Slovenská technická univerzita v Bratislave
Stavebná fakulta

Evidenčné číslo: SvF-5342-87685

Numerický výpočet funkcie času prvého príchodu

Bakalárska práca

2019

Katarína Lacková

Slovenská technická univerzita v Bratislave
Stavebná fakulta

EVIDENČNÉ ČÍSLO: SvF-5342-81233

Numerický výpočet funkcie času prvého príchodu

Bakalárska práca

Študijný program:	Matematicko-počítačové modelovanie
Študijný odbor:	9.1.9. Aplikovaná matematika
Školiace pracovisko:	Katedra matematiky a deskriptívnej geometrie
Vedúci práce:	doc. RNDr. Peter Frolkovič, PhD.

Bratislava 2019

Katarína Lacková

Čestné prehlásenie

Čestne vyhlasujem, že bakalársku prácu s názvom: *Numerický výpočet funkcie času prvého príchodu* som vypracovala samostatne, na základe konzultácií a s použitím uvedených informačných zdrojov a literatúry.

Bratislava 13.4.2019

Katarína Lacková

Poďakovanie

Chcela by som sa poďakovať vedúcemu práce, doc. RNDr. Petrovi Frolkovičovi, PhD. za jeho ochotu, trpezlivosť a množstvo cenných rád, bez ktorých by tvorba tejto práce bola oveľa náročnejšia a ani zďaleka nie tak poučná.

Abstrakt

Abstrakt: Táto práca sa zaoberá hľadaním funkcie času prvého príchodu rozpínajúcej sa krivky pomocou metód úrovňovej množiny. Pracuje sa na modeli zahŕňajúcom aj krivosť pohybujúcej sa krivky. Hlavným cieľom práce je porovnanie nestacionárnej a stacionárnej metódy riešenia funkcie času prvého príchodu pre model uvažujúci krivosť. Na záver sa získané poznatky aplikujú pri riešení funkcie času prvého príchodu rozpínajúceho sa frontu lesného požiaru.

Kľúčové slová: metódy úrovňovej množiny, čas prvého príchodu, vzdialenostná funkcia so znamienkom

Abstract

Abstract: This work deals with level set methods in order to find the first arrival time function of a moving curve. The model includes also a curvature of the propagating curve. The main objective of the work is to compare solutions to both initial and boundary value formulation considering the curvature term. Finally the observations are used to solve the first arrival time of a forest fire front propagation.

Keywords: level set methods, first arrival time, signed distance function

Obsah

Úvod

Problému modelovania pohybu kriviek a jeho aplikáciam, napríklad šíreniu frontu lesného požiaru, sa po mnohé roky venujú vedci a matematici na Katedre matematiky a deskriptívnej geometrie Stavebnej fakulty Slovenskej technickej univerzity v Bratislave. V tejto práci nadviažeme na nimi získané poznatky.

Významnými publikáciami modelujúcimi pohyb lesných požiarov sú napríklad [8] a [9]. V oboch sú popísané takzvané *Langrangeovské metódy* a samotné modely v nich zahŕňajú okrem iných faktorov aj krivosť.

Našim cieľom bude riešiť tento problém pomocou *Metód úrovňovej množiny*. Motivujeme sa preto prácami [12], [10] a [11], ktoré prinášajú rôzne techniky modelovania pohybu kriviek práve Metódami úrovňovej množiny. Podobne ako v práci [11] opíšeme pohyb kriviek prostredníctvom *funkcie času prvého príchodu*. Budeme pracovať na modeli zahŕňajúcom aj krivosť pohybujúcej sa krivky.

Prvým cieľom tejto práce bude odvodiť a riešiť *nestacionárnu rovnicu* funkcie úrovňovej množiny a *lineárnou interpoláciou* časov príchodu nulovej izočiary tejto funkcie získať stacionárne riešenie funkcie času prvého príchodu. V *nestacionárnej numerickej metóde* aplikujeme a porovnáme dve rôzne *diskretizácie krivosti*.

Následne budeme riešiť *stacionárnu rovnicu*, ktorú zlinearizujeme a získaný systém lineárnych rovníc budeme riešiť rôznymi *iteračnými metódami*. Porovnáme ich a spomenieme niektoré ich výhody, respektíve nevýhody pre riešenie rovnice s krivosťou.

Cenným prínosom tejto práce bude práve *porovnanie nestacionárnej a stacionárnej metódy* riešenia funkcie času prvého príchodu pre model uvažujúci krivosť. Oboma metódami vyriešime modelový príklad a odtestujeme ich presnosť a efektivitu pre rovnicu s členom krivosti.

Na záver získané poznatky aplikujeme pri riešení funkcie času prvého príchodu rozpínajúceho sa frontu lesného požiaru. Porovnáme tak na príklade z praxe vlastnosti stacionárnej a nestacionárnej metódy a taktiež poukážeme na vplyv krivosti na vlastnosti riešenie.

Kapitola 1

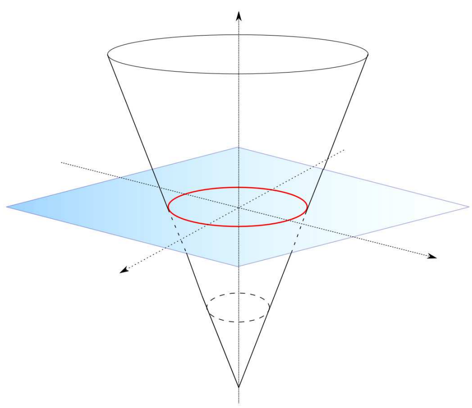
Metódy úrovňovej množiny

Metódy úrovňovej množiny (Level set methods), prvýkrát predstavené americkými matematikmi Stanleym Osherom a Jamesom Sethianom, sú výpočtové techniky slúžiace na opis šíriacich sa rozhraní funkcií. Vďaka svojim vlastnostiam pristupujú rovnako k n rozmerným funkciám, dokážu sa vysporiadať s topologickým zlučovaním či delením a efektívne počítajú šírenie frontov pohybujúcich sa pod vplyvom komplexných zákonov rýchlosti. Metódy úrovňovej množiny sú podrobne popísané v knihách [1] a [2].

Kľúčovou myšlienkou, ktorá podporila vznik metód úrovňovej množiny bol Hamilton - Jacobiho prístup k numerickým riešeniam časovo závislých nelineárnych rovníc pre implicitne zadané krivky a plochy. Viac o Hamilton - Jacobiho prístupe sa dá nájsť v knihe [1].

1.1 Vzdialenostná funkcia so znamienkom

V tejto kapitole objasníme pojem *dvojdimenzionálnej vzdialenostnej funkcie so znamienkom* (signed distance function) Φ.



Obr. 1.1: *Implicitná funkcia Φ rozdeľujúca oblasť na Ω^-, Ω^+ a hranicu $\partial\Omega$.*

V dvojdimenzionálnom prípade front, ktorého pohyb sledujeme, je krivka rozdeľujúca $\mathbb{R}^2$ na podoblasti s nenulovou plochou. Túto krivku reprezentujeme implicitne ako izočiaru nejakej funkcie. Budeme používať stručný názov "*implicitná funkcia*"pre funkcie, ktoré sú definované implicitne. Vo všeobecnosti platí, že izočiara n-rozmernej implicitnej funkcie definovanej na celom priestore $\mathbb{R}^n$ má dimenziu $n-1$. My sa zameriame na izočiary tvoriace uzavretú krivku. Tieto majú jednoznačne definovanú vnútornú a vonkajšiu oblasť.

Ako príklad si vezmime funkciu

$$\Phi = \sqrt{x^2 + y^2} - r, \tag{1.1}$$

kde r je zvolená konštanta.

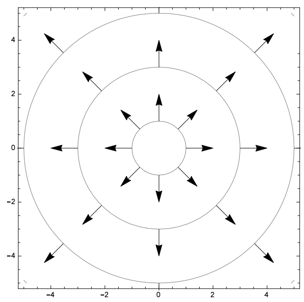
Nulová izočiara $\Phi(\mathbf{x}) = 0$ sa označuje $\partial\Omega$ a je to uzavretá kružnica definovaná ako $\partial\Omega = \{ \mathbf{x} \in \mathbb{R}^2, |\mathbf{x}| = r \}$, kde $|x|$ je Euklidovská norma. Vnútorná oblasť je otvorený kruh $\Omega^- = \{ \mathbf{x} \in \mathbb{R}^2, |\mathbf{x}| < r \}$ a vonkajšia oblasť je $\Omega^+ = \{ \mathbf{x} \in \mathbb{R}^2, |\mathbf{x}| > r \}$. Ďalej platí, že funkcia Φ nadobúda hodnotu 0 na $\partial\Omega$, záporné hodnoty na Ω^- a kladné hodnoty na Ω^+. Vzdialenostné funkcie so znamienkom sú podmnožinou implicitných funkcií spomínaných vyššie. Majme vzdialenostnú funkciu $d(x)$ definovanú ako

$$d(\mathbf{x}) = \min(|\mathbf{x} - \mathbf{x}_I|), \text{ pre všetky } \mathbf{x}_I \in \partial\Omega$$

a zvoľme si $\Phi(\mathbf{x}) = 0$ pre všetky $\mathbf{x} \in \partial\Omega$, $\Phi(\mathbf{x}) = -d(\mathbf{x})$ pre všetky $\mathbf{x} \in \Omega^-$ a podobne $\Phi(\mathbf{x}) = d(\mathbf{x})$ pre všetky $\mathbf{x} \in \Omega^+$. Teda vzdialenostná funkcia so znamienkom spĺňa všetky vlastnosti spomínaných implicitných funkcií. Funkcia Φ v (1.1) je tiež vzdialenostnou funkciou so znamienkom.

1.2 Formulácia nestacionárnej úlohy

Majme vzdialenostnú funkciu so znamienkom Φ. Počiatočnú pozíciu pozorovaného frontu definujme ako nulovú izočiaru tejto funkcie v čase $t = 0$. Ak front stotožníme s rovnakou, v tomto prípade nulovou izočiarou v ľubovoľnom čase t, odvodíme časovo závislý problém s definovanou počiatočnou podmienkou. Inak povedané, vývojom funkcie Φ sa bude meniť aktuálna nulová izočiara, čo efektívne simuluje propagáciu frontu (viď obrázok 1.3).



Obr. 1.2: *Gradient funkcie* (1.1).

Front v čase t je teda definovaný ako nulová izočiara funkcie Φ v čase t

$$\Phi(\boldsymbol{x}(t), t) = 0, \tag{1.2}$$

kde $\mathbf{x}(t)$ je bod ležiaci na fronte.

Reťazovým pravidlom získame

$$\Phi_t + \nabla\Phi(\boldsymbol{x}(t), t) \cdot \boldsymbol{x}'(t) = 0, \tag{1.3}$$

kde dolný index t značí parciálnu deriváciu podľa času.

Nech V je rýchlosť v smere vonkajšej normály, potom $V = \boldsymbol{x}'(t) \cdot \boldsymbol{n}$ pričom platí (viď obr. (1.2))

$$\boldsymbol{n} = \frac{\nabla\Phi}{|\nabla\Phi|}. \tag{1.4}$$

Takto získame rovnicu konvekcie (či advekcie) funkcie Φ s danou počiatočnou podmienkou $\Phi(\boldsymbol{x}, t = 0) = \Phi^0$

$$\Phi_t + V\,|\nabla\Phi| = 0, \tag{1.5}$$

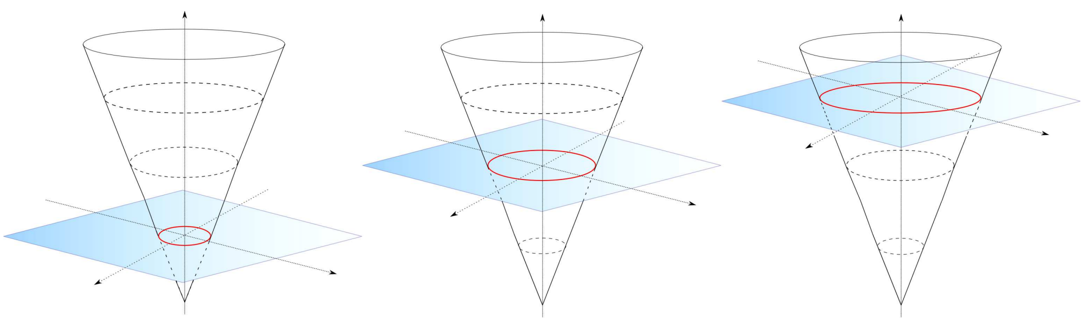
kde V v sebe môže zahŕňať rôzne faktory vplývajúce na zmenu rýchlosti. V našom prípade budeme brať do úvahy konštantnú, alebo po častiach konštantnú kladnú rýchlosť a (aby sa front vyvýjal v smere vonkajšej jednotkovej normály), ktorá modeluje napríklad "horlavosť" oblasti, po ktorej sa pohybuje sledovaný front a krivosť κ. Výsledná rýchlosť, s ktorou budeme počítať, je

$$V = a - b\kappa, \quad \text{pre} \ \ b > 0,\ a > 0.$$

V tomto prípade má naša rovnica na pravej strane difúzny člen [1] $b\kappa|\nabla\Phi|$, a teda jej finálna podoba je

$$\Phi_t + a|\nabla\Phi| = b\kappa|\nabla\Phi|. \tag{1.6}$$

Okrajové podmienky bližšie určíme pri formulácii numerickej metódy pre nestacionárnu rovnicu.



Obr. 1.3: *Vývoj nulovej izočiary vzdialenostnej funkcie so znamienkom v čase.*

1.3 Formulácia stacionárnej úlohy

Majme uzavretú krivku Γ pohybujúcu sa po rovine v smere svojej vonkajšej normály rýchlosťou V. Nech V je kladné, takže Γ sa bude "rozpínať" a do každého bodu roviny sa dostane najviac raz.

Tento pohyb budeme charakterizovať funkciou času prvého príchodu $T(x,y)$ v každom bode roviny x a y. Funkcia $T(x,y)$ je teda zobrazenie $T : \mathbb{R}^2 \to \mathbb{R}$ priraďujúce každému bodu roviny čas, v ktorom krivka "príde"do bodu (x,y).

Podľa knihy [2] odvodíme rovnicu vývoja funkcie $T(x,y)$ pomocou základného vzťahu $dráha = rýchlosť * čas$,

$$dx = V\,(dT) \tag{1.7}$$

a po úprave

$$1 = V\,\frac{dT}{dx}. \tag{1.8}$$

Vo viacdimenzionálnom prípade (my sa budeme venovať dvojdimenzinálnym príkladom) sa parciálna derivácia v (1.8) zmení na gradient funkcie $T(x,y)$.

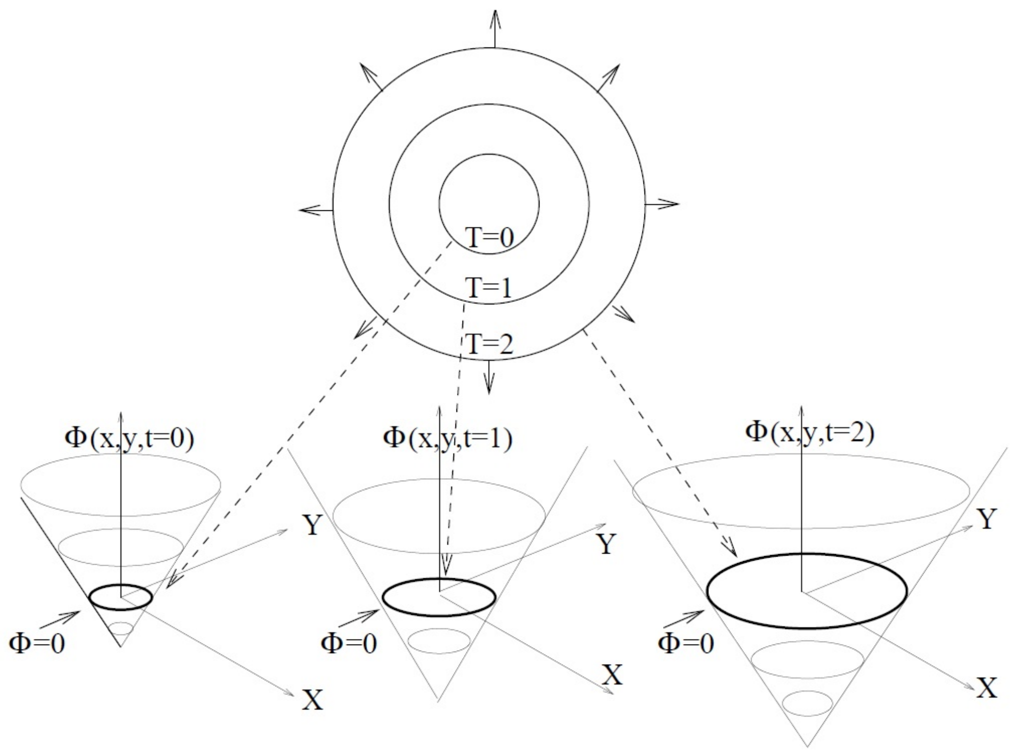
$$|\nabla T|\,V = 1, \quad T = 0 \text{ na } \Gamma \tag{1.9}$$

Aj v tomto prípade rýchlosť V môže nadobúdať rôzne podoby. Tak, ako aj vo formulácií problému s počiatočnou podmienkou, aj teraz budeme predpokladať, že platí $V = a - b\kappa$, s podmienkou $V > 0$.

Naša výsledná rovnica po dosadení za rýchlosť V teda bude

$$a|\nabla T| - b\kappa|\nabla T| = 1 \tag{1.10}$$

Okrajovým podmienkam sa budeme venovať pri formulácii numerickej metódy pre riešenie stacionárnej rovnice. Oba prístupy prinášajú so sebou isté výhody aj nevýhody a v niektorých prípadoch sa dajú navzájom tranformovať. Ich spoločnou charakeristikou je napríklad rovnaká formulácia vo viacdimenzionálnych prípadoch $\mathbb{R}^n$. Viac o podobnostiach a rozdieloch týchto dvoch stratégií sa dá nájsť v diele [2].



Obr. 1.4: *Graficky znázornená transformácia stacionárneho riešenia na nestacionárne riešenie z knihy [2].*

Kapitola 2

Numerické metódy úrovňovej množiny

V nasledujúcej kapitole sa budeme venovať diskretizácií a numerickým metódam riešenia metód úrovňovej množiny a funkcie času prvého príchodu. Nakoľko svet numerických metód je skutočne rozmanitý, priblížime a porovnáme viacero rôznych prístupov.

2.1 Numerické riešenie nestacionárnej rovnice

Prvým krokom numerického riešenia je správna diskretizácia rovnice (1.6). Na karteziánskej sieti by bolo problematické implementovať rovnicu (1.6) s rýchlosťou V definovanou len na samotnom fronte, preto budeme predpokladať, že rýchlosť V je definovaná na celej oblasti Ω (alebo len jednoducho na vopred určenej oblasti obsahujúcej sledovaný front).

Tento prístup však môže spôsobiť nepresnoti. Majme vektorové pole $\boldsymbol{V}$, ktoré je rovné nule takmer na celej oblasti Ω, okrem bodov pozície frontu v aktuálnom čase, kde $\boldsymbol{V} = [1, 0]$. Potom presným riešením by bol konštantný pohyb frontu vpravo rýchlosťou 1. Pri diskretizácií väčšina (alebo všetky) body ležia mimo krivky predstavujúcej front a bude im teda priradená rýchlosť 0 a front sa nebude takmer vôbec hýbať (alebo vôbec, ak ani jeden z bodov siete nebude ležať presne na fronte). Bližšie informácie o problémoch spôsobených diskretizáciou sa dajú nájsť v knihe [1].

Tomuto problému sa môžeme sčasti vyhnúť, ak budeme predpokladať, že rýchlosť je v blízkosti frontu spojitá.

Nech hodnota funkcie Φ v čase t^n je Φ^n. Meniť hodnoty funkcie Φ v čase znamená hľadať nové hodnoty Φ v bodoch siete po nejakom časovom prírastku Δt.

Najprv si uveďme časovú diskretizáciu. Nech má teda hodnota funkcie Φ v novom čase t^{n+1} hodnotu $\Phi(t^{n+1}) \approx \Phi^{n+1}$, kde $t^{n+1} = t^n + \Delta t$.

Pre časovú diskretizáciu rovnice (1.5) použijeme doprednú konečnú diferenciu prvého rádu presnosti.

$$\frac{\Phi^{n+1} - \Phi^n}{\Delta t} + V^n |\nabla \Phi^n| = 0, \tag{2.1}$$

kde V^n predstavuje rýchlosť v bodoch siete v čase t^n a $|\nabla \Phi^n|$ je norma gradientu funkcie Φ v čase t^n.

Osamostatnením člena Φ^{n+1} na ľavej strane získame rovnicu

$$\Phi^{n+1} = \Phi^n - \Delta t \; V^n |\nabla \Phi^n|. \tag{2.2}$$

Rovnica (2.2) predstavuje explicitnú schému, ktorou vypočítame hodnoty funkcie Φ v bodoch siete a v každom čase t^n. Za V^n dosadíme (1.2). Získame tvar

$$\Phi^{n+1} = \Phi^n - a \, \Delta t \, |\nabla \Phi^n| + b \, \Delta t \, \kappa^n \, |\nabla \Phi^n|. \tag{2.3}$$

Pre zabezpečenie stability numerickej metódy časový prírastok Δt zhora ohraničíme vhodnou podmienkou pre $b \neq 0$

$$\Delta t \left(\frac{4b}{h^2} \right) \leq 1, \tag{2.4}$$

kde h predstavuje priestorový krok diskretizácie.

V príklade, kde $b = 0$ je vhodnejšou podmienkou

$$\Delta t \leq \frac{h}{2a}. \tag{2.5}$$

Vo všeobecnosti sa jedná o menej "prísnu"podmienku, ktorá je dostačujúca pre úlohy neuvažujúce člen s krivosťou. V nasledujúcej časti odvodíme priestorovú diskretizáciu.

2.1.1 Rouy-Tourinovej schéma

V nasledujúcej časti uvedieme zhrnutie Rouy-Tourinovej schémy priestorovej diskretizácie z práce [1] pre člen v rovnici (2.3)

$$a \, \Delta t \, |\nabla \Phi^n|. \tag{2.6}$$

Nech h je priestorový krok na osi x a y. Označme $\Phi^n(x_i, y_i) \approx \Phi^n_{ij}$, kde $x_{i+1} = x_i + ih$ a analogicky $y_{j+1} = y_j + jh$, pre i a j idúce od 0 po N.

Potom $|\nabla \Phi^n|$ vo výraze (2.6) bude rovný

$$\sqrt{(\partial_x \Phi^n_{ij})^2 + (\partial_y \Phi^n_{ij})^2}, \tag{2.7}$$

kde výrazy $\partial_x \Phi^n_{ij}$ a $\partial_y \Phi^n_{ij}$ predstavujú aproximácie parciálnych derivácií podľa x a podľa y.

Na ich aproximáciu sa používajú konečné diferencie prvého rádu presnosti podľa nasledujúcej schémy:

$$\partial_x \Phi_{ij}^n = \begin{cases} \frac{\Phi_{ij}^n - \Phi_{i-1j}^n}{h} & pre \quad \Phi_{i-1j}^n = \min\left\{\Phi_{i-1j}^n, \Phi_{ij}^n, \Phi_{i+1j}^n\right\} \\ \frac{\Phi_{i+1j}^n - \Phi_{ij}^n}{h} & pre \quad \Phi_{i+1j}^n = \min\left\{\Phi_{i-1j}^n, \Phi_{ij}^n, \Phi_{i+1j}^n\right\} \\ 0 & pre \quad \Phi_{ij}^n = \min\left\{\Phi_{i-1j}^n, \Phi_{ij}^n, \Phi_{i+1j}^n\right\} \end{cases} \tag{2.8}$$

Na hranici siete, t.j. pre $\partial_x \Phi_{0j}^n, \partial_x \Phi_{Nj}^n$ a $\partial_y \Phi_{i0}^n, \partial_y \Phi_{iN}^n$, zo schémy vypadne dopredná či spätná diferencia podľa toho, na ktorej hranici sa nachádzame. Analogicky sa postupuje pri aproximácií parciálnej derivácie podľa y.

2.1.2 Diskretizácia krivosti κ

Posledným členom rovnice (2.3), ktorý sme zatiaľ nezdiskretizovali je

$$b\ \Delta t\ \kappa^n\ |\nabla \Phi^n|. \tag{2.9}$$

Krivosť frontu je definovaná ako divergencia jednotkovej vonkajšej normály $\boldsymbol{N}$ [1], teda

$$\kappa = \nabla \cdot \boldsymbol{N}. \tag{2.10}$$

Keďže gradient $\nabla \Phi$ je kolmý na front funkcie Φ a pri vhodnej voľbe rovnaký smer ako vonkajšia jednotková normála Φ (viď obr. (1.2)), môžeme ju vyjadriť ako podiel gradientu funkcie Φ a jeho normy.

$$\boldsymbol{N} = \frac{\nabla \Phi}{|\nabla \Phi|} \tag{2.11}$$

Dosadením (2.11) do (2.10) získame

$$\kappa = \nabla \cdot \left(\frac{\nabla \Phi}{|\nabla \Phi|}\right) \tag{2.12}$$

Podrobnejšie [1]

$$\kappa = \frac{\Phi_x^2 \Phi_{yy} - 2\Phi_x \Phi_y \Phi_{xy} + \Phi_y^2 \Phi_{xx}}{|\nabla \Phi|^3}, \tag{2.13}$$

kde dolné indexy značia parciálne derivácie. Teraz máme na výber dva postupy diskretizácie. Oba neskôr porovnáme na konkrétnom príklade. V oboch budeme krivosť na hranici pre jednoduchosť zanedbávať. Na hranici siete tak bude platiť

$$\Phi^{n+1} = \Phi^n - a\Delta t |\nabla \Phi^n|$$

1. **Diskretizácia krivosti pomocou konečných diferencií.**

   Prvým postupom je diskretizácia parciálnych derivácií vo výraze (2.13) centrálnymi konečnými diferenciami druhého rádu presnosti. Použijeme konečné diferencie z knihy [4].

$$
\begin{aligned}
&\Phi_x(x_i, y_j) \approx \frac{\Phi^n_{i+1j} - \Phi^n_{i-1j}}{2h} \quad \Phi_{xx}(x_i, y_j) \approx \frac{\Phi^n_{i+1j} - \Phi^n_{ij} + \Phi^n_{i-1j}}{h^2} \\
&\Phi_{xy}(x_i, y_j) \approx \frac{\Phi^n_{i+1j+1} - \Phi^n_{i+1j-1} - \Phi^n_{i-1j+1} + \Phi^n_{i-1j-1}}{4h^2}
\end{aligned}
\tag{2.14}
$$

   Aproximácie parciálnych derivácií podľa premennej y vyzerajú analogicky.

   Dosadíme konečné diferencie (1) do (2.13). Gradient funkcie Φ budeme taktiež diskretizovať s použitím konečných diferencií Φ_x a Φ_y z (1). Tento postup budeme opakovať v každom čase n a pre každý bod siete x_i a y_j, okrem hraničných bodov, kde krivosť zanedbáme. Vyhneme sa tak problémom s centrálnymi diferenciami na hranici siete, kde susedné hodnoty nie sú známe.

2. **Diskretizácia krivosti metódou konečných objemov**

   Alternatívnym postupom pri diskretizácií krivosti je použitie metódy konečných objemov vo výraze (2.12). Viac o tejto metóde sa dá nájsť v skriptách [3]. V tomto prípade diskretizácia krivosti bude mať tvar

$$
\kappa \approx \frac{\left(\frac{\partial_x \Phi}{|\nabla \Phi|}\right)^n_{i+1/2\ j} - \left(\frac{\partial_x \Phi}{|\nabla \Phi|}\right)^n_{i-1/2\ j}}{h} + \frac{\left(\frac{\partial_y \Phi}{|\nabla \Phi|}\right)^n_{i\ j+1/2} - \left(\frac{\partial_y \Phi}{|\nabla \Phi|}\right)^n_{i\ j-1/2}}{h}, \tag{2.15}
$$

   kde člen $\left(\frac{\partial_x \Phi}{|\nabla \Phi|}\right)^n_{i+1/2\ j}$ má tvar

$$
\left(\frac{\partial_x \Phi}{|\nabla \Phi|}\right)^n_{i+1/2\ j} = \frac{\left(\Phi^n_{i+1j} - \Phi^n_{ij}\right)}{\sqrt{\left(\Phi^n_{i+1j} - \Phi^n_{ij}\right)^2 + \frac{\left(\Phi^n_{ij+1} - \Phi^n_{ij-1} + \Phi^n_{i+1j+1} - \Phi^n_{i+1j-1}\right)^2}{16}}} \tag{2.16}
$$

   Analogicky získame tvar ostatných členov vo výraze (2.15).

   Ďalej postupujeme rovnako ako pri prvej metóde. Aproximáciou pomocou konečných objemov (2.15) budeme počítať krivosť v každom čase n a vo všetkých bodoch siete x_i a y_j, okrem hraničných bodov, kde krivosť zanedbáme.

2.1.3 Výpočet času prvého príchodu

Vyššie opísaným nestacionárnym numerickým riešením metódy úrovňovej množiny získame aproximačné hodnoty implicitnej funkcie Φ v každom čase t^n. Naším cieľom je využiť tieto hodnoty a získať funkciu $T(x, y)$, ktorá bude vracať čas prvého príchodu frontu do bodov siete x_i a y_j.

Nezabudnime, že náš front predstavuje nulová izočiara funkcie Φ v každom čase t. To znamená, že front v časovom intervale (t^n, t^{n+1}) prejde práve tými bodmi x_i a y_j, v ktorých funkcia Φ zmenila znamienko. Teda platí

$$\Phi_{ij}^n \geq 0 \text{ a zároveň } \Phi_{ij}^{n+1} \leq 0. \tag{2.17}$$

Ak by v niektorej z hodnôt platila rovnosť, znamenalo by to, že nulová izočiara funkcie Φ sa nachádza v uzlových bodoch (x^i, y^j) a čas prvého príchodu je t^n, prípade t^{n+1}.

Na výpočet konkrétneho času príchodu frontu do bodu (x^i, y^j) využijeme jednoduchú lineárnu interpoláciu v tvare

$$0 = \Phi_{ij}^n + \eta(\Phi_{ij}^{n+1} - \Phi_{ij}^n). \tag{2.18}$$

Potom je jasné, že v bodoch, pre ktoré platí (2.17), je čas prvého príchodu frontu rovný

$$t^n + \frac{\Phi_{ij}^n}{\Phi_{ij}^n - \Phi_{ij}^{n+1}} \Delta t. \tag{2.19}$$

2.2 Numerické riešenie stacionárnej rovnice

Stacionárne numerické riešenie pozostáva z dvoch častí: linearizácia a iteračná metóda. Linearizáciou získame systém lineárnych rovníc, ktorý riešime iteračne pomocou nejakej iteračnej metódy. Neskôr si ukážeme a porovnáme viacero iteračných metód. Zatiaľ si len povedzme, že linearizácia je v každom prípade rovnaká.

Použime rovnakú priestorovú diskretizáciu ako pri nestacionárnom numerickom riešení, teda $x_{i+1} = x_i + ih$ a $y_{j+1} = y_j + jh$ a nech $T_{ij} = T(x_i, y_j)$.

Rovnica (1.10) má na ľavej strane dva členy. Najskôr odvoďme linearizáciu rovnice pre $b = 0$, teda pre tvar $a|\nabla T| = 1$. Toto zjednodušenie nám pomôže lepšie pochopiť rozdiel v linearizácií rovnice s prvými deriváciami a rovnice, ktorá už obsahuje aj člen s krivosťou κ.

2.2.1 Linearizácia

Rovnicu $a|\nabla T| = 1$ linearizujme do tvaru

$$a\,\frac{\nabla T^k}{|\nabla T^k|}\cdot\nabla T^{k+1} = 1, \tag{2.20}$$

kde index $k = 0, 1, 2, \ldots$ predstavuje k-tu linearizáciu, pričom počiatočný odhad T^0 si musíme zvoliť. Poznamenajme, že získať vhodný počiatočný odhad je vo všeobecnosti netriviálne. Platí $a > 0$, a tak ním môžeme rovnicu predeliť. Po aplikácií skalárneho súčinu získame tvar

$$\frac{\partial_x T^k\ \partial_x T^{k+1}}{|\nabla T^k|} + \frac{\partial_y T^k\ \partial_y T^{k+1}}{|\nabla T^k|} = \frac{1}{a}, \tag{2.21}$$

kde znakom parciálnej derivácie označujeme numerickú aproximáciu parciálnej derivácie podľa x a podľa y. Označme pre body siete

$$U_{ij} := \frac{\partial_x T^k_{ij}}{|\nabla T^k_{ij}|},\quad V_{ij} := \frac{\partial_y T^k_{ij}}{|\nabla T^k_{ij}|}\ . \tag{2.22}$$

Na aproximáciu parciálnych derivácií použijeme Rouy-Tourinovej schému (2.8). Získame tvar

$$U_{ij}\ \partial_x T^{k+1}_{ij} + V_{ij}\ \partial_y T^{k+1}_{ij} = \frac{1}{a}$$

Uvažujme dva rôzne scenáre pre hodnoty U_{ij} a V_{ij}: ak je ich hodnota kladná, deriváciu, pri ktorej stoja, budeme aproximovať konečnou spätnou diferenciou a v opačnom prípade konečnou doprednou diferenciou. Táto schéma sa nazýva Upwind schéma [4]. Definujme

$$[U_{ij}]^+ = \begin{cases} U_{ij} & pre \quad U_{ij} \geq 0 \\ 0 & pre \quad U_{ij} \leq 0 \end{cases} \qquad [U_{ij}]^- = \begin{cases} U_{ij} & pre \quad U_{ij} \leq 0 \\ 0 & pre \quad U_{ij} \geq 0 \end{cases}$$

Analogicky budú vyzerať hodnoty $[V_{ij}]^+$ a $[V_{ij}]^-$.

Po dosadení Upwind schémy získavame tvar linearizácie rovnice

$$[U_{ij}]^+\,\frac{T^{k+1}_{ij} - T^{k+1}_{i-1j}}{h} + [U_{ij}]^-\,\frac{T^{k+1}_{i+1j} - T^{k+1}_{ij}}{h} + [V_{ij}]^+\,\frac{T^{k+1}_{ij} - T^{k+1}_{ij-1}}{h} + [V_{ij}]^-\,\frac{T^{k+1}_{ij+1} - T^{k+1}_{ij}}{h} = \frac{1}{a}.$$

Toto bude zároveň tvar výslednej linearizácie na hranici siete, kde krivosť pre jednoduchosť zanedbáme. Ukážme si teraz, ako vyzerá linearizácia rovnice $-b\kappa|\nabla T| = 1$. Za κ dosaďme tvar z definície (2.12). Získame linearizovanú rovnicu

$$-\,b\ \nabla\cdot\left(\frac{\nabla T^{k+1}}{|\nabla T^k|}\right)|\nabla T^k| = 1 \tag{2.23}$$

Krivosť teraz môžeme aproximovať metódou konečných objemov (2.15) a normu gradientu pomocou centrálnej konečnej diferencie [4]. Po dosadení jednotlivých diskretizácií a po úprave získame tvar

$$-\frac{b}{2h^2}\sqrt{\left(T^k_{i+1j}-T^k_{i-1j}\right)^2+\left(T^k_{ij+1}-T^k_{ij-1}\right)^2}$$

$$\cdot\left(\left(\frac{\partial_x T^{k+1}}{|\nabla T^k|}\right)_{i+1/2\ j}-\left(\frac{\partial_x T^{k+1}}{|\nabla T^k|}\right)_{i-1/2\ j}+\left(\frac{\partial_y T^{k+1}}{|\nabla T^k|}\right)_{i\ j+1/2}-\left(\frac{\partial_y T^{k+1}}{|\nabla T^k|}\right)_{i\ j-1/2}\right)=1$$

Platí analogický tvar definície (2.16). Pre zjednodušenie zápisu označme [1]

$$\alpha_{ij}=\sqrt{\left(T^k_{i+1j}-T^k_{ij}\right)^2+\frac{\left(T^k_{ij+1}-T^k_{ij-1}+T^k_{i+1j+1}-T^k_{i+1j-1}\right)^2}{16}}$$

$$\beta_{ij}=\sqrt{\left(T^k_{ij}-T^k_{i-1j}\right)^2+\frac{\left(T^k_{ij+1}-T^k_{ij-1}+T^k_{i-1j+1}-T^k_{i-1j-1}\right)^2}{16}}$$

$$\gamma_{ij}=\sqrt{\frac{\left(T^k_{i+1j}-T^k_{i-1j}+T^k_{i+1j+1}-T^k_{i-1j+1}\right)^2}{16}+\left(T^k_{ij+1}-T^k_{ij}\right)^2}$$

$$\delta_{ij}=\sqrt{\frac{\left(T^k_{i+1j}-T^k_{i-1j}+T^k_{i+1j-1}-T^k_{i-1j-1}\right)^2}{16}+\left(T^k_{ij}-T^k_{ij-1}\right)^2}$$

$$\lambda_{ij}=\frac{1}{2h}\sqrt{\left(T^k_{i+1j}-T^k_{i-1j}\right)^2+\left(T^k_{ij+1}-T^k_{ij-1}\right)^2}$$

Výsledný tvar linearizácie po diskretizácuií teda bude

$$-\frac{b}{h}\lambda_{ij}\left(\left(\frac{T^{k+1}_{i+1j}-T^{k+1}_{ij}}{\alpha_{ij}}\right)-\left(\frac{T^{k+1}_{ij}-T^{k+1}_{i-1j}}{\beta_{ij}}\right)+\left(\frac{T^{k+1}_{ij+1}-T^{k+1}_{ij}}{\gamma_{ij}}\right)-\left(\frac{T^{k+1}_{ij}-T^{k+1}_{ij-1}}{\delta_{ij}}\right)\right)=1$$

Teraz je len potrebné správne dosadiť obe opísané linearizácie do rovnice (1.10).

2.2.2 Riešenie systému iteračnou metódou

Linearizáciou sme získali systém lineárnych rovníc. Existuje viacero spôsobov, ako takýto systém riešiť. Mnohé metódy sú opísané a rozobraté v knihe [5]. My sme sa roz-

[1] Zamlčíme horný index k kvôli prehľadnosti v iteračných metódach

hodli použiť a následne aj porovnať zopár známych iteračných metód.

Približné riešenie systému budeme hľadať v jednotlivých krokoch, takzvaných iteráciach, kde sa každé nové riešenie bude počítať ako modifkácia predchádzajúceho riešenia násobením iteračnou maticou.

Majme teda iteračný index $e = 0, 1, 2, \ldots$ a označme $C^0_{ij} = T^k_{ij}$ nulté riešenie systému, t.j. pre $e = 0$. Riešenie $\mathbf{C}^{e+1}$ budeme hľadať v tvare $\mathbf{C}^{e+1} = S^e\mathbf{C}^e + V^e\mathbf{p}$, kde $\mathbf{p}$ je pravá strana systému a S^e, V^e sú iteračné matice závisiace od iteračnej metódy, prípadne od iteračného kroku e.

Pozrieme sa na to, ako funguje Jacobiho metóda, Gauss-Seidelova metóda a metóda Fast sweeping. Porovnáme ich efektivitu, presnosť a rôzne výhody, či nevýhody, ktoré so sebou prinášajú.

1. **Jacobiho metóda**

Pri Jacobiho metóde počítame približné riešenie systému $\mathbf{C}^{e+1}$ v tvare

$$C^{e+1}_{ij} = \frac{h + a\left([U_{ij}]^+C^e_{i-1j} - [U_{ij}]^-C^e_{i+1j} + [V_{ij}]^+C^e_{ij-1} - [V_{ij}]^-C^e_{ij+1}\right)}{a\left([U_{ij}]^+ - [U_{ij}]^- + [V_{ij}]^+ - [V_{ij}]^-\right) + b\lambda_{ij}\left(\alpha^{-1}_{ij} + \beta^{-1}_{ij} + \gamma^{-1}_{ij} + \delta^{-1}_{ij}\right)}$$

$$+ \frac{b\lambda_{ij}\left(\frac{C^e_{i+1j}}{\alpha_{ij}} + \frac{C^e_{i-1j}}{\beta_{ij}} + \frac{C^i_{ij+1}}{\gamma_{ij}} + \frac{C^e_{ij-1}}{\delta_{ij}}\right)}{a\left([U_{ij}]^+ - [U_{ij}]^- + [V_{ij}]^+ - [V_{ij}]^-\right) + b\lambda_{ij}\left(\alpha^{-1}_{ij} + \beta^{-1}_{ij} + \gamma^{-1}_{ij} + \delta^{-1}_{ij}\right)} \tag{2.24}$$

Všetky hodnoty na pravej strane rovnice sú známe z linearizácie alebo z predchádzajúcej iterácie. Výhodou tejto metódy je preto jednoduchá paralelizácia.

Iteračné matice pozostávajú z hodnôt vypočítaných pri linearizácii a nezávisia od iteračného kroku e.

Iteračný cyklus končí, keď sa naplní takzvané kritérium konvergencie, t.j dosiahne sa požadovaná blízkosť približného riešenia C^k k presnému riešeniu. Bežným kritériom konvergencie je napríklad norma rezídua r^e definovaná ako

$$r^e = ||\mathbf{p} - A\mathbf{C}^e|| \leq tol, \quad 0 < tol \ll 1, \tag{2.25}$$

kde tol je zvolená malá kladná konštanta.

Iným kritériom konvergencie je napríklad norma korekcie

$$norm^e = ||\mathbf{C}^{e+1} - \mathbf{C}^e|| \leq tol, \quad 0 < tol \ll 1. \tag{2.26}$$

2. **Gauss-Seidelova metóda**

Princíp Gauss-Seidelovej metódy je rovnaký ako pri Jacobiho metóde. Aj v tomto prípade budeme počítať približné riešenie systému $\mathbf{C}^{e+1}$. Kľúčovou zmenou bude využitie zložiek novej iterácie ihneď potom, ako budú vypočítané.

To znamená, že pre $i = 0, 1, 2, \ldots, n$ a $j = 0, 1, 2, \ldots, n$ bude mať výpočet nového približného riešenia tvar

$$C_{ij}^{e+1} = \frac{h + a\left([U_{ij}]^+ C_{i-1j}^{e+1} - [U_{ij}]^- C_{i+1j}^{e} + [V_{ij}]^+ C_{ij-1}^{e+1} - [V_{ij}]^- C_{ij+1}^{e}\right)}{a\left([U_{ij}]^+ - [U_{ij}]^- + [V_{ij}]^+ - [V_{ij}]^-\right) + b\lambda_{ij}\left(\alpha_{ij}^{-1} + \beta_{ij}^{-1} + \gamma_{ij}^{-1} + \delta_{ij}^{-1}\right)}$$

$$+ \frac{b\lambda_{ij}\left(\frac{C_{i+1j}^{e}}{\alpha_{ij}} + \frac{C_{i-1j}^{e+1}}{\beta_{ij}} + \frac{C_{ij+1}^{i}}{\gamma_{ij}} + \frac{C_{ij-1}^{e+1}}{\delta_{ij}}\right)}{a\left([U_{ij}]^+ - [U_{ij}]^- + [V_{ij}]^+ - [V_{ij}]^-\right) + b\lambda_{ij}\left(\alpha_{ij}^{-1} + \beta_{ij}^{-1} + \gamma_{ij}^{-1} + \delta_{ij}^{-1}\right)} \quad (2.27)$$

Vidíme použitie hodnôt nového približného riešenia $\mathbf{C}^{e+1}$ v bodoch $i-1$ a $j-1$, ktoré sú v tomto momente už známe. Táto nenápadná zmena algoritmu vo všeobecnosti môže urýchliť konvergenciu metódy. Nevýhodou je strata paralelizačného potenciálu, ktorý má Jacobiho metóda.

Aj v tomto prípade sa pri ukončení iteračného cyklu budeme riadiť normou rezídua (2.25) alebo normou korekcie (2.26).

3. **Metóda Fast Sweeping**

Algoritmus metódy Fast Sweeping značne vystihuje jej samotný názov - "rýchle zametanie". Svojím charakterom sa podobá na Gauss-Seidelovu metódu.

Pri jednotlivých iteráciach e prechádzame bodmi siete i a j v rôznych "smeroch". To znamená, že budeme rozlišovať štyri druhy cyklov, kde i a j budú nadobúdať hodnoty $i = 0, 1, 2, \ldots, n$ a $j = 0, 1, 2, \ldots, n$, alebo naopak $i = n, n-1, n-2, \ldots, 0$ a $j = n, n-1, n-2, \ldots, 0$, či ďalšie kombinácie.

Vždy budeme pri výpočte nového približného riešenia systému využívať už známe hodnoty $\mathbf{C}^{e+1}$. Výpočet C_{ij}^{e+1} bude mať teda rovnaký tvar ako (2.27) pre hodnoty $i = 0, 1, 2, \ldots, n$ a $j = 0, 1, 2, \ldots, n$ a jeho podoba pre ďalšie iterácie bude mať analogický tvar.

Iteračné kritérium bude opäť (2.25) alebo (2.26).

Poznamenajme, že metóda Fast sweeping pre prípad $b = 0$ konverguje v konečnom počte iterácii [6].

Na záver je dôležité poznamenať, že iteračný proces pre ktorúkoľvek z uvedených metód nemusí vo všeobecnosti konvergovať. Konvergencia iteračných metód zavisí od iteračnej matice. Platí, že ak je matica sústavy diagonálne dominantná, spomenuté metódy vždy konvergujú [5].

Naša matica systému, ktorú sme získali linearizáciou, je diagonálne dominantná. Môžeme to vidieť napríklad na rovnici (2.24). Koeficient pri C_{ij}^k je kladný a pozostáva zo súčtu koeficientov susedných členoch. Môžeme teda predpokladať, že naša metóda s použitím ľubovoľnej zo spomenutých iteračných metód bude konvergovať.

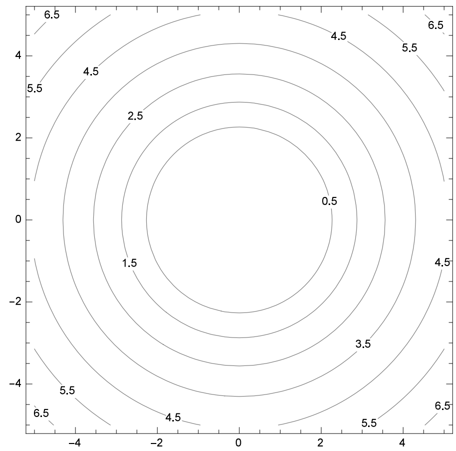
Vyriešením systému lineárnych rovníc získavame priamo stacionárne riešenie, teda čas prvého príchodu pre body siete.

Kapitola 3

Porovnanie numerických metód

Rozobrali sme viacero numerických metód výpočtu času prvého príchodu a ako to už raz býva, každá numerická metóda so sebou prináša rôzne výhody a nevýhody. Je teda na mieste porovnať ich na jednoduchom príklade a demonštrovať tak niektoré ich vlastnosti.

Upozorňujeme však, že akékoľvek správanie sledované na jednom konkrétnom príklade sa nedá zovšeobecniť. Vhodnosť numerických metód sa môže meniť v závislosti od konkrétnej riešenej úlohy. Napriek tomu považujeme prinajmenšom za zaujímavé ich porovnať.



Obr. 3.1: *Graf presného riešenia funkcie času prvého príchodu* (3.1)

Dôležitou časťou tejto kapitoly bude porovnanie nestacionárnej a stacionárnej metódy výpočtu funkcie času prvého príchodu. Tieto dva prístupy sa v mnohom líšia.

Majme implicitnú vzdialenostnú funkciu so znamienkom $\Phi = \sqrt{x^2+y^2} - r$ a nech rýchlosť je $V = 1 - \kappa$, t.j. $a = 1$ a $b = 1$. Našou úlohou je vypočítať čas prvého príchodu frontu pre body siete. Aby sme vedeli jednotlivé metódy efektívne porovnať, potrebujeme poznať presné riešenie tohto problému, ktoré sa odvádza podobne ako v knihe [7]. Za jeho výsledný tvar vďačíme Mgr. Martinovi Balažovjechovi, PhD, ktorý nám poskytol výsledok svojej práce a umožnil nám tak overiť a porovnať presnosť numerických metód.

$$T^p(r) = -r^0 + \ln\left(\frac{e^r\,(-1+r)}{-1+r^0}\right), \tag{3.1}$$

kde r^0 je počiatočný polomer rozpínajúcej sa kružnice.

Nakoľko okrajové podmienky nášho modelu sa líšia od tohto presného riešenia, predpokladáme, že numerické riešenie sa na hranici bude javiť nepresné.

3.1 Riešenie nestacionárnej rovnice

V predchádzajúcej kapitole sme odvodili tvar numerického riešenia nestacionárnej rovnice. Už vieme, že sa jedná o explicitnú metódu, a teda hodnoty funkcie Φ v čase t^{n+1} počítame z jej hodnôt v čase t^n. Pre úplnosť zopakujme, že stacionárne riešenie, takzvanú funkciu času prvého príchodu v bodoch siete, získame lineárnou interpoláciou hodnôt Φ^n a Φ^{n+1} vždy, keď $\Phi^n \geq 0$ a $\Phi^{n+1} \leq 0$.

Počítali sme príklad s danou počiatočnou podmienkou

$$\Phi^0_{ij} = \sqrt{x_i^2 + y_j^2} - 2 \tag{3.2}$$

a s danými parametrami $a = 1, b = 1$ na sieti $x_0 = y_0 = -5$ a $x_N = y_N = 5$.

Zvoľme časový krok $\Delta t = \dfrac{h^2}{4b}$ podľa podmienky (2.4) a priestorový krok nech sa počíta vzťahom $h = \frac{x_N - x_0}{N}$, kde N je počet dielikov na sieti.

Príklad sme riešili s použitím oboch spomenutých diskretizácií krivosti, to znamená pomocou konečných diferencií a pomocou konečných objemov [4]. Zisťovali sme normu chyby riešenia

$$||T^p - T^a||_N = h^2 \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} |T^p_{ij} - T^a_{ij}|, \tag{3.3}$$

kde T^p je presné riešenie (3.1) pre $r^0 = 2$ a T^a je naše numerické riešenie. Normu zámerne nepočítame pre body na hranici siete, kde zanedbávame krivosť κ. Sledujeme taktiež EOC riešenia (experimental order of convengence).

$$EOC = \log_2 \left(\frac{||T^p - T^a||_{\frac{N}{2}}}{||T^p - T^a||_N} \right) \tag{3.4}$$

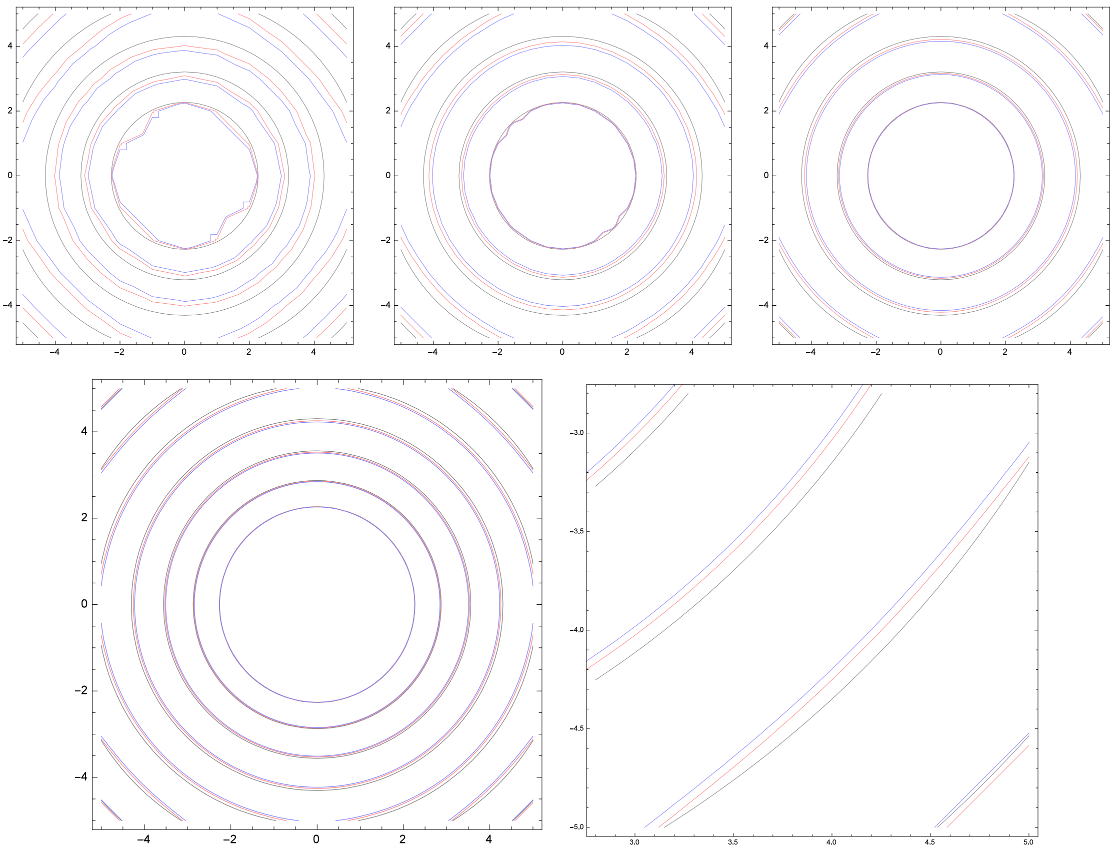
Na základe výsledkov získaných riešením úlohy oboma spomenutými spôsobmi diskretizácie nami vytvoreným programom sme zostavili nasledujúcu tabuľku. Vďaka nej vieme jednoducho porovnať presnosť oboch diskretizačných metód.

Vidíme, že pri riešení príkladu (1.1) s $r^0 = 2$ je metóda konečných diferencií o niečo presnejšia. EOC oboch metód však konverguje k hodnote 1 a dovolíme si tvrdiť, že pre obe diskretizácie krivosti je výsledná metóda prvého rádu presnosti.

Pozorujeme tiež, že chyba narastá v závislosti od času riešenia (viď obr. (3.2)). Jedným z možných vysvetlení tohoto javu je kumulácia chyby v časových krokoch riešenia explicitnej schémy. Každé riešenie v čase $n + 1$ sa počíta z riešenia v čase n. Pri jeho výpočte sa dopúšťame numerickej chyby a tá sa automaticky prenáša do ďaľšieho kroku algoritmu.

Presnosť riešenia						
Parametre riešenia			Konečné diferencie		Konečné objemy	
N	h	Δt	norm	EOC	norm	EOC
10	0.25	1	23.3974	-	39.1057	-
20	0.0625	0.5	15.7495	0.571	26.2496	0.575
40	0.015625	0.25	9.1736	0.780	14.7628	0.830
80	0.00390625	0.125	4.6873	0.969	7.5280	0.972

Tabuľka 3.1: *Tabuľka presnosti riešenia dosiahnutej použitím dvoch rôznych druhov diskretizácie krivosti.*



Obr. 3.2: *Grafy numerických riešenení nestacionárnej (červená) a stacionárnej rovnice (modrá), a presného riešenia (šedá). Prvé tri grafy predstavujú riešenia pre $N = 10, N = 20$ a $N = 40$. Vľavo dole je graf riešení pre $N = 80$ a vpravo dole je detail jeho pravého dolného rohu.*

3.2 Riešenie stacionárnej rovnice

Už sme spomenuli, že numerické riešenie stacionárnej rovnice pozostáva z linearizácie a riešenia systému lineárnych rovníc napríklad iteračnou metódou.

Vhodnú linearizáciu sme si odvodili v sekcii 2.2. Získali sme systém lineárnych rovníc, ktorý sme riešili tromi rôznymi iteračnými metódami. Porovnajme rýchlosť konvergencie a presnosť týchto metód na konkrétnom príklade.

Opäť podotýkame, že výsledky porovnania nie je možné zovšeobecniť a sú spojené s vlastnosťami konkrétneho príkladu.

Zvoľme časový krok $\Delta t = \dfrac{h^2}{4b}$ podľa podmienky (2.4) a priestorový krok nech sa počíta vzťahom $h = \frac{x_N - x_0}{N}$, kde N je počet dielikov na sieti.

Ďalej zvoľme za počiatočný odhad T^0 numerické riešenie T^a, ktoré sme získali v sekcii 3.1 riešením nestacionárnej rovnice s použitím metódy konečných objemov.

Zvoľme maximálny počet linearizácii $kmax = 100$ a maximálny počet iterácii v iteračnej metóde $emax = 25$. V oboch prípadoch zvoľme za iteračné kritérium normu korekcie (2.26).

Sledujeme normu chyby riešenia (3.3) pre izočiary nachádzajúce sa vo vnútri siete, a index k, pri ktorom dokonvergovalo stacionárne riešenie numerickej metódy pri zvolenej tolerancii $tol = 10^{-4}$.

Presnosť riešenia						
Počet dielikov	Jacobiho metóda		Gauss-Seidelova metóda		Metóda Fast Sweeping	
N	k	norm	k	norm	k	norm
10	13	13.7029	13	13.7031	13	13.7031
20	24	5.1034	20	5.1046	20	5.1045
40	50	1.4082	41	1.4101	41	1.4104
80	98	0.3612	79	0.3617	71	0.3606

Tabuľka 3.2: *Tabuľka presnosti a rýchlosti konvergencie stacionárneho riešenia pre rôzne iteračné metódy s počiatočným odhadom T^a zo sekcie 3.1.*

V tabuľke (3.2) môžeme vidieť, že Gauss-Seidelova metóda konverguje rýchlejšie ako Jacobiho metóda. Naopak metóda Fast Sweeping sa v tomto prípade zdá byť podobne efektívna ako Gauss-Seidelova metóda, čo by mohlo znamenať, že pri rovniciach s difúznym členom a pri nami použitej linearizácií Fast Sweeping stráca vlastnosť konvergencie v konečnom počte krokov, ktorú má napríklad pri rovniciach bez difúzneho člena.

Veľký vplyv na rýchlosť konvergencie linearizácie má vhodná voľba *tol* pre iteračnú metódu a hodnota $emax$. Pri malých hodnotách $emax$ sa rozdiel medzi rýchlosťou konvergencie použitím Jacobiho a Gauss-Seidelovej metódy prehlbuje.

3.3 Porovnanie výsledkov

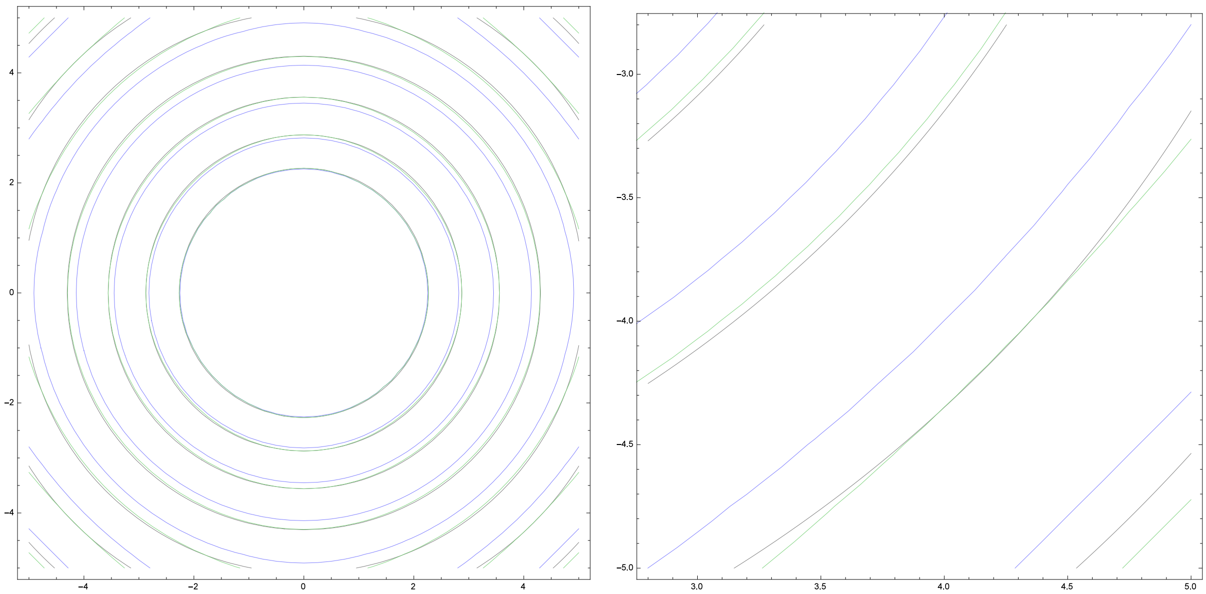
V tabuľke (3.3) vidíme, že riešením stacionárnej rovnice sa nám podarilo získať presnejší výsledok. Na obrázku (3.4) je tiež vidieť, že zanedbanie krivosti na hranici oveľa viac ovplyvnilo riešenie stacionárnej rovnice, ktoré je v bodoch vzdialených od hranice takmer presné. Z dôvodu nepresnosti riešenia na hranici nás zaujímala presnosť riešenie len pre tie izočiary, ktoré sa celé nachádzajú vo vnútri siete. Zvolili sme izočiary, pre ktoré platí $T_{ij} \leq 4.8$.

Porovnanie presnosti (vo vnútri siete)				
Počet dielikov	Nestacionárna rovnica		Stacionárna rovnica	
N	norm	EOC	norm	EOC
10	26.3406	-	13.7031	-
20	17.3344	0.60	5.1046	1.42
40	9.2704	0.90	1.4101	1.86
80	4.8333	0.94	0.3617	1.96

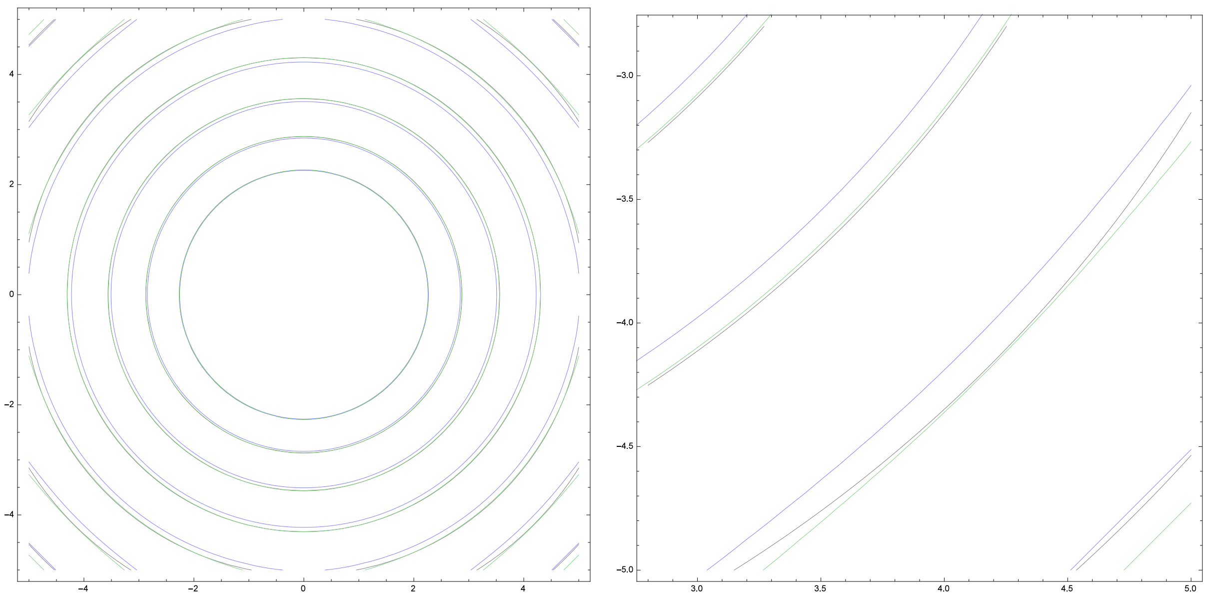
Tabuľka 3.3: *Tabuľka presnosti riešenia vo vnútri siete získaného riešením nestacionárnej rovnice a stacionárnej rovnice.*

Porovnanie presnosti (na celej sieti)				
Počet dielikov	Nestacionárna rovnica		Stacionárna rovnica	
N	norm	EOC	norm	EOC
10	39.1057	-	16.8395	-
20	26.2496	0.58	6.4003	1.40
40	14.7628	0.83	1.9810	1.69
80	7.5280	0.97	0.8679	1.19

Tabuľka 3.4: *Tabuľka presnosti riešenia na celej sieti získaného riešením nestacionárnej rovnice a stacionárnej rovnice.*



Obr. 3.3: *Grafické znázornenie stacionárnych numerických riešenení získaných riešením nestacionárnej (modrá) a stacionárnej (zelená) rovnice pre $N = 40$. Vpravo vidíme detail pravého dolného rohu grafu.*



Obr. 3.4: *Grafické znázornenie stacionárnych numerických riešenení získaných riešením nestacionárnej (modrá) a stacionárnej (zelená) rovnice pre $N = 80$. Vpravo vidíme detail pravého dolného rohu grafu.*

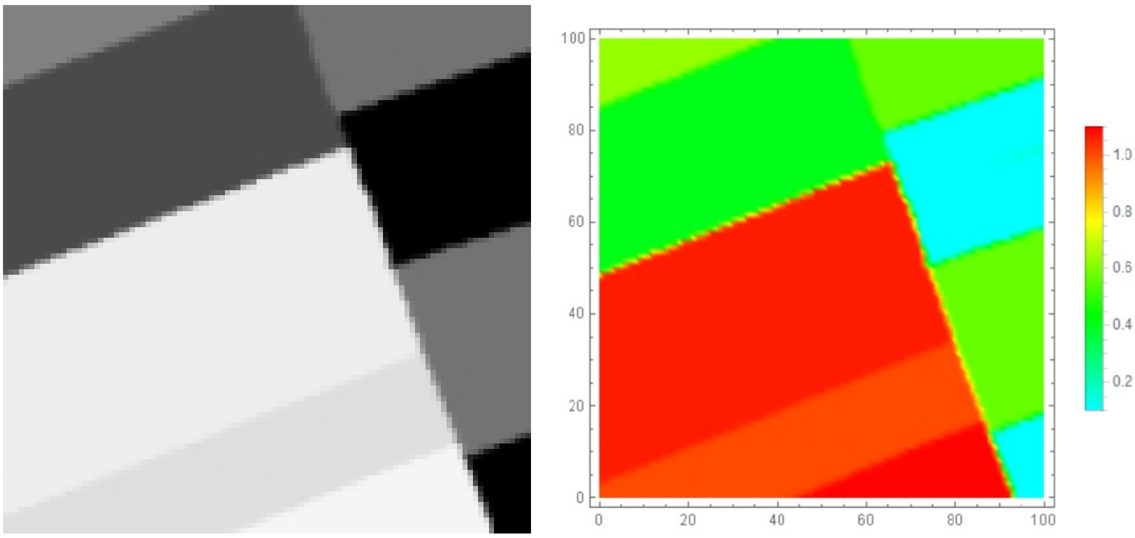
Kapitola 4

Riešenie príkladu z praxe

V tejto kapitole aplikujeme spomenuté numerické metódy na príklad z praxe. Vďaka spolupráci s Botanickým ústavom SAV a útvarom Vojenských lesov a majetkov SR v Malackách máme k dispozícii mapu rovinného terénu oblasti VO Záhorie, na ktorej odtieň šedej reprezentuje horlavosť podložia (tmavší odtieň predstavuje väčšiu horlavosť). Rýchlosť v našom modeli bude

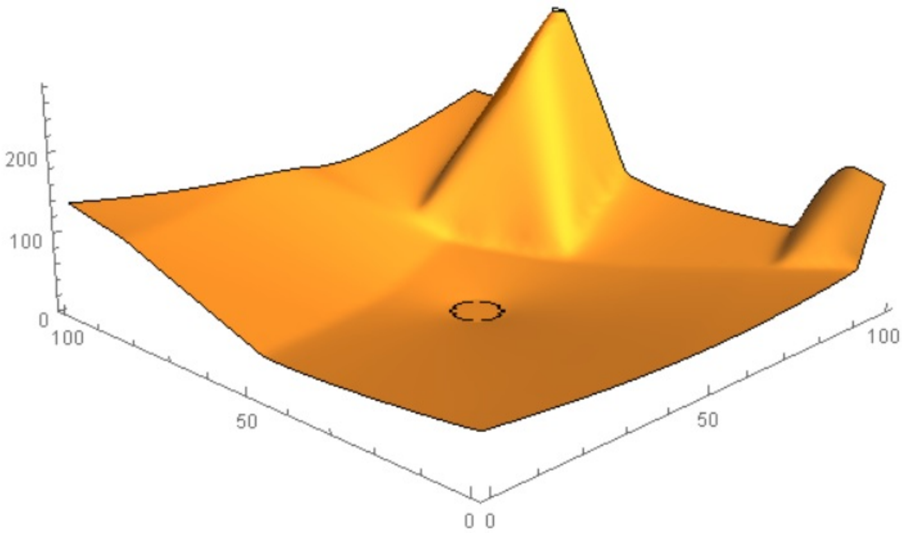
$$V = v(x_i, y_j)(a - b\kappa_{ij}^k), \tag{4.1}$$

kde $v(x_i, y_j)$ je horlavosť podložia v bode (x_i, y_j).

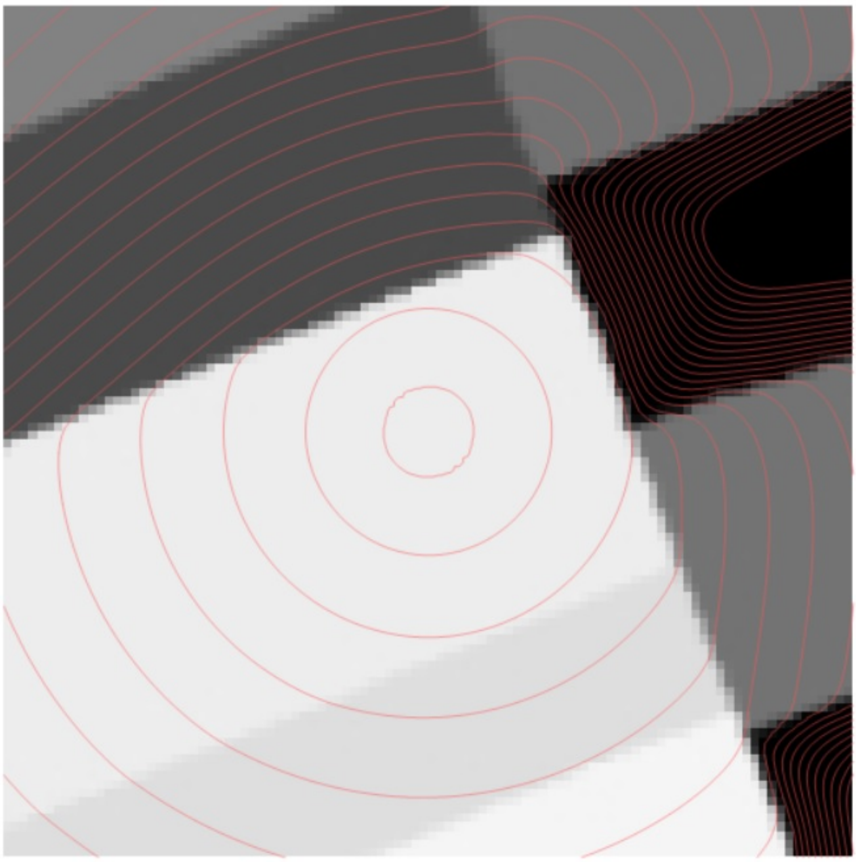


Obr. 4.1: *Vľavo vidíme mapu rovinného terénu v odtieňoch sivej reprezentujúcich horlavosť podložia a vpravo graf rýchlosti $v(x_i, y_j)a$.*

Najskôr budeme riešiť nestacionárnu rovnicu (1.6). Počiatok súradnicovej sústavy umiestníme do ľavého dolného rohu mapy a za počiatočnú podmienku si zvolíme kružnicu $\sqrt{(x-50)^2 + (y-50)^2} - 5$. Úlohu budeme riešiť numerickou metódou a s okrajovými podmienkami opísanými v časti 2.1. Na diskretizáciu krivosti použijeme metódu konečných objemov (2.15).

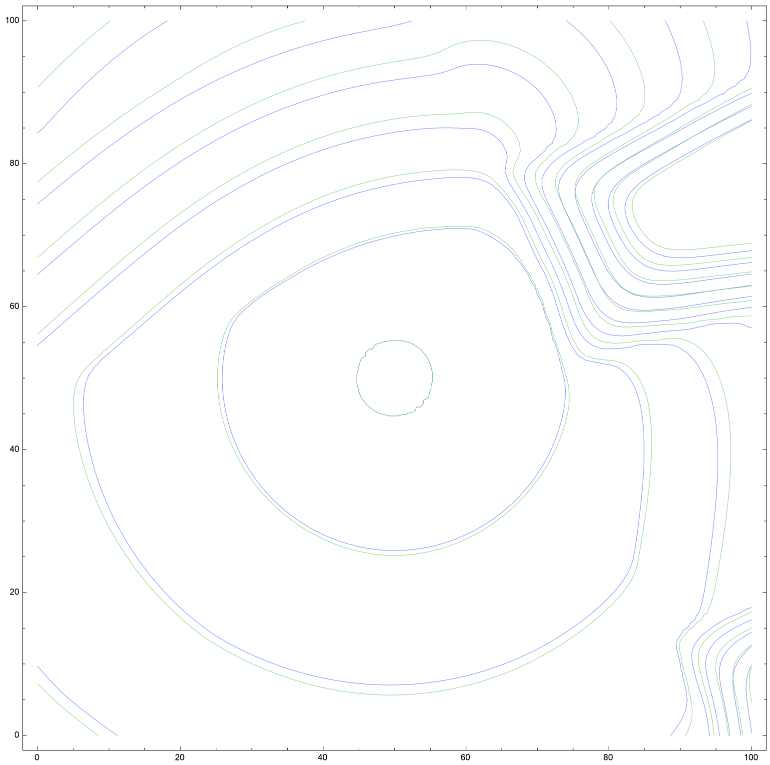


Obr. 4.2: *3D graf stacionárneho riešenia získaného lineárnou interpoláciou časov príchodu získaných riešením nestacionárnej rovnice.*

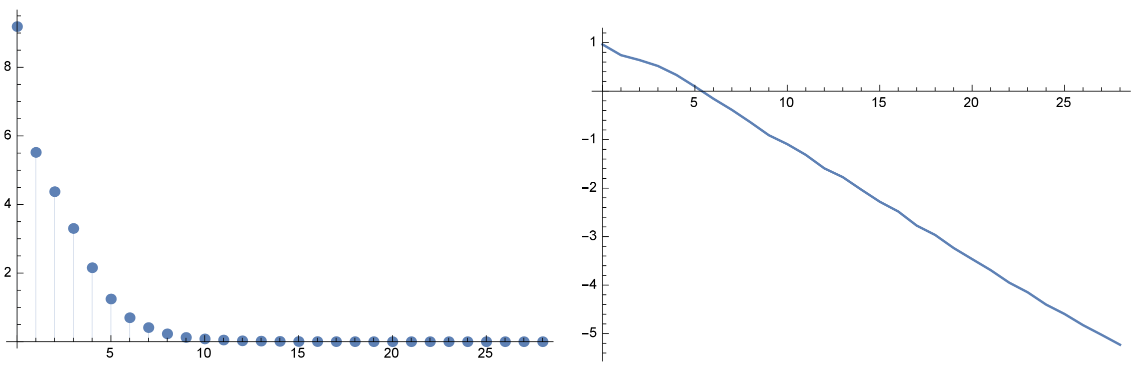


Obr. 4.3: *Izočiary stacionárneho riešenie vykreslené spolu s mapou rovinnej oblasti.*

Riešme teraz stacionárnu rovnicu (1.10) numerickou metódou popísanou v časti 2.2. Za počiatočný odhad si zvoľme riešenie získané predchádzajúcim postupom, riešením nestacionárnej rovnice, a systém lineárnych rovníc riešme Gauss-Seidelovou metódou (2.27).



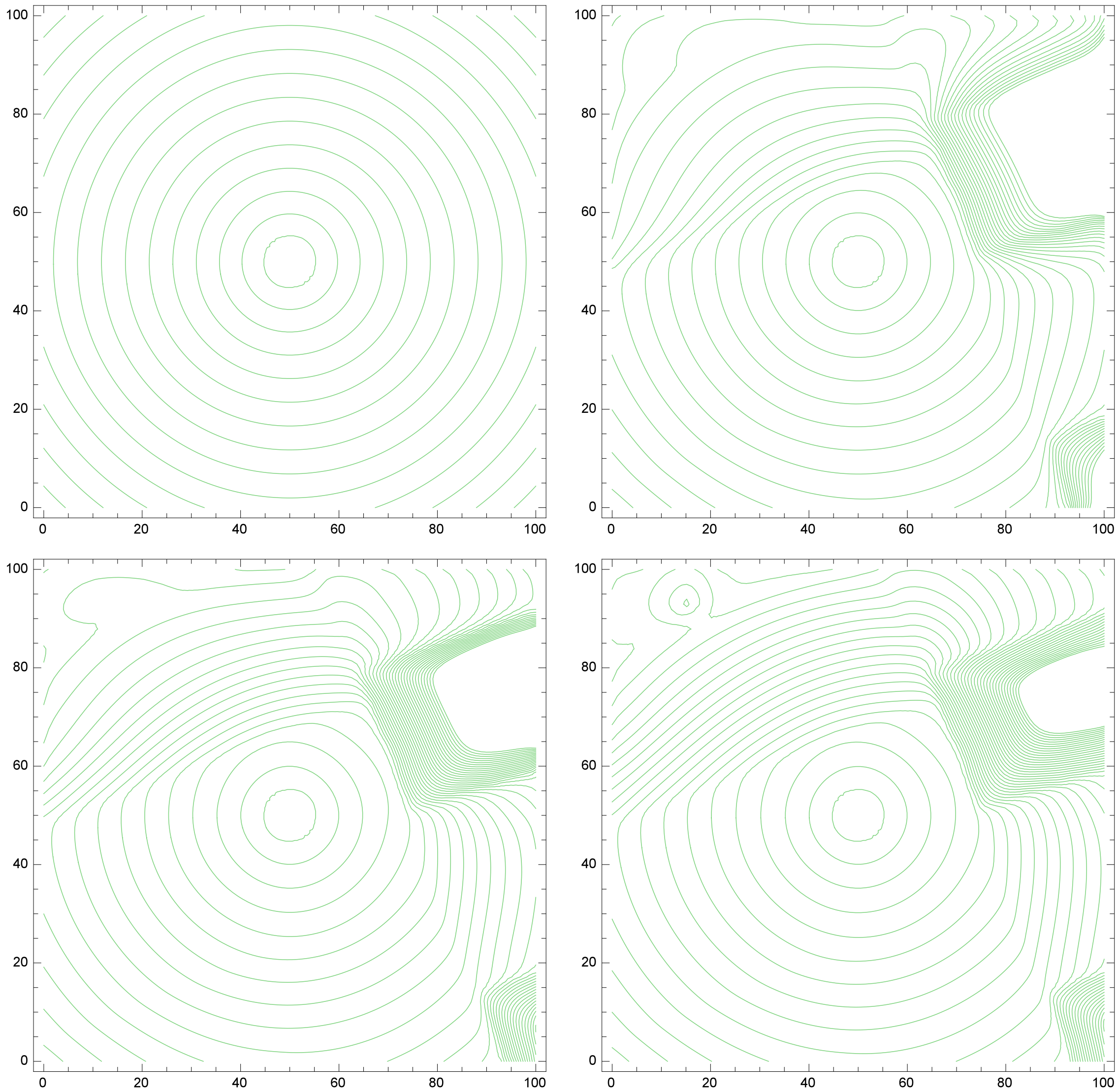
Obr. 4.4: *Grafické porovnanie riešení. Modrou je znázornený výsledok získaný riešením nestacionárnej rovnice a zelenou stacionárnej rovnice.*



Obr. 4.5: *Norma rezídua* (2.25) *v jednotlivých iteráciach linearizácie. Vľavo je graf hodnôt normy rezídua a vpravo sú normy rezídua v logaritmickej mierke.*

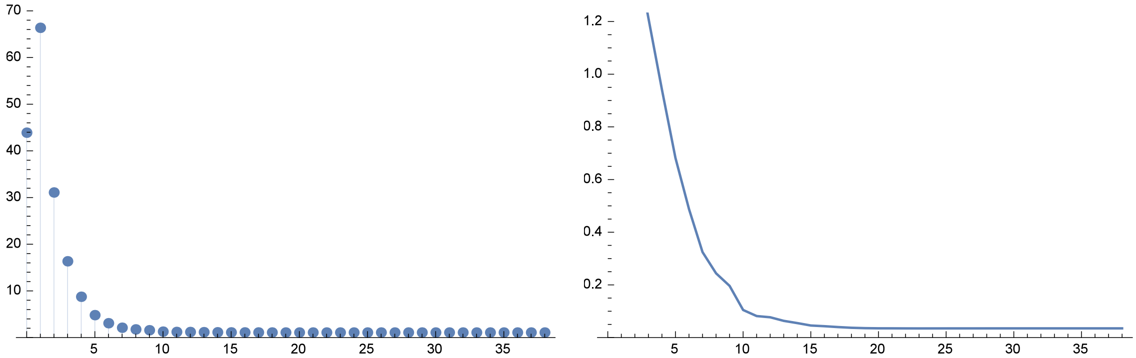
Pozorujeme (viď obr. (4.4)) rovnaký jav ako v časti 3.3. Riešenie získané zo stacionárnej formulácie zvyšuje časy príchodu získané z nestacionárnej formulácie. Nakoľko v predchádzajúcom prípade sa riešenie spresnilo, dovolíme si predpokľadať, že aj teraz je tomu tak.

Pokúsili sme sa riešiť rovnakú úlohu s menej presným počiatočným odhadom. Zvolili sme si zaň rovnicu (3.1), kde r je $\sqrt{(x-50)^2+(y-50)^2}$ a $r^0 = 5$. Ukázalo sa, že numerická metóda s týmto nepresným počiatočným odhadom a so zvolenými počiatočnými podmienkami neprináša správne výsledky.

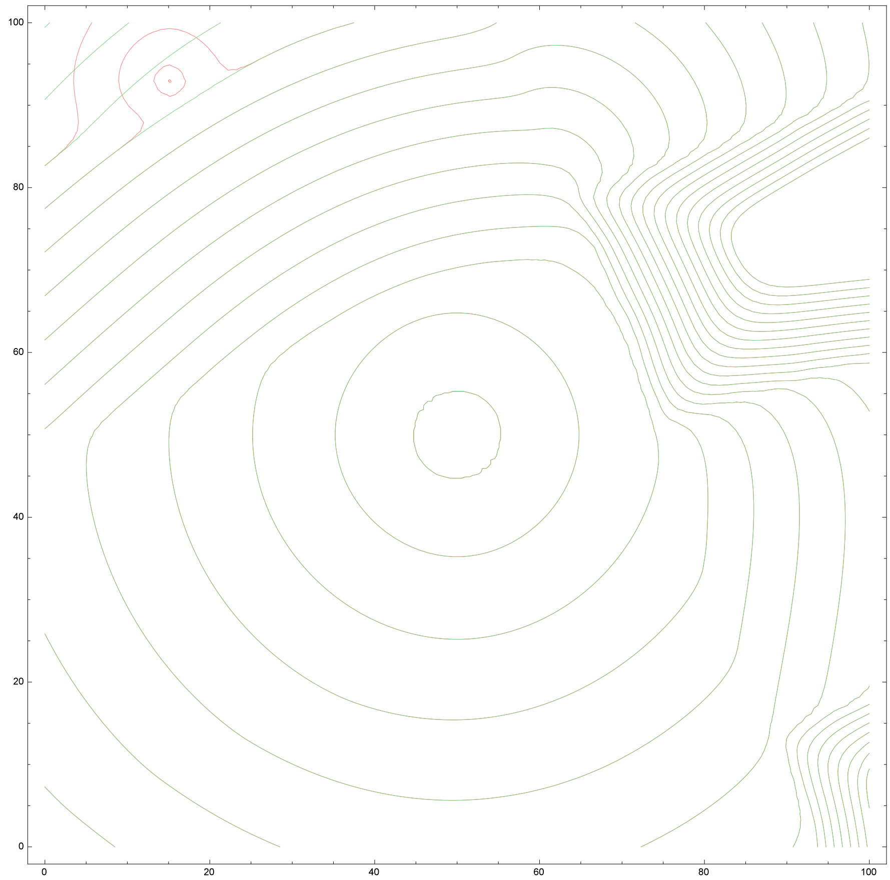


Obr. 4.6: *Vývoj riešenia naprieč linearizačným cyklom. Vľavo hore je riešenie T^0, vpravo hore T^1, vľavo dole T^2 a vpravo dole T^3*

V ľavom hornom rohu grafu riešenia (viď obr. (4.6)) vidíme deformáciu. Norma rezídua tohto riešenia neklesla pod hodnotu 1 (obr. (4.7)). V miestach, kde sa vyskytla deformácia (krúžky), pozorujeme zápornú rýchlosť (4.1), čo je pravdepodobne dôvodom, prečo norma rezídua neklesá. Vo väčšine bodov siete je napriek tomu toto riešenie identické riešeniu získané z presnejšieho počiatočného odhadu (obr. (4.8)).

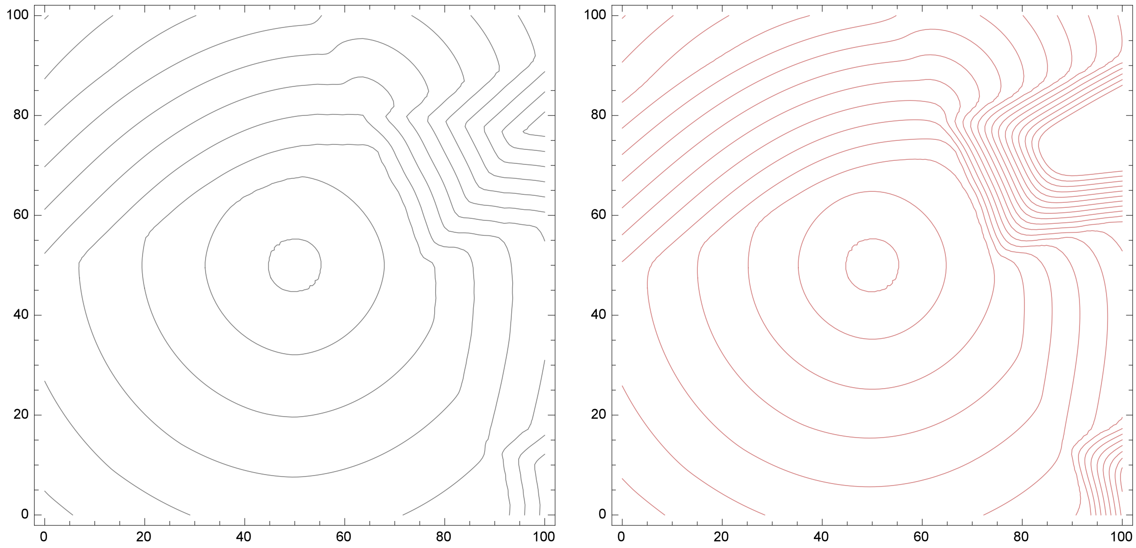


Obr. 4.7: *Norma rezídua* (2.25) *riešenia s nepresným počiatočným odhadom v jednotlivých krokoch linearizácie. Vľavo je graf hodnôt normy rezídua a vpravo sú normy rezídua v logaritmickej mierke.*



Obr. 4.8: *Grafické porovnanie stacionárnych riešení riešených s rôznymi počiatočnými odhadmi. Zelenou je vykreslené riešenie s počiatočným odhadom získáným riešením nestacionárnej rovnice a červenou riešenie, kde počiatočným odhadom bola rovnica* (3.1).

Na záver tejto kapitoly by sme radi poukázali na vplyv členu s krivosťou na riešenie modelu v praxi. Vidíme (obr. (4.9)), že zahrnutím krivosti do modelu sa nám podarilo vyhľadiť ostré "rohy"na izočiarách. Taktiež sa zlepšil tvar kružníc uprostred grafu, ktoré mali v modeli nezahŕňajúcom krivosť tendenciu deformovať sa (špicatiť sa) v smere osí súradnicovej siete.



Obr. 4.9: *Vľavo vidíme riešenie modelu nezahŕňajúcom krivosť a vpravo riešenie modelu s krivosťou.*

Záver

V tejto práci sme sa zaoberali numerickým výpočtom funkcie času prvého príchodu pre úlohy s krivosťou.

Prvým cieľom práce bolo *riešiť nestacionárnu rovnicu metódou úrovňovej množiny*, ktorej formuláciu sme odvodili v časti 1.2. Numerický tvar funkcie času prvého príchodu sme získali numerickou metódou podrobne opísanou v časti 2.1. Stacionárne riešenie funkcie času prvého príchodu sme získali *lineárnou interpoláciu časov príchodu nestacionárneho riešenia*. Dospeli sme k finálnemu riešeniu (časť 3.1), v ktorom sme zároveň porovnali dve rôzne metódy diskretizácie krivosti: *metódu konečných diferencií* a *metódu konečných objemov*. Metóda konečných diferencií sa na konkrétnom príklade javila ako presnejšia, no metóda konečných objemov je kľúčová pri riešení stacionárnej rovnice, čo nás privádza k ďalšiemu cieľu tejto práce.

Druhým cieľom bolo *riešiť stacionárnu rovnicu metódou úrovňovej množiny*. Formuláciu tejto úlohy sme odvodili v časti 1.3 a jej riešeniu pomocou numerickej metódy sme sa venovali v časti 2.2. Popísali sme tri rôzne iteračné metódy: *Jacobiho metódu*, *Gauss-Seidelovu metódu* a *metódu Fast Sweeping*. Všetky tri sme aplikovali pri riešení konkrétneho príkladu a v časti 3.2 sme porovnali dosiahnutú rýchlosť konvergencie. Gauss-Seidelova metóda sa ukázala ako najviac efektívna iteračná metóda pre riešenie nášho systému lineárnych rovníc.

Riešili sme modelový príklad a *porovnali sme obe metódy výpočtu funkcie času prvého príchodu*. Ukázalo sa, že s použitím vhodného počiatočného odhadu je riešenie stacionárnej rovnice presnejšie ako riešenie nestacionárnej rovnice. Touto metódou sa však dajú riešiť len úlohy, kde rýchlosť V je v každom bode siete kladná, na rozdiel od riešenia nestacionárnej rovnice, pri ktorej rýchlosť môže byť ľuboľná. Ďalšou výhodou riešenia nestacionárnej rovnice je použitie metódy konečných diferencií na aproximáciu krivosti, ktorá je priamočiarejšia a ako sa ukázalo, môže byť presnejšia než metóda konečných objemov. Pre úplnosť tiež zopakujme, že zvoliť vhodný počiatočný odhad pri riešení stacionárnej úlohy nie je triviálne. Ako dobrý nápad sa ukazuje použiť práve riešenie nestacionárnej rovnice transformované na stacionárne, ktoré riešením stacionárnej rovnice spresníme (viď tabuľku (3.3)).

V poslednej kapitole 4 sme naše *pozorovania testovali na praktickom príklade*, čo bolo zároveň posledným cieľom tejto práce. Riešili sme príklad, v ktorom sa rýchlosť V

menila v závisloti od úrovne šedi na mape reprezentujúcej horlavosť rovinného terénu. Opäť sa ukázalo výhodné riešiť stacionárnu rovnicu, kde počiatočným odhadom bolo riešenie získané z nestacionárnej formulácie. Pri použití nepresného počiatočného odhadu bolo riešenie pri daných počiatočných podmienkach nepresné. V závere kapitoly sme poukázali na vplyv členu s krivosťou na celkový model. Pozorovali sme vyhladenie kriviek grafu a výrazné zjemnenie deformácií na kružniciach vyskytujúce sa v modeloch neuvažujúcich krivosť. Model uvažujúci krivosť je naviac fyzikálne presnejší.

Zoznam použitej literatúry

[1] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer 2003.

[2] J. Sethian. *Fast Marching Methods and Level Set Methods for Propagating Interfaces.* von Karman Institute Lecture Series, Computational Fluid Mechanics, 1998.

[3] Z. Krivá, K. Mikula, O. Stašová. *Spracovanie obrazu, Vybrané kapitoly z prednášok.* Slovenská Technická Univerzita v Bratislave, 2016.

[4] R. J. LeVeque *Finite Difference Methods for Ordinary and Partial Differential Equations.* University of Washington Seattle, Washington, 1955

[5] G. Okša *Úvod do numerických metód lineárnej algebry.* Slovenská Technická Univerzita v Bratislave, 2009.

[6] H. Zhao A fast sweeping method for eikonal equations. *Mathematics of computation*, 74.250 (2005): 603-627.

[7] M. Balažovjech, P. Frolkovič, R. Frolkovič, K. Mikula *Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems.* Springer, 2014.

[8] K. Mikula, M. Medľa, M. Balažovjech, M. Ambroz *Numerical modeling of wildland surface fire propagation by evolving surface curves.* Advances in Computational Mathematics, 2018

[9] K. Mikula, M. Balažovjech, M. Petrášová, J. Urbán *Lagrangean method with topological changes for numerical modelling of forest fire propagation*, ALGORITMY 2012. Publishing House of STU, 2012

[10] M. Bohunčák *Numerické riešenie úloh advekcie v prostredí DUNE*, Diplomová práca, 2011

[11] R. Blaschke *Numecký výpočet funkcie času príchodu pre hranicu lesného požiaru*, Diplomová práca, 2016

[12] P. Frolkovič, K. Mikula Flux-based level set method: a finite volume method for evolving interfaces, *Applied Numerical Mathematics*, 4(57), 436-454, 2007