

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA STAVEBNÍ

PROGRAM GEODÉZIE A KARTOGRAFIE  
OBOR GEOMATIKA



DIGITÁLNÍ MODEL TERÉNU A JEHO ANALÝZY

AUTOŘI: Bc. LINDA KLADIVOVÁ, Bc. JANA ŠPEREROVÁ

PŘEDMĚT: ALGORITMY DIGITÁLNÍ KARTOGRAFIE A GIS

### Úloha č. 3: Digitální model terénu

Vstup: množina  $P = \{p_1, \dots, p_n\}$ ,  $p_i = \{x_i, y_i, z_i\}$ .

Výstup: polyedrický DMT nad množinou  $P$  představovaný vrstevnicemi doplněný vizualizací sklonu trojúhelníků a jejich expozicí.

Metodou inkrementální konstrukce vytvořte nad množinou  $P$  vstupních bodů 2D Delaunay triangulaci. Jako vstupní data použijte existující geodetická data (alespoň 300 bodů) popř. navrhněte algoritmus pro generování syntetických vstupních dat představujících významné terénní tvary (kupa, údolí, spočinek, hřbet, ...).

Vstupní množiny bodů včetně níže uvedených výstupů vhodně vizualizujte. Grafické rozhraní realizujte s využitím frameworku QT. Dynamické datové struktury implementujte s využitím STL.

Nad takto vzniklou triangulací vygenerujte polyedrický digitální model terénu. Dále provedte tyto analýzy:

- S využitím lineární interpolace vygenerujte vrstevnice se *zadaným krokem* a v *zadaném intervalu*, provedte jejich vizualizaci s rozlišením zvýrazněných vrstevnic.
- Analyzujte sklon digitálního modelu terénu, jednotlivé trojúhelníky vizualizujte v závislosti na jejich sklonu.
- Analyzujte expozici digitálního modelu terénu, jednotlivé trojúhelníky vizualizujte v závislosti na jejich expozici ke světové straně.

Zhodnotte výsledný digitální model terénu z kartografického hlediska, zamyslete se nad slabinami algoritmu založeného na 2D Delaunay triangulaci. Ve kterých situacích (různé terénní tvary) nebude dávat vhodné výsledky? Tyto situace graficky znázorněte.

Zhodnocení činnosti algoritmu včetně ukázek provedte alespoň na tři strany formátu A4.

#### Hodnocení:

Krok	Hodnocení
Delaunay triangulace, polyedrický model terénu.	10b
Konstrukce vrstevnic, analýza sklonu a expozice.	10b
Triangulace nekonvexní oblasti zadáné polygonem.	+5b
Výběr barevných stupnic při vizualizaci sklonu a expozice.	+3b
Automatický popis vrstevnic.	+3b
Automatický popis vrstevnic respektující kartografické zásady (orientace, vhodné rozložení).	+10b
Algoritmus pro automatické generování terénních tvarů (kupa, údolí, spočinek, hřbet, ...).	+10b
3D vizualizace terénu s využitím promítání.	+10b
Barevná hypsometrie.	+5b
<b>Max celkem:</b>	<b>65b</b>

Čas zpracování: 3 týdny

# Obsah

<b>1 Popis a rozbor problému</b>	<b>5</b>
1.1 Údaje o bonusových úlohách . . . . .	5
<b>2 Popis použitých algoritmů</b>	<b>6</b>
2.1 Delaunayho triangulace . . . . .	6
2.1.1 Slovní zápis algoritmu . . . . .	7
2.2 Konstrukce vrstevnic . . . . .	8
2.2.1 Slovní zápis algoritmu . . . . .	9
2.3 Analýza sklonu DMT . . . . .	9
2.3.1 Slovní zápis algoritmu . . . . .	9
2.4 Analýza expozice DMT . . . . .	10
2.4.1 Slovní zápis algoritmu . . . . .	10
<b>3 Informace o bonusových úlohách</b>	<b>11</b>
3.1 Výběr barevných stupnic . . . . .	11
3.2 Automatický popis vrstevnic . . . . .	14
3.3 Algoritmus pro automatické generování terénních tvarů . . . . .	15
3.3.1 Kupa . . . . .	15
3.3.2 Údolí . . . . .	15
3.3.3 Vodorovný hřbet . . . . .	16
3.3.4 Sedlo . . . . .	17
3.3.5 Spočinek . . . . .	19
<b>4 Vstupní data</b>	<b>21</b>
<b>5 Výstupní data</b>	<b>21</b>
<b>6 Ukázka aplikace</b>	<b>22</b>
<b>7 Technická dokumentace</b>	<b>27</b>
7.1 Třídy . . . . .	27
7.1.1 Algorithms . . . . .	27
7.1.2 Generator . . . . .	29
7.1.3 Draw . . . . .	29
7.1.4 SortbyX . . . . .	31
7.1.5 Edge . . . . .	32
7.1.6 QPoint3D . . . . .	32
7.1.7 Triangle . . . . .	33
7.1.8 Widget . . . . .	33
<b>Diskuse</b>	<b>35</b>
<b>Závěr</b>	<b>37</b>

# Seznam obrázků

1	Nepravidelná trojúhelníková síť (TIN) s vytvořenými vrstevnicemi . . . . .	8
2	Analýza sklonu svahu poskytovaná aplikací Analýzy výškopisu, jež spravuje Zeměměřický úřad . . . . .	10
3	Analýza expozice svahu poskytovaná aplikací Analýzy výškopisu, jež spravuje Zeměměřický úřad . . . . .	11
4	Analýza sklonu nad spočinkem . . . . .	12
5	Stupňové intervaly v mapové aplikaci Analýzy výškopisu při analýze sklonu terénu . . . . .	12
6	Intervaly expozice terénu v softwaru ArcGIS . . . . .	13
7	Stupňové intervaly v mapové aplikaci Analýzy výškopisu při analýze orientace terénu . . . . .	13
8	Analýza expozice nad spočinkem . . . . .	14
9	Vygenerovaná kupa s vrstevnicemi . . . . .	16
10	Vygenerovaný vodorovný hřbet s vrstevnicemi . . . . .	17
11	Vygenerované sedlo s vrstevnicemi . . . . .	18
12	Vygenerovaný spočinek s vrstevnicemi . . . . .	19
13	Vygenerovaná mřížka po analýze sklonu . . . . .	20
14	Vygenerovaná mřížka po analýze expozice . . . . .	20
15	Ukázka Delaunay modelu nad body zadanými kursorem myši . . . . .	21
16	Analýza sklonu nad testovacími geodetickými daty ze souboru . . . . .	22
17	Analýza expozice nad testovacími geodetickými daty ze souboru . . . . .	22
18	Analýza sklonu nad kupou . . . . .	23
19	Analýza expozice nad kupou . . . . .	24
20	Analýza sklonu nad sedlem . . . . .	25
21	Analýza expozice nad sedlem . . . . .	25
22	Analýza sklonu nad údolím . . . . .	26
23	Analýza expozice nad údolím . . . . .	26



# 1 Popis a rozbor problému

Cílem úlohy bylo vytvořit model terénu takový, který na podkladě trojúhelníkové Delaunayho sítě umožní tvorbu vrstevnic, sklonu terénu a expozici.

Trojúhelníková síť je základem pro každý digitální model terénu. Nejčastěji se všechny analýzy snaží najít rovnostranné trojúhelníky tak, aby co nejlépe odpovídaly terénu, jenž mají znázorňovat. Jsou však metody, které dbají na uchování důležitých hran a jejich trojúhelníky mají nejrůznější úhly.

Triangulace, v dnešní době však slouží při mnohem vícero věcích, než jen u kartografie. Například na základě triangulace se určuje na mobilech otisk prstu či známé „face ID“ a existuje mnoho dalších způsobů využití. Triangulací je hned několik, Delaunayho, kterou používáme pro řešení této úlohy je jednou z nejčastějších, ale pak je také Greedy triangulace, minimum Weight triangulace, datově závislé triangulace či Constrained triangulace.

Analýza sklonu terénu je dnes využívána hojně na silnicích. Každý se jistě setkal se značkou, která znázorňuje sklon terénu a varuje řidiče, že je zde terén strmější, či prudší. Stejné sklony se používají i u znázornění terénu.

Analýza expozice je dnes hojně využívána nejen při volbě míst, kam sázet, ale také třeba v umění. Když umělec nakreslí kvádr tak, že jdou vidět jen tři strany a vystíní ho, ukazuje nám, jak na něj dopadá světlo. Jedna strana světlá, druhá tmavší a třetí nejtmavší. Stejně jako terén, na kopce nám krásně svítí, ale do nějakých úžin nám téměř nesvítí.

## 1.1 Údaje o bonusových úlohách

V rámci úlohy byly řešeny tři bonusové úlohy:

- Výběr barevných stupnic při vizualizaci sklonu a expozice
- Automatický popis vrstevnic
- Algoritmus pro automatické generování terénních tvarů



## 2 Popis použitých algoritmů

Obecně lze metody triangulace rozdělit podle více aspektů. Jedním aspektem je dělení triangulací dle geometrické konstrukce. Mezi tyto triangulace můžeme zařadit Greedy triangulace, Delaunay triangulace, Minimum Weight triangulace, Constrained triangulace (triangulace s povinnými hranami) a datově závislé triangulace. Dále můžeme triangulace dělit podle použitých kritérií, tedy na lokálně optimální triangulace, globálně optimální triangulace a multikriteriálně optimalizované triangulace.

V této úloze byla naprogramována Delaunay triangulace pomocí metody inkrementální konstrukce. Další metody přímé konstrukce Delaunayho triangulace jsou např. metoda lokálního prohazování, inkrementálního vkládání, Divide and Conquer (rozděl a panuj) či Sweep Line (zametací přímka). Pro nepřímou konstrukci se využívají Voronoi diagramy, v praxi však tento způsob není používaný. Inkrementální konstrukce je vhodná pro vstupní množiny do 1 milionu bodů, pro větší množiny (do 50 milionů) se používá inkrementální vkládání. Pro ještě větší množiny je nutné použít metodu Divide and Conquer, která je však hodně náročná zhlediska implementace.

### 2.1 Delaunayho triangulace

Je metoda pro vytvoření trojúhelníkové sítě, jakožto digitálního modelu terénu a lze ji využít v rovině i v prostoru. Metoda funguje na principu hledání minimální opsané kružnice vedoucí bodem, nevhodnějším k již vytvořené orientované hraně a to v její levé polorovině. Pokud takový bod existuje, změníme orientaci hrany a pokračujeme dále v hledání dalších bodů.

Celý algoritmus má seznam hran – Active Edge List (AEL), ke kterým ještě nebyl nalezen třetí bod. Pokud nalezneme novou hranu, musíme se přesvědčit, zdali hrana s opačnou orientací již v seznamu není. Pokud ne, vložíme nově nalezenou hranu a hledáme další, pokud už je, hrana se nevloží. A pokud k nějaké hraně z AEL nalezneme třetí bod, hranu ze seznamu odstraníme. Celý algoritmus trvá, dokud seznam AEL není prázdný.

Delaunayho triangulace musí splňovat následující vlastnosti:

1. Uvnitř opsané kružnice libovolného trojúhelníku triangulace neleží žádný jiný bod.
2. Maximalizuje minimální úhel v trojúhelníku.
3. Vůči kritériu minimálního úhlu je lokálně i globálně optimální.
4. Triangulace je jednoznačná, pokud čtyři body neleží na kružnici.

### 2.1.1 Slovní zápis algoritmu

1. Nalezení pivota  $q$  s minimální souřadnicí  $X$ :  $q = \min(x_i)$
2. Hledání nejbližšího bodu k pivotovi:  $\|p_1 - q\|_2 = \min$
3. Vytvoření první hrany:  $e = (q, p_1)$
4. Hledání Delaunay bodu:  $\underline{p} = \operatorname{argmin}_{\forall p_i \in \sigma_L(e)} r'(k_i); k_i = (a, b, p_i); e = (a, b)$   
Při nenalezení Delaunay bodu změna orientace:  $\#p : e \leftarrow (p_1, q)$ ; go to 4.
5. Vytvoření zbývajících hran trojúhelníka:  $e_2 = (p_1, \underline{p}); e_3 = (\underline{p}, q)$
6. Přidání hran trojúhelníka do DT:  $DT \leftarrow e; DT \leftarrow e_2; DT \leftarrow e_3$
7. Přidání hran trojúhelníka do AEL (active edges list):  $AEL \leftarrow e; AEL \leftarrow e_2; AEL \leftarrow e_3$
8. Dokud není AEL prázdný proved':

Hledání Delaunay bodu k hraně z AEL (viz 4).

Pokud Delaunay bod existuje. *if*  $\exists \underline{p}$

Vytvoření zbývajících hran trojúhelníku:  $e_2 = (p_1, \underline{p}); e_3 = (\underline{p}, q)$

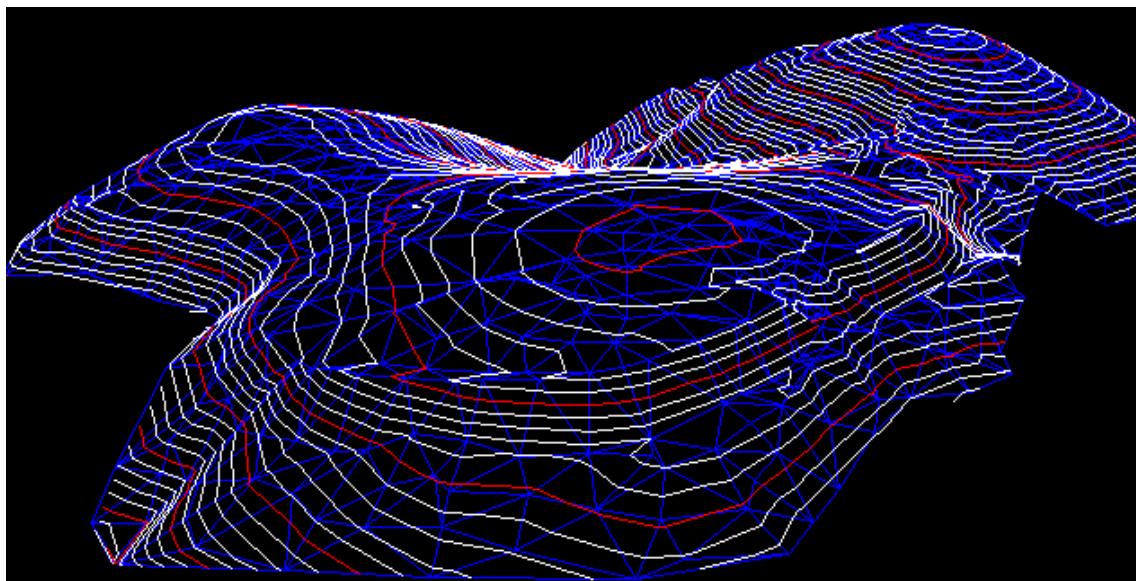
Pokud nová hrana není v AEL, přidej.

## 2.2 Konstrukce vrstevnic

Vrstevnici můžeme označit jako křivku, spojující body o stejně nadmořské výšce. Jinými slovy jde o křivku, která vznikne protnutím terénu vodorovnými rovinami. Vrstevnice dělíme na základní, které jsou ve vzdálenosti intervalu vrstevnic. Dále pak Hlavní nebo také Zvýrazněné, což bývá zpravidla každá 5. základní vrstevnice. Pak jsou vrstevnice doplňkové, které se nachází v různých částech intervalu, nejčastěji polovina či čtvrtina a nakonec vrstevnice pomocné, které slouží v místech, kde potřebujeme lépe naznačit terén a ostatní vrstevnice by tomu nestačily (lomy, sesuvy...).

Vrstevnice, jejich vzhled i popis je dán skrze kartografické zásady. Základní vrstevnice kreslíme tenkou čarou, hlavní pak silnější čarou a doplňkové čárkovanou čarou. Jejich popis by měl být orientován tak, že vršek textu je směřován do kopce a naopak spodek textu je směřován z kopce. Popis bývá většinou barevný a silný, jako vrstevnice, kterou označuje a je potřeba aby vrstevnice byly popisovány rovnoměrně po celé zmapované oblasti. Nejčastěji se popisují pouze hlavní vrstevnice.

Vrstevnice lze tvořit dvěma způsoby. První z nich je za pomoci protínání trojúhelníkové sítě vodorovnými rovinami. Druhý pak za pomoci lineární interpolace, kdy je rozestup mezi dvěma body konstantní a tedy i jejich spád. Tato metoda byla použita při našem řešení.



Obrázek 1: Nepravidelná trojúhelníková síť (TIN) s vytvořenými vrstevnicemi [2]

### 2.2.1 Slovní zápis algoritmu

1. Pro všechny hrany trojúhelníku  $t$ :  $\forall e_i \in t : \text{najdi průsečíky roviny a hrany.}$   
Může nastat několik případů:
  2. Trojúhelník je celý v rovině vrstevnice - žádná vrstevnice se nekreslí.
  3. Hrana náleží rovině vrstevnice  $z$ :  $(z - z_i) \cdot (z - z_{i+1}) = 0 \rightarrow e_i \in \rho$  (vrstevnice se vykreslí v příslušné hraně).
  4. Hrana je průnikem roviny vrstevnice  $z$ :  $(z - z_i) \cdot (z - z_{i+1}) < 0 \rightarrow e_i \cap \rho$  (spočtou se průsečíky a ty se spojí vrstevnicovou hranou).

Výpočet polohových souřadnic:

$$x = \frac{(x_2 - x_1)}{(z_2 - z_1)}(z - z_1) + x_1$$
$$y = \frac{(y_2 - y_1)}{(z_2 - z_1)}(z - z_1) + y_1$$

Vytvoř hranu tvořící vrstevnici.

## 2.3 Analýza sklonu DMT

Sklon nám udává, zdali nám daný terén klesá, stoupá či zůstává neměnný. Máme-li dva normálové vektory  $(0,0,1)$  a normálový vektor roviny trojúhelníku z trojúhelníkové sítě, pak úhel mezi těmito normálovými vektory definuje sklon terénu.

### 2.3.1 Slovní zápis algoritmu

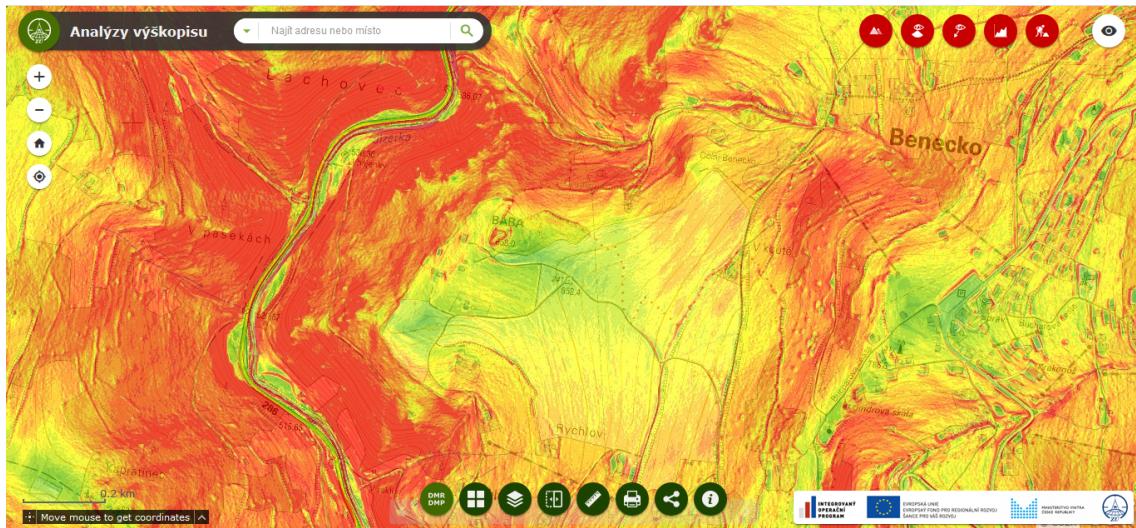
1. Pro všechny trojúhelníky triangulace:  $\forall t_i \in DT :$
2. Výpočet normálového vektoru roviny trojúhelníku:

$$u_x = \Delta x_2, x_1; u_y = \Delta y_2, y_1; u_z = \Delta z_2, z_1;$$

$$v_x = \Delta x_2, x_3; v_y = \Delta y_2, y_3; v_z = \Delta z_2, z_3;$$

$$n_t = (u_y \cdot v_z - u_z \cdot v_y)^2 - (u_x \cdot v_z - u_z \cdot v_x)^2 + (u_x \cdot v_y - u_y \cdot v_x)^2$$

3. Výpočet sklonu:  $\varphi = \arccos \frac{n_z}{|n_t|}$



Obrázek 2: Analýza sklonu svahu poskytovaná aplikací Analýzy výškopisu, jež spravuje Zeměměřický úřad [1]

## 2.4 Analýza expozice DMT

Expozice nám označuje orientaci terénu vzhledem ke světlu, tedy vzhledem k slunci. Jinými slovy nám říká, na jakou část terénu dopadá kolik světla, kde je světla do statek a kde je naopak stín.

Pokud máme rovinu určenou souřadnicovými osami x a y a promítneme do ní normálový vektor roviny trojúhelníku, pak azimut tohoto průmětu definuje expozici terénu.

### 2.4.1 Slovní zápis algoritmu

1. Pro všechny trojúhelníky triangulace:  $\forall t_i \in DT :$

2. Výpočet x a y části normálového vektoru:

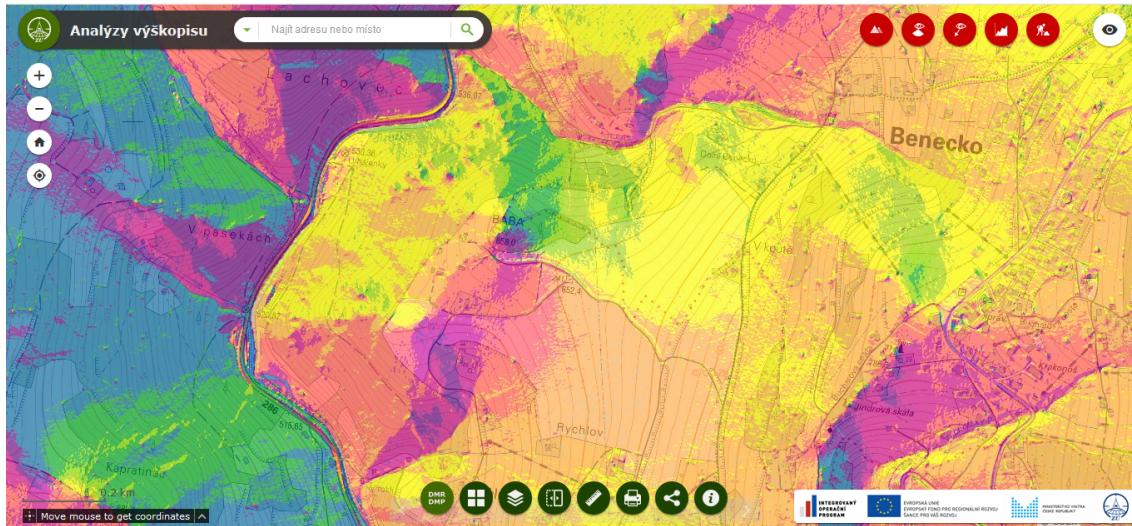
$$u_x = \Delta x_2, x_1; u_y = \Delta y_2, y_1; u_z = \Delta z_2, z_1;$$

$$v_x = \Delta x_2, x_3; v_y = \Delta y_2, y_3; v_z = \Delta z_2, z_3;$$

$$n_x = (u_y \cdot v_z - u_z \cdot v_y)$$

$$n_y = -(u_x \cdot v_z - u_z \cdot v_x)$$

3. Výpočet expozice:  $A = \text{atan}2\left(\frac{n_x}{n_y}\right)$



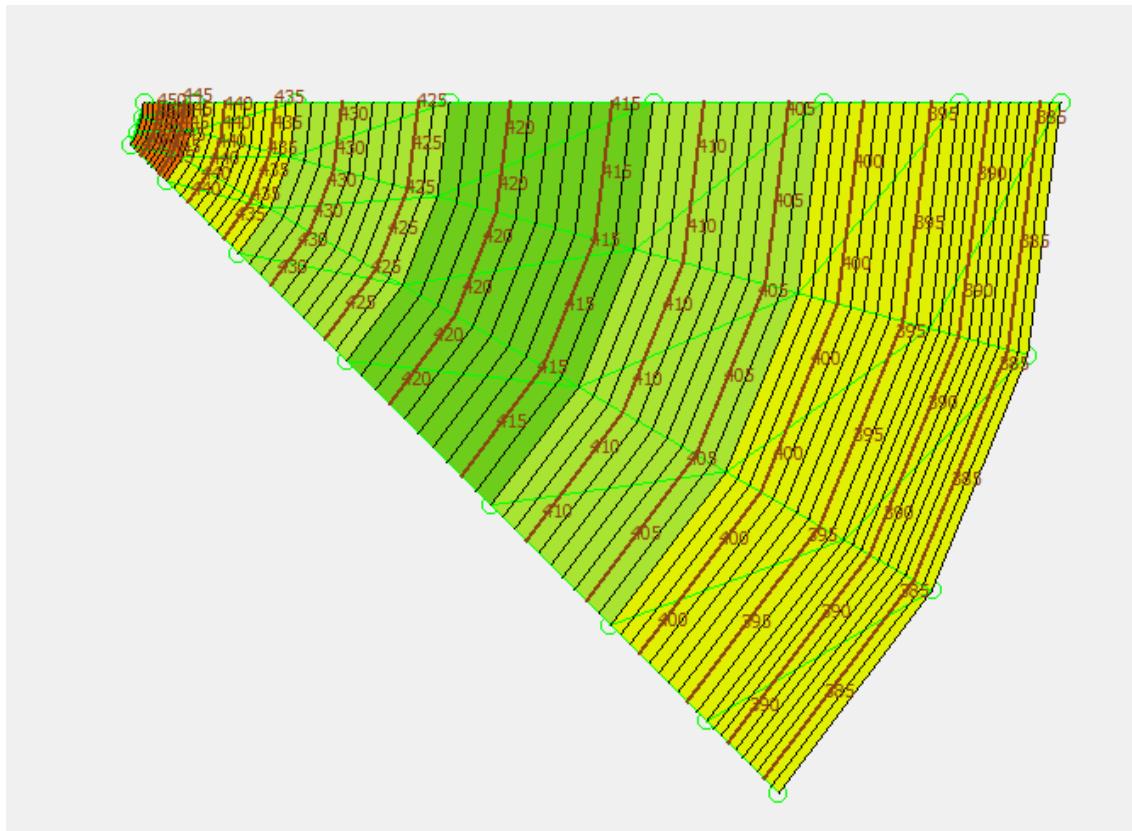
Obrázek 3: Analýza expozice svahu poskytovaná aplikací Analýzy výškopisu, jež spravuje Zeměměřický úřad [1]

## 3 Informace o bonusových úlohách

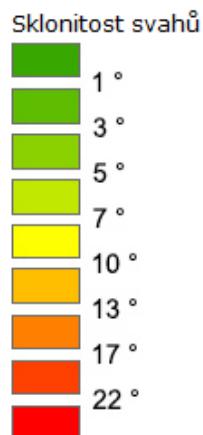
### 3.1 Výběr barevných stupnic

V rámci této bonusové úlohy byly definovány barevné stupnice pro zobrazení sklonu a expozice. Pro sklon byly zvoleny odstíny barev od zelené (téměř rovina) po červenou (strmý svah). Inspirací pro volbu této barevné škály byla aplikace Analýzy výškopisu, jejímž provozovatelem je Zeměměřický úřad.

Stupňové intervaly byly rovněž voleny podle schémata použitého ve výše zmíněné aplikaci při analýze *Slope*, viz obrázek 5. Aby byly dobře vidět i vrstevnice, bylo pracováno s průhledností. Složky RGB barev byly namíchány na webové stránce <https://www.colorsphere.com/rgb-color-wheel/>. Samotné vytvoření barevných škál bylo implementováno ve třídě Draw, kde byly procházeny jednotlivé trojúhelníky Delaunayho triangulace a ke každému trojúhelníku podle spočtené analýzy byla přiřazena konkrétní barva. Výběr barevných stupnic sklonu je nejlépe viditelný při analýzách nad generovanými terénními tvary. Na obrázku 4 je zobrazena analýza sklonu nad spočinkem.

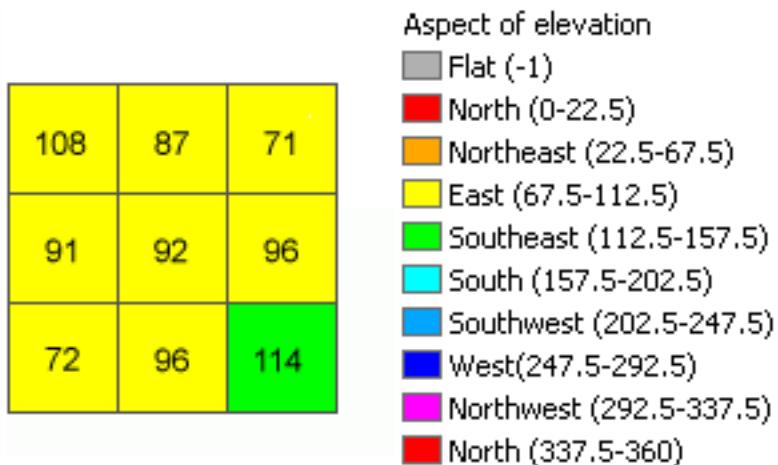


Obrázek 4: Analýza sklonu nad spočinkem



Obrázek 5: Stupňové intervaly v mapové aplikaci Analýzy výškopisu při analýze sklonu terénu

Pro vyznačení expozice byly voleny sytější barvy, která na sebe navazují. Inspirácií pro stupňové intervaly byla barevná stupnice vytvořená pro analýzu *Aspect* v programu ArcGIS, jež je znázorněna na následujícím obrázku 6.



Obrázek 6: Intervaly expozice terénu v softwaru ArcGIS [3]

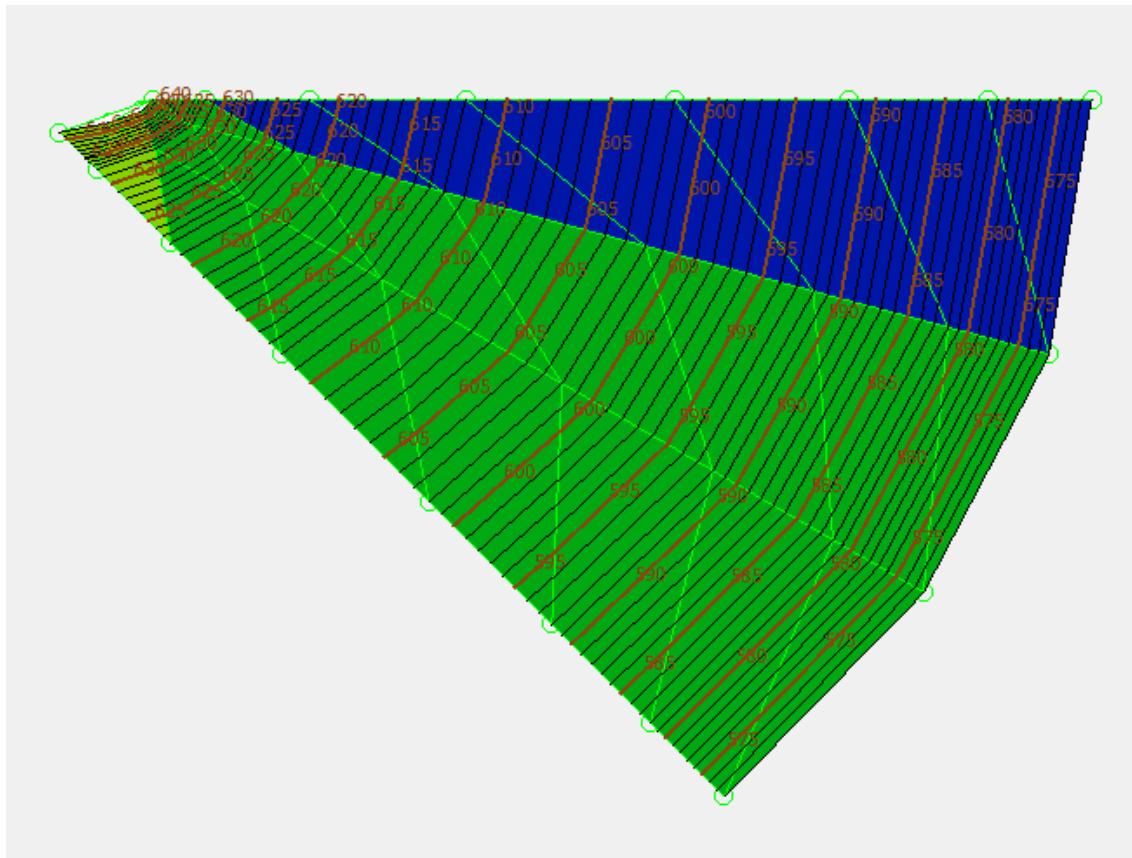
Barevní paleta byla volena podle schémata použitého v aplikaci Analýzy výškopisu, viz obrázek 7. Rovina nebyla uvažována.

#### Orientace svahů vůči světovým stranám

	Sever
	Severovýchod
	Východ
	Jihovýchod
	Jih
	Jihozápad
	Západ
	Severozápad

Obrázek 7: Stupňové intervaly v mapové aplikaci Analýzy výškopisu při analýze orientace terénu

Na obrázku 8 můžeme vidět analýzu sklonu nad vygenerovaným spočinkem. Sklon se nejdříve snižuje a poté zase zvětšuje. Je patrné, že velká část spočinku se nachází v severovýchodním a východním směru.



Obrázek 8: Analýza expozice nad spočinkem

### 3.2 Automatický popis vrstevnic

Automatický popis vrstevnic byl vytvořen pro hlavní vrstevnice (každá pátá). Důležitým krokem bylo pozměnit funkci *CreateContourLines*, aby nevracela pouze vektor hran vrstevnic, ale i vektor jednotlivých výšek. Tyto výšky pak byly pomocí funkce *SetMainContours* převedeny do třídy *Draw*, ve které byly využity přímo při vykreslování hlavních vrstevnic.

Text byl umístěn do průměrných souřadnic mezi počátkem a koncem každé hrany hlavní vrstevnice. Pro umístění popisu byla použita funkce *drawText*. Další vylepšení popisu vrstevnic, jako například natočení vzhledem k hraně, popis směřující hlavou do kopce, rovnoměrné rozložení kót po vrstevnici, či vyneschání vrstevnice v místě popisu, řešeno nebylo. Popis hlavních vrstevnic je patrný z obrázku 8.

### 3.3 Algoritmus pro automatické generování terénních tvarů

V rámci bonusové úlohy byly vytvořeny algoritmy na automatické generování terénních tvarů. Konkrétně byly navrženy algoritmy na generování kupy, údolí, vodorovného hřbetu, sedla a spočinku. Navíc byla implementována možnost generování pravidelného gridu, jelikož takové rozmístění bodů má za následek nejednoznačnost DT, kdy žádné čtyři body nesmí ležet na kružnici. Popis jednotlivých útvarů byl použit ze středoškolských zápisek z předmětu Mapování.

#### 3.3.1 Kupa

Jedná se o výrazně zaoblený vypuklý tvar. Povrch terénu klesá od vrcholu směrem na všechny strany. Tvarovou čárou je uzavřená křivka, která ohraničuje její vrchol a má kruhový, eliptický nebo nepravidelný tvar. V rámci naší aplikace je pro kupu zvoleno pojmenování **Hill**.

Kupa byla generována pomocí několika elips, jejichž střed je vrcholem kupy. Elipsa s větším poloměrem má menší výšku Z, pro klesání byl zvolen krok 20. Vygenerovaná kupa je zobrazena na obrázku 9.

$$X = X_0 + a * \cos(\phi)$$

$$Y = Y_0 + b * \sin(\phi)$$

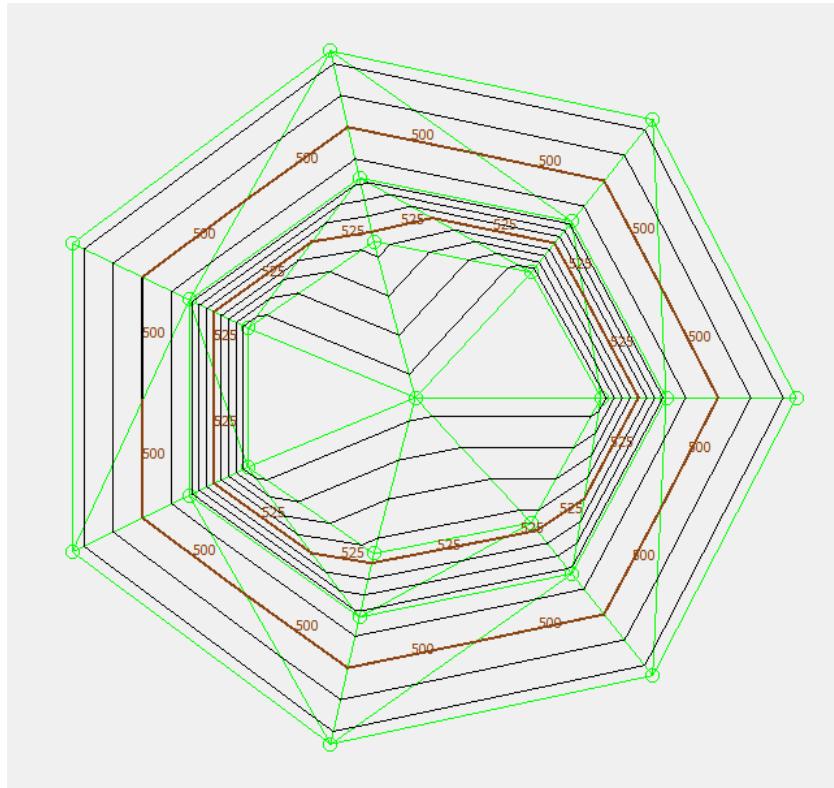
$$Z = Z_0 - j * 20$$

#### 3.3.2 Údolí

Jedná se o terénní útvar tvořící protáhlou vhloubenou plochu, která je tvořena působením vody nebo ledovci. V rámci aplikace je použito označení **Valley**.

Pro údolí byly vygenerovány přímky od nichž se v zadaném kroku vytváří nové přímky s konstantně stoupající hodnotou výšky. V uvedených vzorcích jsou p1 a p2 označení pro vzniklé přímky.

$$X_{p1} = X_0 + a$$



Obrázek 9: Vygenerovaná kupa s vrstevnicemi

$$Y_{p1} = Y_0 + i * 10$$

$$Z_{p1} = Z_0 + a$$

$$X_{p2} = X_0 - a$$

$$Y_{p2} = Y_0 + i * 10$$

$$Z_{p2} = Z_0 + a$$

### 3.3.3 Vodorovný hřbet

Jedná se o protáhlý vypuklý terénní tvar, jehož vrcholová část je zaoblená. V aplikaci je pojmenován jako **Ridge**. Hřbet byl generován podobným způsobem jako údolí, s tím rozdílem, že výška postupně nestoupá, ale klesá. Vygenerovaný vodorovný hřbet je zobrazen na obrázku 10.

$$X_{p1} = X_0 + a$$

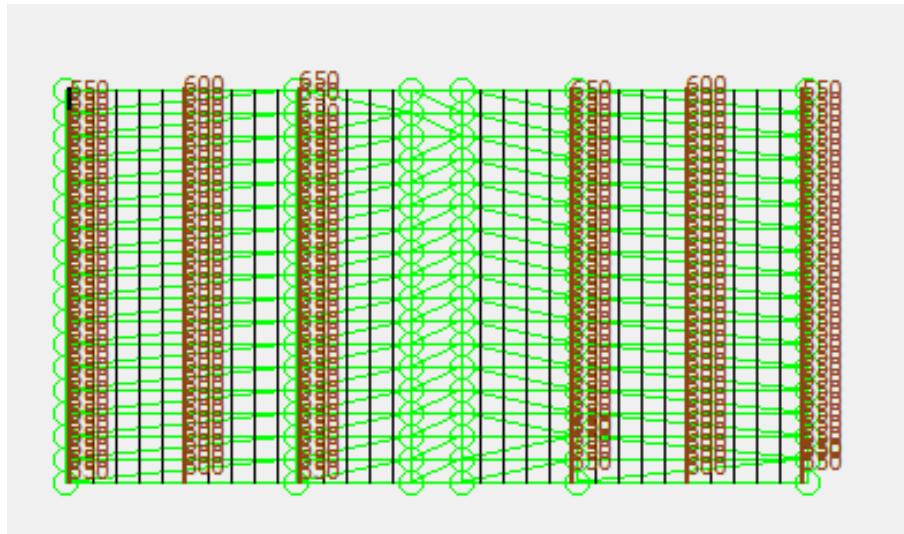
$$Y_{p1} = Y_0 + i * 10$$

$$Z_{p1} = Z_0 - a$$

$$X_{p2} = X_0 - a$$

$$Y_{p2} = Y_0 + i * 10$$

$$Z_{p2} = Z_0 - a$$



Obrázek 10: Vygenerovaný vodorovný hřbet s vrstevnicemi

### 3.3.4 Sedlo

Tento terénní útvar označuje nejnižší plochu mezi dvěma terénními tvary, např. kupami. Stýkají se v něm dvě hřbetnice a dvě údolnice. Vbíhají do něj dvě plochy vypuklé s ubývajícím spádem kolmo na hřbetnici a vybíhají z něj dvě plochy vhloubené s přibývajícím spádem. V aplikaci je použito označení **Saddle**.

Generování sedla zahrnovalo vygenerování dvou kupy a dvou prohlubní. Podstatné bylo rozmístění. Vlevo a vpravo byly generovány kupy, nahoře a dole prohlubně. Středy těchto terénních tvarů byly zvoleny napevno. Následně byly kolem těchto středů generovány elipsy s klesající/rostoucí výškou. Vygenerované sedlo je zobrazeno na obrázku 11. Můžeme si všimnout, že zvláště u vrcholů a prohlubní jsou výsledné vrstevnice generovány chybně. Mohla by pomoci možnost vložení povinných hran.

Tvorba elips u kupy:

$$X = X_0 + i * \text{rand\%}(70) * \cos(j * \phi) + \text{rand\%}(20)$$

$$Y = Y_0 + i * \text{rand\%}(70) * \sin(j * \phi) + \text{rand\%}(20)$$

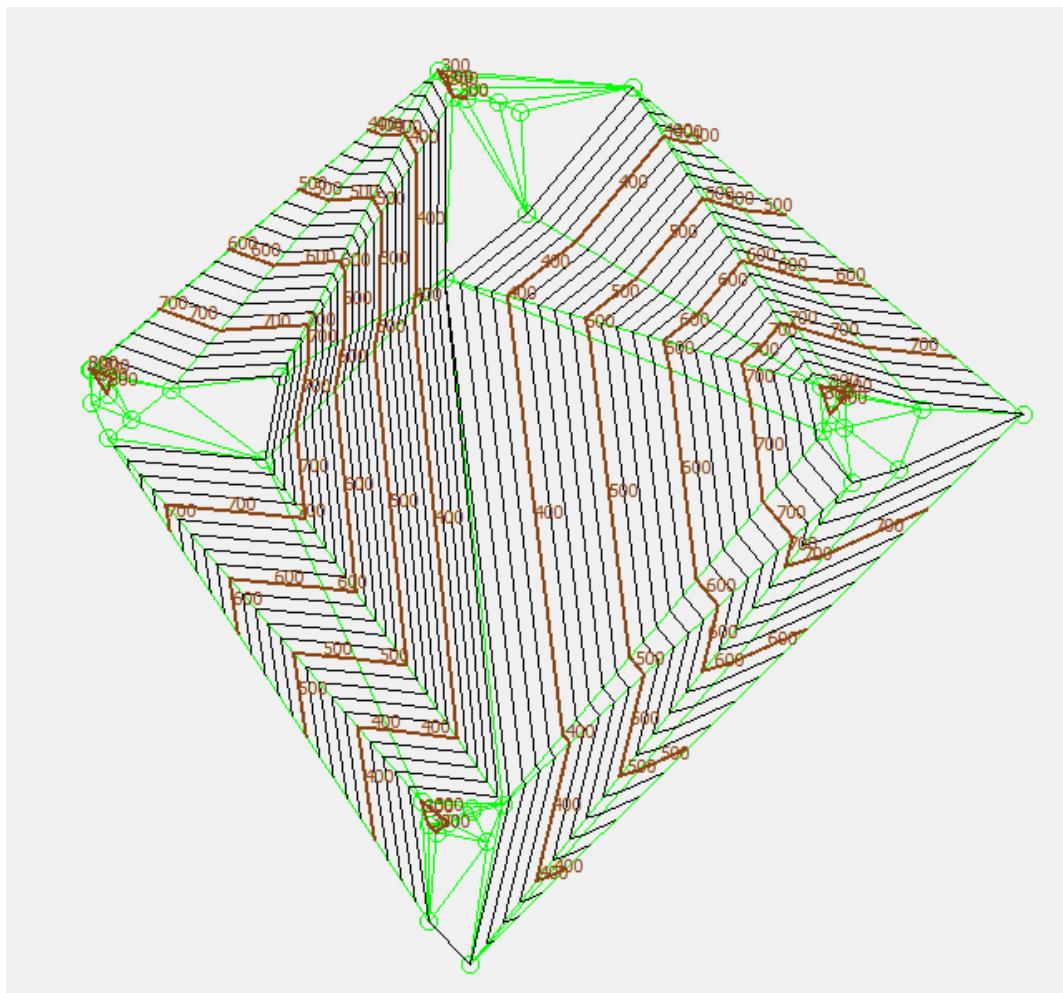
$$Z = Z_0 - i * 50$$

Tvorba elips u prohlubně:

$$X = X_0 + i * \text{rand\%}(70) * \cos(j * \phi) + \text{rand\%}(20)$$

$$Y = Y_0 + i * \text{rand\%}(70) * \sin(j * \phi) + \text{rand\%}(20)$$

$$Z = Z_0 + i * 50$$

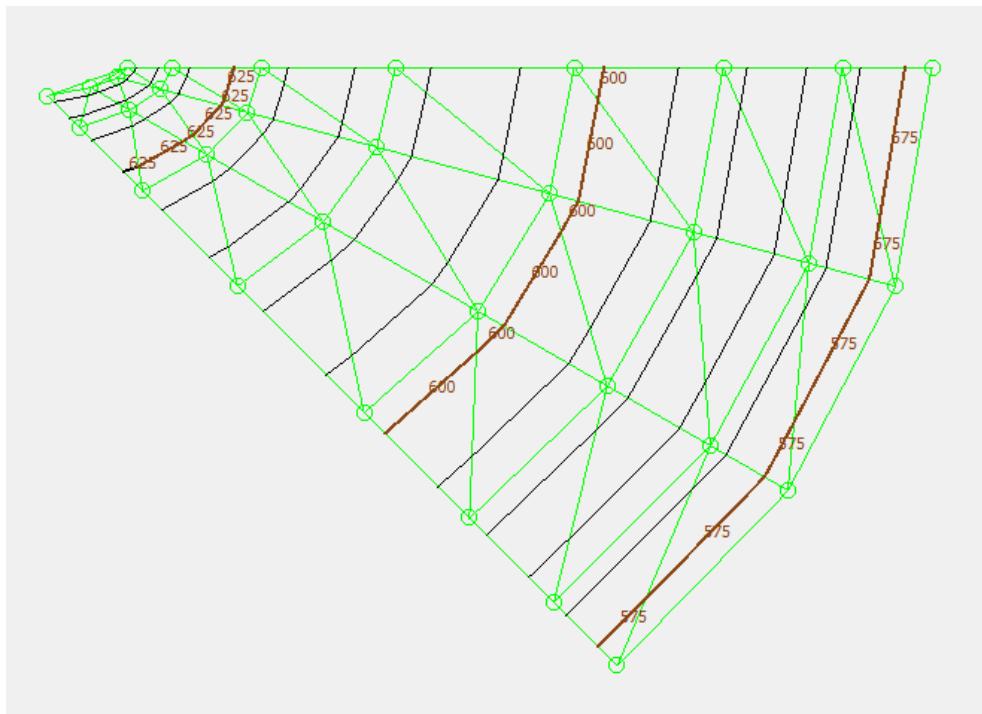


Obrázek 11: Vygenerované sedlo s vrstevnicemi

### 3.3.5 Spočinek

Tento název označuje část hřbetu vyvýšeniny, kde hřbetnice přechází do mírnějšího sklonu, tj. plocha spočinku je oproti zbytku svahu méně skloněná nebo i vodorovná. tvarová čára se skládá ze dvou křivek – jedna ohraničuje vbíhající vypuklou plochu, druhá ohraničuje vybíhající plochu. Rozestupy ve směru spádu se nejprve plynou zvětšují a pak zase změňují. V aplikaci lze spočinek vygenerovat pod názvem **Gentle Ridge**.

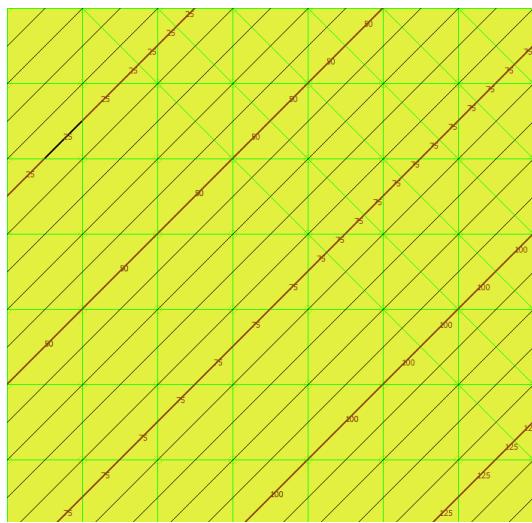
Algoritmus na tvorbu spočinku je nejsložitější. Rozmezí úhlů, v kterých se generují elipsy je omezeno, tak aby výsledný útvar vytvářel šedesáti-stupňový vějíř. Je náhodně vygenerován vrchol, od kterého se pak vytváří části elips. Polovina elips je tvořena se zvyšujícím se poloměrem, tedy rozestupy vrstevnic se plynou zvětšují. Další polovina vrstevnic je tvořena se snižujícím se poloměrem, tedy rozestupy ve směru spádu se plynule zmenšují. Počet elips je pevně daný, generování vrcholu a počtu bodů v elipsách je náhodné. Výškové rozdíly mezi jednotlivými elipsami jsou stejné, co se mění je pouze rozestup mezi nimi. Vygenerovaný spočinek je zobrazen na obrázku 12.



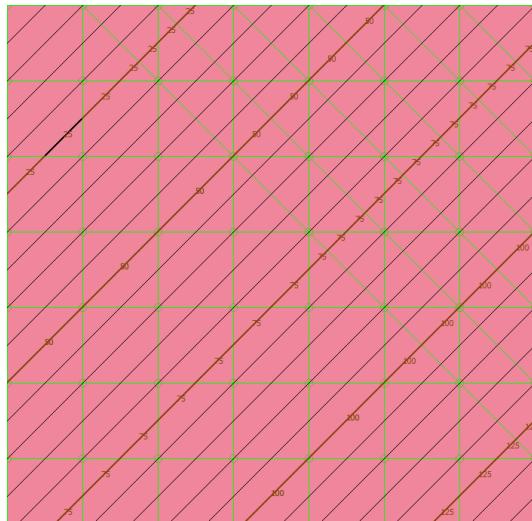
Obrázek 12: Vygenerovaný spočinek s vrstevnicemi

## Grid

Aby bylo možné dobře demonstrovat nejednoznačnost DT, byla naprogramována tvorba čtvercové mřížky. Nejeznačnost je však dobře patrná i u analýz nad hřbetem nebo údolím v nejvyšší resp. nejnižší části. Na obrázcích 13, 14 je vidět nejednoznačnost hledání nejmenší opsané kružnice. Proto se může stát, že výsledné výpočty nebudou správné. V tomto případě však výsledné analýzy vycházejí správně.



Obrázek 13: Vygenerovaná mřížka po analýze sklonu



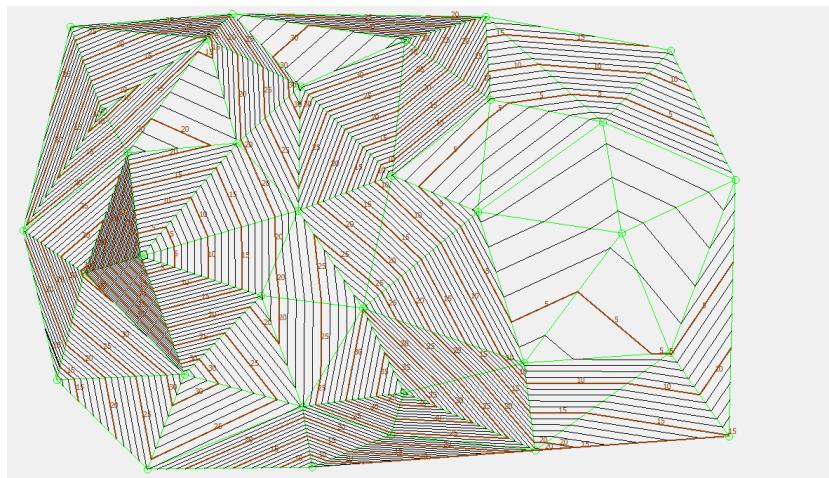
Obrázek 14: Vygenerovaná mřížka po analýze expozice

## 4 Vstupní data

Vstupní data tvoří analyzovaná skupina bodů datového typu  $std :: vector < QPoint3D >$ , která je vygenerována po stisku tlačítka *Generate Points*. Tento generátor vytváří pomocí funkce *rand* množiny významných terénních útvarů s předem definovaným počtem vrcholů. Druhým možným vstupem je zadání bodů kursorem myši (viz obrázek 15). Třetím možným vstupem je import skupiny bodů datového typu  $std :: vector < QPoint3D >$  ze souboru.

Vzor vstupních dat YXYZ ve formátu .txt je:

```
845562.622 998320.147 827.354  
845560.329 998321.145 826.989  
845561.871 998330.141 825.856  
845562.564 998329.139 827.546
```



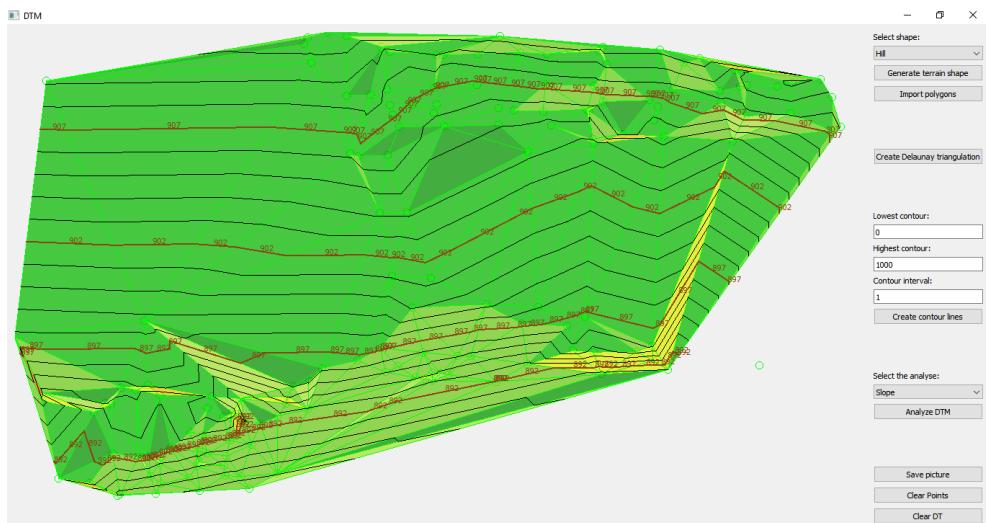
Obrázek 15: Ukázka Delaunay modelu nad body zadanými kursorem myši

## 5 Výstupní data

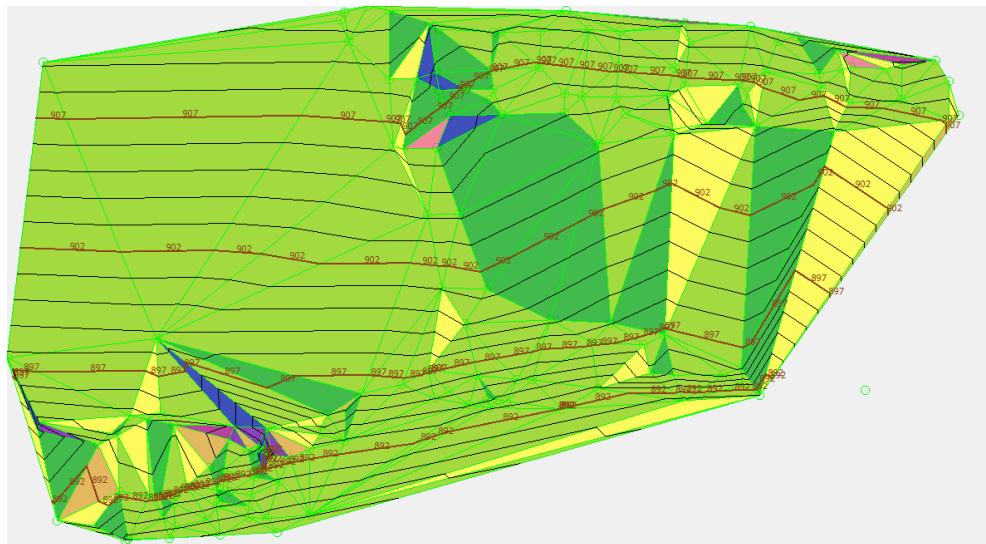
Výstupem je grafická aplikace, která na vstupních datech vytvoří Delaunayho triangulaci. Nad touto triangulací je pak možné vykreslit vrstevnice s automatickým popisem hlavních vrstevnic a analyzovat sklon či expozici terénu.

## 6 Ukázka aplikace

Do této kapitolky je zahrnuto několik ukázků aplikace a výstupů, které lze přímo v aplikaci uložit jako obrázek. V pravé části aplikace může být méně rozestup vrstevnic. V pravé horní části aplikace je možné importovat soubor s připravenými souřadnicemi. Analýzy takového souboru jsou zobrazeny na obrázcích 16 a 17.

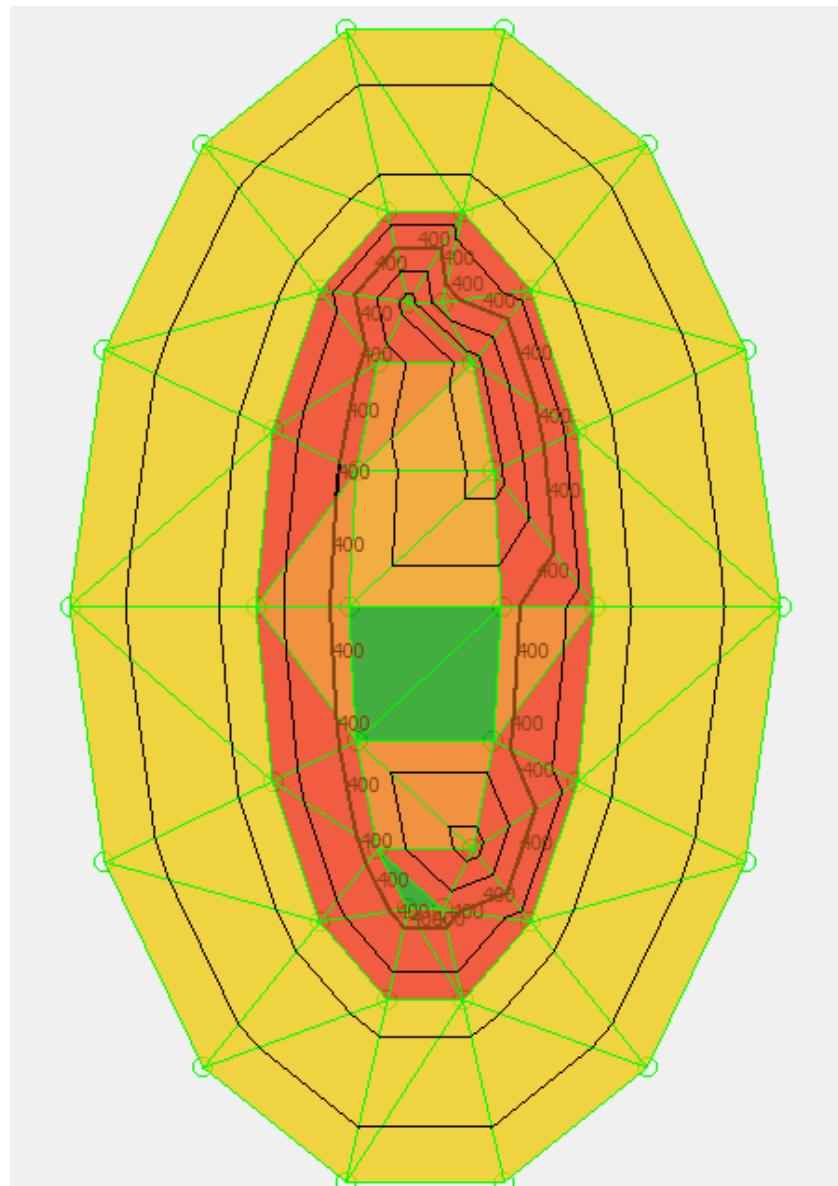


Obrázek 16: Analýza sklonu nad testovacími geodetickými daty ze souboru

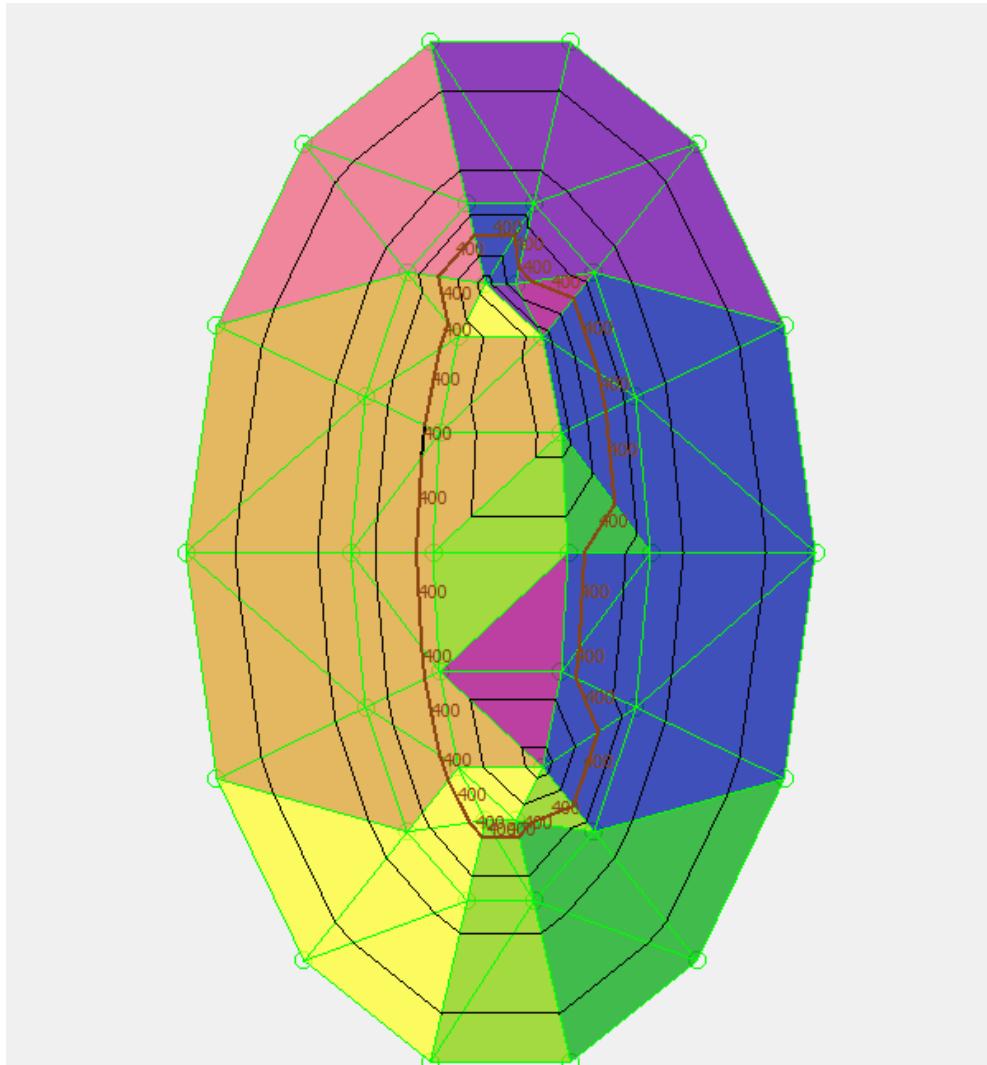


Obrázek 17: Analýza expozice nad testovacími geodetickými daty ze souboru

Následující ukázky jsou věnovány generátoru terénních tvarů, který se nachází v pravé horní části aplikace. Analýzy sklonu a expozice jsou pravděpodobně nejlépe viditelné na spočinku, jehož obrázky jsou obsahem kapitoly 3.3. Analýza expozice je dobře viditelná na obrázku 19.

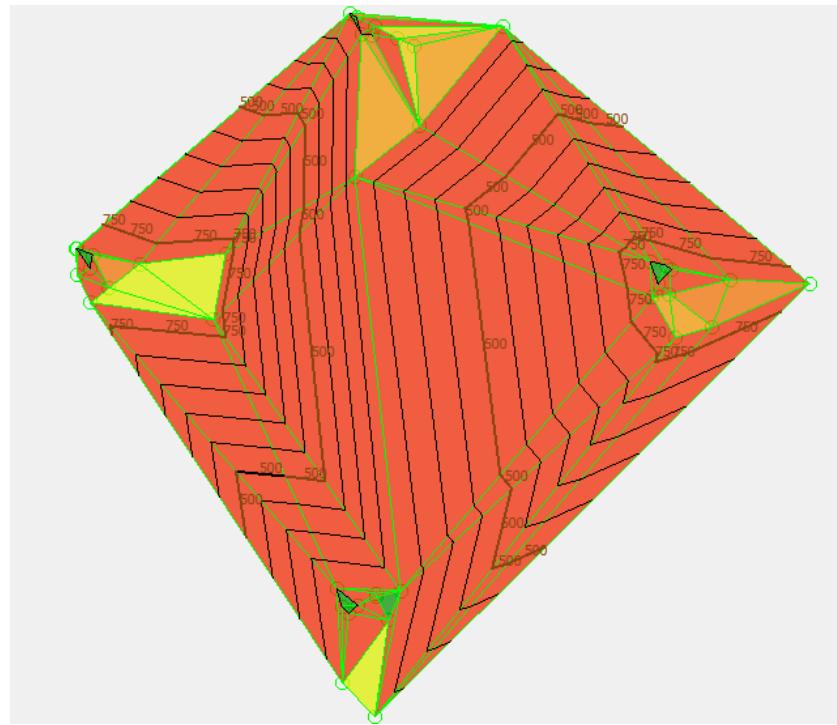


Obrázek 18: Analýza sklonu nad kupou

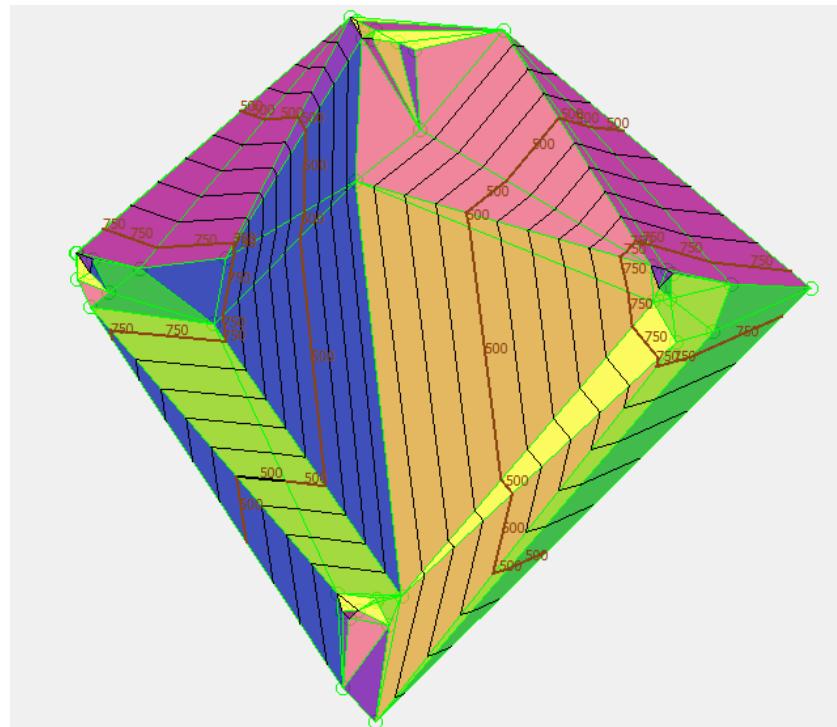


Obrázek 19: Analýza expozice nad kupou

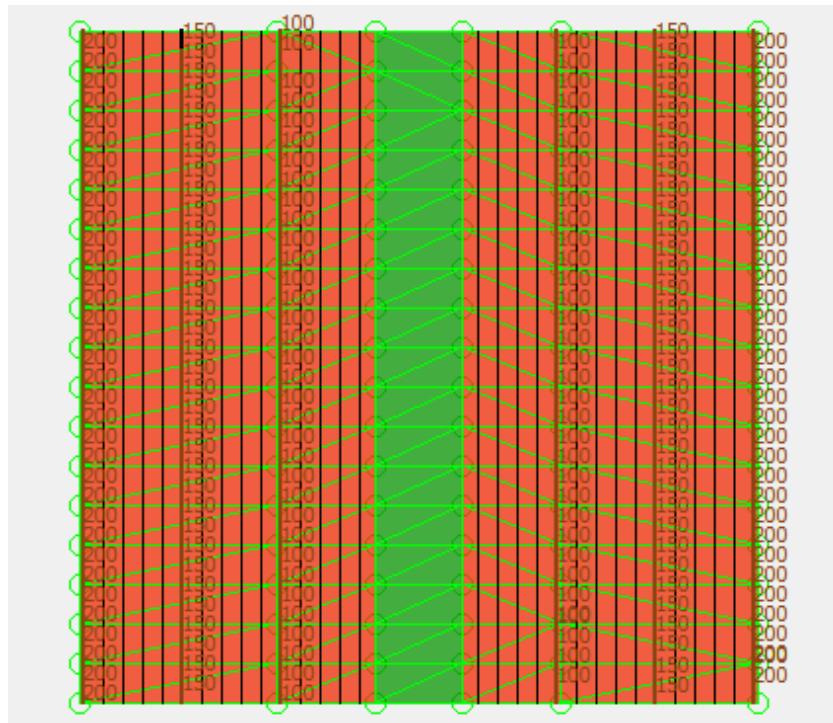
Generování DT u sedla je velmi problematické, tudíž i výsledné analýzy sklonu a expozice nejsou pěkné a to zejména u vrcholových a vhloubených částí. Pokud bychom chtěli výsledek vylepšit, bylo by nutné použít lepší mechanismus na vygenerování DT, která v tomto případě vytváří u vrcholových a vhloubených částí velmi malé, v mnoha případech štíhlé trojúhelníky, která špatně approximují daný tvar. Bylo by vhodné zavést tzv. Constrained Triangulace, které zavádí povinné hrany, jejichž poloha se nesmí měnit.



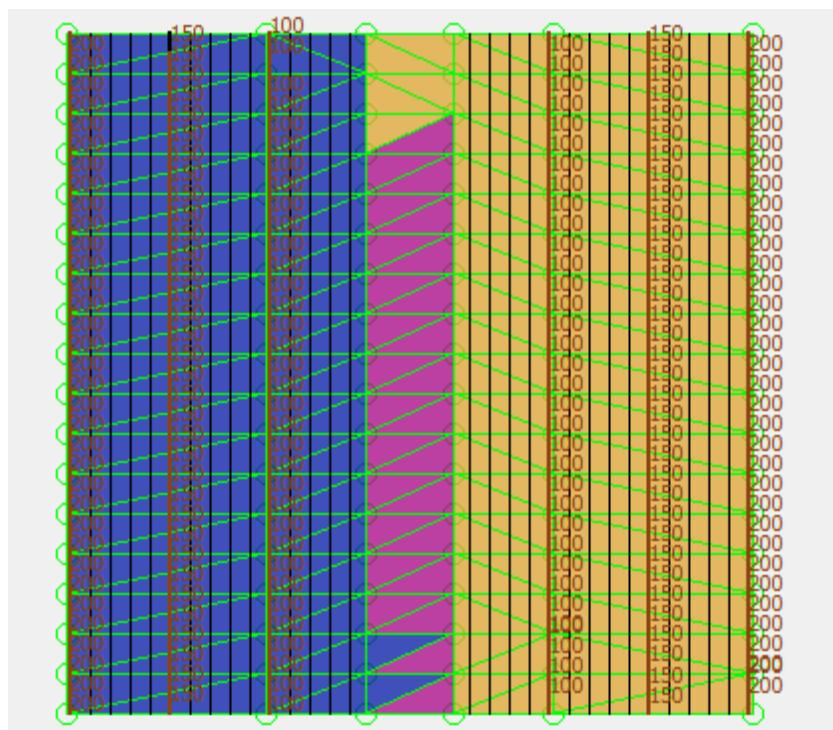
Obrázek 20: Analýza sklonu nad sedlem



Obrázek 21: Analýza expozice nad sedlem



Obrázek 22: Analýza sklonu nad údolím



Obrázek 23: Analýza expozice nad údolím

## 7 Technická dokumentace

### 7.1 Třídy

V aplikaci se nachází celkem sedm tříd - Algorithms, Draw, Generator, SortbyX, Edge, QPoint3D, Triangle a Widget.

#### 7.1.1 Algorithms

Třída Algorithms je nejobsáhlější. Obsahuje konstruktor a dalších několik pomocných i hlavních metod, které jsou určeny pro výpočet algoritmů používaných v digitálním GIS.

**int getPointLinePosition(QPoint3D &q, QPoint3D &p1, QPoint3D &p2)**

Tato funkce má za úkol určit polohu bodu  $q$  vůči přímce zadané dvěma body  $p1$  a  $p2$ . Z vektorů je vypočten determinant. Pokud je determinant větší než 0, bod se nachází v levé polovině a funkce vrací hodnotu 1. Pokud je menší než 0, bod se nachází v pravé polovině a funkce vrací hodnotu 0. Pokud nenastane ani jeden z výše uvedených případů, tedy bod leží na linii, výstupem je hodnota -1.

**double distance2Points(QPoint3D p1, QPoint3D p2)**

Metoda, jejíž návratová hodnota je typu double, vrací velikost spojnice mezi dvěma body.

**double getCircleRadius(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3, QPoint3D &c)**

Metoda jejíž návratová hodnota je typu double, vrací velikost poloměru kružnice tvořené třemi vstupními body.

**int getNearestpoint(QPoint3D &p, std::vector<QPoint3D>&points)**

Tato metoda slouží k nalezení indexu nejbližšího bodu z množiny points k bodu p.

int getDelaunayPoint(QPoint3D &s, QPoint3D &e,

**std::vector<QPoint3D >&points)**

Tato metoda slouží k nalezení indexu bodu, který splňuje Delaunayho vlastnosti.

**std::vector<Edge >DT(std::vector <QPoint3D >&points)**

Metoda vytváří nad vektorem bodů Delaunayho triangulaci, která je reprezentována jako vektor hran.

**QPoint3D getContourPoint(QPoint3D &p1, QPoint3D &p2, double z)**

Metoda vypočte průsečík hrany, která je tvořena 3D body p1 a p2, s rovinou definovanou Z souřadnicí.

**std::vector<Edge >createContourLines(std::vector<Edge >&dt, double z\_min, double z\_max, double dz, std::vector<int >&contour\_heights)**

Metoda z triangulace dt v zadáném intervalu v ose Z  $< z_{min}; z_{max} >$  s intervalem vrstevnic dz vrátí vektor hran definující vrstevnice. Aby bylo možné vytvořit popis vrstevnic, je v rámci této funkce sestavován i vektor výšek typu integer.

**double calculateSlope(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3)**

Tato metoda slouží k vypočetní hodnoty sklonu v trojúhelníku definovaném 3D body p1, p2 a p3.

**double calculateAspect(QPoint3D &p1, QPoint3D &p2, QPoint3D &p3)**

Tato metoda slouží k vypočetní hodnoty expozice v trojúhelníku definovaném 3D body p1, p2 a p3.

**std::vector<Triangle>analyzeDTM(std::vector<Edge >&dt)**

Metoda prochází jednotlivé trojúhelníky Delaunayho triangulace a počítá analýzu sklonu a expozice.



### 7.1.2 Generator

Tato třída obsahuje algoritmy pro automatické generování terénních tvarů.

**std::vector<QPoint3D>generateHill()**

Tvorba kupy.

**std::vector<QPoint3D>generateValley()**

Tvorba údolí.

**std::vector<QPoint3D>generateRidge()**

Tvorba hřbetu.

**std::vector<QPoint3D>generateSaddle()**

Tvorba sedla.

**std::vector<QPoint3D>generateGentleRidge()**

Tvorba spočinku.

**std::vector<QPoint3D>generateGrid()**

Tvorba čtvercového gridu, vstupem je vzdálenost bodů.

### 7.1.3 Draw

Třída Draw obsahuje několik metod. Metody jsou určeny pro generování a vykreslování proměnných.

**void paintEvent(QPaintEvent \*e)**

Nejobsáhlejší metoda sloužící k vykreslení bodů a zobrazení výsledků použitých algoritmů.

**void mousePressEvent(QMouseEvent \*e)**

Metoda uloží bod se souřadnicemi místa kliknutí v zobrazovacím okně a dá tomuto

bodu náhodou souřadnici z.

**void setPoints(std::vector<QPoint3D> &points\_)**

Metoda slouží pro nastavení privátní položky *points*, tak aby byla vidět v třídě Draw.

**std::vector<QPoint3D> & getPoints()**

Metoda slouží pro vrácení privátní položky *points* z třídy Draw.

**std::vector<Edge> & setDT()**

Metoda slouží pro nastavení privátní položky *dt*, tak aby byla vidět z třídy Draw.

**void getDT(std::vector<Edge> &dt\_)**

Metoda slouží pro vrácení privátní položky *dt* z třídy Draw.

**std::vector<Edge> & getDTsize()**

Metoda slouží pro vrácení velikosti Delaunayho triangulace z třídy Draw.

**void setContours(std::vector<Edge> &contours\_,  
std::vector<int> &contour\_heights, int dz\_)**

Metoda slouží pro nastavení privátních položek týkajících se vrstevnic, tak aby byly vidět v třídě Draw.

**std::vector<Edge> & getContours()**

Metoda slouží pro převod vrstevnic do vykreslovacího okna.

**void setMainContours(std::vector<Edge> &mainContours\_,  
std::vector<int> &main\_contour\_heights, int main\_dz\_)**

Metoda slouží pro nastavení privátních položek týkajících se hlavních vrstevnic, tak aby byly vidět v třídě Draw.

**std::vector<Edge> & getMainContours()**

Metoda slouží pro převod hlavních vrstevnic do vykreslovacího okna.

**void setSlope(bool slope\_)**

Metoda slouží jako podmínka TRUE/FALSE pro vykreslení sklonu terénu.

**void setAspect(bool aspect\_)**

Metoda slouží jako podmínka TRUE/FALSE pro vykreslení expozice terénu.

**void setDTM(std::vector<Triangle>&dtm\_)**

Metoda slouží pro nastavení privátních položek týkajících se trojúhelníků a jejich informací o sklonu a expozici terénu.

**void importPolygons(std::string &path, std::vector<QPoint3D>&points, QSizeF &canvas\_size, double &min\_z, double &max\_z)**

Tato metoda slouží pro import bodů ze souboru. Z cesty path se naplní vektor points a uloží hodnotu s minimální a maximální souřadnicí. Proměnná canvas\_size slouží k vykreslení v rozsahu importovaných dat.

**void clearDT()**

Metoda slouží k vymazání celého obsahu okna a k překreslení.

**void clearPoints()**

Metoda slouží pouze k vymazání bodů.

### 7.1.4 SortbyX

Tato jednoduchá třída slouží k setřídění vstupního pole souřadnic podle osy x.

**bool operator()(QPoint &p1, QPoint &p2)**

Přetížený operátor () vrátí bod s větší souřadnicí x z dvojice bodů.

### 7.1.5 Edge

**Edge(QPoint3D &start, QPoint3D &end)**

Třída Edge je zkonstruována ze dvou 3D bodů, počátku a konce hrany. Třída slouží k uložení hrany triangulace i vrstevnice.

**QPoint3D &getStart()**

Metoda vrátí počáteční bod hrany.

**void setStart(QPoint3D & s)**

Metoda nastaví počáteční bod hrany.

**QPoint3D & getEnd()**

Metoda vrátí koncový bod hrany.

**void setEnd(QPoint3D & e)**

Metoda nastaví koncový bod hrany.

**void changeOrientation()**

Metoda prohodí orientaci hrany.

### 7.1.6 QPoint3D

**QPoint3D(double x, double y, double z\_)**

Třída QPoint3D je odvozena z třídy QPointF a složí k uložení bodu s informací o výšce.

**double getZ()**

Metoda vrátí výšku bodu.

**void setZ(double z\_)**

Metoda nastaví výšku bodu.



### 7.1.7 Triangle

**Triangle(QPoint3D &p1\_, QPoint3D &p2\_, QPoint3D &p3\_, double slope\_, double aspect\_)**

Třída QPoint3D složí k uložení trojúhelníku definovaného body p1, p2, p3 a jeho informace o sklonu a expozici.

#### **QPoint3D getP1()**

Metoda vrátí první bod trojúhelníku.

#### **QPoint3D getP2()**

Metoda vrátí druhý bod trojúhelníku.

#### **QPoint3D getP3()**

Metoda vrátí třetí bod trojúhelníku.

#### **double getSlope()**

Metoda vrátí sklon trojúhelníku.

#### **double getAspect()**

Metoda vrátí expozici trojúhelníku.

Obdobně jako gettery byly nastaveny settery.

### 7.1.8 Widget

#### **void on\_createDelaunay\_clicked()**

Při stisknutí tohoto tlačítka se zavolá metoda třídy Algorithms DT a výsledek se zobrazí v okně.

#### **void on\_clearDT\_clicked()**

Při stisknutí tlačítka Clear DT se zavolá metoda třídy Draw clearDT a smaže se celý obsah okna.

**void on\_clearPoints\_clicked()**

Při stisknutí tlačítka Clear points se smažou body.

**void on\_createContours\_clicked()**

Při stisknutí tlačítka Create Contours se zkontroluje, zda byla již vytvořena DT. Pokud ne, vytvoří se a nad ní se poté zavolá metoda třídy Algorithms createContours a výsledek se zobrazí v okně.

**void on\_importPolygons\_clicked()**

Při stisknutí tlačítka Import, se otevře okno pro výběr importovaných dat.

**void on\_analyzeDTM\_clicked()**

Při stisknutí tlačítka AnalyzeDTM se zavolá metoda třídy Algorithms analyzeDTM a podle vybrané analýzy v ComboBoxu se zobrazí v okně buď analýza sklonu, nebo expozice.

**void on\_generateTerrain\_clicked()**

Při stisknutí tlačítka Generate a výběru z comboboxu se vygenerují body terénu.

**void on\_Save\_clicked()**

Při stisknutí tlačítka Save, se otevře okno pro volbu místa uložení vytvořeného obrázku a následně je obrázek uložen.

# Diskuse

V této části se zamyslíme nad slabostmi použitých algoritmů a nad možnými vylepšeními.

## Delaunayho triangulace

Algoritmus DT není jednoznačný pro body na mřížce. Zároveň selhává pro některé terénní tvary (zejména sedlo). Oba tyto problémy by se daly vyřešit triangulací se vstupní podmínkou, která definuje povinné hrany, jež umožňují lepší modelování morfologie terénu. V aplikaci není bohužel možné definovat povinné hrany, proto program není dost dobře použitelný pro data se složitějšími typy terénních tvarů. Tvorba povinných hran by výrazně zkvalitnila aplikaci, neboť by uživatel získával věrohodnější výstupy. S tímto problémem souvisí i to, že jsou následně chybně vykresleny vrstevnice. V rámci úlohy nebyla řešena triangulace nekonvexní oblasti. Tedy i na místech, kde už nejsou podrobné body jsou chybně vytvářeny trojúhelníky. Aplikace je tedy užitečná pro málo členitá výškopisná data v terénu bez povinných hran, například spočinek je zobrazen velmi věrohodně.

## Import dat

Při importu není řešen souřadnicový systém dat. Body by před nahráním do mapového okna měly být transformovány do takového souřadnicového systému, který data vhodně zobrazí. V této úloze se však primárně zabýváme tvorbou digitálního modelu, není tedy na transformaci vstupních dat brán ohled. V testovaném souboru jsou jen takové body, které se v rozhraní správně vykreslí.

## Vykreslení vrstevnic

Pro vykreslení vrstevnic byl použit lineární interpolační algoritmus, který předpokládá, že spád mezi dvěma body, mezi nimiž provádíme interpolaci, je konstantní. Tento algoritmus je sice výpočetně jednoduchý, ale nevystihuje realitu. Pro kvalitní výsledek by bylo vhodnější použít geomorfologickou interpolaci, která předpokládá plynulou změnu sklonu terénu mezi jednotlivými body. Tento typ interpolace se však obtížně algoritmizuje, často je tvořena ručně zkušenými kartografy.

Pokud se zaměříme na vrstevnice, pro údolí a hřbet jsou vykresleny korektně. U vrstevnic kupy již dochází k chybám a to zejména blízko vrcholu. Důvodem je, že body jsou na kružnici, kde Delaunayho triangluace není jednoznačná. Vrstevnice mají zvláštní tvar, někdy se dokonce i kříží, případně dvojí. Velmi zle jsou vrstevnice generovány u sedla, kde blízko vrcholů a prohlubní vytváří maličké uzavřené polygonky a dál od nich chybí úplně. Vrstevnice by se měly chovat rovnoměrně, lahodit oku, což v případě sedlového úvbaru rozhodně neplatí.

### Výpočet a vykreslení sklonu a expozice

Může vzniknout problém v rovinatém území, jak je patrné z obrázku 18. Při výpočtu vznikají zaokrouhlovací chyby, které se promítnou do sklonu a expozice. Proto se terén jeví uživateli jako nerovinný.

Při vykreslení sklonu terénu byly pro ilustraci zvoleny odstíny od zelené po červenou. Sklon nabývá hodnot od 0 do  $180^\circ$ , nicméně reálně jsou hodnoty sklonu o dost nižší, proto byla barevná stupnice volena v intervalu od  $1^\circ$  do  $22^\circ$ .

Vykreslení expozice bylo rovněž obohaceno o barevnou semi-průhlednou paletu, jež byla také odvozena od palety z aplikace Analýzy výškopisu. Výsledná směrnice byla rozdělena do osmi směrů - S, SV, V, JV, J, JZ, Z, SZ. Rovina uvažována nebyla. Byla snaha o velkou barevnou návaznost, aby byl výsledný obrázek přehledný. Nicméně i přesto by bylo rozhodně vhodné u obou analýz do mapového okna doplnit legendu.

Vyhodnocení sklonu i expozice terénu je stejně jako vykreslení vrstevnic v některých případech velmi nejednoznačné. Pro hřbet i údolí je sklon zobrazen správně. Znázornění spočinku je také správné, je krásně patrné, jak směrem dolů rozestup vrstevnic roste a pak se zase snižuje. Pro kupu je nekorektně vyobrazen střed, neboť v něm nejsou korektně vykresleny Delaunay trojúhelníky. Obdobný problém pak platí u sedla. Expozice pro údolí a hřbet je chybná (viz 23), nastává stejný problém, jako při generování gridu, kdy DT je nejednoznačná.

Co se týče importovaných dat, textové soubory neobsahují mnoho výrazných terénní útvarů, které by mohly ovlivnit výpočty. Vrstevnice, sklon i expozice byly pro importovaná data vykresleny víceméně správně (když vezmeme v úvahu lineární interpolaci, přesnější by byla samozřejmě morfologická).

## Závěr

Byla vytvořena aplikace, která má několik funkcionalit. Umí generovat vybrané terénní tvary, importovat soubor se souřadnicemi i kreslit body kursorem myši. Nad všemi těmito množinami dokáže zkonstruovat Delaunay triangulaci, jejíž trojúhelníky jsou ztěžejní při následném generování vrstevnic a provádění analýz sklonu a expozice. Dále má uživatel možnost uložit si vytvořený model jako obrázek, což může výrazně pomoci při práci. V rámci bonusových úloh byly vytvořeny některé terénní útvary, konkrétně kupa, hřbět, sedlo, údolí a spočinek. Co se týče výstupů aplikace, pro některé útvary funguje kód korektně, pro jiné se vyskytují chyby. Ukázky jednotlivých výsledků lze nalézt v kapitole 6. Pro kvalitnější výsledek by bylo vhodné použít nějakou funkci pro vyhlazení vrstevnic či pro tvorbu vrstevnic prostřednictvím morfologické interpolace.

I přesto, že se mnoho věcí podařilo do aplikace naprogramovat, je nespočet funkcionalit, které jsou velmi podstatné a v aplikaci chybí. Jednou z hlavních je zobrazení jednotlivých vrstevnic podle kartografických zásad. Z vygenerované nebo importované množiny bodů není možné odstranit některé body, nad kterými nechci provádět analýzu, nebo vybrat zájmovou oblast pomocí ohraničujícího obdélníku. Dalším vylepšením by byla například schopnost generovat správně DT v nekonvexní oblasti, či umožnit kromě zobrazení vrstevnic i barevnou hypsometrii nebo 3D vizualizaci terénu.

# Literatura

- [1] Analýzy výškopisu. [online], Napsledy navštíveno 10. 11. 2019, dostupné z: <https://ags.cuzk.cz/dmr/>.
- [2] Triangulated irregular network. [online], Napsledy navštíveno 10. 11. 2019, dostupné z: [https://www.wikiwand.com/en/Triangulated\\_irregular\\_network](https://www.wikiwand.com/en/Triangulated_irregular_network).
- [3] Esri: How Aspect works. [online], Napsledy navštíveno 13. 11. 2019, dostupné z: <http://desktop.arcgis.com/en/arcmap/10.3/tools/spatial-analyst-toolbox/how-aspect-works.htm>.