

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra matematiky

METODA FAST MARCHING PRO
HLEDÁNÍ NEJKRATŠÍCH CEST

Bakalářská práce

Plzeň, 2006

Martina SMITKOVÁ

Prohlášení

Předkládám tímto k posouzení a následné obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem předloženou bakalářskou práci vypracovala samostatně s použitím odborné literatury a pramenů, jejichž úplný seznam je součástí práce, a za odborného vedení vedoucího bakalářské práce.

V Plzni dne 16. června 2006

.....

podpis

Poděkování

Na tomto místě bych především ráda poděkovala vedoucímu své bakalářské práce panu Ing. Marku Brandnerovi, Ph.D., a to za zapůjčení odborné literatury, motivaci během práce, čas, který mi věnoval, a v neposlední řadě za velice vstřícný přístup a cenné rady.

Dále bych ráda poděkovala slečně Mgr. Anně Porazilové za poskytnutí několika modelů trojúhelníkových sítí potřebných pro testování algoritmů.

Abstrakt

Tato bakalářská práce se zabývá hledáním nejkratších cest na nepravdělné trojúhelníkové síti (tzv. TIN) s využitím metody Fast Marching.

Tato metoda odvozená v [11] je efektivní numerická metoda vhodná pro řešení eikonalové rovnice $|\nabla T|F = 1$ na obdélníkové pravoúhlé síti. Metodu lze mírně modifikovat a použít ji pro řešení eikonalové rovnice na TIN. Jako aplikaci je poté možné hledat nejkratší cesty na TIN.

V práci je popsána a implementována metoda Fast Marching, a to na pravoúhlé síti i na TIN. Následně je vyvinut vlastní algoritmus pro výpočet nejkratších cest na TIN. Získané výsledky jsou vizualizovány a analyzovány.

Klíčová slova

pohybující se hranice, eikonalová rovnice, numerické metody, metoda Fast Marching, nejkratší cesty, TIN, MATLAB

Abstract

This bachelor thesis deals with construction of shortest paths on a triangular irregular network (TIN) using the Fast Marching Method.

This method developed in [11] is an efficient numerical method for solving the Eikonal equation $|\nabla T|F = 1$ on a rectangular orthogonal mesh. It is possible to extend this method to a TIN and as an application to construct shortest paths on a TIN.

The Fast Marching Method on a mesh as well as on a TIN is described and implemented in the thesis. Then is developed an own algorithm for construction of shortest paths on a TIN. Results are visualised and analysed.

Keywords

moving interface, Eikonal equation, numerical methods, Fast Marching Method, shortest paths, TIN, MATLAB

Obsah

1	Úvod	9
1.1	Motivace řešení problému nejkratší cesty	9
1.2	Struktura práce	10
2	Vysvětlení základních pojmů	11
2.1	Reprezentace zemského povrchu	11
2.1.1	Globální modely	11
2.1.2	Lokální modely	12
2.2	Geodetická křivka	14
2.2.1	Definice geodetické křivky	14
2.2.2	Vlastnosti geodetické křivky	14
2.2.3	Příklady	15
3	Hledání nejkratší cesty na TIN	16
3.1	Výpočet časové funkce T	16
3.1.1	Pohybová rovnice pro pohybující se hranici	18
3.1.2	Diskretizace problému	19
3.1.3	Klasická metoda řešení	22
3.1.4	Metoda Fast Marching	23
3.1.5	Fast Marching na TIN	29
3.2	Vlastní výpočet nejkratší cesty	35
3.2.1	Aproximace gradientu funkce T	36
3.2.2	Výpočet jednotlivých bodů nejkratší cesty	38

4 Implementace a výsledky	40
4.1 Výpočet funkce T na čtvercové síti	40
4.1.1 Výpočet bez váhové funkce F	42
4.1.2 Výpočet s váhovou funkcí F	43
4.1.3 Srovnání metod použitých pro výpočet funkce T	44
4.2 Výpočet funkce T na TIN	47
4.2.1 Výpočet bez váhové funkce F	48
4.2.2 Výpočet s váhovou funkcí F	48
4.2.3 Další ukázky	51
4.3 Výpočet nejkratší cesty na TIN	53
5 Závěr	56
5.1 Shrnutí a zhodnocení	56
5.2 Výhled do budoucna	57
Literatura	59

Seznam obrázků

2.1	Příklad TIN.	13
3.1	Křivka pohybující se normálovým směrem rychlostí F	18
3.2	Diskretizace množiny \mathbb{R}^2	20
3.3	Schéma metody Fast Marching.	26
3.4	Příklad min-haldy.	28
3.5	Jednoduchá trojúhelníková síť.	30
3.6	Skupina trojúhelníků okolo společného centrálního bodu X	31
3.7	Odvození vztahu pro výpočet funkce T na trojúhelníku.	32
3.8	Trojúhelník pro výpočet ∇T	37
3.9	Čtverice trojúhelníků pro výpočet ∇T	38
3.10	Situace, kdy není možné použít schéma druhého řádu.	38
3.11	Počátek výpočtu jednotlivých bodů nejkratší cesty.	39
3.12	Průběh výpočtu jednotlivých bodů nejkratší cesty.	39
4.1	Přesné řešení rovnice $ \nabla T = 1$	41
4.2	Váhová funkce F	41
4.3	Výsledek numerického výpočtu funkce T na čtvercové síti.	42
4.4	Odchyly způsobené diskretizací úlohy.	43
4.5	Výpočet časové funkce T s uvažováním váhové funkce F	43
4.6	Rozdíly výsledků iterační metody a metody Fast Marching (1).	44
4.7	Rozdíly výsledků iterační metody a metody Fast Marching (2).	45
4.8	Iterační metoda bez váhové funkce F po 83 iteracích.	45
4.9	Iterační metoda s váhovou funkcí F po 74 iteracích.	46

4.10	Graf funkce $N/\log_2 N$	47
4.11	Jednoduchá rovinná TIN.	48
4.12	Přesné řešení rovnice $ \nabla T = 1$ na jednoduché TIN.	49
4.13	Výsledek numerického výpočtu funkce T na TIN.	49
4.14	Odchytky mezi přesným řešením a metodou Fast Marching.	50
4.15	Váhová funkce F na TIN.	50
4.16	Výpočet časové funkce T na TIN s úvahou váhové funkce F	51
4.17	Výpočet časové funkce T na prostorové TIN.	52
4.18	Výpočet časové funkce T na triangulované kulové ploše.	52
4.19	Nejkratší cesty na rovinné TIN.	53
4.20	Nejkratší vážené cesty na rovinné TIN.	54
4.21	Nejkratší cesty na prostorové TIN.	54
4.22	Nejkratší cesty na triangulované kulové ploše.	55

Kapitola 1

Úvod

1.1 Motivace řešení problému nejkratší cesty

Hledání nejkratších cest je intenzivně diskutovanou otázkou zejména při řešení navigačních úloh – ať už máme na mysli historickou mořeplavbu či moderní navigaci pomocí globálního polohového systému GPS. Odjakživa bylo potřeba dostat se z výchozího do cílového místa co nejrychleji a s co nejmenšími náklady na cestu a tyto důvody podněcovaly a stále podněcují zájem o problematiku nejkratších cest.

V geodézii se tento problém řeší již velice dlouho a je známý pod názvem *druhá základní geodetická úloha* – ze zadaných souřadnic koncových bodů je úkolem vypočítat délku geodetické křivky mezi těmito dvěma body a azimuty křivky v koncových bodech. Na klasických referenčních plochách je tato úloha jednoduše řešitelná – v rovině převodem kartézských souřadnic na polární, na kulové ploše pomocí aparátu sférické trigonometrie a na rotačním elipsoidu numerickým řešením soustavy obyčejných diferenciálních rovnic.

S nástupem geografických informačních systémů (GIS) a obecně informačních technologií se začaly prosazovat nové reprezentace zemského povrchu, jako je například nepravidelná trojúhelníková síť – tzv. TIN. Jednou z důležitých

vlastností GIS je možnost provádění prostorových analýz – analýz využívajících prostorové vztahy mezi objekty. Jednou z prostorových analýz je i problém hledání nejkratší cesty mezi dvěma body na TIN, kterým se zabývám ve své bakalářské práci – mým cílem je navrhnout a implementovat numerické metody pro hledání nejkratších cest na TIN.

1.2 Struktura práce

Tato práce je rozdělena do pěti kapitol.

První a druhá kapitola práce představují úvod do tématu. První kapitola přibližuje motivaci řešení problému nejkratší cesty. Ve druhé kapitole jsou vysvětleny některé základní pojmy – TIN jako model zemského povrchu a geodetická křivka a její souvislost s nejkratšími cestami.

Nejdůležitějšími částmi práce jsou třetí a čtvrtá kapitola, ve nichž je řešen problém hledání nejkratší cesty mezi dvěma body na TIN. Ve třetí kapitole jsou nejprve teoreticky popsány numerické metody použité pro řešení tohoto problému a ve čtvrté kapitole jsou následně prezentovány praktické výsledky získané těmito metodami.

V závěrečné páté kapitole jsou shrnuty a zhodnoceny výsledky práce a je naznačen jakýsi výhled do budoucna – několik podnětů, kterými je možné se zabývat v další práci.

Kapitola 2

Vysvětlení základních pojmů

Moje práce se týká hledání nejkratších cest na nepravidelné trojúhelníkové síti – TIN. V této kapitole bych tedy ráda naznačila, co to vlastně TIN je a jaké má postavení v hierarchii modelů zemského povrchu. Dále bych chtěla definovat pojem geodetické křivky, zmínit některé zajímavé vlastnosti geodetické křivky a vysvětlit, jak souvisí geodetické křivky s nejkratšími cestami.

2.1 Reprezentace zemského povrchu

Skutečný fyzický povrch Země (terén, dna oceánů) je velice komplikovaná plocha, kterou není možné popsat analyticky. Existuje mnoho způsobů, jak více či méně zjednodušeně reprezentovat zemský povrch. Je možné jej popisovat buď globálně – povrch celé Země – nebo lokálně – matematické vyjádření pouze malé části zemského povrchu. Od velikosti území se následně odvíjí konkrétní způsob reprezentace tohoto území.

2.1.1 Globální modely

Tyto modely se zaměřují na reprezentaci zemského povrchu jako celku a využívají se při analýzách velkého prostorového rozsahu. Globálně bývá fy-

zický povrch Země nahrazován následujícími plochami:

- geoid – hladinová plocha tíhového pole Země
- trojosý elipsoid
- rotační elipsoid (zploštělý na pólech)
- kulová plocha
- jiné složitější plochy

Při výběru nejvhodnější náhradní plochy se přihlíží k požadavkům na přesnost reprezentace, se kterou ale souvisí i komplikovanost náhradní plochy. Globálními modely zemského povrchu se zabývá vyšší geodézie, která tyto plochy nejen užívá, ale i určuje na základě různých měření (délky, úhly, tíže, dálkový průzkum, geodetické družice, letecká data atd.). Pro podrobnější informace odkazují na [8].

2.1.2 Lokální modely

Tyto modely se na rozdíl od globálních modelů zaměřují na detailnější popis menších částí zemského povrchu. Jsou využívány při analýzách malého prostorového rozsahu, ve kterých je kladen důraz na přesnost vstupních dat.

Data o zemském povrchu mohou být uchovávána následujícími dvěma kvalitativně odlišnými způsoby:

- rastrově
- vektorově

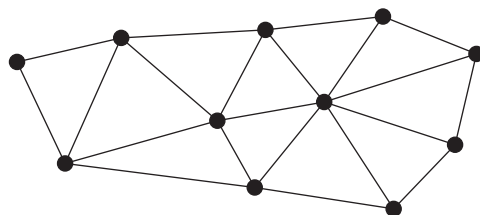
Rastrové vyjádření

Základním stavebním prvkem rastrové struktury je buňka. Buňky jsou organizovány do mozaiky. Nejpoužívanějším typem mozaiky je pravidelná čtvercová mřížka. Jednotlivé buňky obsahují střední hodnoty výškových dat z daného území. Takto je reprezentován například digitální model terénu – DMT.

Výhodou tohoto přístupu je jednoduchost datové struktury, nevýhodou je velká paměťová náročnost při zpracování a uchování dat.

Vektorové vyjádření

Nejběžnějším typem vektorového popisu zemského povrchu je nepravidelná trojúhelníková síť neboli *TIN* (z anglického Triangular Irregular Network). *TIN* reprezentuje povrch jako soubor trojúhelníků, z nichž každý je definován třemi vrcholy – body o známých prostorových souřadnicích. Každá hrana v *TIN* (s výjimkou „obvodu“ *TIN*) je sdílena právě dvěma sousedními trojúhelníky. Proces vytvoření sítě se nazývá triangulace a existuje pro něj řada algoritmů – jako příklad lze uvést Delaunayovu triangulaci.



Obrázek 2.1: Příklad *TIN*.

Výhodou vektorového přístupu je věrnější a přesnější popis tvarů zemského povrchu a optimalizace uložení dat, jeho nevýhodou je složitost datové struktury a tím i algoritmů s ní pracujících.

Mezi vektorovou a rastrovou reprezentací zemského povrchu je samozřejmě možné podle aktuální potřeby vzájemně převádět.

Detailnější informace k tématu lokálních reprezentací zemského povrchu lze nalézt například v [2].

2.2 Geodetická křivka

V této části bude definován pojem geodetické křivky a naznačena souvislost nejkratších cest a geodetických křivek. Není zde zaváděn matematický aparát diferenciální geometrie křivek a ploch, předpokládají se znalosti v rozsahu předmětu Diferenciální geometrie (KMA/DG) – viz [3].

2.2.1 Definice geodetické křivky

Nechť $\mathbf{P}(t)$, $t \in \mathcal{I}$, kde \mathcal{I} je interval, je křivka na ploše κ . Velikost průmětu vektoru první křivosti křivky $\mathbf{P}(t)$ do tečné roviny plochy κ nazýváme *geodetická křivost* křivky na ploše.

Geodetickou křivkou (nebo také *geodetikou*) rozumíme křivku plochy, která obsahuje jen body s nulovou geodetickou křivostí. Alternativně ji lze také definovat jako křivku plochy, která splňuje podmínku, že v každém jejím bodě splývá hlavní normála této křivky s normálou plochy.

2.2.2 Vlastnosti geodetické křivky

Geodetické křivky jsou významné křivky z hlediska hledání nejkratších spojnic mezi body na ploše. Vztah mezi geodetikami a nejkratšími spojnicemi vyjadřuje tato věta:

Pokud mezi dvěma body plochy existuje nejkratší spojnice, pak je geodetikou.

Geodetiky jsou tedy jakési kandidátky na nejkratší spojnice. Také platí, že každým bodem plochy prochází jediná geodetika s danou tečnou.

Další vlastnost geodetické křivky vyjadřuje *Clairauova věta*:

Nechť $\mathbf{G}(t)$, $t \in \mathcal{I}$, je geodetikou na rotační ploše s osou o a nechť $\rho(t)$ je vzdálenost bodu křivky $\mathbf{G}(t)$ od osy o . Označme $\varphi(t)$ azimut (odchylku od meridiánu) křivky $\mathbf{G}(t)$ v daném bodě. Pak součin

$$c = \rho(t) \cdot \sin \varphi(t) \quad (2.1)$$

je pro všechna $t \in \mathcal{I}$ konstantní. Naopak je-li součin $\rho(t) \cdot \sin \varphi(t)$ pro všechna $t \in \mathcal{I}$ konstantní, křivka $\mathbf{G}(t)$ je geodetikou.

2.2.3 Příklady

- Geodetikami v rovině jsou přímky této roviny. Zde splývá množina geodetik s množinou nejkratších spojnic.
- Geodetikami na kulové ploše jsou hlavní kružnice této kulové plochy (kružnice, které mají střed ve středu příslušné kulové plochy a jejich poloměr je roven poloměru této kulové plochy). Každé dva různé body (s výjimkou těch, které leží na opačných pólech kulové plochy) lze spojit dvěma geodetikami – kratším a delším obloukem hlavní kružnice. Body na opačných pólech kulové plochy spojuje nekonečně mnoho geodetik stejné délky.
- Geodetikami na rotační válcové ploše jsou površky, rovnoběžkové kružnice a šroubovice. Zvolíme-li na povrchu dva různé body neležící na téže rovnoběžkové kružnici, existuje mezi nimi nekonečně mnoho geodetik. Jen jedna z nich je jejich nejkratší spojnicí.

Kapitola 3

Hledání nejkratší cesty na TIN

K problému hledání nejkratší cesty na TIN existuje mnoho přístupů. Vstupem algoritmů je vždy TIN a dva různé body na TIN – výchozí bod a cílový bod, výstupem je pak nejkratší cesta po TIN mezi těmito dvěma body. Nepovinným vstupem může být váhová funkce F , která pro každý vrchol TIN definuje možnou rychlost pohybu v tomto vrcholu (například v závislosti na prostupnosti terénu). Výstupem pak je nejkratší vážená cesta, která minimalizuje celkový čas cesty mezi danými dvěma body.

Ve své práci problém hledání nejkratší cesty na TIN řeším ve dvou krocích:

1. Výpočet *časové funkce* T , která pro každý vrchol TIN udává minimální čas cesty do výchozího bodu.
2. Vlastní výpočet nejkratší cesty – „návrat“ z cílového do výchozího bodu podél záporně vzatého gradientu funkce T – gradient udává směr největší změny funkce T a tedy i lokální směr nejkratší cesty.

3.1 Výpočet časové funkce T

Představme si, že na ploše κ je dán výchozí bod a že známe *časovou funkci* T definovanou na ploše κ , která pro každý bod této plochy vyjadřuje minimální

čas cesty do výchozího bodu. Křivky ξ na této ploše, pro které platí

$$\xi = \{(x, y, z) : (x, y, z) \in \kappa \wedge T(x, y, z) = \text{konst}\}, \quad (3.1)$$

nazvěme *hladiny* funkce T . Platí, že vektor ∇T je ortogonální k hladinám funkce T – udává normálový směr pro daný bod křivky. Soustavu hladin funkce T je možné popsat jako jednoparametrický systém křivek $\xi(t)$, pro které platí

$$\xi(t) = \{(x, y, z) : (x, y, z) \in \kappa \wedge T(x, y, z) = t\}. \quad (3.2)$$

Na tento systém je možné nahlížet jako na výchozí křivku, pro kterou platí

$$\xi_0 = \{(x, y, z) : (x, y, z) \in \kappa \wedge T(x, y, z) = 0\}. \quad (3.3)$$

(v našem speciálním případě se jedná o jediný výchozí bod) a která se rozpíná normálovým směrem rychlostí, která je pro každý bod předsána funkcí F .

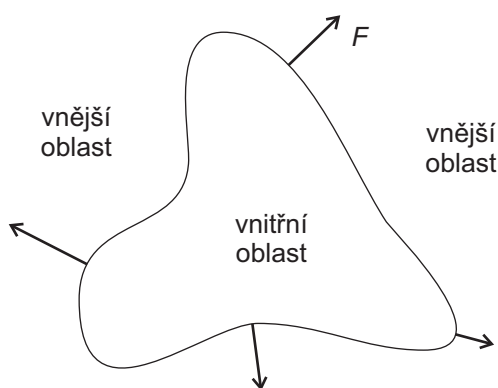
Tímto způsobem je možné ze známé polohy výchozí křivky a z dané funkce F rekonstruovat časovou funkci T .

V této části bude nejprve odvozeno matematické vyjádření pohybující se hranice v \mathbb{R}^2 , následně bude problém diskretizován do rovinné čtvercové sítě a budou hledána numerická schémata pro jeho řešení. Nastíníme klasické iterační řešení a nakonec se zaměříme na efektivní metodu Fast Marching – nejprve na rovinné čtvercové síti a poté rozšíříme algoritmus i pro výpočet na TIN.

V následujícím textu se zabývám obecnějším případem, kdy počátečním stádiem pohybující se hranice je křivka, při implementaci je jako počátek uvažován jediný bod.

3.1.1 Pohybová rovnice pro pohybující se hranici

Uvažujme jakousi hranici, například jednoduchou uzavřenou křivku v \mathbb{R}^2 , která odděluje jednu oblast od jiné. Představme si, že tato křivka se pohybuje směrem, který je pro každý bod křivky určen normálou křivky v tomto bodě a že rychlost pohybu v daném bodě je dána funkcí F (viz obrázek 3.1). Naším cílem je popsat pohyb této hranice v závislosti na funkci F a na výchozí pozici hranice.



Obrázek 3.1: Křivka pohybující se normálovým směrem rychlostí F .

Formulace okrajové úlohy

Nyní předpokládejme, že $F > 0$, takže hranice se vždy pohybuje směrem „ven“. Pozici rozpínající se hranice lze charakterizovat například *časovou funkcí* $T(x, y)$, která pro každý bod (x, y) udává čas, ve kterém hranice dosáhla tohoto bodu. Odvodíme nyní rovnici pro funkci T .

Použijeme-li známý vztah, že $vzdálenost = rychlost \times čas$, pak v jednorozměrném případě dostáváme

$$F \frac{dT}{dx} = 1. \quad (3.4)$$

Ve více dimenzích platí, že ∇T je ortogonální k hladinám funkce T a podobně jako v předchozím případě je jeho velikost nepřímo úměrná rychlosti, tedy

$$|\nabla T| F = 1, \quad T = 0 \text{ na } \Gamma, \quad (3.5)$$

kde Γ je počáteční pozice hraniční křivky. Pro polohu rozpínající se hranice v čase t potom platí

$$\Gamma(t) = \{(x, y) : T(x, y) = t\}. \quad (3.6)$$

Pohyb hranice je tedy charakterizován jako řešení okrajové úlohy. Rovnici $|\nabla T| F = 1$ se říká *eikonalová rovnice*.

3.1.2 Diskretizace problému

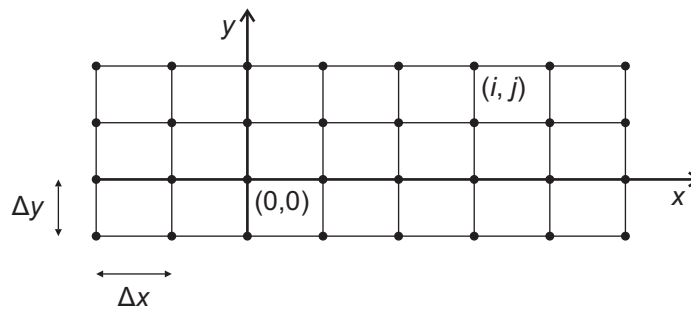
V této části a v celé další práci vycházíme ze znalostí a z terminologie zavedené v předmětu Numerické metody (KMA/NM) – viz [7] a [9].

Pro numerický výpočet časové funkce T je potřeba původní spojitý problém vhodně diskretizovat – množinu \mathbb{R}^2 nahradit rovinnou pravoúhloú výpočetní sítí a $\nabla T = \left(\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y} \right)$ v rovnici $|\nabla T| F = 1$ aproximovat pomocí příslušných

diferenčních schémat.

Diskretizace množiny \mathbb{R}^2

Diskretizaci množiny \mathbb{R}^2 provedeme standardním způsobem – viz obrázek 3.2. Představme si pravidelnou síť bodů, ve směru osy x je mezi body krok Δx a ve směru osy y krok Δy . Pro případ čtvercové sítě platí $\Delta x = \Delta y$. Umístění konkrétního bodu v této síti je pak charakterizováno dvojicí indexů i a j ($i, j \in \mathbb{Z}$) – index i pro pozici ve směru osy x a index j pro pozici ve směru osy y . Označíme $x_i = i\Delta x$ a $y_j = j\Delta y$ a jako bod (i, j) budeme rozumět bod o souřadnicích (x_i, y_j) .



Obrázek 3.2: Diskretizace množiny \mathbb{R}^2 .

Aproximace ∇T

Derivace ve výrazu pro ∇T nahradíme diferencemi odvozenými z Taylorova rozvoje funkce T .

Základní diferenční schémata:

- levá diference podle x : $D^{-x}T = \frac{T(x, y) - T(x - \Delta x, y)}{\Delta x}$
- pravá diference podle x : $D^{+x}T = \frac{T(x + \Delta x, y) - T(x, y)}{\Delta x}$

- centrální diference podle x : $D^{0x}T = \frac{T(x + \Delta x, y) - T(x - \Delta x, y)}{2\Delta x}$

Přejděme nyní na přehlednější značení. Označme $T_{ij} = T(x_i, y_j)$ a základní diferenční schémata zapišme pomocí indexů i a j :

- levá diference podle x v bodě (i, j) : $D_{ij}^{-x}T = \frac{T_{ij} - T_{i-1,j}}{\Delta x}$
- pravá diference podle x v bodě (i, j) : $D_{ij}^{+x}T = \frac{T_{i+1,j} - T_{ij}}{\Delta x}$
- centrální diference podle x v bodě (i, j) : $D_{ij}^{0x}T = \frac{T_{i+1,j} - T_{i-1,j}}{2\Delta x}$

Schémat pro levou a pravou diferenci jsou schémata prvního řádu, zatímco schéma pro centrální diferenci je schématem druhého řádu. Na první pohled se tedy může zdát výhodné použít schéma pro centrální diferenci. Nyní ale uvidíme, že řád schématu nemusí být jediným kritériem při volbě nejvhodnějšího diferenčního schématu.

Které z výše uvedených schémat tedy použít?

- při použití D_{ij}^{-x} se hodnota diference v bodě (i, j) určuje z hodnot T v bodech (i, j) a $(i - 1, j)$, tj. informace se šíří zleva doprava
- při použití D_{ij}^{+x} se hodnota diference v bodě (i, j) určuje z hodnot T v bodech (i, j) a $(i + 1, j)$, tj. informace se šíří zprava doleva
- při použití D_{ij}^{0x} se hodnota diference v bodě (i, j) určuje z hodnot T v bodech $(i - 1, j)$ a $(i + 1, j)$, tj. informace se šíří z obou stran

Je vhodné, aby se směr šíření skutečné informace shodoval s matematickým směrem šíření informace prostřednictvím diferenčních operátorů. Zkoumaná rozpínající se hranice se v různých bodech pohybuje různými směry. Bylo

by ideální, kdyby pro jednotlivé body mohla být nezávisle volena optimální diferenční schémata.

Představme si, že v daném bodě se hranice šíří zleva doprava. Chtěli bychom tedy, aby byl použit diferenční operátor D^{-x} . Jestliže se hranice šíří tímto směrem, pak napravo od daného bodu se nacházejí větší hodnoty T než nalevo a pro D^{-x} i pro D^{+x} (označme nyní tyto dva operátory souhrnně jako $D^{\pm x}$) platí $D^{\pm x} > 0$. Naopak pokud se v daném bodě hranice šíří zprava doleva, pak bychom rádi použili operátor D^{+x} a platí $D^{\pm x} < 0$.

Tyto požadavky splňuje následující schéma:

$$\frac{\partial T}{\partial x} \approx [\max(D_{ij}^{-x}T, 0) + \min(D_{ij}^{+x}T, 0)] . \quad (3.7)$$

Všechny výše uvedené úvahy lze samozřejmě provést analogicky i pro difference podle y . Dostáváme tedy schéma

$$\frac{\partial T}{\partial y} \approx [\max(D_{ij}^{-y}T, 0) + \min(D_{ij}^{+y}T, 0)] . \quad (3.8)$$

Pro velikost gradientu platí

$$|\nabla T| = \left[\left(\frac{\partial T}{\partial x} \right)^2 + \left(\frac{\partial T}{\partial y} \right)^2 \right]^{\frac{1}{2}} . \quad (3.9)$$

3.1.3 Klasická metoda řešení

Pro okrajovou úlohu (3.5) je v [11] odvozeno toto numerické schéma prvního řádu:

$$\left[\begin{array}{l} \max(D_{ij}^{-x}T, 0)^2 + \min(D_{ij}^{+x}T, 0)^2 + \\ + \max(D_{ij}^{-y}T, 0)^2 + \min(D_{ij}^{+y}T, 0)^2 \end{array} \right]^{\frac{1}{2}} = \frac{1}{F_{ij}}, \quad (3.10)$$

kde $F_{ij} = F(x_i, y_j)$.

Za předpokladu, že okolní hodnoty T jsou dány, je to vlastně kvadratická rovnice pro T_{ij} . Je možné ji řešit iteračně podle následujícího postupu:

```
for iter = 1 : n
  for i, j = 1 : N
    řešení kvadratické rovnice pro  $T_{ij}^{iter+1}$ 
    z daných hodnot  $T_{i-1,j}^{iter}$ ,  $T_{i+1,j}^{iter}$ ,  $T_{i,j-1}^{iter}$ ,  $T_{i,j+1}^{iter}$ 
  end
end,
```

kde N je počet bodů v každé dimenzi výpočetní sítě. Za optimistického předpokladu, že řádově po N iteracích obdržíme dostatečně přesné řešení (podle [11] však počet iterací může ve skutečnosti být i mnohem větší), je výpočetní složitost tohoto postupu $\mathcal{O}(N^3)$.

3.1.4 Metoda Fast Marching

Nášim cílem je vytvořit algoritmus pro řešení eikonalové diferenciální rovnice

$$|\nabla T| F = 1. \quad (3.11)$$

Připomeňme, že výpočet funkce T stále řešíme na rovinné čtvercové síti bodů. Jako vhodné schéma pro numerické řešení rovnice (3.11) se podle [11] jeví

$$\left[\begin{array}{l} \max(D_{ij}^{-x}T, -D_{ij}^{+x}T, 0)^2 + \\ + \max(D_{ij}^{-y}T, -D_{ij}^{+y}T, 0)^2 \end{array} \right]^{\frac{1}{2}} = \frac{1}{F_{ij}}. \quad (3.12)$$

Klasická metoda pro řešení okrajové úlohy vyžaduje iterace. V této části bude popsána metoda Fast Marching, která umožňuje řešit okrajovou úlohu (3.5) bez nutnosti iteračního výpočtu. K tomu je využito optimálního řazení bodů sítě.

Princip metody

Jaká je tedy hlavní idea v pozadí metody Fast Marching? Řešení T je v daném bodě vhodné konstruovat tím směrem, kterým se v tomto bodě šíří pohybující se hranice. Ze struktury schématu (3.12) plyne, že informace se může šířit „jednosměrně“, což v tomto případě znamená od menších hodnot T k větším hodnotám.

Popis metody

Metoda Fast Marching systematicky buduje řešení od okrajové podmínky – nejmenších hodnot T – směrem ven. Okolo stávající hranice je dynamicky vytvářen jakýsi úzký pás *dočasných* bodů uvažovaných při výpočtu dalšího stádia hranice, v každém kroku metody je hodnota T jednoho *dočasného* bodu označena jako *známá* a bod je z pásu vyřazen a naopak jiné body jsou do úzkého pásu přidány. Otázkou zůstává, který *dočasný* bod má být vybrán jako *známý*.

Je potřeba si uvědomit, že funkce T pro každý bod výpočetní sítě udává minimální čas cesty do výchozího bodu, a proto hodnota T žádného bodu nemůže být ovlivněna body s většími hodnotami T . Jako příklad uvažujme

dva různé body X a Y , pro které platí $T(X) < T(Y)$, a dále předpokládejme, že minimální čas cesty z bodu Y do bodu X je $\tau > 0$. Minimální čas cesty z výchozího bodu do bodu X přes bod Y je pak roven $T(Y) + \tau$ a platí, že $T(Y) + \tau > T(Y) > T(X)$. Čas cesty do bodu X přes bod Y tedy nemůže být menší než původní $T(X)$, a proto bod Y neovlivní hodnotu $T(X)$.

Z úvah v předchozím odstavci plyne, že bodem, který má být vybrán jako *známý*, musí být *dočasný* bod s minimální hodnotou T .

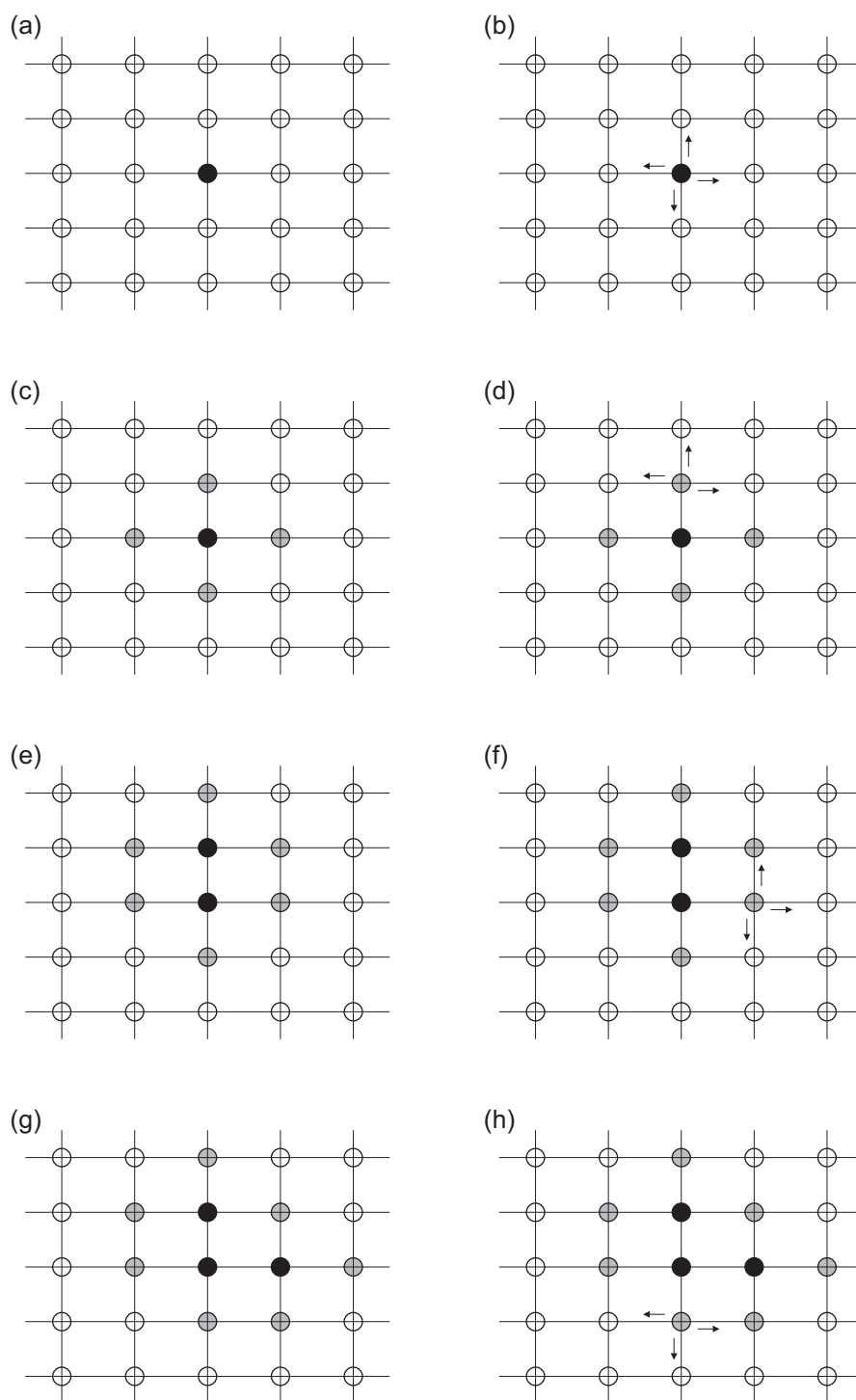
Do úzkého pásu *dočasných* bodů potom budou přidány body sousedící s právě označeným *známým* bodem a podle schématu (3.12) jim budou vypočítány hodnoty T . Pokud některý sousední bod již je označený jako *dočasný*, pak je hodnota T v tomto bodě vypočítána znovu. Mohou nastat dvě možnosti:

- Nově vypočítaná hodnota T je menší než původní hodnota pro tento bod. Pak je původní hodnota T nahrazena nově vypočítanou hodnotou.
- Nově vypočítaná hodnota T je větší než původní hodnota pro tento bod. Pak původní hodnota T zůstává.

Výpočet hodnot T tedy probíhá analogicky se šířením pohybující se hranice – hodnota T v jednom bodě je vždy označena jako *známá* a dojde k přepočítání hodnot T v sousedních bodech. Nikdy není nutné se vracet a znovu přepočítávat hodnoty T v bodech označených jako *známé* – tyto hodnoty po celý zbytek výpočtu zůstávají neměnné. Výpočet funkce T tedy probíhá bez iterací.

Stručný algoritmus

1. Body na počáteční pozici hranice se označí jako *známé* a hodnota T se pro tyto body nastaví na nulu.



Obrázek 3.3: Schéma metody Fast Marching.
 Černě označeny *známé*, šedě *dočasné* a bíle *vzdálené* body.

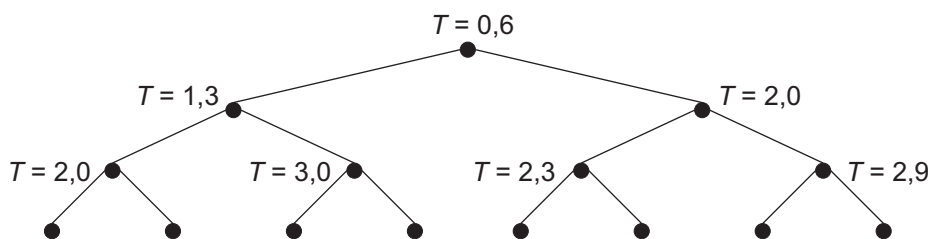
2. Body, které přímo sousedí se *známými* body, se označí jako *dočasné*.
3. Všechny ostatní body se označí jako *vzdálené*.
4. Následuje cyklus:
 - (i) Nechť A je *dočasný* bod o minimální hodnotě T .
 - (ii) Bod A je přidán do *známých* a odebrán z *dočasných*.
 - (iii) Všechny body, které sousedí s A a nejsou *známé*, se označí jako *dočasné*. Pokud byl bod *vzdálený*, odebere se z této množiny bodů.
 - (iv) Všem *dočasným* bodům přímo sousedícím s A je podle rovnice (3.12) přepočítána hodnota T .

Optimální řazení bodů

K tomu, aby tento algoritmus byl efektivní, je nutné navrhnout rychlý způsob nalezení *dočasného* bodu s minimální hodnotou T . Jako nejvhodnější přístup se podle [11] ukázala aplikace heap sortu. V tomto případě se používá datová struktura *min-halda*, ve které se pro každý *dočasný* bod uchovává hodnota T v tomto bodě spolu s údaji o umístění bodu ve čtvercové síti.

Tato min-halda je úplný binární strom¹, který splňuje vlastnost, že hodnota T v libovolném uzlu je menší nebo rovna hodnotám T v potomcích tohoto uzlu. V praxi je ovšem efektivnější reprezentovat haldu sekvenčně jako pole – platí, že pokud je rodič uchováván na pozici k , pak jeho potomky lze nalézt na pozicích $2k$ a $2k+1$. Naopak k prvku na pozici k lze nalézt rodiče na pozici $k/2$. Z definice min-haldy plyne, že *dočasný* bod s minimální hodnotou T je uchováván v kořeni stromu, v případě reprezentace polem tedy na pozici $k = 1$.

¹U čtenáře předpokládám znalost základních pojmů z teorie grafů – viz například [1].



Obrázek 3.4: Příklad min-haldy.

Algoritmus metody Fast Marching s využitím heap sortu pracuje následujícím způsobem. Nejprve se v min-haldě hledá *dočasný* bod s minimální hodnotou T . Tato operace `FindSmallest` zahrnuje smazání kořene haldy a úpravu haldy pomocí operace `DownHeap` tak, aby uspořádání zbylých prvků opět splňovalo vlastnost haldy. Dále se pokračuje přidáváním nových *dočasných* bodů do haldy za použití operace `Insert` – tato operace zvětší velikost haldy o jedna a pomocí operace `UpHeap` dojde ke kontrole a případné úpravě haldy tak, aby uspořádání zbylých prvků opět splňovalo vlastnost haldy. Operaci `Upheap` je nutné také provést vždy, když měníme hodnotu T bodu, který už prvkem haldy je.

Podrobnosti týkající se implementace haldy a operací, které jsou používány při práci s haldou, lze nalézt například v [12].

Výpočetní složitost

Operace `DownHeap` a `UpHeap` v nejhorším případě přenášejí prvek od kořene haldy až k jejím listům nebo naopak. Toto trvá $\mathcal{O}(\log_2 M)$ času za předpokladu, že v haldě je M prvků. Halda jako úplný binární strom totiž zůstává vždy vyvážená. Operace hledání sousedních prvků kořene haldy má – při uchování údajů o umístění bodu ve čtvercové síti – časovou složitost $\mathcal{O}(1)$. Jelikož změna hodnoty jednoho prvku a následné přeuspořádání haldy má náročnost $\mathcal{O}(\log_2 M)$, na síti o M bodech je pak celkový počet operací $\mathcal{O}(M \log_2 M)$.

Představme si dvojrozměrnou síť o $N \times N$ bodech. Metoda Fast Marching tedy snižuje celkový počet operací na $\mathcal{O}(N^2 \log_2 N)$.

Schémata vyšších řádů

Metoda Fast Marching, tak jak ji používám ve své práci, je metodou prvního řádu, neboť pro aproximaci gradientu používá pouze jednoduché diferenční schéma prvního řádu. Samozřejmě je možné odvodit a aplikovat i schémata vyšších řádů a tím zvyšovat přesnost výpočtu funkce T . Na druhou stranu se mohou během výpočtu vyskytnout problematická místa, kde není dobré používat schémata vyšších řádů. Například v [11] je v souladu s touto zásadou odvozeno schéma druhého řádu, které se v případě potřeby automaticky přepíná na schéma prvního řádu. Problematikou schémat vyšších řádů jsem se ale zatím nezabývala.

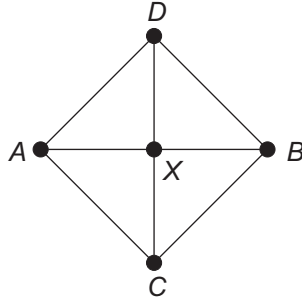
3.1.5 Fast Marching na TIN

Zbývá poslední krok – upravit algoritmus metody Fast Marching tak, aby bylo možné metodu používat nejen na rovinné čtvercové síti, ale i na TIN. Základní princip rozšířené metody vysvětlíme na speciální jednoduché trojúhelníkové síti a následně myšlenku aplikujeme na výpočet funkce T na obecné TIN.

Schéma pro jednoduchou trojúhelníkovou síť

Uvažujme jednoduchou trojúhelníkovou síť vzniklou z rovinné čtvercové sítě jako na obrázku 3.5.

Zaměříme se na trojúhelník tvořený body X , A a C s hodnotami T_X , T_A a T_C a předpokládejme, že hodnoty T_A a T_C jsou *známé* a že hodnotu T_X chceme



Obrázek 3.5: Jednoduchá trojúhelníková síť.

určit. Představme si, že hodnoty T v těchto třech bodech definují jakousi rovinu, kartézský souřadnicový systém má počátek v bodě X a mezi body původní čtvercové výpočetní sítě je krok h , tedy $h = |AX| = |CX|$. Rovnice roviny vyjádřené jako funkce dvou proměnných pak je

$$\left(\frac{T_X - T_A}{h}\right)x + \left(\frac{T_X - T_C}{h}\right)y + T_X = T(x, y). \quad (3.13)$$

Gradient této funkce je

$$\nabla T = \left(\frac{T_X - T_A}{h}, \frac{T_X - T_C}{h}\right). \quad (3.14)$$

Řešíme eikonalovou diferenciální rovnici $|\nabla T| F = 1$, musí tedy platit

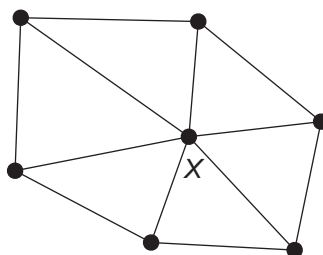
$$\left(\frac{T_X - T_A}{h}\right)^2 + \left(\frac{T_X - T_C}{h}\right)^2 = \frac{1}{F^2}. \quad (3.15)$$

Lze říci, že hledanou hodnotou $T_X \geq T_A, T_C$ zdviháme rovinu tak, aby velikost gradientu byla rovna $1/F$. Výsledná kvadratická rovnice pro výpočet hodnoty T_X je nakonec stejná jako při výpočtu funkce T na rovinné čtvercové

síti. Všimněme si také, že v tomto případě vektor záporně vzatého gradientu směřuje z bodu X do trojúhelníku, ze kterého se počítá hodnota T_X . Podle [11] je tato podmínka důležitá pro zajištění monotónnosti funkce T . V případě obecné TIN ovšem její splnění není automaticky zajištěno a podmínku je nutno ověřovat.

Schéma pro ostroúhlou TIN

Ostroúhlou TIN rozumějme TIN, jejíž trojúhelníky nejsou tupoúhlé. Uvažujme trojúhelníkovou síť jako na obrázku 3.6.

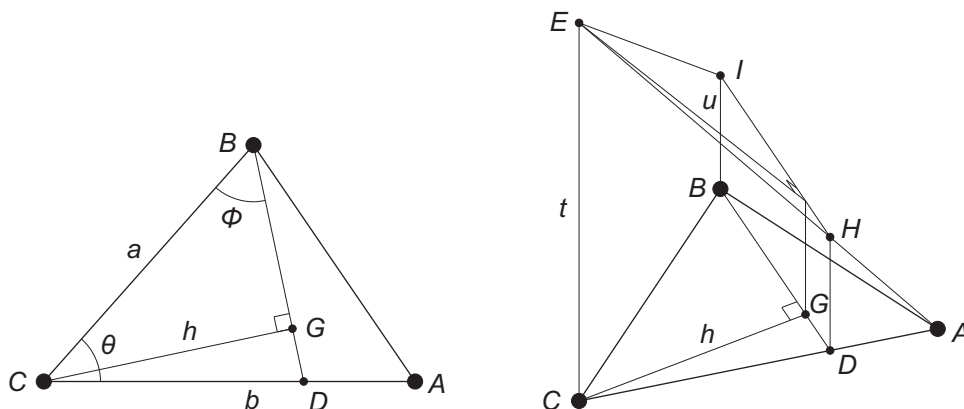


Obrázek 3.6: Skupina trojúhelníků okolo společného centrálního bodu X .

V obecnějším případě ostroúhlé TIN může velké množství trojúhelníků sdílet centrální bod (viz bod X na obrázku 3.6). Pomocí následujícího postupu, inspirovaného jednoduchou trojúhelníkovou sítí z předchozí části, se pokoušíme vypočítat vhodnou hodnotu T pro centrální bod z každého trojúhelníku, který má vrchol v tomto centrálním bodě. Může ovšem dojít k situaci, že z různých trojúhelníků obdržíme více různých přípustných hodnot T a musíme tedy jednu z nich vybrat – v tomto případě volíme minimum z možných hodnot T .

Uvažujme netupoúhlý trojúhelník ABC (viz obrázek 3.7), ve kterém chceme určit $T(C)$. Předpokládejme, že $T(B) > T(A)$ a $u = T(B) - T(A)$. Pro $T(C)$

platí $T(C) = T(A) + t$.



Obrázek 3.7: Odvození vztahu pro výpočet funkce T na trojúhelníku.

Obrázek vlevo – rovinný pohled na trojúhelník ABC , označení některých bodů, úseček a úhlů.

Obrázek vpravo – prostorový pohled na trojúhelník ABC , označení dalších bodů a úseček. Platí $t = |CE| = T(C) - T(A)$ a $u = |BI| = T(B) - T(A)$. Body A, E a I definují myšlenou rovinu funkce T „nad“ trojúhelníkem ABC .

Pro myšlenou rovinu funkce T „nad“ trojúhelníkem ABC platí

$$|\nabla T| = \frac{t - u}{h}, \quad (3.16)$$

kde h je výška v trojúhelníku BCD . Hledáme tedy $t = |EC|$, které splňuje eikonalovou rovnici

$$\frac{t - u}{h} = \frac{1}{F}. \quad (3.17)$$

Označme $a = |BC|$ a $b = |AC|$. Z podobnosti trojúhelníků AEC a AHD plyne

$$\frac{t}{b} = \frac{|DH|}{|AD|}, \quad (3.18)$$

takže

$$|CD| = b - |AD| = b - \frac{bu}{t} = \frac{b(t-u)}{t}. \quad (3.19)$$

Z kosinové věty plyne

$$|BD|^2 = a^2 + |CD|^2 - 2a|CD| \cos \theta \quad (3.20)$$

a ze sinové věty

$$\sin \phi = \frac{|CD|}{|BD|} \sin \theta. \quad (3.21)$$

Potom z pravoúhlého trojúhelníka CBG dostáváme

$$h = a \sin \phi = a \frac{|CD|}{|BD|} \sin \theta = \frac{a|CD| \sin \theta}{\sqrt{a^2 + |CD|^2 - 2a|CD| \cos \theta}}. \quad (3.22)$$

Po dosazení za h z rovnice (3.17) a za $|CD|$ z rovnice (3.19) a jednoduchých úpravách máme kvadratickou rovnici pro t :

$$\begin{pmatrix} F^2 (a^2 + b^2 - 2ab \cos \theta) t^2 + \\ + 2buF^2 (a \cos \theta - b) t + \\ + b^2 (u^2 F^2 - a^2 \sin \theta) \end{pmatrix} = 0. \quad (3.23)$$

Řešení t musí splňovat nerovnost $t > u$ a dále požadujeme, aby záporně vzatý vektor gradientu směřoval dovnitř trojúhelníku, což zajistíme pomocí této nerovnice:

$$a \cos \theta < \frac{b(t-u)}{t} < \frac{a}{\cos \theta}. \quad (3.24)$$

Dostáváme tedy následující způsob výpočtu:

$$\text{Pokud } u < t \text{ a zároveň } a \cos \theta < \frac{b(t-u)}{t} < \frac{a}{\cos \theta},$$

$$\text{pak } T(C) = \min \{T(C), T(A) + t\};$$

$$\text{jinak } T(C) = \min \left\{ T(C), T(A) + \frac{b}{F}, T(B) + \frac{a}{F} \right\}.$$

Tento předpis pro výpočet nových hodnot funkce T ve vrcholech TIN je určitou analogií schématu (3.12), které jsme používali pro výpočet nových hodnot funkce T v bodech rovinné čtvercové výpočetní sítě.

Spojíme-li tedy dohromady princip metody Fast Marching (dynamické rozdělení bodů na *známé*, *dočasné* a *vzdálené*, heap sort atd. – viz podrobný popis metody v předchozí části) s nově odvozeným způsobem výpočtu hodnot funkce T , dostáváme metodu Fast Marching pro výpočet funkce T na ostroúhlé TIN.

Rozšíření pro obecnou TIN

V [11] je velmi stručně naznačeno, jak rozšířit metodu Fast Marching tak, aby bylo možné ji použít i pro obecně tupouhrou TIN. Nejjednodušší způsob je přetvořit tupouhrou TIN na ostroúhrou a tímto převést problém na již vyřešený případ ostroúhlé TIN. Jinou možností je pomocí rozvinování sousedních trojúhelníků do jedné roviny přidávat jakési *virtuální hrany*, které tupé úhly v trojúhelnících rozdělí na dva ostré. Problematikou tupouhrou TIN jsem se ale zatím nezabývala.

3.2 Vlastní výpočet nejkratší cesty

V úvodu této kapitoly bylo uvedeno, že výpočet nejkratší cesty vlastně znamená „návrat“ z cílového do výchozího bodu podél záporně vzatého gradientu časové funkce T – gradient udává směr největší změny funkce T a tedy i lokální směr nejkratší cesty.

Laicky řečeno – lze si představit, že bod na pohybující se hranici, který dosáhl cílového bodu, se celou dobu pohyboval ve směru gradientu funkce T , protože tímto směrem se pohybovala celá hranice. Zároveň platí, že trajektorie tohoto bodu byla nejkratší váženou cestou, protože čas cesty po této trajektorii byl minimální. Proto se nyní chceme vracet po stejné trajektorii, ale v opačném směru.

Matematicky řečeno – řešíme soustavu obyčejných diferenciálních rovnic

$$\frac{d\mathbf{X}(s)}{ds} = -\nabla T, \quad (3.25)$$

kde $\mathbf{X}(s)$ popisuje nejkratší cestu z cílového do výchozího bodu.

Pro řešení soustavy diferenciálních rovnic (3.25) na TIN jsem navrhla postup popsany v následujícím textu. Nejprve jsem se zabývala problémem aproximace gradientu funkce T a poté jsem řešila otázku výpočtu jednotlivých bodů hledané nejkratší cesty. Při aproximaci gradientu jsem funkci T uvažovala jako funkci pouze dvou proměnných – $T = T(x, y)$ – a zanedbávala jsem třetí rozměr TIN. Ráda bych v budoucnu odvodila metodu, která by aproximovala gradient funkce T přímo vzhledem k TIN.

3.2.1 Aproximace gradientu funkce T

K řešení diferenciální rovnice použijeme schéma druhého řádu s případným přepínáním na schéma prvního řádu.

Schéma prvního řádu

Funkci T pro jednotlivé trojúhelníky nahrazujeme rovinou o rovnici

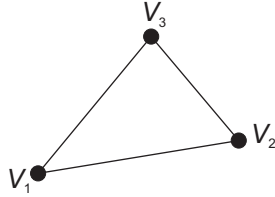
$$Ax + By + C = T, \quad (3.26)$$

kde koeficienty A , B a C určíme ze souřadnic vrcholů daného trojúhelníka (viz obrázek 3.8) a z hodnot funkce T v těchto vrcholech jako řešení soustavy rovnic

$$\begin{aligned} Ax_1 + By_1 + C &= T_1 \\ Ax_2 + By_2 + C &= T_2 \\ Ax_3 + By_3 + C &= T_3 \end{aligned} \quad (3.27)$$

Pro gradient pak platí

$$\nabla T \approx (A, B) \quad (3.28)$$



Obrázek 3.8: Trojúhelník pro výpočet ∇T .

Schéma druhého řádu

Nyní funkci T pro jednotlivé trojúhelníky nahrazujeme plochou druhého stupně o rovnici

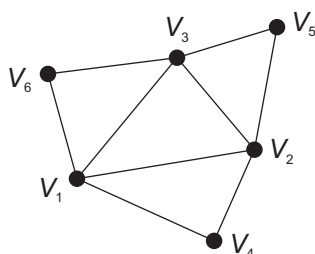
$$Ax^2 + By^2 + Cxy + Dx + Ey + F = T, \quad (3.29)$$

kde koeficienty A , B , C , D , E a F určíme ze souřadnic vrcholů daného trojúhelníka a tří okolních trojúhelníků (viz obrázek 3.9) a z hodnot funkce T v těchto vrcholech jako řešení soustavy rovnic

$$Ax_i^2 + By_i^2 + Cx_iy_i + Dx_i + Ey_i + F = T_i \quad i = 1, \dots, 6 \quad (3.30)$$

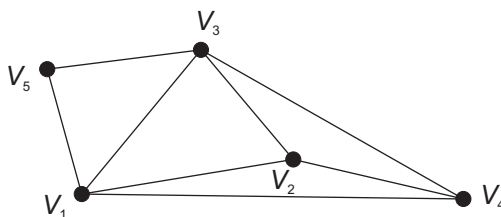
Pro gradient pak platí

$$\nabla T \approx (2Ax + Cy + D, 2By + Cx + E) \quad (3.31)$$



Obrázek 3.9: Čtveřice trojúhelníků pro výpočet ∇T .

Nesmí ovšem dojít k situaci jako na obrázku 3.10. Plocha by byla určena jen pěti body a soustava rovnic by měla nekonečně mnoho řešení. V této situaci je nutné vyvolat výpočet prostřednictvím schématu prvního řádu.



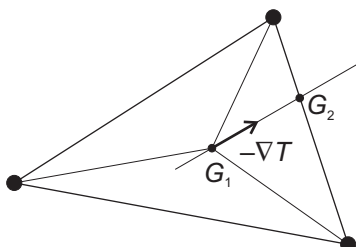
Obrázek 3.10: Situace, kdy není možné použít schéma druhého řádu.

3.2.2 Výpočet jednotlivých bodů nejkratší cesty

Nejkratší cestu do výchozího bodu pak získáme následujícím způsobem:

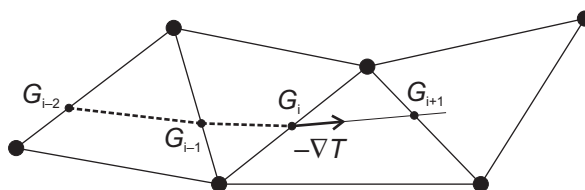
1. Zvolíme trojúhelník v TIN. Jeho těžiště bude bodem G_1 – prvním bodem nejkratší cesty. Pro tento trojúhelník vypočteme aproximaci gradientu funkce T . Bod G_1 a záporně vzatý vektor gradientu určují přímku, určíme tedy průsečík této přímky s některou ze stran výchozího

trojúhelníka. Nalezený průsečík je bodem G_2 – druhým bodem nejkratší cesty.



Obrázek 3.11: Počátek výpočtu jednotlivých bodů nejkratší cesty.

2. Máme určený bod G_i – i -tý bod nejkratší cesty a hledáme bod G_{i+1} . Bod G_i leží na hraně, která je společná dvěma trojúhelníkům – přes jeden trojúhelník jsme již nejkratší cestou přešli a nyní chceme přejít přes druhý. Podobně jako v předchozím případě určíme aproximaci gradientu funkce T pro trojúhelník, přes který chceme nyní přejít, a pomocí bodu G_i a záporně vzatého vektoru gradientu vyjádříme přímku a vypočteme její průsečík s některou ze zbylých dvou hran trojúhelníka. Nalezený průsečík je bodem G_{i+1} – dalším bodem nejkratší cesty.



Obrázek 3.12: Průběh výpočtu jednotlivých bodů nejkratší cesty.

3. Jsme-li již dostatečně blízko bodu, do kterého máme za úkol se dostat, pak tento bod přidáme jako poslední bod do hledané nejkratší cesty a ukončíme výpočet.

Kapitola 4

Implementace a výsledky

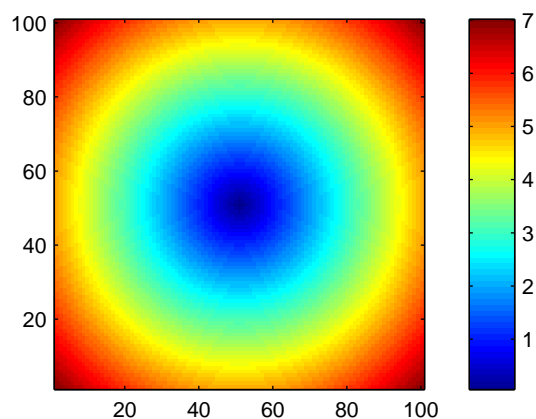
Vlastní implementace algoritmů popsaných v předchozím textu byla provedena v programu MATLAB. Pro práci s výsledky a pro jejich vizualizaci tedy bylo možné využít vestavěné funkce programu MATLAB, jako například funkce pro měření času `tic – toc` a vizualizační funkce `imagesc` pro znázornění čtvercové výpočetní sítě a `trisurf` pro znázornění trojúhelníkové sítě. Vytvořené programy jsou k dispozici na příloženém CD.

4.1 Výpočet funkce T na čtvercové síti

Výpočet funkce T probíhal na čtvercové síti 101×101 bodů. Krok mezi body výpočetní sítě byl zvolen jako $\Delta x = \Delta y = 0,1$. Jako počáteční bod byl uvažován bod uprostřed sítě – jeho souřadnice v síti jsou $i = j = 51$.

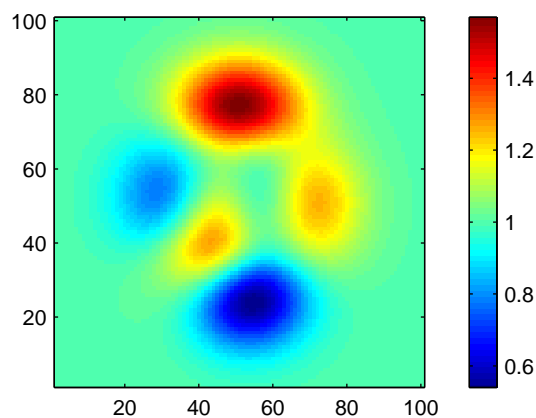
Numerické řešení rovnice $|\nabla T|F = 1$ bylo prováděno nejprve pomocí klasické iterační metody a poté pomocí metody Fast Marching – vždy jednou bez váhové funkce F (respektive s triviální verzí váhové funkce, kdy v každém bodě platí $F = 1$) a jednou s netriviální váhovou funkcí F (jako váhová funkce byla využita mírně modifikovaná funkce `peaks` programu MATLAB).

Pokud platí $F = 1$, pak by hodnota funkce T pro jednotlivé body měla být rovna jejich vzdálenosti od výchozího bodu. Vizualizace přesného řešení rovnice $|\nabla T| = 1$ na výše popsané výpočetní síti je na obrázku 4.1.



Obrázek 4.1: Přesné řešení rovnice $|\nabla T| = 1$.

Netriviální váhová funkce F použitá při některých výpočtech časové funkce T je znázorněna na obrázku 4.2. Připomeňme, že pro funkci F má platit $F > 0$.

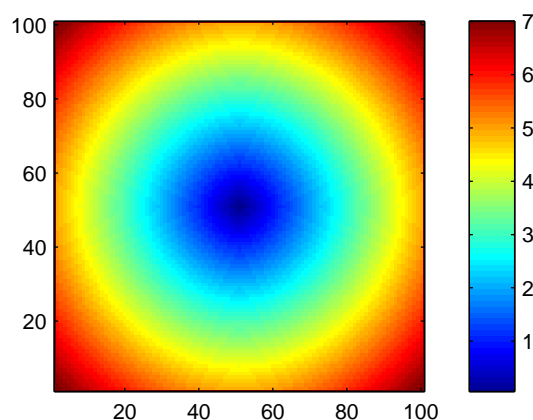


Obrázek 4.2: Váhová funkce F .

Výpočet byl sice prováděn zvlášť iterační metodou a zvlášť metodou Fast Marching, v následujících dvou částech ale budou ukázány pouze výsledky získané pomocí metody Fast Marching. Důvody, které k tomu vedly, jsou vysvětleny v podkapitole 4.1.3.

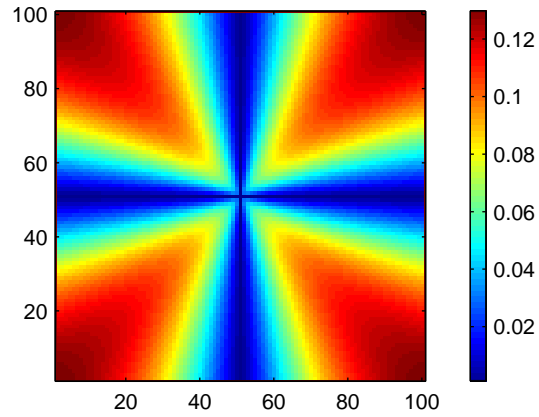
4.1.1 Výpočet bez váhové funkce F

Při numerickém výpočtu časové funkce T na rovinné čtvercové síti podle schématu (3.12) bez váhové funkce F jsme dostali výsledek, který je znázorněný na obrázku 4.3.



Obrázek 4.3: Výsledek numerického výpočtu funkce T na čtvercové síti.

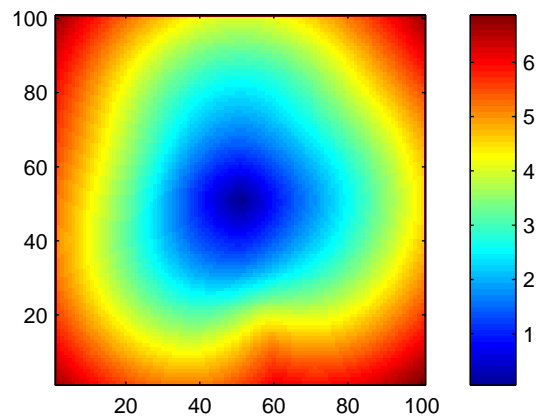
Odchylky tohoto řešení od analyticky vypočítaného řešení (viz obrázek 4.1) způsobené diskretizací úlohy jsou znázorněny na obrázku 4.4. Odchylky v jednotlivých bodech se liší v závislosti na poloze bodu ve výpočetní síti. Použitím schématu vyššího řádu by pravděpodobně došlo ke zmenšení těchto odchylek a ke zpřesnění výpočtu funkce T .



Obrázek 4.4: Odchylyky způsobené diskretizací úlohy.

4.1.2 Výpočet s váhovou funkcí F

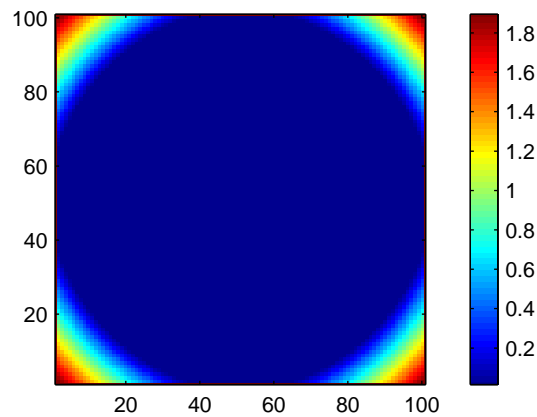
Při tomto výpočtu byla uvažována váhová funkce podle obrázku 4.2. Výsledek je vizualizovaný na obrázku 4.5. Při výpočtu s uvažováním netriviální váhové funkce bohužel neznáme dopředu správný výsledek a není tedy možné naše řešení s ničím porovnat.



Obrázek 4.5: Výpočet časové funkce T s uvažováním váhové funkce F .

4.1.3 Srovnání metod použitých pro výpočet funkce T

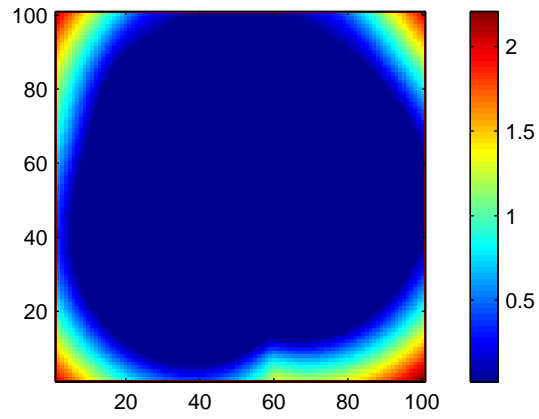
Výsledky získané iterační metodou a výsledky získané metodou Fast Marching byly (až na časovou náročnost) naprosto totožné, ale jen v určité oblasti výpočetní sítě. Rozdíly výsledků získaných těmito dvěma metodami vidíme na obrázcích 4.6 (výpočet bez váhové funkce F) a 4.7 (výpočet s váhovou funkcí F).



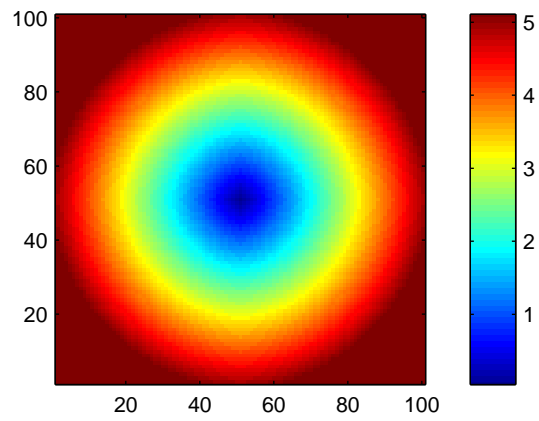
Obrázek 4.6: Rozdíly výsledků iterační metody a metody Fast Marching (1).

Iterační metoda totiž začíná selhávat v okamžiku, kdy se výpočet funkce T dostane příliš blízko k hranici výpočetní sítě. V případě výpočtu bez váhové funkce F to je při 84. iteraci a v případě výpočtu s váhovou funkcí F podle obrázku 4.2 to je při 75. iteraci. Ukončíme-li tedy výpočet po 83., respektive 74. iteraci, zůstanou některé oblasti výpočetní sítě nedopočítané (viz obrázky 4.8 a 4.9 – oproti obrázkům 4.3 a 4.5 mají bohužel mírně posunutou barevnou stupnici).

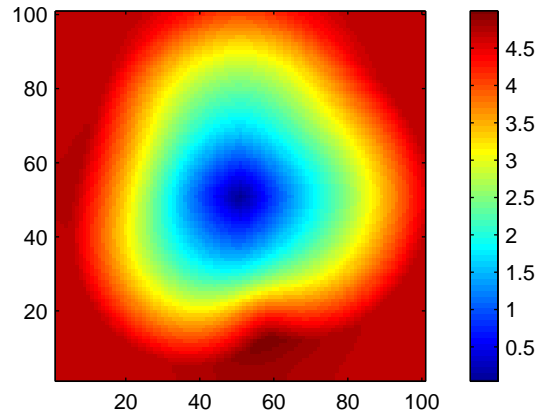
Tento problém by samozřejmě bylo možné řešit například zvětšením rozměru výpočetní sítě – nedopočítané oblasti by potom zůstaly vně původní sítě 101×101 bodů. To by ale vedlo k dalšímu zpomalení výpočtu, který už nyní



Obrázek 4.7: Rozdíly výsledků iterační metody a metody Fast Marching (2).



Obrázek 4.8: Iterační metoda bez váhové funkce F po 83 iteracích.



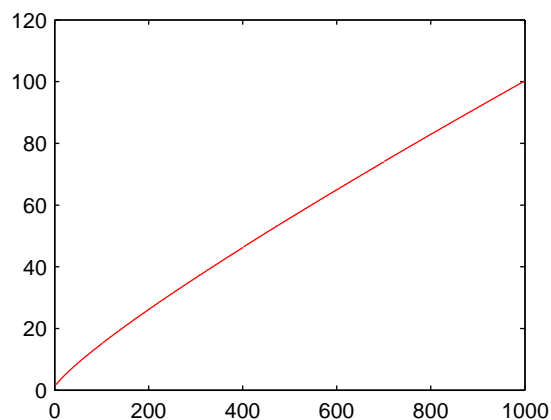
Obrázek 4.9: Iterační metoda s váhovou funkcí F po 74 iteracích.

je oproti metodě Fast Marching výrazně pomalejší.

Výpočet funkce T na síti 101×101 bodů metodou Fast Marching trvá přibližně 5,7 sekundy, zatímco výpočet iterační metodou trvá zhruba 81 sekund při 83 iteracích (bez váhové funkce F) a 71 sekund při 74 iteracích (s váhovou funkcí F)¹. Iterační metoda na síti pouhých 101×101 bodů je tedy mnohem pomalejší než metoda Fast Marching.

Připomeňme, že teoreticky odvozená výpočetní složitost na síti $N \times N$ bodů je $\mathcal{O}(N^2 \log_2 N)$ pro metodu Fast Marching a $\mathcal{O}(N^3)$ pro iterační metodu. Funkce $N/\log_2 N$ tedy řádově udává, kolikrát pomalejší je výpočet iterační metodou oproti výpočtu metodou Fast Marching v závislosti na rozměru výpočetní sítě. Z grafu této funkce (viz obrázek 4.10) je vidět, že se zvětšujícím se N náročnost výpočtu iterační metodou roste mnohem rychleji než náročnost výpočtu pomocí metody Fast Marching.

¹Čas výpočtu změřený pomocí funkcí `tic` – `toc` je sice pouze orientační informace (čas výpočtu nezávisí pouze na výpočetní složitosti metody, ale například i na kvalitě programového řešení), nicméně pro naše účely postačuje.



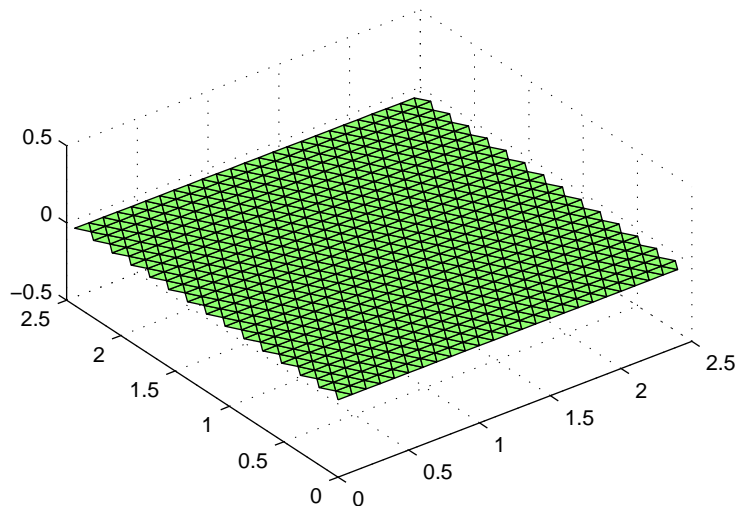
Obrázek 4.10: Graf funkce $N/\log_2 N$.

Z výše uvedeného tedy vyplývá, že vzhledem k výpočetní náročnosti iterační metody oproti metodě Fast Marching a také s přihlédnutím k problémům s výpočtem hodnot funkce T na okrajích výpočetní sítě při použití iterační metody není příliš přínosné se iterační metodou dále zabývat. Při výpočtu funkce T na TIN se tedy budeme soustředit pouze na metodu Fast Marching.

4.2 Výpočet funkce T na TIN

Časová funkce T na TIN byla stejně jako v předchozím případě rovinné pravoúhlé výpočetní sítě určována jako řešení eikonalové rovnice $|\nabla T|F = 1$. Numerický výpočet funkce T probíhal pomocí metody Fast Marching, mírně modifikované pro TIN. Podrobný popis této metody lze nalézt v podkapitolách 3.1.4 a 3.1.5.

Pro testovací účely byla použita jednoduchá rovinná TIN složená z rovnostranných trojúhelníků – viz obrázek 4.11.



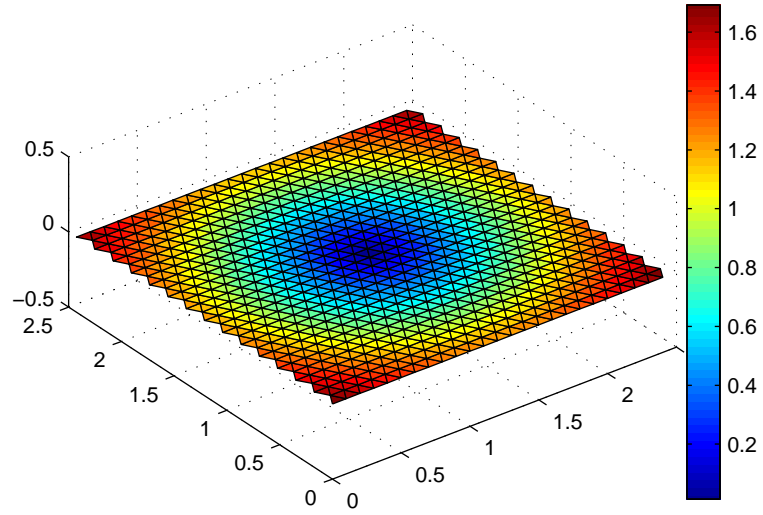
Obrázek 4.11: Jednoduchá rovinná TIN.

4.2.1 Výpočet bez váhové funkce F

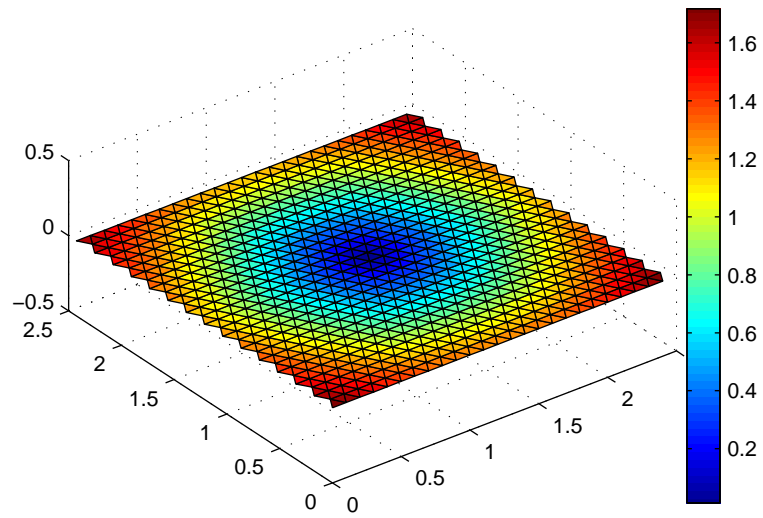
Budeme-li jako výchozí bod uvažovat vrchol uprostřed této TIN a budeme-li předpokládat, že váhová funkce $F = 1$, pak by hodnota časové funkce T v každém vrcholu TIN měla být rovna vzdálenosti tohoto vrcholu od výchozího bodu. Vizualizace přesného řešení je na obrázku 4.12. Vizualizace řešení eikonalové rovnice metodou Fast Marching bez uvažování váhové funkce F je na obrázku 4.13. Odchyly těchto dvou řešení jsou znázorněny na obrázku 4.14. Je vidět, že odchyly jsou opět závislé na poloze bodů v TIN.

4.2.2 Výpočet s váhovou funkcí F

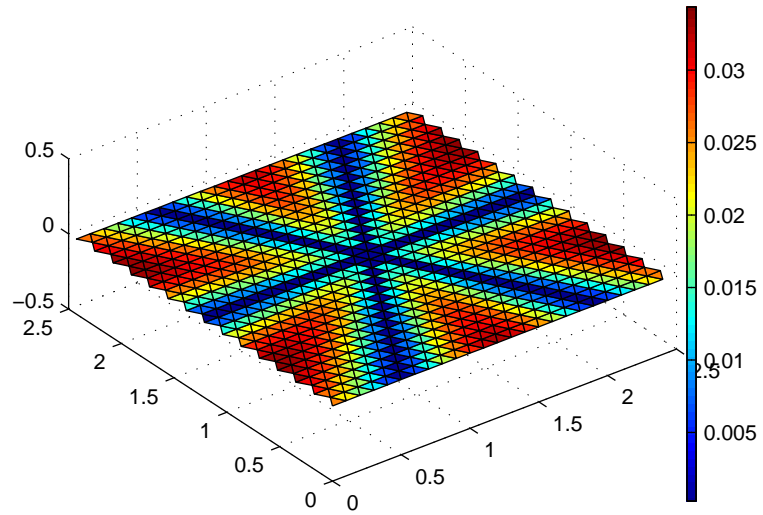
Můžeme vyzkoušet i výpočet s uvažováním váhové funkce F . Pro vytvoření váhové funkce opět použijeme funkci `peaks` programu MATLAB. Funkce F je znázorněná na obrázku 4.15. Výsledná funkce T je na obrázku 4.16.



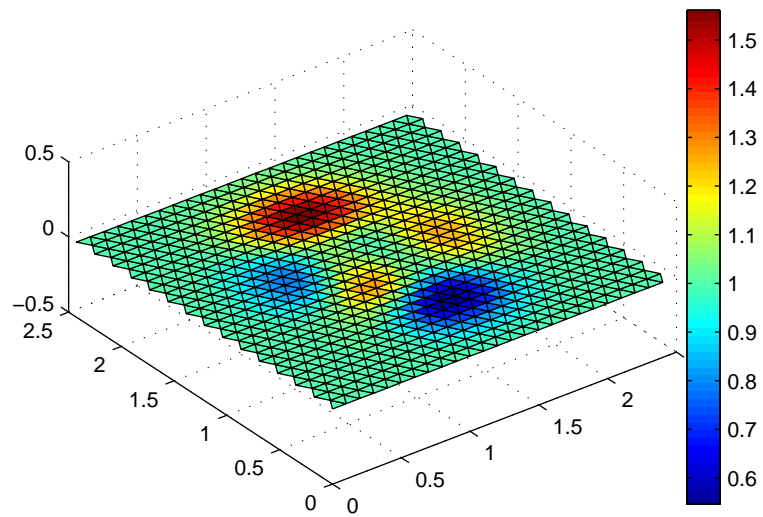
Obrázek 4.12: Přesné řešení rovnice $|\nabla T| = 1$ na jednoduché TIN.



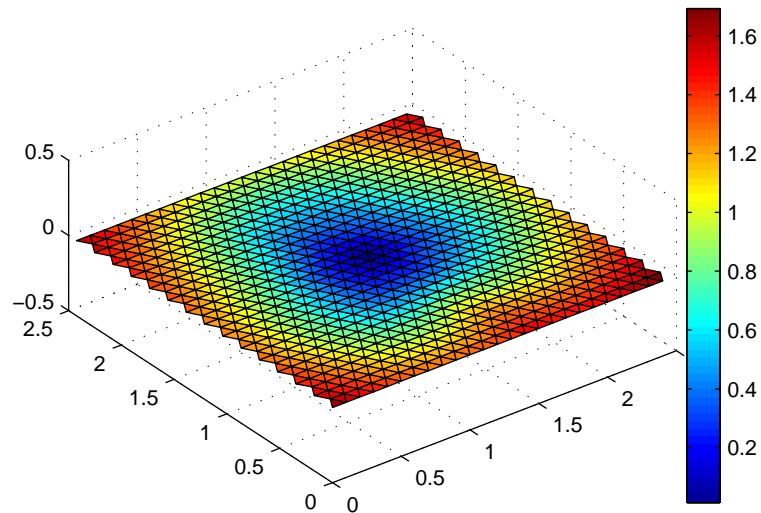
Obrázek 4.13: Výsledek numerického výpočtu funkce T na TIN.



Obrázek 4.14: Odchyly mezi přesným řešením a metodou Fast Marching.



Obrázek 4.15: Váhová funkce F na TIN.



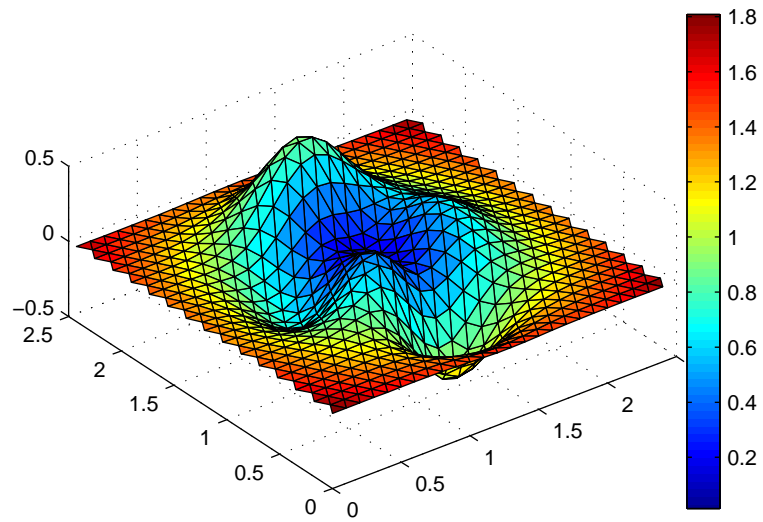
Obrázek 4.16: Výpočet časové funkce T na TIN s úvahou váhové funkce F .

4.2.3 Další ukázky

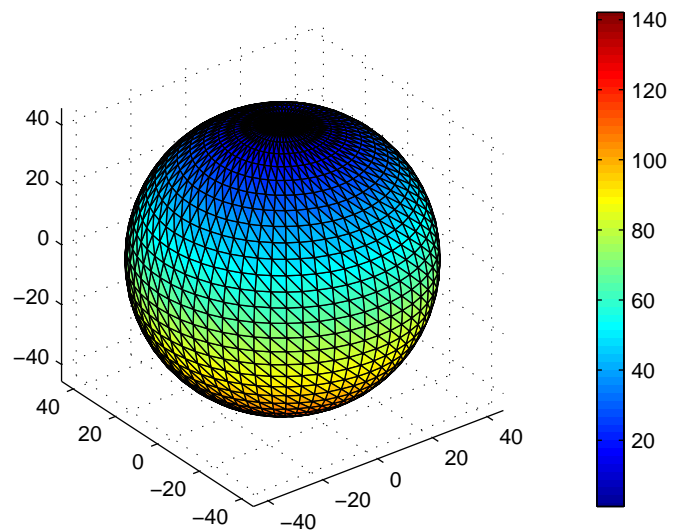
Nemusíme se samozřejmě omezovat jen na rovinnou TIN, algoritmus pro výpočet časové funkce T je určen pro prostorovou TIN.

Ukázka výpočtu funkce T na prostorové TIN je na obrázku 4.17. Při vytváření této TIN byla opět využita funkce `peaks` programu MATLAB.

Další ukázku – výpočet funkce T na triangulované kulové ploše – je možné si prohlédnout na obrázku 4.18.



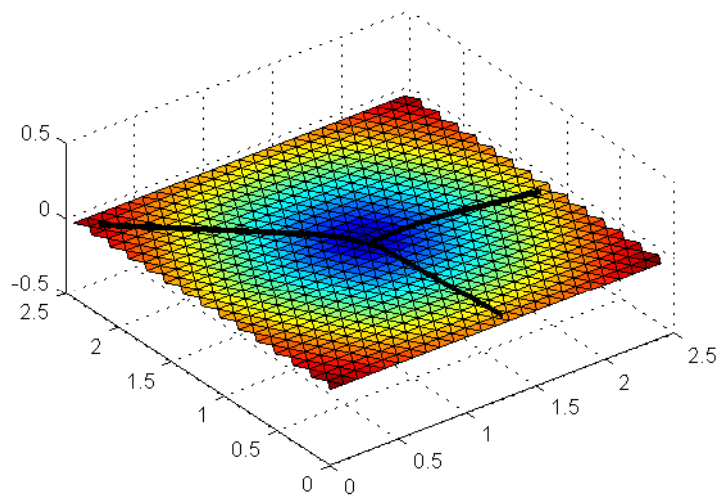
Obrázek 4.17: Výpočet časové funkce T na prostorové TIN.



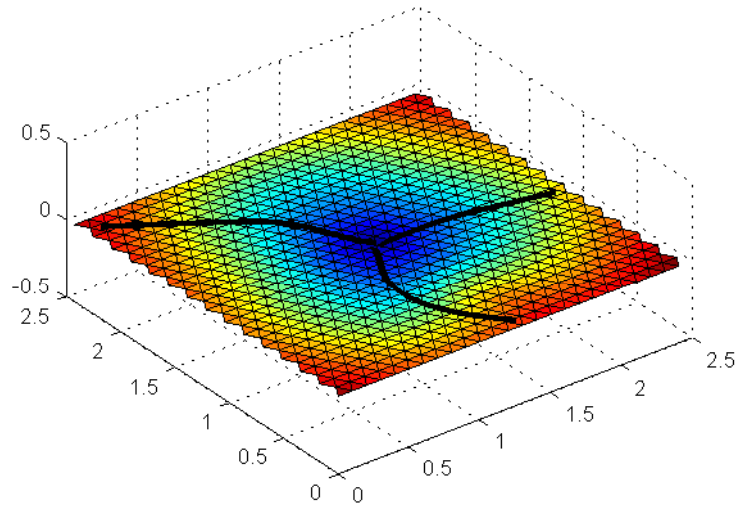
Obrázek 4.18: Výpočet časové funkce T na triangulované kulové ploše.

4.3 Výpočet nejkratší cesty na TIN

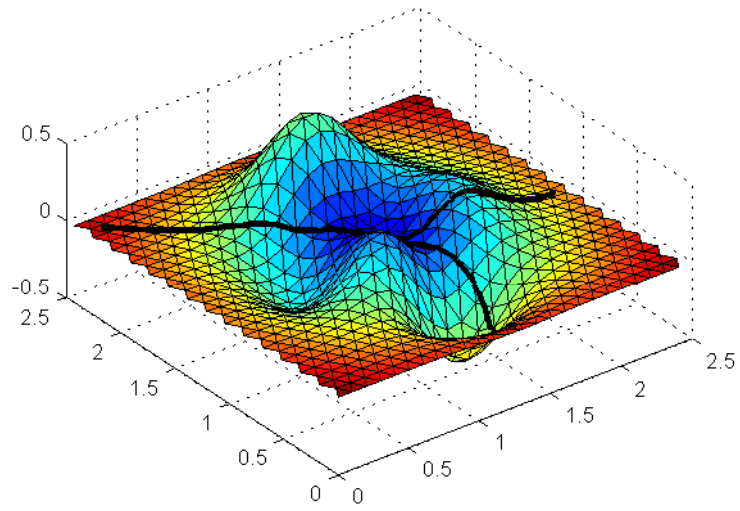
Chybí už jen poslední krok – pomocí vypočtené funkce T začít konstruovat nejkratší cesty na TIN podle algoritmu popsaného v podkapitole 3.2. Výsledky jsou vizualizovány na následujících obrázcích.



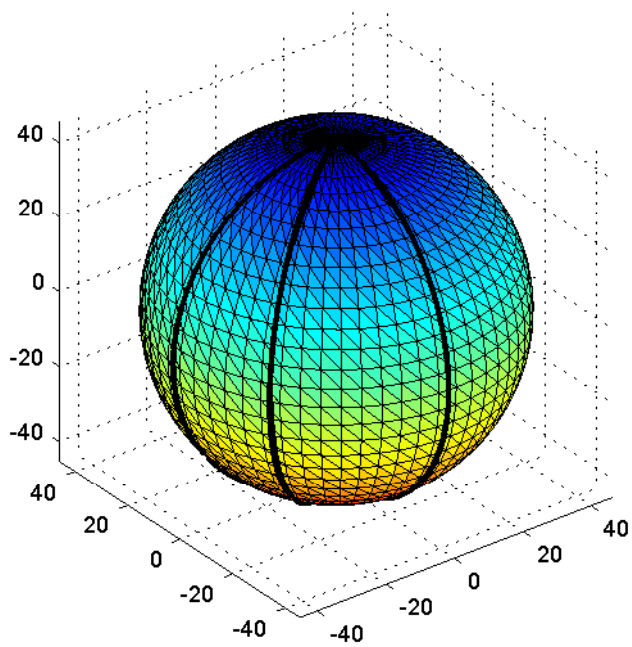
Obrázek 4.19: Nejkratší cesty na rovinné TIN.



Obrázek 4.20: Nejkratší vážené cesty na rovinné TIN.



Obrázek 4.21: Nejkratší cesty na prostorové TIN.



Obrázek 4.22: Nejkratší cesty na triangulované kulové ploše.

Kapitola 5

Závěr

5.1 Shrnutí a zhodnocení

Cílem mojí práce bylo navrhnout a implementovat numerické metody pro hledání nejkratších cest na TIN. Řešení probíhalo ve dvou krocích. Nejprve jsem řešením eikonalové parciální diferenciální rovnice

$$|\nabla T|F = 1, \tag{5.1}$$

kde F je daná váhová funkce, získala časovou funkci T , která pro každý vrchol TIN udává minimální čas cesty do výchozího bodu, a následně jsem řešením soustavy obyčejných diferenciálních rovnic

$$\frac{d\mathbf{X}(s)}{ds} = -\nabla T \tag{5.2}$$

získala funkci $\mathbf{X}(s)$, která popisuje nejkratší cestu mezi dvěma body na TIN.

Při výpočtu časové funkce T jsem postupovala od jednoduchého ke složitějšímu. Nejdříve jsem rovnici (5.1) numericky řešila na dvojrozměrné čtvercové

výpočetní síti, a to dvěma různými způsoby – klasickou iterační metodou a metodou Fast Marching. Metoda Fast Marching se ukázala jako velice efektivní, a proto jsem při výpočtu funkce T na TIN používala již pouze tuto metodu.

Pro následné numerické řešení rovnice (5.2) na TIN jsem navrhla vlastní metodu, která je podrobně popsána v podkapitole 3.2. Nejkratší cesta mezi dvěma body na TIN je v mém pojetí určena jako posloupnost diskrétních bodů na hranách TIN.

Domnívám se tedy, že cíl práce se podařilo splnit.

Snad nejdůležitějším zjištěním plynoucím z mé práce je to, že při numerickém řešení reálného problému je klíčové si všimnout vlastností tohoto problému a snažit se jich maximálně využít při návrhu numerického řešení – řešení „šité na míru“ konkrétnímu problému totiž může být mnohem efektivnější než jiné univerzální řešení.

Jsem si vědoma toho, že problematika nejkratších cest je velice komplikovaná a že já jsem ve své bakalářské práci zatím pronikla jen do nepatrné hloubky tohoto problému. Svou bakalářskou práci tedy v této chvíli chápu především jako základ k budoucí diplomové práci.

5.2 Výhled do budoucna

V průběhu zpracovávání bakalářské práce se objevilo několik podnětů, kterými je možné se dále zabývat.

- Odvození a implementace metod vyšších řádů pro výpočet časové funkce T na rovinné čtvercové výpočetní síti i na TIN.

- Vylepšení metody pro výpočet funkce T na TIN tak, aby bylo možné ji použít i pro obecně tupoúhlou TIN.
- Použití nových přístupů k výpočtu funkce T , jako je například metoda Level Set popsaná v [11].
- Navržení nových metod pro numerické řešení rovnice (5.2) na TIN – zejména se pokusit najít lepší způsob aproximace ∇T .
- Modifikace parametrů již implementovaných metod (například kroků Δx a Δy pravoúhlé výpočetní sítě nebo velikosti trojúhelníků v TIN) a sledování vlivu těchto změn na výpočet.
- Optimalizace implementace jednotlivých metod.
- Testování a porovnávání dosavadních i nových metod a jejich hlubší analýza. Pro testování přesnosti jednotlivých metod je například možné použít Clairautovu větu nebo možnost analytického vyjádření rovnice geodetiky v rovině a na kouli atd.

Literatura

- [1] ČADA, Roman – KAISER, Tomáš – RYJÁČEK, Zdeněk. *Diskrétní matematika*. 1. vydání. Západočeská univerzita. Plzeň, 2004. ISBN 80-7082-939-7.
- [2] JEDLIČKA, Karel. *Úvod do geografických informačních systémů* [online]. Západočeská univerzita. Plzeň, 2006. Dostupné z: [⟨http://gis.zcu.cz/studium/ugi/elearning/index1.htm⟩](http://gis.zcu.cz/studium/ugi/elearning/index1.htm).
- [3] JEŽEK, František. *Diferenciální geometrie. Pomocný učební text* [online]. Západočeská univerzita. Plzeň, 2005. Dostupné z: [⟨http://home.zcu.cz/~jezek/www/DG/DG_Final-05.pdf⟩](http://home.zcu.cz/~jezek/www/DG/DG_Final-05.pdf).
- [4] KIMMEL, Ron – SETHIAN, James. *Computing Geodesic Paths on Manifolds*. Proceedings of National Academy of Sciences, USA, 95(15): 8431-8435, 1998.
- [5] KUFNER, Alois. *Přednášky předmětu Obyčejné diferenciální rovnice*. Západočeská univerzita. Plzeň, 2005.
- [6] LÁVIČKA, Miroslav. *Geometrie 1. Pomocný učební text* [online]. Západočeská univerzita. Plzeň, 2005. Dostupné z: [⟨http://home.zcu.cz/~lavicka/subjects/G1/texty/G1_texty.pdf⟩](http://home.zcu.cz/~lavicka/subjects/G1/texty/G1_texty.pdf).
- [7] MÍKA, Stanislav – BRANDNER, Marek. *Numerické metody I*. 1. vydání. Západočeská univerzita. Plzeň, 2000. ISBN 80-7082-619-3.

- [8] NOVÁK, Pavel. *Vyšší geodézie. Přednáškové fólie* [online]. Západočeská univerzita. Plzeň, 2006. Dostupné z: http://gis.zcu.cz/studium/gev/prednaskove_folie/.
- [9] PŘÍKRYL, Petr – BRANDNER, Marek. *Numerické metody II*. 1. vydání. Západočeská univerzita. Plzeň, 2000. ISBN 80-7082-699-1.
- [10] SETHIAN, James. *Fast Marching Methods*. SIAM Review. Vol. 41, No. 2, pp. 199–235. Society for Industrial and Applied Mathematics. Philadelphia, 1999.
- [11] SETHIAN, James. *Level Set Methods and Fast Marching Methods. Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science*. 2nd edition. Cambridge University Press. Cambridge, 1999. ISBN 0-521-64204-3.
- [12] ŠAFAŘÍK, Jiří. *Počítače a programování 2* [online]. Západočeská univerzita. Plzeň, 2005. Dostupné z: <http://www.kiv.zcu.cz/~netrvalo/vyuka/ppa2/ekniha/online/>.