left) and computed solution (right) at time $t = 1$ for the example 2.5.2.1 with speed $c(t, x) = \sin(2\pi t)x_1$.

In our case we have the effect that the front is rounded near the boundary of D if the line rotates counterclockwise. Thus we take only those points into account which lie in the smaller domain $[-0.9, 0.9]^2$.

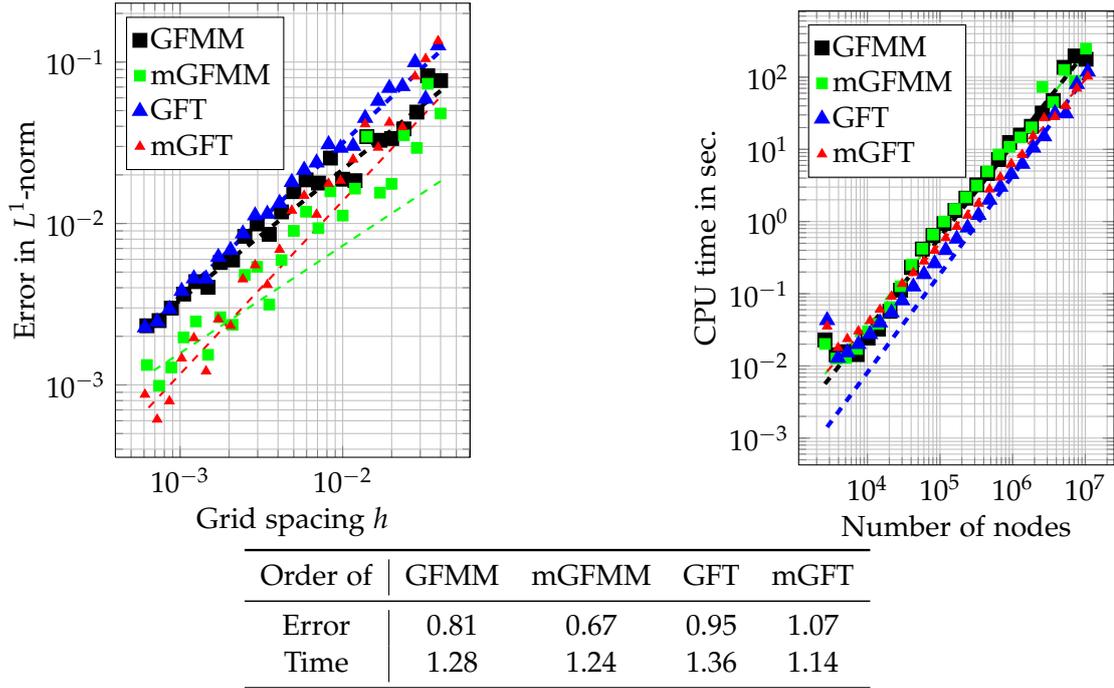


Figure 2.5.5: The approximated error in the L^1 -norm for the example in 2.5.2.1 with the non constant speed $c(t, x) = \sin(2\pi t)x_1$.

This issue can be very important because computations are restricted to bounded domains. Therefore we must ensure that either the front never reaches the boundary of the computational domain D or that at least the characteristics at the points of the front show outward when touching the boundary of D .

The time for the computation on a notebook with Intel P8400 processor for grid spacing $h = 0.005$ in the case of the non-constant speed is about 1.5 sec. for the GFMM, 1.3 sec. for the mGFMM and the GFT, and 1.9 sec. for the mGFT. This is about 80-times faster then the computational time presented in [CFFMo8].

2.5.2.2 Propagation of a circle

In this example we compute how a circle propagates due to the speed $c(t, x) = t/10 - x_1$. The initial data is given as the zero level set of $P(0, x) = x_1^2 + x_2^2 - 1$. Thus we have for θ in this case

$$\theta(0, x) = \begin{cases} 1 & \text{if } x_1^2 + x_2^2 - 1 < 0, \\ -1 & \text{otherwise.} \end{cases}$$

Using the formulas of the preceding subsection we gain for the coefficient of $P(t, x) = (x_1 - x_{0,1}(t))^2 + (x_2 - x_{0,2}(t))^2 - R(t)^2$ the ODE

$$\begin{cases} \dot{x}_{0,1}(t) = -R(t), \\ x_{0,1}(0) = 0, \end{cases} \quad \begin{cases} \dot{x}_{0,2}(t) = 0, \\ x_{0,2}(0) = 0, \end{cases} \quad \begin{cases} \dot{R}(t) = -x_{0,1}(t) + t/10, \\ R(0) = 1. \end{cases}$$

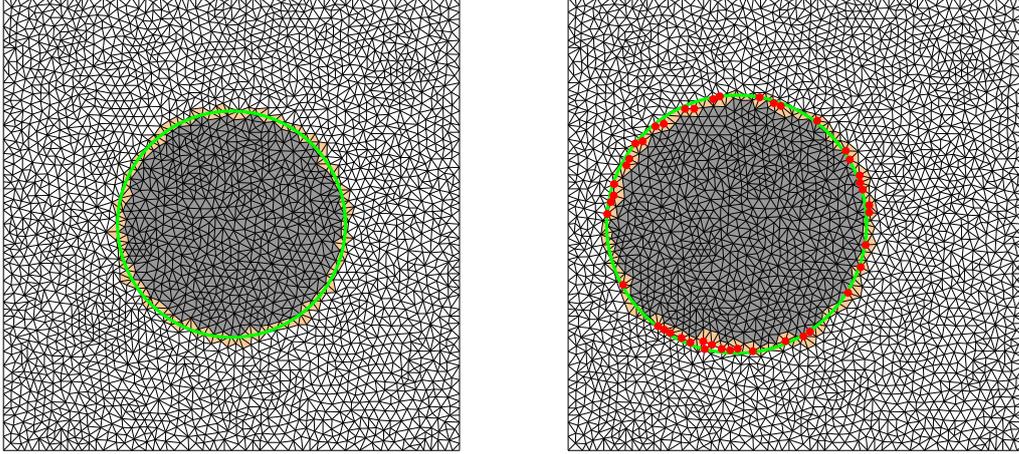


Figure 2.5.6: Initial data (left) and computed solution (right) at time $t = 0.5$ for the example 2.5.2.2.

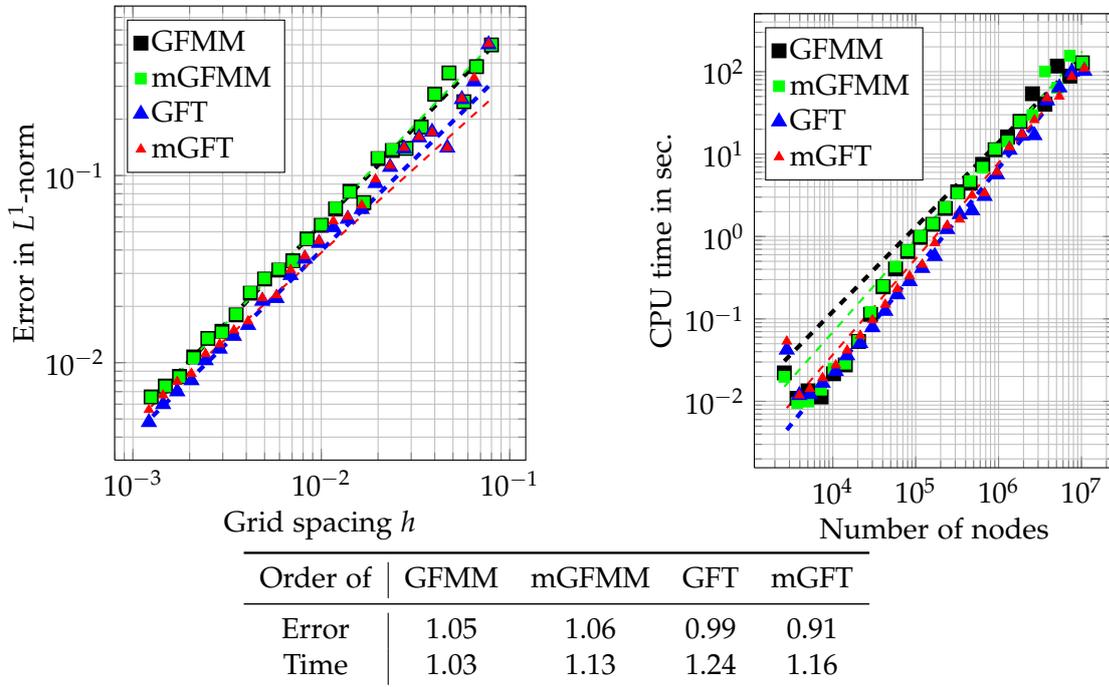


Figure 2.5.7: The approximated error in the L^1 -norm for the example in 2.5.2.2.

Hence we get for the solution

$$x_{0,1}(t) = (t - 11 \sinh(t))/10 \quad \text{and} \quad R(t) = (11 \cosh(t) - 1)/10,$$

which means that the circle moves to the left and expands. We use $D = [-2, 2]^2$ for the computational domain. Thus we know that the front will not touch the boundary till the time $T = 0.5$. In figure 2.5.6 we see the initial state and the result of our computation.

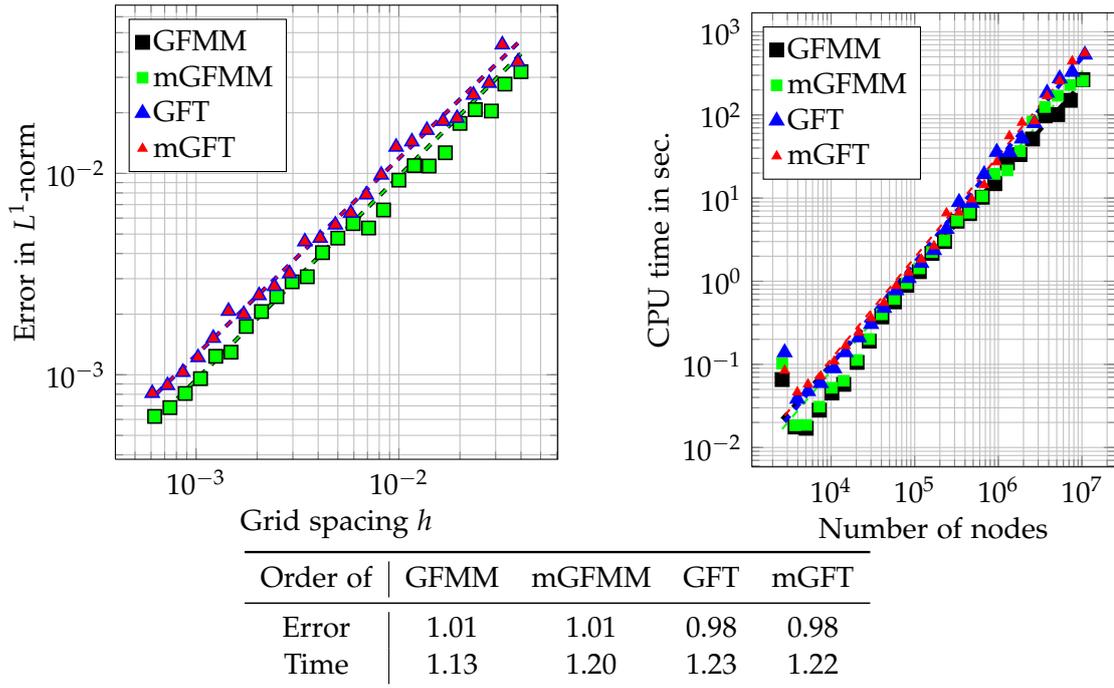


Figure 2.5.8: The approximated error in the L^1 -norm for the example in 2.5.2.3.

The mesh used in this figure is the same as used in the previous example, but it was scaled by a factor of 2. Thus we have a corresponding grid spacing of $h = 0.0773$.

In this example GFMM and mGFMM and likewise GFT and mGFT do not compute the same solution, because the speed changes its sign, but the difference is so small that it has no visible effect. The convergence is linear, as can be seen in figure 2.5.7.

We reuse this setting in the example in 2.5.2.5 where we test the effect of degenerated meshes and the usage of virtual updates.

2.5.2.3 Comparison to FMM

In this example we show that the generalized fast marching methods can in principle also be used as a FMM, although a pure FMM is of course much faster than the generalized one.

We want to compute the solution of

$$\begin{aligned} c(x) |DT(x)| &= 1 & x \in \Omega \\ T(x) &= 0 & x \in \partial\Omega. \end{aligned}$$

This problem is connected to the evolution of a front as shown in subsection 1.3.3 because the speed c does not change sign and is independent of t . Thus we are able to solve the minimum time problem using the equivalent representation as a evolution.

We set here $c(x) = 1$ and $\partial\Omega = \partial B(0, 1/2)$ and compute the T for $x \in [-2, 2]^2 \setminus B(0, 1/2)$. Due to the choice $c(x) = 1$ we know that $T(x)$ is just the distance between x and $B(0, 1/2)$ whose exact solution can easily be determined.

We must keep track in our algorithms of the time when the front crosses a point. In the GFMM and GFT this information is stored in the variable u_i^n . In the mGFMM and mGFT we have no such variable because we do not use this information. Therefore I added in the mGFMM and mGFT a variable u_i^n which stores this information in exactly the same way as in the GFMM and GFT, but we do not use this information for further computations within the algorithm.

In our example we have a constant speed. Thus GFMM and mGFMM and likewise GFT and mGFT produce the same results. We can see in figure 2.5.8 that the convergence is linear, and therefore the generalized fast marching methods are really an extension of the plain FMM. The FMM has the advantage that one can construct methods of high order accuracy whereas the generalized methods presented here are restricted to be of first order.

2.5.2.4 Collapsing circles

In this example we show an example that involves topological changes of the front. We set the computational domain as $D = [-2.6, 2.6]^2$. The initial state is

$$\theta(0, x) = \begin{cases} 1 & \text{if } x \in \Omega_0 \\ -1 & \text{otherwise,} \end{cases}$$

with $\Omega_0 = B((-1, -1/2), 1) \cup B((1, 1/2), 1)$, that means we start with two circles with radius 1, one centred at $p_1 = (-1, -1/2)$ the other at $p_2 = -p_1$. We choose $c(t, x) = 1 - t$ for the speed.

The analytical solution can be obtained by using the Huygens principle. For $0 \leq t \leq 1$ the circles expand and join. At $t = 1$ we have $\theta(1, x) = 1_{\Omega_1}(x)$ with $\Omega_1 = B(p_1, 1.5) \cup B(p_2, 1.5)$. The corners of $\partial\Omega_1$ are at $s_1 = (-\sin(\arctan(1/2)), \cos(\arctan(1/2))) = (-1/\sqrt{5}, 2/\sqrt{5})$ and $s_2 = -s_1$. For $1 \leq t$ the joined circles shrink again. We can construct $\theta(t, x)$ by regarding two circles at p_1 and p_2 with radius $R(t)$ and two circles at s_1 and s_2 with radius $r(t)$. For the radii $R(t)$ and $r(t)$ we get two simples ODEs, namely

$$\begin{cases} \dot{R}(t) = c(t), \\ R(0) = 1, \end{cases} \quad \text{and} \quad \begin{cases} \dot{r}(t) = -c(t), \\ r(1) = 0 \end{cases}$$

with $c(t) = c(t, x) = 1 - t$ as the speed function which is in our case independent of x . Thus we conclude that the front separates at time $1 + \sqrt{2}$ and that it disappears at time $t = 1 + \sqrt{3}$.

As we can see in figure 2.5.11 we have a slightly reduced order for convergence. The reason for this is that the mesh is still too coarse to enter a convergent regime but nevertheless the algorithms are still working fine.

In the figures 2.5.9 and 2.5.10 we show this evolution column by column for twelve different times. The circles which we used above for the construction are plotted green.

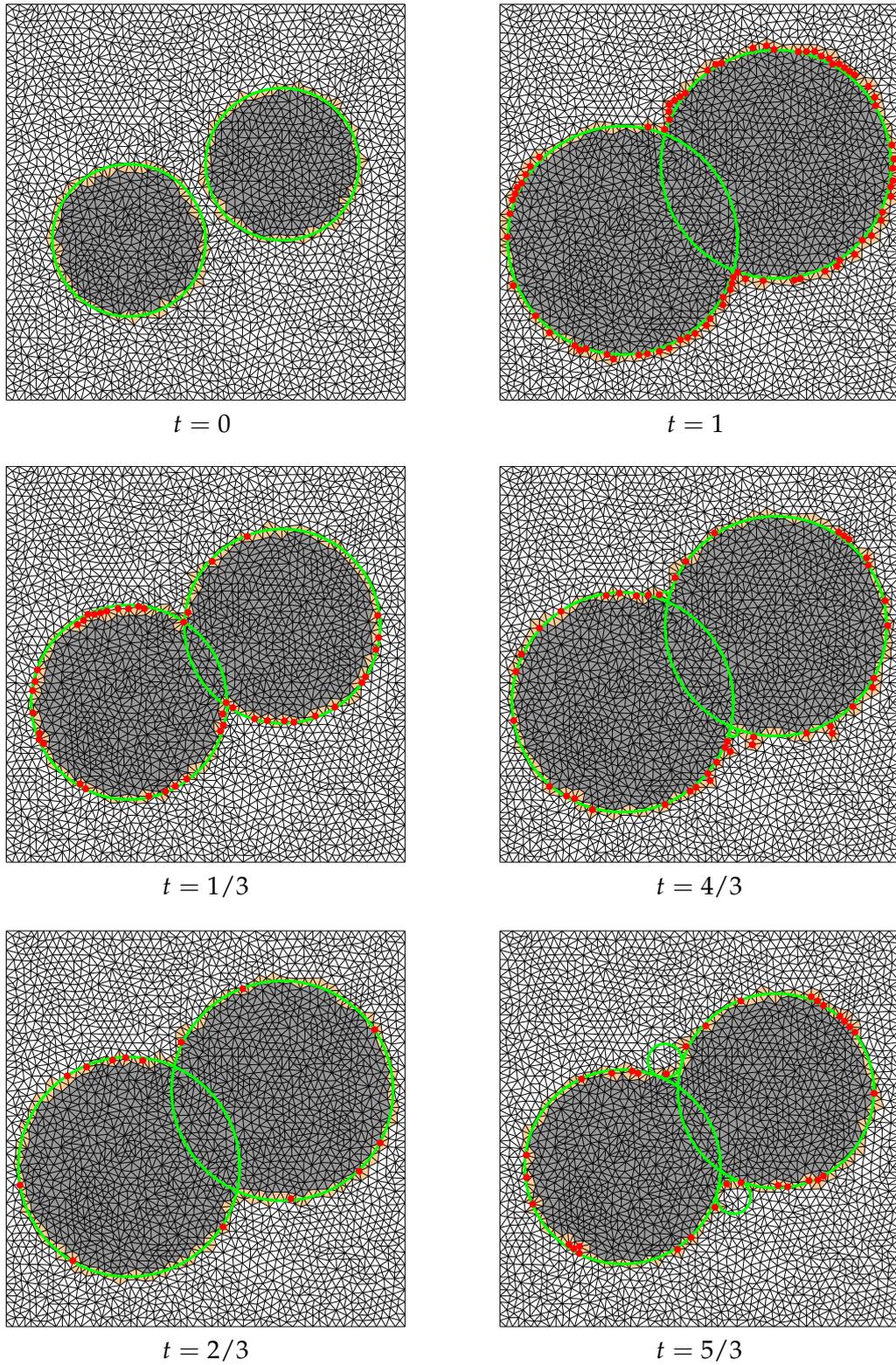
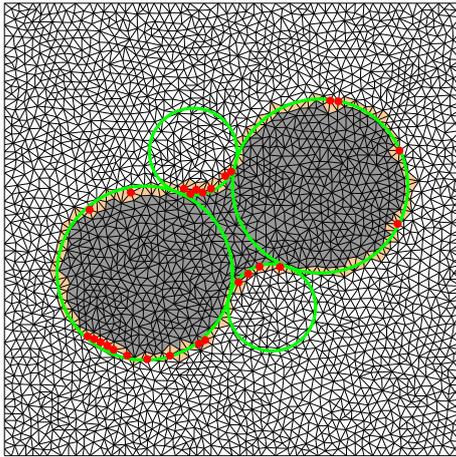
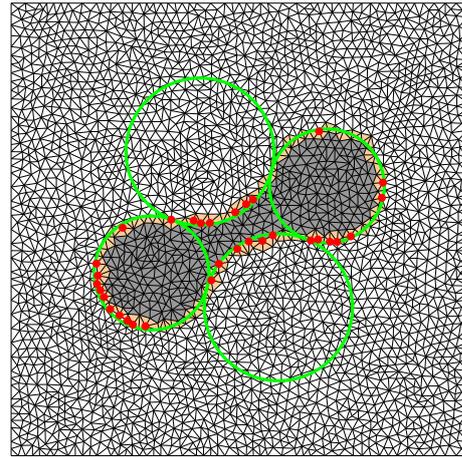


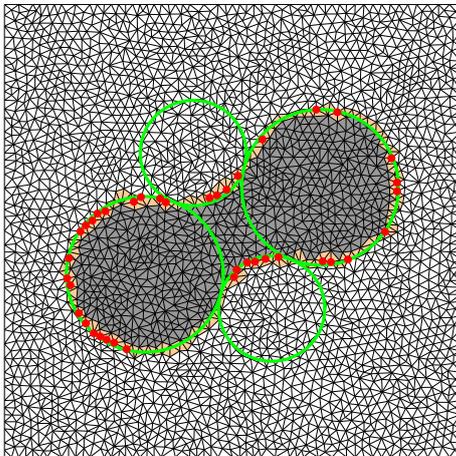
Figure 2.5.9: Evolution of example 2.5.2.4.



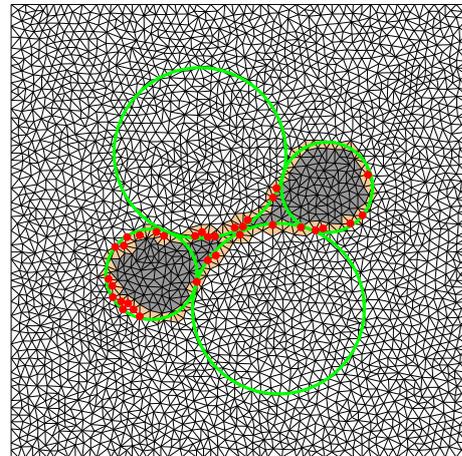
$t = 2$



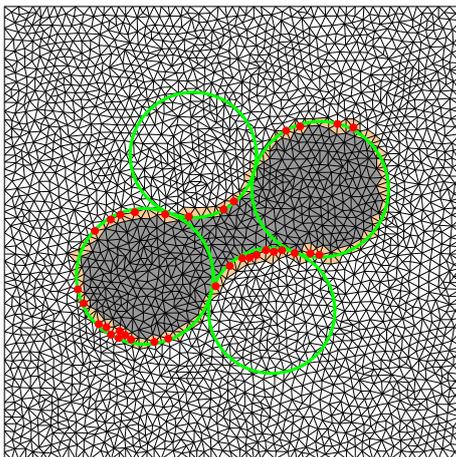
$t = 2.3$



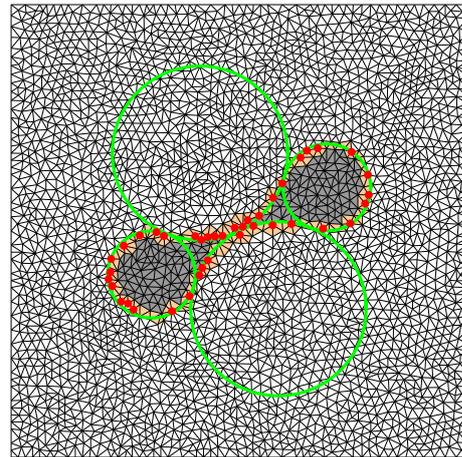
$t = 2.1$



$t = 2.4$



$t = 2.2$



$t = 1 + \sqrt{2}$

Figure 2.5.10: Evolution of example 2.5.2.4.

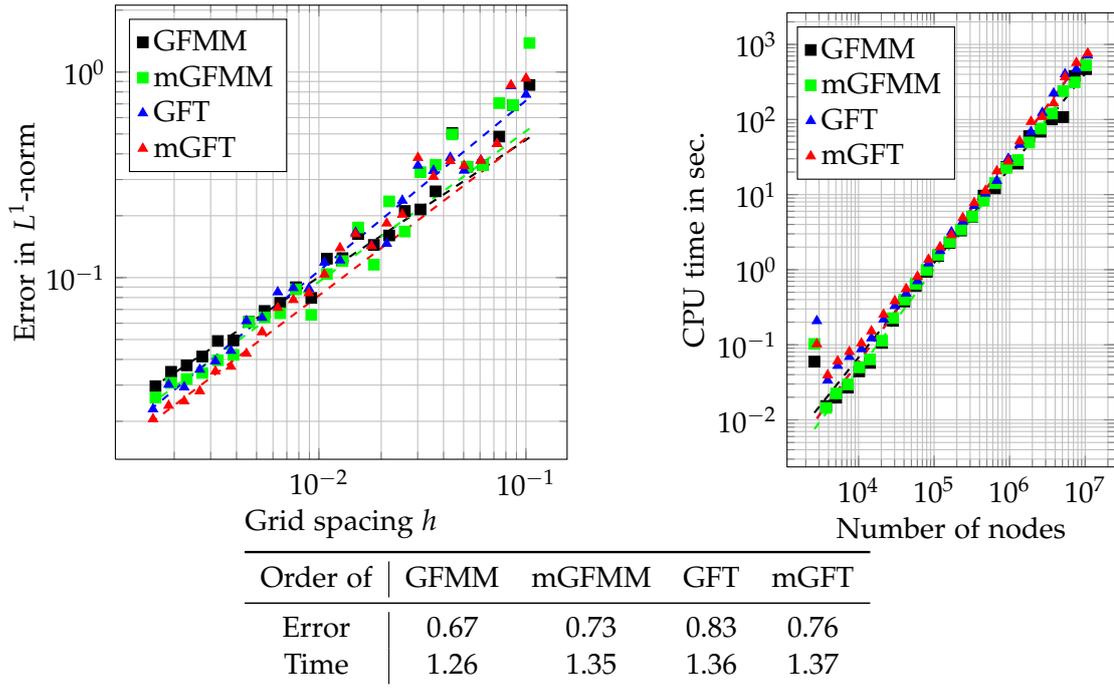


Figure 2.5.11: The approximated error at time $t = 1 + \sqrt{2}$ in the L^1 -norm for the example in 2.5.2.4.

2.5.2.5 Virtual updates

In this example we look at the effect of meshes with obtuse triangles and the usage of virtual updates. We reuse the settings of example 2.5.2.2, with the difference that we propagate the front till time $T = 0.6$ instead of $T = 0.5$. This is just to see a greater difference between computing with or without virtual updates. The mesh was generated by stretching a high quality mesh in x_2 -direction by a factor of 10.

In figure 2.5.12 we clearly see that the GFT does not converge if we do not use the method of virtual updates. Using virtual updates gives us at the beginning linear convergence which slows down to an order ≈ 0.4 . This effect may be related to the fact that using virtual updates, we do not exactly compute the underlying Hopf-Lax-updates but only an approximation.

In the figures 2.5.13 and 2.5.14 we plotted the result on a mesh with 1202 nodes. We clearly see that the front does not move fast enough to the left if we do not use virtual updates. Thus a systematical error is introduced and no convergence can be obtained. On the other hand using virtual updates we see that the front evolves properly and the error is only located in a small band near the analytical front, as in the other examples.

We conclude that from a practical point of view it is reasonable to use virtual updates every time obtuse triangles appear, but we cannot guarantee in a theoretical manner that the numerical solution converges to the analytic one. Even the comparison principle no longer holds because using virtual updates we heavily change the meaning of neighbourhood which is used in the proof of proposition 2.3.5.

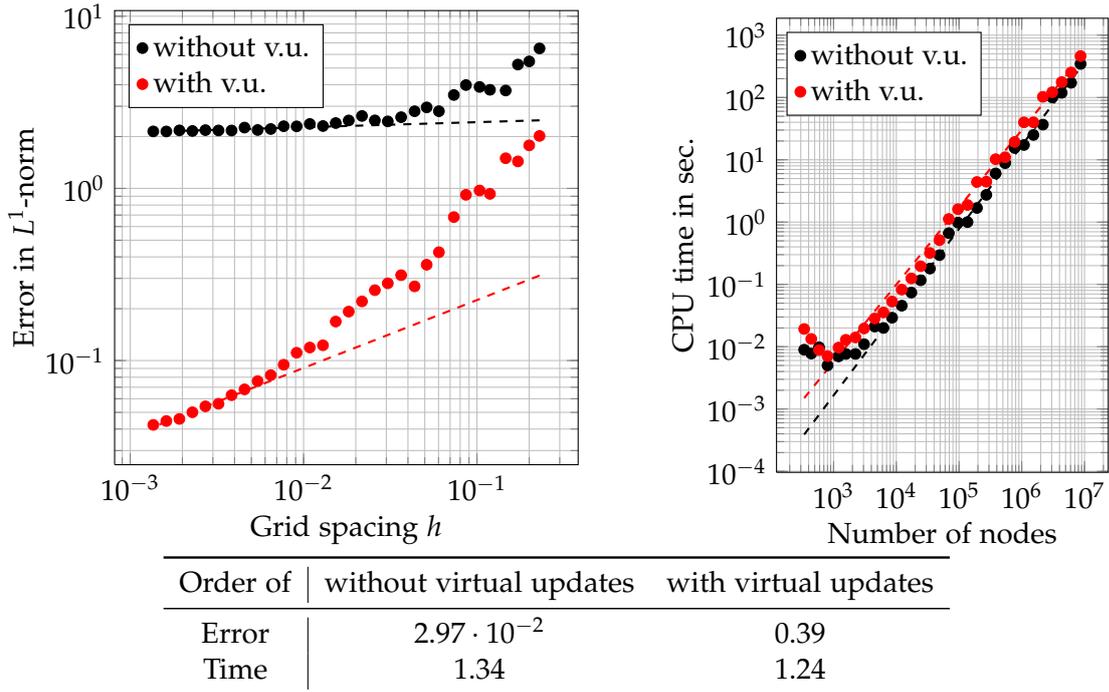


Figure 2.5.12: The approximated error in L^1 -norm for the example in 2.5.2.5.

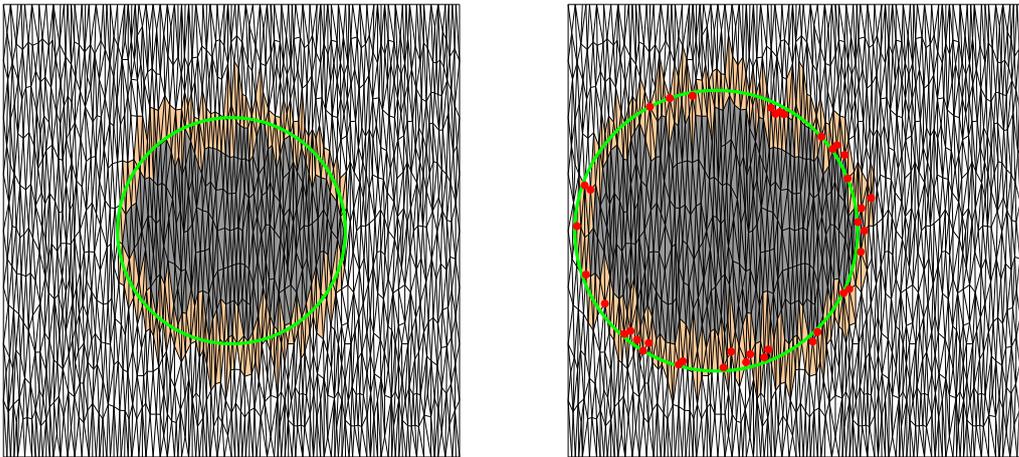


Figure 2.5.13: Initial data (left) and computed solution (right) at time $t = 0.6$ for the example 2.5.2.5 using virtual updates.

2.6 GFT for anisotropic speed metrics

In this section we extend the idea of the front propagation driven by a speed c in normal direction to speed functions that are induced by symmetric and definite matrices. Thus we will use the term metric instead of speed. Furthermore we apply the GFT to this problem to get an algorithm that solves this problem. We propose this algorithm and

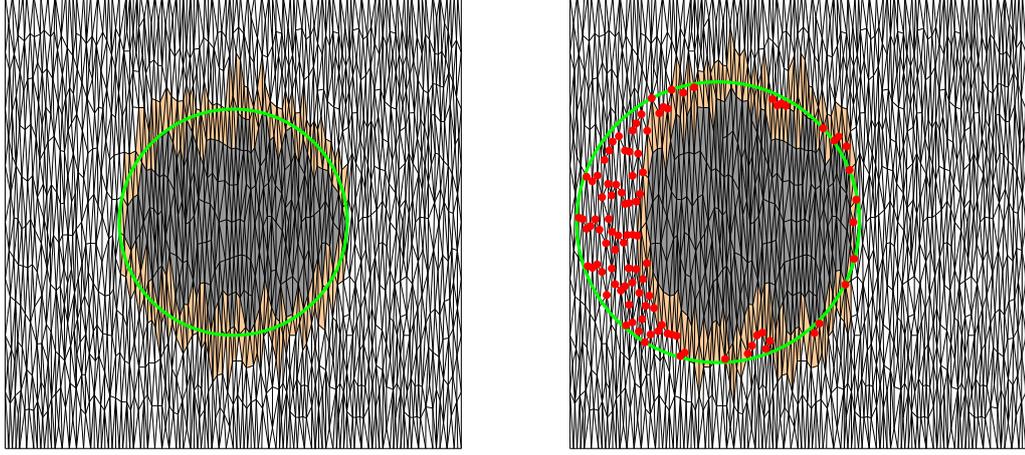


Figure 2.5.14: Initial data (left) and computed solution (right) at time $t = 0.6$ for the example 2.5.2.5 without virtual updates.

give also a numerical example.

2.6.1 Front propagation for anisotropic speed.

Until now we regarded the equation (2.0.1) which can be written for $c(t, x) > 0$ in the form

$$\begin{cases} \theta_t(t, x) - |\theta_x(t, x)|_{M(t, x)} = 0 & \text{on } (0, T) \times \mathbb{R}^d, \\ \theta(0, \cdot) = 1_{\Omega_0} - 1_{\Omega_0^c}, \end{cases}$$

with a symmetric positive definite (spd) matrix $M(t, x) = c(t, x)^2 \cdot I_d$. Now let us extend the definition of the square root in this section for all $x \in \mathbb{R}$ by $\sqrt{x} := \text{sgn}(x)\sqrt{|x|}$. With this kind of signed square root we can rewrite (2.0.1) for arbitrary functions c now by

$$\begin{cases} \theta_t(t, x) - |\theta_x(t, x)|_{M(t, x)} = 0 & \text{on } (0, T) \times \mathbb{R}^d, \\ \theta(0, \cdot) = 1_{\Omega_0} - 1_{\Omega_0^c}, \end{cases}$$

with $M(t, x) = c(t, x) |c(t, x)| \cdot I_d$. The matrix $M(t, x)$ is spd (or symmetric negative definite (snd)) if and only if $c(t, x)$ is positive (or negative). We remark that for a snd matrix M we get that $|p|_M = \sqrt{\langle Mp, p \rangle} \leq 0$ and that $-|\cdot|_M$ is a norm.

Until now we have only rewritten equation (2.0.1), but now we can extend the front propagation to anisotropic metrics. Let M be a continuous mapping $M : [0, \infty] \times \mathbb{R}^d \rightarrow \{-S^d \cup 0 \cup S^d\}$, where S^d is the set of the symmetric positive definite $d \times d$ matrices. Then we regard the equation

$$\begin{cases} \theta_t(t, x) - |\theta_x(t, x)|_{M(t, x)} = 0 & \text{on } (0, T) \times \mathbb{R}^d, \\ \theta(0, \cdot) = 1_{\Omega_0} - 1_{\Omega_0^c}. \end{cases} \quad (2.6.1)$$

2.6.2 Generalized fast marching for anisotropic speed functions

We want to apply the GFT to get an algorithm which we call generalized fast marching method for anisotropic metrics (GFA) that can compute a numerical solution of (2.6.1). First we look where in the GFT the speed c_I^n occurs. The first appearance is the definition 2.3.2 of the regularized speed \hat{c}_I^n . We extend that definition to

DEFINITION 2.6.1:

Given the metric $M_I^n := M(t_n, x_I)$. We define the regularized speed by the function

$$\hat{M}_I^n := \begin{cases} 0 & \text{if there exists } K \in V(I) \text{ such that } ((M_I^n \in S^d, M_K^n \in -S^d) \text{ or } (M_I^n \in -S^d, M_K^n \in S^d)) \\ & \text{and } |I - K|_{\text{abs}(M_K^n)^{-1}} \leq |I - K|_{\text{abs}(M_I^n)^{-1}} \\ M_I^n & \text{otherwise.} \end{cases}$$

Here $I - K$ denotes the vector from node K to node I .

REMARK 2.6.2:

In definition 2.6.1 we replace the condition $c_I^n c_K^n < 0$, which means that the speed c has a different sign at points I and K , with the condition $((M_I^n \in S^d, M_K^n \in -S^d) \text{ or } (M_I^n \in -S^d, M_K^n \in S^d))$ which says that the metric M is at one node positive definite and at the other negative definite.

The other condition $|c_I^n| \leq |c_K^n|$ can be interpreted as follows: We reach node I from node K with speed $|c_K^n|$ before we would reach K from node I with speed $|c_I^n|$. That means the time to reach node I from node K with speed $|c_K^n|$ is $h / |c_K^n|$ which is smaller than the time $h / |c_I^n|$ if we use speed c_I^n .

We first extend the notion of the absolute value. Thus we introduce for this section the notion $\text{abs}(\cdot)$ for matrices $M \in \{-S^d \cup 0 \cup S^d\}$ by

$$\text{abs}(M) = \begin{cases} -M & \text{if } -M \in S^d, \\ M & \text{otherwise.} \end{cases}$$

Thus $\text{abs}(\cdot)$ denotes for these matrices a kind of absolute value in the sense that $\text{abs}(M)$ is always a positive definite matrix (or the zero matrix).

With this notion we transfer the condition $|c_I^n| \leq |c_K^n|$ to $|I - K|_{\text{abs}(M_K^n)^{-1}} \leq |I - K|_{\text{abs}(M_I^n)^{-1}}$, that means we look in which metric we can go faster from I to K .

With this notion of \hat{M} we still have a monotone mapping $M \rightarrow \hat{M}$ as shown in the following proposition. It is analogous to proposition 2.4.7

PROPOSITION 2.6.3 (Monotonicity of $M \rightarrow \hat{M}$):

The mapping $M \rightarrow \hat{M}$ is monotone, that means if we have $M_v \geq M_u$ then we get $\hat{M}_v \geq \hat{M}_u$.

Proof. Let $M_v \geq M_u$. We will prove that we have for all nodes $I \in \mathcal{N}$

$$\hat{M}_{v,I} \geq \hat{M}_{u,I}. \quad (2.6.2)$$

We proof only the case $\hat{M}_{v,I} = M_{v,I}$ and $\hat{M}_{u,I} \neq M_{u,I}$. All other cases are done like in the proof of 2.4.7.

2 Generalized fast marching method

Case 3: $\widehat{M}_{v,I} = M_{v,I}$ and $\widehat{M}_{u,I} \neq M_{u,I}$.

Due to $\widehat{M}_{u,I} \neq M_{u,I}$ we have $\widehat{M}_{u,I} = 0$. We assume by contradiction that

$$\widehat{M}_{v,I} = M_{v,I} < 0 = \widehat{M}_{u,I}. \quad (2.6.3)$$

Thus we have $M_{v,I} \in -\mathcal{S}^d$. Due to $M_v \geq M_u$ we get that $M_{u,I} < 0$, that means $M_{u,I} \in -\mathcal{S}^d$. Since $\widehat{M}_{u,I} \neq M_{u,I}$, we know that there exists $K \in V(I)$ with $M_{u,K} \in \mathcal{S}^d$ and

$$|I - K|_{\text{abs}(M_{u,K})^{-1}} \leq |I - K|_{|M_{u,I}|^{-1}}. \quad (2.6.4)$$

Now we show that $\widehat{M}_{v,I} = 0$.

We have $M_{v,I} \in -\mathcal{S}^d$ and $M_{v,K} \geq M_{u,K} > 0$ thus $M_{v,K} \in \mathcal{S}^d$. Thus we get

$$\begin{aligned} |I - K|_{\text{abs}(M_{v,K})^{-1}} &\leq |I - K|_{\text{abs}(M_{u,K})^{-1}} && \text{because } M_v \geq M_u \text{ and } M_{u,K} \in \mathcal{S}^d \\ &\leq |I - K|_{\text{abs}(M_{u,I})^{-1}} && \text{due to 2.6.4} \\ &\leq |I - K|_{\text{abs}(M_{v,I})^{-1}} && \text{because } M_v \geq M_u \text{ and } M_{v,I} \in -\mathcal{S}^d \end{aligned}$$

Thus we have $\widehat{M}_{v,I} = 0$ and we have a contradiction to (2.6.3) and therefore we have proved (2.6.2). □

The other point which we have to change in the GFT is step 5 of the algorithm. The distinction due to the sign of c_I^n is replaced by the condition whether $M_I^n \in \pm\mathcal{S}^d$.

5. Compute \tilde{u}^{n-1} on F^{n-1} as

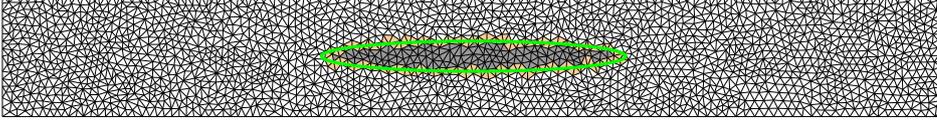
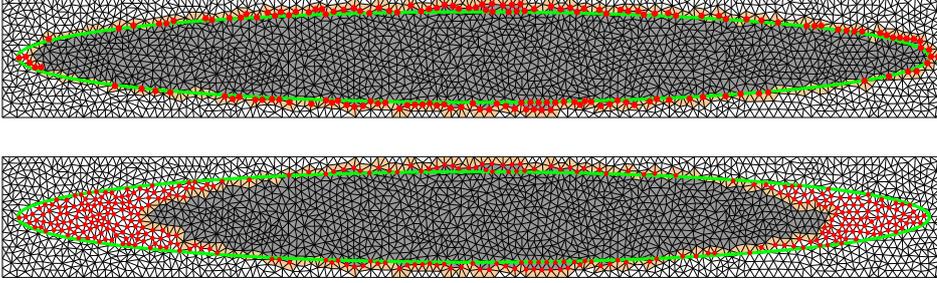
Let $I \in F_{\pm}^{n-1}$, then

- a) if $\pm\widehat{M}_I^{n-1} \in \{0 \cup \mathcal{S}^d\}$ then $\tilde{u}_I^{n-1} = \infty$.
- b) if $\pm\widehat{M}_I^{n-1} \in -\mathcal{S}^d$ then we compute \tilde{u}_I^{n-1} via the Hopf-Lax update. We have to distinguish two cases:
 - if $I \in F_-^{n-1}$, then $\tilde{u}_I^{n-1} = (\Lambda\widehat{u}_+^n)(I)$ with the speed $\text{abs}(\widehat{M}_I^{n-1})$.
 - if $I \in F_+^{n-1}$, then $\tilde{u}_I^{n-1} = (\Lambda\widehat{u}_-^n)(I)$ with the speed $\text{abs}(\widehat{M}_I^{n-1})$.

REMARK 2.6.4:

We change in step 5 of the GFT only the condition $\pm\widehat{c}_I^{n-1} \geq 0$ to $\pm\widehat{M}_I^{n-1} \in \{0 \cup \mathcal{S}^d\}$ and $\pm\widehat{c}_I^{n-1} < 0$ to $\pm\widehat{M}_I^{n-1} \in -\mathcal{S}^d$ which means that we compute only updates for points which lie on the upwind side of the front. The Hopf-Lax update has to be computed as mentioned in subsection 2.1.4. Thus we internally compute with a mesh that is locally transformed by $(M_I^n)^{-1/2}$. So the mesh can be heavily deformed and therefore we should use the methods of virtual updates. Furthermore this implies that we have no comparison principle like 2.3.5, because we need an acute triangulation for this and due to the local transformations of the grid in the Hopf-Lax update we cannot give any general predictions whether the transformed grids are acute or not.

Fortunately we have at least a method which works in practice and gives good numerical solutions of (2.6.1). In the following subsection we present such an example.

Figure 2.6.1: Initial data for the example in 2.6.3 with $Y = 10$.Figure 2.6.2: Computed solution for the example in 2.6.3 with $Y = 10$ at time $t = \pi$ with virtual updates (top) and without (bottom).

2.6.3 Numerical example for GFA

For this example we choose for the metric and the initial condition

$$M(t, x) = \begin{pmatrix} 1 & 0 \\ 0 & 1/Y^2 \end{pmatrix} \cdot \sin(t) \quad \text{and} \quad \theta(0, x) = \begin{cases} 1 & \text{if } x \in \Omega_0, \\ -1 & \text{otherwise,} \end{cases} \quad (2.6.5)$$

with $\Omega_0 = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 + (Yx_2)^2 \leq 1\}$. This can be interpreted as an ellipse with the major axis x_1 and minor axis x_2 with radii $R = 1$ and $r = 1/Y$. We choose the metric such that we can give the analytic solution, namely

$$\theta(t, x) = \begin{cases} 1 & \text{if } x \in \Omega_t \\ -1 & \text{otherwise,} \end{cases}$$

with $\Omega_t = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 + (Yx_2)^2 \leq (2 - \cos(t))^2\}$. The computational domain is $D = [-3.1, 3.1] \times [-(3Y + 0.1), 3Y + 0.1]$. In figure 2.6.1 we plot the initial data for $Y = 10$.

In figure 2.6.2 we see the state for the evolution at time $t = \pi$. We clearly see that we have to use virtual updates. Otherwise large errors are produced and we get an inappropriate approximation. Using this anisotropic metric has the same effect as compressing the hole domain in x_1 -direction by the factor \sqrt{Y} . Thus internally we compute with a highly irregular mesh as in example 2.5.2.5 and it is reasonable to use virtual updates.

Figure 2.6.3 shows the computed solution at time $t = 3\pi/2$. The effect of using virtual updates is more pronounced than in figure 2.6.2, and we see that there is no possibility of convergence if we do not use virtual updates.

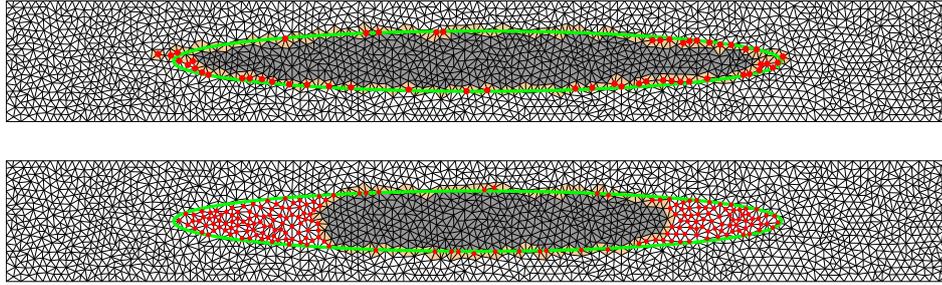


Figure 2.6.3: Computed solution for the example in 2.6.3 with $Y = 10$ at time $t = 3\pi/2$ with virtual updates (top) and without (bottom).

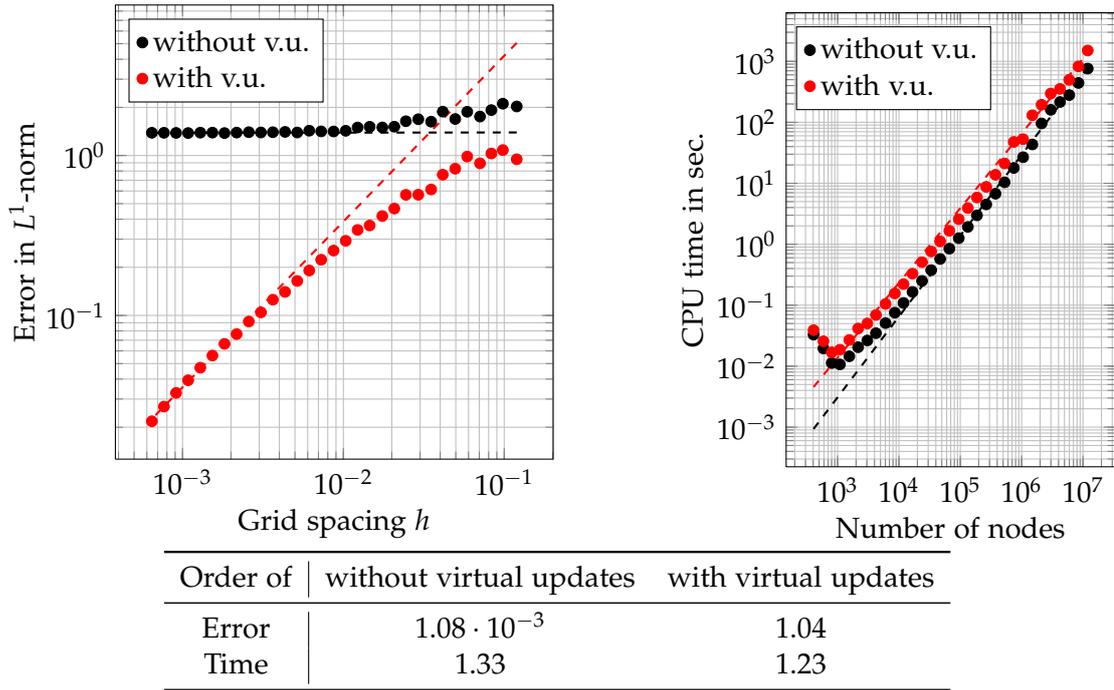


Figure 2.6.4: The approximated error in the L^1 -norm for the example in 2.6.3 with $Y = 10$ at time $t = \pi$.

In figure 2.6.4 we clearly see that without virtual updates there is no convergence. Of course we have some additional effort due to the virtual updates but this is in this example about a factor of 2 and it does not affect the order of time.

More interesting is the figure 2.6.5. Here we display the convergence for different values of Y and for the times $t = \pi$ and $t = 3\pi/2$. For $t = \pi$ we recognise that we have for all three values of Y in essence linear convergence. Thus the method works very well with virtual updates and even better than in example 2.5.2.5. But for $t = 3\pi/2$ we observe that for coarse meshes we have also nearly linear convergence. However, for finer grids we see that the rate of convergence slows down the more the anisotropy

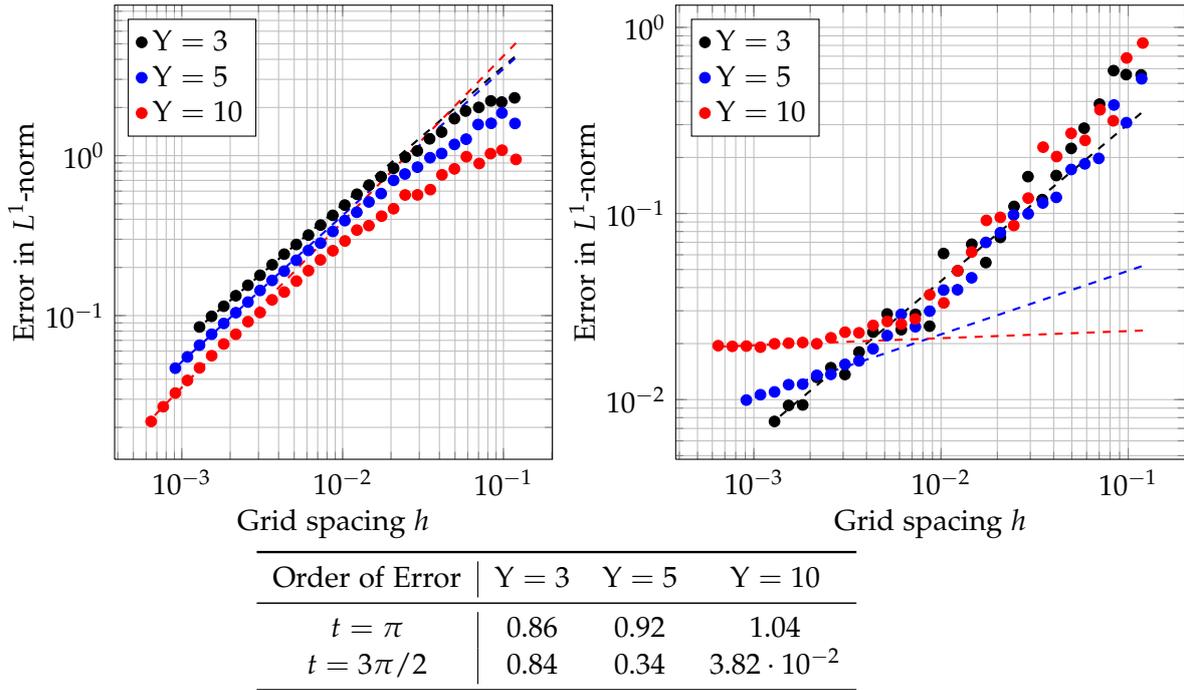


Figure 2.6.5: The approximated error in the L^1 -norm for the example in 2.6.3 at time $t = \pi$ (left) and $t = 3\pi/2$ (right).

Y increases. For $Y = 10$ we have practically no convergence anymore, but for $Y = 3$ we still have linear convergence. Thus the situation is curious. On the one hand using virtual updates we gain very good results, but if the front changes its direction, it seems that the higher the anisotropy the slower the convergence. Also in example 2.5.2.5 the front changes its direction on a small part and this could be the reason for the slower convergence. But there is no theoretical reasoning for this. Furthermore no such effect using virtual updates has been described in the literature.

2.7 Conclusion and further problems

In this section we briefly discuss the results of the preceding sections and discuss some starting points for further work on this topic.

2.7.1 Conclusion

In section 2.2 we recalled the GFMM and the mGFMM where we also corrected the algorithm of mGFMM in remark 2.2.13. We saw in section 2.3 that we can extend the GFMM and the mGFMM to unstructured grids by using the same heuristic arguments. The main steps are to use the right interpretation of neighbourhood and a suitable update formula. In both cases we can recover the comparison principle if the underlying triangulation is acute. The proofs of these results are essentially the same because

they are based only on the notion of neighbourhood and a monotone update scheme. The numerical examples in section 2.5, which are inspired by the ones in [CFFMo8], show that all variants of (m)GFMM and (m)GFT give in essence the same results. The numerical examples for the mGFMM are new too, because there are no numerical results in the literature till now. In a practical and heuristic sense the (m)GFT is therefore a real extension to the (m)GFMM.

The situation is worse if we ask for a proof of convergence. In [CFFMo8] the structure of the regular grid is heavily used in the proof of the convergence theorem and this cannot be transferred to unstructured grids. Some results for the convergence proof of the GFMM can be recovered, see subsection 2.7.6 but it is still an open problem to prove convergence for (m)GFT.

The step from isotropic speed to anisotropic metrics in section 2.6 is straight forward for unstructured grids and it may be of interest in the future. In the GFA there arise bigger theoretic problems and even a comparison principle as in the GFT no longer holds.

2.7.2 Further variants of (m)GFMM

In the (m)GFMM we use a regular grid and the neighbourhood of a node x_I consists of the points x_K with $\|x_I - x_K\|_1 = \Delta x$. These are the centers of the faces of the d -dimensional cube around x_I with edge length $2\Delta x$. If we use instead the corners of this cube as the neighbourhood $V(I)$ then we can use the same description of the algorithm but we must use in another update scheme in step 5 in the GFMM and step 4 in the mGFMM. If we use a consistent and monotone update scheme, e.g. induced by the Hopf-Lax update, then we know that a comparison principle like 2.2.7 for the GFMM and 2.2.16 for the mGFMM will still hold, because the proof of these two results depends on a suitable notion of the neighbourhood and the update scheme, see also the proof of 2.3.5 and 2.4.5. Using this extended neighbourhood we expect that the error will decrease but still have the same order, and on the other hand the computational effort will increase. It should be tested which kind of neighbourhood would be more efficient in practice. Further we expect that this scheme would be still convergent but have to check whether the proof stills applies here.

Another extension would be to use anisotropic metrics in the (m)GFMM like in section 2.6 instead of a scalar speed function. We can regard this method as a special case of the GFA. Using the methods of chapter 3 we can take effort of this special structure and implement the algorithm. Using anisotropic metric on a regular grid would be interesting, for example in the image sciences.

2.7.3 Theoretic results for anisotropic speed

In the example for the GFA we saw that convergence slowed down if we had a high anisotropy and changing directions in the front. In the literature I found no rigorous proof or disproof whether the fast marching methods converges if one uses virtual updates. Thus the algorithms presented are only justified in an heuristic manner and we have no proof whether the method will converge.

2.7.4 Adaptivity

The (m)GFMM and (m)GFT use a static mesh. In the examples 2.5.2.1 and 2.5.2.2 we effectively used only a small portion of the mesh. The greater part of the mesh was not used, so we wasted a lot of memory, but no computational time, because that depends only on the number of grid points we used. For some applications as in image processing it seems to be reasonable to regard the problem in a static mesh. But in other situations it would be useful to have an adaptive setting. We need for an adaptive algorithm on the one hand suitable data structures and on the other hand we need error estimators which we use to decide where we should coarsen or refine the mesh. Using adaptive meshes we have the advantage that we save memory and that we can use a mesh that is as fine as it is necessary to achieve a certain precision.

2.7.5 Applications for (m)GFT

The (m)GFT can be used like the (m)GFMM for several applications. In [FLGG08] one uses the mGFMM for a method for image segmentation. If we have data on an irregular mesh we can directly apply the GFT or the mGFT to this problem. In this case there are no analytic difficulties and one has only to implement this method with the right speed function which also depends on the current front.

Another application for the (m)GFT would be the extension to non-local speed functions like the one presented for the mGFMM in [CFM]. In this work the authors extended the mGFMM to problems with non-local speed functions and they proved therein that the mGFMM would still be convergent. For the numerical computations they used the GFMM instead of the mGFMM. This was probably done to simplify the effort for the implementation, but no reasons are given in this article. In an algorithmic view one can extend the problem to an irregular mesh and we can use mGFT instead of mGFMM. A demanding task is to transfer the computation of the non-local speed which is done in [CFM, Section 8] via Fast Fourier Transform. This application of the FFT for the computation of a convolution explicitly uses the regular structure of the grid and it will be difficult to find a fast method for the computation of convolution on irregular grid, which we have in the (m)GFT. So the problem is not to have any working algorithm but to make an implementation which is efficient.

2.7.6 Convergence of (m)GFT

In this subsection we discuss which parts of the convergence proof in [CFFM08] can be transferred to the GFT and in which parts one explicitly uses the structure of the regular mesh and therefore we cannot use their methods.

We saw in section 2.4 that the construction of the patch P_I^k was not unique. We have there some more degrees of freedom than in the construction of the square cells S_I^k for the (m)GFMM. For the proof of the comparison principle 2.3.5 and 2.4.5 we know that this choice of P_I^k was suitable but it was only heuristically motivated and it is not clear whether we can prove convergence with this construction.

2 Generalized fast marching method

Before we can speak about convergence we define half-relaxed limits of θ^ε :

$$\bar{\theta}^0(t, x) = \limsup_{\varepsilon \rightarrow 0, y \rightarrow x, s \rightarrow t} \theta^\varepsilon(s, y), \quad \underline{\theta}^0(t, x) = \liminf_{\varepsilon \rightarrow 0, y \rightarrow x, s \rightarrow t} \theta^\varepsilon(s, y).$$

This is the same definition as used in [CFFMo8, Section 2.2] and [For09, Section 3].

With this definition we gain for the (m)GFT the same symmetry result as [CFFMo8, Lemma 3.1].

LEMMA 2.7.1 (Symmetry of the (m)GFT algorithm):

We denote by $\bar{\theta}^0[\theta^0, c]$ and $\underline{\theta}^0[\theta^0, c]$ the functions constructed by the (m)GFT algorithm with initial condition θ^0 and velocity c . Then we have

$$\bar{\theta}^0[\theta^0, c] = -\underline{\theta}^0[-\theta^0, -c].$$

The proof is exactly the same as in [CFFMo8, Lemma 3.1].

We use in the following the assumption

(A) The velocity $c \in W^{1,\infty}(\mathbb{R}^2 \times [T, 0])$, for some constant $L > 0$ we have $|c(x', t') - c(x, t)| \leq L/|x' - x| + |t' - t|$, and Ω_0 is a C^2 open set, with bounded boundary $\partial\Omega_0$.

Further we denote by H the maximum diameter of any triangle in the triangulation \mathcal{T} .

In the following we show that we can recover the lemmata 4.1, 4.2 and 4.3 of [CFFMo8]. The proofs of these results for the (m)GFT are mainly the same as for the (m)GFMM. This is because one uses mostly ideas that do not explicitly use structure of the regular grid. We only show the results for the GFT. For the mGFT we would get analogous results as for the mGFMM and the differences are mainly of notational kind.

First we gain for the GFT the analogous result of [CFFMo8, Lemma 4.1]

LEMMA 2.7.2 (time character of \hat{u}):

Assume that there exists $\delta > 0$ and $(I, n) \in \mathcal{N} \times \mathbb{N}$ such that $c(t_n, x_I) \geq \delta > 0$, $\theta_I^{n-1} = -1$, and $\theta_I^n = 1$ (resp., $c(t_n, x_I) \leq \delta < 0$, $\theta_I^{n-1} = 1$, and $\theta_I^n = -1$) then for any $K \in V(I) \cap F_+^{n-1}$ (resp. $K \in V(I) \cap F_-^{n-1}$), we have for $H \leq \frac{\delta^2}{16L}$

$$\hat{u}_{+,K}^{n-1} = \sup\{t_m \leq t_{n-1}, \theta_K^{m-1} = -1, \theta_K^p = 1 \text{ for } m \leq p \leq n-1\} > t_n - \frac{4H}{\delta},$$

with the convention that $\hat{u}_{+,K}^{n-1} = 0$ if $\theta_K^p = 1$ for $0 \leq p \leq n-1$

(resp., $\hat{u}_{-,J}^{n-1} = \sup\{t_m \leq t_{n-1}, \theta_K^{m-1} = 1, \theta_K^p = -1 \text{ for } m \leq p \leq n-1\} > t_n - \frac{4H}{\delta}$

with the convention that $\hat{u}_{+,K}^{n-1} = 0$ if $\theta_K^p = -1$ for $0 \leq p \leq n-1$).

The proof is essentially the same as in [CFFMo8, Lemma 4.1]. We changed the index J to K and we have to regard the maximum diameter H of the triangulation instead of the grid spacing Δx . We have only to change the estimate of $\hat{u}_I^{n-p-1} - \hat{u}_K^{n-p-1} \leq \frac{2\Delta x}{\delta}$. Due to the Hopf-Lax update we know that

$$\hat{u}_I^{n-p-1} - \hat{u}_K^{n-p-1} \leq H \frac{1}{\hat{c}_K^{n-p-1}} \leq \frac{2H}{\delta}.$$

The rest of the proof is exactly the same as in [CFFMo8].

Now we state that also [CFFMo8, Lemma 4.2] works in the setting of GFT.

LEMMA 2.7.3 (Error estimate between t_n and \tilde{t}_n):

Assume that there exists $I \in NA^n$ such that $|\hat{c}_I^{n-1}| \geq \delta > 0$. Then the following estimate holds:

$$(t - \tilde{t}_n)^+ \leq \frac{2L}{\delta^2} H \Delta t \text{ if } \Delta t \leq \frac{\delta}{2L}$$

The proof differs at some points from the proof in [CFFMo8], so we give here the full proof.

Proof. We treat only the case $c_I^{n-1} \geq \Delta > 0$. The other case is similar.

Assume that $\tilde{t}_n < t_n$ (otherwise the result is obviously true). Then we know that $t_n = t_{n-1}$. We define $p > 0$ such that $t_{n-p-1} < t_{n-p} = \dots = t_{n-1} = t_n$. In particular we have

$$t_{n-p} \leq \tilde{t}_{n-p} \leq \tilde{u}_K^{n-p-1} \quad \forall K \in F^{n-p-1}$$

and

$$\tilde{t}_n = \tilde{u}_I^{n-1} \leq \tilde{u}_K^{n-1} \quad \forall K \in F^{n-1}.$$

Now we show that $I \in F_-^{n-p-1}$. We assume by contradiction that $I \notin F_-^{n-p-1}$. By using that $\theta_I^{n-p-1} = -1$ (since $\hat{c}_I > 0$), we deduce that, for all $K \in V(I) \cap F_+^{n-1}$, we have $\theta_K^{n-p-1} = -1$ and thus we have $\hat{u}_{+,K}^{n-1} = t_n$. This implies that also the node K has been accepted at the physical time t_n , which gives us that $\tilde{u}_I^{n-1} > t_n$, and this is a contradiction.

Further we know due to $t_{n-p} - t_{n-p-1} \leq \Delta t$ that $\hat{c}_I^{n-p-1} \geq \frac{\delta}{2}$ for $\Delta t \leq \frac{\delta}{2L}$.

We compare $\hat{u}_{+,K}^{n-1}$ and $\hat{u}_{+,K}^{n-p-1}$ for $K \in V(I) \cap F_+^{n-1}$. If $K \notin F_+^{n-p-1}$, then u_K changes its value during the iterations $n-p \leq m \leq n-1$, and for such m we have $\hat{u}_K^{n-1} = u_K^m = t_m = t_n$. Since $\tilde{t}_n < t_n$ we know that the node $J \in V(I)$ does not contribute to the Hopf-Lax update of node I .

Thus we have $\tilde{t}_n = \tilde{u}_I^{n-1} = (\Lambda \hat{u}^{n-1})(I) = (\Lambda \hat{u}^{n-p-1})(I)$ with speed \hat{c}_I^{n-1} and $\tilde{u}_I^{n-p-1} = (\Lambda \hat{u}^{n-p-1})(I)$ with speed \hat{c}_I^{n-p-1} .

Now we get the estimate

$$\begin{aligned} t_n - \tilde{t}_n &\leq \tilde{u}_I^{n-p-1} - \tilde{u}_I^{n-1} \\ &\leq H \left(\frac{1}{|\hat{c}_I^{n-1}|} - \frac{1}{|\hat{c}_I^{n-p-1}|} \right) \leq H \frac{|\hat{c}_I^{n-p-1}| - |\hat{c}_I^{n-1}|}{|\hat{c}_I^{n-p-1}| |\hat{c}_I^{n-1}|} \\ &\leq H \frac{|\partial_t c|_\infty |t_{n-1} - t_{n-p-1}|}{|\hat{c}_I^{n-p-1}| |\hat{c}_I^{n-1}|} \leq \frac{2L}{\delta^2} H \Delta t. \end{aligned}$$

□

Even [CFFMo8, Lemma 4.3] holds in the same form:

LEMMA 2.7.4 (Separation of the phases of θ^ε by the level set of a test function):

Let $\phi \in \mathcal{C}^2$ in a neighbourhood of V of (t_0, x_0) such that $\phi_t(t_0, x_0) > 0$ (resp., $\phi_t(t_0, x_0) < 0$). There exists $\delta_0 > 0$, $r > 0$, $\tau > 0$ such that, if $\max_{\bar{V}}((\theta^\varepsilon)^* - \phi)$ is reached at $(t_\varepsilon, x_\varepsilon) \in B((t_0, x_0), \delta_0) \subset V$ with $(\theta^\varepsilon)^*(t_\varepsilon, x_\varepsilon) = 1$, then there exists $\Psi_\varepsilon \in \mathcal{C}^2(B(x_0, r), (t_0 - \tau, t_0 + \tau))$ such that

2 Generalized fast marching method

(i) for all $(t_m, x_K) \in Q_{\tau,r}(t_0, x_0) = (t_0 - \tau, t_0 + \tau) \times B(x_0, r_0)$

$$\theta^\varepsilon(t_m, x_k) = 1 \Rightarrow t_m \geq \Psi_\varepsilon(x_K) \text{ (resp., } t_m \leq \Psi_\varepsilon(x_K)\text{)}.$$

(ii) There exists $(I, n) \in \mathcal{N} \times \mathbb{N}$ such that $(t_\varepsilon, x_\varepsilon) \in \bar{Q}_I^n = [t_{k_n}, t_{k_{n+1}}] \times \bar{P}_I$, $(\theta)^\ast(t_{k_n}, x_I) = 1$, $t_{k_n} = \Psi_\varepsilon(x_I)$, and $\theta^{\bar{n}} = 1$, $\theta_I^m = -1$, $m_0 \leq m \leq \bar{b} - 1$ (resp. $\theta^{\bar{n}} = -1$, $\theta_I^m = 1$, $m_0 \leq m \leq \bar{n} - 1$), where $\bar{n} = \inf\{k : k_n \leq k \leq k_{n+1} - 1, \theta_I^k = 1 \text{ (resp. } \theta_I^k = -1)\}$ and $m_0 = \inf\{m : t_m \geq t_0 - \tau\}$.

(iii) The following Taylor expansion holds:

$$\Psi_\varepsilon(x_K) = \Psi_\varepsilon(x_I) - \frac{D\phi(t_0, x_0)}{\phi_t(t_0, x_0)}(x_K - x_I) + H\mathcal{O}(H + |x_I - x_0| + |t_{k_n} - t_0|).$$

(iv) If $\phi_t(t_0, x_0) < 0$, then for all $(t_{k_n}, x_K) \in Q_{\tau,r}(t_0, x_0) = (t_0 - \tau, t_0 + \tau) \times B(x_0, r_0)$

$$\theta^\varepsilon(t_{k_{n-1}}, x_k) = 1 \text{ and } \theta^\varepsilon(t_{k_n}, x_k) = -1 \Rightarrow t_{k_n} \leq \Psi_\varepsilon(x_K).$$

The proof is exactly the same as in [CFFMo8]. We have only to set $Q_K^{n-1} =]t_{k_n}, t_{k_{n+1}}[\times \text{int}(P_K)$ instead of $Q_K^{n-1} =]x_K + \Delta x[\times]t_{k_n}, t_{k_{n+1}}[$. Further we replace the grid spacing Δx by the maximum diameter H of the triangulation.

In the lemmata 4.4, 4.5 and 4.6 and also in the proof of the convergence theorem [CFFMo8, Theorem 2.5] the structure of the regular grid is used extensively and thus I could not transfer these results to the setting of the GFT.

Nevertheless the (m)GFT is based on the same heuristic arguments as the (m)GFMM and thus it seems to be reasonable that also the (m)GFT converges under the same conditions as the (m)GFMM. Thus I state here the conjecture of convergence for (m)GFT:

CONJECTURE 2.7.5:

Under assumption (A), $\bar{\theta}^0$ (resp. $\underline{\theta}^0$) is a viscosity subsolution (resp., supersolution) of (2.0.1). In particular, if (2.0.1) satisfies a comparison principle, then $\bar{\theta}^0 = (\underline{\theta}^0)^\ast$ and $(\bar{\theta}^0)_\ast = \underline{\theta}^0$ is the unique discontinuous viscosity solution of (2.0.1).

3 Anisotropic fast marching on Cartesian grids

3.1 Introduction

In many applications, such as image science, one has to solve the anisotropic eikonal equation, namely

$$\|Du(x)\|_{M(x)} = 1 \text{ on } \Omega, \quad u|_{\partial\Omega} = 0. \quad (3.1.1)$$

where $M : \Omega \rightarrow \mathcal{S}^2$ is a continuous mapping and \mathcal{S}^2 denotes the set of symmetric positive definite 2×2 -matrices.

Some algorithms are known in the literature. The ordered upwind method (OUM) was introduced in [SV03]; it can solve general static Hamilton-Jacobi equations. In his PhD-Thesis [Lino3] Lin adapted – at least theoretically – the OUM for the anisotropic eikonal equation to regular grids, but no numerical results for the OUM are given. Lin also introduced a kind of anisotropic fast marching; the resulting algorithm, however, does not converge to the analytic solution [Lino3, Section 8.3.2]: he uses a consistent discretization, but the plain fast marching scheme is not suitable in this case. We will see in section 3.3, why the ansatz of Lin cannot work in general. Further, we list the cases in which Lin’s algorithm will work.

Spira and Kimmel proposed in [SK04] a variant of the fast marching method that computes the solution of the eikonal equation on a parametric manifold. Their methods are very similar to the anisotropic fast marching method that will be proposed in section 3.2, where we will also discuss the differences between these two methods. Unfortunately, Spira and Kimmel did not recognize that they had almost solved the problem for the anisotropic eikonal equation.

In [BR06] an adaptive Gauss-Seidel method was proposed to solve (3.1.1) on unstructured meshes. Further, [Ras07] presented a fast marching variant of this method. We will mainly use their ideas for constructing an anisotropic fast marching method.

In [KSC⁺07] and [SKD⁺07] a recursive anisotropic FMM, respectively an anisotropic FMM is proposed. Despite the name “fast marching method” this method is not a single-pass method as would be characteristic for fast marching methods. The authors describe an adaptive Gauss-Seidel method with two priority queues instead of one FIFO-queue as in [BR06]. In section 3.4 we will also compare this method and give numeric results.

In [WDB⁺08] an algorithm to approximate distances maps on parametric surfaces is proposed using ansatz different from [SK04]. They use a massively parallel algorithm and sweeping strategies to gain, in total, an $\mathcal{O}(N)$ algorithm if the parametrization of the surface satisfies suitable conditions.

3.2 The AFM algorithm

In this section we will construct an anisotropic fast marching method (AFM). In the first subsection 3.2.1 we give a heuristic motivation of the algorithm. After this we will state the AFM in subsection 3.2.2 and in the following subsection 3.2.3 we show how we can construct the needed virtual triangles. In subsection 3.2.4 and 3.2.5 we give some estimates for the virtual triangles. In the last subsection 3.2.6 we discuss the differences to the method in [SKo4].

3.2.1 Motivation

There are two different methods to construct a convergent method that solves the anisotropic eikonal equation. The first one is the ordered upwind method. This method was introduced in [SV03] for triangulated meshes and can of course also be used on Cartesian grids. The key point of the OUM is that one constructs a narrow-band which is large enough to contain the domain of numerical dependency. I think this method is the most sophisticated and one obtains the best result with optimal numerical complexity, but it is hard to understand and to implement.

The second way to solve this equation is to use iterative methods e.g. Gauß-Seidel methods. Using Gauß-Seidel there is no need to care about the domain of numerical dependency, because by iterating one implicitly construct the right domain of dependency. The advantage of Gauß-Seidel is that it is easy to understand and it is also easy to implement. There are many different special cases of this Gauß-Seidel iteration but all methods share the property that they are in essence iterative methods. The problem of all these methods is, that one has in general a complexity of at least $\mathcal{O}(N^{3/2})$.

In the isotropic case the fast marching method is a hybrid of these two ways. On the one hand fast marching is a very special Gauß-Seidel-method, on the other hand it is a narrow-band method like the ordered upwind method. The causality is the main property in the fast marching method. With this it is possible to construct a single pass algorithm and there is no need to iterate. In the case of the anisotropic eikonal equation we have to take care of the causality of the fast marching method and this is done by constructing virtual triangles and the usage of a consistent update formula.

3.2.2 Description of the AFM algorithm

The AFM algorithm is a pure fast marching algorithm. So the structure is like the one in [Set99a, chapter 8]. The only change is in step (4) due to the concrete formulation of the update formula.

- (1) Begin loop: Let A be the *Trial* point with the smallest u value.
- (2) Add the point A to *Known*; remove it from *Trial*.
- (3) Tag all neighbours of A that are not *Known* as *Trial*. If the neighbour is in *Far*, remove, and add to the set *Trial*.
- (4) Recompute the values of u at all *Trial* neighbours of A .

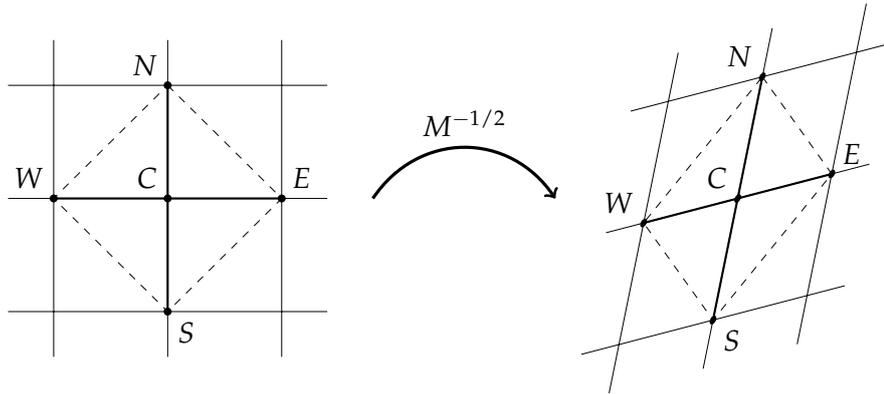


Figure 3.2.1: Transformation of the grid by a symmetric positive definite matrix $M^{-1/2}$. Squares become to parallelograms. The transformed triangles CSE and CNW on the right side are obtuse.

(5) Return to top of loop.

We describe now the update formula for the 4-point neighbourhood. The the case of the 8-point neighbourhood can be handled in the completely same way. Using the 4-point neighbourhood a node C has four neighbours, namely N , E , S and W . These neighbours induce a triangle patch around C with the four triangles CEN , CSE , CWS and CNW . For this triangle patch we use the Hopf-Lax update for the anisotropic eikonal equation introduced in [BR06, Section 7], see also subsection 2.1.2. The main idea we use here is that the generalized eikonal equation (3.1.1) is solved by locally transforming the mesh and solving the isotropic eikonal equation in the transformed mesh. In figure 3.2.1 such a transformation is displayed.

We may use this update scheme only if the transformed triangles are acute. If we have to update an obtuse triangle we use the idea of virtual triangles and split the obtuse triangle into two acute ones. The construction of suitable virtual triangle is described in the following subsection 3.2.3.

3.2.3 Finding virtual triangles

For this subsection we regard a triangle CAB in the transformed mesh that has an obtuse angle at node C , see for example figure 3.2.2. Furthermore, we assume that C lies at the origin of our coordinate system and we identify the nodes with the vectors connecting them from the origin.

We search a node P of the mesh in the splitting section such that the two triangles CAP and CPB are acute. Hereby the splitting section is defined as the set of points x such that the angles $\angle(ACx)$ and $\angle(xCB)$ are acute. We know that all grid points have the form $\alpha A + \beta B$ with $\alpha, \beta \in \mathbb{Z}$ because we started with a Cartesian grid and transformed the whole grid by $M^{-1/2}$.

The conditions that the triangles CAP and CPB have to be acute at C give us with

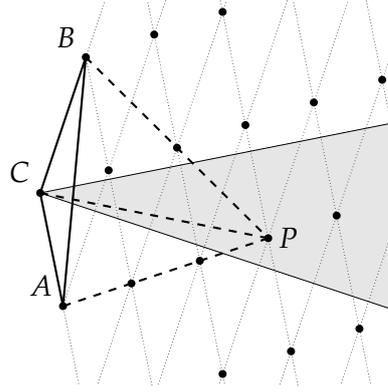


Figure 3.2.2: The obtuse triangle CAB is split into the two acute triangles CAP and CPB . Here is $P = 4 \cdot A + 3 \cdot B$.

$P = \alpha A + \beta B$ the conditions for α and β :

$$\begin{aligned}\langle A, P \rangle &= |A|^2 \alpha + \langle B, A \rangle \beta \geq 0 \\ \langle B, P \rangle &= \langle A, B \rangle \alpha + |B|^2 \beta \geq 0.\end{aligned}$$

Furthermore it is reasonable to enforce that α and β are positive and that $|P|$ should be as small as possible.

So we end up with the following nonlinear integer programming problem for $P = \alpha A + \beta B$:

$$\min |P|$$

with the restrictions

$$\begin{aligned}\langle A, P \rangle &= |A|^2 \alpha + \langle B, A \rangle \beta \geq 0, \\ \langle B, P \rangle &= \langle A, B \rangle \alpha + |B|^2 \beta \geq 0, \\ \alpha, \beta &\in \mathbb{N}^+.\end{aligned}\tag{3.2.1}$$

To find a suitable point P would be difficult because P is determined by a nonlinear integer programming problem. To simplify the problem we weaken the assumption on P and we do not minimize $|P|$ anymore, but we minimize $\langle A^\perp, P \rangle = \langle A^\perp, B \rangle \beta$ instead. Hereby A^\perp denotes a normalized vector that is perpendicular to A such that $\angle(ACA^\perp) = \pi/2$.

Thus we get the following linear integer programming problem:

$$\min \langle A^\perp, P \rangle$$

with the restrictions

$$\begin{aligned}\langle A, P \rangle &= |A|^2 \alpha + \langle B, A \rangle \beta \geq 0, \\ \langle B, P \rangle &= \langle A, B \rangle \alpha + |B|^2 \beta \geq 0, \\ \alpha, \beta &\in \mathbb{N}^+.\end{aligned}$$

Due to the assumption on the triangle CAB we have $\langle A^\perp, P \rangle = \beta \langle A^\perp, B \rangle > 0$. Thus it suffices to minimize β . Rewriting the conditions of the angles (3.2.1) we get

$$\frac{-\langle A, B \rangle}{|A|^2} \beta \leq \alpha \leq \frac{-|B|^2}{\langle A, B \rangle} \beta.$$

This problem can be solved very fast because we just carry out a linear search increasing β .

Thus the construction of suitable virtual triangles is done by solving the following linear integer programming problem:

Find minimal $\beta \in \mathbb{N}^+$ such that the interval

$$\left[\frac{-\langle A, B \rangle}{|A|^2}; \frac{-|B|^2}{\langle B, A \rangle} \right] \beta \quad (3.2.2)$$

contains a natural number α .

It may happen that the interval in (3.2.2) contains more than one natural number. In this case we just take the smallest one. Therefore we know that if this interval contains a natural number, then we have $\alpha = \left\lceil \frac{-\langle A, B \rangle}{|A|^2} \beta \right\rceil$.

At the end of this section we give two remarks. The simplified problem (3.2.2) still constructs a point in the splitting section as it is needed for the causality property of the fast marching algorithm, but we cannot guarantee that we have found a point that is as close as possible to C . The second remark is that we could also minimize $\langle B^\perp, P \rangle = \alpha \langle B^\perp, A \rangle$ with $\angle(B^\perp CB) = \pi/2$ instead of $\langle A^\perp, P \rangle = \beta \langle A^\perp, B \rangle$. This choice was just made by chance because there is no canonical choice for this.

3.2.4 Estimates for virtual triangles in the 4-point neighbourhood

In this subsection we derive some estimates for α and β in the 4-point neighbourhood. To do this, we regard the triangle CEN with the coordinates $C = (0, 0)^T$, $E = (1, 0)^T$ and $N = (0, 1)^T$. Furthermore we have a symmetric positive definite matrix V which plays the role of $M^{-1/2}$. We assume that V has eigenvalues $\lambda_1 = Y > 1$, $\lambda_2 = 1$ and eigenvectors $v_1 = (\cos \phi, \sin \phi)$ and $v_2 = (\sin \phi, -\cos \phi)$ with $0 \leq \phi < \pi$. Thus the matrix V is

$$V = \begin{pmatrix} Y \cos^2 \phi + \sin^2 \phi & (Y - 1) \cos \phi \sin \phi \\ (Y - 1) \cos \phi \sin \phi & \cos^2 \phi + Y \sin^2 \phi \end{pmatrix}.$$

For the transformed points we get

$$\begin{aligned} V \cdot C &= (0, 0), \\ A = V \cdot E &= \begin{pmatrix} Y \cos^2 \phi + \sin^2 \phi \\ (Y - 1) \cos \phi \sin \phi \end{pmatrix}, \\ B = V \cdot N &= \begin{pmatrix} (Y - 1) \cos \phi \sin \phi \\ \cos^2 \phi + Y \sin^2 \phi \end{pmatrix}. \end{aligned}$$

Thus we get

$$|A|^2 = Y^2 \cos^2 \phi + \sin^2 \phi,$$

$$\begin{aligned} |B|^2 &= (1 + Y^2 + (1 - Y^2) \cos(2\phi)) / 2, \\ \langle A, B \rangle &= (Y^2 - 1) \cos \phi \sin \phi. \end{aligned} \quad (3.2.3)$$

One can see that $\langle A, B \rangle < 0$ for $\pi/2 < \phi < \pi$. Thus for $\pi/2 < \phi < \pi$ we have a obtuse triangle and we will regard this case for the rest of this subsection. The lower and upper bounds for the interval in (3.2.2) are

$$\begin{aligned} \frac{-\langle A, B \rangle}{|A|^2} &= \frac{(1 - Y^2) \cos \phi \sin \phi}{Y^2 \cos^2 \phi + \sin^2 \phi}, \\ \frac{-|B|^2}{\langle B, A \rangle} &= \frac{\cot \phi + Y^2 \tan \phi}{1 - Y^2}. \end{aligned}$$

Finally we get for the length of the interval

$$l := \left| \left[\frac{-\langle A, B \rangle}{|A|^2}; \frac{-|B|^2}{\langle B, A \rangle} \right] \right| = \frac{2Y^2 \csc(2\phi)}{(1 - Y^2)(Y^2 \cos^2 \phi + \sin^2 \phi)}.$$

We know that this interval contains a natural number if its length is at least 1. Thus we know, that $\beta \leq \lceil l^{-1} \rceil$. A short computation with Mathematica shows that the maximum of l^{-1} is attained at

$$\arccos \left(-\sqrt{2 / \left(5 - 3Y^2 + 9\sqrt{9 - 14Y^2 + 9Y^4} \right)} \right) \rightarrow \frac{5\pi}{6} \quad \text{for } Y \rightarrow \infty$$

and we gain

$$\beta \lesssim \frac{\sqrt{3}(Y^2 - 1)(1 + 3Y^2)}{16Y^2} = \mathcal{O}(Y^2).$$

With equation (3.2.2) we get for α the bound (we left the ceiling operation for simplicity)

$$\alpha \leq \left\lceil \left[\lceil l^{-1} \rceil \frac{\langle A, B \rangle}{-|A|^2} \right] \right\rceil \approx \frac{(Y^2 - 1)^2 \sin^2(2\phi)}{4Y^2} = \mathcal{O}(Y^2).$$

We carried out some numerical experiments to validate whether these boundaries are sharp or not. For this we compute the values for α and β explicitly and compare the results with the analytic estimates.

We see in figures 3.2.3 and 3.2.4 that the analytic estimates for α and β are sharp, but we assume that the coefficients are smaller in the average case.

To determine the average complexity for the calculation of α and β we do the following numerical experiment. We compute the values of α and β for the angle $\pi/2 < \phi < \pi$ at 10000 equidistant sampled points and a given anisotropy coefficient Y . Afterwards we just calculate the mean of α and β and divide this mean by Y again.

In figure 3.2.5 we see how this ratio depends on the anisotropy coefficient Y . This figure leads to the conjecture that the average complexity to determine a suitable virtual triangle is $\mathcal{O}(Y)$ instead of $\mathcal{O}(Y^2)$ for the worst-case analytic estimate.

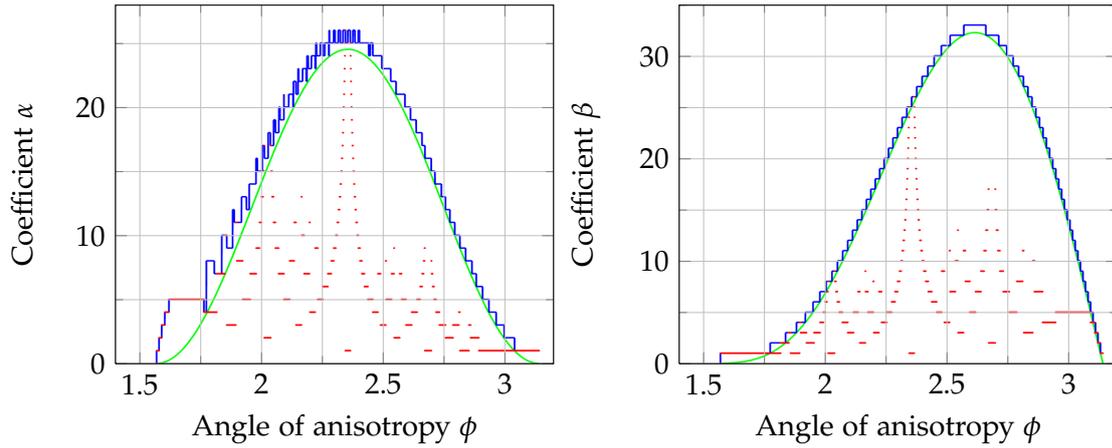


Figure 3.2.3: Computed values (red) for α (left) and β (right) compared with the exact boundaries (blue) and boundaries without ceiling operation (green). We use an anisotropy $Y = 10$ and sample the angle $\pi/2 < \phi < \pi$ at 100000 equidistant points.

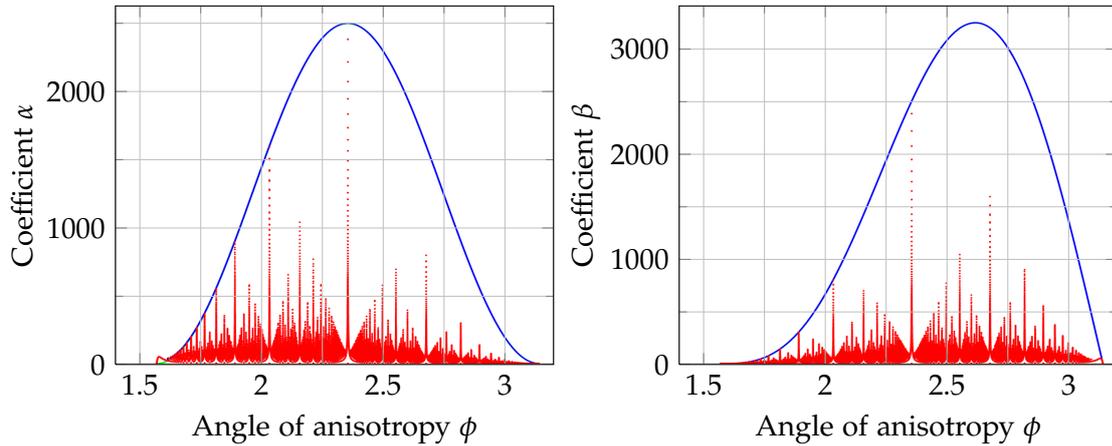


Figure 3.2.4: The same experiment as in figure 3.2.3 but with anisotropy $Y = 100$.

3.2.5 Estimates for virtual triangles in the 8-point neighbourhood

In this subsection we do the same analysis for the 8-point neighbourhood using the same techniques as in the case of the 4-point neighbourhood. Let us regard the triangle $C(NE)N$ with the coordinates $C = (0,0)^T$, $NE = (1,1)^T$ and $N = (0,1)^T$ and we use the same matrix V as before.

Now we get for the transformed points

$$\begin{aligned} V \cdot C &= (0,0) \\ A = V \cdot (NE) &= \begin{pmatrix} Y \cos^2 \phi + \sin^2 \phi + (Y-1) \cos \phi \sin \phi \\ (Y-1) \cos \phi \sin \phi + \cos^2 \phi + Y \sin^2 \phi \end{pmatrix} \end{aligned}$$

3 Anisotropic fast marching on Cartesian grids

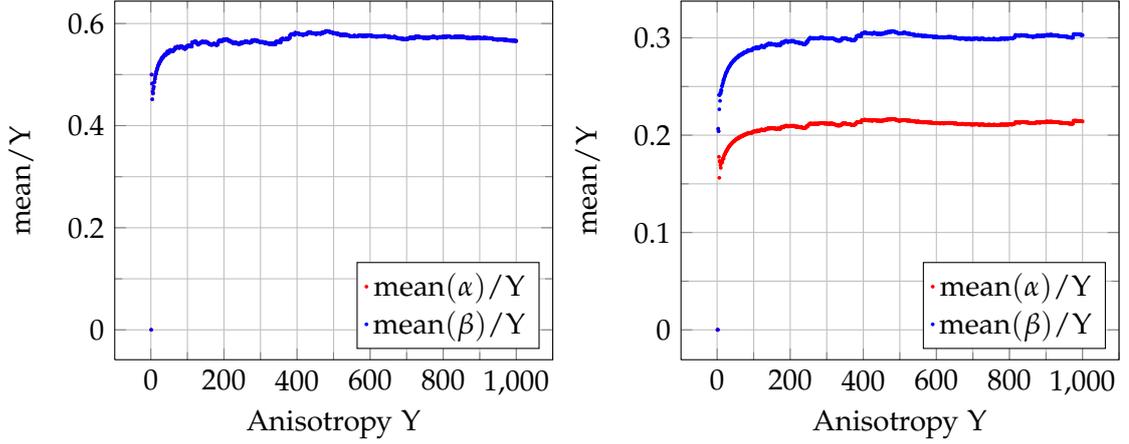


Figure 3.2.5: Mean of α (resp. β) divided by the anisotropy coefficient Y plotted against Y . In the left figure we regard the 4-point neighbourhood and the angle $\pi/2 < \phi < \pi$ was sampled at 10000 equidistant points. The mean of α and β is the same, thus we see only one of them. In the right plot we regard the same but in the 8-point neighbourhood and the angle $3\pi/4 < \phi < \pi$, see also subsection 3.2.5.

$$B = V \cdot N = \begin{pmatrix} (Y-1) \cos \phi \sin \phi \\ \cos^2 \phi + Y \sin^2 \phi \end{pmatrix} \quad (3.2.4)$$

Thus we have

$$\begin{aligned} |A|^2 &= (Y^2 - 1) \sin(2\phi) + Y^2 + 1 \\ |B|^2 &= ((1 - Y^2) \cos(2\phi) + Y^2 + 1) / 2 \\ \langle A, B \rangle &= ((Y^2 - 1) (\sin(2\phi) - \cos(2\phi)) + Y^2 + 1) / 2. \end{aligned} \quad (3.2.5)$$

One can verify that $\langle A, B \rangle > 0$ for $1 \leq Y < 1 + \sqrt{2}$. Further we have $\langle A, B \rangle > 0$ for $0 < \phi < 3/4\pi$. Thus we will assume $\langle A, B \rangle < 0$ for the rest of this subsection which implies that $3\pi/4 < \phi < \pi$ and $Y > 1 + \sqrt{2}$. The lower and upper bounds for the interval are

$$\begin{aligned} \frac{-\langle A, B \rangle}{|A|^2} &= \left(\left(\frac{(Y^2 + 1) \sec(2\phi)}{Y^2 - 1} + \tan(2\phi) \right)^{-1} - 1 \right) / 2 \\ \frac{-|B|^2}{\langle B, A \rangle} &= - \left(\frac{(Y^2 - 1) \sin(\phi) \cos(\phi)}{Y^2 \sin^2(\phi) + \cos^2(\phi)} + 1 \right)^{-1}. \end{aligned} \quad (3.2.6)$$

Finally we get for the length of the interval

$$l := \left| \left[\frac{-\langle A, B \rangle}{|A|^2}; \frac{-|B|^2}{\langle B, A \rangle} \right] \right|$$

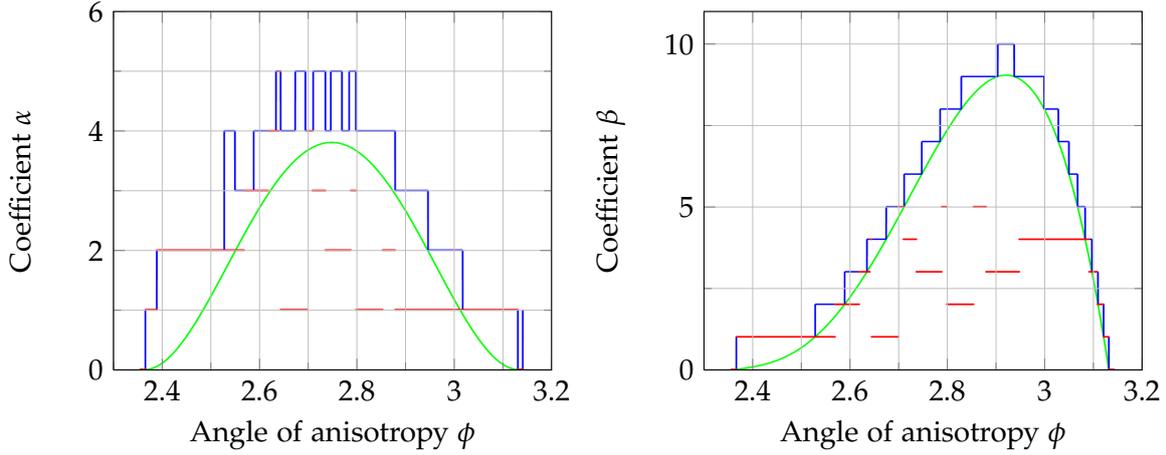


Figure 3.2.6: Computed values (red) for α (left) and β (right) compared with the exact boundaries (blue) and boundaries without ceiling operation (green). We use an anisotropy $Y = 10$ and sample the angle $3\pi/4 < \phi < \pi$ at 100000 equidistant points.

$$= -\frac{2Y^2}{((Y^2 - 1) \sin(2\phi) + Y^2 + 1) ((Y^2 - 1) (\sin(2\phi) - \cos(2\phi)) + Y^2 + 1)}.$$

We can compute with Mathematica the angle ϕ^* which maximizes β . We omit the analytic formula but a further use of Mathematica gives us

$$\lim_{Y \rightarrow \infty} \phi^* = \pi/2 + \arctan(2 + \sqrt{7})$$

which can also be verified by numeric experiments. Thus we get the boundary for β

$$\beta \lesssim \frac{(7\sqrt{7} - 17) Y^4 - 2(1 + \sqrt{7}) Y^2 - 5\sqrt{7} - 13}{16Y^2} = \mathcal{O}(Y^2).$$

For α we get the bound (again without ceiling operation)

$$\alpha \leq \left\lceil \left[l^{-1} \frac{-\langle A, B \rangle}{|A|^2} \right] \right\rceil \approx \frac{((Y^2 - 1) (\sin(2\phi) - \cos(2\phi)) + Y^2 + 1)^2}{4Y^2} = \mathcal{O}(Y^2).$$

As in the case of the 4-point neighbourhood we carried out some numerical experiments to see whether these boundaries are sharp or not. As above we computed the values for α and β explicitly and compared the results with the analytic estimates.

We see in figures 3.2.6 and 3.2.7 that the analytic estimates for α and β are sharp, but we expect that in the average case we the coefficients α and β are smaller. An experiment for the average case is displayed in figure 3.2.5.

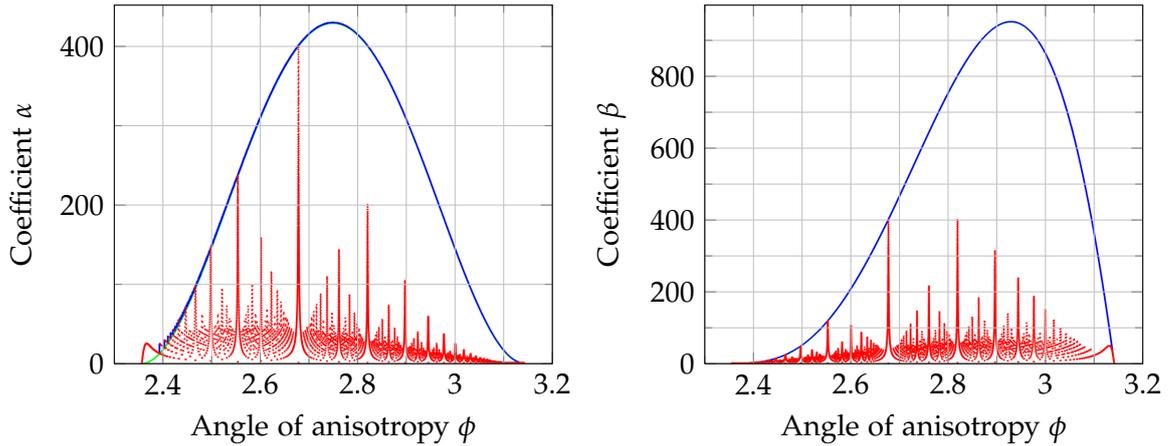


Figure 3.2.7: Same experiment as in figure 3.2.6 but with anisotropy $\Upsilon = 100$.

3.2.6 Differences to [SK04]

In [SK04] the authors proposed a fast marching method to solve the eikonal equation on a 2-dimensional parametric manifold. The main idea is that they transport the Cartesian grid on \mathbb{R}^2 to the manifold. The transported grid is in general non-orthogonal and they use also the idea of virtual updates to overcome the problems of obtuse triangles. They end up in equation (10) with the same problem as in (3.2.2) but it is not mentioned and derived that the constructed point is minimal due to the projection to A^\perp . The algorithm to calculate n and m (resp. α and β) differs in its form, but is in essence the same. Due to their ansatz with the metric tensor they could not give as detailed an analysis for the dependency of the virtual triangles.

In main idea in the AFM is, that we solve the anisotropic eikonal equation by locally transforming the grid and solving the ordinary eikonal equation in the transformed grid. Thus this idea applies for every metric given by a mapping $M : \Omega \rightarrow \mathcal{S}^2$, especially for M induced by the metric tensor of a manifold. In [SK04] they use always the point of view to solve the eikonal equation on the manifold and they lift up the update scheme to the manifold. Thus one does not see that the same idea can also work with the more general setting of any metric $M : \Omega \rightarrow \mathcal{S}^2$.

3.3 Restrictions for the plain fast marching method on cartesian grids

A straight forward approach for the fast marching method is to transform the update formula to get a consistent update scheme. In [Lino3] this was done and Lin recognized that this approach does not work in general. Lin also saw that the reason for this was the causality property, which was not fulfilled anymore. But no closer analysis was given and also no other suitable extension for the fast marching method.

In [JBT⁺08] the authors applied the idea of [Lino3] in \mathbb{R}^3 . They developed an exact update scheme, but only used the plain fast marching method. Thus their algorithm

cannot converge in general. In the 3-dimensional setting one has to split obtuse simplexes into acute one. This could be possible by using the same ideas as in the construction of virtual triangles, but no such methods has been presented.

The causality of the update scheme is an important principle for constructing the fast marching method. Further we know that the causality property holds if and only if the underlying triangulation is non-obtuse. With the preceding section we are able to list the cases in which the plain fast marching method still converges for anisotropic metrics.

In the case of the 4-point neighbourhood we see by equation (3.2.3) that the corresponding triangle are non-obtuse if and only if the angle of anisotropy is a multiple of $\pi/4$. Thus we have to use virtual updates if the anisotropy does not point along the edges of the grid.

The case of the 8-point neighbourhood is more complicated. Like above, we see with equation (3.2.5) that we need no virtual updates if the angle of the anisotropy is a multiple of $\pi/4$. Let $V = M^{-1/2}$ with M from (3.1.1) and $\lambda_1 \geq \lambda_2$ the eigenvalues of M . Then we know by equation (3.2.5) that in the case $\lambda_1/\lambda_2 \leq \sqrt{1 + \sqrt{2}}$ that no obtuse angle appear and we need no virtual updates. If the anisotropy is bigger, it depends on the angle ϕ , whether obtuse angle appears or not. Furthermore in the 8-point neighbourhood the coefficients α and β are always smaller than in the 4-point neighbourhood.

3.4 Numerical tests

In this section we present some numerical tests for the different methods. We abbreviate the anisotropic fast marching using the 4-point neighbourhood by AFM₄ and using the 8-point neighbourhood by AFM₈. The usual fast marching method with a consistent update scheme but without virtual update is called FMM₄ or FMM₈ depending on the underlying neighbourhood. We abbreviate the adaptive Gauß-Seidel methods by AGS and the method introduced in [KSC⁺07] by 2QGS (double queued Gauß-Seidel).

3.4.1 Implementational issues

We give here some comments on the implementation of the different methods.

In practical computations the grid is bounded and therefore we cannot ensure that we find a suitable virtual triangle. If we determine a point P in the construction of virtual triangles which is outside the computational domain we just project it to the boundary of the domain and are no longer concerned whether the triangles induced by the projected point are acute.

Another issue is that due to the symmetry we only have to check one triangle in the 4-point stencil. In the case of the 8-point stencil it suffices to check 4 triangles. If we find an obtuse one we can stop the search, because there are either no obtuse triangles or two of them which are opposite.

At first sight, it seems that we have to do the transformation with $M^{-1/2}$ explicitly. But we need only the scalarproduct and the norm, so it suffices only to calculate M^{-1} which is much simpler.

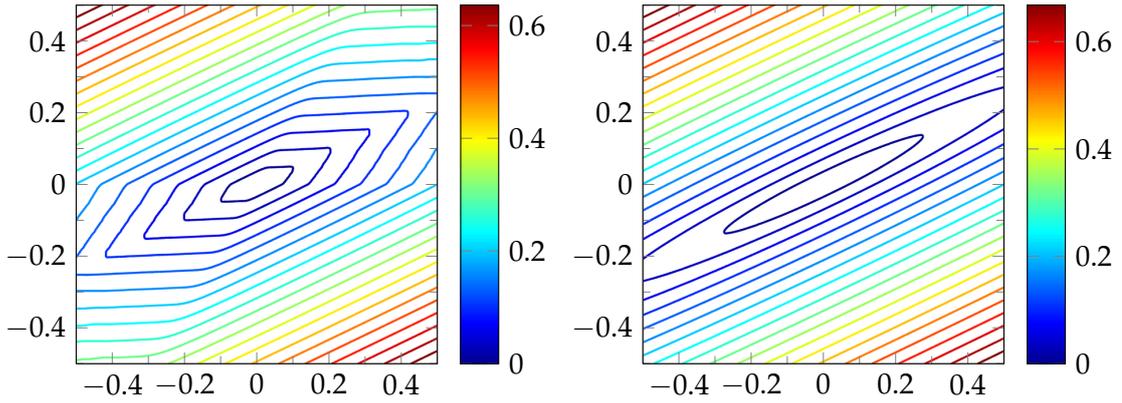


Figure 3.4.1: Computed solution of example 3.4.2 with stepsize $h = 2^{-8}$ using FMM4 (left) and the AFM8 (right).

For the AGS we used a code provided by the authors of [BC11], see also the example in subsection 3.4.6. The update scheme is the one which was introduced in [Lino3]. The 2QGS is based on the AGS. We only changed the handling of the queue so we directly compare the influence of choosing the order of iterations. We further introduced a tolerance in the 2QGS like in AGS. With this tolerance we prevent the 2QGS from propagating updates with too small an enhancement.

3.4.2 Constant speed

In the first example we test whether the algorithm works with a constant metric. We set for the metric $\phi = \pi/7$ and $\lambda_1 = 100$, $\lambda_2 = 1$, thus $Y = 10$ and the domain is $\Omega = [-1/2, 1/2]^2$.

We see in figure 3.4.1 the typical corners of the level sets using the plain FMM, whereas AFM gives a good result.

In figure 3.4.2 we can recognize convergence using AFM or iterative methods. The plain FMM does not converge. Its update scheme is consistent but it does not cope with the causality property. Furthermore, all methods have in practice a linear complexity. This is suspected for FMM and AFM, but AGS and 2QGS has also linear complexity due to the special structure of this problem. In a non-constant metric one finds that the complexity of AGS is $\mathcal{O}(N^{3/2})$.

3.4.3 Monotonicity of AFM

In section 2.1.3 we remarked that, using virtual updates, it may happen that the update scheme is no longer monotone. This example clearly show this effect. We use the same settings as before, but now with $\lambda_1 = 400$, thus $Y = 20$.

We see in figure 3.4.3 that without enforced monotonicity the result has multiple local minima. Enforcing monotonicity we have no local minima but still some corners in the level sets. These corners develop at the beginning of the computation, because we find virtual triangles for the update, but these triangles do not contain any useful

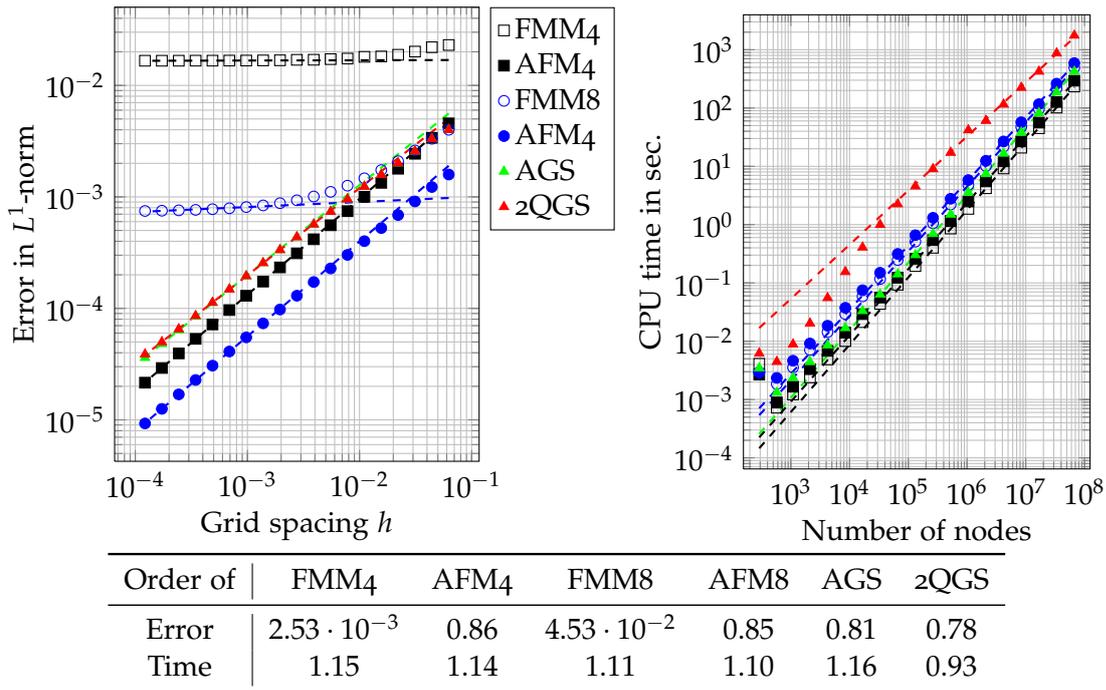


Figure 3.4.2: Left: Convergence of the different algorithms. One sees that without the usage of virtual triangles one does not achieve convergence.
 Right: Time to compute the solution. We have nearly linear complexity because the constant of the $N \log(N)$ term is very small.

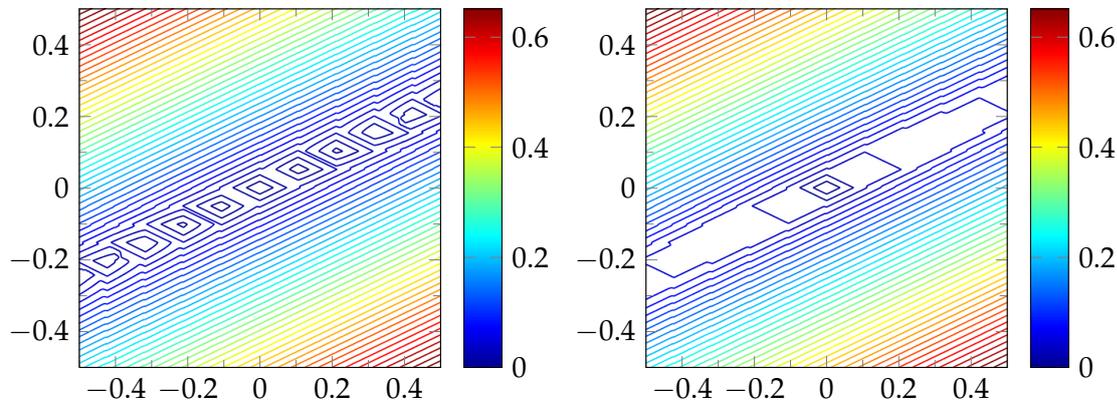


Figure 3.4.3: Computed solution of example 3.4.3 with stepsize $h = 2^{-8}$ using AFM₄ without monotonicity (left) and enforcing monotonicity (right).

information at this stage, and thus we effectively compute the first points as with plain FMM. Of course this effect is weaker using the 8-point neighbourhood but we used for the figure 3.4.3 the 4-point neighbourhood to clearly show that there is such an effect.

Enforcing monotonicity adds numerical errors. This effect is shown in figure 3.4.4,

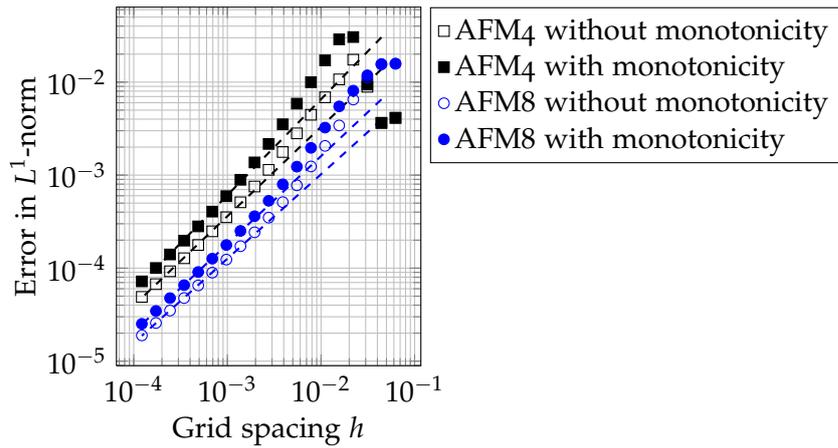


Figure 3.4.4: Convergence of AFM4 and AFM8 with and without monotonicity.

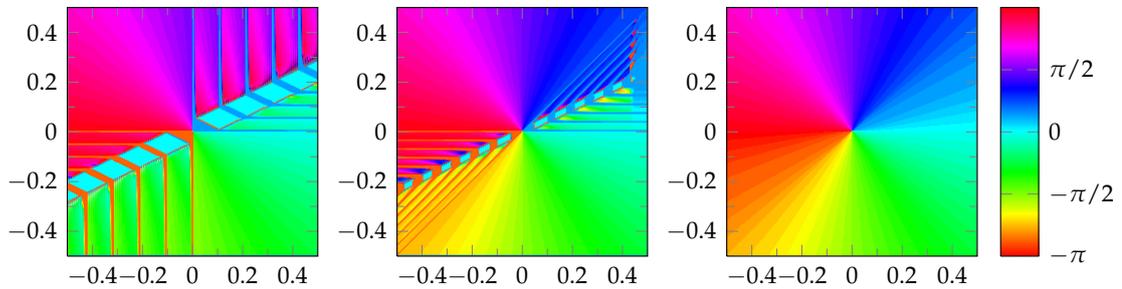


Figure 3.4.5: Computed characteristics using AFM4 (left), AFM8 (middle) and AGS (right).

but this effect is not very strong and the order of convergence is still preserved.

In all examples the default setting is to enforce monotonicity because I think is important to preserve this property.

3.4.4 Characteristics

The virtual updates in the AFM has some side effect for the characteristics. In this example we show the effect of virtual updates for the approximation of characteristics. The setting is the same as in the preceding example.

We see in figure 3.4.5 the approximation for the characteristics. In the right picture we see that AGS does a good approximation of the characteristics which are straight lines beginning at the origin. The AFM badly approximates the characteristics in the quadrant, using 4 neighbours, respectively in the octant, using 8 neighbours, where virtual updates are carried out. This bad approximation is not even restricted to the points where corners in the level-set appear; this error is transported to the whole quadrant, respectively octant.

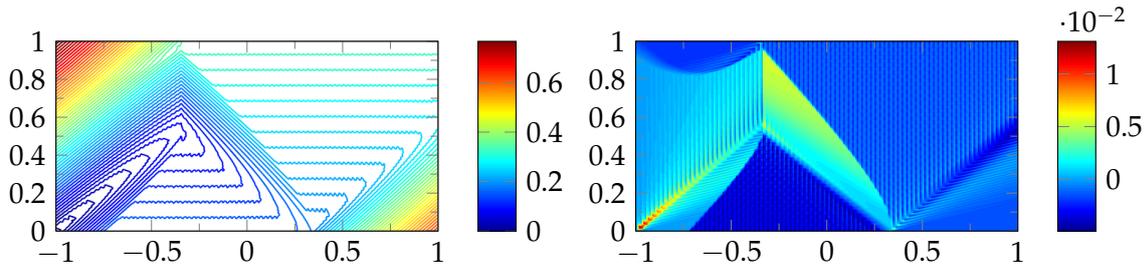


Figure 3.4.6: Left: Computed solution of example 3.4.5 using AFM8.
Right: Estimated error.

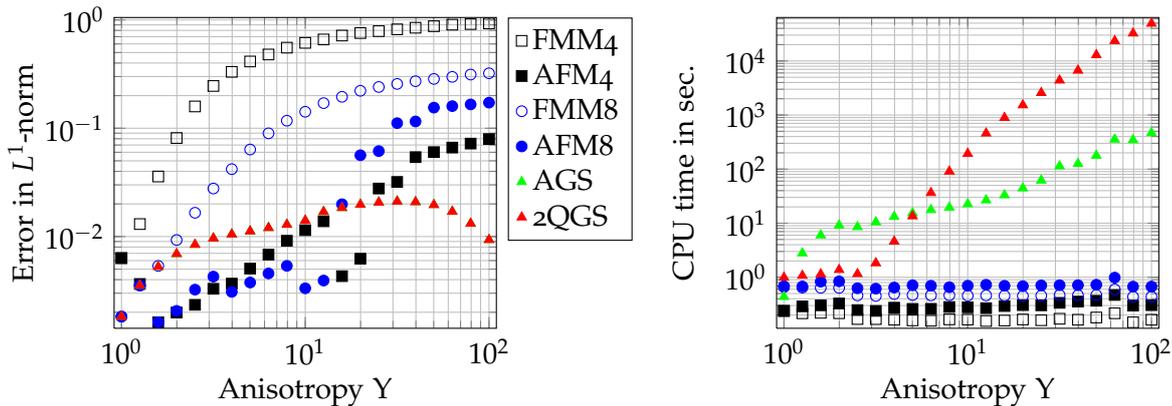


Figure 3.4.7: Error vs. anisotropy (left) and CPU time vs. anisotropy (right)

This effect not so surprising because the direction of the characteristics is an information of the first derivative of the solution and using virtual updates we locally coarsen the mesh and thus we cannot expect that this information is well approximated.

Thus the AFM is not suitable for directly approximating the characteristics or the derivative of the value function u .

3.4.5 Three regions

In this example we show the effect of varying the anisotropy. We have the following setting: $\Omega = [-1, 1] \times [0, 1]$, source point $(-1, 0)$, the metric is defined by $\phi = 2\pi/9$ in $([-1, -1/3] \cup [1/3, 1]) \times [0, 1]$ and $\phi = -2\pi/9$ in $[-1/3, 1/3] \times [0, 1]$, $\lambda_2 = 1$ and $\lambda_1 = Y^2$. We have chosen a stepsize $h = 2^{-8}$.

In figure 3.4.6 we see the computed solution and the estimated error. We used a solution computed with stepsize $h = 2^{-9}$ and the AGS to estimate the error.

The dependency of Y and the error respectively the CPU time is displayed in figure 3.4.7. We see in the left graph that for an anisotropy up to 20 AFM and AGS/2QGS give results with similar accuracy. For higher anisotropy the AFM is less accurate than AGS and 2QGS. In the right graph we see that the computational time for the AFM is nearly

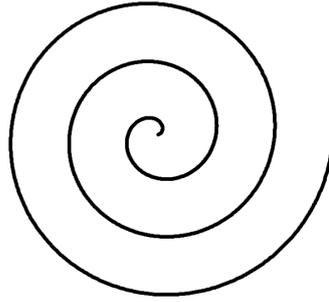


Figure 3.4.8: Source image for anisotropic metric in example 3.4.6.

independent to the anisotropy. AGS is more sensitive to an increase of anisotropy and 2QGS is even worse.

In this example the 2QGS is very inefficient for high anisotropy, because it propagates the information in a bad way; namely already visited points with the smallest value are updated first. Thus it takes too long to transport the information along the characteristics. This effect is even more pronounced if we use no tolerance in the algorithm as in the AGS. The AGS instead transports the information more equally because the FIFO-queue therein does not depend on the function values.

3.4.6 Spiral

This last example is adapted from [BC11, Section 2.3]. It is motivated by the field of medical images processing. In figure 3.4.8 we see the source image with 548×600 pixels which defines the anisotropic metric. The metric is defined as follows: Along the black line we travel fast with anisotropy Υ ; near the black line we walk fast to the line, but slow in parallel direction; in the outer region we walk slow in all directions.

We computed the distance map with starting point at one end of the line with different solvers using a notebook with Intel P8400 processor. For $\Upsilon = 10$ we need about 40 sec. for AGS, 5.2 sec. for 2QGS and 6.2 sec. for AFM8. The computed distance map is for all algorithms practically the same and the minimizing path is nearly identical too, see figure 3.4.9.

In this example 2QGS is slightly faster than AFM8 and much more efficient than AGS. In the example above 2QGS was much slower than AGS. The reason may be that we have only in a small part of the image an anisotropy and in the isotropic part of the image 2QGS acts nearly as a plain FMM which speeds up the algorithm.

Using a higher anisotropy Υ the computational time heavily increases for AGS to 132 sec. but 2QGS needs 5.4 sec. and AFM8 needs 6.4 sec. We see that the effort for 2QGS and AFM8 is nearly independent of the anisotropy but the AGS is very sensitive.

The results of AGS and 2QGS are practically the same, and also the distance map computed by AFM8 seems to be good, but the approximation of characteristics is bad in the region with high anisotropy which is the part we are interested in. The minimizing path for AFM8 in figure 3.4.10 breaks down because the path tries to go along the characteristics but the characteristics are badly approximated using virtual updates.

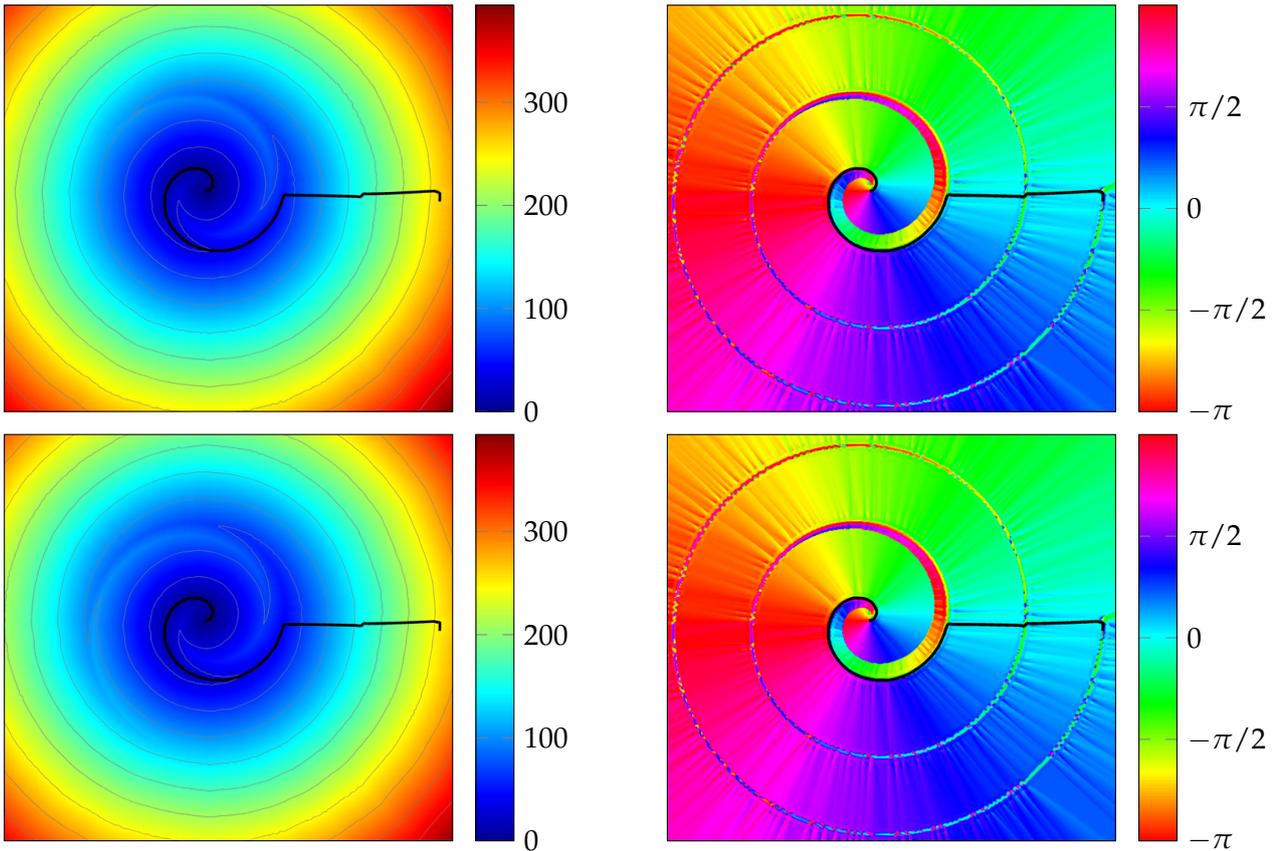


Figure 3.4.9: Distance map (left) and plot of direction of the characteristics (right) with minimizing path (black) for AGS (top) and AFM8 (bottom) for anisotropy $Y = 10$.

So the problem is here less the computed value for the distance but more the local errors in the first derivative. If we smooth the distance map with a Gaussian filter with $\sigma = 5$ and a size of 20 pixel, then the characteristics are also smoothed and we get a minimizing path which seems to be the same as using AGS or 2QGS.

3.5 Conclusion

We saw in the previous examples that the AFM converges to the solution of the anisotropic eikonal equation. Like the other methods AGS and 2QGS the convergence is of first order. The main advantage of the AFM is that the computational time of the method is in practice independent of the anisotropy, but for higher anisotropy we lose accuracy. We have the typical trade off between accuracy and effort. This loss of accuracy is a consequence of the local grid transformation and the construction of virtual updates so that we effectively coarsen the mesh. This leads to the effect that we have on the one hand a good approximation of the value function, but the estimates of

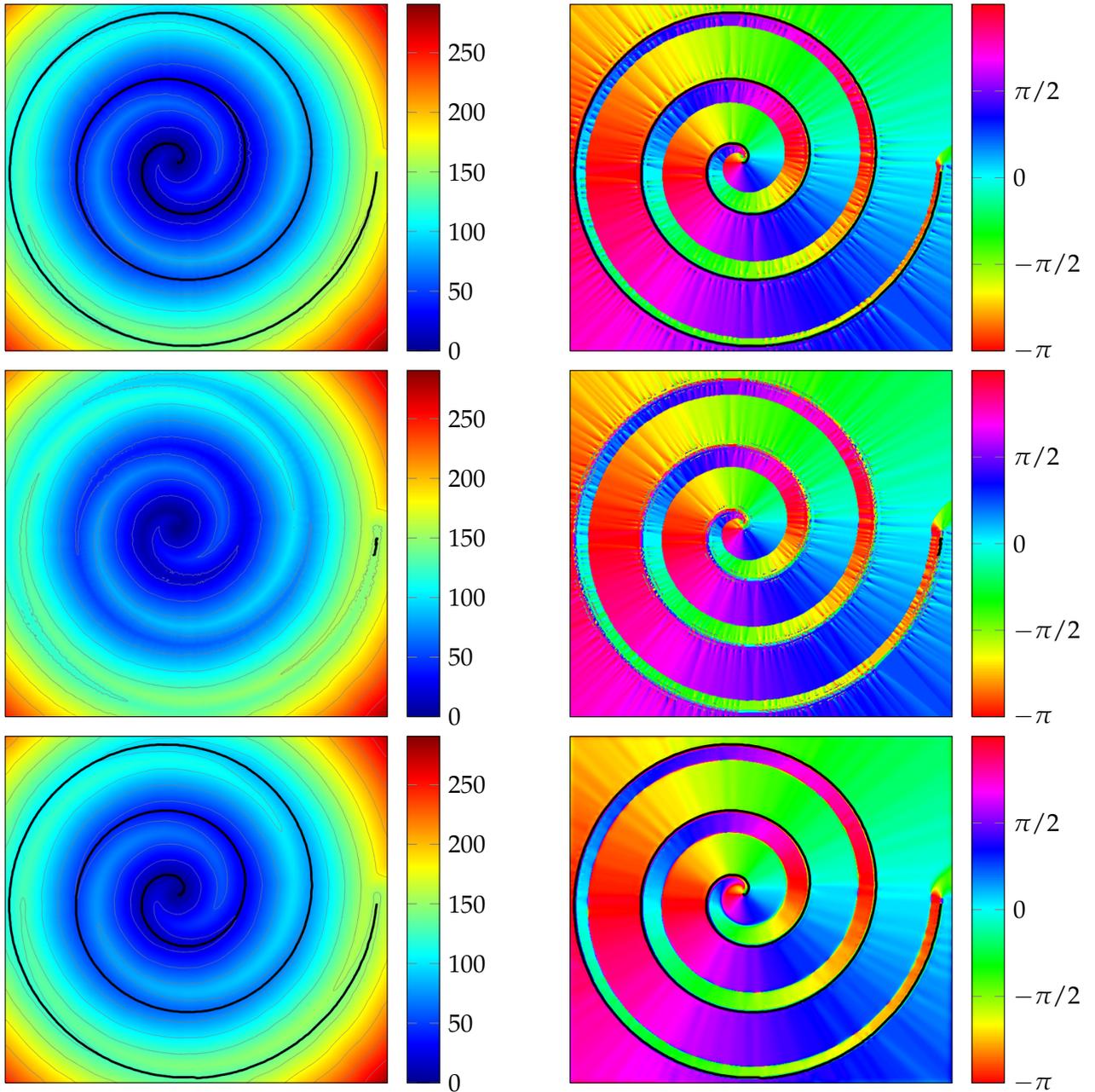


Figure 3.4.10: Distance map (left) and plot of the direction of the characteristics (right) with minimizing path (black) for AGS (top), AFM8 (middle) and smoothed distance (bottom) for anisotropy $\Upsilon = 100$.

the first derivative and in particular the direction of characteristics are not satisfying. The problems with the approximation of the first derivative and the monotonicity appear also in the plain FMM if we use virtual updates, but we are able to clearly show this effect.

For the AGS and the 2QGS we have in general more accurate results but the computational time is higher and is not predictable. In particular for the 2QGS we cannot predict the computational time. In example 3.4.5 we saw that this method is very inefficient but in the spiral example 3.4.6 it was even faster than the AFM. It would be worth while investigating in which cases the 2QGS is efficient and in which applications one can use this method.

For applications where the computational time is a criterion it seems to be better to use the AFM, because we can predict the time much better and nearly guarantee it. The iterative solvers AGS and 2QGS are on the other hand more accurate but we cannot give estimates for the computational time.

An important difference between the AFM and the algorithm of [SK04] is that we regard here only local transformations of the mesh and that we do not need any manifold in the background. The article [WDB⁺08] computes also geodesic distances on manifolds like [SK04] but they use a parallel approach. Also in this article the authors restrict themselves to manifolds, but it seems that it should also be able to solve the more general anisotropic eikonal equations. So it would be interesting to extend their method to general anisotropic eikonal equation, because under some suitable assumptions they can guarantee linear complexity.

Symbols and notation

\mathbb{R}	the set of real numbers
\mathbb{R}^d	the euclidean d -dimensional space
$\langle p, q \rangle$	the euclidean scalar product $\sum_{i=1}^d p_i q_i$ of $p = (p_1, \dots, p_d), q = (q_1, \dots, q_d) \in \mathbb{R}^d$
$ x $	the euclidean norm of $x \in \mathbb{R}^d$
$B(E)$	the space of bounded functions $u : E \rightarrow \mathbb{R}$ with $ u _\infty < \infty$
$UC(E)$	the space of uniformly continuous functions $u : E \rightarrow \mathbb{R}$
$BUC(E)$	the space $B(E) \cap UC(E)$
$C^n(E)$	the space of functions $u : E \rightarrow \mathbb{R}$ that are n -times differentiable
$C^{0,1}(E, X)$	the space of Lipschitz-continuous functions $u : E \rightarrow X$
$W^{1,\infty}(E)$	the space of functions $u \in C^1(E)$ with $ u' _\infty < \infty$
S^d	the set of the $d \times d$ symmetric positive definite matrices
$\partial\Omega$	the boundary of the set Ω
$\bar{\Omega}$	the closure of the set Ω
Ω^C	the complement of the set Ω
$B(x_0, r)$	the open ball $\{x \in \mathbb{R}^d : x - x_0 < r\}$
$\overline{B(x_0, r)}$	the closed ball $\{x \in \mathbb{R}^d : x - x_0 \leq r\}$
u^*	the upper semi-continuous envelope of u
u_*	the lower semi-continuous envelope of u
\dot{u}	the derivative u of with respect to t
2QGS	recursive fast marching method according to [KSC ⁺ 07]
AFM _{4/8}	fast marching method for anisotropic metrics on structured grids using the 4-point or 8-point neighbourhood, see section 3.2
AGS	adaptive Gauss-Seidel iteration according to [BR06]
FMM _{4/8}	plain fast marching method according to [Set99b] using the 4-point or 8-point neighbourhood
GFA	generalized fast marching method for anisotropic metrics, see section 2.6

Symbols and notation

GFMM	generalized fast marching method according to [CFFM08]
mGFMM	modified generalized fast marching method according to [For09]
GFT	generalized fast marching method for unstructured triangulations, see section 2.3.1
mGFT	modified generalized fast marching method for unstructured triangulations, see section 2.3.2

Bibliography

- [Bar93] BARLES, Guy: Discontinuous viscosity solutions of first-order Hamilton-Jacobi equations: a guided visit. In: *Nonlinear Analysis* 20 (1993), Nr. 9, 1123 - 1134. DOI – 10.1016/0362-546X(93)90098-D (Cited on p. 16.)
- [Bar97] BARDI, Martino: Some Applications of Viscosity Solutions to Optimal Control and Differential Games. In: *Lecture Notes in Mathematics* 1660 (1997), S. 44–97. DOI – 10.1007/BFb0094293 (Cited on p. 12.)
- [BC11] BENMANSOUR, Fethallah ; COHEN, Laurent: Tubular Structure Segmentation Based on Minimal Path Method and Anisotropic Enhancement. In: *International Journal of Computer Vision* 92 (2011), 192-210. DOI – 10.1007/s11263-010-0331-0 (Cited on p. 112, 116.)
- [BCD97] BARDI, Martino ; CAPUZZO-DOLCETTA, Italo: *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations (Systems & Control: Foundations & Applications)*. 1. Birkhäuser Boston, 1997. DOI – 10.1007/978-0-8176-4755-1 (Cited on p. 3, 8, 12.)
- [BP90] BARLES, G. ; PERTHAME, B.: Comparison principle for dirichlet-type Hamilton-Jacobi equations and singular perturbations of degenerated elliptic equations. In: *Applied Mathematics and Optimization* 21 (1990), S. 21–44 (Cited on p. 12.)
- [BR06] BORNEMANN, Folkmar ; RASCH, Christian: Finite-element Discretization of Static Hamilton-Jacobi Equations based on a Local Variational Principle. In: *Computing and Visualization in Science* 9 (2006), 57-69. DOI – 10.1007/s00791-006-0016-y (Cited on p. ix, 11, 23, 24, 25, 40, 46, 101, 103, 121.)
- [BSS93] BARLES, Guy ; SONER, Halil M. ; SOUGANIDIS, Panagiotis E.: Front Propagation and Phase Field Theory. In: *SIAM Journal on Control and Optimization* 31 (1993), Nr. 2, 439-469. DOI – 10.1137/0331021 (Cited on p. 13, 15, 16.)
- [CCF06] CARLINI, Elisabetta ; CRISTIANI, Emiliano ; FORCADEL, Nicolas: A Non-Monotone Fast Marching Scheme for a Hamilton-Jacobi Equation Modelling Dislocation Dynamics. Version: 2006. DOI – 10.1007/978-3-540-34288-5-70. In: CASTRO, Alfredo B. (Hrsg.) ; GÓMEZ, Dolores (Hrsg.) ; QUINTELA, Peregrina (Hrsg.) ; SALGADO, Pilar (Hrsg.): *Numerical Mathematics and Advanced Applications*. Springer Berlin Heidelberg, 2006. – ISBN 978-3-540-34288-5, S. 723–731 (Cited on p. 21.)
- [CEL84] CRANDALL, M. G. ; EVANS, L. C. ; LIONS, P. L.: Some Properties of Viscosity Solutions of Hamilton-Jacobi Equations. In: *Transactions of the American*

- Mathematical Society* 282 (1984), Nr. 2, pp. 487-502. DOI – 10.1090/S0002-9947-1984-0732102-X (Cited on p. 2, 8.)
- [CFFM] CARLINI, Elisabetta ; FALCONE, Maurizio ; FORCADEL, Nicolas ; MONNEAU, Régis: Convergence of a Generalized Fast Marching Method for a non-convex eikonal equation. <http://hal.archives-ouvertes.fr/hal-00109596/en/> (Cited on p. 31.)
- [CFFMo8] CARLINI, Elisabetta ; FALCONE, Marizio ; FORCADEL, Nicolas ; MONNEAU, Régis: Convergence of a Generalized Fast-Marching Method for an Eikonal Equation with a Velocity-Changing Sign. In: *SIAM Journal on Numerical Analysis* 46 (2008), Nr. 6, 2920-2952. DOI – 10.1137/06067403X (Cited on p. ix, 21, 26, 27, 31, 32, 33, 44, 46, 50, 76, 77, 78, 80, 82, 96, 97, 98, 99, 100, 122.)
- [CFM] CARLINI, Elisabetta ; FORCADEL, Nicolas ; MONNEAU, Régis: A Generalized Fast Marching Method for dislocation dynamics. <http://hal.archives-ouvertes.fr/hal-00415902/en/> (Cited on p. 37, 76, 97.)
- [CIL92] CRANDALL, Michael G. ; ISHII, Hitoshi ; LIONS, Pierre-Louis: User's guide to viscosity solutions of second order partial differential equations. In: *Bulletin of the American Mathematical Society* 27 (1992), Nr. 1, 1–67. DOI – 10.1090/S0273-0979-1992-00266-5 (Cited on p. 2, 14.)
- [CL81] CRANDALL, Michael G. ; LIONS, Pierre-Louis: Condition d'unicité pour les solutions généralisées des équations de Hamilton-Jacobi du premier ordre. In: *Comptes Rendus Mathématique, Paris* 292 (1981), S. 183–186 (Cited on p. 2.)
- [CL83] CRANDALL, Michael G. ; LIONS, Pierre-Louis: Viscosity Solutions of Hamilton-Jacobi Equations. In: *Transactions of the American Mathematical Society* 277 (1983), Nr. 1, 1–42. DOI – 10.2307/1999343 (Cited on p. 2.)
- [CL84] CRANDALL, Michael G. ; LIONS, Pierre-Louis: Two Approximations of Solutions of Hamilton-Jacobi Equations. In: *Mathematics of Computation* 43 (1984), Nr. 167, 1–19. DOI – 10.2307/2007396 (Cited on p. 2.)
- [Cri07] CRISTIANI, Emiliano: *Fast Marching and semi-Lagrangian methods for Hamilton-Jacobi equations with applications*, Sapienza - Università di Roma, Diss., February 2007. <http://www.emiliano.cristiani.name/research.htm> (Cited on p. 21.)
- [Eva10] EVANS, Lawrence C.: *Partial Differential Equations: Second Edition (Graduate Studies in Mathematics)*. 2. American Mathematical Society, 2010 (Cited on p. 2, 3, 4, 5.)
- [FLGG08] FORCADEL, Nicolas ; LE GUYADER, Carole ; GOUT, Christian: Generalized fast marching method: applications to image segmentation. In: *Numerical Algorithms* 48 (2008), 189-211. <http://dx.doi.org/10.1007/s11075-008-9183-x> (Cited on p. 37, 97.)

- [Fom97] FOMEL, Sergey: A variational formulation of the fast eikonal solver / Stanford Exploration Project. 1997 (95, pp. 127–149). – Technical Report (Cited on p. 23.)
- [For09] FORCADEL, Nicolas: Comparison Principle for a Generalized Fast Marching Method. In: *SIAM Journal on Numerical Analysis* 47 (2009), Nr. 3, 1923-1951. DOI – 10.1137/080718991 (Cited on p. 21, 26, 33, 34, 37, 39, 40, 47, 51, 52, 57, 76, 98, 122.)
- [Ish87] ISHII, Hitoshi: A Simple, Direct Proof of Uniqueness for Solutions of the Hamilton-Jacobi Equations of Eikonal Type. In: *Proceedings of the American Mathematical Society* 100 (1987), Nr. 2, pp. 247-251. DOI – 10.1090/S0002-9939-1987-0884461-3 (Cited on p. 9.)
- [JBT⁺08] JBABDI, S. ; BELLEC, P. ; TORO, R. ; DAUNIZEAU, J. ; PÉLÉGRINI-ISSAC, M. ; BENALI, H.: Accurate anisotropic fast marching for diffusion-based geodesic tractography. In: *Journal of Biomedical Imaging* 2008 (2008), January, 2:1–2:12. DOI – 10.1155/2008/320195 (Cited on p. 110.)
- [KS98] KIMMEL, Ron ; SETHIAN, James A.: Computing Geodesic Paths on Manifolds. In: *Proceedings of the National Academy of Sciences of the United States of America* 95 (1998), Nr. 15, pp. 8431-8435. <http://www.jstor.org/stable/45775> (Cited on p. ix, 25, 50.)
- [KSC⁺07] KONUKOGLU, Ender ; SERMESANT, Maxime ; CLATZ, Olivier ; PEYRAT, Jean-Marc ; DELINGETTE, Hervé ; AYACHE, Nicholas: A Recursive Anisotropic Fast Marching Approach to Reaction Diffusion Equation: Application to Tumor Growth Modeling. Version: 2007. DOI – 10.1007/978-3-540-73273-0_57. In: *IPMI*. 2007, 687-699 (Cited on p. 101, 111, 121.)
- [Lino3] LIN, Qingfen: *Enhancement, Detection, and Visualization of 3D Volume Data*. SE-581 83 Linköping, Sweden, Linköping University, Diss., May 2003. <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-54344>. – Dissertations No. 824, ISBN 91-7373-657-0 (Cited on p. 101, 110, 112.)
- [Lio82] LIONS, Pierre-Louis: *Generalized solutions of Hamilton-Jacobi equations*. Research Notes in Mathematics, 1982 (Cited on p. 2, 8, 9.)
- [Ras07] RASCH, Christian Mark A.: *Numerical Discretization of Static Hamilton-Jacobi Equations on Triangular Meshes*. München, Technische Universität München, Dissertation, 2007. <http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20070206-604594-0-3> (Cited on p. ix, 9, 10, 25, 77, 101.)
- [RS09] RASCH, Christian ; SATZGER, Thomas: Remarks on the implementation of the fast marching method. In: *IMA Journal of Numerical Analysis* 29 (2009), Nr. 3, 806-813. DOI – 10.1093/imanum/drm028 (Cited on p. 51.)

- [RT92] ROUY, Elisabeth ; TOURIN, Agnès: A Viscosity Solutions Approach to Shape-From-Shading. In: *SIAM Journal on Numerical Analysis* 29 (1992), Nr. 3, 867-884. DOI – 10.1137/0729053 (Cited on p. 22, 31.)
- [Set96a] SETHIAN, James A.: A Fast Marching Level Set Method for Monotonically Advancing Fronts. In: *Proceedings of the National Academy of Sciences of the United States of America* 93 (1996), Nr. 4, pp. 1591-1595. <http://www.jstor.org/stable/38628> (Cited on p. ix, 22.)
- [Set96b] SETHIAN, James A.: Theory, algorithms, and applications of level set methods for propagating interfaces. In: *Acta Numerica* 5 (1996), 309-395. DOI – 10.1017/S0962492900002671 (Cited on p. 22.)
- [Set99a] SETHIAN, James A.: Fast Marching Methods. In: *SIAM Review* 41 (1999), Nr. 2, 199-235. DOI – 10.1137/S0036144598347059 (Cited on p. 102.)
- [Set99b] SETHIAN, James A.: *Level set methods and fast marching methods. Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science.* Cambridge Monographs on Applied and Computational Mathematics, 1999. DOI – 10.2277/0521645573 (Cited on p. 22, 23, 24, 121.)
- [SK04] SPIRA, Alon ; KIMMEL, Ron: An efficient solution to the eikonal equation on parametric manifolds. In: *Interfaces Free Bound.* 6 (2004), Nr. 3, S. 315–327. DOI – 10.4171/IFB/102 (Cited on p. vi, 101, 102, 110, 119.)
- [SKD⁺07] SERMESANT, Maxime ; KONUKOGLU, Ender ; DELINGETTE, Hervé ; COUDIÈRE, Yves ; CHINCHAPATNAM, Phani ; RHODE, Kawal ; RAZAVI, Reza ; AYACHE, Nicholas: An Anisotropic Multi-front Fast Marching Method for Real-Time Simulation of Cardiac Electrophysiology. Version: 2007. DOI – 10.1007/978-3-540-72907-5_17. In: SACHSE, Frank (Hrsg.) ; SEEMANN, Gunnar (Hrsg.): *Functional Imaging and Modeling of the Heart Bd.* 4466. Springer Berlin / Heidelberg, 2007, 160-169 (Cited on p. 101.)
- [Sou97] SOUGANIDIS, Panagiotis E.: Front Propagation: Theory and Applications. In: *Lecture Notes in Mathematics* 1660 (1997), S. 186–242. DOI – 10.1007/BFb0094293 (Cited on p. 13.)
- [SV03] SETHIAN, James A. ; VLADIMIRSKY, Alexander: Ordered Upwind Methods for Static Hamilton–Jacobi Equations: Theory and Algorithms. In: *SIAM Journal on Numerical Analysis* 41 (2003), Nr. 1, 325-363. DOI – 10.1137/S0036142901392742 (Cited on p. 101, 102.)
- [Tsi95] TSITSIKLIS, John N.: Efficient algorithms for globally optimal trajectories. In: *IEEE Transactions on Automatic Control* 40 (1995), Nr. 9, S. 1528–1538. DOI – 10.1109/9.412624 (Cited on p. 23.)
- [WDB⁺08] WEBER, Ofir ; DEVIR, Yohai S. ; BRONSTEIN, Alexander M. ; BRONSTEIN, Michael M. ; KIMMEL, Ron: Parallel algorithms for approximation of

distance maps on parametric surfaces. In: *ACM Trans. Graph.* 27 (2008), November, 104:1–104:16. DOI – 10.1145/1409625.1409626 (Cited on p. 101, 119.)

- [YBS05] YATZIV, Liron ; BARTESAGHI, Alberto ; SAPIRO, Guillermo: $O(N)$ Implementation of the Fast Marching Algorithm. In: *Journal of Computational Physics* 212 (2005), 393–399. DOI – 10.1016/j.jcp.2005.08.005 (Cited on p. 23, 51.)