

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**  
**STAVEBNÁ FAKULTA**

**NUMERICKÉ MODELOVANIE LAVÍNY**

Diplomová práca

**BRATISLAVA 2020**

**Bc. Michal Žeravý**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**

**Stavebná fakulta**

Evidenčné číslo: SvF-5343-75449

# **NUMERICKÉ MODELOVANIE LAVÍNY**

**Diplomová práca**

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

**BRATISLAVA 2020**

**Bc. Michal Žeravý**



## ZADANIE DIPLOMOVEJ PRÁCE

Študent: **Bc. Michal Žeravý**  
ID študenta: 75449  
Študijný program: matematicko-počítačové modelovanie  
Študijný odbor: matematika  
Vedúci práce: doc. RNDr. Peter Frolkovič, PhD.

Názov práce: **Numerické modelovanie lavíny**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

Práca sa bude zaoberať numerickým modelovaním pohybu lavín na základe rovníc pre plytkú vodu.

Riešenie zadania práce od: 30. 09. 2019

Dátum odovzdania práce: 14. 05. 2020

**Bc. Michal Žeravý**  
študent

**prof. RNDr. Radko Mesiar, DrSc.**  
vedúci pracoviska

**prof. RNDr. Karol Mikula, DrSc.**  
garant študijného programu

## **POKYNY na vypracovanie diplomovej práce**

### **Úvodné ustanovenie**

V zmysle zákona č. 131/2002 Z. z. o vysokých školách a o zmene a doplnení niektorých zákonov v znení neskorších predpisov je súčasťou štúdia podľa každého študijného programu aj záverečná práca. Jej obhajoba patrí medzi štátne skúšky. Záverečnou prácou pri štúdiu podľa študijného programu druhého stupňa je diplomová práca. Podkladom na vypracovanie diplomovej práce je zadanie diplomovej práce

### **Štruktúra záverečnej práce**

- titulný list,
- zadanie záverečnej práce,
- pokyny na vypracovanie,
- vyhlásenie autora,
- názov a abstrakt v slovenskom a v anglickom jazyku (spolu v rozsahu jednej strany),
- obsah s očíslovaním kapitol,
- zoznam príloh,
- zoznam skratiek a značiek,
- text samotnej práce (odporúčané členenie),
  - úvod,
  - súčasný stav problematiky,
  - ciele záverečnej práce,
  - vlastné riešenie členené na kapitoly podľa charakteru práce,
  - zhodnotenie dosiahnutých výsledkov resp. navrhnutých riešení,
  - záver,
- resumé v slovenskom jazyku v rozsahu spravidla 10 % rozsahu ZP (len pre práce vypracované v cudzom jazyku),
- zoznam použitej literatúry,
- prílohy (výkresy, tabuľky, mapy, náčrty) vrátane postera s rozmermi 1000x700 mm.

### **Rozsah a forma**

1. Obsah a forma záverečnej práce musí byť spracovaná v zmysle vyhlášky MŠVVaŠ SR č. 233/2011 Z. z., ktorou sa vykonávajú niektoré ustanovenia zákona č. 131/2002 Z. z. a v zmysle Metodického usmernenia č. 56/2011 o náležitostiach záverečných prác.
2. Vyžadovaný rozsah diplomovej práce je 30 až 50 strán. Odovzdáva sa v dvoch vyhotoveniach. Jedno vyhotovenie musí byť viazané v pevnej väzbe (nie hrebeňovej) tak, aby sa jednotlivé listy nedali vyberať. Rozsiahle grafické prílohy možno v prípade súhlasu vedúceho práce odovzdať v jednom vyhotovení.

3. Autor práce je povinný vložiť prácu v elektronickej forme do akademického informačného systému. Autor zodpovedá za zhodu listinného aj elektronického vyhotovenia.
4. Po vložení záverečnej práce do informačného systému, predloží autor fakulte ním podpísaný návrh licenčnej zmluvy. Návrh licenčnej zmluvy je vytvorený akademickým informačným systémom.
5. Odporúčaný typ písma je Times New Roman, veľkosť 12 a je jednotný v celej práci. Odporúčané nastavenie strany - riadkovanie 1,5, okraj vnútorný 3,5 cm, vonkajší 2 cm, zhora a zdola 2,5 cm, orientácia na výšku, formát A4.
6. Obrázky a vzorce sa čísloujú v rámci jednotlivých kapitol (napr. obr. 3.1 je obrázok č. 1 v kapitole 3). Vzorce sa čísloujú na pravom okraji riadku v okrúhlych zátvorkách - napr. (3.1).
7. Všetky výpočty musia byť usporiadané tak, aby bolo možné preveriť ich správnosť.
8. Pri všetkých prevzatých vzorcoch, tabuľkách, citovaných častiach textu musí byť uvedený prameň.
9. Citovanie literatúry vrátane elektronických materiálov sa uvádza podľa STN ISO 690 (01 0197): 2012. *Informácie a dokumentácia. Návod na tvorbu bibliografických odkazov na informačné pramene a ich citovanie.*
10. Príklad zoznamu bibliografických odkazov:  
 ABELOVIČ, J. a kol.: *Meranie v geodetických sieťach*. Bratislava, Alfa 1990, ISBN 0-1554-9173.  
 MICHALČÁK, O. – ADLER, E.: Výskum stability dunajských hrádzí. In: *Zborník vedeckých prác Stavebnej fakulty SVŠT*. Bratislava: Edičné stredisko SVŠT 1976, s. 17-28. ISBN 0-3552-5214.  
 ŠÜTTI, J.: Určovanie priestorových posunov stavebných objektov. *Geodetický kartografický obzor*. 2000, roč. 2, č. 3, s. 8-16. ISSN 0811-6900.  
 Article 18. Technical Cooperation. <http://www.lac.uk/iso/tc456> (2013-09-28)
11. Za jazykovú a terminologickú správnosť záverečnej práce zodpovedá diplomant.
12. Formu postera (elektronická alebo aj tlačaná) určí garant študijného programu.
13. Vzor pre poster je uvedený na dokumentovom serveri v akademickom informačnom systéme univerzity.

.....  
 podpis garanta študijného programu

Ustanovenia týchto pokynov som vzal na vedomie. Som si vedomý(á), že ak nebude moja diplomová práca vypracovaná v súlade s týmito pokynmi, nebude prijatá na obhajobu.

V Bratislave .....

.....  
 podpis študenta

### Čestné vyhlásenie

Čestne vyhlasujem, že diplomovú prácu s názvom: „Numerické modelovanie lavíny“ som vypracoval samostatne, na základe konzultácií a s použitím uvedených informčných zdrojov a literatúry.

V Bratislave, 14.5.2020

.....

Podpis autora práce

## **Pod'akovanie**

Ďakujem vedúcemu práce doc. RNDr. Petrovi Frolkovičovi, PhD. za ochotu venovať sa problematike modelovania lavín a s ňou spojené odborné konzultácie, ktoré boli prínosom pri písaní diplomovej práce. Ďakujem Ing. Michalovi Kollárovi, PhD. za pomoc a rady pri spracovaní dát. Ďakujem Stredisku lavínovej prevencie Horskej záchranej služby za poskytnuté dáta reálneho terénu.

# Abstrakt

V tejto práci sme sa oboznámili so softvérom OpenFoam, ktorý obsahuje modul pre modelovanie lavín. Popisujeme funkcionality softvéru a jeho základné časti. Na jednoduchom vzorovom príklade si osvojíme prácu so softvérom a oboznamujeme sa s vizualizáciou výsledkov v softvéri ParaView. Naším hlavným cieľom je na dátach, ktoré predstavujú reálny terén, modelovať lavínu podobnú lavíne zaznamenatej v danej oblasti na video. Za tým účelom spracúvame dáta, získané zo Strediska lavínovej prevencie Horskej záchranej služby, do požadovaného formátu. Popisujeme a vysvetľujeme jednotlivé nastavenia simulácie priebehu lavíny a vizualizujeme ju v softvéri ParaView. Zobrazujeme veličiny spojené s lavínou a popisujeme ich vývoj počas simulácie. Simuláciu lavíny porovnávame s reálnou situáciou. V závere vizualizujeme trajektórie častíc lavíny na základe rýchlostného poľa.

**Kľúčové slová:** rovnice plytkej vody, OpenFoam, ParaView, konečné plochy, lavína

# Abstract

In this thesis, we are getting familiar with OpenFoam software, which contains avalanche modelling module. We describe the functionality of the software and its basic parts. With the simple example from tutorials we learn how to work with the software and we are getting familiar with the visualization of the results in the ParaView software. Our main goal is to model avalanche similar to an avalanche recorded on video. Data for avalanche represents the real topography. We process data for this purpose to desired format. We obtain a data from the Avalanche prevention center of the Mountain rescue service. We describe and explain individually the options of the avalanche simulation and we visualized it in ParaView. We show the variables which are associated with the avalanche and we describe the evaluation of them during the simulation. The simulation of avalanche is compared with a real situation. In the end of this thesis we visualize the trajectories of the avalanche particles based on the velocity field.

**Key words:** shallow water equations, OpenFoam, ParaView, finite areas, avalanche



# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Matematický model</b>	<b>4</b>
<b>3</b>	<b>Diskretizácia metódou konečných plôch</b>	<b>6</b>
3.1	Priestorová diskretizácia . . . . .	7
3.2	Okrajové podmienky . . . . .	8
3.3	Časová diskretizácia . . . . .	8
3.4	Počiatočná podmienka . . . . .	8
3.5	Priebeh výpočtu . . . . .	9
<b>4</b>	<b>OpenFoam softvér</b>	<b>10</b>
4.1	Štruktúra súborov . . . . .	10
4.2	Rozmerové jednotky . . . . .	12
4.3	Reprezentácia geometrie . . . . .	12
<b>5</b>	<b>Príklad s jednoduchým terénom</b>	<b>16</b>
<b>6</b>	<b>Modelovanie lavíny v oblasti Príslopu v Západných Tatrách</b>	<b>20</b>
6.1	Spracovanie dát . . . . .	20
6.2	Nastavenie a spustenie výpočtu . . . . .	23
6.3	Vizualizácia výsledkov pomocou softvéru ParaView . . . . .	24
6.4	Porovnanie výsledkov . . . . .	27
6.5	Trajektórie častíc lavíny . . . . .	30
<b>7</b>	<b>Záver</b>	<b>35</b>

# 1 Úvod

Modelovanie priebehu lavín má význam pre ohodnotenie lavínového rizika v horských oblastiach.

Už v bakalárskej práci sme sa oboznámili s problematikou simulovania lavín. V tejto oblasti sme sa rozhodli pokračovať aj v diplomovej práci. Daná problematika je taktiež veľmi rozšírená vo svete, čoho dôkazom môže byť aj nový modul v softvéri OpenFoam, s ktorým v tejto práci pracujeme a ktorý je jeden z najrozšírenejších „open source“ softvérov na numerické simulácie. Jeho relatívne nový modul s názvom *avalanche*, ako už z názvu vyplýva, umožňuje modelovať lavíny.

V prvotnej fáze sme sa zoznámili so softvérom pomocou jednoduchých vzorových príkladov a následne sme ich čiastočne modifikovali. Hlavným cieľom diplomovej práce je vytvorenie simulácie lavíny na reálnom teréne, ktorá bude zodpovedať lavíne v danej oblasti zaznamenatej na video. Tieto dáta sme získali zo Strediska lavínovej prevencie Horskej záchranej služby na akademické účely. Naša simulácia nemá len napodobniť reálnu lavínu, ale môže mať aj veľkú výpovednú hodnotu. To znamená, že pomocou simulácie získame informácie o rýchlosti lavíny, maximálnej výške snehu a mnoho ďalších dôležitých informácií.

Vizualizácia výsledkov prebieha v softvéri ParaView, kde sme za pomoci rôznych filtrov schopný interpretovať rôzne vlastnosti lavín. V tomto softvéri sa pokúsime aj vykresliť trajektórie vybraných častíc lavíny.

## 2 Matematický model

Numerické modelovanie snehových lavín modelujeme ako prúdenie plytkej kvapaliny po zakrivenej ploche  $S$ . Kvapalina je reprezentovaná ako zrnitý materiál pozostávajúci z drobných pevných častíc. Pri odvodení matematického modelu [3] pre prúdenie plytkej vody vychádzame zo zákona zachovania hmotnosti

$$\frac{\partial}{\partial t} \int_S h dS + \int_L \mathbf{n} \cdot h \bar{\mathbf{u}} dL = 0, \quad (1)$$

kde  $h$  je výška hladiny kvapaliny,  $\mathbf{n}$  je vonkajšia normála k ploche  $S$ ,  $\bar{\mathbf{u}}$  je cez hĺbku priemerná rýchlosť a  $S$  je plocha s hranicou  $\partial S = L$ . Ďalej vychádzame zo zákona zachovania hybnosti

$$\begin{aligned} \frac{\partial}{\partial t} \int_S h \bar{\mathbf{u}} dS + \int_L \mathbf{n} \cdot \xi h \bar{\mathbf{u}} \bar{\mathbf{u}} dL = & -\frac{1}{\rho} \int_S \boldsymbol{\tau} dS + \int_S h \mathbf{g} dS \\ & -\frac{1}{2\rho} \int_L \mathbf{n} h p dL - \frac{1}{\rho} \int_S \mathbf{n} p dS, \end{aligned} \quad (2)$$

kde  $\xi$  je konštantny faktor tvaru terénu,  $\rho$  je konštantná hustota kvapaliny,  $\boldsymbol{\tau}$  je trenie medzi kvapalinou a terénom,  $\mathbf{g}$  je gravitačné zrýchlenie a  $p$  je tlak. Tretia rovnica, z ktorej vychádzame, je okrajová podmienka

$$\bar{\mathbf{u}} \cdot \mathbf{n} = 0. \quad (3)$$

Trenie medzi kvapalinou a terénom popisuje vzťah:

$$\boldsymbol{\tau} = \mu(I)p \frac{\bar{\mathbf{u}}}{|\bar{\mathbf{u}}| + \mu_0}. \quad (4)$$

Koeficient trenia  $\mu$  je vyjadrený ako funkcia bezrozmerného inerciálneho čísla  $I$ :

$$\mu = \mu_s + \frac{\mu_2 - \mu_s}{I_0/I + 1}, \quad (5)$$

$$I = \frac{5}{2} \frac{d|\bar{\mathbf{u}}|/h}{\sqrt{p/\rho_p}}, \quad (6)$$

s materiálovými parametrami  $d$  priemer častíc,  $\rho_p$  hustota častíc a parametrami  $\mu_s$ ,  $\mu_2$  a  $I_0$  opísanými v [3]. Na odvodenie modelu prúdenia je potrebné [3] skombinovať zákon zachovania hybnosti (2) s okrajovou podmienkou (3) a rozdeliť rovnicu na časť rovnobežnú s terénom a časť kolmú na terén. Rovnobežnú časť rovnice budeme riešiť pre rýchlosť a kolmú časť rovnice budeme riešiť pre tlak. V klasickom odvodení sa toto rozdelenie prirodzene vyskytuje v dôsledku zakriveného súradnicového systému. My sa

vyhneme tejto transformácii súradníc a namiesto toho vykonáme projekciu. Keďže túto projekciu nie je možné použiť na rovnice v integrálnej forme, musíme rovnice prepísať do diferenciálnej podoby.

Rovnica zákona zachovania hmotnosti (1) môže byť prepísaná do tvaru:

$$\int_S \frac{\partial h}{\partial t} dS + \int_S \nabla \cdot (h \bar{\mathbf{u}}) dS = 0, \quad (7)$$

použitím Gaussovej vety. Pretože rovnica platí pre každú oblasť  $S$ , integrand musí byť rovný nule,

$$\frac{\partial h}{\partial t} + \nabla \cdot (h \bar{\mathbf{u}}) = 0. \quad (8)$$

Rovnakým postupom môžeme prepísať rovnicu zákona zachovania hybnosti (2):

$$\frac{\partial(h\bar{\mathbf{u}})}{\partial t} + \xi \nabla \cdot (h \bar{\mathbf{u}} \bar{\mathbf{u}}) = -\frac{1}{\rho} \boldsymbol{\tau} + h \mathbf{g} - \frac{1}{2\rho} \nabla(hp) - \frac{1}{\rho} \mathbf{n} p. \quad (9)$$

Rovnica kombinuje klasickú rovnicu zákona zachovania hybnosti a rovnicu pre tlak. Pre získanie riešenia pre tlak môžeme použiť projekciu kolmú na terén pre rovnicu (9), ktorú realizujeme vynásobením tejto rovnice normalovým vektorom  $\mathbf{n}$ :

$$\underbrace{\mathbf{n} \cdot \frac{\partial h \bar{\mathbf{u}}}{\partial t}}_{=0} + \mathbf{n} \cdot \xi \nabla \cdot (h \bar{\mathbf{u}} \bar{\mathbf{u}}) = -\frac{1}{\rho} \underbrace{\mathbf{n} \cdot \boldsymbol{\tau}}_{=0} + h \mathbf{n} \cdot \mathbf{g} - \frac{1}{2\rho} \mathbf{n} \cdot \nabla(hp) - \frac{1}{\rho} \underbrace{\mathbf{n} \cdot \mathbf{n}}_{=1} p. \quad (10)$$

Z okrajovej podmienky (3) dostaneme prvý člen rovný nule a taktiež člen s trením nám vypadne. Dodatočným násobením rovnice (10) s normálovým vektorom  $\mathbf{n}$  dostaneme rovnicu pre tlak vrátane jeho smeru  $\mathbf{n}p$  v tvare

$$\xi \nabla_n \cdot (h \bar{\mathbf{u}} \bar{\mathbf{u}}) = h \mathbf{g}_n - \frac{1}{2\rho} \nabla_n(hp) - \frac{1}{\rho} \mathbf{n} p, \quad (11)$$

s gradientom kolmým na terén v tvare

$$\nabla_n = (\mathbf{n} \mathbf{n}) \cdot \nabla, \quad (12)$$

a normálovou zložkou gravitačného zrýchlenia

$$\mathbf{g}_n = (\mathbf{n} \mathbf{n}) \cdot \mathbf{g}. \quad (13)$$

Prvý člen v rovnici (11) predstavuje odstredivé sily a druhý člen v rovnici predstavuje gravitačné sily. Tretí člen, ktorý sa neuvažuje v klasickom prístupe, berie do úvahy, že

gradient tlaku na steny oblasti obsahuje zložky kolmé na terén v dôsledku zakrivenia terénu. Tento člen je malý, ale je potrebný na získanie rýchlosti v ďalšom kroku, ktorá je rovnobežná s terénom. Hrá však malú úlohu v tlaku a následne v trení. Rovnicu (11) odpočítame od rovnice (9), aby sme získali rovnicu pre hybnosť rovnobežnú s terénom v nasledujúcom tvare

$$\frac{\partial h\bar{\mathbf{u}}}{\partial t} + \xi \nabla_s \cdot (h\bar{\mathbf{u}}\bar{\mathbf{u}}) = -\frac{1}{\rho}\boldsymbol{\tau} + h\mathbf{g}_s - \frac{1}{2\rho}\nabla_s(h, p), \quad (14)$$

s gradientom rovnobežným s terénom v tvare

$$\nabla_s = (\mathbf{I} - \mathbf{nn}) \cdot \nabla. \quad (15)$$

Počiatočnú podmienku pre model lavíny definujeme pre výšku hladiny kvapaliny  $h$  a rýchlosť prúdenia  $\bar{\mathbf{u}}$  v počiatočnom čase  $t_0$ . Počiatočnú výšku hladiny kvapaliny môžeme reprezentovať ako výšku snehu v mieste spustenia lavíny a označíme si ju ako  $h_0$  a počiatočnú rýchlosť si označíme ako  $u_0$ , potom počiatočnú podmienku máme definovanú v tvare

$$\begin{aligned} h(t_0) &= h_0 \\ \bar{\mathbf{u}}(t_0) &= u_0. \end{aligned} \quad (16)$$

Výsledný matematický model pozostáva z rovníc (8), (11) a (14) spolu s (4), (5) a (6) a počiatočnými podmienkami (16).

### 3 Diskretizácia metódou konečných plôch

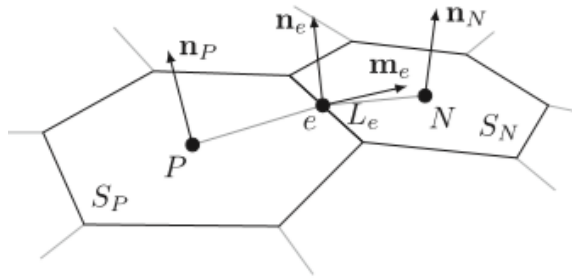
Pre numerické riešenie modelu pre prúdenie definovaného v Sekcii 2, musíme diskretizovať výpočtovú oblasť v priestore a čase. Časový interval simulácie je rozdelený do časových krokov a rovnice sú riešené postupne pre každý časový krok. Priestorová oblasť, v našom prípade zakrivený povrch v trojrozmernom priestore, je rozdelená do konečného počtu konvexných polygonálnych kontrolných plôch ohraničených ľubovoľným počtom hrán, vid' Obr. 1. Kontrolné plochy pokrývajú celú priestorovú oblasť.

Diskretizované rovnice sú riešené sekvenčne implicitnou numerickou schémou. Implicitné riešiče vyjadrujú väčšinu výrazov ako funkcie neznámych hodnôt veličín, t.j. hodnôt pre nový časový krok. Sekvenčný prístup znamená, že rieši v čase vždy len

jednu riadiacu rovnicu, každá rovnica dáva výsledok pre jednu veličinu. Tento prístup bol zvolený z dôvodu komplexného nelineárneho prepojenia rovníc pre rýchlosť a výšku hladiny kvapaliny.

### 3.1 Priestorová diskretizácia

Diskretizácia metódou konečných plôch je založená na integrálnej forme rovníc zákonov zachovania, ktoré dostaneme integráciou riadiacich rovníc cez kontrolnú plochu.



Obr. 1: Polygonálne kontrolné plochy [3]

Obrázok Obr. 1 reprezentuje dve polygonálne kontrolné plochy  $S_P$  a  $S_N$  s reprezentáčnymi bodmi  $P$  a  $N$ , ktoré sú umiestnené v príslušnom ťažisku. Hrany s dĺžkou  $L_e$  sú označené ako  $e$ . Normálové vektory kolmé na plochu sú označené ako  $\mathbf{n}_P$  a  $\mathbf{n}_N$  pre plochy a ako  $\mathbf{n}_e$  pre hrany. Normálové vektory  $\mathbf{n}_{i0}$  na bočných plochách  $S_{i0}$  teraz majú formu binormálnych vektorov na hrane  $e$  a normálovom vektore  $\mathbf{n}_e$ , sú označené ako  $\mathbf{m}_e$ . Použitím pravidla stredného bodu s presnosťou druhého rádu na aproximáciu integrálov cez povrch  $S$  pre bunku  $P$  dostaneme:

$$\int_S \Phi dS \approx \Phi_P |S_P|$$

kde  $\Phi_P$  je hodnota poľa  $\Phi$  v reprezentáčnom bode  $P$  a  $|S_P|$  je veľkosť plochy  $S_P$ .

Operátory gradientu a divergencie sú použitím Gaussovej vety prepísané ako hraničné integrály po hranici  $\partial S = L$  a diskretizované nasledovne:

$$\int_S \nabla \Phi dS = \int_L \mathbf{n} \Phi dL \approx \sum_e \mathbf{m}_e \Phi_e L_e, \quad (17)$$

$$\int_S \nabla \cdot \Phi dS = \int_L \mathbf{n} \cdot \Phi dL \approx \sum_e \mathbf{m}_e \cdot \Phi_e L_e, \quad (18)$$

kde  $\Phi_e$  reprezentuje hodnotu  $\Phi$  v strede hrany  $e$ . Hodnoty  $\Phi_e$  v stredoch hrán  $e$  v rovniciach (17) a (18), sú interpolované z hodnôt susedných buniek.

### 3.2 Okrajové podmienky

Ak hrana  $e$  je umiestnená na okraji  $L$  výpočtovej oblasti  $S$ , tak nemôžeme použiť diskretizáciu riadiacej diferenciálnej rovnice. Namiesto nej okrajová podmienka musí poskytnúť informácie potrebné na výpočet okrajových hodnôt  $\Phi_e$ .

Dirichletové okrajové podmienky predpisujú hodnotu na hranici, ako napríklad  $\Phi(\mathbf{x}) = \Phi_B, \forall \mathbf{x} \in L$ , teda dostávame priamo požadovanú hodnotu:

$$\Phi_e = \Phi_B \quad (19)$$

Von Neumannové okrajové podmienky predpisujú gradient v smere normály na hranici. V najjednoduchšom prípade  $\nabla\Phi(\mathbf{x}) \cdot \mathbf{n}_{io} = 0, \forall \mathbf{x} \in L$ , máme hodnotu na hranici:

$$\Phi_e = \Phi_P \quad (20)$$

Tieto jednoduché prípady sú dostatočné na zabezpečenie okrajových podmienok pre prítok a odtok pre daný model pre plytké prúdenie.

### 3.3 Časová diskretizácia

Diskretizácia časovej derivácie v čase  $t^n$  je vykonaná implicitnou schémou druhého rádu presnosti, nazývanou spätná schéma.

$$\left(\frac{\partial\Phi}{\partial t}\right)^n \approx \frac{3\Phi^n - 4\Phi^{n-1} + \Phi^{n-2}}{2\Delta t}. \quad (21)$$

Nový časový krok máme s indexom  $n$ .

Táto schéma vyžaduje prepočítanie diferenciálnej rovnice iba v jednom časovom kroku. Diferenciálna rovnica je aproximovaná v novom časovom kroku  $t^n$  a preto numerickú schému nazývame implicitná.

### 3.4 Počiatočná podmienka

Počiatočná podmienka musí poskytnúť hodnoty  $\Phi^{n-1}$  na celej priestorovej oblasti  $S$ , tak ako vyžaduje časová derivácia. Pre zabránenie potreby poskytnúť hodnoty  $\Phi^{n-2}$ ,

prvý časový krok sa vykoná pomocou Eulerovej implicitnej schémy prvého rádu:

$$\left(\frac{\partial\Phi}{\partial t}\right)^n \approx \frac{\Phi^n - \Phi^{n-1}}{\Delta t}. \quad (22)$$

Pre granulárne lavíny je výška hladiny kvapaliny  $h \neq 0$  zadaná v nejakej podoblasti celej výpočtovej oblasti  $S$ . Počiatočná rýchlosť je zvyčajne nulová rovnako ako výška hladiny kvapaliny mimo podoblasti počiatočnej podmienky.

### 3.5 Priebeh výpočtu

Matematický model tvorí systém prepojených nelineárnych diferenciálnych rovníc. Diskretizácia vedie k sústave algebraických rovníc, ktoré môžu byť riešené numericky. Iteračným prístupom je rovnica (11) najskôr riešená pre tlak  $p$  pomocou starého riešenia pre výšku hladiny kvapaliny  $h$  a rýchlosti  $\bar{\mathbf{u}}$ . V druhom kroku sa rovnica (14) rieši pre rýchlosť  $\bar{\mathbf{u}}$  s prepočítanými hodnotami pre tlak  $p$  a starými hodnotami pre výšku hladiny kvapaliny  $h$ . Nakoniec rovnica (8) je riešená pre výšku hladiny kvapaliny  $h$ . Algoritmus iteruje pokiaľ nedosiahneme požadované rezíduum.



## 4 OpenFoam softvér

OpenFoam je bezplatný „open source“ softvér, ktorý zahŕňa širokú škálu funkcií na riešenie mnohých inžinierskych aplikácií od zložitých tokov tekutín zahŕňajúcich chemické reakcie, turbulencie a prenos tepla, až po akustiku, mechaniku tuhých látok a elektromagnetickú energiu. OpenFoam je napísaný v jazyku C++ a využíva objektovo orientované techniky a preťaženie operátora, aby sa vytvorila syntax úrovne podobná tenzorovému zápisu parciálnych diferenciálnych rovníc. Softvér obsahuje rozhrania na predspracovanie dát, ako je napríklad vytvorenie výpočtovej siete a na následné spracovanie výsledkov ako napríklad ich vizualizácia.

Hlavná časť softvéru, knižnica pre metódu konečných objemov, rieši trojrozmerné parciálne diferenciálne rovnice na neštruktúrovaných a mnohostenných sieťach. Časť softvéru, ktorú sme využívali, je knižnica pre tzv. metódu konečných plôch, ktorá rieši kvázi dvojrozmerné parciálne diferenciálne rovnice na neštruktúrovaných mnohouholníkových sieťach na zakrivených plochách. V implementácii je výpočtová oblasť v metóde konečných plôch hranicou nejakej oblasti z metódy konečných objemov. Metóda konečných plôch bola integrovaná do softvéru v roku 2017, čo umožnilo rozšíriť škálu funkcií tohto softvéru o modelovanie lavín. Toto rozšírenie softvéru obsahuje nový model riešenia, ktorý počíta s účinkami ako nanášanie snehu, odtrhnutie časti snehu a trenie s terénom. Rozšírenie bolo pridané len do vývojovej verzie softvéru OpenFoam a preto práca s týmto rozšírením nebola jednoduchá. Poznatky sme získavali len zo vzorových príkladov a vedeckých článkov.

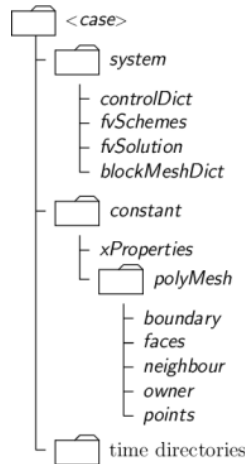
Softvér OpenFoam nemá grafické užívateľské prostredie a aplikácie sa spúšťajú v príkazovom riadku. Obsahuje aplikácie na vytvorenie diskretizovanej výpočtovej oblasti, spustenie výpočtu alebo napríklad využitie programu ParaView pre vizualizáciu dát.

V nasledujúcich podsekciiach si popíšeme časti softvéru, s ktorými sme sa zoznámili a pracovali s nimi [1].

### 4.1 Štruktúra súborov

V tejto časti si opíšeme použitie najdôležitejších vstupných a riadiacých textových súborov pre použitie v softvéri OpenFoam.

Na začiatku každého súboru v OpenFoam sa nachádza hlavička, ktorá obsahuje logo softvéru s číslom aktuálnej verzie, adresu oficiálnej internetovej stránky a informácie o danom súbore. V informáciach o súbore sa nachádza číslo verzie súboru, jeho formát, trieda, umiestnenie a druh obsahu súboru.



Obr. 2: Základná štruktúra priečinkov a súborov pri riešení problému pomocou OpenFoam [1]

Priečinkov s názvom *constant* obsahuje priečinkov *polyMesh*, kde sa nachádza plný popis siete terénu daného problému. Tento priečinkov si neskôr dôkladne popíšeme. Súbor, ktorý špecifikuje fyzikálne vlastnosti, sa nazýva *xProperties*. Ak sa jedná o veličiny súvisiace s prúdením tekutiny, tak máme súbor s názvom *transportProperties*, ktorý si popíšeme neskôr na konkrétnom príklade.

Ďalší dôležitý priečinkov má názov *system*. V ňom sa nachádzajú všetky nastavenia spojené s postupom riešenia problému. Obsahuje minimálne tri súbory. Súbor *controlDict* slúži na nastavenie parametrov riadenia výpočtu vrátane počiatočného a koncového času, veľkosti časového kroku a parametrov pre výstupné dáta. V súbore *fvSchemes/faSchemes* sú definované numerické schémy pre diferenciálne operátory, ktoré sa používajú pri výpočte. Posledný súbor je súbor s názvom *fvSolution/faSolution*, v ktorom sa nastavujú rovnice popisujúce problém, ktorý riešime, tolerancia presnosti výpočtu a predpokladovače.

Ako poslednú si uvedieme skupinu priečinkov *time directories*, ktoré obsahujú individuálne súbory dát pre konkrétne veličiny. Tieto dáta sú výsledky vypočítané a zapísané pomocou softvéru OpenFoam, okrem súboru, v ktorom sa nachádzajú počiatočné podmienky. Priečinky s výsledkami sú pomenované podľa času, v ktorom

boli výsledky zaznamenané. Keďže zvyčajne začíname simuláciu v čase  $t = 0$  s, tak počiatočné podmienky sú uložené v prierečinku s názvom 0.

## 4.2 Rozmerové jednotky

Je známe, že v mechanike kontinua sú vlastnosti látok reprezentované v nejakých zvolených jednotkách. Algebraické operácie sa musia vykonávať na týchto vlastnostiach s použitím konzistentných jednotiek merania. Ako ochrana pred implementáciou nezmyselnej operácie OpenFoam pripája rozmerové jednotky k dátovým poliam a fyzikálnym vlastnostiam a vykonáva kontrolu rozmerových jednotiek pri každej operácii. OpenFoam je kompatibilný s rôznymi sústavami jednotiek nielen SI.

P.č.	Vlastnosť	SI jednotka
1	Hmotnosť	kilogram (kg)
2	Dĺžka	meter (m)
3	Čas	sekunda (s)
4	Teplota	Kelvin (K)
5	Látkové množstvo	mol (mol)
6	Elektrický prúd	ampér (A)
7	Svietivosť	kandela (cd)

Tabuľka 1: Základné jednotky SI sústavy [1]

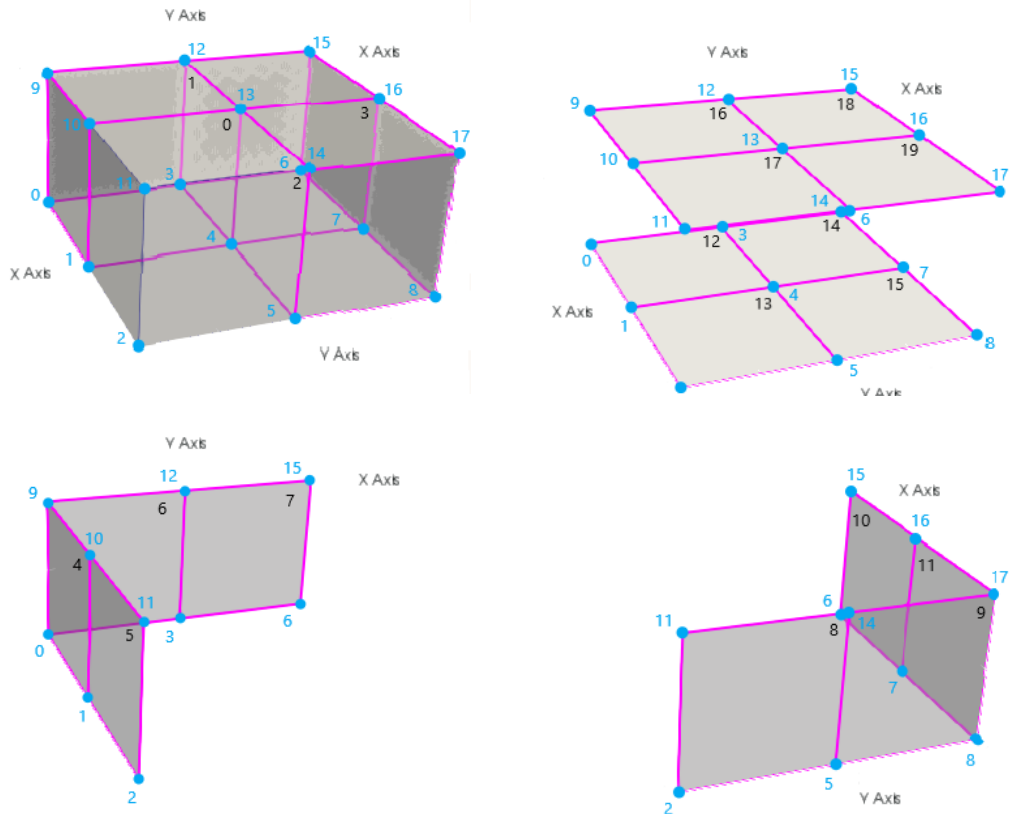
Vstupno-výstupný formát zadávania rozmerových jednotiek je sedem čísel ohraničených hranatými zátvorkami, za ktorými je uvedená hodnota danej veličiny, napríklad

$$[0\ 1\ -20000] 10 .$$

V hranatých zátvorkách každé číslo zodpovedá mocnine každej zo základných jednotiek uvedených v Tabuľke (1). Vo vyššie uvedenom príklade teda vidíme reprezentáciu zrýchlenia s hodnotou 10 a jednotkou  $m/s^2$ .

## 4.3 Reprezentácia geometrie

V tejto časti si popíšeme súbory, ktoré sa nachádzajú v spomínanom prierečinku *polyMesh*. Sú to súbory *points*, *faces*, *owner*, *neighbour* a *boundary*.



Obr. 3: Vľavo hore vidíme očíslované vnútorné steny oblasti, vpravo hore vidíme očíslované steny, ktoré tvoria spodnú a vrchnú časť oblasti, vľavo a vpravo dole vidíme bočné steny oblasti. Čísla stien sú čierne, čísla vrcholov sú modré.

Ako ukážku reprezentácie geometrie problému sme vytvorili jednoduchú oblasť, na ktorej si vysvetlíme význam všetkých súborov pre sieť na Obr. 3. Oblasť tvorí celkovo osemnásť vrcholov a dvadsať stien, ktoré vytvárajú štyri rovnaké bunky v tvare kocky. Ďalej popísané súbory sú k dispozícii ako príloha diplomovej práce.

V súbore *points* sú uložené informácie o bodoch, ktoré tvoria oblasť. Ako prvá hodnota je celkový počet bodov. V ďalšom riadku sú v zátvorkách súradnice jednotlivých bodov. V jednom riadku sa nachádza len jeden bod, čo určuje jeho poradové číslo. Prvý bod má index 0 a posledný v našom prípade 19.

Súbor *faces* obsahuje všetky steny, ktoré sa nachádzajú v oblasti. Jednu stenu tvorí usporiadaná  $n$ -tica bodov v závislosti od tvaru steny. Pri obdĺžnikových a štvorcových stenách sú to usporiadané štvorice bodov. V každej oblasti rozlišujeme okrajové a vnútorné steny. Pri vytváraní týchto stien musia platiť isté pravidlá. Pomocou stien, ktoré sú dvojdimenzionálne bunky sa vytvoria trojdimenzionálne bunky, pomocou ktorých máme reprezentovanú našu trojdimenzionálnu oblasť. Tieto bunky sú taktiež

zoradené a majú svoje indexy. V našej jednoduchej oblasti má bunka vľavo hore index 0, vpravo hore 1, vľavo dole 2 a vpravo dole 3. Pri definícii vnútorných stien musia byť body usporiadané tak, aby podľa pravidla pravej ruky smeroval normálový vektor vždy od bunky s nižším indexom k bunke s vyšším indexom. Pre vonkajšie steny platí pravidlo, že normálový vektor musí smerovať von z bunky. Pravidlo pravej ruky hovorí, že ak ohnuté prsty ukazujú smer v ktorom sú usporiadané body, tak vystretý palec ukazuje smer normálového vektora. V súbore *faces* máme na začiatku počet všetkých stien v oblasti, za ktorým nasledujú usporiadané  $n$ -tice. Ako prvé musíme mať v súbore zapísané vnútorné steny. Vonkajšie steny sú zapísané tak, aby indexy stien, ktoré tvoria spoločnú hranicu oblasti, nasledovali po sebe. Pred každou zátvorkou je uvedený počet vrcholov tvoriacich príslušnú stenu.

V nasledovnom súbore *owner* máme informácie o bunke „majiteľovi“ každej steny. To znamená, že má rovnaký počet riadkov ako súbor *faces*. V každom riadku je index bunky, ktorej patrí príslušná stena. Keďže vnútorné steny ležia na hranici medzi dvomi bunkami, musia jednej patriť a s tou druhou len susediť. Pre vnútorné steny máme súbor *neighbour*, v ktorom máme pre všetky vnútorné steny zapísané indexy buniek, s ktorými susedia. Tento súbor je dôvod, prečo musia byť vnútorné bunky zapísané ako prvé v súbore *faces*, inak by sme nevedeli rozlišovať vnútorné a vonkajšie steny.

Ako posledný máme súbor *boundary*, kde definujeme hranice našej oblasti. Ako aj v predošlých súboroch máme tu informáciu o počte hraníc oblasti. Každá hranica má svoje pomenovanie, typ, počet stien, ktoré ju tvoria a index začiatkovej steny.

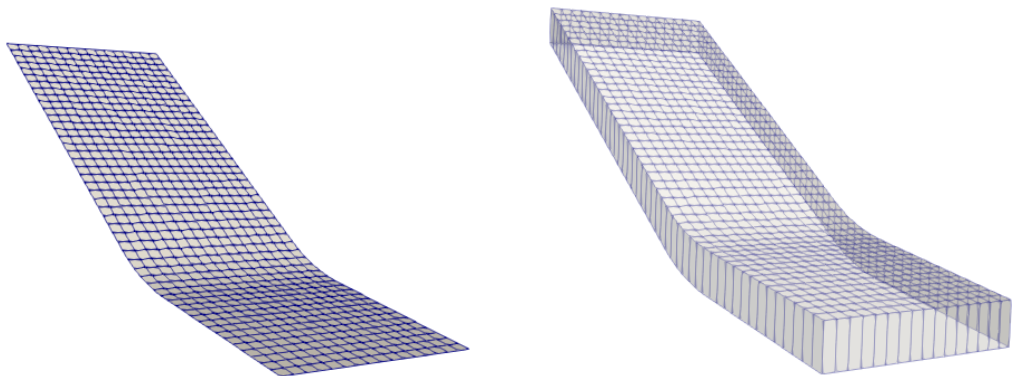
points	faces	owner	neighbour	boundary
18	20	20	4	6
((0 0 0)	(4(1 4 13 10)	(0	(2	teren
(10 0 0)	4(4 3 12 13)	0	1	
(20 0 0)	4(5 4 13 14)	2	3	(type patch;
(0 10 0)	4(4 7 16 13)	1	3)	nFaces 4;
(10 10 0)	4(0 1 10 9)	0		startFace 12;)
(20 10 0)	4(1 2 11 10)	2		
(20 10 0)	4(3 0 9 12)	0		⋮
⋮	⋮	⋮		
(20 20 10))	4(13 14 17 16))	3)		

Tabuľka 2: Ukážka súborov pre vytvorenú jednoduchú oblasť. V tabuľke máme zobrazených prvých osem riadkov s údajmi a posledný riadok. Pri súbore *boundary* máme zobrazený popis jednej hranice zo šiestich.

## 5 Príklad s jednoduchým terénom

Prácu so softvérom OpenFoam si popíšeme na príklade s jednoduchým terénom, ktorý sa nachádza v novom module *avalanche* ako vzorový príklad v adresári s názvom *tutorials*. Jedná sa o príklad pohybu lavíny po naklonenej rovine s dvomi rôznymi uhlami sklonu.

Ako prvý krok si musíme zostaviť geometriu. V tomto vzorovom príklade je na to napísaný skript s názvom *slopeMeshDict*, ktorý vygeneruje požadovanú naklonenú rovinu. V tomto skripte si môžeme nastaviť rozmery naklonenej roviny ako aj uhly sklonu. Pri vytváraní geometrie myslíme aj na hustotu výpočtovej siete, ktorú si tu tiež môžeme nastaviť. Softvér OpenFoam pracuje s trojrozmernými oblasťami a preto musíme zadať aj hodnotu tretej dimenzie, najčastejšie volíme nenulovú konštantu, pozri Obr. 4 pre vygenerovanú 2D a 3D geometriu. Po spustení skriptu dostaneme všetkých päť súborov, ktoré reprezentujú geometriu.



Obr. 4: Vľavo vidíme vygenerovanú naklonenú rovinu, vpravo máme zobrazenú celú oblasť.

V nasledujúcom kroku si nastavíme počiatočnú podmienku. V súbore s názvom *releaseArea* definujeme parametre pre veličinu  $h$  reprezentujúcu výšku snehu. Nastavíme jej počiatočnú hodnotu pomocou formátu pre zadávanie rozmerových jednotiek a následne zdefinujeme oblasť, v ktorej bude hodnota výšky snehu rovná konštantnej nenulovej hodnote. V tomto prípade zadáme súradnice stredu kruhu, polomer kruhu a konštantnú výšku snehu, pozri Obr. 5. Súbor slúži ako vstupný parameter pre aplikáciu na vytvorenie počiatočnej podmienky s názvom *releaseAreaMapping*.

V súbore *transportProperties* zadávame hodnoty veličín ako je napríklad hustota, počiatočné hodnoty pre rýchlosť a pre výšku snehu mimo podoblasti zadanej

v súbore *releaseArea*, pričom volíme najčastejšie veľmi malé hodnoty blízke nule. Ďalej v tomto súbore určíme či chceme počítat' aj s trením medzi snehom a terénom, nanášaním snehu alebo s odtrhom časti snehu. Ak zvolíme nejaký efekt, ktorý budeme uvažovať vo výpočte, musíme zadať koeficienty pre model, ktorý reprezentuje daný efekt. V tomto príklade nemeňte žiadne preddefinované nastavenie pre fyzikálne vlastnosti a uvažujeme len efekt trenia, ktorý je reprezentovaný modelom  $\mu I$  vid' rovnicu (4).

Teraz môžeme prejsť na súbor *controlDict*, kde si skontrolujeme všetky nastavenia pre výpočet a následné ukladanie výsledkov. V súbore máme nastavenú aplikáciu pre výpočet a to *faSevageHutterFoam*, ktorá rieši náš matematický model. Ďalej máme nastavený počiatkový čas na 0 a koncový čas na 15. Dĺžka časového kroku je rovná hodnote 0.1 a môže sa počas výpočtu meniť. Výsledky sa nám budú zapisovať v časovom intervale 0.5.

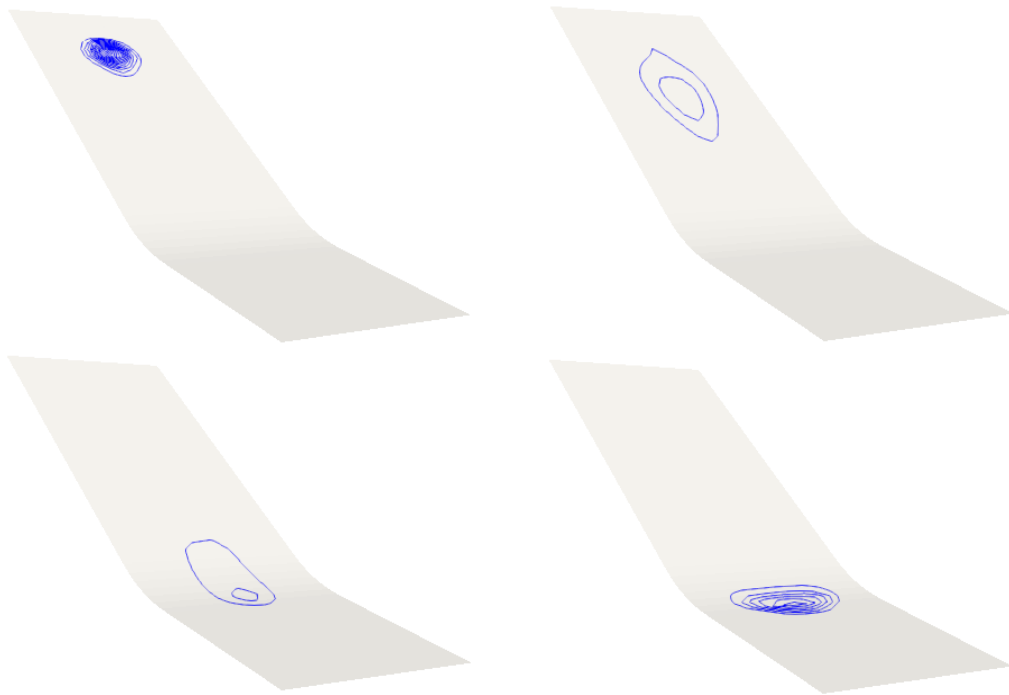
Spúšťanie aplikácie na riešenie daného problému môžeme robiť postupne manuálne alebo pomocou shellovského skriptu s názvom *Allrun*, v ktorom sú postupne napísane príkazy na spustenie všetkých aplikácií pre výpočet pomocou príkazu *runApplication* s argumentom príslušnej aplikácie. V prípade manuálneho spúšťania príkazov budeme dostávať výpis z aplikácie do príkazového riadku a v prípade, že spustíme shellovský skript, tak sa po ukončení každej aplikácie vytvorí takzvaný *logfile* pre každú aplikáciu, v ktorom sú uložené všetky výpisy danej aplikácie. Tieto súbory sú užitočné v prípadoch, keď nám výpočet nezbehne alebo zbehne nesprávne, pretože obsahujú aj chybové výpisy a môžeme potom ľahšie detekovať problém s výpočtom.

Po ukončení výpočtu sa nám v adresári zobrazia priečinky s dátami pre časové kroky podľa toho po akom časovom intervale sme si nastavili ich zapisovanie. Tieto výsledky ľahko zobrazíme v softvéri ParaView. Na vizualizáciu pomocou ParaView napíšeme do príkazového riadku *paraFoam*. Otvorí sa program a v ľavom hornom okne vidíme súbor s názvom nášho príkladu s príponou *.foam*. Tento súbor bol vytvorený softvérom OpenFoam po zadaní príkazu *paraFoam*. Pod oknom s názvom príkladu máme okno s možnosťami zobrazenia. Tu si môžeme zvoliť, ktorú časť geometrie chceme zobrazit'. V menu máme na výber všetky okrajové oblasti tak ako sú pomenované v súbore *boundary* plus možnosť zobrazit' celú oblasť. Ďalej si môžeme vybrať, či chceme načítat' všetky veličiny spojené s našim príkladom alebo len niektoré ako napríklad výšku snehu



$h$  alebo rýchlostné pole  $U$ . Po potvrdení všetkých počiatočných nastavení tlačidlom *Apply* vidíme našu naklonenú rovinu bez počiatočnej podmienky ako na Obr. 5.

Pre zobrazenie snehu musíme aplikovať zobrazenie jednej alebo viacerých vrstevníc funkcie  $h$ . Na zobrazenie vrstevníc aplikujeme filter *Contour*. Hodnoty pre konkrétne vrstevnice si môžeme zadať dvomi spôsobmi. Prvý spôsob predstavuje postupné zadávanie hodnôt podľa nášho výberu alebo pomocou druhého spôsobu, kedy zadáme minimálnu a maximálnu hodnotu a počet vrstevníc, ktoré chceme zobraziť na danom intervale. Ako minimálnu hodnotu zvolíme číslo o trochu väčšie ako 0 a pre maximálnu hodnotu môžeme zvoliť hodnotu počiatočnej podmienky alebo, keďže simulujeme pohyb lavíny, pri ktorom má sneh tendenciu sa nahromadiť, je rozumné zvoliť aj hodnotu väčšiu ako je v počiatočnej podmienke. Chceme, aby vrstevnice zodpovedali výške snehu a preto zvolíme veličinu  $h$  ako zdroj pre tento filter.

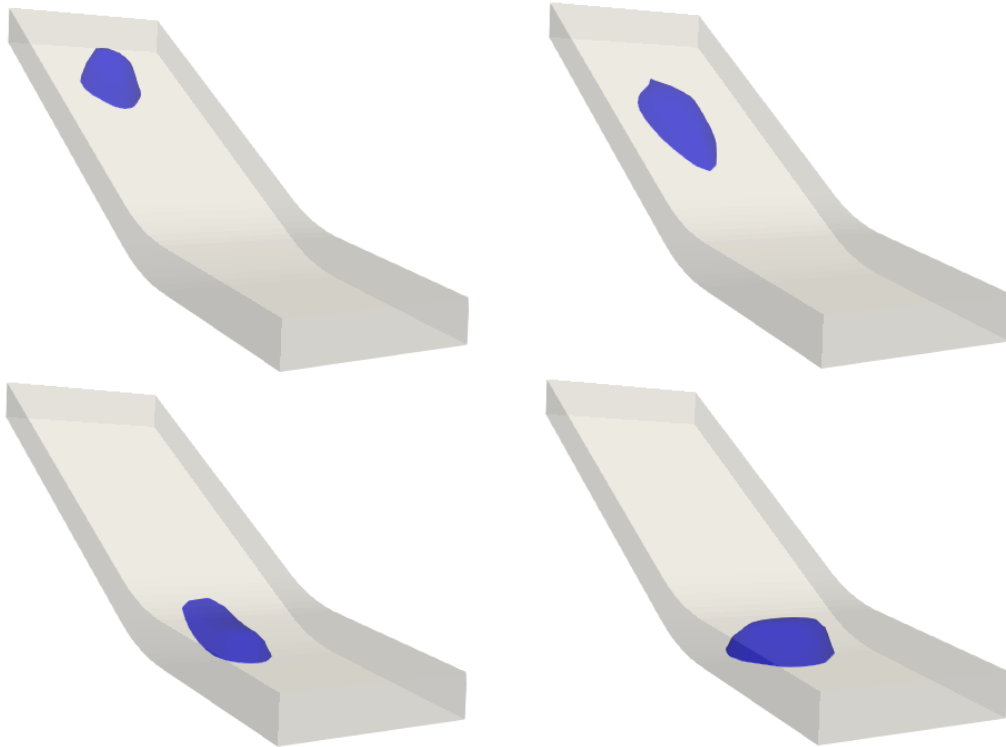


Obr. 5: Vľavo hore vidíme počiatočnú podmienku, vpravo hore vidíme polohu lavíny v čase  $t = 2.5$  s, vľavo dole vidíme polohu lavíny v čase  $t = 5$  s a vpravo dole vidíme, kde sa lavína zastavila v čase  $t = 7$  s. Zobrazených máme 10 rovnomerne rozdelených vrstevníc na intervale od 0.01 po 0.17, čo je výška snehu v koncovom čase.

Po aplikovaní filtra vidíme počiatočnú podmienku. Výsledné dáta sú uložené po každom časovom kroku a preto si vieme zobrazíť animáciu pohybu lavíny po naklonenej rovine podobne ako na Obr. 5. Pre zobrazenie iných veličín ako je výška snehu,

napríklad rýchlosť alebo tlak, stačí zmeniť veličinu, ktorú zobrazujeme.

Vieme, že OpenFoam pracuje s trojrozmernými oblasťami a preto si môžeme pre lepšiu vizualizáciu pomôcť aj tretím rozmerom, pozri Obr. 6. V tomto prípade filter *Contour* nezobrazuje vrstevnice s rovnakou výškou snehu, ale plochy s rovnakou výškou tzv. izoplochy. Pri tejto vizualizácii je praktickejšie zobrazovať iba jednu izoplochu.



Obr. 6: Vľavo hore vidíme počiatočnú podmienku, vpravo hore vidíme pohyb lavíny v čase  $t = 2.5$  s, vľavo dole vidíme pohyb lavíny v čase  $t = 5$  s a vpravo dole vidíme, kde sa lavína zastavila v čase  $t = 7$  s.

## 6 Modelovanie lavíny v oblasti Príslopu v Západných Tatrách

Po zvládnutí viacerých jednoduchých príkladov a ich modifikácii sa pustíme do modelovania lavíny na reálnom teréne, ktorý zodpovedá oblasti Príslopu v Západných Tatrách. Pre túto oblasť sme sa rozhodli po návšteve Strediska lavínovej prevencie Horskej záchranej služby, pri ktorej sme konzultovali problematiku simulovania lavín. Počas návštevy sme sa dozvedeli, že v danej oblasti sa lavíny nielen vyskytujú, ale ich aj majú zaznamenané na videu. Kľúčová informácia bola zistenie, že táto oblasť bola zdigitalizovaná. Tieto dáta nám boli poskytnuté na akademické účely.

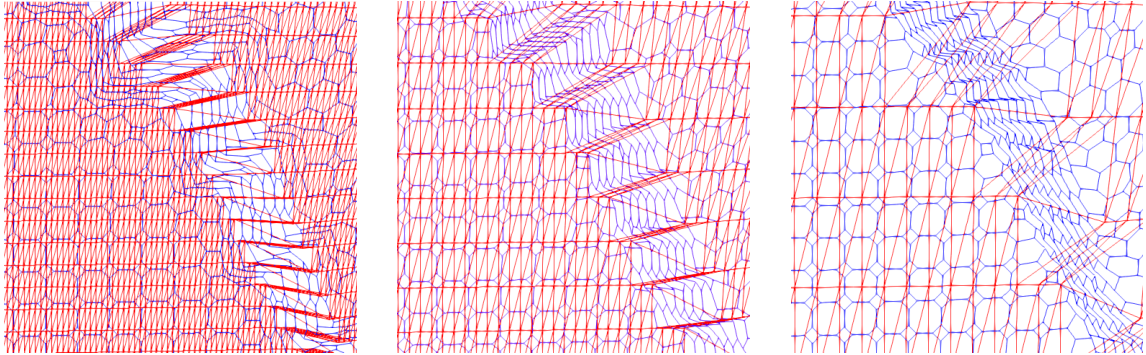
### 6.1 Spracovanie dát

Digitalizácia oblasti prebehla na základe spracovania snímok získaných z dronu, ktorý podrobne snímал oblasť. Model oblasti bol vo formáte `.tiff`. Tento formát slúži na ukladanie rastrovej počítačovej grafiky. Softvér OpenFoam dokáže načítať topografiu vo formáte `.obj`, ktorý reprezentuje 3D geometriu tak, že obsahuje všetky súradnice každého vrcholu a tiež všetky steny, ktoré tvoria objekty, definované ako zoznam bodov. Naša úloha bola vytvoriť súbor vo formáte `.obj` z poskytnutého súboru vo formáte `.tiff`. Ďalší problém, ktorý sme museli riešiť, je veľkosť dát. Dáta obsahujú oblasť s pomerne veľkou rozlohou a pri presnosti 35 centimetrov majú veľké pamäťové nároky.

Postupovali sme nasledovne. Prvé spracovanie dát bolo do formátu `.grd`, ktorý obsahuje informácie o počte bodov v smere osi  $x$  a  $y$ , následne maximálne a minimálne hodnoty pre súradnice  $x$ ,  $y$  a  $z$  a po nich nasledujú hodnoty súradnice  $z$  pre jednotlivé body. Tento súbor má veľkosť 1.1 GB a preto je potrebné našu oblasť zmenšiť len na okolie žľabu, do ktorého padajú lavíny. Nový súbor zmenšenej oblasti má veľkosť 240 MB. Ďalší krok spracovania dát bol vytvoriť zo súboru `.grd` súbor `.obj`. Na toto preformátovanie sme si vytvorili jednoduchý program v jazyku C++, ktorý je súčasťou tejto diplomovej práce a je k dispozícii ako príloha s názvom `converter.cpp`. V tomto programe sme načítali dáta z `.grd` súboru a následne sme ich spracovali a zapísali do nového súboru tak, aby spĺňal požiadavky pre formát `.obj`.

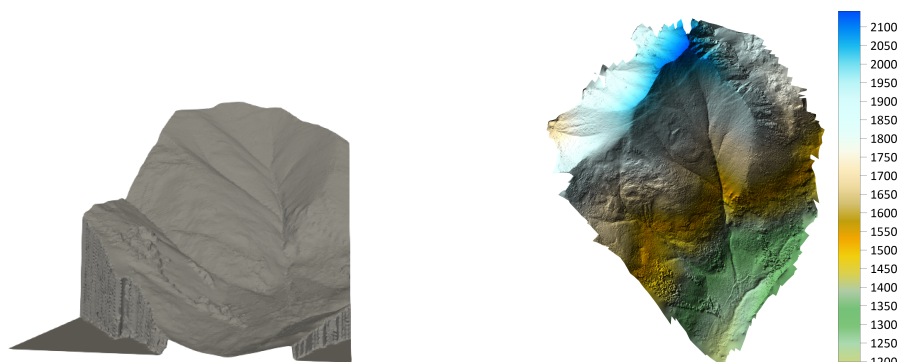
Ak by sme dodržali presnosť pôvodných dát, ktoré tvori približne 13 500 000 vrcholov, tak výsledný súbor by mal niekoľkokrát väčšiu veľkosť ako `.grd` súbor a výpočty

na takej oblasti by boli časovo a pamäťovo náročné. Pre zníženie týchto nárokov bolo potrebné pri vytváraní výsledného `.obj` súboru vytvoriť hrubšiu sieť. Pôvodnú sieť sme zriedili najskôr 100-krát, potom 400-krát a nakoniec 1600-krát.



Obr. 7: Zľava doprava vidíme červenou farbou sieť nami vytvoreného súboru `.obj`, ktorá je postupne 100, 400, 1600 -krát redšia. Modrou farbou vidíme vygenerovanú výpočtovú oblasť softvérom OpenFoam pre príslušný súbor `.obj`.

Počas generovania hrubších sietí sme si pre všetky tri nové siete merali výpočtový čas. Vygenerovanie 100-krát redšej siete trvalo 82 sekúnd, pre 400-krát redšiu sieť sme namerali čas 19 sekúnd a najrýchlejšie sa vygenerovala sieť 1600-krát redšia s časom 7 sekúnd. Vidíme, že výpočtový čas klesá so znižovaním nárokov na jemnosť siete. Pri všetkých troch sieťach softvér OpenFoam dokázal vytvoriť takmer rovnakú výpočtovú sieť a pri všetkých troch by sme teda dostali približne rovnaké výsledky. Výpočtový čas na vytvorenie siete v softvéri OpenFoam je približne rovnaký pre všetky nami vygenerované vstupné siete. Pre náš výpočet sme sa rozhodli pre sieť 400 krát redšiu z dôvodu optimálneho pomeru jemnosti siete voči výpočtovému času. Vybraná sieť reprezentuje našu oblasť pomocou približne 33 800 vrcholov.



Obr. 8: Vpravo vidíme pôvodnú oblasť, vľavo vidíme výrez oblasti na hlavný žľab reprezentovaný sieťou 400 krát redšou ako pôvodné dáta.

Teraz si opíšeme hlavné časti nami vytvoreného programu. Prvý krok je načítanie dát zo súboru do jednorozmerného poľa s názvom *data*. V tomto poli máme uloženú *z*-ovú súradnicu každého bodu oblasti. Ďalej zapisujeme do nového súboru vrcholy oblasti tak, ako vidíme v ukážke kódu

Kód 1: Zápis vrcholov do súboru .obj

```

for (i = 0; i < rows; i+=scalerows)
  for(j = 0; j < columns; j+=scalecolumns)
  {
    index0 = i * columns + j;
    myfile << "v" << " " << (i * 0.35) << " "
      << (-j * 0.35) << " " << data[index0] << "\n";
  };
```

kde hodnoty premenných *rows* a *columns* sme načítali z pôvodného súboru. Premenné *scalerows* = 40 a *scalecolumns* = 10 reprezentujú škálovacie koeficienty v smere *y* a *x*. V ukážke kódu zapisujeme vrcholy terénu. Aby sme dosiahli trojrozmernú oblasť, musíme ešte zapísať ďalšie vrcholy nad terén. Postupovať budeme analogicky a zmeníme len zápis *z*-ovej súradnice na konštantu rovnú maximálnej *z*-ovej súradnice zväčšenú o 10.

V ďalšom kroku zapisujeme do súboru steny, ktoré tvoria celú oblasť. Pre každú časť oblasti vytvoríme trianguláciu a tieto trojuholníky následne zapíšeme do súboru.

Kód 2: Zápis triangulácie do súboru .obj pre časť terén

```

int rows1 = (rows / scalerows)+1;
int columns1 = (columns / scalecolumns)+1;
int rxc = rows1 * columns1;

for (i = 1; i < rows1; i++)
  for (j = 1; j < columns1; j++)
  {
    index = columns1*(i - 1) + j;
    myfile << "f" << " " << index << " " << index + columns1
```

```

    << "┘" << index + columns1 + 1 << "\n";
myfile << "f" << "┘" << index << "┘"
    << index + columns1 + 1 << "┘" << index + 1 << "\n";
};

```

Trianguláciu vytvoríme podobne aj pre ostatných 5 častí oblasti. Po zbehnutí programu dostaneme reprezentáciu našej oblasti vo formáte `.obj` na Obr. 8.

## 6.2 Nastavenie a spustenie výpočtu

Novovytvorený súbor reprezentujúci geometriu presunieme do adresára s názvom *triSurface*, ktorý sa nachádza v priečinku *constant*. V priečinku *system* v súbore *meshDict* nastavíme premennej *surfaceFile* cestu k nášmu súboru. V súbore *meshDict* môžeme zadať nastavenia pre novú sieť, ktorú vygeneruje softvér OpenFoam, ako nastaviť maximálnu veľkosť bunky siete alebo jednotlivým častiam oblasti nastaviť úroveň zjemňovania siete, ak je to potrebné.

Nenulovú počiatočnú podmienku pre výšku lavíny zadefinujeme na štvoruholníkovej podoblasti, ktorej konkrétne štyri body zadáme do súboru *releaseArea*. Súradnice bodov definujúcich štvoruholníkovú podoblasť sme získali pomocou softvéru ParaView. Načítali sme súbor so sieťou vygenerovanou pomocou OpenFoam a vyznačili sme si oblasť pre počiatočnú podmienku. Súradnice vyznačených bodov si prepíšeme do súboru *releaseArea*. Počiatočnú konštantnú výšku snehu v tejto štvoruholníkovej oblasti zadáme s hodnotou  $h_0 = 0.5$  m.

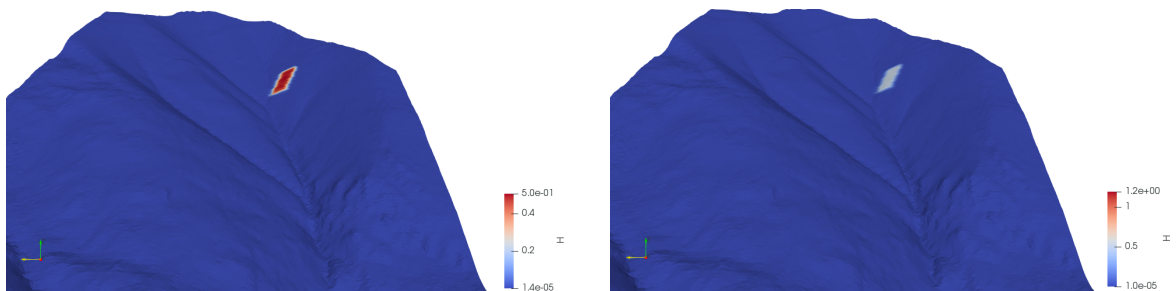
Fyzikálne vlastnosti snehu v súbore *transportProperties* sme na rozdiel od príkladu na naklonenej rovine trochu pozmenili v rozumnej miere s cieľom dosiahnutia výsledkov podobných reálnemu priebehu lavíny na videu. Minimálnu výšku lavíny na celej oblasti sme nastavili na číslo takmer rovné 0, čiže  $hmin = 1 \times 10^{-5}$  m. Hustotu snehu sme nastavili na hodnotu  $\rho = 200 \text{ kg m}^{-3}$ , ktorá zodpovedá pevnému suchému snehu [4]. Počiatočnú hodnotu pre výšku snehu  $h$  na celej oblasti sme zadefinovali ako  $h_0 = 1 \times 10^{-4}$  m a počiatočnú hodnotu pre veľkosť rýchlosti sme zvolili rovnako  $u_0 = 1 \times 10^{-4} \text{ m s}^{-1}$ . Pre trenie sme zvolili model Voellmy s parametrami  $\mu = 0.4$  a  $\xi = 750$ , ktoré sa javili vhodné pre náš model [2].

Priebeh výpočtu nastavíme v súbore *controlDict*. Na samotný výpočet použijeme aplikáciu *faSevageHutterFoam*, ďalej nastavíme, že výpočet sa bude realizovať od pr-

vého časového kroku, keďže v nultom máme zadefinované počiatočné podmienky, po koncový čas  $t = 120$  s. Dĺžku časového kroku sme nastavili na hodnotu 0.2, aby sme dosiahli čo najpresnejšie výsledky. Zápis výsledkov nastavíme tak, aby sme mali zapísanú hodnotu po každých piatich časových krokoch, preto premennú *writeInterval* nastavíme na 1. Formát výsledkov nastavíme na ASCII s presnosťou na 6 desatinných miest. Časový formát nastavíme na *general*, toto nastavenie popisuje formát pomenovania časových prierečinkov, v ktorých budú uložené výsledky. Presnosť pre čas nastavíme rovnako na 6 desatinných miest. Hodnota *yes* pri premennej *runTimeModifiable* nám dovoľuje počas výpočtu meniť jeho parametre. Premenná *adjustTimeStep* definuje možnosť modifikovať si dĺžku časového kroku pre efektívnejší výpočet.

### 6.3 Vizualizácia výsledkov pomocou softvéru ParaView

Softvér ParaView ponúka široké spektrum možností na vizualizáciu prúdenia. Obsahuje rôzne filtre a tie, s ktorými sme pracovali, si popíšeme. Veličiny a vektorové polia si zobrazíme veľmi jednoducho. Máme zobrazený terén našej oblasti, na ktorom nemáme žiadne farebné rozlíšenie a teda celá oblasť je jednofarebná. Práve zmenou zafarbenia oblasti na základe zvolenej veličiny dosiahneme jej vizualizáciu. Musíme ale dbať na rozsah farebnej škály, ktorý závisí od maximálnej a minimálnej hodnoty veličiny alebo vektorového poľa.

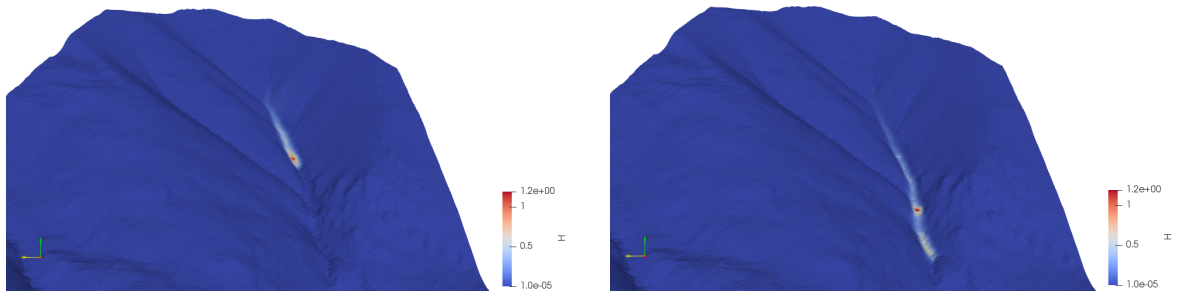


Obr. 9: Vľavo vidíme zobrazenú oblasť zafarbenú podľa výšky snehu s rozsahom farebnej škály aktuálneho časového kroku, vpravo vidíme zobrazenie s rozsahom farebnej škály získanej zo všetkých časových krokov

Na Obr. 9 máme zobrazenú veličinu  $h$  v čase  $t = 0$  s, teda vidíme počiatočnú podmienku. Rozsah hodnôt na obrázku vľavo je získaný z počiatočnej podmienky a preto vidíme oblasť s hodnotou  $h = 0.5$  m zafarbenú farbou pre maximálnu hodnotu, červenou. Ak by sme spustili simuláciu, nikde by sme nevideli oblasť, kde je výška

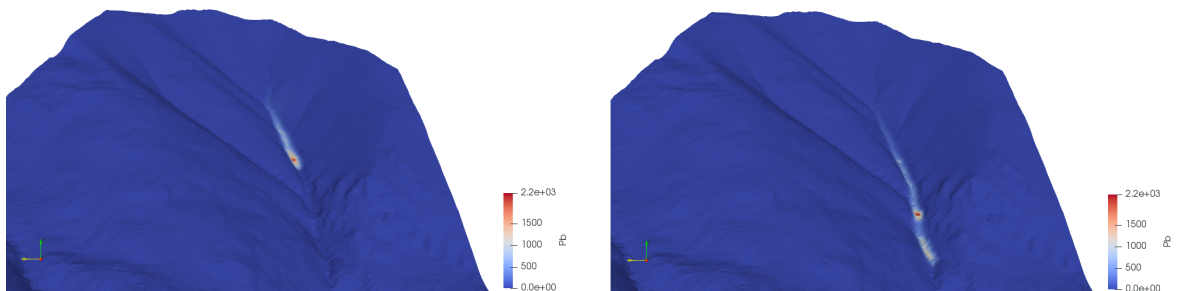
snehu väčšia ako  $h = 0.5$  m, pretože je to maximálna hodnota. Na obrázku vpravo sú hodnoty pre farebnú škálu získanu zo všetkých časových krokov. Červená farba teraz prislúcha maximálnej hodnote, ktorá je dosiahnutá počas simulácie.

Veličiny a vektorové polia, ktoré máme uložené pre každý interval zápisu, sú pre rýchlosť, tlak a výšku snehu. Vizualizácia každej z týchto veličín ma veľkú výpovednú hodnotu a preto si ich zobrazíme podľa opísaného postupu so škálou získanou zo všetkých časových krokov.



Obr. 10: Vľavo máme zobrazenú výšku snehu na celej oblasti v čase  $t = 60$  s a vpravo máme zobrazenú výšku snehu v koncovom čase  $t = 120$  s

Pri zobrazení výšky snehu pozorujeme, že maximálna hodnota  $h = 1.2$  m v čase  $t = 60$  s sa nachádza na čele lavíny. V koncovom čase maximálna hodnota už nie je na čele lavíny, ako by sme to mohli predpokladať. Je to spôsobené členitosťou terénu v mieste, kde vidíme vyznačenú maximálnu hodnotu v koncovom čase. Nachádza sa tam priehlbina alebo prekážka, ktorá zastavila časť lavíny a ďalej sa dostala už len jej malá časť.

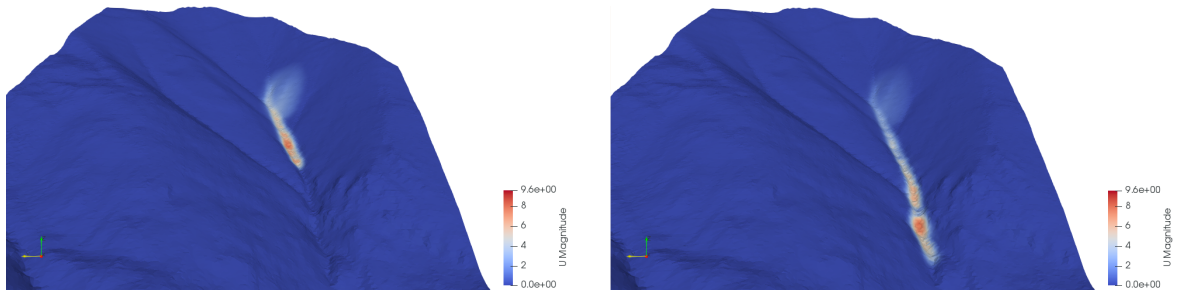


Obr. 11: Vľavo máme zobrazenú hodnotu tlaku na celej oblasti v čase  $t = 60$  s a vpravo máme zobrazenú hodnotu tlaku v koncovom čase  $t = 120$  s

Tlak je veličina, ktorá závisí od výšky snehu a preto vidíme takmer identické obrázky pre tieto dve veličiny. Najväčší tlak je v mieste s maximálnou výškou snehu.

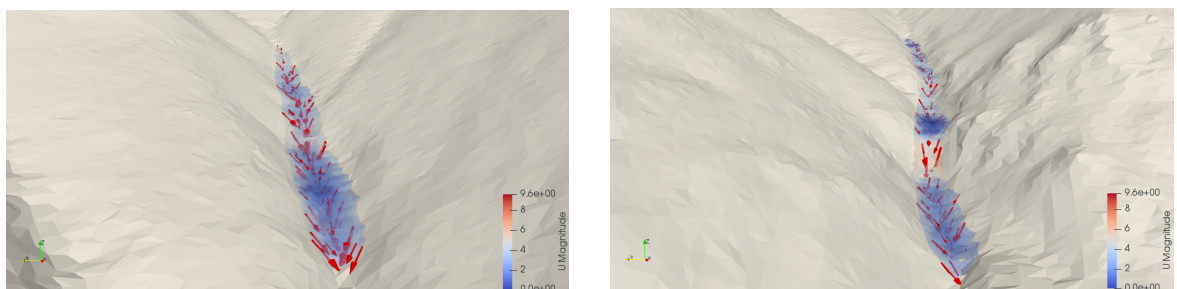


Ak by sme mali k dispozícii len vizualizáciu pre tlak, vedeli by sme určiť miesto s maximálnou výškou snehu.



Obr. 12: Vľavo máme zobrazenú veľkosť rýchlostného poľa v čase  $t = 60$  s a vpravo máme zobrazenú veľkosť rýchlostného poľa v koncovom čase  $t = 120$  s

Na Obr. 12 vidíme zobrazenú veľkosť (normu) vektorového poľa pre rýchlosť. V čase  $t = 60$  s máme rozloženie rýchlosti pomerne rovnomerné. V prednej a zadnej časti lavíny pozorujeme menšiu rýchlosť ako v strede. Stredná časť sa nachádza za oblasťou s maximálnou výškou snehu v danom časovom kroku. Teda pozorujeme zaujímavú vlastnosť našej simulácie a to, že najväčšia rýchlosť nie je na čele lavíny. V čase  $t = 120$  s pozorujeme, že najväčšia rýchlosť je znova niekde v centrálnej časti. V tomto prípade je to ale spôsobené sklonom terénu, pretože z vizualizácie výšky snehu vieme, že najväčšia časť lavíny sa zastavila už pred touto centrálnou časťou a hodnoty výšky snehu v tejto časti sú veľmi malé.



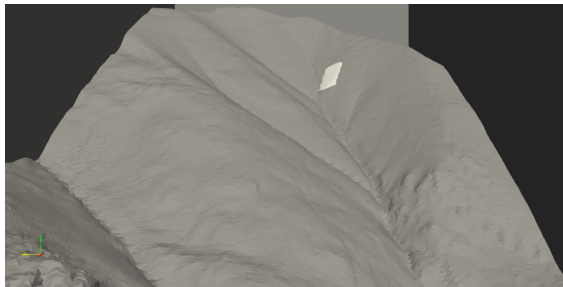
Obr. 13: Vľavo máme zobrazené rýchlostné pole len na oblasti so snehom pomocou šípok v čase  $t = 60$  s a vpravo máme zobrazené rýchlostné pole pomocou šípok v koncovom čase  $t = 120$  s

Jeden z výhodných spôsobov vizualizácie vektorového poľa je jeho reprezentácia pomocou šípok. V softvéri ParaView máme k dispozícii filter s názvom *Glyph*. Na výber nám ponúka reprezentáciu nielen pomocou šípok, ale aj napríklad pomocou jednoduchých čiar, sfér a iných objektov. Pre náš príklad sme si zvolili šípky. Zobrazovať ich

budeme len na oblasti so snehom a preto sme najskôr aplikovali filter *Contour*, ktorý nám zobrazuje výšku snehu, a následne len na túto oblasť sme aplikovali filter *Glyph*. Ako prvé musíme zvoliť pole na základe ktorého budú šípky orientované, teda zvolíme rýchlostné pole  $U$ . Rovnaké pole zvolíme aj pre škálovanie šípok so škálovacím faktorom s hodnotou 4, je to optimálna hodnota pre zobrazenie šípok v našom príklade. Šípky môžeme zobrazovať buď z každého bodu oblasti, čo nie je v našom príklade prehľadné, alebo môžeme zvoliť možnosť výberu bodov na základe rovnomerného priestorového rozloženia, alebo vyberať len každý  $n$ -tý bod z oblasti. Vyberáme každý dvadsiaty bod z oblasti. Na Obr. 13 vidíme v čase  $t = 60$  s, že šípky na ľavej a pravej strane oblasti smerujú do jej stredu, čo je spôsobené vplyvom terénu, ktorý je na oboch stranách naklonený. Šípky v strede smerujú v smere pohybu lavíny. Keď porovnáme tento obrázok s Obr. 12 vidíme, že najdlhšie šípky sú tam kde máme najväčšiu veľkosť rýchlosti. V koncovom čase si všimneme oblasť, kde sa zastavila veľká časť lavíny. V tomto mieste máme veľmi malé šípky, čo súhlasí s tým, že sa táto časť už nepohybuje. Šípky na čele lavíny sú síce väčšie, ale zobrazujú pohyb len časti snehu s minimálnou výškou a nie pohyb celej lavíny.

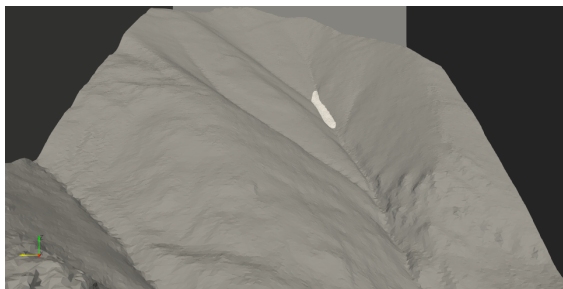
## 6.4 Porovnanie výsledkov

Simuláciu lavíny na reálnom teréne, ktorú sa nám podarilo namodelovať, môžeme po kvalitatívnej stránke porovnať s videom. Všetky parametre pre výpočet, vlastnosti snehu a počiatkové podmienky sme nastavili tak, aby sa približne podobali na lavínu v oblasti Príslopu, ktorá bola zaznamenaná na video. Porovnávať budeme obrázky zaznamenané v rovnakých časoch od odtrhnutia lavíny.



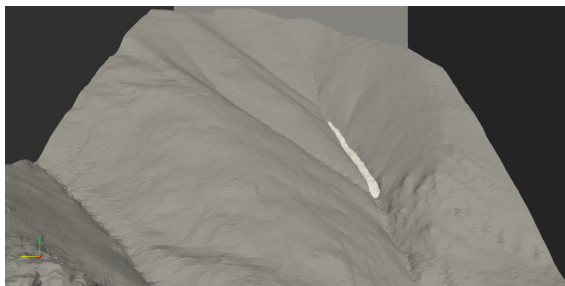
Obr. 14: Porovnanie numerického modelu lavíny naľavo s reálnou lavínou v danej oblasti v čase odtrhu  $t = 0$  s.

V počiatočnom časovom kroku vidíme, že sa nám podarilo dobre zvoliť hranice počiatočnej podmienky. Oblasť, na ktorej simulujeme lavínu, máme zobrazenú v 3D z dôvodu zobrazenia izoplôch. Týmto dosiahneme reálnejšiu reprezentáciu snehu a môžeme lepšie porovnať simuláciu.



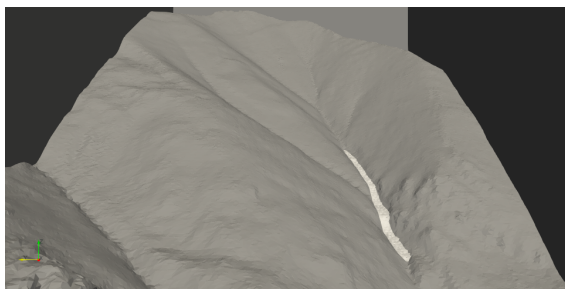
Obr. 15: Porovnanie numerického modelu lavíny naľavo s reálnou lavínou v danej oblasti v čase  $t = 30$  s.

Po niekoľkých časových krokoch v čase  $t = 30$  s pozorujeme, že sneh sa aj v simulácii aj na videu zosunul do oblasti žľabu. Tým sa zvýšila výška snehu a lavína sa natiahne v smere žľabu. Ďalej bude postupovať iba v tomto smere, čo spôsobí jej mierne zrýchlenie.



Obr. 16: Porovnanie numerického modelu lavíny naľavo s reálnou lavínou v danej oblasti v čase  $t = 80$  s.

V čase  $t = 80$  s si všimneme, že lavína na videu je ďalej ako lavína v simulácii. Toto je spôsobené aj tým, že približne v čase  $t = 40$  s sa k lavíne postupujúcej žľabom pripojila ďalšia menšia lavína. Objem lavíny sa zväčšil a na danom mieste to spôsobilo mierne zvýšenie rýchlosti lavíny na videu. Lavína v simulácii naďalej postupuje žľabom bez tohto efektu. V našej simulácii by sme vedeli jednoducho modelovať takú počiatočnú podmienku, kde by sme mali viacej oblastí spustenia lavíny. Problém je, že je to počiatočná podmienka a zo všetkých oblastí by sa spustila lavína naraz, čo by nezodpovedalo tejto situácii na videu. Z časových dôvodov sme sa nevenovali spôsobu zadávania odtrhu lavín v rôznych časoch, čo je jedna z možností ďalšieho vývoja práce na danej problematike.



Obr. 17: Porovnanie numerického modelu lavíny naľavo s reálnou lavínou v danej oblasti v čase zastavenia lavíny  $t = 120$  s.

V čase, keď sa obe lavíny zastavili, pozorujeme, že čelo oboch lavín končí na približne rovnakom mieste. Pri lavíne na videu predpokladáme, že bola dosiahnutá väčšia výška snehu z dôvodu pripojenia ďalšej lavíny. V jednotlivých časových krokoch môžeme pozorovať veľmi podobný spôsob pohybu lavíny na videu s našou simuláciou. Z tohto porovnania predpokladáme vhodnosť zvolenia parametrov modelu. Na ich výstižné určenie je potrebné použiť nielen ich subjektívne určenie na základe vizualizácie, ale i na základe kvantitatívneho porovnania nameraných a vypočítaných dát.

## 6.5 Trajektórie častíc lavíny

Ukázali sme si, ako vieme vizualizovať a interpretovať rôzne veličiny, polia, izoplochy a izočiaru. Všetky reprezentácie výsledkov, ktoré sme doteraz dosiahli, môžu byť veľmi užitočné. Sú to ale reprezentácie lavíny ako celku. Naš matematický model popisuje lavínu ako granulárne prúdenie. To znamená, že sneh reprezentujeme ako látku zloženú z malých častíc. Predstavme si situáciu, že lavína zo sebou strhne nejaký objekt a nás by zaujímalo, kde sa tento objekt ocitne v čase, keď sa lavína zastaví. Tieto myšlienky nás priviedli k tomu, že získať takúto informáciu by mohlo byť veľmi cenné a užitočné, preto sme sa rozhodli pokúsiť zobraziť i trajektórie častíc lavíny. Pripomíname, že tento zámer nebolo jednoduché dosiahnuť, keďže mnohé časti softvéru nie sú dostatočne opísané v užívateľských príručkách alebo sa nie je jednoduché k nim dostať.

Softvér ParaView ponúka niekoľko filtrov na zobrazenie trajektórií ako sú napríklad *StreamTracer*, *ParticleTracer*, *ParticlePath* a iné. Všetky tieto filtre sme skúsili aplikovať na našu simuláciu, ale ani jeden nám nedal požadované výsledky. Problém s filtermi je ten, že my máme dáta uložené pre každý časový krok zvlášť. Filtre vedeli zobraziť trajektórie bodov, ale len pre aktuálny časový krok. To znamená, že ak sme zobrazili trajektórie v nejakom časovom kroku, tak v nasledujúcom časovom kroku softvér zobrazí nové trajektórie, ale tie staré vymaže.

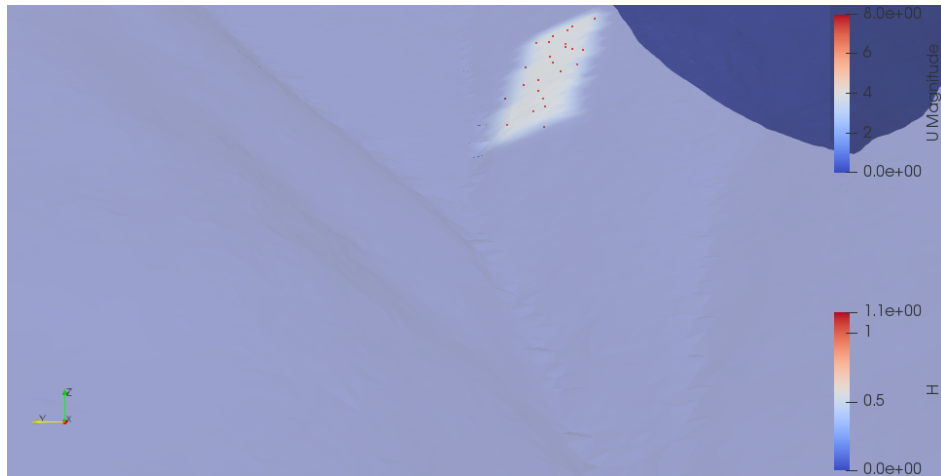
Všetky tieto pokusy nás priviedli k filteru *TemporalInterpolator*. Tento filter konvertuje dáta, ktoré sú definované v diskretných časových krokoch, do jedného časového kroku, ktorý je definovaný v priebehu času pomocou lineárnej interpolácie polí medzi po sebe idúcimi časovými krokmi. To je presne to, čo sme potrebovali. Tento filter je taktiež vhodný pre situácie, v ktorých by sme nemali k dispozícii všetky časové kroky, ale len niektoré. Filter by pomocou interpolácie vyplnil chýbajúce časové kroky, aby

sme dostali súvislý priebeh simulácie.

Chceme zobrazovať trajektórie častíc snehu a preto aplikujeme znovu filter *Contour*, ktorý bude tvoriť zdroj pre filter na zobrazenie trajektórii. Vyššie spomínané filtre na zobrazenie trajektórii, ktoré ponúka softvér ParaView, nám stále nedávajú rozumné výsledky.

Jediný filter, ktorý nám zatiaľ zobrazí súvislé trajektórie pre celý priebeh simulácie, je filter *TemporalParticlesToPathlines*. Nastavenia pre daný filter sú podobné ako pre *Glyph* filter.

Ako prvé nastavíme počet častíc, pre ktoré budeme sledovať trajektórie. Nastavíme hodnotu  $n = 100$ , na základe ktorej filter vyberie každý  $n$ -tý bod. Ďalej nastavíme maximálnu dĺžku trajektórie na hodnotu 2500. Zvolili sme dostatočne veľké číslo, aby sme videli zobrazenú celú trajektóriu. Ak by sme zadali nejaké malé číslo a dĺžka trajektórie by bola väčšia, tak by sa trajektória začala od jej začiatku vymazávať. Podobne ako pri dĺžke trajektórii teraz nastavíme maximálnu vzdialenosť pre časticu vo všetkých troch smeroch vzhľadom na jeden časový krok. Túto hodnotu sme zvolili 200. Všetky dôležité nastavenia sme takto zadali a môžeme aplikovať filter.

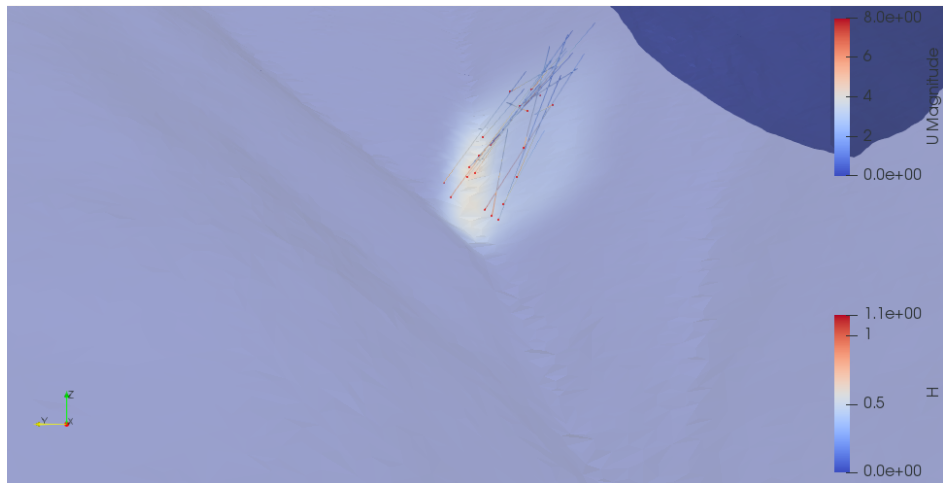


Obr. 18: Zobrazenú máme počiatočnú podmienku s bodmi, pre ktoré zobrazujeme trajektórie.

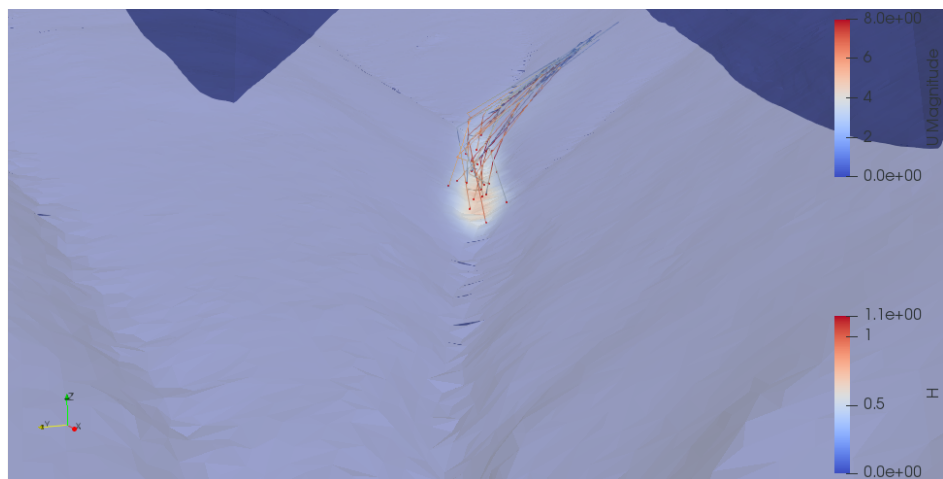
Zobrazenú oblasť na Obr. 18 máme zafarbenú podľa výšky snehu v aktuálnom čase. Keďže sme aplikovali na túto oblasť aj jej 50 % priehľadnosť, farba oblasti sa mierne líši od legendy pre veličinu  $h$  zobrazenej na pravej strane.

Trajektórie budú zafarbené podľa rýchlosti častíc  $U$ . Na obrázku Obr. 19 s trajektóriami vidíme, že body sa presunuli do oblasti, v ktorej sa nachádza sneh v aktuál-

nom časovom kroku. V oblasti počiatočnej podmienky sa ešte nachádzajú pozorované častice a preto predpokladáme zmenu trajektórií na tejto oblasti aj v ďalších časových krokoch.

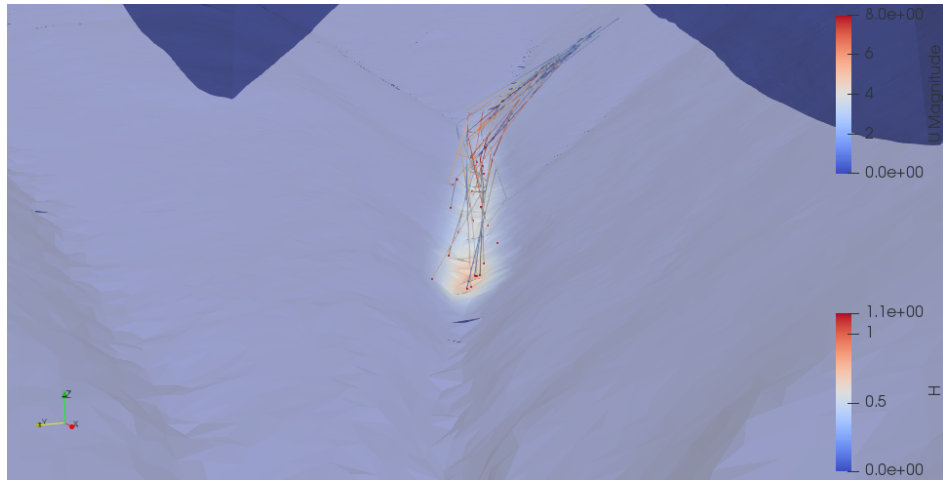


Obr. 19: Trajektórie zobrazené v čase  $t = 14.222$  s

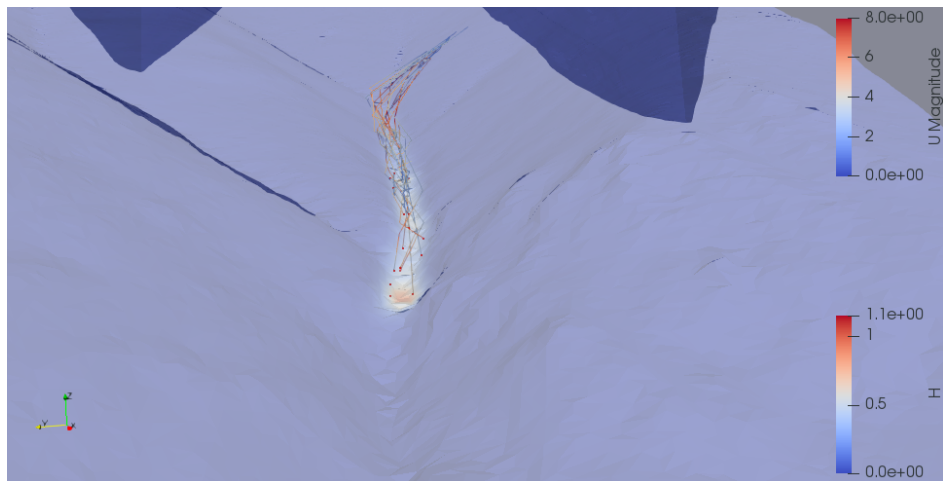


Obr. 20: Trajektórie v čase  $t = 40.666$  s.

Počet viditeľných trajektórií na Obr. 20 narástol a pozorujeme už pohyb všetkých častíc. Častice, ktorých trajektórie vidíme, výrazne zmenili smer do žľabu, čo vidíme na ich trajektóriách. V ďalšom priebehu simulácie už neočakávame takéto výrazné zmeny. Častice sa pohybujú už len v rámci žľabu z jednej strany na druhú. Rýchlosť pozorovaných častíc sa po tejto zmene smeru trochu zmenšila, čo si môžeme všimnúť na zafarbení jednotlivých trajektórií.



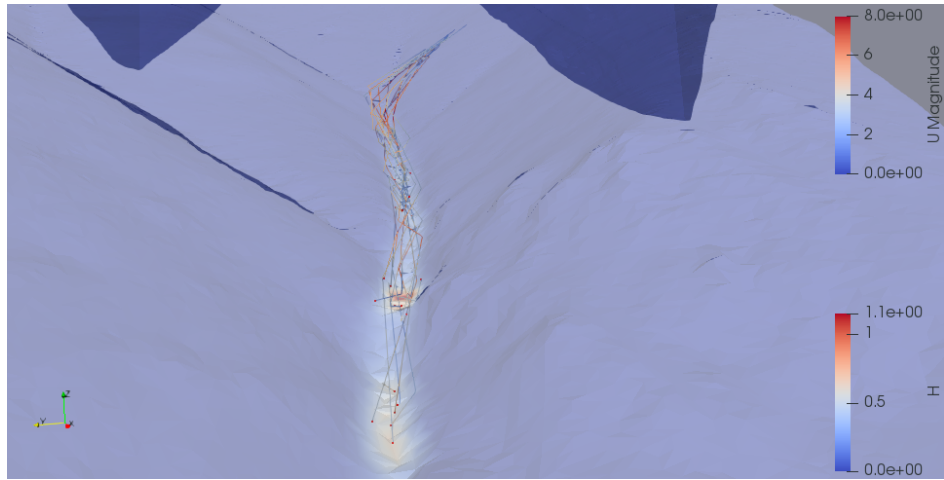
Obr. 21: Trajektórie v čase  $t = 53.88$  s.



Obr. 22: Trajektórie v čase  $t = 93.55$  s.

Smer trajektórií na Obr. 22 sa mení minimálne, čo si ale môžeme všimnúť je ich dĺžka. V čase  $t = 93.55$  s sa čelo lavíny nachádza v mieste kde zastane jej časť. Sledované častice sa pomaly približujú k tejto oblasti. V koncovom čase vidíme, že len časť zo všetkých sledovaných častíc sa dostala až tam, kam čelo lavíny.





Obr. 23: Trajektórie v koncovom čase  $t = 120$  s.

Časové kroky, v ktorých nám filter poskytol trajektórie sú relatívne dlhé. Nepodarilo sa nám manuálne zmeniť tieto dĺžky časového kroku, preto sme sa zatiaľ z časových dôvodov uspokojili s výsledkami, ktoré sme pri zobrazovaní trajektórií dosiahli.

## 7 Záver

Cieľom diplomovej práce bolo nielen osvojiť si samotný softvér OpenFoam a prácu s ním, ale predovšetkým jeho využitie pri simulovaní lavín. Táto práca môže takto slúžiť aj ako úvod na numerické simulácie so softvérom OpenFoam a predovšetkým s jeho modulom *avalanche*.

Po osvojení a modifikovaní niekoľkých vzorových príkladov, ktoré sme čiastočne v práci opísali, sme sa odhodlali simulovať lavínu na reálnom teréne. Pre túto simuláciu sme museli pomocou vlastného programu upraviť a preformátovať dáta, získane zo Strediska lavínovej prevencie Horskej záchranej služby tak, aby sme mohli importovať reálny terén do softvéru OpenFoam.

Po vytvorení vhodnej oblasti v požadovanom formáte sme študovali fyzikálne parametre potrebné pre simuláciu lavíny a určujúce jej priebeh. Spolu s týmito experimentami sme sa venovali vizualizácii všetkých výsledkov. Na vizualizáciu sme používali softvér ParaView, ktorý dokáže priamo zobrazíť výsledky zo softvéru OpenFoam.

Za hlavný prínos práce považujeme vytvorenie simulácie, ktoré je kvalitatívne porovnateľné s lavínou v danej oblasti zaznamenanou na video. Navyše sa nám do veľkej miery podarilo úspešne vizualizovať trajektórie jednotlivých vybraných častíc lavíny, čo je výrazná pridaná hodnota tejto simulácie.

Ciele práce, ktoré sa nám pri zadaní diplomovej práce javili značne netriviálne, keďže sme nemali žiadne skúsenosti so softvérom OpenFoam, sa nám podarilo splniť nad naše očakávania. Takto môžeme odporúčať použitie tohto nástroja nielen na modelovanie priebehu lavín, ale i na iné inžinierske aplikácie. Ako ďalšie možnosti vývoja problematiky simulácie lavín vidíme jej detailnejšie modelovanie (napríklad možnosť odtrhu lavíny v rôznych časoch simulácie alebo flexibilnejšiu vizualizáciu trajektórii), ako aj využitie vypočítaných výsledkov na kvantitatívne porovnanie s reálnymi dátami.

## Literatúra

- [1] OpenCFD Ltd (ESI Group). Openfoam official documentation, 2020. <https://www.openfoam.com/documentation>.
- [2] T.E. Lang M. Martinelli and A. I. Mears. Calculations of avalanche friction coefficients from field data. *Journal of Glaciology*, 26(94):109–118, 1980.
- [3] M. Rauter and Ž. Tuković. A finite area scheme for shallow granular flows on three-dimensional surface. *Computers and Fluids*, 166:184–199, 2018.
- [4] Horská záchranná služba. Lavíny hzs — náuka o snehu a lavínach - metodika, 2020. <http://www.laviny.sk/metodika-laviny/nauka-o-snehu-a-lavinach>.