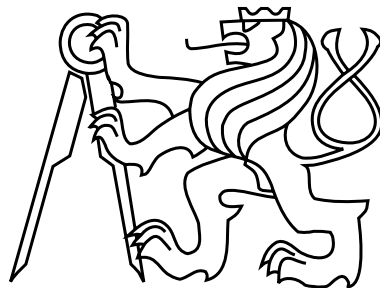


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ
OBOR GEODÉZIE A KARTOGRAFIE



DOKUMENTACE PROJEKTU INFORMATIKA 2 (PIN2)

TUTORIAL „PROGRAMOVÁNÍ V PYTHONU
PRO ARCGIS 10.0“ PRO ZAČÁTEČNÍKY

MICHAL BEČIČKA, JAN ZAJÍČEK

15. května 2013

Obsah

1	Úvod	2
1.1	Poznámka ke kompatibilitě	2
1.2	Moduly	2
2	Jak začít skriptovat	2
2.1	Založení nového skriptu	2
2.2	Obecné poznatky pro psaní skriptů	5
2.2.1	Vybrané syntaktické zásady Pythonu	5
2.2.2	Help	6
2.2.3	Načtení vstupních parametrů	6
2.2.4	Výpisy zpráv a atributových typů	7
2.2.5	Přístup k prvkům atributové tabulky	7
2.2.6	Nastavení pracovního adresáře a práce s pozicemi na disku	8
2.2.7	Práce s Feature Class a Feature Layer	8
2.2.8	Seznam parametrů	9
2.2.9	Testování existence atributového sloupce	9
2.2.10	Balíček NumPy a jeho použití	10
3	Závěr	11

1 Úvod

Tato dokumentace by měla sloužit jako manuál pro začátečníky, kteří chtějí začít psát skripty v prostředí ArcGIS verze 10.0.

1.1 Poznámka ke kompatibilitě

Na úvod jen pár základních informací o skriptování v ArcGIS. Každá nová verze ArcGIS má integrovanou nejnovější verzi Pythonu. Stejně tak se vyvíjí i moduly potřebné k využívání ArcGIS funkcí v Pythonu. To bohužel neumožňuje zpětnou kompatibilitu. Např. zde byl využíván ArcGIS 10.0 a Python 2.6. Při vyhledávání informací a návodů je dobré mít toto na paměti a využívat zejména zdroje odkazující na čtenářovu aktuální verzi prostředí ArcGIS.

1.2 Moduly

Při skriptování v ArcGIS musíme implementovat i další žádoucí moduly. Jedním z nejdůležitějších modulů je *ArcPy*, který pod sebou zahrnuje standardní toolboxy v ArcGISu. Jeho cílem je vytvořit užitečný a produktivní způsob, jak provést geografickou analýzu dat, konverze dat a správu dat. Dalším velice významným modulem je balíček *NumPy* užívaný pro vědecké výpočty. Umožňuje podporu velkých, multi-dimenzionálních polí a matic spolu s velkou knihovnou matematických funkcí pro práci s těmito poli. Modul *Math* podporuje práci s matematickými funkcemi a objekty. Např. goniometrické funkce, konstanty atd. Dalším užitečným modulem může být *Os*, *Sys* a *String*. *String* umožňuje práci s textovými řetězci. Např. funkci *print*. *Os* a *Sys* podporují mj. práci se systémovými operacemi. Např. práce s pozicemi na disku apod. Všechny moduly se zavolají pomocí funkce *import*. Např. `import arcpy` zpřístupní knihovny v balíčku *arcpy*.

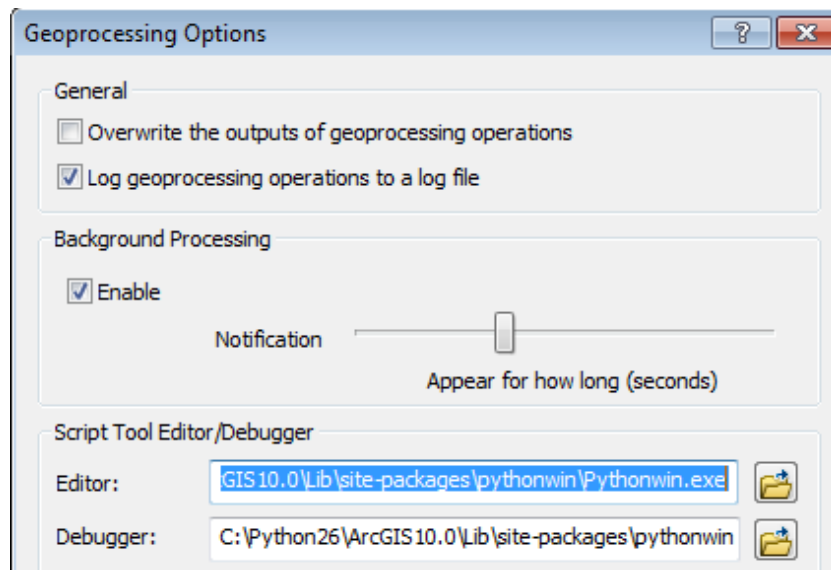
2 Jak začít skriptovat

2.1 Založení nového skriptu

Nejprve je potřeba v libovolném adresáři založit soubor s příponou *.py*. Tento soubor bude obsahovat zdrojový kód. Kód můžeme editovat jak v externím IDLE, tak v IDLE integrovaném v instalaci ArcGIS jmenující se PythonWin. V každém případě je vhodné toto prostředí v ArcGIS určit, aby mohl ArcGIS s editorem spolupracovat.

1. **Nastavení editoru v ArcGIS** - *Geoprocessing* → *Geoprocessing options*

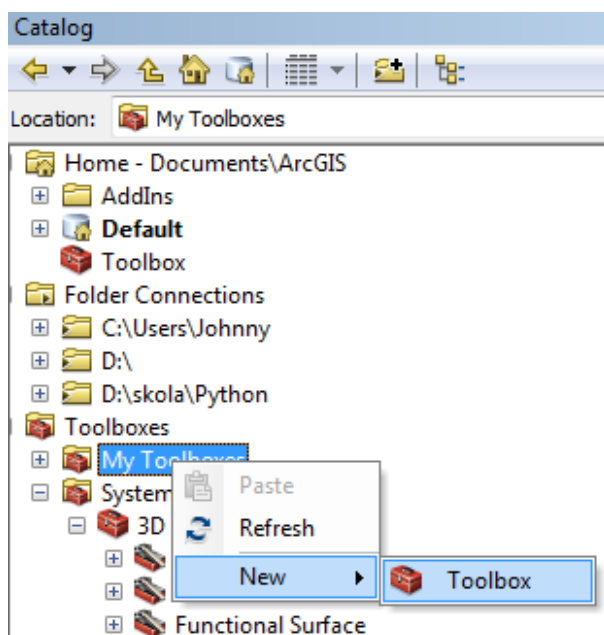
Zde v poli *Geoprocessing tool* → *editor/debugger* zvolíme námi preferovaný editor. Zadá se cesta k .exe souboru programu PythonWin.



Obrázek 1: Nastavení editoru a debuggeru

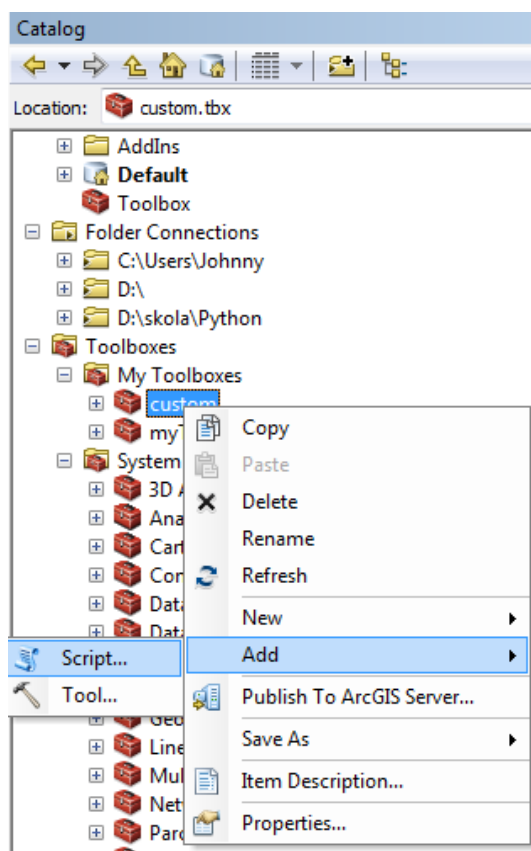
Nyní při volbě editace skriptu přímo v ArcGIS se nám zdrojový kód otevře v daném editoru viz níže.

2. **Vytvoření nástroje (tool)** Nové toolboxy a nástroje se vytvářejí stejně jako jiné objekty v ArcGIS - tedy v ArcCatalogu. Nový toolbox se vytvoří po kliknutí pravým tlačítkem myši na adresář *Toolboxes/My Toolboxes* a zvolení položky *New → Toolbox*. V takovémto uživatelském Toolboxu můžeme přidat skript. Opět přes pravé tlačítko myši *Add → Script* najdeme soubor *.py*, do kterého budeme psát zdrojový kód. Nyní při opětovném zapnutí ArcGIS stačí najít v ArcCatalogu tento skript, kliknout pravým a zvolit *Edit...* Zdrojový kód se automaticky otevře v definovaném editoru.



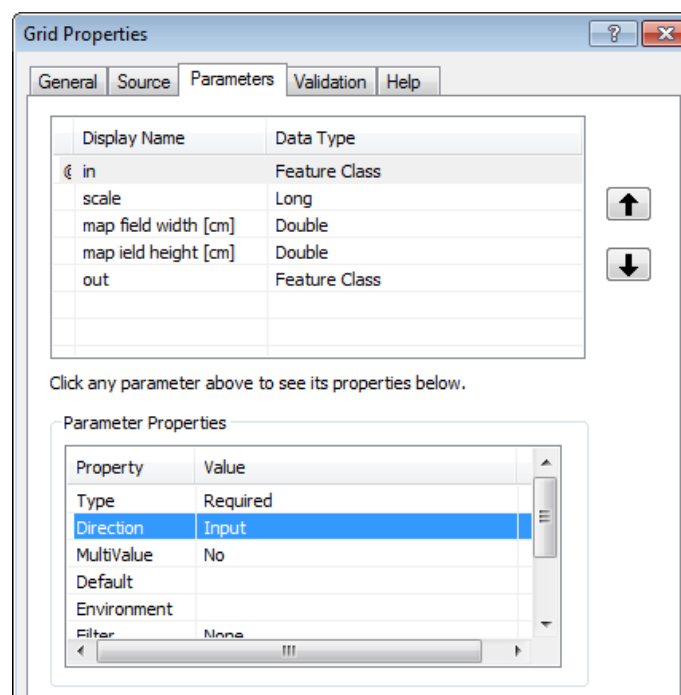
Obrázek 2: Vytvoření nového toolboxu

Při přidávání skriptu můžeme zadávat vstupní a výstupní parametry. Tento krok můžeme přeskočit a editovat tyto parametry později.



Obrázek 3: Přidání skriptu

3. **Editace vstupních a výstupních parametrů** Pro editaci parametrů klikneme pravým na skript v ArcCatalogu a zvolíme *Properties...* Zde můžeme mimo jiné editovat i cestu k souboru *.py*. V záložce *Parameters* je seznam parametrů. Po kliknutí na parametr se ve spodním okně objeví jejich atributy z nichž nejdůležitější je *Direction* tj., zda-li se jedná o vstupní nebo výstupní parametr. Význam dalších atributů je jasný z jejich názvu a nabídky jejich hodnot. Při definování parametrů je potřeba zvolit správný datový typ parametru z předdefinované nabídky.



Obrázek 4: Editace parametrů

2.2 Obecné poznatky pro psaní skriptů

Před psaním skriptu je samozřejmě žádoucí si nejprve osvojit základní syntaxi Pythonu. Poté můžeme psát skript pro ArcGIS. Protože úvod do Pythonu je perfektně zpracován v mnoha dokumentech, zde jen stručně uvedeme pár zásad.

2.2.1 Vybrané syntaktické zásady Pythonu

Python rozeznává **bílé znaky**. Např. tělo cyklu je označeno pouze odsazením tabulátorem. V C++ by funkce bílých znaků byla nahrazena složenými závorkami .

Python rozlišuje **velká a malá písmena**. **Komentáře** se značí #, označuje celý řádek. Víceřádkový komentář není možný.

Pole je nejprve nutné inicializovat, pak je možné plnit jej pomocí indexu.

Další informace jsou pěkně sepsány např. zde <http://www.sallyx.org/sally/python/index.php>

2.2.2 Help

Pro úvod do skriptování v ArcGIS je nejvýznamnějším pomocníkem Help přímo od společnosti ESRI:

<http://help.arcgis.com/en/arcgisdesktop/10.0/help/>

Help je v angličtině a je zde přehledně popsána hledaná funkce a její použití je demonstrováno na ukázce kódu, což je obrovskou výhodou pro porozumění, jak daná funkce pracuje a funguje.

2.2.3 Načtení vstupních parametrů

```
arcpy.GetParameterAsText(index)
```

Indexy v Pythonu začínají od 0 a zde označují pořadí parametru, jak byl definován v ArcCatalogu.

Vždy je potřeba parametr uložit do nějaké proměnné, tedy správná syntaxe bude

```
NazevPromenne = arcpy.GetParameterAsText(0)
```

Stejně tak vytvoříme proměnnou, která bude reprezentovat výstupní parametr. Do této proměnné pak ukládáme výsledek.

PŘÍKLAD

Skript, který zkopíruje shapefile a uloží jej pod uživatelem zvoleným názvem do zvoleného adresáře:

```
import arcpy
inShapefile = arcpy.GetParameterAsText(0)
outShapefile = arcpy.GetParameterAsText(1)
arcpy.Copy_management (inShapefile, outShapefile)
```

2.2.4 Výpisy zpráv a atributových typů

Pro jakýkoliv textový výstup do výsledného dialogu se používá funkce *AddMessage()*. Pokud chceme touto funkcí vypsát hodnotu atributu nebo jiné proměnné, která není datového typu *string*, použijeme přetypování *str(proměnná)*. Pro zjištění atributu objektu se často využívá funkce *Describe*, která umožní práci s množstvím atributových typů. PŘÍKLAD

```
import arcpy
inShapefile = arcpy.GetParameterAsText(0)
ShpShapeType = arcpy.Describe(inShapefile).shapeType
```



```
arcpy.AddMessage("Shape Type: " + str(ShpShapeType))
```

Na příkladu je demonstrováno, jak vypsat typ vstupního rastru. Vedle výpisu zprávy *AddMessage* existují ještě výpis varování *AddWarning* (text je zvýrazněný) a výpis chybové hlášky *AddError* (běh programu se ukončí, ve výsledném dialogu v ArcGIS se vypíše daná hláška, a výpočet je vyhodnocen jako neúspěšný).

2.2.5 Přístup k prvkům atributové tabulky

K prvkům atributové tabulky se dostaneme pomocí funkce *SearchCursor*. Ta nám vytvoří pole *ntit* (jednotlivých řádků) tabulky. Ty se pak dají kromě indexování procházet např. cyklem *for*. V tomto cyklu můžeme porovnávat hodnoty atributu s hledanou hodnotou atributu. Tu nalezneme jako *ntita.název_atributu*. V příkladu je to formule *row.POINT_X*. Hodnota FID je hodnota zadaná uživatelem, *ntity* jsou body z shapefile. Kód v příkladu uloží souřadnice bodů do polí X, Y.

PŘÍKLAD

```
import arcpy
inShapefile = arcpy.GetParameterAsText(0) #typu shapefile
FID_P1 = arcpy.GetParameterAsText(1) #typu long
FID_P2 = arcpy.GetParameterAsText(2) #typu long

# inicializace dvouprvkových polí a pomocné proměnné
X = [0,0]
Y = [0,0]
a=0

for row in rows:
    if i==int(FID_P1) or i==int(FID_P2):
        X[a] = math.radians(row.POINT_X)
        Y[a] = math.radians(row.POINT_Y)
        a = a+1
    i = i+1
```

Pokud bychom nehledali *ntitu* podle FID, které tvoří vždy posloupnost přirozených čísel, můžeme proměnné *i* přiřadit místo čísla například jméno města *i* =

row.Name. Pak by ovšem vstupní parametr musel být typu *String*, byl by povinný a neobsahoval by přednastavené hodnoty.

2.2.6 Nastavení pracovního adresáře a práce s pozicemi na disku

Pokud není nastaven pracovní adresář (tzv. workspace), ArcGIS automaticky nabízí cestu k výstupním parametrům podle parametrů vstupních. Defaultně je nastavena cesta k systémové geodatabázi ArcGISu *Default.gdb*. Pokud chceme toto změnit můžeme pracovní adresář nastavit příkazem

```
arcpy.env.workspace = "cesta"
```

Pracovní adresář může být složka i geodatabáze.

2.2.7 Práce s Feature Class a Feature Layer

Některé funkce pracují s Feature Class a některé s Feature Layer. Některé funkce umí pracovat s oběma typy dat. Musíme se tedy rozhodnout, která data použít a popř. je vytvořit.

Rozdíl mezi těmito typy je v tom, že Feature Class jsou data uložená pevně na disku. Tedy například shapefile nebo vrstva geodatabáze. Typ Feature Layer je uchovávan v paměti počítače a obsahuje aktuální situaci vrstvy. Především tedy pouze selektované prvky.

Chceme-li mít přístup k selektovaným prvkům, musíme vždy použít typ Feature Layer. PŘÍKLAD

Chceme-li naplnit zásobník identifikátory všech prvků vrstvy nezávisle na tom, jestli jsou některé označeny, použijeme Feature Class. Chceme-li naplnit zásobník identifikátory aktuálně označených prvků, použijeme Feature Layer.

Vytvoření Feature Layer z Feature Class:

```
path = arcpy.GetParameterAsText(0)
arcpy.MakeFeatureLayer_management(path, "layer")
```

2.2.8 Seznam parametrů

Některé funkce jako např. *Merge_management* používají jako vstupní parametr seznam parametrů (vrstev). Ten se vytváří jako běžný seznam, který ovšem obsahuje textové řetězce určující cestu k daným vrstvám:

```
merge_list = []
while pathToFeatureClass:
merge_list.append('pathToFeatureClass')
arcpy.Merge_management(merge_list, outFeatureClass)
```

2.2.9 Testování existence atributového sloupce

Pokud chceme přidat atributový sloupec funkcí *AddField_management*, je vhodné nejprve otestovat, zda-li již neexistuje. To se provede pomocí seznamu sloupců:

```
fieldList = arcpy.ListFields(inFeatureClass, 'section')
fieldCount = len(fieldList)

if (fieldCount != 1):
    arcpy.AddField_management(inFeatureClass, 'section', 'SHORT')
```

2.2.10 Balíček NumPy a jeho použití

Balíček NumPy má obrovský význam při práci s poli a maticemi. Pokud máme na vstupu rastr, je třeba ho převést do podoby, ve které budou hodnoty v jednotlivých buňkách snadno dostupné a možné matematické operace s nimi. K tomuto využijeme funkce *RasterToNumPyArray()*:

```
arcpy.RasterToNumPyArray(InputRaster)
```

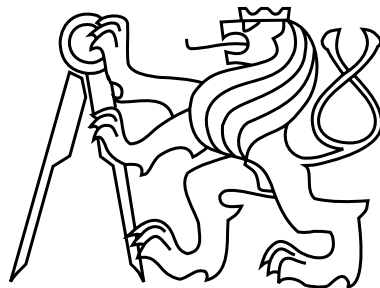
Tato funkce převede na rastr na matici a jí už můžeme bez problému indexovat a provádět s ní početní operace. Jestliže potřebujeme dostat jako výsledek opět rastr, převedeme výslednou matici opačným postupem na rastr:

```
NumPyArrayToRaster (matice, {levý_dolní_roh}, {velikost_pixelu_x},
{velikost_pixelu_y}, {hodnota_pro_NoData})
```

3 Závěr

V tomto dokumentu byly demonstrovány ukázky a případy pro úplné začátky při programování skriptu spustitelného v prostředí ESRI ArcGIS 10.0. Kompletní zdrojové kódy, ze kterých se čerpaly ukázky jsou ke stažení na stránkách předmětu PIN2.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ
OBOR GEODÉZIE A KARTOGRAFIE



DOKUMENTACE K PROJEKTU INFORMATIKA 2

AUTOMATIZACE SESTAVENÍ ATLASU
EROZNÍHO SMYVU A TRANSPORTU
SPLAVENIN VE VELKÝCH POVODÍCH

MICHAL BEČIČKA

15. května 2013

Obsah

1	Úvod	2
1.1	O tématu	2
1.2	Stručný souhrn	2
2	Popis prací	3
2.1	Nástroj pro tvorbu gridu	3
2.1.1	Úvodní dialog	4
2.1.2	Výsledný log	5
2.1.3	Výsledné vrstvy	5
2.1.4	Činnost nástroje podrobně	6
2.2	Generalizace	9
2.2.1	Úvodní dialog	9
2.2.2	Výsledný log	10
2.3	Finální úpravy a export	10
3	Problémy a možnosti rozšíření	10
3.1	Připojování externích tabulek	10
3.2	Doba běhu tvorby gridu	11
3.3	Možnosti rozšíření	11
4	Závěr	12
5	Přílohy	13
5.1	Výsledný log tvorby gridu	13
5.2	Výsledný log generalizace	14

1 Úvod

1.1 O tématu

Téma bylo určeno doc. Krásou z katedry Hydromeliorací a krajinného inženýrství pro prezentování výsledků výpočtu erozního smyvu a transportu splavenin. Přesto, že je nástroj primárně určen pro toto použití, byla snaha jej napsat co nejuniverzálněji.

Cílem projektu je vytvořit atlas sestavený z jednotlivých tématických map. Tyto mapy jsou vytvořeny pro cca třetinu území České Republiky. V potřebném měřítku (cca 1:50 000) je nemožné vytvářet spojené tématické mapy. Bylo tedy nutné nejprve vyhodnotit, jak nejlépe výsledky prezentovat, poté vytvořit nástroj, který mapy podle požadavků upraví (generalizuje vizualizované prvky), vygeneruje jednotlivé mapové listy a jejich přehledku. Dále vygeneruje seznam mapových listů pro jednotlivá povodí a další potřebné údaje.

Nástroj je psán v jazyce Python pro ArcGIS 10.0 .

1.2 Stručný souhrn

Práce se skládá z několika částí.

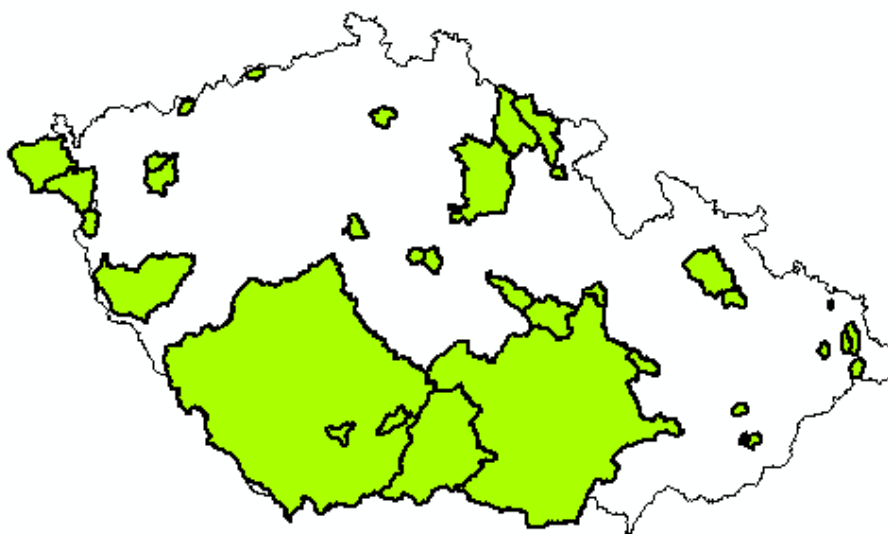
1. **Vytvoření gridu** - Tato část je nejrozsáhlejší. Jedná se o skript, který pro daná povodí vytvoří rozsah mapových listů (tzv. index grid layer) a do atributových tabulek uloží potřebné hodnoty viz dále.
2. **Generalizace obsahu mapy** - Generalizace nakonec proběhla velmi jednoduše a to kategorizací hodnot kumulovaného smyvu.
3. **Úprava vzhledu a export** - Tato část byla prováděna manuálně v arcgis layout view. Po vytvoření kýženého vzhledu poté proběhlo generování jednotlivých mapových listů pomocí toolbaru Data Driven Pages.

2 Popis prací

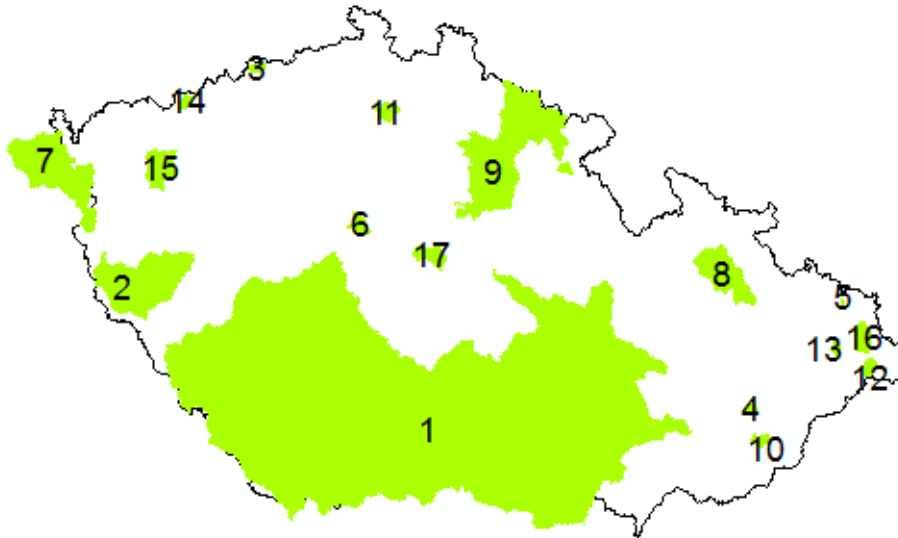
2.1 Nástroj pro tvorbu gridu

Tento nástroj vytvoří polygonovou vrstvu obsahující grid a to podle zadaného měřítka a velikosti mapového pole. Do atributové tabulky gridu vloží sloupec s názvy mapových listů a sloupec s čísly sekcí (bude vysvětleno dále). Zároveň přidá do vstupní polygonové vrstvy sloupec s čísly sekcí a sloupec s názvy mapových listů, na kterých se daný polygon (povodí) nachází. Dále vytvoří vrstvu *název_gridu_dissolve* pro přehledku sekcí.

Pojem sekce Sekce je tvořena všemi dotýkajícími se polygony. Mapové listy jsou ke každé sekci tvořeny zvlášť a jejich názvy jsou poté rozšířeny o číslo sekce. Příklad číslování mapových listů: 1-A1, 1-A2, ..., 2-A1, 2-A2, ...



Obrázek 1: Vstupní polygonová vrstva

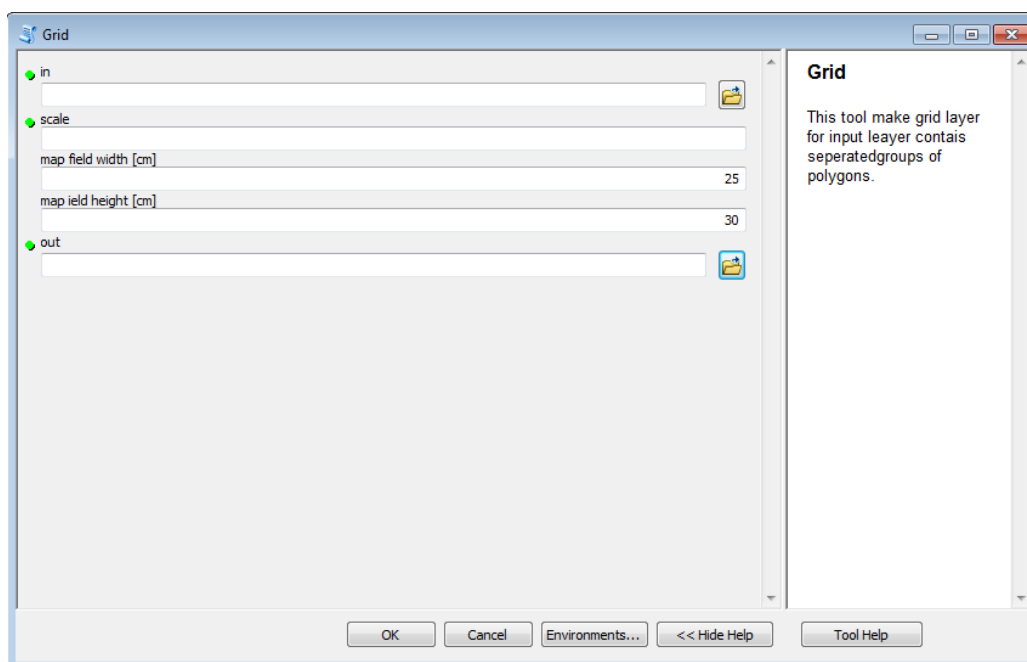


Obrázek 2: Očíslované sekce polygonové vrstvy

2.1.1 Úvodní dialog

Úvodní dialog nástroje obsahuje pole pro 5 parametrů:

- **In** - vstupní parametr, datový typ *Feature class*
Vstupní vrstva polygonů.
- **Scale** - vstupní parametr, datový typ *Long*
Požadované měřítko.
- **Map field width** - vstupní parametr s nastavenou defaultní hodnotou 25, datový typ *Double*
Šířka výsledného mapového rámce v cm.
- **Map field height** - vstupní parametr s nastavenou defaultní hodnotou 30, datový typ *Double*
Výška výsledného mapového rámce v cm.
- **Out** - výstupní parametr, datový typ *Feature class*
Výstupní vrstva gridu.



Obrázek 3: Grid - Úvodní dialog

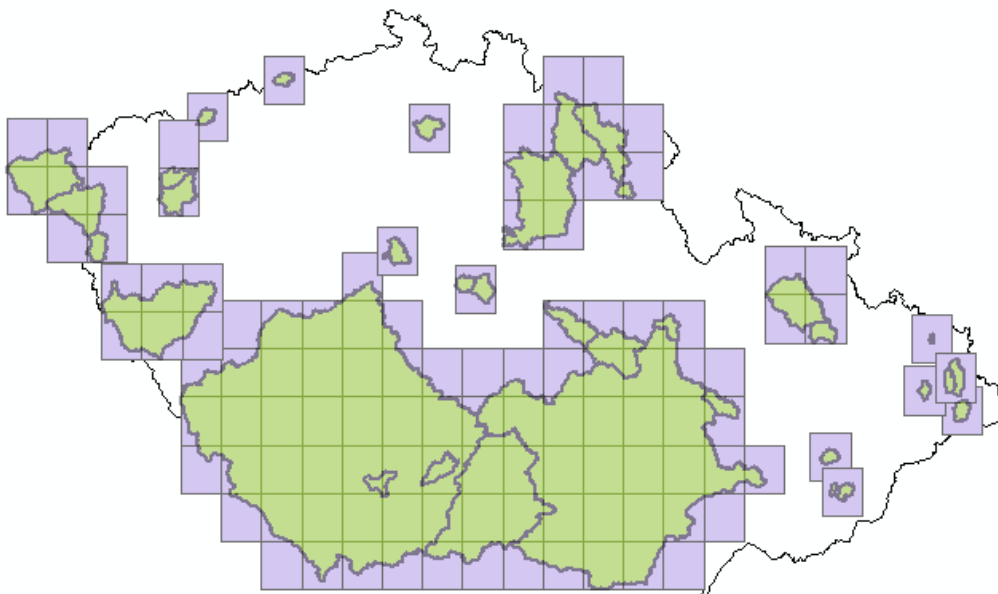
2.1.2 Výsledný log

Kompletní příklad výsledného logu je k nalezení v příloze 5.1. Obsahuje tyto informace:

- Umístění používané databáze na disku
- Dále v každém cyklu:
 - Pořadové číslo cyklu
 - Aktuální počet nezpracovaných polygonů
 - Číslo vytvářené sekce
 - Identifikátory polygonů tvořících danou sekci
- Počet sekcí
- Počet mapových listů v každé sekci

2.1.3 Výsledné vrstvy

Výsledný grid pro měřítko 1 : 80 000 a defaultní velikost mapového pole (25 x 30 cm) vypadá následovně



Obrázek 4: Grid - Výsledná vrstva

2.1.4 Činnost nástroje podrobně

1. Přípravné činnosti

V prvních řádcích kódu jsou vytvořeny pomocné proměnné a pole, z nichž pro další činnost je potřeba zmínit zásobník *IDs*. Ten je naplněn identifikátory povodí.

Pokud již neexistují, jsou ve vstupní polygonové vrstvě vytvořeny atributové sloupce *hard_ID*, *section*, *MapSheets*.

- *hard_ID* - Do tohoto sloupce byly zkopírovány identifikátory, aby i po rozdělení do několika vrstev měly polygony jednoznačný identifikátor.
- *section* - Po výpočtu sekcí sem bude pro každý polygon zapsáno číslo sekce, ve které se nachází.
- *MapSheets* - Atribut se seznamem mapových listů, na kterých se dané povodí nachází.

Ukázka tvorby atributového sloupce:

```
fieldList = arcpy.ListFields(inFeatureClass, 'MapSheets')
```

```

fieldCount = len(fieldList)

if (fieldCount != 1):
    arcpy.AddField_management(inFeatureClass, 'MapSheets', 'TEXT')

```

2. Hlavní výpočetní část

Hlavní část je celá v cyklu `while (len(IDs) > 0)` tj. cyklus probíhá dokud jsou v zásobníku *IDs* hodnoty. Na konci cyklu jsou vždy identifikátory zpracovaných povodí ze zásobníku mazány. V každém průběhu cyklu jsou selektovány povodí tvořící 1 sekci, k takové sekci je poté vytvořen grid a jsou plněny nové atributové sloupce.

Ukázka hlavního cyklu:

```

while len(IDs) > 0 :
    ...označení povodí i-té sekce...
    rows_sel = arcpy.SearchCursor('povodi_class')
    for row_sel in rows_sel:
        IDs.remove(row_sel.OBJECTID)

```

V každém průběhu cyklu je zpracována 1 sekce - tj. shluk povodí, které mají společnou část hranice. Sekce je tvořena na základě dotazu *SelectByLocation* s definicí *BOUNDARY_TOUCHES*. Dotaz se opakuje v cyklu, dokud se počet označených povodí zvětšuje. K tomu slouží proměnné *matchcount1*, *matchcount2*, které obsahují počet označených polygonů.

Vytvoření sekce:

- Označení prvního nezpracovaného polygonu

```

first_ID = str(IDs[0])
fieldName = "OBJECTID"
Query = "\"" + fieldName + "\" = " + first_ID
arcpy.SelectLayerByAttribute_management('povodi_class', 'NEW_SELECTION', Query)
matchcount1 = int(arcpy.GetCount_management('povodi_class').getOutput(0))

```

- První průběh *SelectByLocation*

```

arcpy.SelectLayerByLocation_management('povodi_class', 'BOUNDARY_TOUCHES',
'povodi_class')
matchcount2 = int(arcpy.GetCount_management('povodi_class').getOutput(0))

```

- Další průběhy *SelectByLocation*

```

while matchcount1 < matchcount2:
matchcount1 = matchcount2
    arcpy.SelectLayerByLocation_management('povodi_class', 'BOUNDARY_TOUCHES',
'povodi_class')
    matchcount2 = int(arcpy.GetCount_management('povodi_class').getOutput(0))

```

Označené polygony jsou uloženy jako polygonová vrstva do vstupní geodatabáze. Tato vrstva je dále použita pro tvorbu gridu.

Vytvoření gridu:

Před samotnou tvorbou bylo potřeba zjistit rozsah sekce. Pokud byla sekce menší než 1 mapový list, bylo třeba definovat souřadnice polohy listu tak, aby povodí leželo na středu listu. Do zásobníku byly načteny souřadnice lomových bodů povodí a byly z nich vybrány maximální a minimální hodnoty. Jejich rozdíl pak určil rozsah sekce.

Příklad plnění zásobníku souřadnicema:

```

desc = arcpy.Describe(outputFeatureClassPath+'/' +output_name)
shape_field_name = desc.ShapeFieldName
rows3 = arcpy.SearchCursor(outputFeatureClassPath+'/' +output_name)

mbr_X=[]
mbr_Y=[]

for row3 in rows3:

    # Create the geometry object
    feat = row3.getValue(shape_field_name)
    partnum = 0

    for pnt in feat.getPart(partnum):

        mbr_X.append(pnt.X)
        mbr_Y.append(pnt.Y)

        partnum += 1

```

Vytvoření gridu poté probíhá buď s definovanou nebo s nedefinovanou polohou počátečního mapového listu. Pro vytvoření gridu je použita funkce *GridIndexFeatures_cartography*.

Ukázka:

```
arcpy.GridIndexFeatures_cartography(outputFeatureClassPath+'/' +grid_name,
outputFeatureClassPath+'/' +output_name, "", "USEPAGEUNIT", scale_str,
sheet_width_str, sheet_high_str, str(grid_start_pnt_X)+' ' +str(grid_start_pnt_Y))
```

Grid je tedy vytvořen pro jednu sekci. Po všech průbězích cyklu jsou takto vytvořené vrstvy gridů spojeny do jedné finální vrstvy. V cyklu jsou dále plněny nové atributové sloupce - čísla sekcí a názvy mapových listů a to jak do nové (výstupní) vrstvy gridu, tak do vstupní vrstvy povodí.

3. Vytvoření finálních vrstev, tisk výsledků Jak bylo řečeno, na závěr jsou spojeny jednotlivé vrstvy gridů. Vedle toho je vytvořena vrstva spojených mapových listů po sekcích - funkcí *Dissolve* - pro tvorbu přehledky. Jsou vytisknuty výsledné hodnoty viz 2.1.2.

2.2 Generalizace

Pro generalizaci byl napsán skript, který načítá liniovou vrstvu toků. Tento skript byl již psán na míru podkladovým datům a při načítání je v něm využito znalosti struktury daných podkladových dat. Skript k vrstvě připojí externí tabulku a vytvoří atributový sloupec *kategorie*. Podle dat z externí tabulky pak plní nový sloupec hodnotami 1 - 5 podle množství unášeného sedimentu. Tato hodnota je pak použita při vizualizaci dat.

Zajímavostí je, že pro načítání musela být tabulka indexována viz 3.1.

2.2.1 Úvodní dialog

Úvodní dialog obsahuje dva parametry:

- **Input Feature Class** Vstupní liniová vrstva říčních úseků.
- **Table** Vstupní tabulka obsahující hodnoty říčních splavenin.

2.2.2 Výsledný log

Výsledný log je k nahlédnutí v příloze 5.2. Obsahuje tyto informace:

- Počet všech říčních úseků
- Počet úseků v jednotlivých kategoriích

2.3 Finální úpravy a export

Jak již bylo zmíněno finální vzhled mapového listu byl vytvořen manuálně v *Layout view* a pomocí toolbaru *Data Driven Pages*. Tento postup je časově ekonomičtější a je jím dosaženo lepších výsledků díky přehlednosti. Toolbar *Data Driven Pages* pak umožňuje vkládat dynamické textové pole, které se automaticky mění s měnícím se mapovým listem. Takovým polem je např. název mapového listu. Stačí tedy definovat vzhled jednoho mapového listu, v toolbaru *Data Driven Pages* navolit příslušné pole - ty byly vytvořeny nástrojem Grid viz 2.1. Dynamicky bylo potřeba propojit s mapovým oknem i přehledku, která byla vytvořena jako druhý Data Frame. To se provedlo definováním rozsahu podle hlavního mapového okna v Properties -> Data Frame -> Extant a definováním *Definition Query* ve vlastnostech vrstvy. Dalším dynamickým prvkem jsou označení sousedních mapových listů a číslo stránky. Na takto sestavenou stránku s dynamickými prvky bylo možné použít export mapy do pdf a Atlas byl vygenerován.

Příklad dynamického textu pro název mapového listu:

```
<dyn type="page" property="PageName"/>
```

3 Problémy a možnosti rozšíření

3.1 Připojování externích tabulek

Při připojování externích tabulek musely být indexovány hodnoty atributového sloupce použitého k párování. To proběhlo příkazem `arcpy.AddIndex_management ("layer", "HydroID")`. Připojení tabulky musí probíhat k Feature layer. Proto bylo nutné nejprve z Feature Class vytvořit Feature Layer¹. Poté mohla být tabulka připojena:

```
arcpy.AddIndex_management ("layer", "HydroID")
arcpy.AddJoin_management("layer", "HydroID", table, "Riverid")
```

¹Rozdíl mezi Feature Layer a Feature Class je vysvětlen v příloženém Tutorialu programování v Pythonu pro ArcGIS

Zde nastal problém s přístupem k atributům. Obecně nesmí názvy atributových sloupců obsahovat mezery. K hodnotám je přístupováno přes jednotlivé řádky s parametrem *název_sloupce*:

```
rows= arcpy.SearchCursor("layer")
for row in rows:
value=row.název_sloupce
```

Připojená tabulka ovšem měla v názvech sloupců mezeru. K takto pojmenovanému sloupci muselo být přístupováno příkazem *getValue*:

```
rows= arcpy.SearchCursor("layer")
for row in rows:
    value=row.getValue('název tabulky$.název sloupce')
```

3.2 Doba běhu tvorby gridu

Doba běhu nástroje pro tvorbu gridu je znepokolivě dlouhá. Bylo provedeno měření času běhu jednotlivých částí kódu funkcí *time* z modulu *time*. Výsledky ukázaly, že nejdéle trvá vytváření nových vrstev. Vrstva se může vytvářet až 20 sekund. Snaha o optimalizaci donutila nahradit vytváření vrstev s minimálním ohraničujícím obdélníkem sekvenčním ukládáním souřadnic lomových bodů polygonů, z nichž pak funkcemi *min* a *max* byly tyto hodnoty získány. Průběh se tak urychlil o desítky sekund (v jednom průběhu cyklu přibližně o 15 sekund). Další možností bylo využít prostorového indexu dat. Tato možnost nebyla využita, protože ArcGIS používá k prostorovému indexování pravidelný grid. Získání MBR² nebylo nalezeno. Využití indexu by ovšem problém neřešilo. Jak již bylo zmíněno nejdéle trvá vytváření nových vrstev. Bohužel funkce *GridIndexFeatures_cartography* pro vytvoření gridu neumí pracovat se selektovanými daty. Tedy v každém cyklu musí být uložena polygonová vrstva tvořící 1 sekci. Poté mohla být vytvořena vrstva tvořící grid. Tedy v každém průběhu cyklu jsou vytvářeny 2 nové vrstvy. Jelikož autor nepřišel na způsob jiné optimalizace je nástroj pomalý. Pro zadaný dataset s 58mi povodími jeho činnost probíhá desítky minut.

3.3 Možnosti rozšíření

Do skriptu by se dal zakomponovat export seznamu mapových listů pro každé povodí. V současné verzi je tvořen exportem atributové tabulky.

²minimum bounding rectangle - minimální ohraničující obdélník

4 Závěr

Byly vytvořeny nástroje pro automatizaci tvorby Atlasu pro mapy erozního smyvu a transportu splavenin. Tvorba Atlasu probíhá v několika krocích. Vizualizované prvky jsou generalizovány. Dále je vytvořen Grid, který je použit jako indexová vrstva pro funkci *Data Driven Pages*. Tou jsou vytvořeny jednotlivé mapové listy. Pomocí dynamických prvků je pak vytvořen vzhled, který se mění se změnou mapového listu. Nedostatkem je rychlost nástroje pro tvorbu gridu. Přesto, že je nástroj pomalý, je zcela funkční. V případě potřeby dalšího rozšíření bude ale vhodné výsledky načítat do nového nástroje. Vzorový výtisk výsledného atlasu je přiložen. Vzhledem k tomu, že zpracovávaná data jsou rozsáhlá a podléhají autorskému zákonu, je vytvořen jen vzorový výtisk z 6ti povodí. Nástroje jsou psány v jazyce Python 2.6 pro ArcGIS 10.0. Při práci byla využívána především online nápověda ArcGIS a diskuzní programátorská fóra. Kompletní zdrojový kód je ke stažení na stránkách předmětu PIN2.

5 Přílohy

5.1 Výsledný log tvorby gridu

```
Executing: Grid
J:\Atlas\vzorek\temp_select\povodi_vzorek.gdb\pokusne_povodi
70000 25 30 J:\Atlas\vzorek\temp_select\povodi_vzorek.gdb\grid70
Start Time: Wed May 15 02:05:51 2013
Running script Script2...
Output folder: J:\Atlas\vzorek\temp_select\povodi_vzorek.gdb
-----

count of basin polygon to process: 6
-----

RUN 1
count of basins in 1. section: 3
Their ID:
      ID: 1
      ID: 3
      ID: 6
-----

count of basin polygon to process: 3
-----

RUN 2
count of basins in 2. section: 1
Their ID:
      ID: 2
-----

count of basin polygon to process: 2
-----

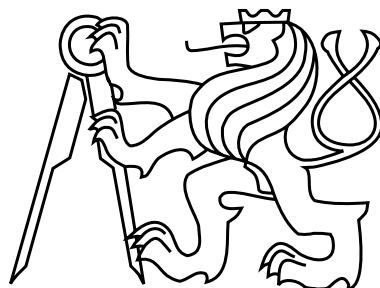
RUN 3
count of basins in 3. section: 2
Their ID:
      ID: 4
      ID: 5

----- RESULT -----
Number of sections: 3
Number of map sheets...
  ... in section 1: 13.
  ... in section 2: 1.
  ... in section 3: 2.
Completed script Grid...
Succeeded at Wed May 15 02:13:59 2013 (Elapsed Time: 8 minutes 8 seconds
```

5.2 Výsledný log generalizace

```
Executing: generalization J:\Atlas\vzorek\2011_NAZV_nadrze\2011_watemsedem\
NAZV_toky\VT28_fin.shp
J:\Atlas\vzorek\2011_NAZV_nadrze\2011_watemsedem\28_vypocet\
projectriversediment.xlsx\projectriversediment$
Start Time: Wed May 15 07:56:24 2013
Running script generalization...
Category created
Total count of river parts: 2879
  Count of category 1: 2428
  Count of category 2: 118
  Count of category 3: 68
  Count of category 4: 20
  Count of category 5: 245
Completed script generalization...
Succeeded at Wed May 15 07:56:38 2013 (Elapsed Time: 14,00 seconds)
```

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ
OBOR GEODÉZIE A KARTOGRAFIE



DOKUMENTACE K PROJEKTU INFORMATIKA 2

STANOVENÍ SMĚRU ODTOKU
NA RASTROVÉM DIGITÁLNÍM MODELU
A IMPLEMENTACE ALGORITMU
DO PROSTŘEDÍ ARCGIS

BC. JAN ZAJÍČEK

5. června 2013

Obsah

Obsah	1
1 Úvod	2
2 Nejběžnější existující algoritmy	3
3 Flow Direction D8	4
3.1 Obecný princip D8	4
3.2 Popis algoritmu D8	5
4 Popis algoritmu MD∞	6
4.1 Obecný princip MD ∞	6
4.2 Technické řešení MD ∞	8
5 Výsledky	11
5.1 Reálná data - Býkovice	11
5.2 Reálná data - Nučice	12
6 Závěr	13
Reference	14
7 Přílohy	15
7.1 Býkovice a Nučice vstupní DMT rastry	15
7.2 Býkovice výsledky	16
7.3 Nučice výsledky	17

1 Úvod

Tato dokumentace byla zpracována v rámci předmětu Projekt Informatika 2. Semestrální práce bude použita jako z jedna částí autorovi diplomové práce s názvem SMODERP 2D rozšíření, řešící vztahy soustředěného a plošného odtoku na daném území. Důvodem zpracování tohoto tématu, byla potřeba použití, v diplomové práci, více než jednoho algoritmu pro určení směru odtoku v prostředí ArcGIS. V současné době ArcGIS využívá algoritmus D8, viz. kapitola 3, jako jediný algoritmus pro výpočet směru odtoku. Tento algoritmus je součástí Hydrology tools. Nástroje ze skupiny Hydrology se používají k modelování chování toku vody na povrchu. Nejběžnější používaná digitální data povrchu Země jsou rastry digitálního modelu terénu DMT složené z jednotlivých buněk, pixelů. Tyto DMT rastry jsou použity jako vstupy pro různé charakteristiky povrchu. DMT rastr je souvislý povrch, obvykle odkazující na povrch Země. Přesnost dat je dána rozlišením rastru, taktéž vzdáleností mezi body rastru. Informace o tvaru zemského povrchu je užitečná pro mnoho oborů, jako je např. zemědělství, lesnictví, životní prostředí, nebo regionální plánování. Tyto obory vyžadují pochopení jak voda teče v rámci území a jak změny na tomto území následně ovlivní její tok.

Cílem projektu bylo naprogramovat algoritmus, který v roce 2007 publikovali Jan Seibert a Brian L. McGlynn s názvem MD_{∞} [1]. Tento algoritmus je vylepšením stávajících algoritmů pro řešení určení směru odtoku na rastru, viz. kapitola 2. Pro ověření správnosti výsledků byla použita reálná data dvou pozemků u obcí Býkovice a Nučice. Testování probíhalo na rastroch o rozlišení 5 metrů. Algoritmus byl naprogramován v jazyce Python v prostředí NetBeans IDE 6.9. Pro spuštění v ArcGISu je potřeba přidat do prostředí vytvořený toolbox, upravit aktuální cestu ke skriptu *main.py* a tento model následně spustit. Vstupem modelu je rastr DMT a výstupní adresář, do kterého se po doběhnutí modelu uloží výsledný rastr s hodnotami směru odtoků 1,...,255. Algoritmus MD_{∞} je testován na verzi ArcGIS 10.0. Na nejnovější verzi ArcGIS 10.1, která vyšla v červnu 2012 není ověřena kompatibilita algoritmu v důsledku nefunkčnosti více než jedné verze tohoto programu na jednom počítači.

2 Nejběžnější existující algoritmy

Existují tři nejběžnější algoritmy, každý odlišně řešící problematiku odtoku vody z rastru do sousedních buněk. Prvním z nich je algoritmus D8 z roku 1984. Algoritmus je podrobně popsán v následující kapitole 3. Jelikož odtok nastává vždy jen do jedné sousední buňky, dochází často ke koncentraci vodních sítí a síť odtoku pak neodpovídá skutečnosti. Na druhou stranu je jeho výhodou jeho jednoduchost a fakt, že nedochází k nepochtené disperzi, jako je tomu u druhého algoritmu.

Druhým algoritmem je MD8, neboli multiple flow direction algorithm. MD8 algoritmem je voda odváděna do všech níže položených sousedních buněk a poměr vody je rozdělen procentuálně podle sklonu. Ve výsledku má algoritmus tendenci vytvářet více realistické prostorové sítě toku než D8 tím, že nedochází k takové koncentraci vody v síti. U D8 má mnohem větší vliv malý rozdíl výšek mezi sousedními buňkami na to, která buňka přijme veškerou vodu. Použitím MD8 tyto malé výškové rozdíly nemají zdaleka takový vliv, jelikož je množství vody rozděleno poměrem mezi tyto buňky. Největší nevýhodou tohoto algoritmu je jeho přílišná disperze na konvergentních svazích v důsledku odtoku do všech níže položených buněk.

V roce 1997 byl navržen algoritmus pojmenovaný D_{∞} Davidem G. Tarbotonem z americké univerzity v Utahu. Byl navržen přístup, který je založen na osmi trojúhelníkových ploškách, viz. obrázek 3. Díky tomuto postupu jsou odstraněny limitace v podobě možnosti pouze osmi směrů odtoku z buňky. Proto je algoritmus pojmenován D_{∞} , jelikož může nastat nekonečně možností výsledného směru odtoku. Tento přístup umožňuje pouze jeden směr odtoku, rozdělený mezi jednu nebo dvě buňky v závislosti na směru odtoku.

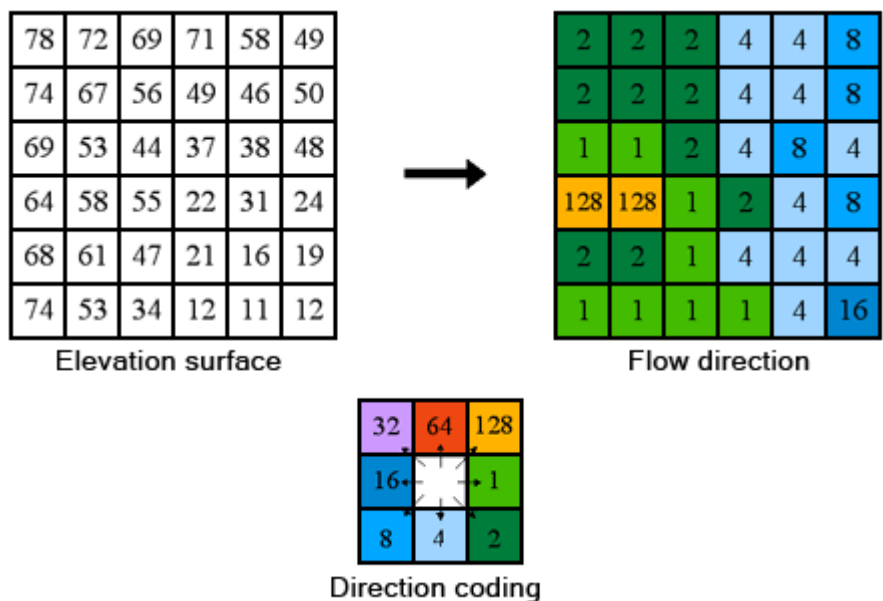
Poslední dva zmíněné algoritmy MD8 a D_{∞} mají oproti D8 nevýhodu ve větší časové náročnosti výpočtu. V kapitole 4 je popsán algoritmus MD_{∞} , který využívá výhody MD8 algoritmu za použití trojúhelníkových plošek obdobně jako D_{∞} algoritmus.

3 Flow Direction D8

Pro pochopení problematiky a zjevnost rozdílů mezi algoritmy je vhodné uvést, jak funguje nástroj, který používá ArcGIS. Základ algoritmu MD_{∞} , o němž pojednává tato dokumentace, je založen na podobných postupech.

3.1 Obecný princip D8

Princip nástroje určení směru odtoku (Flow Direction), který používá prostředí ArcGIS je následující. Tento nástroj (tool) vezme vstupní rastr digitálního modelu terénu, neboli výškový rastr. Provede výpočet, jehož výstupem je rastr, který v každé buňce rastru obsahuje hodnotu určující směr odtoku z této buňky. Hodnot je osm, podle osmi sousedních buněk, do kterých může odtékat voda, viz. obrázek 1. Právě podle osmi směrů odtoku se tento přístup nazývá D8 (eight-direction approach).



Obrázek 1: Flow Direction D8 [3]

3.2 Popis algoritmu D8

Samotný výpočet algoritmu je v zásadě primitivní. Pro všechny sousední buňky se vypočte maximální sklon [3]:

$$\text{maximální_sklon} = \text{rozdíl_výšek} / \text{vzdálenost} * 100$$

Rozdíl výšek je rozdíl výšky v aktuálním bodě a konkrétní sousední buňky. Vzdálenost je vypočtena mezi středy buněk a liší se v závislosti na poloze sousední buňky. Pro buňky sousedící vrcholem je hodnota velikosti pixelu přenásobena o odmocninu ze dvou. Pro zbylé buňky sousedící hranou je velikost vzdálenosti rovna velikosti pixelu. Po nalezení nejstrmějšího směru odtoku je hodnota uložena v podobě bitové hodnoty, reprezentující daný směr. Před použitím nástroje Flow Direction se obvykle zpracuje vstupní rastr pomocí nástroje Sink. Tento algoritmus zaplní bezodtoké oblasti rastru nejnižší hodnotou z okolí a do této nejnižší sousední buňky pak pošle vodu. Poté již nemůže dojít k situaci, kdy všechny sousední buňky k dané buňce mají vyšší hodnoty výšek a nedocházelo by tedy k odtoku.

Výsledný rastr nemusí obsahovat pouze buňky s hodnotami $2^0, 2^1, \dots, 2^7$ [4]. Jestliže nastane situace, kdy má více sousedních buněk nejnižší hodnotu výšek, pak je výsledná hodnota součtem všech směrů. Např. hodnota maximálního sklonu bude stejná pro sousední buňku vlevo a nahore, odtok tedy bude směrem na západ (hodnota = 16) a na sever (hodnota = 64), výsledná hodnota bude součtem 16 a 64, tedy 84.

4 Popis algoritmu MD_∞

4.1 Obecný princip MD_∞

Obecný postup je převzat z [1]. Stejně jako v případě algoritmu D_∞ byly zavedeny trojúhelníkové plošky pro výpočet lokálního sklonu a směru pro každou trojúhelníkovou plošku. Okolo středového bodu M bylo vytvořeno osm trojúhelníků, viz. obrázek 3. Každý se zbývajícími dvěma vrcholy $P1, P2$ ve dvou sousedících buňkách. Pro každý z těchto trojúhelníků je vypočten směr největšího sklonu. Pokud označíme výšky bodů $M, P1, P2$ jako h_M, h_{P1}, h_{P2} , rozdíly výšek se vypočtou takto:

$$z_1 = h_{P1} - h_M, z_2 = h_{P2} - h_M \quad (1)$$

Obdobně se vypočtou rozdíly v x-ových a y-ových souřadnicích. Po výpočtu rozdílu jednotlivých souřadnic je vypočten normálový vektor:

$$n = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = \begin{pmatrix} z_1 y_2 - z_2 y_1 \\ z_1 x_2 - z_2 x_1 \\ y_1 x_2 - y_2 x_1 \end{pmatrix} \quad (2)$$

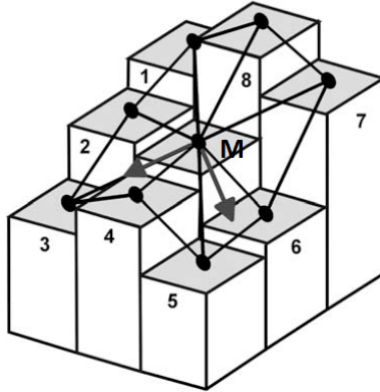
Směr odtoku d a sklon s na trojúhelníkové ploše se vypočte pomocí rovnice 3. Hodnota nula pro d označuje směr osy y a hodnota $3\pi/2$ označuje směr osy x .

$$d = \begin{cases} 0, & n_x = 0, n_y \geq 0 \\ \pi, & n_x = 0, n_y < 0 \\ \pi/2 - \arctan\left(\frac{n_y}{n_x}\right), & n_x > 0 \\ 3\pi/2 - \arctan\left(\frac{n_y}{n_x}\right), & n_x < 0 \end{cases} \quad (3)$$

$$s = -\tan\left(\arccos\left(\frac{n_z}{\sqrt{n_x^2 + n_y^2 + n_z^2}}\right)\right) \quad (4)$$

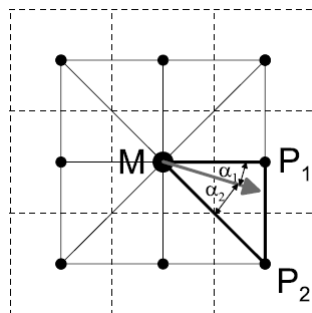
Jestliže největší sklon od bodu M je mimo rozpětí $0-45^\circ$ ($\pi/4$ radián), je použit sklon strmější k jednomu z bodů $P1, P2$ a zároveň i sklon je spočten k tomuto bodu. V případě, že oba sousední body jsou výše položeny než bod M , směr není pro tento trojúhelník uvažován. Po spočtení všech osmi směrů a sklonů pro jednotlivé plošky, jsou tyto směry považovány za lokálně nejstrmější pro každý trojúhelník v rozsahu 45° . Směry vedoucí přímo do bodu

$P1$ nebo $P2$, tedy ty, kde $d = 0$ nebo $d = \pi/4$ jsou uvažovány pouze v případě, kdy obě vedlejší sousední buňky mají směr k bodu $P1$ nebo $P2$. Tato situace je znázorněna na obrázku 2.



Obrázek 2: Ukázka určení směrů odtoku z buňky [1]

První případ je směr mířící mezi buňky 5 a 6. Druhý případ je směr, který vede přímo do buňky 3. Tato situace nastane, jestliže pro trojúhelník M43 vyjde lokální směr $d = \pi/4$ a pro trojúhelník M32 směr vyjde $d = 0$. Poté, co je směr odtoku pro buňku vypočten, je celkový odtok rozdělen do všech výsledných směrů na základě sklonu. Řešení této problematiky je více popsáno v sekci 4.2. Pro směry mířící mezi body $P1$ a $P2$ je odtok rozdělen do dvou buněk na základě největšího sklonu. Poměr odtoku do buněk s body $P1$ a $P2$ bude v tomto pořadí $\alpha_2/45^\circ$ a $\alpha_1/45^\circ$.



Obrázek 3: Trojúhelníkové plošky [1]

4.2 Technické řešení MD_{∞}

Výše již bylo zmíněno, že algoritmus je napsán v jazyce Python. Algoritmus je rozdělen do dvou skriptů. První z nich obsahuje funkce a druhý je hlavní. Vstupem do modelu je rastr digitálního modelu terénu a výstupní adresář, do kterého se uloží finální rastr směru odtoků, viz. obr. 4. Po načtení rastru ze vstupních parametrů je tento rastr převeden na matici NumPy pomocí funkce:

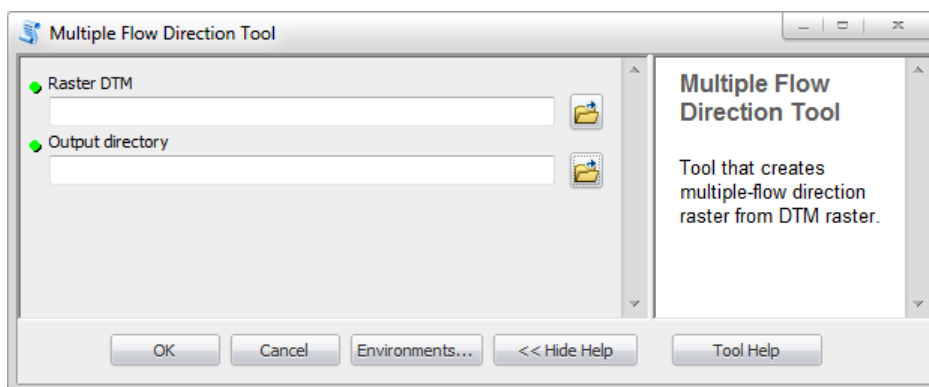
```
arcpy.RasterToNumPyArray( InputRaster ).
```

NumPy [5] je rozšíření programovacího jazyka Python, je to balíček užívaný pro vědecké výpočty. Umožňuje podporu velkých, multi-dimenzionálních polí a matic, spolu s velkou knihovnou matematických funkcí pro práci s těmito poli.

Algoritmus cyklem projíždí všechny prvky matice a u těch, které mají hodnotu různou od NoData nalezne okolí. Hledání okolí buňky provádí funkce *neighbors*, ta je definována takto:

```
def neighbors( i , j , array , x , y ) :  
    ...  
    return nb1 , nb2 , nb3 , nb4 , nb5 , nb6 , nb7 , nb8
```

Vstupem funkce je pozice v matici, dále celá matice a její rozměry. Funkce *neighbors* vypočítá hodnoty výšek sousedních osmi buněk a vrátí je na výstup.



Obrázek 4: Dialogové okno nástroje MD_{∞}

Výstup z funkce *neighbors* slouží jako jeden z vstupních parametrů pro další funkci.

Funkce byla pojmenována *dir_slope*, jelikož jejím výstupem je směr odtoku a sklon tohoto směru.

```
def dir_slope (pointM , nbrs , vpix , spix ) :
    ...
    return direction , slope
```

Pro každý trojúhelník byla spočtena hodnota d , viz. vzorec 3. Hodnota s byla po špatných a nelogických výsledcích počítána podle jiného vzorce, který již dával správné výsledky:

$$s_0 = \sqrt{\frac{z_1^2}{2y_1^2} + \frac{z_2^2}{y_2^2}} \quad (5)$$

Vzorec se liší pro každý trojúhelníček, uvedený je pro výpočet sklonu, kdy sousední buňka je severozápadní a severní. Tímto postupem bylo zbytečné počítat z-tovou složku vektoru n , vzorec 2. Byly ošetřeny případy, kdy sousední buňky aktuálního počítaného trojúhelníku jsou obě výše položené, nebo obě mimo rastr (NoData). Další případ nastane, jestliže je jeden bod vně rastru a druhý v rastru leží. Směr je potom určen přímo do toho bodu, který náleží rastru. Taktéž i sklon je počítán pouze mezi tímto bodem a bodem M . Lokální směr d v každé plošce po vypočtení nesmí překročit 45° . V situaci, že tuto hodnotu překročí, je směru přiřazena hodnota 0 nebo 45° v závislosti na vzájemném porovnání výšek bodů v trojúhelníku. Takto jsou zpracovány všechny plošky a je zjištěno, zda-li do každého trojúhelníku nastává směr odtoku nebo ne. Na konci funkce *dir_slope* se kontroluje případ, kdy směr sousedních trojúhelníků odkazuje přesně do stejného bodu, tedy hodnoty směrů musí být pro první trojúhelník $d = 45^\circ$ a pro následující sousední trojúhelník $d = 0$. Výstupem funkce jsou dva vektory o osmi prvcích, směr a sklon pro každou plošku.

Další fází výpočtu je zjištění poměru vody, která poteče do jednotlivých směrů. K výpočtu poměrů byl použit tento vzorec [2]:

$$f_i = \frac{s_i^x}{\sum_{j=1}^8 s_j^x} \quad (6)$$

, kde s vyznačuje sklon vypočteného směru konkrétní trojúhelníkové plošky a x je proměnný exponent. Pokud by byl exponent zvolen 1, bude výsledná síť obdobná jako při použití MD8 algoritmu. Při zvětšování exponentu x k nekonečnu, bude síť stejná jako při použití D8 metody. Výchozí hodnota exponentu byla zvolena $x = 4$, což je kompromis mezi oběma variantami.

Změna exponentu se v současné podobě neprojeví na výsledném rastru, jelikož výsledek prezentuje pouze směr odtoku a ne procentuální část. Tohoto poměru bude využito až při spojení skriptu s 2D modelem SMODERP.

V další části výpočtu je rozdělen již zjištěný poměr z celkového odtoku do dvou, případně pouze do jedné buňky v závislosti na směru odtoku d . Ve výsledku je vypočten vektor osmi čísel, kde každá hodnota od 0 do 1 představuje již procentuální poměr z celkového odtoku do každé sousední buňky.

Pro výslednou reprezentaci směru odtoku na výstup byl pro každou buňku vytvořen vektor obsahující 1 a 0 říkající, zda nastává či nenastává odtok do sousední buňky. Tento vektor byl převeden na celočíselnou hodnotu pomocí funkce *bool2int*:

```
def bool2int(x):
    y = 0
    for i, j in enumerate(x):
        if j: y += 1<<i
    return y
```

Funkce má na vstupu vektor jediček a nul. Například pro vektor 00000111, kdy voda teče na severozápad, sever a severovýchod, se funkce chová takto. Je testováno, jestli platí $j = 1$, tedy prvních pět nul vektoru funkce přeskočí a pro první jedničku se přičte k proměnné y 2^0 , v dalším kroku 2^1 a v posledním 2^2 . Operátor \ll vyjadřuje totiž bitový posun směrem doleva. Výsledkem je na výstupu číslo $y = 1 + 2 + 4 = 7$.

Pro všechny buňky byla vypočtena hodnota reprezentující směr odtoku. Tato hodnota se uloží do matice a ta se následně převede na výsledný rastr.

```
flow_dir = arcpy.NumPyArrayToRaster(value, LLcorner,
    spix, vpix, "#")
```

<i>value</i>	hodnota reprezentující směr odtoku,
<i>LLcorner</i>	souřadnice levého dolního rohu,
<i>spix</i>	šířka pixelu,
<i>vpix</i>	výška pixelu.

5 Výsledky

Algoritmus MD_{∞} byl testován na skutečných datech dvou pozemků u obcí Nučice ve Středočeském kraji a Býkovice v kraji Jihomoravském, rastry jsou v příloze 7.1. Oba dva rastry jsou o rozlišení 5 metrů. Finální rastry jsou porovnávány s rastry vytvořenými v prostředí ArcGIS nástrojem Flow Direction metodou $D8$.

5.1 Reálná data - Býkovice

Výsledné rastry Flow Direction pro území Býkovice jsou uvedeny v příloze 7.2. Oba rastry byly pro hodnoty $2^0, 2^1, \dots, 2^7$ obarveny shodnou barvou kvůli lepší názornosti a porovnatelnosti výsledků. Rastr, který obsahuje z celkového počtu 44807 buněk celkem 23655 buněk s hodnotami výšek po proběhnutí algoritmy, dal následující výsledky:

FD	D8	MD_{∞}
1	3650	737
2	666	917
4	207	18
8	21	7
16	94	37
32	791	85
64	8818	1045
128	9408	14983
Σ	23655	14983

Tabulka 1: Tabulka porovnání obou algoritmů pro rastr Býkovic

FD je zkratka pro Flow Direction, čili směr odtoku reprezentovaný celočíselnou hodnotou. Ostatní hodnoty u výsledného rastru, na který byl použit algoritmus MD_{∞} , jsou v rozpětí $0, \dots, 246$. V rozmezí hodnot směrů odtoku $246, \dots, 255$ se nevyskytují žádné buňky. Čas potřebný pro vykonání skriptu byl v případě $D8$ 4 vteřiny. Pro algoritmus MD_{∞} byl čas 15 vteřin.

5.2 Reálná data - Nučice

Výsledné rastry jsou uvedeny v příloze 7.3. Opět bylo použito shodné obarvení pro základní směry u obou rastrů. Rastr Nučice obsahoval data na 82732 buňkách z celkové počtu 125449 buněk. Výsledky jsou pro přehlednost uvedeny v tabulce:

FD	D8	MD ∞
1	12186	1233
2	12441	8703
4	16907	1027
8	9826	16423
16	2516	288
32	5010	2417
64	7788	366
128	16058	25475
Σ	82732	55932

Tabulka 2: Tabulka porovnání obou algoritmů pro rastr Nučic

Ostatní hodnoty u výsledného rastru, na který byl použit algoritmus MD ∞ , jsou v rozpětí 0,...255. Čas potřebný pro vykonání skriptu byl v případě D8 6 vteřin. Pro algoritmus MD ∞ byl čas 50 vteřin.

6 Závěr

Cílem projektu bylo vytvořit funkční skript obsahující algoritmus pojmenovaný MD_{∞} [1], který řeší odtok do sousedních buněk rastru. Algoritmus byl napsán v programovacím jazyce Python a testován na reálných datech. Výsledky byly mírným překvapením, kdy zejména hodnoty směru 128, tedy severovýchodního se vyskytovaly ve výsledku velmi často, viz. tabulky 1 a 2. Předpokladem bylo, že se odtok rozloží do více směrů a počet buněk s hodnotami základních směrů 1,2,4,8,16,32,64 a 128 bude méně než v případě algoritmu $D8$. K rozkladu do více směrů sice došlo, ale počet některých buněk se základními směry byl větší u algoritmu MD_{∞} .

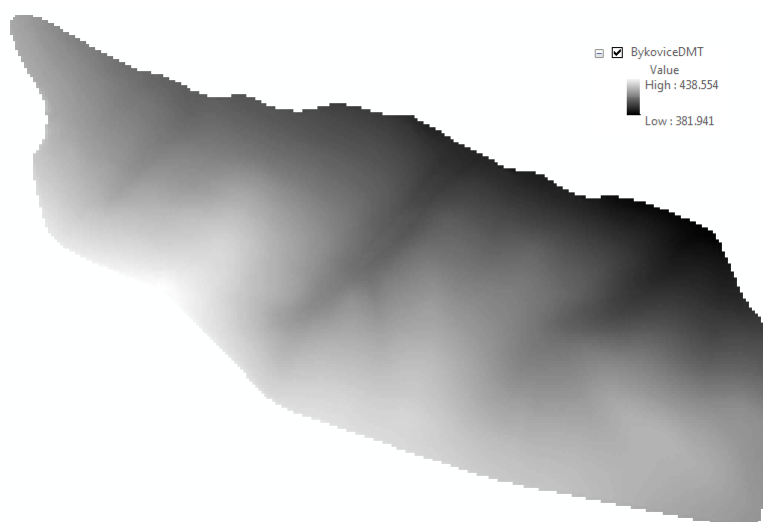
Co se týče délky doby běhu algoritmu je algoritmus MD_{∞} pomalejší, viz. časy v kapitole 5, než $D8$ z několika zřejmých důvodů. Především algoritmus $D8$ provádí mnohem méně početních operací z důvodu jeho menší složitosti, viz. 3. Dalším důvodem je také rozdíl v jazyce, ve kterém jsou oba algoritmy napsány. $D8$ je napsán v programovacím jazyce C++, který na větších datech je mnohonásobně rychlejší než Python. Přepsáním algoritmu MD_{∞} do jazyka C++ by mohla být cesta k budoucímu zvýšení rychlosti práce algoritmu.

Reference

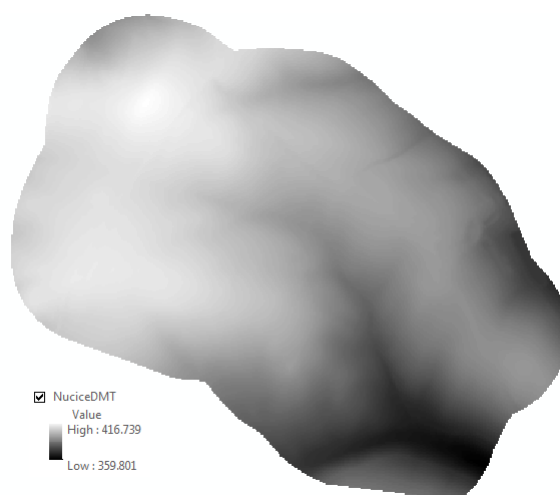
- [1] SEIBERT Jan, McGlynn B.L.: *A new triangular multiple flow direction algorithm for computing upslope areas from gridded digital elevation models* [online], [cit. 13.5.2013]. Dostupné ke stažení z URL: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.79.977&rep=rep1&type=pdf>>.
- [2] HOLMGREN, Peter: *Multiple direction algorithms for runoff modelling in grid based elevation models: An empirical evaluation* [online], [cit. 13.5.2013]. Dostupný z URL: <[ftp://www.lwr.kth.se/Common/UllaM/For_Imran/\[1994\]%20holmgren%20-%20multiple%20flow%20direction%20algorithms%20for%20runoff%20modelling%20in%20grid%20based%20elevation%20models.pdf](ftp://www.lwr.kth.se/Common/UllaM/For_Imran/[1994]%20holmgren%20-%20multiple%20flow%20direction%20algorithms%20for%20runoff%20modelling%20in%20grid%20based%20elevation%20models.pdf)>.
- [3] ESRI: *ESRI WebHelp - How Flow Direction works*: [online], [cit. 13.5.2013]. Dostupné z URL: <<http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/009z00000063000000.htm>>.
- [4] ESRI: *ESRI Webhelp - Flow Direction (Spatial Analyst)*: [online], [cit. 13.5.2013]. Dostupné z URL: <<http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/009z00000052000000.htm>>.
- [5] WIKIPEDIA: *NumPy*: [online], [cit. 13.5.2013]. Dostupné z URL: <<http://en.wikipedia.org/wiki/NumPy>>.
- [6] INFORMIT: *Working with operators*: [online], [cit. 13.5.2013]. Dostupné z URL: <<http://www.informit.com/articles/article.aspx?p=2002831&seqNum=6>>.

7 Přílohy

7.1 Býkovice a Nučice vstupní DMT rastry

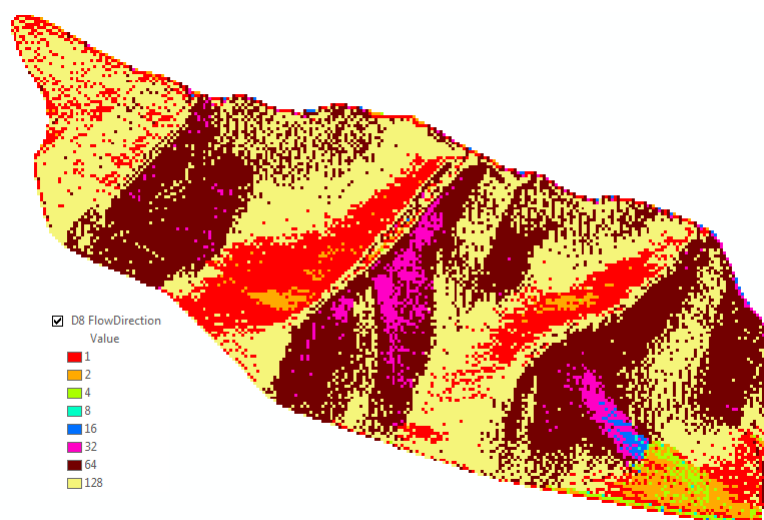


Obrázek 5: DMT Býkovice

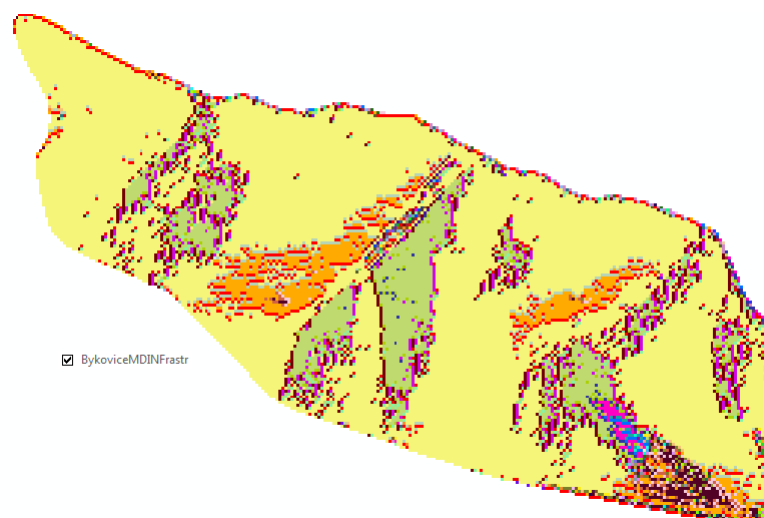


Obrázek 6: DMT Nučice

7.2 Býkovice výsledky

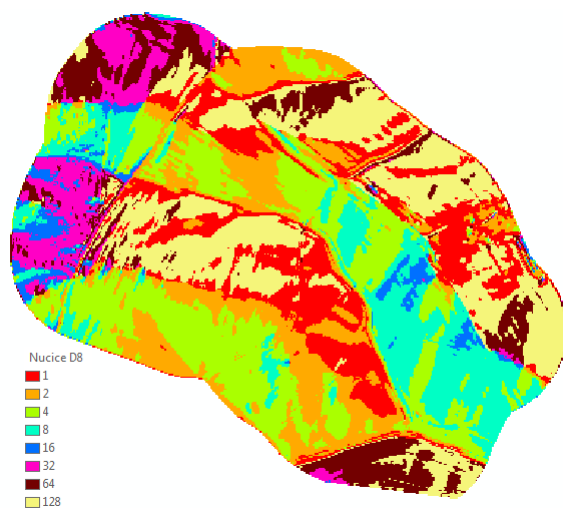


Obrázek 7: Flow Direction D8

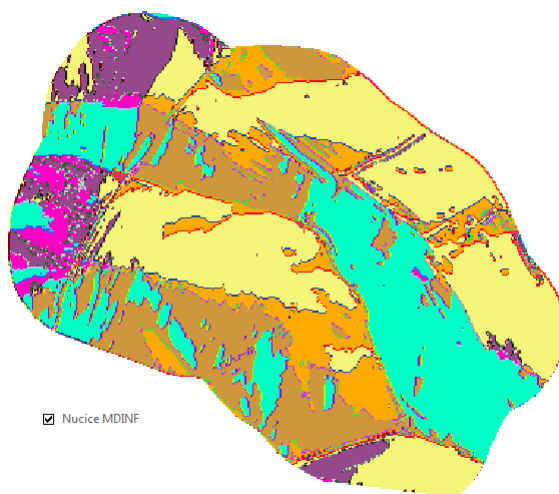


Obrázek 8: Flow Direction MD∞

7.3 Nučice výsledky



Obrázek 9: Flow Direction D8



Obrázek 10: Flow Direction MD ∞