

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V
BRATISLAVE**

STAVEBNÁ FAKULTA

**NUMERICKÉ MODELOVANIE ŠÍRENIA LESNÝCH
POŽIAROV**

BAKALÁRSKA PRÁCA

Kód: SvF-5342-58955

Študijný program: Matematicko-počítačové modelovanie
Študijný odbor: 9.1.9 aplikovaná matematika
Pracovisko (katedra/ústav): Katedra matematiky a deskriptívnej geometrie
Vedúci záverečnej práce/školiteľ: Mgr. Martin Balažovjeh, PhD.

Bratislava 2011

Filip Pafčo

Čestné prehlásenie:

Čestne prehlasujem, že som bakalársku prácu „Numerické modelovanie šírenia lesných požiarov“ vypracoval samostatne s použitím uvedenej literatúry a s odbornou pomocou vedúceho práce.

Vlastnoručný podpis

Pod'akovanie:

Týmto by som sa chcel poďakovať Mgr. Martinovi Balažovjehovi, PhD. za cenné pripomienky a odborné konzultácie počas vypracovania tejto práce.

Abstrakt:

Témou práce je riešenie problému topologických zmien pri spájaní viacerých požiarov, ktoré sa šíria z viacerých miest. Čelo požiaru je reprezentované rovinnou krivkou, riadenou externými silami a jej krivosťou. Pre simulovanie jej vývoja riešime numericky nelineárnu parciálnu diferenciálnu rovnicu Lagrangeovským prístupom. Na výpočet používame numerickú schému vyššieho rádu Crank-Nicolsonovho typu. Výhodou Lagrangeovského prístupu je efektivita výpočtu. Tento však prirodzene nezahŕňa topologické zmeny. Cieľom práce je napísať vlastný program na riešenie pohybovej rovnice pre šírenie požiaru, ktorá zahŕňa algoritmus pre topologické zmeny týkajúce sa delenia kriviek a vytvoriť vlastný algoritmus pre spájanie kriviek a otestovať ho na príkladoch pre šírenie požiaru.

Abstract:

Our work is focused on a treatment of topological changes arising in the fire front merging when the fire spreads from several different places. Fire front is represented by a plane curve driven by external forces and curvature. For simulation of curve evolution we solve numerically a nonlinear partial differential equation by the so-called Lagrangean approach. For solving the equation we use a higher order Crank-Nicolson type scheme. The advantage of the Lagrangean approach is the efficiency of computation. On the other hand, it does not treat naturally topological changes which may arise in front propagation. The aim of the work is to write own program for solving the equation of motion of fire front propagation, which includes algorithm for topological changes of curve splitting and to create own algorithm for merging evolving curves and test it on examples of the fire front propagation.

Obsah

Úvod	1
Numerické metódy modelovania šírenia lesných požiarov	2
Farsite	2
Vývoj krivky v rovine	3
Pohybová rovnica	4
Numerická diskretizácia pohybovej rovnice	6
Riešiteľnosť sústavy	10
Vstupné údaje	11
Ciele Práce	11
Topologické zmeny pri šírení lesných požiarov	12
Delenie krivky	12
Algoritmus pre delenie kriviek	13
Popis algoritmu delenia	14
Spájanie viacerých kriviek	16
Algoritmus pre spájanie kriviek	17
Popis algoritmu spájania	18
Zhodnotenie dosiahnutých výsledkov	20
Záver	21
Literatúra	22

Úvod

Lesné požiare každoročne sužujú krajinu po celom svete. Pre niekoho to môže predstavovať "len" vypálené porasty, avšak lesy tvoria tzv. "pľúca Zeme". Sú zdrojom kyslíka potrebného pre všetok život na našej planéte. Sú domovom pre množstvo rozmanitých druhov živočíchov či rastlín a spolu tak vytvárajú jedinečné ekosystémy, hodné ochrany. Nekontrolované lesné požiare rovnako ako na prírode zanechávajú škody na majetkoch obyvateľstva, v najhorších prípadoch aj straty na ľudských životoch.

Na vzniku požiaru sa podieľajú tri hlavné faktory - kyslík, palivo a iniciátor vzniku požiaru. Tým môžu byť blesky, vulkanické aktivity, v obdobiach sucha aj samotné slnko, kde dochádza k spontánnemu samovznieteniu, či iskry spôsobené padaním skál. Napriek tomu však najčastejšie stojí za vznikom požiaru samotný človek. Nedbalou manipuláciou s otvoreným ohňom, ohorkom z cigarety, úmyselným podpaľovaním (pyrománia) dokáže prerásť malý plameň v katastrofický scenár.

Podstatný vplyv na šírenie požiaru má vietor, ktorý ho zásobuje kyslíkom. Za bezvetria sa oheň šíri rádovo rýchlosťou 1m/min. Pri vetre rýchlosť šírenia výrazne stúpa. Ďalším dôležitým faktorom je palivo - horľavý materiál a jeho vertikálne usporiadanie. Pri horení smerom do kopca dochádza k rýchlejšiemu vysušovaniu a tým aj šíreniu požiaru ako z kopca.

Podľa paliva rozlišujeme niekoľko druhov požiarov:

- **podzemný** - najčastejšie sa vyskytuje na rašeline, koreňoch stromov a inom podzemnom organickom materiále. Rýchlosť je pomerne malá, len zopár metrov za deň, avšak takýto požiar je ťažko lokalizovateľný a kontrolovateľný.
- **pozemný** - horí lístie, popadané ihličie, konáre a krovinatý porast. Je najčastešie sa vyskytujúcim druhom požiaru u nás. Je podporovaný hlavne vetrom, pričom požiar má zvyčajne eliptický charakter.
- **korunový** - plameň sa šíri korunami stromov. Vznikajú vzdušné prúdy a miestne vetrom vyvolané požiare. Takýto požiar je veľmi nebezpečný a jeho hasenie náročné.

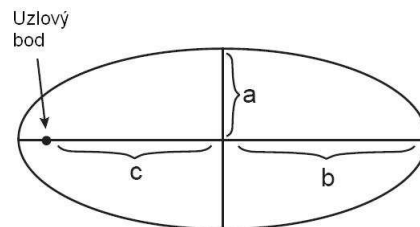
Protipožiarne aktivity závisia od druhu požiaru. Pri pozemných požiaroch sa väčšinou využíva hasenie z povrchu a vytváranie tzv. protipožiarnych pásov. Pri korunových požiaroch už takéto opatrenia nezaberajú, vytvárajú sa izolačné pásy. Takéto pásy predstavujú 30 - 50 metrov široký pás odstránený o horľavý materiál. V takýchto prípadoch je nutné použitie leteckej techniky.

V posledných rokoch dochádza k vývoju počítačových softvérov, ktoré sú schopné predikovať šírenie lesného požiaru, čím dokážu protipožiarne sily účinnejšie a včasne reagovať na možné nebezpečenstvo. Ich nedostatky však spočívajú v čase, ktorý je potrebný na výpočet.

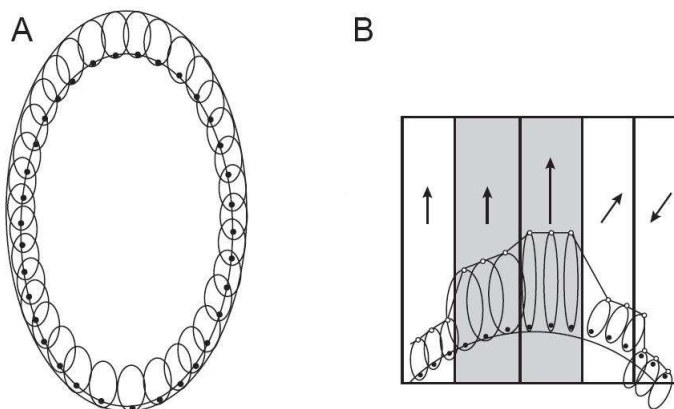
Numerické metódy modelovania šírenia lesných požiarov

Farsite

Farsite je momentálne najznámejší a najpoužívanejší softvér pre simuláciu šírenia lesných požiarov používajúci empiricko-matematický model založený na Huygensovom princípe [7]. Tento princíp sa používa pri šírení vln, kde podstatnú úlohu zohráva myšlienka, že každý bod takejto vlny je zdrojom jej ďalšieho šírenia. V prípade softvéru FARSITE sa však namiesto kružníc využívajú elipsy. Osy tejto elipsy, jej orientácia a vzdialenosť uzlového bodu záležia od vplyvov ako sú horľavosť materiálu, sila a smer vetra (Obr. 1, Obr. 2.B). Čelo požiaru je reprezentované krivkou, po ktorej su rozmiestnené uzlové body. Pre každý takýto bod sú potom charakteristické jednotlivé elipsy (Obr. 1.A). Následne sa vytvorí obálka takto vytvoreného polygónu a celý postup sa opakuje, čím sa dostatočným počtom krokov dokáže vytvoriť simulácia šírenia čela lesného požiaru.



Obr. 1: Osy elipsy a vzdialenosť uzlového bodu od stredu.



Obr. 2: A - Využitie Huygensovho princípu pri simulácii krivky, B - vplyv horľavosti materiálu, sila a smer vetra.

Okrem už spomínaných faktorov zohľadňuje pri materiále jeho aktuálny stav (vlhkosť), teplotu a vlhkosť vzduchu, no taktiež sklon terénu. Softvér FARSITE rieši už spomínané pozemné a korunové druhy lesných požiarov.

Vývoj krivky v rovine

Pre vývoj kriviek sa zvyčajne využívajú dva prístupy a to Eulerovský a Lagrangeovský. Obidva však majú svoje pozitíva i negatíva.

Eulerovský: Pri takomto prístupe je výpočtová sieť reprezentovaná pevne zvolenými mriežkovými bodmi rovnomerne rozdelenými na výpočtovej oblasti Ω . Vývoj krivky Γ , ktorá tvorí podmnožinu oblasti Ω je daný riešením diferenciálnych rovníc pre celú diskretizovanú oblasť Ω . Táto metóda kladie veľké nároky na výpočtový čas. Výhodou je, že vo svojej podstate zahŕňa topologické zmeny. Príkladom je *level-set* metóda.

Lagrangeovský: V tomto prípade je výpočtová sieť tvorená len pohyblivými bodmi, ktoré reprezentujú krivku Γ . Vďaka tomu oproti Eulerovskému prístupu podstatne znižuje pamäťové nároky rovnako ako aj časovú náročnosť. Na druhej strane však nezahŕňa topologické zmeny a je ich pre tento model nutné naprogramovať dodatočne. Odvođením numerickej schémy pre vývoj rovinatej krivky s Lagrangeovským prístupom, rovnako ako aj topologickými zmenami, sa budeme zaoberať v nasledujúcich častiach tejto práce.

Pohybová rovnica

V našom prípade čelo požiaru reprezentujeme uzavretou krivkou Γ v dvojrozmernom priestore. Krivka Γ je daná pomocou polohového vektora $\mathbf{r} = [x, y]$. Jej pohyb v každom bode môžeme rozložiť do tangenciálneho a normálového smeru a vystihuje ho parciálna diferenciálna rovnica:

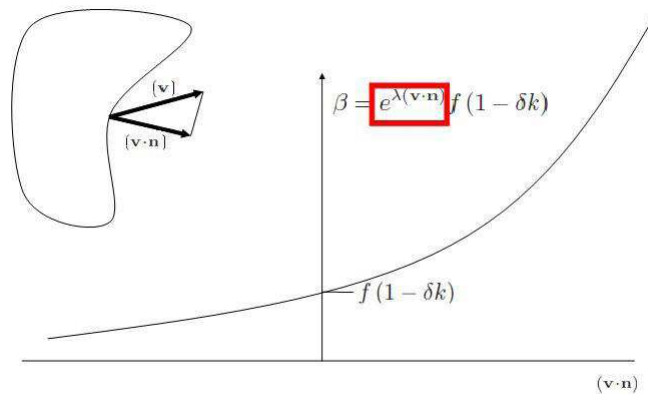
$$\partial_t \mathbf{r} = \beta \mathbf{n} + \alpha \mathbf{t} \quad (1)$$

kde rýchlosť $\partial_t \mathbf{r}$, daná časovou deriváciou polohového vektora, pozostáva z rýchlosti β v normálovom smere \mathbf{n} a z rýchlosti α v tangenciálnom smere \mathbf{t} . Koeficient β priamo vplýva na šírenie krivky, zatiaľ čo koeficient α nemá na šírenie žiaden vplyv a zabezpečuje rovnomerné rozdelenie uzlových bodov pozdĺž celého obvodu krivky a tým aj celkovú stabilitu systému.

Ako už bolo spomínané, na vývoj požiaru majú vplyv veľkosť a smer vetra, rovnako ako aj horľavosť materiálu. Preto musí normálová rýchlosť β zahŕňať takéto parametre.

Z empirického pozorovania šírenia lesných požiarov je známe, že pri prúdení vetra v normálovom smere (v smere šírenia požiaru), rýchlosť požiaru v tomto smere rapídne narastá. Na druhej strane pri prúdení vetra proti smeru normály, rýchlosť klesá. V smere kolmom na čelo požiaru je vplyv vetra nulový (Obr. 4).

Preto je najvhodnejším kandidátom pre popísanie vplyvu vetra na šírenie čela požiaru exponenciálna funkcia tvaru $e^{\lambda(\mathbf{v} \cdot \mathbf{n})}$, kde skalárny súčin $\mathbf{v} \cdot \mathbf{n}$ je priemet vektora rýchlosti vetra na normálu. Parameter λ je miera vplyvu vetra.



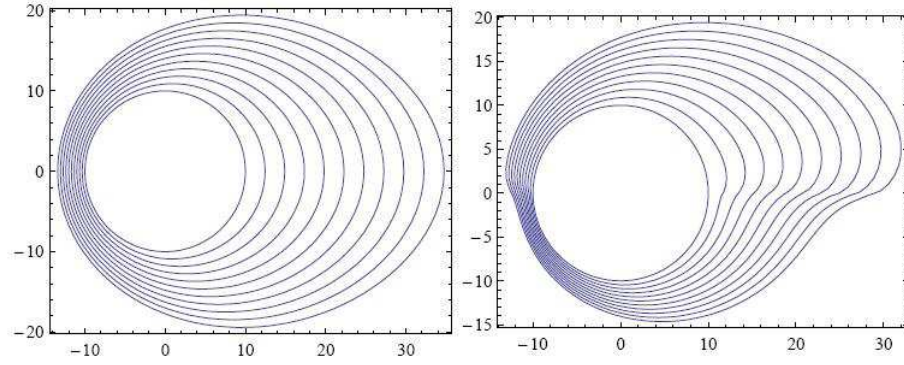
Obr. 3: Projekcia veľkosti a smeru vetra na normálu krivky.

Palivo, čiže horľavosť lesa reprezentuje skalárne rýchlostné pole f . Na vývoj krivky sa rovnako podieľa jej krivosť k , ktorá má vlastnosť krivku zťahovať. Mierou vplyvu krivosti označíme parameter δ .

Normálovú rýchlosť potom definujeme ako súčin týchto zložiek vzťahom:

$$\beta = e^{\lambda(\mathbf{v} \cdot \mathbf{n})} f(1 - \delta k) \quad (2)$$

Takto zvolená β nám zabezpečí, že čelo požiaru neprenikne nehorľavým materiálom.



Obr. 4: Vývoj krivky s vplyvom západného vetra homogénneho a nehomogénne horľavého podkladu.

Keďže platia vzťahy $k\mathbf{n} = -\partial_s\mathbf{t} = -\partial_{ss}\mathbf{r}$, môžeme rýchlosť $\beta\mathbf{n}$ z pohybovej rovnice (1) prepísať v tvare

$$\beta\mathbf{n} = e^{\lambda(\mathbf{v}\cdot\mathbf{n})}f\mathbf{n} + e^{\lambda(\mathbf{v}\cdot\mathbf{n})}f\delta k\partial_{ss}\mathbf{r} \quad (3)$$

Rovnako použitím vzťahov $\mathbf{t} = \partial_s\mathbf{r}$ pre tangenciálu v rovnici (1) a $\mathbf{n} = \mathbf{t}^\perp = (\partial_s\mathbf{r})^\perp$ pre normálu vo vzťahu (3), dostávame pre pohybovú rovnicu nasledujúci tvar:

$$\partial_t\mathbf{r} = \delta f e^{\lambda(\mathbf{v}\cdot\mathbf{n})}\partial_{ss}\mathbf{r} + \alpha\partial_s\mathbf{r} + e^{\lambda(\mathbf{v}\cdot\mathbf{n})}f(\partial_s\mathbf{r})^\perp \quad (4)$$

Tangenciálnu rýchlosť α získame riešením rovnice:

$$\partial_s\alpha = \beta k - \frac{1}{L} \int_{\Gamma} \beta k ds + \left(\frac{L}{g} - 1\right)\omega \quad (5)$$

Kde časť $\beta k - \frac{1}{L} \int_{\Gamma} \beta k ds$ nám zabezpečuje zachovanie pomeru dĺžok elementov krivky Γ . Zatiaľ čo druhá časť $\left(\frac{L}{g} - 1\right)\omega$ zabezpečuje rovnomernú redistribúciu uzlových bodov po obode krivky, čím sa upevňuje stabilita systému. Relaxačný parameter ω určuje rýchlosť redistribúcie uzlových bodov, g je parametrizácia.

Voľba vhodnej tangenciálnej rýchlosti je popísaná v článkoch [5, 8].

Numerická diskretizácia pohybovej rovnice

Pomocou metódy konečných objemov odvodíme numerickú schému pre vývoj krivky.

Pohybovú rovnicu (4) zintegrujeme na oblasti $\left[\mathbf{r}_{i-\frac{1}{2}}, \mathbf{r}_{i+\frac{1}{2}} \right]$, pričom $\mathbf{r}_{i-\frac{1}{2}}$ predstavuje stred elementu medzi bodmi \mathbf{r}_{i-1} a \mathbf{r}_i , podobne pri $\mathbf{r}_{i+\frac{1}{2}}$ je stred medzi \mathbf{r}_i a \mathbf{r}_{i+1} . Nech $h_i = |\mathbf{r}_i - \mathbf{r}_{i-1}|$ a $h_{i+1} = |\mathbf{r}_{i+1} - \mathbf{r}_i|$ potom:

$$|\mathbf{r}_{i-\frac{1}{2}} - \mathbf{r}_{i+\frac{1}{2}}| = \frac{h_i + h_{i+1}}{2} ,$$

$$\int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \partial_t \mathbf{r} ds = \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \delta f e^{\lambda(\mathbf{v} \cdot \mathbf{n})} \partial_{ss} \mathbf{r} ds + \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \alpha \partial_s \mathbf{r} ds + \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} e^{\lambda(\mathbf{v} \cdot \mathbf{n})} f (\partial_s \mathbf{r})^\perp ds \quad (6)$$

Keďže hodnoty δ , f , $e^{\lambda(\mathbf{v} \cdot \mathbf{n})}$, α pre rovnicu (6) uvažujeme v bodoch \mathbf{r}_i konštantné, môžeme ich následne vybrať pred integrál.

$$\int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \partial_t \mathbf{r} ds = \delta f e^{\lambda(\mathbf{v} \cdot \mathbf{n})} \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \partial_{ss} \mathbf{r} ds + \alpha \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \partial_s \mathbf{r} ds + e^{\lambda(\mathbf{v} \cdot \mathbf{n})} f \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} (\partial_s \mathbf{r})^\perp ds \quad (7)$$

Jednotlivé časti rovnice (7) zintegrujeme postupne.

1. Použitím divergenčnej vety a centrálnej diferencie dostávame:

$$\int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \partial_{ss} \mathbf{r} ds = [\partial_s \mathbf{r}]_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} = \frac{\partial \mathbf{r}_{i+\frac{1}{2}}}{\partial s} - \frac{\partial \mathbf{r}_{i-\frac{1}{2}}}{\partial s} = \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{h_{i+1}} - \frac{\mathbf{r}_i - \mathbf{r}_{i-1}}{h_i} \quad (8)$$

2.

$$\alpha \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \partial_s \mathbf{r} ds = [\mathbf{r}_i]_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} = (\mathbf{r}_{i+\frac{1}{2}} - \mathbf{r}_{i-\frac{1}{2}}) = \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{2} \quad (9)$$

3. Vypočítame integrál $\int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} (\partial_s \mathbf{r})^\perp ds$. Keďže $\partial_s \mathbf{r} = \left[\frac{dx}{ds}, \frac{dy}{ds} \right]$, pre kolmicu naň platí $(\partial_s \mathbf{r})^\perp = \left[\frac{dy}{ds}, -\frac{dx}{ds} \right]$, preto tento vektor zintegrujeme postupne po zložkách.

$$\int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \frac{dy_i}{ds} ds = [y_i]_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} = (y_{i+\frac{1}{2}} - y_{i-\frac{1}{2}}) = \frac{y_{i+1} - y_{i-1}}{2}$$

Podobným spôsobom dostaneme zintegrovaním zložky $-\frac{dx}{ds}$:

$$\int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} -\frac{dx_i}{ds} ds = - \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \frac{dx_i}{ds} ds = - [x_i]_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} = (x_{i-\frac{1}{2}} - x_{i+\frac{1}{2}}) = \frac{x_{i-1} - x_{i+1}}{2}$$

Označme si vektor týchto zložiek ako

$$\tilde{\mathbf{n}}_i = \left[\frac{y_{i+1} - y_{i-1}}{2}, \frac{x_{i-1} - x_{i+1}}{2} \right] \quad (10)$$

4. Keďže $\partial_t \mathbf{r}$ sa na intervale $(i - \frac{1}{2}, i + \frac{1}{2})$ nemení, zintegrovaním dostaneme:

$$\int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \partial_t \mathbf{r} ds = \frac{h_{i+1} + h_i}{2} \partial_t \mathbf{r}_i \quad (11)$$

Dosadením vztahov (8,9,10,11) pre pohybovú rovnicu (7) dostaneme sústavu obyčajných diferenciálnych rovníc - **semidiskrétnu schému**.

$$\frac{h_{i+1} + h_i}{2} \partial_t \mathbf{r}_i = \delta_i f_i e^{\lambda(\mathbf{v}_i \cdot \mathbf{n}_i)} \left(\frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{h_{i+1}} - \frac{\mathbf{r}_i - \mathbf{r}_{i-1}}{h_i} \right) + \alpha_i \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{2} + e^{\lambda(\mathbf{v}_i \cdot \mathbf{n}_i)} f_i \tilde{\mathbf{n}}_i \quad (12)$$

Z rovnice (8) vidno, že prvá a druhá derivácia polohového vektora vzľadom na dĺžku krivky sú aproximované nasledujúcimi diferenciami:

$$\partial_s \mathbf{r}_i = \frac{\mathbf{r}_{i+1} - \mathbf{r}_{i-1}}{2} / \frac{h_{i+1} + h_i}{2}$$

$$\partial_{ss} \mathbf{r}_i = \left(\frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{h_{i+1}} - \frac{\mathbf{r}_i - \mathbf{r}_{i-1}}{h_i} \right) / \left(\frac{h_{i+1} + h_i}{2} \right)$$

$\mathbf{r}_i = [x_i, y_i]^T$ je polohový vektor pre i -ty uzlový bod krivky Γ pre $i=1, \dots, n$, kde n je počet uzlových bodov krivky, $h_i = |\mathbf{h}_i|$, $\mathbf{h}_i = \mathbf{r}_i - \mathbf{r}_{i-1}$, a kde normálový vektor je daný

$$\mathbf{n}_i = [n_{x_i}, n_{y_i}] = \left(\frac{\mathbf{r}_{i-1} - \mathbf{r}_{i+1}}{h_{i+1} + h_i} \right)^\perp = \left[\frac{y_{i+1} - y_{i-1}}{h_{i+1} + h_i}, \frac{x_{i-1} - x_{i+1}}{h_{i+1} + h_i} \right].$$

Diskretizáciou diferenciálnej rovnice pre skalárnu rýchlosť α v tangenciálnom smere dostaneme:

$$\frac{\alpha_i - \alpha_{i-1}}{h_i} = \beta_{i+\frac{1}{2}} k_{i+\frac{1}{2}} - \frac{\sum_{i=1}^n \beta_{i+\frac{1}{2}} k_{i+\frac{1}{2}} h_i}{\sum_{i=1}^n h_i} + \left(\frac{\sum_{i=1}^n h_i}{nh_i} - 1 \right) \omega$$

kde

$$k_{i+\frac{1}{2}} = \frac{\text{sgn}(\det(\mathbf{h}_{i-1}, \mathbf{h}_{i+1}))}{2h_i} \arccos \left(\frac{\mathbf{h}_{i-1} \cdot \mathbf{h}_{i+1}}{h_{i-1} h_{i+1}} \right)$$

Kôli cyklickosti okrajových podmienok využívame hodnoty pre $\mathbf{r}_0 = \mathbf{r}_n$ a $\mathbf{r}_{n+1} = \mathbf{r}_1$.

Ak aproximujeme časovú deriváciu $\partial_t \mathbf{r}_i$ v rovnici (12) diferenciou v tvare $\frac{\mathbf{r}_i^{m+1} - \mathbf{r}_i^m}{\Delta t}$ dostaneme **úplne diskrétnu schému**.

Základné typy týchto schém sú:

- **Explicitná schéma** - pre výpočet polohového vektora \mathbf{r}_i^{m+1} využíva všetky hodnoty zo starého časového kroku m . Nevýhodou je nestabilita.

$$\frac{h_{i+1}^m + h_i^m}{2} \frac{\mathbf{r}_i^{m+1} - \mathbf{r}_i^m}{\Delta t} = \delta_i f_i^m e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)} \left(\frac{\mathbf{r}_{i+1}^m - \mathbf{r}_i^m}{h_{i+1}^m} - \frac{\mathbf{r}_i^m - \mathbf{r}_{i-1}^m}{h_i^m} \right) + \alpha_i^m \frac{\mathbf{r}_{i+1}^m - \mathbf{r}_{i-1}^m}{2} + e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)} f_i^m \tilde{\mathbf{n}}_i^m$$

- **Plne implicitná schéma** - hodnoty vektora \mathbf{r}_i^{m+1} získame riešením nasledujúceho trojdiagonálneho systému rovníc.

$$\frac{h_{i+1}^{m+1} + h_i^{m+1}}{2} \frac{\mathbf{r}_i^{m+1} - \mathbf{r}_i^m}{\Delta t} = \delta_i f_i^{m+1} e^{\lambda(\mathbf{v}_i^{m+1} \cdot \mathbf{n}_i^{m+1})} \left(\frac{\mathbf{r}_{i+1}^{m+1} - \mathbf{r}_i^{m+1}}{h_{i+1}^{m+1}} - \frac{\mathbf{r}_i^{m+1} - \mathbf{r}_{i-1}^{m+1}}{h_i^{m+1}} \right) +$$

$$+\alpha_i^{m+1} \frac{\mathbf{r}_{i+1}^{m+1} - \mathbf{r}_{i-1}^{m+1}}{2} + e^{\lambda(\mathbf{v}_i^{m+1} \cdot \mathbf{n}_i^{m+1})} f_i^{m+1} \tilde{\mathbf{n}}_i^{m+1}$$

Kde koeficienty nachádzajúce sa v matici systému uvažujeme z nového časového kroku a získame ich pomocou iterácií. Nevýhoda tejto schémy spočíva vo väčšom nároku na výpočtový čas.

- **Semi implicitná schéma** - hodnoty vektora \mathbf{r}_i^{m+1} získame riešením trojdiagonálneho systému rovníc.

$$\begin{aligned} \frac{h_{i+1}^m + h_i^m}{2} \frac{\mathbf{r}_{i+1}^{m+1} - \mathbf{r}_i^m}{\Delta t} = \delta_i f_i e^{\lambda(\mathbf{v}_i \cdot \mathbf{n}_i)} \left(\frac{\mathbf{r}_{i+1}^{m+1} - \mathbf{r}_i^{m+1}}{h_{i+1}^m} - \frac{\mathbf{r}_i^{m+1} - \mathbf{r}_{i-1}^{m+1}}{h_i^m} \right) + \alpha_i^m \frac{\mathbf{r}_{i+1}^{m+1} - \mathbf{r}_{i-1}^{m+1}}{2} + \\ + e^{\lambda(\mathbf{v}_i \cdot \mathbf{n}_i)} f_i^m \tilde{\mathbf{n}}_i^m \end{aligned} \quad (13)$$

Kde koeficienty nachádzajúce sa v matici systému uvažujeme z nového aj starého časového kroku. Takáto semi-implicitná schéma sa bežne používa na výpočet vývoja kriviek [3, 4, 5].

Presnosť týchto schém je prvého rádu.

V našom prípade využívame **Higher order** schému, ktorej presnosť je druhého rádu. Prvý krát bola uvedená v [1] bez tangenciálnej redistribúcie a s tangenciálnou redistribúciou v [2]. Je tvorená ako súčet explicitnej a plne implicitnej schémy.

$$\begin{aligned} \frac{h_{i+1}^m + h_i^m + h_{i+1}^{m+1} + h_i^{m+1}}{2} \frac{\mathbf{r}_{i+1}^{m+1} - \mathbf{r}_i^m}{\Delta t} = \\ = \delta_i^m f_i^m e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)} \left(\frac{\mathbf{r}_{i+1}^m - \mathbf{r}_i^m}{h_{i+1}^m} - \frac{\mathbf{r}_i^m - \mathbf{r}_{i-1}^m}{h_i^m} \right) + \alpha_i^m \frac{\mathbf{r}_{i+1}^m - \mathbf{r}_{i-1}^m}{2} + e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)} f_i^m \tilde{\mathbf{n}}_i^m + \\ + \delta_i^{m+1} f_i^{m+1} e^{\lambda(\mathbf{v}_i^{m+1} \cdot \mathbf{n}_i^{m+1})} \left(\frac{\mathbf{r}_{i+1}^{m+1} - \mathbf{r}_i^{m+1}}{h_{i+1}^{m+1}} - \frac{\mathbf{r}_i^{m+1} - \mathbf{r}_{i-1}^{m+1}}{h_i^{m+1}} \right) + \\ + \alpha_i^{m+1} \frac{\mathbf{r}_{i+1}^{m+1} - \mathbf{r}_{i-1}^{m+1}}{2} + e^{\lambda(\mathbf{v}_i^{m+1} \cdot \mathbf{n}_i^{m+1})} f_i^{m+1} \tilde{\mathbf{n}}_i^{m+1} \end{aligned} \quad (14)$$

Za účelom získania riešenia hodnôt \mathbf{r}_i^{m+1} , $i = 1, \dots, n$, riešime iteračne nasledujúci semi-implicitný cyklický trojdiagonálny systém rovníc.

$$\begin{aligned} \left(\frac{h_{i+1}^m + h_i^m + h_{i+1}^{m(l)} + h_i^{m(l)}}{2\Delta t} + \delta_i^{m(l)} f_i^{m(l)} e^{\lambda(\mathbf{v}_i^{m(l)} \cdot \mathbf{n}_i^{m(l)})} \left(\frac{1}{h_{i+1}^{m(l)}} + \frac{1}{h_i^{m(l)}} \right) \right) \mathbf{r}_i^{m(l+1)} + \\ + \left(\frac{\alpha_i^{m(l)}}{2} - \frac{\delta_i^{m(l)} f_i^m e^{\lambda(\mathbf{v}_i^{m(l)} \cdot \mathbf{n}_i^{m(l)})}}{h_i^{m(l)}} \right) \mathbf{r}_{i-1}^{m(l+1)} - \left(\frac{\alpha_i^{m(l)}}{2} + \frac{\delta_i^{m(l)} f_i^m e^{\lambda(\mathbf{v}_i^{m(l)} \cdot \mathbf{n}_i^{m(l)})}}{h_{i+1}^{m(l)}} \right) \mathbf{r}_{i+1}^{m(l+1)} = \end{aligned}$$

$$\begin{aligned}
&= \left(\frac{h_{i+1}^m + h_i^m + h_{i+1}^{m(l)} + h_i^{m(l)}}{2\Delta t} - \delta_i^m f_i^m e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)} \left(\frac{1}{h_{i+1}^m} + \frac{1}{h_i^m} \right) \right) \mathbf{r}_i^m - \\
&- \left(\frac{\alpha_i^m}{2} - \frac{\delta_i^m f_i^m e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)}}{h_i^m} \right) \mathbf{r}_{i-1}^m + \left(\frac{\alpha_i^m}{2} + \frac{\delta_i^m f_i^m e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)}}{h_{i+1}^m} \right) \mathbf{r}_{i+1}^m + \\
&+ \delta_i^m f_i^m e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)} \tilde{\mathbf{n}}_i^m + \delta_i^{m(l)} f_i^{m(l)} e^{\lambda(\mathbf{v}_i^{m(l)} \cdot \mathbf{n}_i^{m(l)})} \tilde{\mathbf{n}}_i^{m(l)} \tag{15}
\end{aligned}$$

Iteračný proces sa zastavuje, ak rozdiel celkových dĺžok po sebe nasledujúcich iterovaných krivkách je menší ako predpísaná tolerancia. $|L^{m(l)} - L^{m(l+1)}| \leq TOL$. Celková dĺžka je vyjadrená ako $L^{m(l)} = \sum_{i=1}^n h_i^{m(l)}$.

Pre výpočet koeficientov v explicitnej časti schémy využívame vťahy

$$h_i^m = \sqrt{(x_i^m - x_{i-1}^m)^2 + (y_i^m - y_{i-1}^m)^2}, \quad f_i^m = f(\mathbf{r}_i^m)$$

$$\alpha_0^m = 0, \quad \alpha_i^m = \alpha_{i-1}^m + h_i^m \beta_{i+\frac{1}{2}}^m k_{i+\frac{1}{2}}^m - h_i^m \frac{\sum_{i=1}^n \beta_{i+\frac{1}{2}}^m k_{i+\frac{1}{2}}^m h_i^m}{\sum_{i=1}^n h_i^m} +$$

$$+ \left(\frac{\sum_{i=1}^n h_i^m}{n} - h_i^m \right) \omega^m$$

$$\tilde{\mathbf{n}}_i^m = \left[\frac{y_{i-1}^m - y_{i+1}^m}{2}, \frac{x_{i+1}^m - x_{i-1}^m}{2} \right]$$

a upravené koeficienty pre iterácie v implicitnej časti

$$h_i^{m(l)} = \sqrt{(x_i^{m(l)} - x_{i-1}^{m(l)})^2 + (y_i^{m(l)} - y_{i-1}^{m(l)})^2}, \quad f_i^{m(l)} = f(\mathbf{r}_i^{m(l)})$$

$$\alpha_0^{m(l)} = 0, \quad \alpha_i^{m(l)} = \alpha_{i-1}^{m(l)} + h_i^{m(l)} \beta_{i+\frac{1}{2}}^{m(l)} k_{i+\frac{1}{2}}^{m(l)} - h_i^{m(l)} \frac{\sum_{i=1}^n \beta_{i+\frac{1}{2}}^{m(l)} k_{i+\frac{1}{2}}^{m(l)} h_i^{m(l)}}{\sum_{i=1}^n h_i^{m(l)}} +$$

$$+ \left(\frac{\sum_{i=1}^n h_i^{m(l)}}{n} - h_i^{m(l)} \right) \omega^{m(l)}$$

$$\tilde{\mathbf{n}}_i^{m(l)} = \left[\frac{y_{i-1}^{m(l)} - y_{i+1}^{m(l)}}{2}, \frac{x_{i+1}^{m(l)} - x_{i-1}^{m(l)}}{2} \right]$$

Systém riešime pomocou cyklického trojdiagonálneho solveru, ktorý je založený na efektívnej implementácii Gaussovej eliminácie.

Riešiteľnosť sústavy rovníc

Trojdiagonálny cyklický systém rovníc (15) môžeme zapísať v tvare

$$A\mathbf{r}_{i-1}^{m+1} + B\mathbf{r}_i^{m+1} + C\mathbf{r}_{i+1}^{m+1} = D$$

kde koeficienty mimo diagonály sú

$$A = \left(\frac{\alpha_i^{m(l)}}{2} - \frac{\delta_i^{m(l)} f_i^m e^{\lambda(\mathbf{v}_i^{m(l)} \cdot \mathbf{n}_i^{m(l)})}}{h_i^{m(l)}} \right), \quad C = - \left(\frac{\alpha_i^{m(l)}}{2} + \frac{\delta_i^{m(l)} f_i^m e^{\lambda(\mathbf{v}_i^{m(l)} \cdot \mathbf{n}_i^{m(l)})}}{h_{i+1}^{m(l)}} \right),$$

koeficienty na diagonále sú

$$B = \left(\frac{h_{i+1}^m + h_i^m + h_{i+1}^{m(l)} + h_i^{m(l)}}{2\Delta t} - A - C \right) \quad (16)$$

,

a pravá strana

$$\begin{aligned} D = & \left(\frac{h_{i+1}^m + h_i^m + h_{i+1}^{m(l)} + h_i^{m(l)}}{2\Delta t} - \delta_i^m f_i^m e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)} \left(\frac{1}{h_{i+1}^m} + \frac{1}{h_i^m} \right) \right) \mathbf{r}_i^m - \\ & - \left(\frac{\alpha_i^m}{2} - \frac{\delta_i^m f_i^m e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)}}{h_i^m} \right) \mathbf{r}_{i-1}^m + \left(\frac{\alpha_i^m}{2} + \frac{\delta_i^m f_i^m e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)}}{h_{i+1}^m} \right) \mathbf{r}_{i+1}^m + \\ & + \delta_i^m f_i^m e^{\lambda(\mathbf{v}_i^m \cdot \mathbf{n}_i^m)} \tilde{\mathbf{n}}_i^m + \delta_i^{m(l)} f_i^{m(l)} e^{\lambda(\mathbf{v}_i^{m(l)} \cdot \mathbf{n}_i^{m(l)})} \tilde{\mathbf{n}}_i^{m(l)} \end{aligned}$$

Sústava je riešiteľná vtedy, keď je matica sústavy diagonálne dominantná, teda platí

$$|A| + |C| \leq |B|$$

Keďže $B > 0$, môžeme prechádzajúci vzťah zapísať v tvare $|A| + |C| \leq B$. . Ak teraz B vyjadríme dosadením vzťahu (16) dostávame nerovnosť

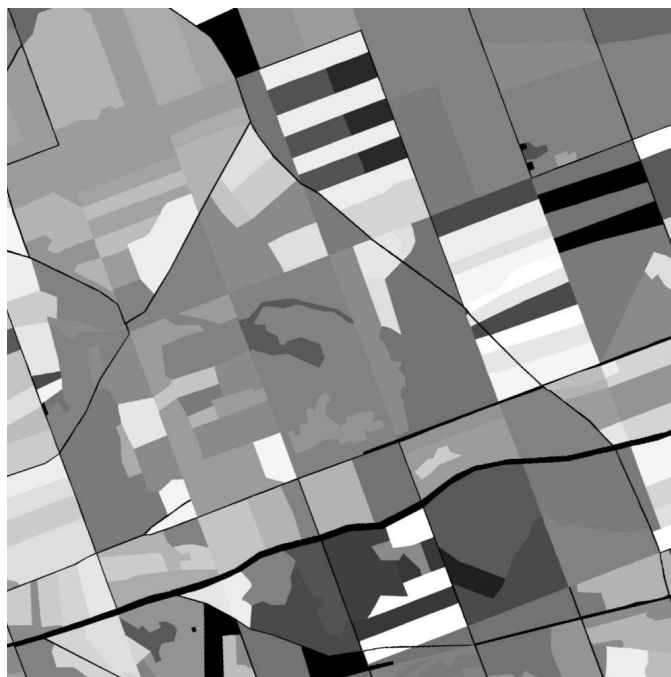
$$|A| + |C| \leq \frac{h_{i+1}^m + h_i^m + h_{i+1}^{m(l)} + h_i^{m(l)}}{2\Delta t} - A - C \quad (17)$$

Zo vzťahu (17) dostaneme podmienku riešiteľnosti sústavy pre časový krok

$$\Delta t = \frac{h_{i+1}^m + h_i^m + h_{i+1}^{m(l)} + h_i^{m(l)}}{2(|A| + |C| + A + C)}$$

Vstupné údaje

Vstupné dáta pre našu prácu bude predstavovať mapa terénu. Táto bola špeciálne pripravená vďaka spolupráci s Mgr. Máriou Petrášovou z Botanického ústavu SAV a útvarom Vojenských lesov a majetkov SR v Malackách. Keďže náš model ešte nezahŕňa vplyv sklonu terénu, je preto rovinatá oblasť VO Záhorie vhodným územím.



Obr. 5: Mapa terénu v odtieňoch šedi reprezentujúcich horľavosť podložia.

Mapa predstavuje terén o rozlohe 1 x 1 km štvorcových, pričom jeden pixel obrázku reprezentuje jeden meter štvorcový. Obrázok je v odtieňoch šedi v hodnotách od 0 po 255, ktoré predstavujú horľavosť podložia. Pri hodnote 0 je materiál nehorľavý, naopak pri hodnote 255 je horľavosť materiálu maximálna.

Vplyv vetra pre náš problém budeme uvažovať s konštatnou veľkosťou a smerom.

Ciele Práce

1. Cieľom našej práce je vyvinúť efektívny algoritmus pre zahrnutie topologických zmien do výpočtu pohybovej rovnice pre šírenie požiaru, ktorý sa týka spájania viacerých požiarov šíracích sa z rôznych miest.
2. Napísať vlastný program na riešenie pohybovej rovnice pre šírenie požiaru, ktorý bude obsahovať aj spomínaný algoritmus zahŕňajúci topologické zmeny.

Topologické zmeny pri šírení lesných požiarov

Ako sme už na začiatku spomínali, topologické zmeny nie sú obsiahnuté v Lagrangeovskom prístupe vývoja kriviek. Pri šírení lesných požiarov dochádza k dvom typom topologických zmien.

Delenie krivky

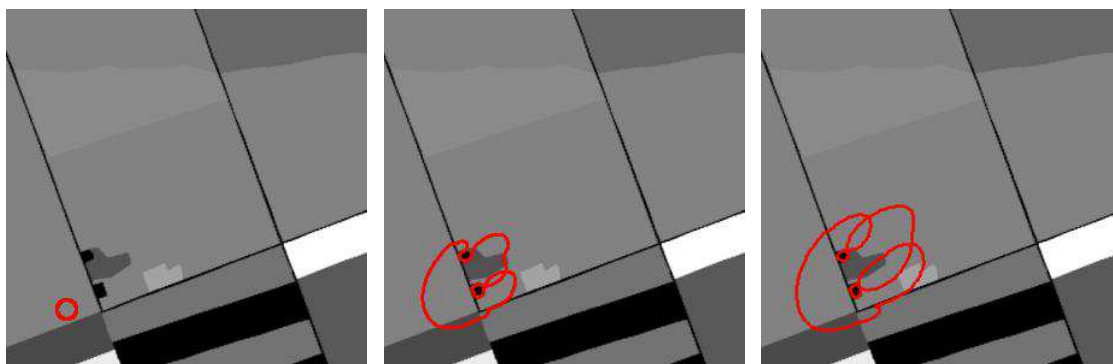
Delenie krivky nastáva napríklad v prípadoch, keď oheň obkolesuje nehorľavú oblasť. Na druhej strane sa dva konce tej istej krivky spoja dokopy, krivka sa rozdelí na dve samostatné, pričom prvá pokračuje smerom od prekážky a druhá krivka smerom k prekážke, pokiaľ sa nestiahne na minimum a zanikne. Takýto algoritmus rovnako ošetruje situácie, keď krivka v niektorých miestach nadobúda silne konvexný charakter. Body na okrajoch konvexného oblúku môžu mať v určitých situáciách dostatočne veľkú normálovú rýchlosť na to, aby stihli “prebehnúť” z jednej strany konvexnej časti na druhú, čím sa vytvorí akási slučka. Takýto efekt je pre simulovanie šírenia lesného požiaru nežiadúci.

Spôsob delenia je nasledovný:

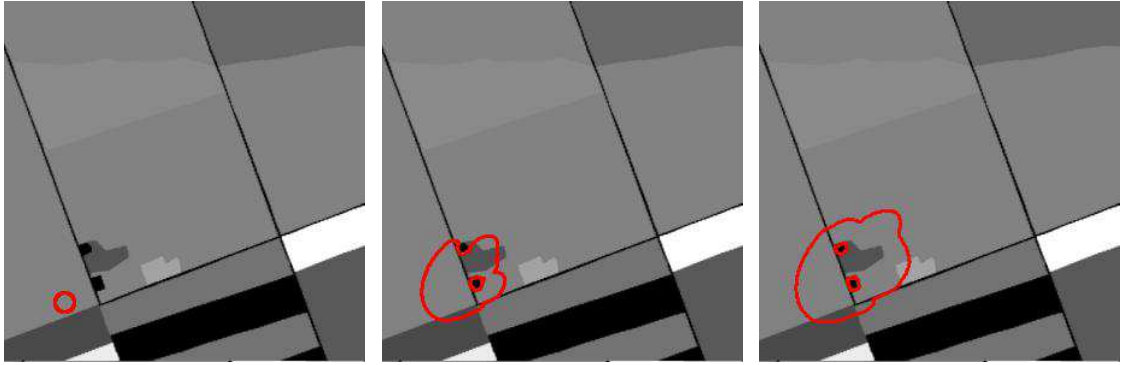
1. Vypočítame najmenšiu vzdialenosť dvoch susedných bodov krivky. $m1 = \min [h_i^m], i = 1, \dots, n$.
2. Nájdeme vzájomnú vzdialenosť všetkých uzlov metódou “každý s každým” okrem susedných uzlov. $m2 = |\mathbf{r}_i - \mathbf{r}_j|, i = 1, \dots, n, j = \dots, i - 3, i - 2, i + 2, i + 3, \dots$.
3. Následne testujeme podmienku $m1 > C m2$, kde C je nejaká konštanta. Ak je táto podmienka splnená, dochádza v konkrétnych uzlových bodoch \mathbf{r}_i^m a \mathbf{r}_j^m k deleniu krivky.

Rovnice pre šírenie sa potom riešia pre každú krivku osobitne. Táto myšlienka je rozobraná v [6].

Nasledujúce obrázky demonštrujú šírenie krivky pre prípad bez algoritmu na delenie kriviek a s ním.



Obr. 6: Šírenie požiarov bez algoritmu pre delenie krivky.



Obr. 7: Šírenie požiarov s použitím algoritmu pre delenie krivky.

Algoritmus pre delenie kriviek

```

for(g=0;g<fn;g++)
{
  do{
    // Hľadáme najmenšiu vzájomnú vzdialenosť uzlových bodov (m2)
    h=1000.;
    for(i=1;i<=gpn[g]-3;i++)
    {
      for(j=i+2;j<=gpn[g]-1;j++)
      {
        hh=sqrt((c_x[g][i]-c_x[g][j])*(c_x[g][i]-c_x[g][j])+
                (c_y[g][i]-c_y[g][j])*(c_y[g][i]-c_y[g][j]));
        if(hh<h) {pos1=i;pos2=j;h=hh;}
      }
    }

    for(j=2;j<=gpn[g]-2;j++)
    {
      hh=sqrt((c_x[g][gpn[g]]-c_x[g][j])*(c_x[g][gpn[g]]-c_x[g][j])+
              (c_y[g][gpn[g]]-c_y[g][j])*(c_y[g][gpn[g]]-c_y[g][j]));
      if(hh<h) {pos2=gpn[g];pos1=j;h=hh;}
    }

    // Podmienka delenia
    if(0.8*h<c_hmin[g])
    {
      if((gpn[g]-fabs(pos2-pos1))>6)
      {
        if(fabs(pos2-pos1)>6)
        {
          pos=1;
          for(i=pos1+1;i<pos2;i++)

```

```

        {
            c_x[fn][pos]=c_x[g][i];
            c_y[fn][pos]=c_y[g][i];
            pos++;
        }
        gpn[fn]=pos-1;
        fn++;
    }
    pos=pos1;
    for(i=pos2+1;i<=gpn[g];i++)
    {
        c_x[g][pos]=c_x[g][i];
        c_y[g][pos]=c_y[g][i];
        pos++;
    }
    gpn[g]=pos-1;
}

else{
    pos=1;
    for(i=pos1+1;i<pos2;i++)
    {
        c_x[g][pos]=c_x[g][i];
        c_y[g][pos]=c_y[g][i];
        pos++;
    }
    gpn[g]=pos-1;
}
}
}while(0.8*h<c_hmin[g]);
}

```

Popis algoritmu delenia

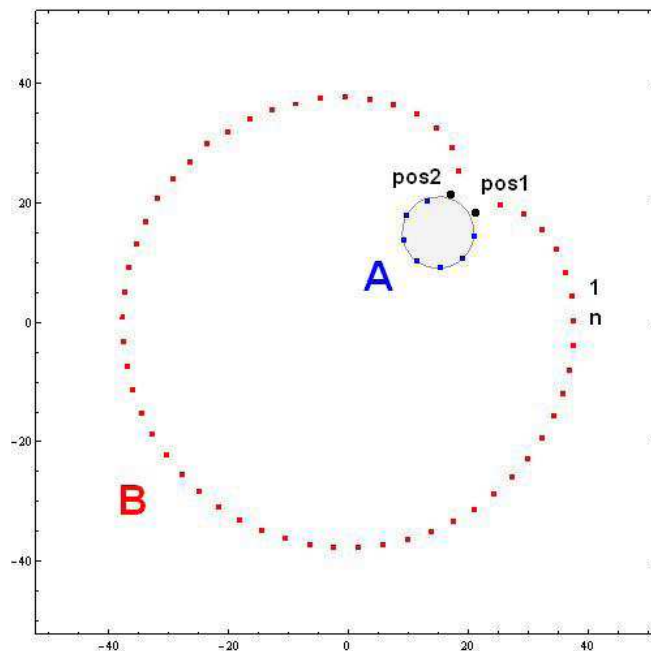
V každom časovom okamihu zisťujeme, či má nastať delenie pre každú krivku osobitne. V našom programe reprezentujeme polohový vektor $\mathbf{r}_i = [x, y]$ vyvíjajúcej sa krivky Γ pomocou dvoch dvojrozmerných polí pre x -ovú a y -ovú zložku - $c_x[g][i]$, $c_y[g][i]$. Prvý index g označuje konkrétnu krivku (dôležité pri vyvíjaní sa viacerých kriviek na pozorovanej oblasti), druhý index označuje jednotlivé uzlové body. Počet uzlových bodov n sa uchováva pre každú krivku zvlášť v jednorozmernom poli $gpn[g]$. Minimálnu vzdialenosť dvoch susedných uzlov $m1 = \min [h_i^m]$, $i = 1, \dots, n$ udáva pole $c_hmin[g]$. Tieto hodnoty sa počítajú v inej časti nášho programu, preto sa zameráme na krok 2. a 3. pre delenie krivky.

Aby sme ušetrili počet výpočtov, hľadáme vzájomnú vzdialenosť $m2 = |\mathbf{r}_i - \mathbf{r}_j|$ tak, že indexy i a j iterujeme v rozmedzí $i = 1, \dots, gpn[g] - 3$ a $j = i + 2, \dots, gpn[g] - 1$. Osobitne počítame túto vzdialenosť pre n -tý uzlový bod, kde $i = gpn[g]$ a $j = 2, \dots, gpn[g] - 2$. V programe túto hodnotu zapisujeme do premennej hh a následne ich minimum do premennej h . Pozície uzlových bodov najmenšej vzdialenosti $|\mathbf{r}_i - \mathbf{r}_j|$ zaznamenáme do premenných $pos1=i$, $pos2=j$.

V 3. kroku testujeme kritickú hodnotu najmenšej vzdialenosti podmienkou ($0.8 \cdot h < c_hmin[g]$). Pokiaľ je splnená dochádza v bodoch r_i a r_j k deleniu krivky, ktorých pozície sú uložené v premenných $pos1$, $pos2$. Vo väčšine prípadov sa používaním tohto algoritmu oddeľuje len veľmi malá časť, z toho dôvodu nie je potrebné zakaždým zachovať obe časti krivky. Na tento účel si označíme tú časť krivky, ktorej body r_i sa nachádzajú v intervale $(pos1, pos2)$, ako úsek A, zvyšnú časť ako úsek B (Obr. 8). Podľa toho, ktorý úsek sa zachová, rozlišujeme tri situácie:

1. Zostanú oba úseky A a B, pričom každý z nich bude v novom časovom kroku tvoriť osobitne sa vyvíjajúcu krivku. V programe najprv testujeme či je oblasť B dostatočne veľká - `if((gpn[g]-fabs(pos2-pos1))>6)`, následne testujeme oblasť A - `if(fabs(pos2-pos1)>6)`. Ak A vyhovuje podmienke, vytvoríme novú krivku do prázdnych polí `c_x[fn][]`, `c_y[fn][]`. Premenná `fn` udáva aktuálny počet vyvíjajúcich sa kriviek (netreba zabudnúť, že v jazyku C sa polia iterujú od nuly). Nové polia vytvárame tak, že ich plníme hodnotami z polí `c_x[g][i]`, `c_y[g][i]`, pričom index $i = pos1 + 1, \dots, pos2 - 1$. Uzlové body v nových poliach indexujeme pomocou iterovanej premennej `pos`. Nakoniec zadáme do `gpn[fn]` celkový počet bodov novej krivky, ktorú teraz tvorí pôvodná oblasť A a počet kriviek `fn` sa zvýši o 1. Či už sme časť A zanechali, alebo nie, algoritmus pre zachovanie oblasti B je v oboch situáciách 1. a 2. rovnaký, preto je súčasťou jednej podmienky a budeme sa mu venovať v nasledovnom kroku.
2. Pri zachovaní oblasti B už nevytvárame nové pole. Hodnoty r_i v poliach `c_x[g][i]`, `c_y[g][i]` v rozmedzí $i = 1, \dots, pos1 - 1$ zostávajú rovnaké. Preto sa zameráme len na úsek $pos2 + 1, \dots, gpn[g]$, ktorý napojíme na zostávajúci úsek oblasti B.
3. Ak nenastane ani jedna z prvých dvoch situácií, automaticky chápeme, že zostáva len oblasť A. Algoritmus v tomto prípade je takmer totožný ako v prípade 1. s tým rozdielom, že nevytvárame nové pole, ale prepisujeme pôvodné, podobne ako v prípade 2. Počet kriviek `fn` zostáva rovnaký.

Keďže v danom časovom okamihu sa môže šíriť niekoľko kriviek naraz, už spomínaný index `g`, reprezentujúci konkrétnu krivku, iterujeme v rozmedzí $0, \dots, fn - 1$. Algoritmus delenia krivky potom pre každú krivku zvlášť aplikujeme dovtedy, kým hodnota `m2` nebude kritická.



Obr. 8: Rozdelenie krivky na úseky A a B.

Všetky doteraz popísané odvodenia pre šírenie krivky v Lagrangeovskom ponímaní s vplyvom paliva a vetra, či samotné delenie krivky, už sú momentálne naprogramované a funkčné aj pre zložitejšie podmienky podobné reálnym situáciám. Spájanie kriviek je teda nosnou témou tejto práce a novým prínosom pre veľmi sľubne sa vyvíjajúci projekt.

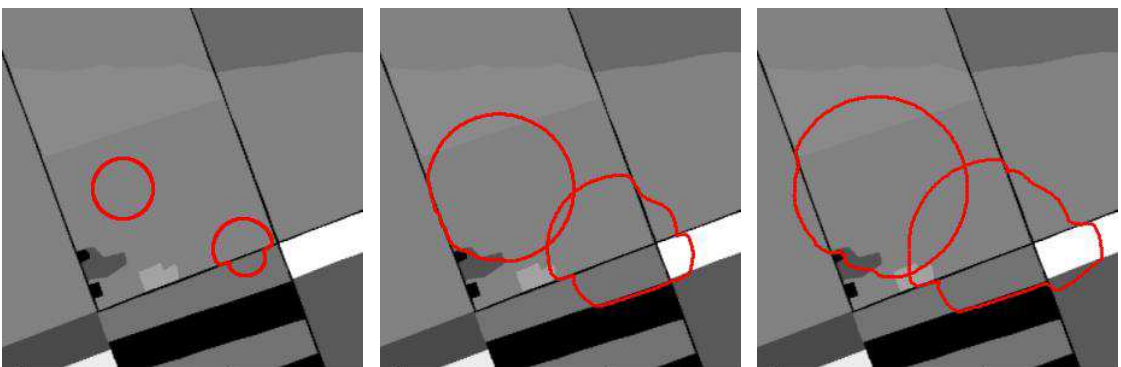
Spájanie viacerých kriviek

Pri šírení viacerých požiarov na pozorovanej oblasti dochádza k ich stretom a následnému zlučovaniu sa do jedného požiaru. Rovnako ako pri delení je nutné doprogramovať spájanie sa kriviek pre náš algoritmus.

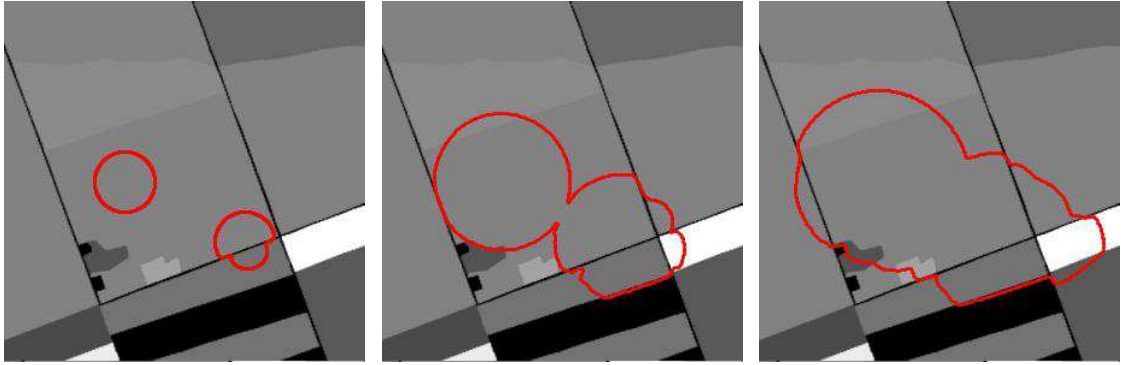
Pre spájanie viacerých kriviek reprezentujúcich lesné požiare je podstatné aby sme vedeli určiť aká je ich vzájomná vzdialenosť. Označme si tieto krivky ako Γ^g a Γ^l dané polohovými vektormi $\mathbf{r}^g, \mathbf{r}^l$ riešime v každom časovom kroku nezávisle dve sústavy rovníc. Keďže časový krok pri spájaní nie je podstatný, budeme v tejto časti využívať horný index pre špecifikáciu krivky.

Akonáhle sa krivky priblížia k sebe dostatočne blízko, dva požiare splynú do jedného, čo algoritmicke ošetríme tak, že dve krivky spojíme nasledujúcim postupom.

1. Vypočítame najmenšiu vzdialenosť susedných bodov pre obe krivky. $m1 = \min(\min(h_i^g), \min(h_j^l))$, $i = 1, \dots, n_g, j = 1, \dots, n_l$, kde n_g a n_l je počet uzlových bodov jednotlivej krivky Γ^g a Γ^l .
2. Hľadáme ich najmenšiu vzájomnú vzdialenosť kriviek Γ^g a Γ^l označenú $m2 = \min|\mathbf{r}_i^g - \mathbf{r}_j^l|$.
3. Následne testujeme či $m1 > C m2$. Ak je podmienka splnená, v bodoch $\mathbf{r}_i^g, \mathbf{r}_j^l$, pre ktoré máme vypočítanú hodnotu $m2$, spojíme krivky Γ^g a Γ^l do jednej, pre ktorú riešime samostatnú sústavu rovníc. Počet uzlových bodov novej krivky je $n = n_g + n_l - 2$, pretože body s minimálnou vzdialenosťou z výpočtu vynecháme.



Obr. 9: Šírenie požiarov bez algoritmu pre spájanie kriviek.



Obr. 10: Šírenie požiarov s použitím algoritmu pre spájanie kriviek.

Algoritmus pre spájanie kriviek

```

for(g=0;g<fn-1;g++)
{
    for(l=g+1;l<fn;l++)
    {
        // hľadáme najmenšiu vzájomnú vzdialenosť kriviek m2 (h)
        h=1000;
        for(i=1;i<=gpn[g];i++)
        {
            for(j=1;j<=gpn[l];j++)
            {
                hh=sqrt((c_x[g][i]-c_x[l][j])*(c_x[g][i]-c_x[l][j]) +
                    (c_y[g][i]-c_y[l][j])*(c_y[g][i]-c_y[l][j]));
                if(h<hh) {pos1=i; pos2=j; h=hh;}
            }
        }
        // zisťujeme hodnotu m1 (h_min)
        if(c_hmin[g]<c_hmin[l]) h_min=c_hmin[g];
        else h_min=c_hmin[l];

        if(0.8*h<h_min)
        {
            // vytváranie novej krivky
            pos=1;
            for(i=1;i<pos1;i++)
                {join_x[pos]=c_x[g][i]; join_y[pos]=c_y[g][i]; pos++;}
            for(i=pos2+1;i<=gpn[l];i++)
                {join_x[pos]=c_x[l][i]; join_y[pos]=c_y[l][i]; pos++;}
            for(i=1;i<pos2;i++)
                {join_x[pos]=c_x[l][i]; join_y[pos]=c_y[l][i]; pos++;}
        }
    }
}

```

```

for(i=pos1+1;i<=gpn[g];i++)
    {join_x[pos]=c_x[g][i]; join_y[pos]=c_y[g][i]; pos++;}

gpn[g]=pos-1;
for(i=0;i<=gpn[g];i++){c_x[g][i]=join_x[i]; c_y[g][i]=join_y[i];}

// posun polí
for(i=1;i<fn-1;i++)
    {for(j=0;j<=gpn[i+1];j++)
        {
            c_x[i][j]=c_x[i+1][j];
            c_y[i][j]=c_y[i+1][j];
        }
        gpn[i]=gpn[i+1];
    }

fn--; // počet kriviek sa zmenší o 1
}
}

```

Popis algoritmu spájania

V danom okamihu testujeme vzdialenosť práve dvoch konkrétnych kriviek Γ^g a Γ^l dané polohovými vektormi $\mathbf{r}^g, \mathbf{r}^l$. V predošlých častiach sme tieto krivky reprezentovali pomocou polohového vektora $\mathbf{r}_i = [x_i, y_i]$, pre $i = 1, \dots, n$, kde index i označuje konkrétny uzlový bod krivky. Pre náš účel používame v algoritme dvojrozmerné polia $c_x[k][i]$, $c_y[k][i]$ reprezentujúce zložky polohového vektora \mathbf{r}_i^k , kde horný index k označuje ľubovoľnú krivku Γ^k .

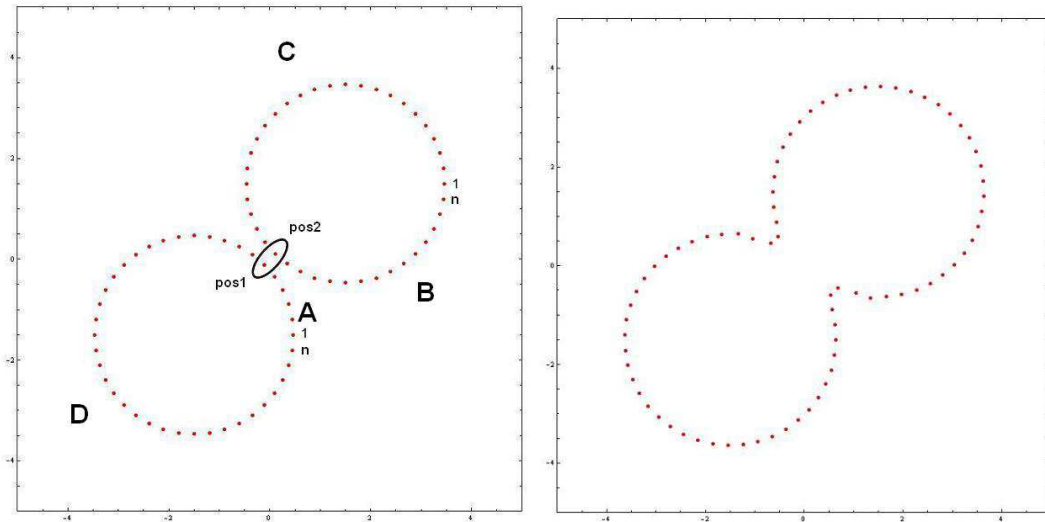
Platí teda :

$k = 0, \dots, fn - 1$ kde fn je celkový počet kriviek,

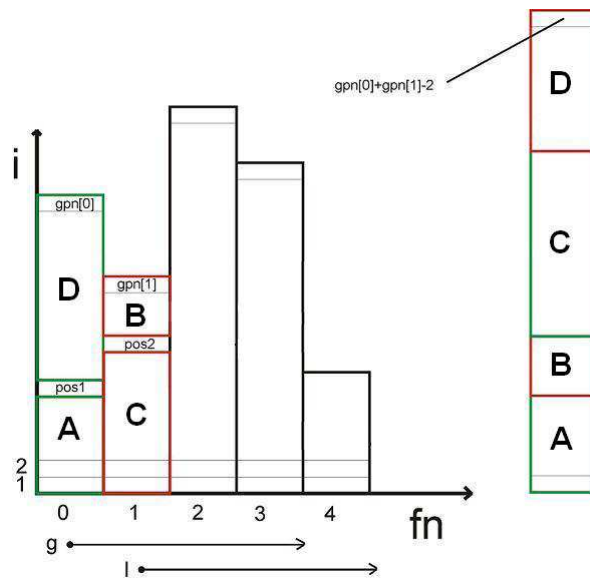
$i = 0, \dots, gpn[k] + 1$, kde $gpn[k]$ je celkový počet uzlových bodov krivky Γ^k . Pozície $i = 0$ a $i = gpn[k] + 1$ sú potrebné kôli cyklickým okrajovým podmienkam.

Aby sme zabezpečili, že sa nebude testovať jedna krivka so samou sebou, alebo aby sme nekontrolovali jednotlivú dvojicu Γ^g a Γ^l viac krát, využijeme pre indexy g a l nasledujúce vzťahy: $g = 0, \dots, fn - 2$ a $l = g + 1, \dots, fn - 1$. Následne počítame najmenšiu vzdialenosť h (m2) kriviek Γ^g a Γ^l , pre ktorú si označíme pozície uzlových bodov $pos1 = i$ a $pos2 = j$. Taktiež nájdeme hodnotu h (m1) ako minimum z hodnôt v premenných $c_hmin[g]$ a $c_hmin[l]$. Keď už máme tieto hodnoty vypočítame, môžeme testovať vzdialenosť kriviek pomocou podmienky $\text{if}(0.8*h < h_min)$. Ak je splnená, nastáva samotný proces spájania. Jednotlivé oblasti A, B, C, D (Obr. 11) oboch kriviek spojíme do jednej. Na tento účel používame pomocné polia $join_x[]$, $join_y[]$, ktoré plníme spôsobom dostatočne vysvetľujúcim pomocou Obr. 11 a Obr. 12.

Takto novovytvorenú krivku Γ potom zapíšeme do polí pre spájanú krivku Γ^g . Druhá krivka Γ^l zanikne, pričom na mieste l v dvojrozmerných poliach $c_x[l][]$, $c_y[l][]$ vznikne “prázdne miesto”. Preto je potrebné krivky Γ_k , pre ktoré platí, že $l < k < fn$, poposúvať smerom “doľava”, aby nedošlo k zlyhaniu programu.



Obr. 11: Rozdelenie dvoch kriviek na jednotlivé oblasti, následne spojené v jednu krivku.

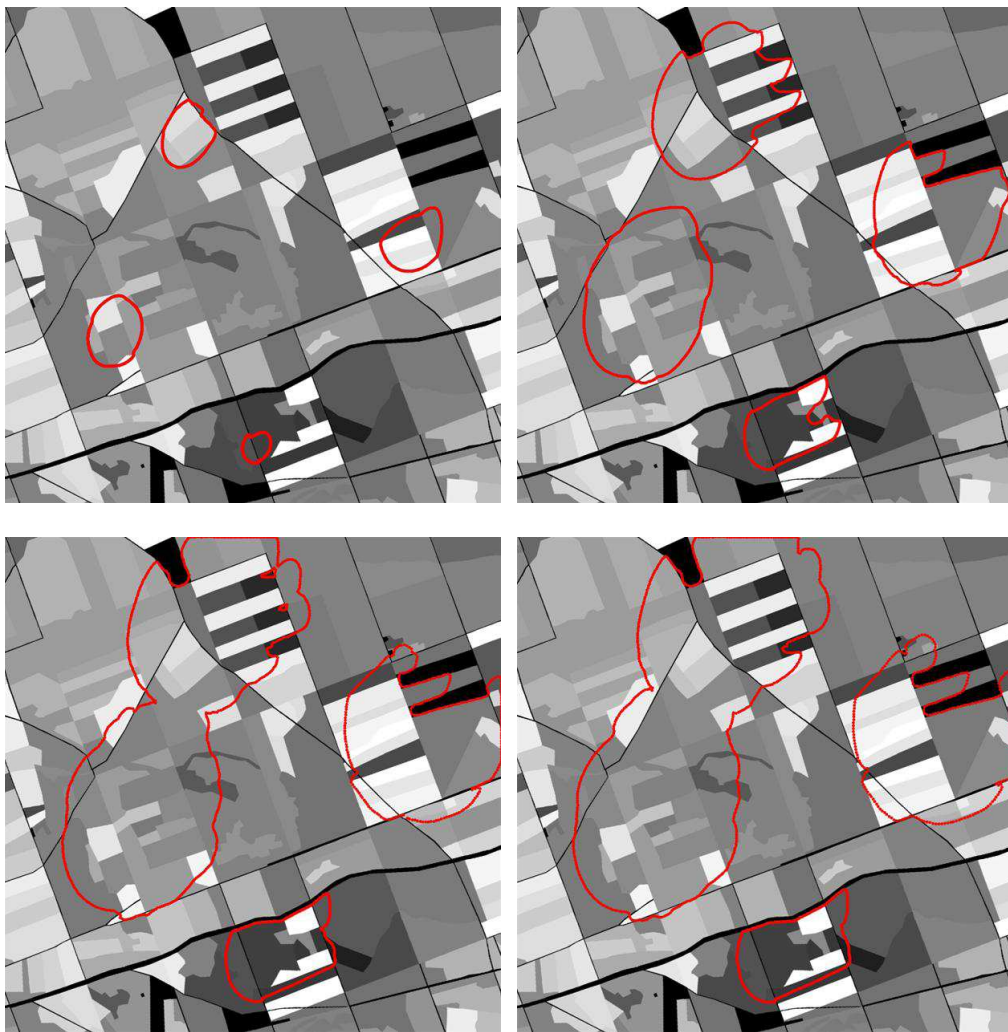


Obr. 12: Postupné spájanie častí polí dvoch kriviek do jedného.

Zhodnotenie dosiahnutých výsledkov

1. Vytvorili sme nový efektívny algoritmus pre zahrnutie topologických zmien do výpočtu pohybovej rovnice pre šírenie požiaru, ktorý sa týka spájania viacerých požiarov šíriacich sa z rôznych miest.
2. Naprogramovali sme vlastný program na riešenie pohybovej rovnice pre šírenie požiaru, ktorý obsahuje algoritmus pre topologické zmeny týkajúce sa delenia kriviek a algoritmus pre šírenie požiaru z bodu 1.

Výsledky prezentujeme na nasledujúcich obrázkoch. Vizualizujeme šírenie viacerých požiarov v nehomogénom prostredí s vplyvom vetra. Na spodných obrázkoch je vidieť funkčné spájanie a delenie kriviek.



Obr. 13: Šírenie viacerých lesných požiarov v nehomogénom prostredí.

Záver

Práca sa zaoberá problémom topologických zmien pri spájaní viacerých požiarov, ktoré sa šíria z viacerých miest. Napísali sme program na riešenie pohybovej rovnice pre šírenie požiaru. Vytvorili sme nový, vlastný algoritmus pre spomínané topologické zmeny, ktorý sme zahrnuli do programu.

Literatúra

- [1] M. Balazovjeh, K. Mikula, *A higher order scheme for the curve shortening flow of plane curves*, In: Proceedings of ALGORITMY(2009), pp. 165-175.
- [2] M. Balazovjeh, K. Mikula, *A higher order scheme for tangentially stabilized curve shortening flow in a plane with a driving force*, submitted to SIAM J. Appl. Math.
- [3] K. Mikula, D. Sevcovic, *Solution of nonlinearly curvature driven evolution of plane curves*, Applied Numerical Mathematics, 31 (1999), pp. 191-207.
- [4] K. Mikula, D. Sevcovic, *Evolution of plane curves driven by a nonlinear function of curvature and anisotropy*, SIAM J. Appl. Math., 61 (2001), pp. 1473-1501.
- [5] K. Mikula, D. Sevcovic, M. Balazovjeh, *A simple, fast and stabilized flowing finite volume method for solving general curve evolution equations*, Communications in Computational Physics, Vol. 7, No. 1 (2010), pp. 195-211.
- [6] P. Paus M. Benes, *Algorithm for topological changes of parametrically described curves*, Proceedings of ALGORITMY(2009), pp. 176-184.
- [7] Richards, GD. 1995. *A general mathematical framework for modeling two-dimensional wildland fire spread*, Int. J. Wildl. Fire. 5(2): pp 63-72.
- [8] J. Urbán, *Modelovanie lesných požiarov pomocou evolúcie rovinných kriviek*, ŠVK 2011 STU Bratislava , s 8-10.