

Univerzita Palackého v Olomouci

Přírodovědecká fakulta

Katedra geoinformatiky



Jan BOJKO

**STATISTICKÉ MODELOVÁNÍ
HYDROLOGICKÝCH JEVŮ**

diplomová práce

Vedoucí práce: Mgr. Pavel TUČEK

Olomouc 2009

Prohlašuji, že jsem zadanou diplomovou práci vypracoval sám, a že jsem uvedl veškerou použitou literaturu.

Hrádek, 11.5.2009

.....

Děkuji Mgr. Pavlu Tučkovi za vedení mé diplomové práce, zvláště za poskytnutí statistického modelu a pomoc se statistickými analýzami.

Obsah

Obsah.....	1
1 Úvod.....	4
2 Cíle.....	5
3 Metody.....	6
3.1 Teoretické předpoklady.....	6
3.2 Modely.....	6
3.3 Software.....	6
3.4 Typografické konvence.....	7
4 Postup zpracování.....	8
4.1 Teoretická část.....	8
4.2 Praktická část.....	8
5 Hydrologie.....	9
5.1 Proudění vody.....	9
5.1.1 Vizualizace proudění.....	10
5.1.2 Laminární proudění.....	10
5.1.3 Rovinné proudění.....	11
5.1.4 Turbulentní proudění.....	11
5.1.5 Viskozita.....	12
5.1.6 Reynoldsovo číslo.....	13
5.1.7 Rozložení rychlosti proudění.....	13
5.2 Koryto.....	14
5.2.1 Tvar koryta.....	14
5.2.2 Profil koryta.....	14
5.3 Hydrometrie.....	14
5.3.1 Měření hloubek.....	15
5.3.2 Měření vzdáleností.....	15
5.3.3 Měření rychlosti proudění.....	16
6 Data a modely.....	18
6.1 Data.....	18
6.1.1 Body koryta.....	18
6.1.2 Bodové rychlosti.....	19
6.2 Kriging.....	20
6.2.1 Prostorová proměnná.....	20

6.2.2 Variogram.....	21
6.2.3 Modely variogramu.....	22
6.2.4 Teorie Krigingu.....	23
6.3 Rychlost proudění v řece.....	24
6.3.1 Břehové linie (2D).....	24
6.3.2 Proudnice.....	25
6.3.3 Úbytek rychlosti proudění.....	26
6.3.4 Hledání projekce k odhadovanému bodu.....	27
7 Softwarové prostředky.....	29
7.1 Matlab.....	29
7.1.1 Nearest neighbour (nejbližší soused).....	29
7.1.2 Poly2mask.....	30
7.1.3 Práce s polynomy.....	30
7.1.4 Skripty a funkce	31
7.1.5 Kompilace.....	31
7.2 R.....	32
7.3 C++.....	32
7.3.1 Třídy.....	33
7.3.2 Abstraktní základní třída.....	33
7.4 Knihovna Qt.....	35
7.4.1 Hierarchie objektů knihovny Qt.....	35
7.4.2 QVector.....	36
7.4.3 Modularita.....	36
7.4.4 QProcess.....	36
8 VOR.....	37
8.1 Instalace.....	37
8.2 Koncept.....	38
8.3 Funkcionalita.....	40
8.4 Nástroje.....	42
8.4.1 Vizualizace.....	42
8.4.2 Interpolace.....	42
8.5 Jiné přístupy.....	44
9 Aplikace modelů.....	45
9.1 Kriging	46
9.2 Model rychlosti proudění v řece.....	49

10 Výsledky.....	54
11 Diskuze.....	56
12 Závěr.....	57
13 Summary.....	58
Literatura.....	59
Seznam příloh.....	61

1 Úvod

V přírodě probíhá značná část procesů a pochodů, které jsou na hranici lidského chápání. Abychom tyto složité děje lépe pochopili, vytváříme jejich modely. Model jako takový se snaží postihnout a popsat čím jak největší počet podmínek, které ovlivňují reálný proces a vystihují tak jeho chování.

V diplomové práci Statistické modelování hydrologických jevů se zabýváme stochastickým modelem pro zjištění rychlosti proudění v kterémkoliv bodě sledovaného říčního koryta. Model se snaží odhadnout danou rychlost na základě měřených bodů čistě statisticky, a proto není uvažovaný model kompletním popisem situace, ale je nucen celý jev značně zjednodušit.

Zadaná práce z velké části přejímá matematické vyjádření problému a snaží se posunout vývoj modelu do fáze verifikace. Podstatnou částí práce je tedy výpočetní zpracování zadaných matematických tezí, které se opírají o výzkum v této oblasti na univerzitě Bodon Kultur (BOKU) v Rakousku.

2 Cíle

Rozložení rychlosti proudění v celém úseku říčního koryta vychází ze složitých procesů a je ovlivněno celou řadou faktorů. Pouhé vizuální sledování řeky nestačí k pochopení těchto procesů, a proto je nutné interpretovat velké množství naměřených dat. Naměřit rychlost ve všech místech určitého koryta pomocí hydrometrických nástrojů je velice časově a finančně náročné. Existují však statistické modely, které se pokouší ze sady náhodně či pravidelně rozmístěných měření odhadnout situaci v celém říčním úseku. Tyto statistické modely často nejsou aplikovány na reálná data, což snižuje jejich využitelnost v praxi.

Naším prvním cílem je tedy za pomoci statistických modelů odhadnout rychlost proudění z naměřených dat v kterémkoliv bodě sledovaného říčního koryta. Jeho součástí je zpracování odborné literatury a hlavně statistických modelů do podoby algoritmu.

Dalším cílem je realizování tohoto algoritmu s pomocí výpočetní techniky a vytvoření programového kódu, který bude dále editovatelný a přístupný pro potřeby vývoje modelu. Výstupem této části je samostatná a modulární aplikace.

Matematické a statistické odhady jsou často pouhou změtí čísel, a tak bývá jejich interpretace velice obtížná. Třetím cílem této práce je proto vytvořit vhodný výstup, který bude zobrazovat situaci v korytě. Tato vizualizace se stane prvním konkrétnějším pohledem na výstupy ze sledovaného modelu rychlosti proudění v říčním korytě.

3 Metody

3.1 Teoretické předpoklady

Tato diplomová práce se opírá o celou řadu dříve popsanych faktů z oblasti hydrologie a zvláště pak hydrodynamiky. Zjištěné informace se staly základem pro správné aplikování a interpretování zadaného modelu. V hydrologické části patří mezi nejdůležitější poznatky chování laminárních proudů a jejich vliv na rozložení rychlosti proudění v otevřeném korytě, jelikož se dále popsané modely snaží tyto dynamické procesy vyjádřit pomocí statistiky.

3.2 Modely

V práci byly využity dva statistické modely pro odhad prostorového rozmístění zkoumaného jevu na základě naměřených dat. Prvním modelem je **Kriging**, který se hojně využívá u geostatistického modelování. Jeho „síla“ odhadu je nesporná, jelikož se snaží odhadovat hodnoty na základě nejmenší chyby. Kriging má při modelování rozložení rychlosti proudění některé nedostatky, i přesto se stal v první fázi této práce důležitým nástrojem pro pochopení zkoumaného jevu.

Model **Rychlost proudění v řece**, kterým se zabývají Mgr. Pavel Tuček a Mgr. Jaroslav Marek Phd., je speciálně zaměřen na odhad rychlosti v říčním korytě. Jeho výhodou při modelování je fakt, že bere v potaz zákonitosti, které jsou pro proudění specifické, a tak je „šitý“ na míru problematice laminárního proudění. Celý model je stále ve vývoji, a proto je logicky dosti zjednodušený oproti reálné situaci.

3.3 Software

Programový kód byl vytvářen za pomoci několika komerčních i nekomerčních softwarů, jejichž výběr nebyl podroben systematickému testování, ale které byly vybírány podle potřeb práce.

Kriging byl zpracován v programu R, který obsahuje velice kvalitní krigovací nástroj z knihovny geoR. Knihovna je volně dostupná v systému balíčků¹, které mohou být staženy

¹ Balíček v tomto kontextu představuje modul do aplikace, který se po stáhnutí automaticky nainstaluje.

prostřednictvím internetového spojení přímo z aplikace. Kriging je v programu R řešen pouze přes příkazovou řádku, avšak pro autora i vedoucího práce bylo jeho ovládání intuitivnější než v obdobných nástrojích jiných softwarů.

Aplikace VOR, která je jedním z výstupů práce, byla programována v C++ s využitím knihovny Qt od společnosti Nokia. Qt je plnohodnotný C++ framework pro tvorbu aplikací s grafickým uživatelským prostředím, který je šířen od verze 4.5 pod komerční, LGPL a GPL licenci.

Programování složitých matematických rovnic je prováděno v komerčním softwaru Matlab, který pro účely diplomové práce poskytla Katedra matematické analýzy a aplikací matematiky. Matlab obsahuje v základní instalaci řadu propracovaných funkcí pro jednotlivé matematické operace a jejich vizualizace, ale je také rozšiřitelný o celou řadu toolboxů, které ho dále vylepšují.

Bližší charakteristika použitých softwarů je popsána v kapitole 7.

3.4 Typografické konvence

Syntaxe, proměnné a parametry zdrojového kódu v textu:

```
plot(x, y)
```

Komentáře jsou ve zdrojových kódech uvozeny %, příkazová řádka >>, zdrojové kódy a jejich příklady vypadají následovně:

```
% Vykreslení promenných A a B
>> A = 1; B = 2;
>> plot(A,B);
```

Matematické vzorce budou mít následující podobu:

(N)
$$c^2 = a^2 + b^2$$
,

kde N je pořadové číslo vzorce.

V celé práci budeme pro desetinnou čárku používat tečku, jelikož oba použité softwary pracují s touto podobou reálných čísel.

4 Postup zpracování

4.1 Teoretická část

V úvodní části textu je zpracována teorie o proudění, faktorech, které ho ovlivňují, rozložení rychlosti v úseku koryta, morfologii koryta, základech hydrometrie atd. Autor práce si je vědom, že se jedná o problematiku velice složitou, a proto byly vybrány pouze nejdůležitější informace, které mají bezprostřední vliv na pochopení modelu.

Druhým krokem při zpracování zadaného úkolu bylo popsání dostupných datových sad na základě jejich zobrazení a formátu uložení, zvláště pak umístění jednotlivých měření v říčním korytě. Dále byly definovány dva modely pro statistické odhady.

Teoretická část pokračuje popisem jednotlivých programů a programovacích jazyků, které byly použity. Byly nastíněny důležité nástroje a postupy pro realizaci výsledných algoritmů. Programový kód je popisován spíše abstraktně a obecně, aby si čtenář udělal představu o jeho koncepci a nebyl zahlcen kvantem příkazů, skriptů, funkcí, definic tříd atd.

V neposlední řadě je popsáno vytvoření programu VOR, u kterého je zmíněn postup instalace, jeho funkcionalita, nástroje použité při modelování rychlosti proudění a zjednodušený UML model závislosti jednotlivých částí programu.

4.2 Praktická část

Praktická část obsahuje algoritmy pro vytvoření výstupů a je popsána v kapitole 9, která je rozdělena do dvou podkapitol.

V první podkapitole je popsán postup pro vytvoření výstupu v podobě gridu metodou Kriging. Tato kapitola popisuje „rozkouskovaný“ skript programu R, který je nastaven podle zadaných dat. Jednotlivé části jsou v textu seřazeny a strukturovány tak, aby vyčlenily nejdůležitější kroky při nastavení jednotlivých vlastností tohoto modelu. Výstupy z této části jsou v podkapitole popsány a interpretovány na základě dostupných teoretických předpokladů.

Poslední podkapitola obsahuje postup zpracování dat metodou popisovanou v kapitole 6.3, která popisuje aplikování vyvíjeného statistického modelu na reálná data. Za pomoci programu VOR je celý postup modelování postupně popsán a vizualizován do podoby gridu.

5 Hydrologie

Hydrologie jako vědní obor má velice široký záběr zkoumání, studuje vodu v atmosféře, řekách, jezerech, bažinách, podzemních vodách atd. V této kapitole se budeme zabývat výhradně hydrologií řek, která se mimo jiné zabývá pohybující se vodou v korytech řek. Pohyb vody v korytě je složitý proces, který je ovlivněn celou řadou faktorů, které působí z vnějšího i vnitřního prostředí. Mezi vnější faktory patří drsnost koryta a jeho morfologie, okolní atmosféra nebo ledová pokrývka. Mezi vnitřní faktory pak hlavně viskozita.

Podmínkou pro pochopení některých složitějších pochodů je zpracování a vizualizace diskrétních dat, která se většinou získávají přístrojovým měřením. Měřením různých veličin v korytě se dále zabývá hydrologická disciplína s názvem hydrometrie. V práci budou popsány pouze základní postupy měření dat, které jsou potřebné pro náš výzkum. Kapitola bude dále popisovat proudění vody obecně a typy proudění, rozložení rychlosti proudění, koryto řeky atd.

5.1 Proudění vody

Řeky jsou přírodní vodní proudy, které tečou napříč územím z míst o vyšší nadmořské výšce do míst o nižší nadmořské výšce (HUGGETT, 2003). Proudění vody v otevřených korytech je podmíněno **gravitační a třecí silou**. Gravitace „nutí“ vodu stékat dolů ze svahu, zatímco třecí síla působí proti ní odporovou silou. Gravitační síla vždy působí ve směru vertikálním, a pokud se koryto svažuje, stává se hlavní hnací silou a je rovna hmotnosti vodní masy. Existují dva druhy třecích sil v otevřených korytech. První je způsobena viskozitou a druhá třením tekoucí vody o povrch koryta. Velikosti těchto působících sil celkově ovlivňují proudění vody.

Existují dva druhy proudění vody v korytě – laminární a turbulentní. Jedním ze základních faktorů ovlivňujících druh proudění je viskozita kapaliny. Výpočet Reynoldsova čísla je jednou z početních metod, na jejímž základě se zjišťuje, jestli je proudění laminární nebo turbulentní.

5.1.1 Vizualizace proudění

Abychom mohli proudění popsat, musíme ho nějakým způsobem **vizualizovat**. V podkapitolách 5.1.2 a 5.1.4 budeme k vizualizaci proudění používat tzv. proudnice (MATOUŠEK, 2009 a BRDIČKA, 2000), které zobrazují dráhy pohybu jednotlivých pevných částic korytem v průběhu času. Proudnice je množina bodů, na niž jsou vektory rychlosti proudění v daném okamžiku tečné.

Tečna k libovolnému bodu na proudnici je rovna směru bodové rychlosti pohybu částice v kapalině. **Bodová rychlost** je uspořádaný vektor u , který je roven derivaci uspořádaného vektoru posunutí (změny polohy) s v čase t (čas posunutí), tzn. za jakou dobu urazí částice vzdálenost z bodu A do bodu B.

$$(1) \quad \boxed{u = \frac{ds}{dt}}$$

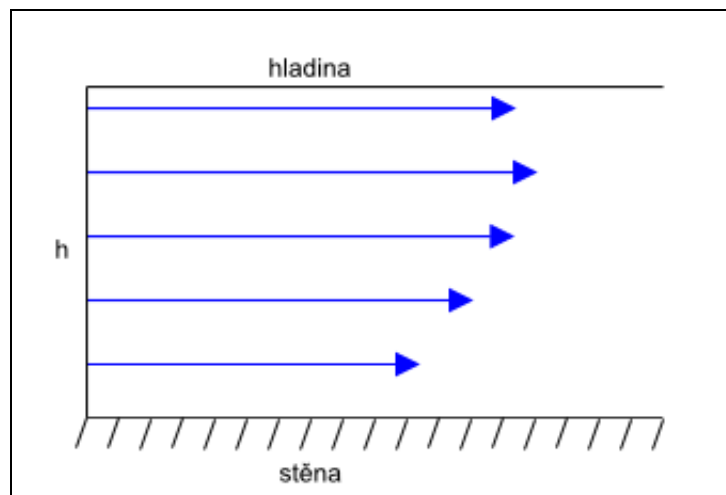
Proudnice tedy představují linie spojující body o stejné bodové rychlosti proudění v určitém okamžiku a zakreslují se pomocí šipek, které vyjadřují směr proudění. Rozdílem v délce šipek bude také vyjádřen rozdíl v bodové rychlosti proudění.

5.1.2 Laminární proudění

Laminární proudění (z latinského lamina: destička) je pohyb částic ve vrstvách, které se vzájemně nekříží. Mezi vrstvami nedochází k přenosu hmotných částic. Bodové rychlosti jsou neměnné v čase a jejich hodnota se mění pouze ve vzdálenosti od břehů a dna koryta.

Při laminárním proudění vzniká odpor mezi každou vrstvou, což je způsobeno molekulární viskozitou (HUGGETT, 2003). Největší odpor vzniká mezi vrstvou, která bezprostředně přiléhá k břehům nebo dnu koryta. Pro tuto vrstvu platí, že má nejmenší rychlost proudění. Vrstvy, které k této „nejspodnější“ vrstvě přiléhají, jsou zpomalovány již méně, a proto se směrem ode dna a břehů koryta rychlost proudění zvyšuje.

Obr. 1 znázorňuje rovnoběžný průběh částic, které jsou ovlivňovány třením o okolní prostředí. Můžeme si všimnout, že částice nejbližší hladiny je vlivem atmosféry zpomalována. Na obrázku jsou zanedbány některé další faktory, jako například sklon koryta.



Obr. 1: Zjednodušený nákres rovnoběžného průběhu proudnic v korytě s ohledem na rychlost proudění (upraveno z [12])

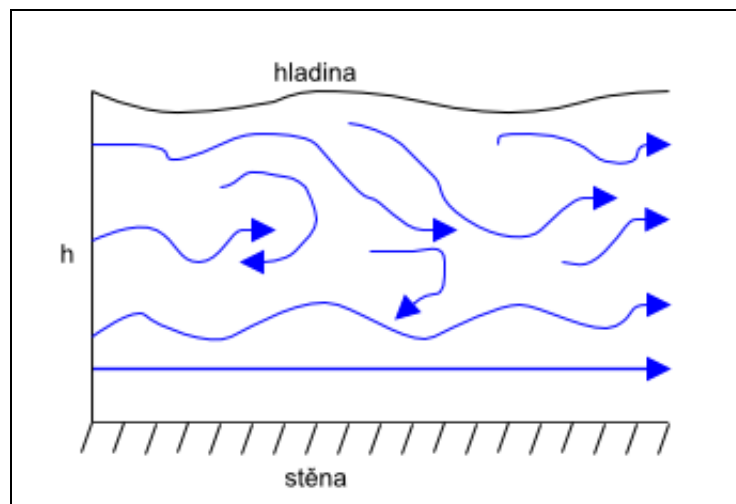
5.1.3 Rovinné proudění

Rovinným prouděním nazýváme takový pohyb tekutiny, při němž je vektor rychlosti všech částic stále rovnoběžný s pevnou rovinou a je stejný ve všech bodech kolmice na tuto rovinu (BRDIČKA, 2000). V těchto případech je proudění dvourozměrné (rovinné) a vyšetřuje se v rovině x, y .

V otevřeném korytě řeky bychom takto definované proudění hledali velice těžko, vzhledem k celé řadě působících faktorů. Pokud tedy chceme zkoumat rozložení rychlosti proudění v jednotlivých hloubkách (rovinách), je nutné volit studované úseky řeky tak, aby se alespoň blížily laminárnímu proudění. Budeme tedy studovat jednotlivé vrstvy v 3D prostoru jako rovinné proudění v ploše x, y .

5.1.4 Turbulentní proudění

Turbulentní proudění je převládajícím typem proudění v korytě toku (HUGGETT, 2003). Na vzniku tohoto proudění se podílí molekulární a turbulentní odpor mezi vrstvami, které se chaoticky kříží a vlní ve vodním toku, a dochází tak k výkyvům rychlosti proudění. Pohyb částic kapaliny je nepravidelný a neuspořádaný, dochází k míchání a vzniku vírů uvnitř proudu.



Obr. 2: Zjednodušený náčrt vírového průběhu proudnic v korytě s ohledem na rychlost proudění (upraveno z [12])

Turbulentní proudění vzniká z laminárního proudění ve chvíli, kdy vznikají nestability (roste smykové napětí) na rozhraní vrstev, což může být zapříčiněno vyšší rychlostí proudění, překážkou v korytě, tvarem koryta atd. Dynamická viskozita u vyšších rychlostí proudění nezvládá tlumit tyto nestability a dochází ke vzniku vírů, které ovlivňují i ostatní vrstvy. Turbulentní proudění se dále rozvíjí za předpokladu, že se zvyšuje rychlost proudění.

5.1.5 Viskozita

Tekutiny, oproti pevným částicím, nemají vlastní (stálý) tvar, jelikož jsou přitažlivé síly mezi částicemi mnohonásobně menší. Můžeme říci, že jejich částice jsou velmi pohyblivé a celkově je tedy kapalina např. v korytě velmi náchylná na změnu tvaru. Odpor tekutiny proti změně nazýváme viskozitou (BRDIČKA, 2000). Síla viskozity je přímo úměrná gradientu rychlosti, tzn. čím větší je rychlost proudění, tím má viskozita větší vliv na proudící kapalinu.

Viskozita ovlivňuje rozložení rychlosti ve vzdálenosti od pevných břehů. Newtonův zákon viskozity říká, že pokud se kapalina pohybuje ve směru x (směr toku), ve vzdálenosti y od břehu koryta a o rychlosti proudění v_x , tak platí, že je odpor ve směru x roven:

$$(2) \quad \tau_x = \mu \frac{d v_x}{d y},$$

kde μ představuje molekulární viskozitu kapaliny.

Viskozitu můžeme dělit na kinematickou a dynamickou (DOUGLAS, 1979). Koeficient dynamické viskozity může být definován jako odporová síla působící na jednotku plochy, která je potřebná na přesun jedné vrstvy o určité rychlosti přes druhou vrstvu v určité vzdálenosti od ní. Kinematická viskozita je podíl dynamické viskozity a hustoty kapaliny.

5.1.6 Reynoldsovo číslo

Reynoldsovo číslo je bezrozměrné číslo, které umožňuje určit, zda je proudění laminární nebo turbulentní (HUGGETT, 2003). Toto číslo je ovlivňováno následujícími vlastnostmi:

- rychlost
- hloubka
- hustota kapaliny
- viskozita

Může být vypočítáno z průměrné rychlosti v , hydraulického poloměru R a kinematické viskozity, která vyjadřuje poměr mezi hustotou kapaliny ρ a molekulární viskozitou μ .

$$(3) \quad Re = \frac{\rho v R}{\mu}$$

Laminární proudění má hodnotu Reynoldsova čísla menší než 500. Turbulentní proudění bylo potvrzeno při hodnotách větších než 2000. Pokud se číslo pohybuje mezi hodnotami 500 a 2000, to znamená, že jsou přítomny oba druhy proudění.

5.1.7 Rozložení rychlosti proudění

Podmínky pro pohyb vody v otevřeném korytě nejsou ve všech bodech úseku koryta stejné. Největší ztráty na rychlostech vznikají v oblasti dna a břehů koryta a na hladině. Směrem od břehů, dna a hladiny se intenzita faktorů působících na rychlost proudění snižuje a podmínky pro pohyb částic se tedy zlepšují. Rychlost je více ovlivňována u dna koryta než u hladiny, proto se částice u nezamrzlých úseků pohybují rychleji blíže hladiny.

Na základě těchto publikovaných faktů můžeme říci, že proudnice s nejlepšími podmínkami a tedy nejvyšší rychlostí proudění se v rovných úsecích nachází ve střední části koryta, blíže hladiny. Dále předpokládáme, že rozložení rychlosti proudění v ploše se snižuje směrem od proudnice ke korytu s největší rychlostí.

5.2 Koryto

Většina řek prochází velkou řadou proudění (LEOPOLD, 1995), které je ovlivněno mimo jiné morfologií, odporem a nerovnostmi koryta. Úsek koryta nebo sledovaný profil může přenášet různé množství vody, která do něj přitéká z horní části toku. Pro hydraulické charakterizování určitého úseku toku se často využívá průměrná rychlost, průměrná hloubka a šířka vodního koryta.

Proudění je ovlivňováno nerovnostmi (drsností) břehů a dna koryta, což je dále umocňováno vegetací a dalšími faktory. Tyto faktory jsou ovšem velice obtížně změřitelné v reálném čase. V dnešní době moderních technologií je velice málo přírodních koryt, na kterých jsou všechny tyto faktory měřeny, a proto jsme odkázáni na pozorování a nepřímé výpočty. Drsnost prakticky nelze počítat přímo, ale musí být počítána z rychlosti, hloubky, sklonu atd.

5.2.1 Tvar koryta

Tvar profilu v kterémkoliv bodě říčního koryta je závislý na proudění, množství a charakteru sedimentů pohybujících se profilem, dále charakterem a složením materiálu, který tvoří dno koryta a jeho břehy. V přírodě je v neposlední řadě tvar koryta závislý na vegetaci. Proudění působí tlakem na dno a břehy, a pokud je tlak působící na každý bod koryta řeky vyvážen odporem dna a břehů, může koryto nabýt neměnného (stabilního) tvaru.

5.2.2 Profil koryta

Profil koryta u většiny řek není parabolický a není ani polokruhový. Profil má tendenci být v rovných úsecích lichoběžník. Jelikož se proudnice s největší rychlostí v obloucích neustále vychyluje od ideálního středu koryta, působí voda na břehy různou intenzitou. V nerovných úsecích proto bývá profil asymetrický. Dále po proudu se vzhled přibližuje trojúhelníku, v místech, kde se šířka zvětšuje rychleji než hloubka. Ve všech částech studovaného úseku koryta (viz. 6.1.1) můžeme říci, že se tvar příčných profilů blíží lichoběžníku.

5.3 Hydrometrie

Hydrometrie je věda, která se zabývá měřením proudění v otevřených korytech včetně metod, postupů a přístrojové techniky použité v hydrologii a managementu vodních zdrojů

(BOITEN, 2000). Hydrometrické charakteristiky sbírané v říčním úseku jsou důležité pro pochopení základního chování vodních proudů.

Měření potřebná pro zjištění morfometrických charakteristik se opírají o geodetická měření, která mohou být polohopisná a výškopisná (KŘÍŽ, 1988). Kapitola 5.3 se bude zabývat pouze základními principy pořizování dat, aby si čtenář udělal představu o jejich sběru. Práce využívá data z měření hloubek, vzdáleností (délka a šířka koryta) a rychlosti proudění.

5.3.1 Měření hloubek

Hloubka vody představuje svislou vzdálenost dna koryta od jeho hladiny. Měření hloubek se provádí většinou přizpůsobenou tyčí hydrometrické vrtule, měřičskou latí nebo jejího závěsu, popřípadě jinými metodami. Při větších hloubkách se používá ozvěnový hloubkoměr (echolot), který měří hloubku na základě času mezi vysláním a příjmem ultrazvukového signálu. Při měření hloubek bývají často sledovány i další veličiny např. rychlost proudění. Echolot se používá pouze pro měření hloubky, a proto se aplikuje především tam, kde je hlavním cílem zjištění hloubky, oproti tomu hydrometrická vrtule může odečítat hloubku a zároveň měřit rychlost proudění. Měření obou veličin zároveň je v některých případech výhodnější, jelikož eliminuje datovou nekonzistentnost, která je způsobena měřením v jiných časech.

5.3.2 Měření vzdáleností

Měření vzdáleností provádíme zpravidla u menších toků pásmem, u větších toků předem nataženým měřičským lankem. Při měření pásmem určujeme naměřenou vzdálenost na centimetry, při použití lanka, naměřenou vzdálenost odhadujeme na 10 cm. Kromě těchto pomůcek lze zejména na širokých tocích využít i geodetických metod, GPS nebo měření různými typy optických či laserových dálkoměrů, které však měří jen od jisté minimální vzdálenosti.

Měření vzdáleností umožňuje umístit prováděné měření rychlosti a hloubky v ploše. Naměřené souřadnice mohou být relativní nebo absolutní. Relativní souřadnice se vztahují pouze k určitému úseku toku nebo mohou být vzdálenosti počítány od pramene a břehů. Absolutní poloha může být měřena za pomoci GPS zařízení a měření jsou tedy umístěna v některém z geodetických systémů.

5.3.3 Měření rychlosti proudění

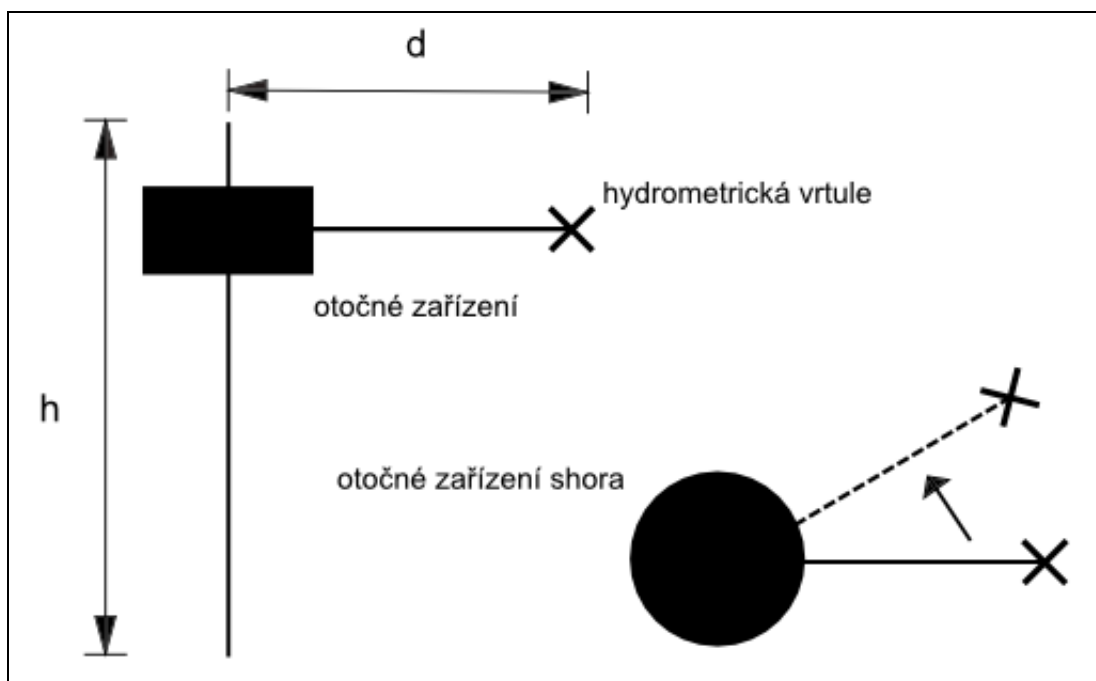
Pro hydrometrická měření rychlosti proudění se nejčastěji používají hydrometrické vrtule², které určují rychlost proudění v bodě, ve svislici či příčném profilu. Při měření se určuje počet otáček vrtule za zvolený časový interval nebo čas potřebný pro určitý počet otáček. Velikost hydrometrické vrtule se přizpůsobuje konkrétní situaci na úseku měření.

Pokud nelze zajistit vhodné podmínky měření, používají se jiné metody měření např. měření pomocí plováků nebo chemických stopovačů, které určují čas, za který částice vody urazí danou vzdálenost nebo mohou být rychlosti proudění počítány na základě hydraulických rychlostních vzorců. Existují také novější zařízení, která měří rychlost proudění na základě elektromagnetické indukce či ultrazvuku.

Umístění měření do určitého bodu v ploše se nejčastěji provádí na základě speciálního lanovkového zařízení (hydrometrická vrtule v závěsu), u kterého odečítáme uraženou vzdálenost od břehů (tzv. staničení). Bodová měření se mohou umísťovat pravidelně (viz. obr. 6), body měření jsou rozmístěny ve svislicích s určitými plošnými rozestupy, které rovnoměrně postihují celý měřený úsek. Ve svislicích mají body měření rovněž pravidelné rozmístění ve směru od hladiny ke dnu koryta.

Druhý přístupem k rozmístění bodů měření může být jejich subjektivní kladení tzn. body jsou umísťovány tak, aby postihly určitý sledovaný jev např. proudnici s největší rychlostí proudění. Takovým zařízením může být speciálně upravená hydrometrická vrtule s otočnou posuvnou tyčí, která měří rychlost proudění ve zvolených horizontálních vzdálenostech od středové tyči (svislá tyč). Po naměření rychlosti v určitém směru se pootočí o zvolený úhel a znovu měří rychlosti, takto provede měření pro celých 360°. Zobrazená měření mají tvar připomínající hvězdici (viz. obr. 5). Zařízení se může ve vertikálním směru posouvat po středové tyči, a proto jsou hvězdice měřeny i pro zvolené hloubky. Schéma zařízení pro měření rychlosti proudění v tvaru hvězdic je znázorněno na obr. 3.

2 Hydrometrická vrtule se skládá z pohyblivé části (vrtule), těla vrtule, signálního zařízení a kormidla (určení směru proudění). Součástí každé hydrometrické vrtule je soutyčí, které slouží k měření hloubek, nebo závěs s manipulačním zařízením.



Obr. 3: Schéma otočného zařízení pro měření rychlosti (d je vzdálenost od středové tyče, h je hloubka)

6 Data a modely

V následující kapitole budou popsána použitá data sledovaného říčního úseku, na která budou aplikovány dva statistické modely pro odhady hodnot z měřených dat.

6.1 Data

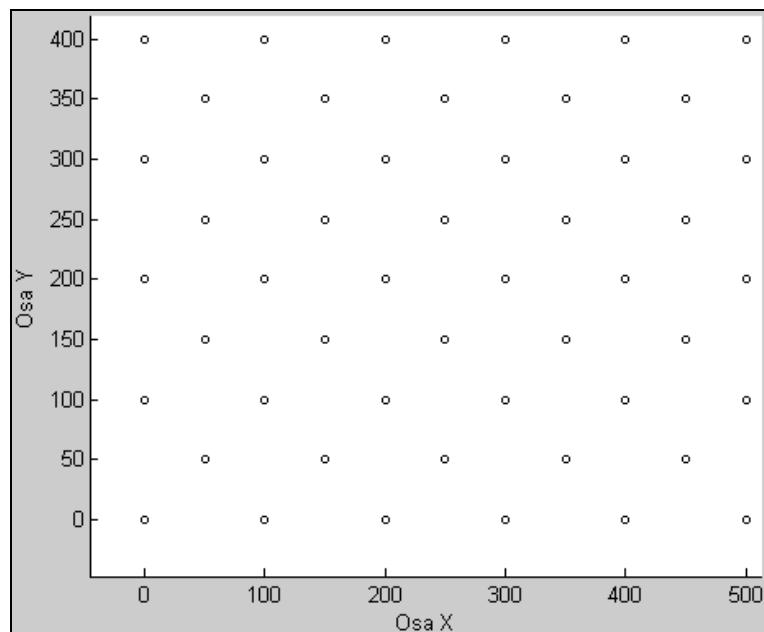
Naměřená data jsou ve formátu csv souborů a jsou dostupná na přiloženém CD. V souborech jsou hodnoty ve sloupcích odděleny pomocí tabelátoru. Záznamy vždy obsahují prostorovou složku, tzn. jsou pomocí pravoúhlých souřadnic x , y a z umístěny do sledované oblasti s říčním korytem. Hodnoty souřadnic x , y a z jsou uvedeny v milimetrech.

K dispozici máme data bodového měření koryta a bodového měření rychlosti, která byla naměřena v rámci projektu na univerzitě Boden Kultur (BOKU) v Rakousku. Data pochází z horního úseku Dunaje.

6.1.1 Body koryta

V souboru *labor.txt* je bodovým měřením definováno koryto toku, kdy každý záznam představuje souřadnici x , y a z . Osa x je kladena ve směru toku, osa y kolmo na směr toku a osa z ve směru naměřené hloubky. Body jsou v ploše x a y rozmístěny do tvaru obdélníku a postihují celý úsek koryta.

Zobrazíme-li body v ose x a y zjistíme, že mají určitou strukturu, naměřené body jsou v profilech střídavě tvořeny sudým a lichým počtem bodů. V „lichých“ profilech je o jeden bod méně a jsou posunuty na ose y . Vzdálenost mezi body se však nemění, můžeme tedy říci, že bodové měření bylo v ploše pravidelné (obr. 4).



Obr. 4: Pravidelné rozmístění bodů z měření koryta

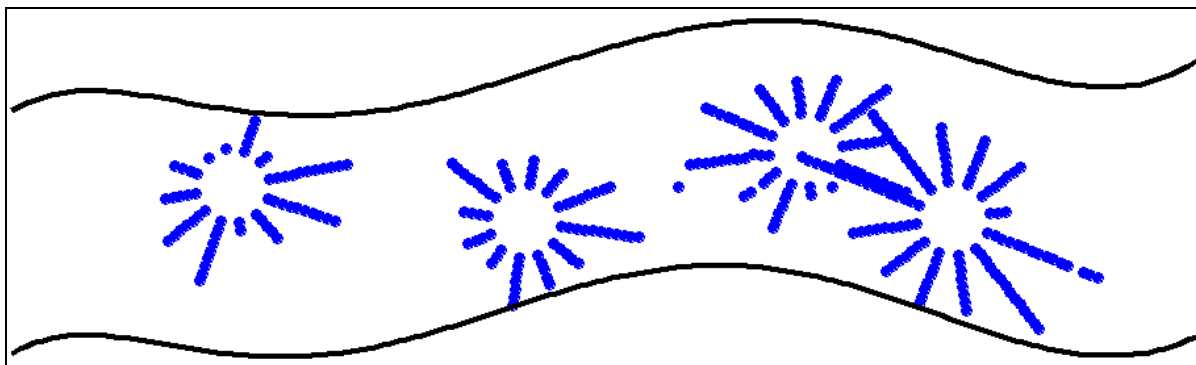
6.1.2 Bodové rychlosti

Soubory *starmesure.txt* a *crossmesure.txt* obsahují měření vektorových rychlostí proudění, každý záznam tedy představuje měření v bodě x, y a z.

Sloupce:

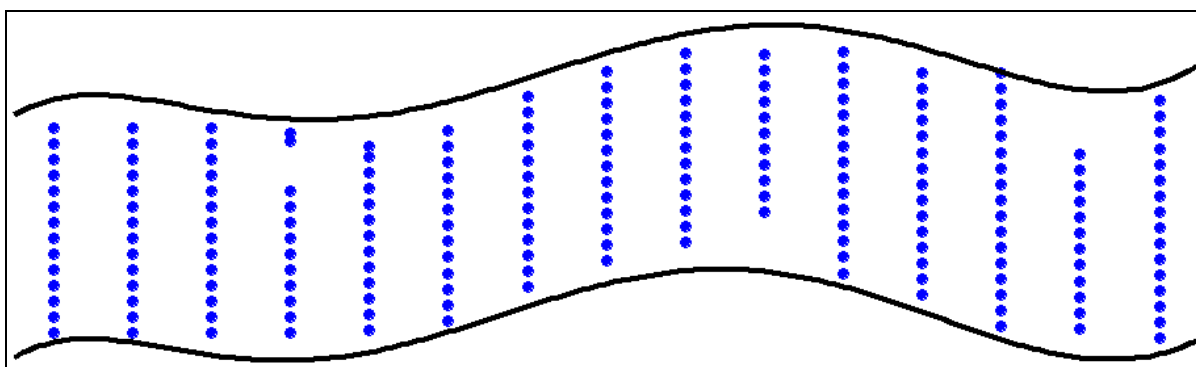
1. souřadnice x
2. souřadnice y
3. souřadnice z (hloubka)
4. vektorová rychlost proudění vody ve směru x
5. vektorová rychlost proudění vody ve směru y

Rozdíl mezi soubory *starmesure.txt* a *crossmesure.txt* je v rozmístění měřených bodů v korytě. První soubor představuje měření vektorových rychlostí, které jsou prostorově rozmístěny tak, že jejich tvar připomíná hvězdici. Předpokladem pro použití tohoto tvaru je, že proudnice s největší rychlostí bude procházet nebo se bude blížit středu hvězdice. Hvězdice byly umísťovány na základě tvaru koryta, z kterého se usuzovala přibližná poloha této proudnice. Obr. 5 ukazuje body měření shora („pohled racka“).



Obr. 5: Bodové měření rychlosti ve tvaru hvězdic (černě – břehová linie, modře – body měření)

Druhý soubor představuje body měření vektorových rychlostí, které jsou umístěné příčně (profilově) ke směru toku viz. obr. 6. Předpokladem je, že nalezneme alespoň jeden bod s největší naměřenou rychlostí proudění v každém profilu. Nalezené body budou tvořit základ pro výpočet proudnice s největší rychlostí.



Obr. 6: Bodové měření rychlosti profilové (černě – břehová linie, modře – body měření)

6.2 Kriging

6.2.1 Prostorová proměnná

Prostorová proměnná (DUTTER, 2000 a ARMSTRONG, 1998) je proměnná, která se nachází v nějakém prostoru, který může být n -rozměrný. Proměnná většinou představuje konkrétní jev, jako např. ložisko ropy či rozmístění počtu obyvatel. Příkladem v oblasti hydrologie může být pro 1D rozměr výška vodní hladiny v čase či v 3D prostoru rozložení

teploty vody v úseku řeky. Zjistit hodnoty proměnné (sledovaného jevu) pro všechny body x v definovaném prostoru je prakticky nereálné, a proto se používají modely pro odhad hodnot v místech, kde nebylo měření provedeno.

Máme-li tedy funkci náhodné proměnné $Z=Z(x)$, která může představovat náhodné rozmístění měření podle funkce Z v bodech x , můžeme v těchto bodech očekávat (odhadovat) určité hodnoty. Tento odhad E je vyjádřen funkcí $m(x)$, která se často označuje jako „drift“ a může představovat vážený průměr $Z(x)$. Drift je pro všechny náhodné proměnné v našem pojetí stejný a má následující podobu:

$$(4) \quad m = m(x) = E[Z(x)] .$$

Rozdíl mezi driftem a náhodnou proměnnou se nazývá rozptyl (variance), kterou budeme označovat jako var a její matematické vyjádření je následující:

$$(5) \quad var(Z(x)) = E[(Z(x) - m(x))^2] .$$

Kovariance C pak určuje rozdíl (vztah) proměnných na základě jejich vzdálenosti h . Jestliže máme dvě náhodné proměnné $Z(x_1)$ a $Z(x_2)$, kde $Z(x_1)$ je v prostoru určena hodnotou x , $Z(x_2)$ je pak určena hodnotou $x+h$, potom mají tyto náhodné veličiny svoji kovarianci, která je vyjádřena následující funkcí dvou lokací x a $x+h$:

$$(6) \quad C(h) = E[Z(x)Z(x+h)] - m^2 .$$

Vztah mezi kovariancí a rozptylem je pak následovný:

$$(7) \quad C(0) = var[Z(x)] .$$

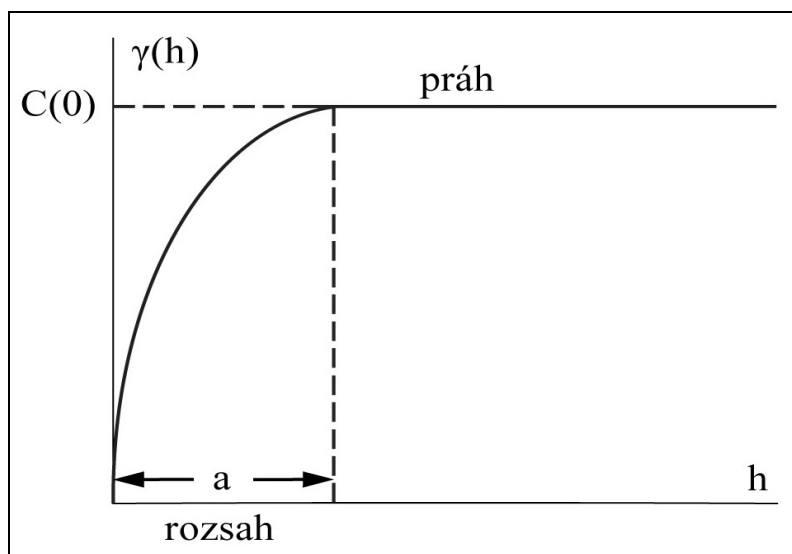
6.2.2 Variogram

Výpočet variogramu (semi-variogramu) je jedním z kroků pro tvorbu Krigingu. Variogram představuje rozptyl rozdílu náhodných proměnných $Z(x+h)$ a $Z(x)$, popisuje tak variabilitu sledovaných proměnných v závislosti na jejich prostorovém rozmístění (zejména na vzdálenosti) a odchylek od driftu. Variogram je definován následovně:

$$\gamma(h) = \frac{1}{2} var[Z(x+h) - Z(x)] .$$

Typický graf variogramu (obr. 7) se znázorňuje jako funkce vzdálenosti h a variogramu $\gamma(h)$, která určuje míru rozptylu hodnot v závislosti na vzdálenosti. Funkce může mít některé nastavení a vlastnosti, např. práh, který označuje limitní hodnotu vzdálenosti h , po

kterou jsou hodnoty variogramu brány v potaz. Práh je na ose variogramu roven nulové kovarianci (rozptylu).



Obr. 7: Typický průběh variogramu (upraveno z [23])

6.2.3 Modely variogramu

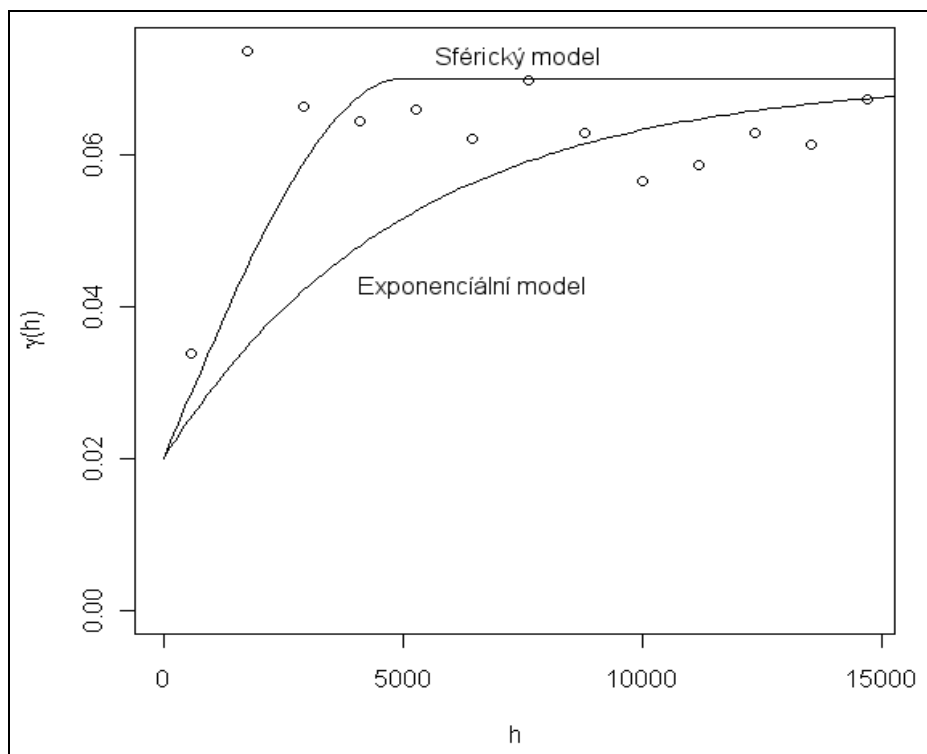
Velice důležitou částí při vytváření Krigingu je správné vystihnutí („nafitování“) variogramu modelem. Model zde představuje funkci nahrazující diskrétní hodnoty variogramu (viz. obr. 8), tato funkce je vybírána tak, aby nejlépe kopírovala obecný trend dat. Modelů³, které se snaží matematicky popsat průběh variogramu je celá řada:

- kubický model
- Gaussův model
- lineární model
- základní sinový model
- exponenciální model
- sférický model

Vzhledem k potřebám a rozsahu diplomové práce si na obr. 8 ukážeme pouze poslední dva modely, které byly aplikovány na data bodového měření koryta v profilovém tvaru (viz. 6.1.2). Sférický (spherical) model je jeden z nejpoužívanějších modelů, který vystihuje data za pomoci polynommické funkce, exponenciální model pak na základě exponenciální funkce.

³ Modely a jejich matematické vyjádření jsou popsány v [23].

Oba modely jsou lineární a jsou vhodné pro malé vzdálenosti. Rozdíly v modelech, které byly aplikovány na reálná data jsou patrné na první pohled. Sférický model narůstá velice strmě až do určitého prahu, který je přibližně $h=5000$ a od této hodnoty je konstantní. Exponenciální model je pozvolnější a narůstá po celou dobu.



Obr. 8: Exponenciální a sférický model

Při prokládání variogramu modelem se často setkáváme s tím, že model na základě automatického postupu vytvoří tzv. nugget effect, který je definován jako $\gamma(h)=0$. Jestliže tato situace nastane, bývá ošetřena definováním limitní hodnoty pro model. Z obr. 8 je patrné, že limitní hodnota variogramu byla nastavena na 0.02.

6.2.4 Teorie Krigingu

Základním úkolem Krigingu je odhadnout hodnotu sledovaného jevu v místě, které může být definováno bodem či blokem, toto místo bývá označováno písmenem V . Máme-li tedy n prostorových proměnných $Z(x_i)$, kde $i=1 \dots n$, můžeme definovat odhad hodnoty jevu Z'_v následujícím vztahem:

$$(8) \quad \boxed{Z'_v = \sum \lambda_i Z(x_i)} ,$$

kde λ_i představuje váhu pro každou prostorovou proměnnou $Z(x_i)$. Výpočet vah se většinou počítá na základě následujícího systému matic:

$$(9) \quad \begin{bmatrix} \gamma_{11} & \gamma_{12} & \dots & \gamma_{1n} & 1 \\ \gamma_{21} & \gamma_{22} & \dots & \gamma_{2n} & 1 \\ \dots & \dots & \dots & \dots & \dots \\ \gamma_{n1} & \gamma_{n2} & \dots & \gamma_{nn} & 1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_n \\ \mu \end{bmatrix} = \begin{bmatrix} \bar{\gamma}(x_1, V) \\ \bar{\gamma}(x_2, V) \\ \dots \\ \bar{\gamma}(x_n, V) \\ 1 \end{bmatrix},$$

kdy např. γ_{12} představuje hodnotu variogramu mezi $Z(x_1)$ a $Z(x_2)$, $\bar{\gamma}(x_1, V)$ pak představuje průměr variogramů v bodě x a v místě V . Hodnota Langrangeova multiplikátoru μ je blíže diskutována v [23]. Takto sestavený systém matic je součástí krigovací techniky s názvem Univerzální Kriging. Existují i další techniky, které však nebudou v této práci dále rozebírány.

Výsledkem je tedy odhad bodu, který se počítá na základě váženého průměru ze všech okolních prostorových proměnných.

6.3 Rychlost proudění v řece

Model rozložení rychlosti proudění v řece (TUČEK, 2006) se snaží odhadnout snižování rychlosti proudění ve směru od proudnice s největší rychlostí. Model je konstruován pro 3D úbytek rychlosti, avšak v práci bude algoritmicky zpracována jeho 2D modifikace. Rychlosti proudění jsou odhadovány na základě principů laminárního proudění.

6.3.1 Břehové linie (2D)

V 2D prostoru je břehová linie odhadována polynomem n -tého stupně, pokud předpokládáme, že n bude nabývat hodnot 3, 4 nebo 5. Tvar břehové linie je definován na základě vzorce:

$$(10) \quad \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ P(x) \end{pmatrix},$$

kde x představuje x-ovou souřadnici bodu koryta a y-ová souřadnice je pak vyjádřena obecnou definicí polynomu $P(x)$ (BARTSCH, 1979):

$$(11) \quad P(x) = a_n x^n + a_{(n-1)} x^{(n-1)} + \dots + a_1 x + a_0,$$

pro $n \in N$; $n = 3, 4, 5$; $a_k \in R$; $k = 0, 1 \dots n$; $a_n \neq 0$.

6.3.2 Proudnice

Tvar proudnice je definován (TUČEK, 2008 a TUČEK, 2009) ve vzorci (12), kde x představuje x-ovou souřadnici měřeného bodu rychlosti proudění, y-ová souřadnice je pak definována funkcí $\beta_1(x)$ a z-ová souřadnice funkcí $\beta_2(x)$. Funkce β_1 a β_2 jsou vyjádřeny pomocí polynomu 5-tého stupně, který byl stanoven na základě výzkumu. Parametry (koeficienty) $\gamma_i, i = 1, 2 \dots 6$ aproximují souřadnici x v y-ovém směru a parametry $\delta_i, i = 1, 2 \dots 6$ ve směru z-ovém.

$$(12) \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x \\ \beta_1(x) \\ \beta_2(x) \end{pmatrix} = \begin{pmatrix} x \\ \gamma_1 + \gamma_2 x + \gamma_3 x^2 + \gamma_4 x^3 + \gamma_5 x^4 + \gamma_6 x^5 \\ \delta_1 + \delta_2 x + \delta_3 x^2 + \delta_4 x^3 + \delta_5 x^4 + \delta_6 x^5 \end{pmatrix}$$

Při odhadu proudnice vycházíme z naměřených hodnot rychlosti proudění X^x, Y^x, Z^x a V^x , které představují vektory rychlosti proudění řeky v bodech x, y a z. Data měření jsou tedy reprezentována parametrem β , který popisuje souřadnice v bodě měření a rychlost v tomto bodě. Parametr β má následující podobu:

$$(13) \quad \beta = (X_1^x, Y_1^x, Z_1^x, V_1^x \dots X_I^x, Y_I^x, Z_I^x, V_I^x) ,$$

kde index I je roven počtu zkoumaných bodů.

Pro odhad projekce bodů na proudnici použijeme parametr Θ , který představuje koeficienty polynomu $\gamma_i, i = 1, 2 \dots 6$ a $\delta_i, i = 1, 2 \dots 6$ a má následující podobu:

$$(14) \quad \Theta = (\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6, \delta_1, \delta_2, \delta_3, \delta_4, \delta_5, \delta_6)$$

Projekce měřených bodů na odhadovanou proudnici je definována v parametru γ , který obsahuje vektory $X^\Delta, Y^\Delta, Z^\Delta$ s těmito body v souřadnicích x, y a z. Parametr dále obsahuje konstanty v_0, κ_1, κ_2 a κ_3 , které jsou dále popsány v kapitole 6.3.3. Parametr γ má pak následující podobu:

$$(15) \quad \gamma = (X_1^\Delta, Y_1^\Delta, Z_1^\Delta, V_1^\Delta \dots X_I^\Delta, Y_I^\Delta, Z_I^\Delta, V_I^\Delta, v_0, \kappa_1, \kappa_2, \kappa_3) ,$$

kde index I je opět roven počtu zkoumaných bodů.

Parametry Θ, β a γ pak musí splňovat následující podmínky:

$$(16) \quad V_i^x - v_0 + (X_i^x - X_i^\Delta)^2 \kappa_1 + (Y_i^x - Y_i^\Delta)^2 \kappa_2 + (Z_i^x - Z_i^\Delta)^2 \kappa_3 = 0 ,$$

$$(17) \quad \begin{aligned} &X_i^x - X_i^\Delta + (Y_i^x - Y_i^\Delta)[\gamma_2 + 2\gamma_3 X_i^\Delta + 3\gamma_4 (X_i^\Delta)^2 + 4\gamma_5 (X_i^\Delta)^3 + 5\gamma_6 (X_i^\Delta)^4] \\ &+ (Z_i^x - Z_i^\Delta)[\delta_2 + 2\delta_3 X_i^\Delta + 3\delta_4 (X_i^\Delta)^2 + 4\delta_5 (X_i^\Delta)^3 + 5\delta_6 (X_i^\Delta)^4] = 0 \end{aligned} ,$$

$$(18) \quad \boxed{\gamma_1 + \gamma_2 X_i^\Delta + \gamma_3 (X_i^\Delta)^2 + \gamma_4 (X_i^\Delta)^3 + \gamma_5 (X_i^\Delta)^4 + \gamma_6 (X_i^\Delta)^5 - Y_i^\Delta = 0} ,$$

$$(19) \quad \boxed{\delta_1 + \delta_2 X_i^\Delta + \delta_3 (X_i^\Delta)^2 + \delta_4 (X_i^\Delta)^3 + \delta_5 (X_i^\Delta)^4 + \delta_6 (X_i^\Delta)^5 - Z_i^\Delta = 0} ,$$

kde $i = I$.

6.3.3 Úbytek rychlosti proudění

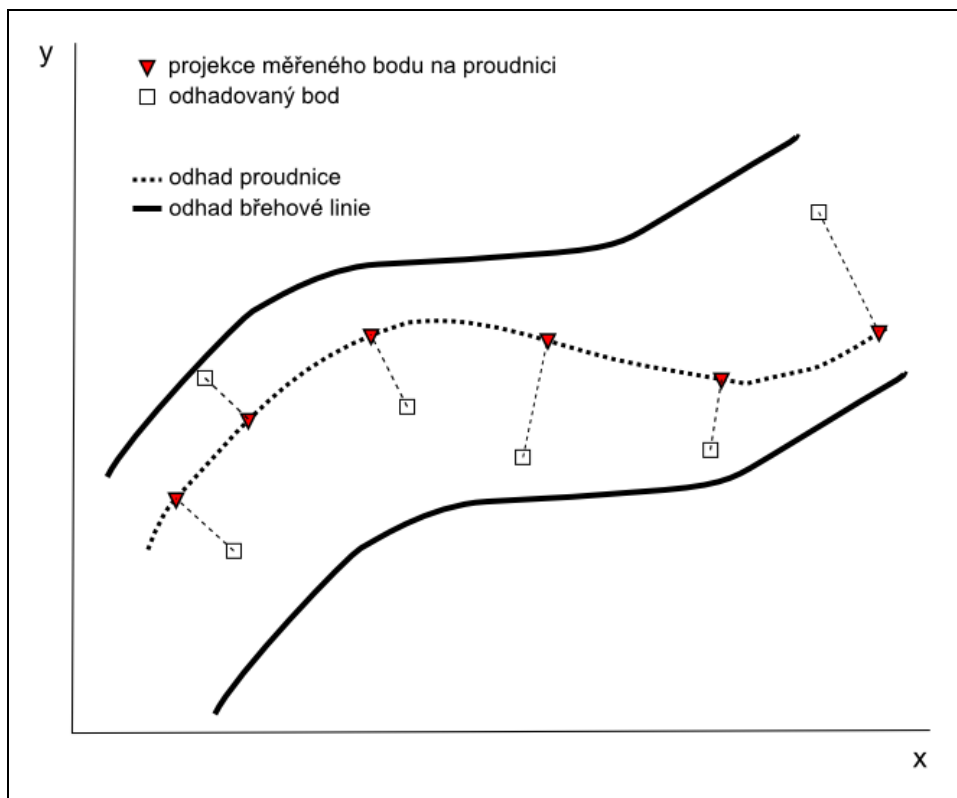
Úbytek rychlosti proudění vzhledem k rychlosti na odhadnuté proudnici je určen vztahem (20). Vektory X , Y a Z představují souřadnice x , y a z hledaného bodu v říčním korytě. Ke každému hledanému bodu je nalezena jeho projekce na proudnici o souřadnicích x , y a z , tyto projekce jsou definovány ve vektorech X^Δ , Y^Δ a Z^Δ .

Rychlost proudění na proudnici je definována parametrem v_0 , který je v této fázi modelu určen konstantně, avšak může být nahrazen vektorem V_0 , který by definoval rychlost proudění pro každý bod proudnice.

Parametry κ_i , $i = 1, 2, 3$ představují konstanty pro snižování rychlosti od proudnice a mohou obsahovat podmínky, které mají vliv na rozložení rychlosti proudění. V současném stavu modelu představuje konstanta κ vliv viskozity na zpomalování rychlosti od proudnice, avšak může být dále rozšiřována. Parametr κ_1 popisuje snižování rychlosti proudění od proudnice ve směru x , parametr κ_2 ve směru y a κ_3 ve směru osy z .

$$(20) \quad \boxed{V_i = v_0 - (X_i - X_i^\Delta)^2 \kappa_1 - (Y_i - Y_i^\Delta)^2 \kappa_2 - (Z_i - Z_i^\Delta)^2 \kappa_3}$$

Výsledkem výpočtu vztahu (20) je vektor rychlostí V_i , který představuje odhad rychlosti v bodech x , y a z , počítaný na základě vzdálenosti k projekci bodu na proudnici (viz. obr. 9), rychlosti proudění v této projekci a koeficientů snižování rychlosti κ .

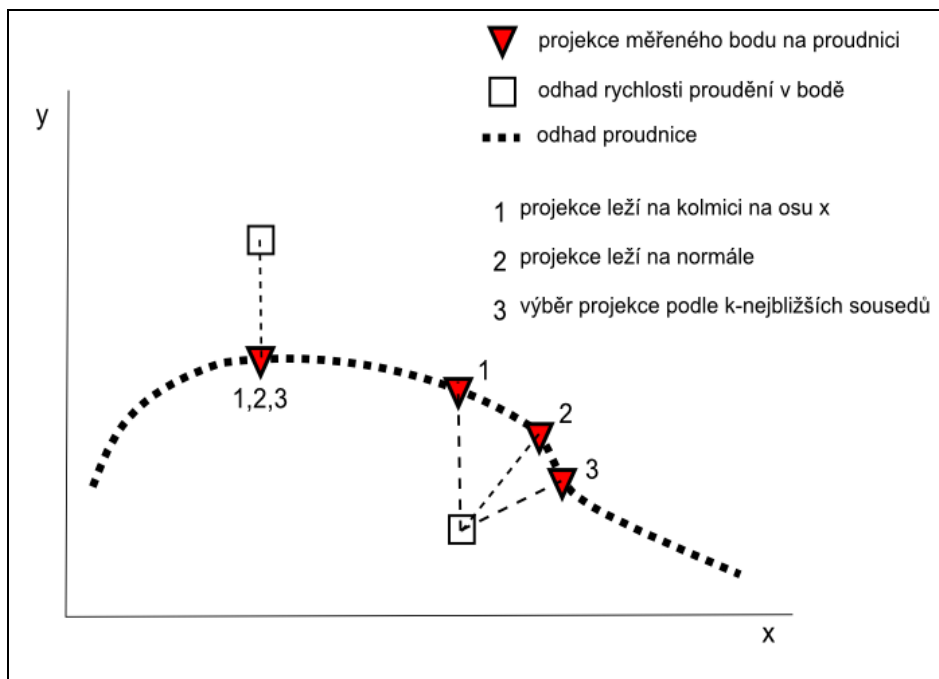


Obr. 9: Odhadované body a projekce měřených bodů

6.3.4 Hledání projekce k odhadovanému bodu

Situace na obr. 9 znázorňuje pouze jeden přístup hledání projekce k odhadovanému bodu, kdy odhadovaný bod a jeho projekce leží na normále k proudnici. Normála představuje kolmici vedenou od odhadovaného bodu k tečně na proudnici. Existují však i jiné přístupy k nalezení projekce na proudnici.

V diplomové práci jsme vytvořili algoritmy pro kolmou projekci na osu x a projekci, která využívá algoritmus k -nejbližších sousedů. V prvním případě jsme algoritmicky zpracovali situaci, kdy je spojnice mezi odhadovaným bodem a jeho projekcí na proudnici kolmá na osu x a v druhém případě jsme k odhadovanému bodu hledali projekci na základě algoritmu k -nejbližších sousedů. Algoritmus k -nejbližších sousedů hledá v našem případě právě jednoho nejbližšího souseda na základě euklidovské vzdálenosti.



Obr. 10: Výběr projekce podle různých přístupů

Obr. 10 zobrazuje situace, ve kterých může a nemusí hrát roli výběr metody pro hledání projekce k odhadovanému bodu. První situace (na obr. 10 vlevo) ukazuje, že v některých případech mohou všechny 3 metody nalézt stejnou projekci na proudnici. V této situaci tedy nemá volba metody hledání žádný vliv a bude teoreticky nastávat u tvarů proudnice, které jsou směrem k odhadovanému bodu vypouklé (konvexní) nebo v případech, kdy se bude proudnice blížit tvaru rovnoběžnému s osou x. Druhá situace (na obr. 10 vpravo) je opačná a dokazuje, že volbou metody hledání můžeme ovlivnit výběr projekce na proudnici.

Upravíme-li rovnici (20) pro potřeby kolmé projekce na osu x, zjistíme, že výsledný vektor bude závislý pouze na úbytku rychlosti ve směru os y a z:

$$(21) \quad V_i = v_0 - (Y_i - Y_i^\Delta)^2 \kappa_2 - (Z_i - Z_i^\Delta)^2 \kappa_3,$$

jelikož hodnoty vektorů X_i a X_i^Δ jsou pro všechna i totožné.

7 Softwarové prostředky

7.1 Matlab

MATLAB (MATrix LABoratory = maticová laboratoř) je výkonné interaktivní prostředí pro vědecké výpočty. Spojuje technické výpočty, vizualizaci dat a programovací jazyk v jednom prostředí. Společně s množstvím dostupných modulů vytváří ideální prostředek pro inženýry, vědce, matematiky a učitele při řešení problémů z mnoha oblastí. Aplikaci vyvíjí firma MathWorks a dodává ji s celou řadou rozšíření (toolboxů). Webové stránky firmy najdete na adrese <http://www.mathworks.com>. [24]

V této podkapitole si popíšeme vybrané stěžejní funkce a postupy, které byly využity při modelování a pro jejichž použití je potřeba mít k dispozici program Matlab. Předpokládá se, že základní funkce Matlabu, které budou také později zmíněny v našem modelu, má čtenář již osvojeny nebo použije některé z dostupných zdrojů pro jejich nastudování např. [24] nebo [11]. V naší práci jsme pracovali v Matlabu verze 7.0.0.0.19920 (R14) s těmito nainstalovanými toolboxy:

- Image Processing Toolbox
- MATLAB Compiler

7.1.1 Nearest neighbour (nejbližší soused)

Funkce *nearestneighbour* byla vyvinuta Richardem Brownem (University of Canterbury) v roce 2006 a její zdrojový kód je volně k dispozici. V naší práci využijeme syntaxi, při které funkce hledá nejbližší body ve vektoru P z vektoru X na základě euklidovské vzdálenosti. Výstupem jsou indexy nejbližších bodů vektoru P .

```
index = nearestneighbour(P,X);
```

Funkce může mít následující parametry:

- NumberOfNeighbours – hledá určitý počet nejbližších sousedů
- Radius – nastavení poloměru hledání
- DelaunayMode – pro hledání bodů je použita Delaunayho triangulace

```
% Syntaxe s parametrem NumberOfNeighbours a Radius  
% hleda K nejbližsich sousedu v okolí s polomerem M
```

```
index = nearestneighbour(P,X,'n',K,'r',M);
```

7.1.2 Poly2mask

Funkce *poly2mask* je součástí *Image Processing Toolboxu*, konvertuje vektorový polygon na matici s hodnotami 0 a 1. Hodnoty 1 představují pixely uvnitř polygonu (x, y) a hodnoty 0 pixely vně tohoto polygonu. Pokud je vstupní vektor (x, y) neuzavřený, spojí funkce automaticky počáteční a koncový bod.

```
maska = poly2mask(x,y,m,n);
```

Pokud prochází polygon středem pixelu, bývá pixel zařazen dovnitř nebo vně polygonu na základě speciálního algoritmu funkce *poly2mask*. Další informace o tomto algoritmu a o tom, jak vyhodnocuje sporné situace, které většinou vznikají v ostrých rozích polygonu, se můžete dočíst v [10] nebo v nápovědě o programu Matlab.

7.1.3 Práce s polynomy

Diplomová práce často operuje s polynomy (mnohočleny), a proto si nyní ukážeme jak s nimi můžeme pracovat v programu Matlab. Ke zjištění hodnot polynomu budeme využívat funkci *polyval* a ke zjištění koeficientů polynomu funkci *polyfit*. Polynomy se v Matlabu zadávají jako vektory, jejichž prvky tvoří koeficienty polynomu (KARBAN, 2006). Polynom řádu n je tedy vyjádřen vektorem o $n+1$ prvcích. Pokud některý z koeficientů v polynomu chybí, je prvek vektoru roven 0.

Chceme-li vypočítat následující polygon 4. řádu:

$$(22) \quad p(x) = 2x^4 + 4x^3 - x + 1,$$

použijeme vektor o 5 prvcích p a funkci *polyval*, která má první vstupní parametr vektor koeficientů p a druhým je vektor hodnot, pro které je polynom počítán.

```
>> p = [2 4 0 -1 1];  
>> x = [1 2 3];  
>> y = polyval(p,x)  
y = 6 63 268 % odhadnute hodnoty y
```

Funkce *polyfit* odhaduje koeficienty polynomu vstupních dat v bodech x, y a zadaného řádu polynomu n na základě metody nejmenších čtverců. Tato metoda se používá k proložení naměřených dat.

```
koeficient = polyfit(x,y,n);
```

7.1.4 Skripty a funkce

V Matlabu je možné psát příkazy přímo do příkazové řádky, což je vhodné pro drobné úlohy nebo pro úlohy, které jsou prováděny pouze jednou. Z hlediska automatického přístupu je vhodnější využít možnosti skriptů. Skript je posloupnost příkazů zapsaná v jednoduchém textovém souboru, který se v Matlabu označuje jako M-soubor (přípona souboru je .m). Po spuštění M-souboru jsou procedurálně zpracovány všechny příkazy v něm zapsané. Proměnné vytvořené za běhu programu jsou globální a lze je po spuštění použít v příkazovém řádku. Skripty nemají definované vstupní a výstupní parametry.

M-soubory se speciální strukturou se nazývají funkce a mají podobnou syntaxi jako funkce v běžných programovacích jazycích. Při psaní funkce definujeme její vstupní a výstupní parametry, které mohou být i prázdné. Výhodou funkcí je jejich zapouzdření, tzn. nemusíme znát programový kód, stačí její stručný popis a popis vstupních a výstupních parametrů.

Obsah souboru s funkcí pro výpočet mocniny může vypadat takto:

```
function [vystup] = mocnina(x)
vystup = x*x;
```

V příkazovém řádku Matlabu potom zavoláme⁴ funkci následovně:

```
>> y = mocnina(3);
```

7.1.5 Kompilace

Toolbox MATLAB Compiler obsahuje funkci *mcc*, která překládá M-soubory do C nebo C++ kódu (ROTH, 2005), jehož výstupem může být samostatný spustitelný soubor nebo knihovna. Tyto soubory jsou platformně závislé, a proto v našem případě budeme generovat .exe soubory pro systém Microsoft Windows. Matlab automaticky generuje tzv. wrapper, který tvoří rozhraní pro komunikaci mezi přeloženým M-souborem do formátu C nebo C++ a výsledným formátem souboru (.exe, .dll, .mex atd.). Celá kompilace probíhá v těchto krocích:

- překlad souboru do C nebo C++ zdrojového kódu
- vytvoření wrapperu
- kompilace a linkování zdrojového kódu

⁴ M-soubor musí být v adresáři, který Matlab prohledává (viz. kapitola *Current Directory Browser and Search Path* v nápovědě k programu Matlab).

Funkce *mcc* má mnoho parametrů, avšak v naší práci využijeme pouze kompilaci do .exe souborů s použitím wrapperu. Následující kód vytvoří z M-souboru *funkce.m* binární program *funkce.exe*. Volba *-W* definuje wrapper a volba *-T* definuje formát výstupního souboru. Lze použít pouze volbu *-m* (makro), která zjednodušuje syntaxi příkazu a je s první verzí kompatibilní.

```
%Cela syntaxe pro generovani .exe souborů
>> mcc -W main -T link:exe funkce.m
%Makro zjednodusujici syntaxi pro generovani .exe souboru
>> mcc -m funkce.m
```

Výhodou takto kompilovaných funkcí je přenositelnost kódu mezi uživateli. Koncový uživatel nemusí mít nainstalovaný Matlab, i přesto může funkci využívat. Binární verze funkce, stejně jako funkce v M-souboru, očekává vstupní parametry a předává zpět volajícímu programu⁵ výstupní parametry.

7.2 R

R je nástroj pro statistické a matematické výpočty a jejich vizualizaci. Uživateli nabízí grafické prostředí nad vlastním programovacím jazykem. Systém R byl vyvinut v Bell Laboratories Johnem Chambersem a kolektivem autorů. Program je šířen pod GNU licencí a je tedy volně šiřitelný i se zdrojovými kódy. Výhodou R je jeho rozšiřitelnost celou řadou volně dostupných modulů, proto má uživatel s internetovým připojením možnost kdykoliv stáhnout potřebný balíček tzv. „on the fly“.

V rámci R byl použit volně dostupný balíček *geoR*, který obsahuje statistické funkce potřebné pro výpočet variogramu a Krigingu. Syntaxe a postup zpracování dat je ukázán v kapitole 9.1.

7.3 C++

Jazyk C++ byl vyvinut na počátku 80. let Bjarnem Stroustrupem v Bell Labs. C++ je programovací jazyk, který vylepšuje procedurální jazyk C o podporu objektově orientovaného programování (OOP)⁶.

⁵ Volajícím programem může být např. příkazová řádka systému Windows nebo jiný program.

Objektový programovací jazyk byl vybrán proto, že se studovaný model (viz. 6.3) stále vyvíjí, a proto je žádoucí, abychom mohlo během vývoje schopni kdykoliv přidávat, upravovat či odebírat moduly⁷, bez nutnosti velkých úprav v jádru programu. Bylo také nutné zvolit vhodný programovací jazyk pro matematické a statistické výpočty.

7.3.1 Třídy

Podstatou objektově orientovaného programování je vytváření nových datových typů (tříd). Datové typy definují způsob uložení dat, tzv. datové elementy (PRATA, 2007) a jejich metody. Pokud je datový typ navržen můžeme z něj vytvořit jakékoliv množství objektů a ty mohou mít mezi sebou vztah. Často bývá jednodušší naprogramovat (navrhnout) novou třídu na míru než použít již vytvořené datové koncepty.

V našem případě třída definuje jedno měření v bodě a může se jednat o měření rychlosti proudění nebo měření koryta. Datovými elementy jsou pak hodnoty souřadnic popř. rychlosti proudění a metodami jsou nástroje pro zobrazení těchto hodnot, výpočet statistik a jejich zobrazení.

Třídy jsou navrženy tak, aby měly některé datové elementy společné, a v nich aby byly uloženy vypočtené statistiky pro všechny objekty stejné třídy. Tyto datové elementy tedy určují vztah mezi měřeními (objekty). Daným vztahem může být např. průměrná rychlost proudění, která se počítá z počtu všech vytvořených objektů třídy a sumy jejich rychlostí.

Využití tříd má při počítačovém modelování výhody, které vycházejí z obecně známých předností objektově orientovaného přístupu. Pokud budeme při vytváření programu používat tento přístup, může být vývoj programu postupný, jelikož přistupujeme ke každé třídě zvlášť. Změnou jednotlivých tříd nenarušíme chod celé aplikace, a proto můžeme jednotlivé třídy neustále vylepšovat pro potřeby modelu.

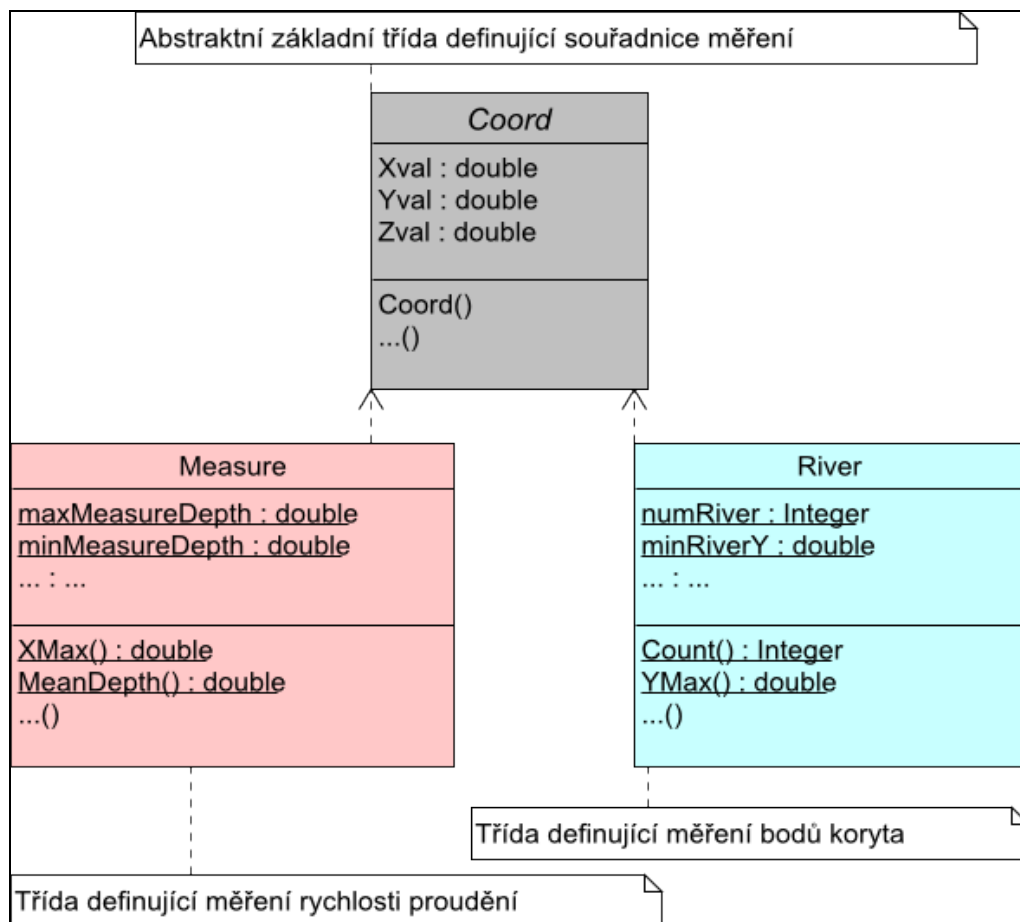
7.3.2 Abstraktní základní třída

Abstraktní základní třída je zvláštní a velmi pokročilý koncept OOP pro reprezentaci vzájemně podobných objektů či dat. Podstatou této filozofie je nalezení společných vlastností dvou tříd, které jsou poté definovány v základní třídě (abstraktní základní třída). Všechny

6 Existují dva základní přístupy k programování: procedurální (klade důraz na algoritmy) a objektové (klade důraz na data). Stavebními kameny OOP jsou třída (definice objektu) a objekt (instance třídy). Další informace o jednotlivých přístupech můžete získat např. v [3] nebo [4].

7 Modulem je zde myšlen objekt nebo funkce .

další třídy od ní odvozené (zděděné) mají zaručenou minimálně stejnou funkcionalitu, která je v základní třídě definována.



Obr. 11: Třídy definující načtená data

Máme-li data z měření rychlosti proudění a data z měření koryta, zjistíme, že obě datové sady vyjadřují měření v bodě x, y a z a jsou tedy v těchto vlastnostech podobné. Základem naší reprezentace dat bude společná abstraktní základní třída, která představuje prostorové určení bodových měření. Od ní pak budou odvozeny třídy s měřením rychlosti proudění (třída Measure) a měřením bodů koryta (třída River), tyto třídy rozvíjí souřadnice základní třídy o další vlastnosti např. třída Measure bude obsahovat vlastnosti s největší, průměrnou a nejmenší rychlostí proudění. Celou předchozí situaci naznačuje UML⁸ diagram tříd na

8 UML (Unified Modeling Language) je jazyk pro návrh aplikace.

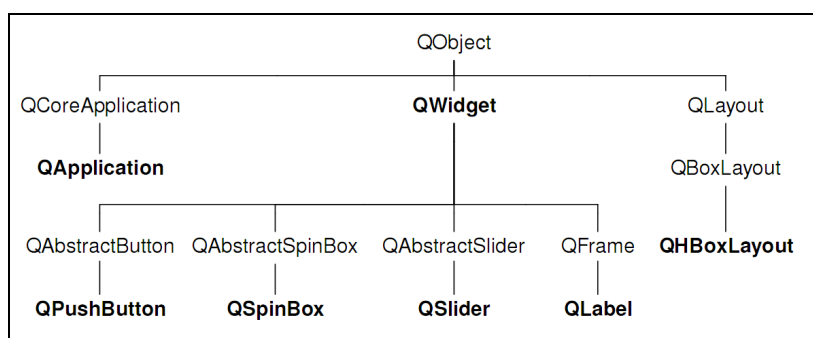
obrázku 11. Tyto třídy jsou obsaženy v hlavičkovém a implementačním souboru⁹ s názvem `RiverData`, které jsou uloženy v příloze 1.

7.4 Knihovna Qt

Qt¹⁰ je C++ framework pro vývoj multiplatformních GUI aplikací (BLANCHETTE, 2006), umožňující napsat jeden programový kód, který je možné kompilovat pro více operačních systémů. Programový kód může být kompilován pro systém Windows, Linux, Mac OS X, Solaris, HP-UX a další verze Unixových systémů. Použití knihovny Qt již předpokládá znalosti C++ objektového programování.

V kapitole 7.4 je popsána hierarchie objektů knihovny Qt, třídy a přístupy, které jsou autorem diplomové práce považovány za stěžejní při řešení této práce. Další informace může čtenář získat z těchto zdrojů: [20], [18] a v nápovědě programu Qt. Zvláště pak kapitoly Tutorials a Qt Examples jsou velice cenným zdrojem inspirace při tvorbě vlastní aplikace.

7.4.1 Hierarchie objektů knihovny Qt



Obr. 12: Dědičnost objektů Qt (stromová struktura). [20]

Knihovna Qt je objektovou knihovnou to znamená, že je tedy každý element¹¹ programu objektem s určitými vlastnostmi a metodami. Jednotlivé objekty jsou ve stromové struktuře hierarchicky provázány, tzn. existuje kořenový objekt či rodič a další objekty jsou jeho

9 Hlavičkové a implementační soubory jsou textové soubory s koncovkami `h` a `cpp`. Hlavičkový soubor obsahuje definici třídy. Implementační soubor obsahuje implementaci metod definovaných v hlavičkové třídě. Více informací například zde [3].

10 Zkratka Qt vznikla spojením písmen *Q*, které nemělo žádný význam, pouze vypadalo pěkně v Emacs editoru a *t* od slova toolkit, což znamená soubor nástrojů. Všechny třídy knihovny Qt mají jako prefix písmeno *Q*.

11 Element v tomto kontextu znamená např. tlačítko, nástrojová lišta, toolbar, dialogové okno atd.

potomky (dědí jeho vlastnosti). Potomci mohou používat veřejné metody a vlastnosti svých rodičů. Obr. 12 graficky znázorňuje stromovou strukturu dědičností, kdy kořenovým objektem je *QObject* a jeho potomky jsou např. *QWidget*, *QLayout*, ale také *QPushButton*.

Objekty knihovny Qt jsou sdružovány do modulů (Qt's Modules), které jsou dále děleny na komerční, nekomerční, unixové atd. V práci jsme využili pouze nekomerční moduly QtCore a QtGUI. QtCore je základní modul obsahující všechny potřebné negrafické třídy, QtGUI je naopak modul, který sdružuje všechny grafické třídy.

7.4.2 QVector

QVector je třída podobná třídě *vector* ze standardní knihovny C++, obsahuje metody pro práci s polem objektů. *QVector* je šablona třídy, která poskytuje uživateli všechny výhody dynamických polí, umožňuje indexovat uložené prvky a přistupovat k nim pomocí těchto indexů.

7.4.3 Modularita

Vyvíjená aplikace je modulární, což znamená, že máme hlavní program, který může být rozšířen o další moduly bez zásahu do jeho kódu. Rozšíření (tzv. pluginy) jsou do aplikace přidávány pomocí dynamických nebo statických modulů. Statické moduly musí být vždy dostupné z hlavní aplikace, jelikož jsou do programu přidány při jeho startu. Dynamické moduly představují knihovny, které mohou být z aplikace zavolány za běhu. Postup vytvoření dynamických a statických modulu v aplikaci VOR se inspiroval příklady ke knihovně Qt (Qt examples – Plug & Paint).

7.4.4 QProcess

QProcess je třída, která umožňuje spustit externí aplikace z hlavního programu a řídit s ní komunikaci. *QProcess* spustí tuto aplikaci na základě cesty ke spouštěcímu souboru, který má v systému Windows formát exe. Externí aplikaci mohou být předány argumenty, které v našem případě představují např. cesty k datovým souborům, hodnoty zadané uživatelem, hodnoty vypočtené v hlavním programu atd. Hlavní program udržuje spojení a čeká na výstupy z externí aplikace. Tato komunikace probíhá ve dvou „kanálech“, první kanál zachytává normální hlášení a druhý chybové hlášení. *QProcess* má spousty dalších metod, které jsou dále popsány v nápovědě ke knihovně Qt nebo v [18].

8 VOR

Aplikace VOR (Velocity Of the River) slouží k výpočetnímu zpracování modelu popsaného v kapitole 6.3. VOR je napsán v programovacím jazyce C++, který využívá nekomerční knihovnu Qt zvláště pro grafický interface, ale také pro jiné negrafické úlohy. Její součástí jsou moduly zkompilovaných Matlab skriptů, pro které byla vytvořena uživatelská dialogová okna. Matlab skripty jsou volány pomocí Qt objektu Qprocess.

VOR v současné době striktně očekává data v textovém formátu, které obsahují měřené hodnoty. Očekávaná struktura datových sad je popsána v kapitole 6.1.

8.1 Instalace

Aplikaci VOR není nutné přímo instalovat, jelikož obsahuje všechny potřebné knihovny a neukládá žádná data přímo do registru systému. VOR ke svému chodu využívá M-soubory Matlabu, které jsou zkompilované do formátu .exe, a proto je celá aplikace platformně závislá na systému Microsoft Windows, což neplatí o programovém kódu. Aplikace byla testována na systému Windows XP, instalace probíhala v následujících dvou krocích:

- instalace MCR
- kopírování souborů aplikace VOR

MCR (MATLAB Component Runtime) obsahuje soubor knihoven pro běh zkompilovaných M-souborů na počítači, který nemá nainstalovaný program Matlab. MCR podporuje prakticky všechny funkce jazyka Matlab, kromě jistých výjimek např. MATLAB C/C++ Math nebo některých grafických knihoven. Instalátor MCR (soubor *MCRInstaller.msi*) je součástí Matlab Compiler toolboxu a standardně se nachází v podsložce tohoto toolboxu¹², instalátor bude také k dispozici na přiloženém CD.

Na počítači, který má nainstalovaný MCR je možné zkopírovat aplikaci VOR kamkoliv, avšak je nutné dodržet stejnou adresářovou strukturu aplikace jako na přiloženém CD, protože aplikace používá relativní cesty. Aplikaci můžeme spustit přímo z CD, avšak pro „svižnější“ chod se doporučuje VOR zkopírovat na pevný disk.

¹² Relativní cesta k MCR instalátoru začínající od kořenového adresáře Matlabu:

`\toolbox\compiler\deploy\win32\MCRInstaller.msi`

Při chodu aplikace mohou nástroje Matlabu (viz. 8.4) hlásit chybové hlášení, které je potřeba vyřešit na konkrétním počítači. Tyto hlášení jsou způsobeny chybami v Matlab Compiler toolboxu, který se stále vyvíjí. V diplomové práci jsme neměli k dispozici nejnovější verzi, a proto se vyskytly chyby, které jsou u novějších verzí opraveny. V naší verzi Matlab Compiler 4.0 se objevily tyto chyby:

- mkl_def.dll
- matlabrc at 81

Chybové hlášení *mkl_def.dll* je způsobeno absencí knihovny pro konkrétní procesor, který používá cílový počítač. Vyřešit tento problém je možné editací souboru *blas.spec*¹³ a doplněním typu procesoru a cesty k příslušné dynamické knihovně. Další informace o problému můžete nalézt zde [10].

Chyba *matlabrc at 81* je způsobena tím, že skript Matlabu běží v jiném *Místním nastavení* Windows XP než *Angličtina(Spojené státy)*¹⁴, tento problém je možné vyřešit změnou nastavení na tuto hodnotu.¹⁵

Poslední zjištěnou chybou, která je spíše upozorněním a na chod aplikace VOR nemá velký vliv je *pathdef.m*. Chybové hlášení můžeme opravit vložením prázdného souboru *pathdef.m* do složky s lokálním nastavením MCR.¹⁶

8.2 Koncept

Aplikace VOR je řešena objektově, a proto jsou jednotlivé části programu zapouzdřeny. Zapouzdření zjednodušeně znamená sdružení kódu do jednotlivých tříd, jejichž vývoj probíhá samostatně. Hlavičkové, implementační soubory¹⁷ a samotné třídy programu VOR jsou znázorněny na UML diagramu (obr. 14) pomocí bílých obdélníků, které jsou dále sjednoceny do logických bloků. Tyto bloky jsou zobrazeny na obrázku 14 pomocí barevných

13 Relativní cesta k souboru *blas.spec* začínajícím od kořenového adresáře MCR:

\MATLAB Component Runtime\v70\bin\win32\blas.spec

14 Ovládací panely – Místní a jazykové nastavení – Standardy a formáty

15 Více informací na této stránce:

<http://www.mathworks.com/support/solutions/data/1-22FK61.html?product=CO&solution=1-22FK61>

16 Relativní cesta ke složce s lokálním nastavením začínajícím od kořenového adresáře MCR:

\MATLAB Component Runtime\v70\toolbox\local\

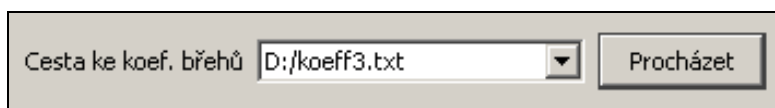
17 Hlavičkové a implementační soubory jsou textové soubory, ve kterých jsou uloženy definice tříd a jejich metod. Více o těchto souborech se můžete dočíst např. v [3].

složek. Červené obdélníky pak představují zkompilované Matlab skripty, které jsou pojmenovány podle názvu binárních souborů.

Šipky na obrázku 14 představují závislosti jednotlivých složek. Přerušovaná šipka je dědičnost a šipka s plnou čarou znázorňuje předání argumentů Matlab skriptům a jejich spuštění. Třídy ve složkách mají závislosti i mezi sebou, avšak ty nebyly pro přehlednost diagramu zobrazeny. V následujícím textu kapitoly si jednotlivé bloky (složky) dále popíšeme.

Složka VOR obsahuje třídu PluginDialog, která je použita pro informativní zobrazení načtených knihoven. RiverData je hlavičkový a implementační soubor s třídami pro uložení dat (viz. 7.3.2). Třída MainWindow definuje základní uživatelské prostředí, vytváří vzhled a funkčnost jednotlivých prvků (tlačítka, menu, tabulka atd.) hlavního uživatelského okna. MainWindow je funkčně nejobsáhlejší, jelikož se stará o základní chod aplikace.

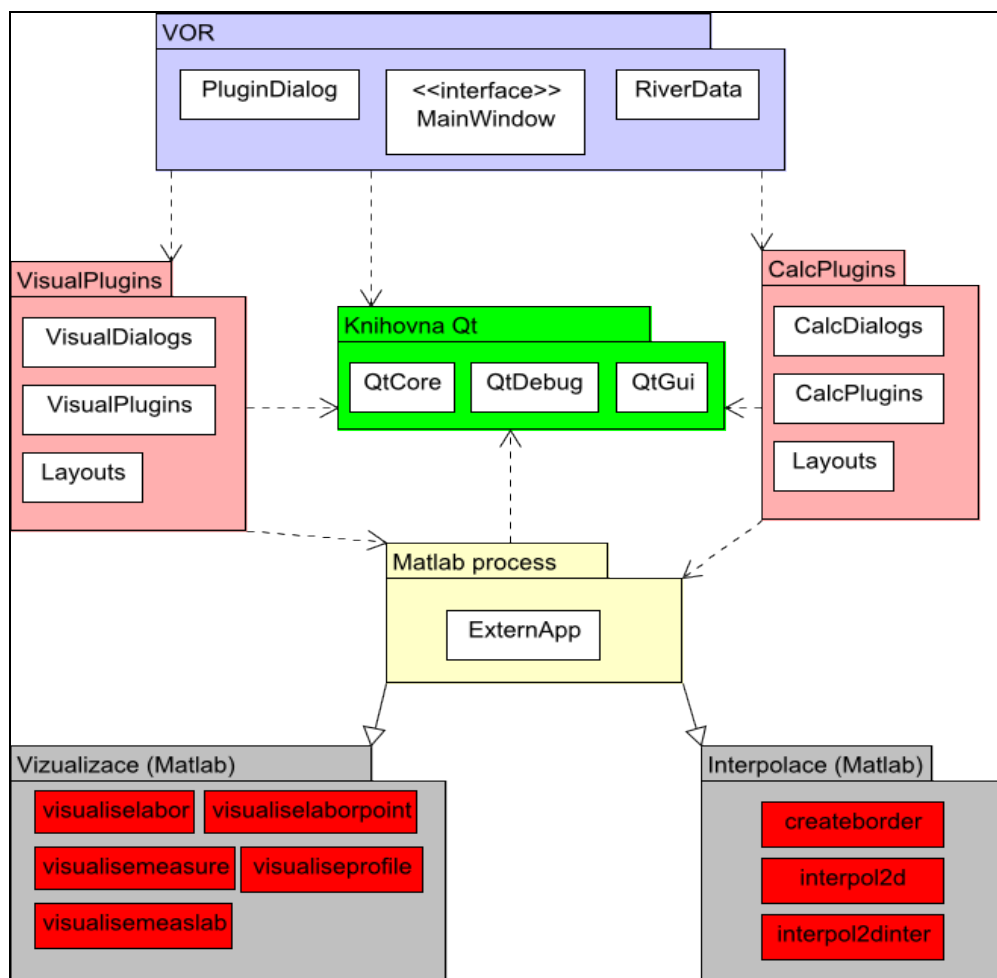
Složka VisualPlugins sdružuje třídy VisualDialogs, VisualPlugins a Layouts, které společně tvoří tzv. statický modul. Třída VisualPlugins definuje název menu tlačítka pro Nástroje (viz. kapitola 8.4) a vazbu tlačítka na příslušný dialog. Vzhled a funkce jednotlivých dialogů jsou definovány v třídě VisualDialogs. Třída Layouts obsahuje vazbu mezi jednotlivými grafickými ovládacími prvky (viz. obr. 13), např. pro načtení souboru se používá text (QLabel), výběrový seznam (QComboBox) a tlačítko Procházet (QPushButton). Všechny tyto prvky jsou následně na dialog volány přes jeden objekt.



Obr. 13: QLabel, QComboBox a QPushButton

Složka CalcPlugins sdružuje třídy CalcDialogs, CalcPlugins a Layouts, které společně tvoří, na rozdíl od složky VisualPlugins, tzv. dynamický modul. Třídy mají stejný význam jako je tomu u tříd složky VisualPlugins.

Složka Matlab process obsahuje třídu ExternApp, která definuje postupy a nastavení pro komunikaci a spuštění externí aplikace. Jejím úkolem je na základě cesty k externí aplikaci a parametrů spustit tuto aplikaci a všechny výstupy z ní přesměrovat do hlavního programu.



Obr. 14: UML diagram tříd programu VOR

8.3 Funkcionalita

Aplikace VOR má standardní vzhled grafických aplikací, obsahuje menu, nástrojové lišty, hlavní a vedlejší panel. Menu má 4 hlavní položky:

- *Soubor*
- *Zobrazit*
- *Nástroje*
- *Nápověda*

Soubor obsahuje tlačítka pro otevření dat koryta a dat měření. Po zadání cesty k příslušnému datovému souboru uloží VOR data automaticky do datových objektů (viz. 7.3.2), které jsou dále uloženy do objektu QVector (viz. 7.4.2). QVector zaručí, že může aplikace VOR k těmto datovým objektům přistupovat přes indexy. K funkcím pro otevření

dat můžeme také přistupovat přes klávesové zkratky „Ctrl – K“ a „Ctrl – M“ nebo přes ikony v toolbaru.

Soubor dále obsahuje tlačítka pro uložení načtených dat. Načtená data z měření rychlosti mohou být uložena pro všechny hloubky nebo pouze pro určitou hloubku. Po zmáčknutí tlačítka *Uložit měření jako..* se otevře dialog s výběrem hloubek. Zvolením příslušné hloubky uložíme data pouze v této hloubce.

Obdobný postup může být aplikován na soubor s korytem, s tím rozdílem, že výběr hloubek se aktivuje v případě, že jsou v programu načtena i data z měření rychlostí. Jestliže uživatel uloží koryto pro určitou hloubku, dojde k zjištění rozdílu průměrných hodnot v prvním a posledním profilu z dat koryta. Přičtením a odečtením tohoto rozdílu k vybrané hloubce vznikne interval pro výběr hloubek. Potom je u všech objektů koryta zkontrolováno, zda jejich hodnota hloubky odpovídá intervalu, a pokud ano, jsou uloženy do výsledného souboru. Tímto postupem vytvoříme datovou sadu, která se snaží ošetřit rozdíl hladin v horní a dolní části koryta, který je způsoben sklonem koryta.

Posledním možností menu *Soubor* je tlačítko pro vypnutí aplikace.

V menu *Zobrazit* jsou tlačítka pro zobrazení a skrytí nástrojových lišt a statistik. Statistika jsou automaticky vytvářeny při načtení datové sady a jsou zobrazeny v plovoucích oknech s tabulkami (viz. obr. 15). V těchto tabulkách jsou některé základní statistiky pro datové sady. Dalším důležitým prvkem, který může být zobrazen, je panel *Výstupy*, který vypisuje standardní i chybové hlášení externí aplikace, a také další informace pro uživatele aplikace.

Menu *Nápověda* v současné době neobsahuje nápovědu k programu, pouze zobrazuje načtené moduly (tlačítko *Načtené moduly*). Funkce načtené moduly je s drobnými úpravami převzatý kód z příkladů ke knihovně Qt (Qt Examples).

	Stat. proměnná	Hodnota
1	Obecné	
2	Počet záznamů	1001
3	Souřadnice X	
4	Max	13742.2
5	Min	0
6	Souřadnice Y	
7	Max	3596.73
8	Min	0
9	Hloubka	
10	Průměr	-209.241
11	Max	-50
12	Min	-750
13	Rychlost proudění	
14	Průměr	0.407138
15	Max	2.59307
16	Min	0.0141421

Obr. 15: Panel se základní statistikou datové sady

8.4 Nástroje

Nástroje (menu *Nástroje*) programu VOR jsou tvořeny dialogy, které jsou grafickým rozhraním pro zasílání argumentů externí aplikaci. Dialogy obsahují tlačítko *Start*, které slouží pro spuštění externí aplikace pomocí Qt objektu *QProcess* (viz. 7.4.4). Pokud nejsou vyplněny požadované údaje¹⁸, není externí aplikace spuštěna. Po správném vyplnění a stisknutí tlačítka *Start* se zobrazí tzv. progress dialog, což znamená, že výpočet právě probíhá. Zmáčknutím tlačítka *Cancel* na progress dialogu je možné výpočet přerušit.

8.4.1 Vizualizace

Aplikace VOR v současné době obsahuje 5 vizualizačních nástrojů, které můžeme nalézt v menu *Nástroje – Vizualizace*. Následující výčet obsahuje názvy těchto nástrojů v tomto menu a v závorce argumenty, které očekávají:

- *3D koryto* (načtená data z měření koryta)
- *Body koryta* (načtená data z měření koryta)
- *Body měření* (načtená data z měření rychlosti)
- *3D koryto a body měření* (načtená data z měření koryta, načtená data z měření rychlosti a zadaná hloubka měření rychlosti)
- *Příčný profil* (načtená data z měření koryta, vzdálenost na ose x a zvolený způsob zobrazení)

Poznámky: Nástroje *3D koryto* a *3D koryto a body měření* očekávají pravidelnou síť bodů koryta, a proto není možné vykreslit data koryta, která byla upravena pro určitou hloubku, např. nástrojem *Uložit koryto jako* (viz. kapitola 8.3). Nástroj *Příčný profil* vykreslí profil ve vzdálenosti na ose x pouze v případě, že pro zadanou vzdálenost existují naměřené body.

8.4.2 Interpolace

Aplikace VOR dále obsahuje 3 interpolační nástroje, které můžeme nalézt v menu *Nástroje – Interpolace*. Následující výčet obsahuje názvy těchto nástrojů v tomto menu:

- *Vytvořit koeficienty břehů*
- *2D Interpolace*

¹⁸ Pozn.: Musí být vždy načtena buď data měření nebo data koryta, anebo oba soubory v hlavním programu.

- *2D Interpolace (interaktivní)*

Nástroj *Vytvořit koeficienty břehů* vytváří 2D břehovou linii. Uživatel interaktivně zadá odpovídající počet bodů, které jsou dále interpolovány na základě polynomu n -tého stupně. Koeficienty těchto polynomů jsou uloženy do textového souboru, který je možné opětovně použít při dalších analýzách. Tento nástroj očekává následující argumenty:

- *Soubor – Cesta ke korytu* (načtená data z měření koryta)
- *Nastavení – Počet bodů pravého břehu* (celé a kladné číslo)
- *Nastavení – Počet bodů levého břehu* (celé a kladné číslo)
- *Nastavení – Stupeň polynomu* (na výběr polynom 5, 4 a 3-tího stupně)
- *Výstup – Koeficienty břehů* (cesta k souboru s uložením koeficientů výsledných polynomů)

Nástroj *2D Interpolace* slouží k vytvoření gridu rozložení rychlosti proudění podle modelu 6.3. Tento nástroj na základě zadaných koeficientů úbytku κ (viz. 6.3) a zadané rychlosti proudění na proudnici stanoví rychlost ve všech bodech 2D říčního koryta. Říční koryto je ohraničeno břehovou linií a obsahuje interpolovanou proudnici. *2D Interpolace* očekává tyto argumenty:

- *Soubor – Cesta ke korytu* (načtená data z měření koryta)
- *Soubor – Cesta k měření* (načtená data z měření rychlosti)
- *Soubor – Cesta ke koef. břehů* (cesta k souboru s uloženými koeficienty břehové linie)
- *Projekce* (na výběr projekce kolmá a nearest neighbour – nejbližší soused)
- *Nastavení proudnice – Koeficienty* (reálné číslo; očekává koeficienty odděleny čárkou, podle počtu koeficientů je zvolen stupeň polynomu)
- *Nastavení proudnice – Rychlost na proudnici* (reálné číslo)
- *Nastavení proudnice – Kappa 1* (reálné číslo)
- *Nastavení proudnice – Kappa 2* (reálné číslo)
- *Nastavení – Hloubka* (výběr hloubky podle načteného měření rychlostí)

Nástroj *2D Interpolace (interaktivně)* slouží také k vytvoření gridu rozložení rychlosti proudění podle modelu 6.3. Proudnice je na rozdíl od výše zmíněného nástroje zadávána interaktivně uživatelem. Nejprve jsou zobrazeny body měření, z kterých uživatel vybere

body proudnice, body proudnice jsou potom automaticky interpolovány pomocí polynomu. Nástroj má tyto doplňující nastavení:

- *Nastavení – Počet bodů proudnice* (celé a kladné číslo)
- *Nastavení – Stupeň polynomu* (na výběr polynom 5, 4 a 3-tího stupně)
- *Nastavení – Nepřichytávat* (algoritmus použije pro výpočet proudnice přesně body, které zadal uživatel)
- *Nastavení – Přichytávat vše* (uživatel zadá body a algoritmus k nim vybere nejbližší body z měření rychlosti)
- *Nastavení – Přichytávat první a poslední* (první a poslední bod je použit podle nastavení *Nepřichytávat* a ostatní podle nastavení *Přichytávat vše*)

8.5 Jiné přístupy

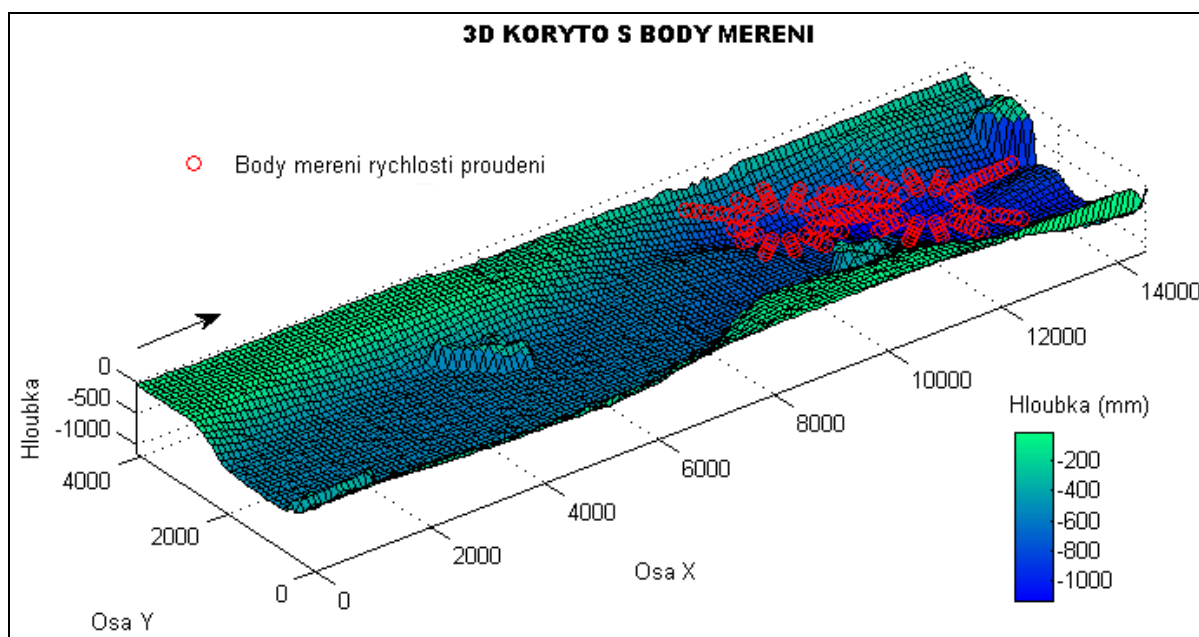
Pro modelování proudění existuje celá řada jiných komerčních i nekomerčních softwarů. V této oblasti převažují spíše komerční softwary, vzhledem k tomu, že vývoj takovýchto nástrojů je značně časově i finančně náročný. Z těchto softwarů uvedeme pouze výčet, jelikož často nejsou dostupné pro obyčejného uživatele. Z volně dostupných softwarů můžeme uvést např. FESWMS, který je určen pro 2D modelování. Jeho zdrojové kódy, které jsou z velké části napsané ve Fortranu jsou dostupné z níže uvedeného internetového zdroje.

- 1D modely
 - HEC-RAS (<http://www.hec.usace.army.mil/>)
 - Hydrocheck (<http://www.hydrosoft.eu/>)
 - SOBEK (<http://delftsoftware.wldelft.nl/>)
- 2D modely
 - FESWMS (<http://water.usgs.gov/software/FESWMS-2DH/>)
 - MIKE 21C (<http://www.dhigroup.com/>)
 - Telemac 2D (<http://www.telemacsystem.com/>)
- 3D modely
 - FLOW 3D (<http://www.flow3d.com/>)
 - Delft 3D (<http://delftsoftware.wldelft.nl/>)
 - FLUENT (<http://www.fluent.com/>)

9 Aplikace modelů

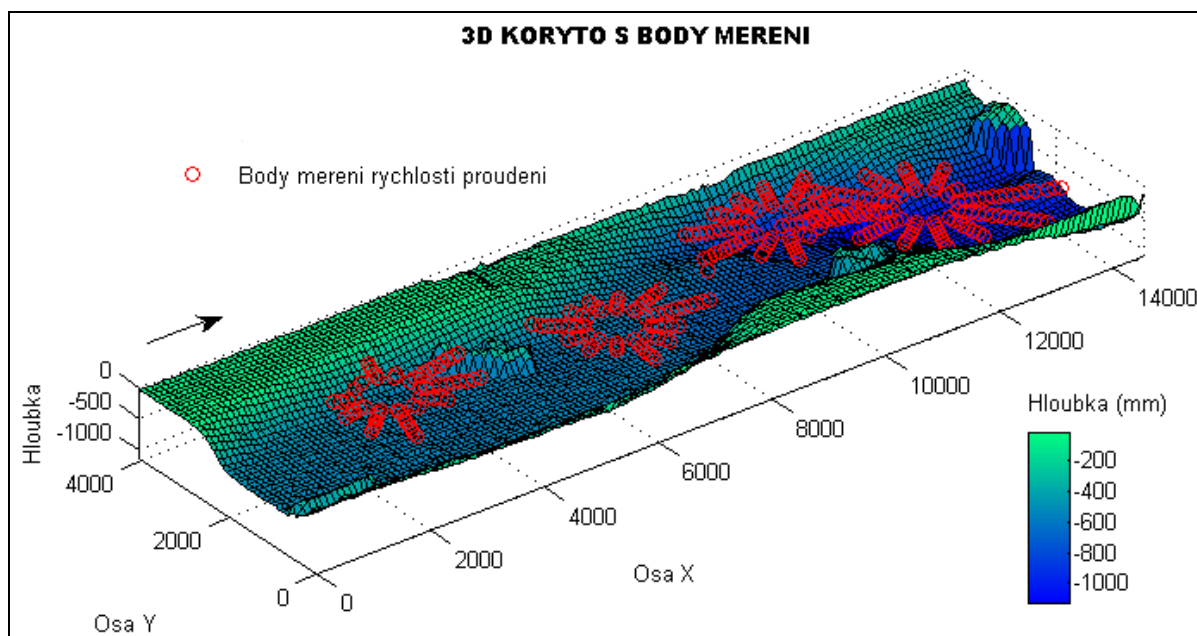
V této kapitole výpočetně zpracujeme data z horního úseku Dunaje za pomoci definovaných statistických modelů (viz. kapitola 6). Výpočty provedeme pro data z měření blízko hladiny (hloubka 150 mm), jelikož v této hloubce je rychlost proudění nejvyšší a směrem od ní se snižuje. V této hloubce byl také, odečtením maximální a minimální rychlosti, zjištěn největší rozsah měření. Největší hodnota rozsahu s velkou pravděpodobností zajišťuje, že bude rozdíl v hodnotách výsledného gridu dobře viditelný.

Vizualizací měření v programu VOR jsme také zjistili, že data z měření rychlostí ve tvaru hvězdic nevystihují celé koryto ve všech hloubkách. Zobrazíme-li měření z hloubky -250 mm zjistíme, že měření proběhla pouze v dolním úseku sledovaného koryta (viz. obr. 16), což může být způsobeno rozdílem hloubek v horní a dolní části úseku.



Obr. 16: 3D koryto s body měření rychlosti pro hloubku -250 mm

Data z měření v hloubce -150 mm postihují celý průběh koryta, a proto byly naměřené body z těchto hloubek použity pro výpočet obou dále popsáných modelů. V této hloubce jsme také odhadovali koeficienty úbytku rychlosti vzhledem k proudnici s největší rychlostí a rychlost na této proudnici.



Obr. 17: 3D koryto s body měření rychlosti pro hloubku -150 mm

9.1 Kriging

Kriging má velkou řadu nastavení, a proto záleží na nastavení jednotlivých parametrů pro získání kvalitních výstupů. V našem případě předpokládáme, že existuje proudnice s největší rychlostí proudění, od které se bude rychlost proudění snižovat. Rychlost snižování však musíme vystihnout na základě dalšího zpracování dat.

Nejprve načteme knihovnu *geoR* a textový soubor s měřením ve tvaru příčných profilů. soubor *profily_R.txt*, který obsahuje tato měření, má 3 sloupce se souřadnicemi x, y a hodnotu skalární rychlosti proudění.

```
>> library(geoR) %nacteni knihovny
>> data=read.table("profily_R.txt") %nacteni souboru
      profily_R.txt
```

Následující kód vytvoří z textového souboru objekt *geodata*, který je nutný pro další zpracování Krigingu. Objekt vyžaduje jako vstupní hodnoty souřadnice x, y a hodnoty rychlostí proudění. Příkazem *summary(gdata)* můžeme zobrazit některé statistiky, které charakterizují data. Příkaz *plot(gdata)* pak charakterizuje datovou sadu pomocí grafů.

```
>> gdata=as.geodata(data, coords.col=c(1,2), data.col=3)
```

Následující syntaxe vytvoří pro geodata variogram, jehož podobu již známe z obr. 8 na straně 23. Z tohoto obrázku můžeme vyčíst, že rozptyl hodnot u blízkých bodů (přibližně 0.5 m) je vzhledem k celkovému rozptylu malý. Avšak od této vzdálenosti prudce narůstá až k vzdálenosti přibližně 3 m, od které je hodnota variogramu již konstantní (přibližně 0.05). Můžeme říci, že body ve vzdálenosti větší než 3 m již na sebe prakticky nemají vliv. Tato podoba variogramu je logická, jelikož šířka koryta je maximálně 4 m, a hodnoty jednotlivých měření u laminárního proudu by se měly navzájem ovlivňovat hlavně v příčném směru.

```
>> variog=variog(gdata)
>> plot(variog,xlab="h",ylab=expression(gamma * '(h)'))
```

Abychom vystihli tento strmý nárůst variogramu, použijeme exponenciální model. Pro vytvoření modelu se používá objekt *vario.fit*, jehož vstupními parametry jsou variogram a počáteční nastavení rozsahu variogramu. U modelu nastavíme práh variogramu na 0.05 a rozsah na 3000 (metoda *cov.pars()*). Hodnotu nugget efektu nastavíme na 0.02 (metoda *nugget()*).

```
>> ini.vals=expand.grid(seq(0,var(gdata$data),
  l=100),seq(0,12000,l=100)) %pocatecni rozsah variogramu
>> vario.fit=variofit(variog,ini=ini.vals) %objekt modelu
>> vario.fit$cov.pars=c(0.05,3000) %nastaveni prahu a
  rozsahu modelu
>> vario.fit$nugget=0.02 %nugget efekt modelu
>> lines(vario.fit) %zobrazeni modelu
```

Nyní načteme soubor s body břehových linií (*border.txt*), který obsahuje x, y souřadnice levého a pravého břehu v tomto pořadí.

```
>> bor=read.table("border.txt")
>> bor=bor[,1:2]
```

V dalším kroku vytvoříme prázdný grid, který bude mít rozměry výsledného gridu. Nastavíme škálu pro barevné provedení gridu (v systému rgb) a dále nastavíme názvy jednotlivých souřadnicových os.

```
>> loc=expand.grid(seq(0, 16000, l = 200),seq(0,
  4000,l=100))
>> image(kc2,borders=bor,col = rgb(red=0,
  green=0,blue=(13:29)/29,),xlab="Osa X",ylab="Osa Y")
```

V poslední fázi nastavíme funkci *contour()* vzhled izotach a provedeme výpočet Krigingu.

```
>> contour(kc2,n=10,borders=bor,add=T,col="white")

>> kc2=krige.conv(gdata,loc=loc,krige=krige.control
(obj.model= vario.fit))
```

Podle výše zmíněného postupu vznikl grid Krigingu naměřených rychlostí proudění (příloha 2) na základě exponenciálního modelu. Na příloze 2 vidíme rozložení rychlosti ve sledovaném korytě, kde body, ve kterých nebyla rychlost měřena, jsou vypočítány na základě váženého průměru známých bodů. Světle modrá místa představují největší rychlosti proudění, od kterých se rychlost směrem k břehům snižuje (tmavě modrá barva). Jak je vidět na tomto obrázku, metoda Kriging odhadla tvar proudnice s největší rychlostí víceméně správně. Tato proudnice v horním rovném úseku se nachází prakticky ve středu koryta. Při kontaktu s konvexním břehem dochází k zakřivení její dráhy směrem k břehu konkávnímu.

Nyní vytvoříme Kriging i pro sférický model tak, že předešlý postup prakticky opakujeme avšak nastavíme proměnnou *cov.model* objektu *vario.fit()* na hodnotu *spherical*.

```
>> vario.fit$cov.model = "spherical"
```

Výsledný grid pro sférický model je zobrazen v příloze 3. Grid sférického modelu se od předchozího liší pouze nepatrně v odhadnutých hodnotách. Podle obou těchto modelů můžeme říci, že Kriging dokáže zachytit úbytek rychlostí ve směru od naměřených hodnot s největší rychlostí a odhadnout tak teoretický tvar proudnice.

Podle zobrazených izotach však zjišťujeme, že v korytě vznikají umělé turbulence. Turbulence jsou na zobrazených gridech znázorněny uzavřenými izotachami. Tyto umělé turbulence jsou více patrné na sférickém modelu a nacházejí se v oblasti koryta s největší rychlostí. Jelikož je z dalších výzkumů dokázáno, že sledované koryto má povahu laminárního proudění, u kterého mají všechny proudnice souběžný průběh a nedochází k jejich křížení (viz. 5.1.2.), můžeme tedy tvrdit, že tyto turbulence vznikly pouze vlivem modelu a můžeme je tedy považovat za umělé.

Ze zjištěných faktů vyplývá, že Kriging je vhodný pouze pro prvotní vizualizace rozložení rychlosti proudění, jelikož je schopen zobrazit teoretické rozložení rychlostí, ale vytváří umělé chyby.

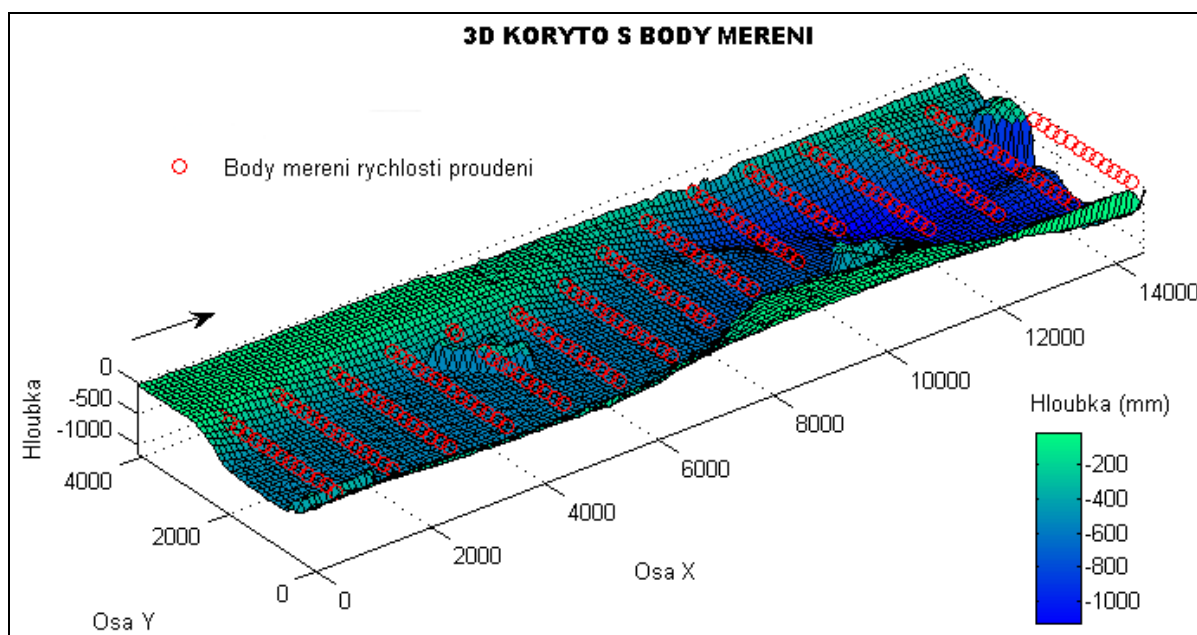
9.2 Model rychlosti proudění v řece

Nyní aplikujeme model rychlosti proudění v říčním korytě (TUČEK, 2009) na reálná data. K zpracování dat do výsledné podoby gridu budeme využívat aplikaci VOR, ve které můžeme provést všechny kroky modelu.

Postup bude probíhat v následujících krocích:

- načtení souboru labor.txt a crossmeasure.txt
- vizualizace a charakterizování úseku řeky
- vytvoření břehových linií
- 2D interpolace

Tento model rychlosti proudění v řece budeme aplikovat na data z měření rychlosti proudění v příčných profilech v hloubce -150 mm. Načteme tedy soubory do aplikace VOR pomocí menu *Soubor – Otevřít koryto* a menu *Soubor – Otevřít měření*. Po načtení byly automaticky vypočítány statistiky pro koryto a z nich zjišťujeme, že největší rychlost

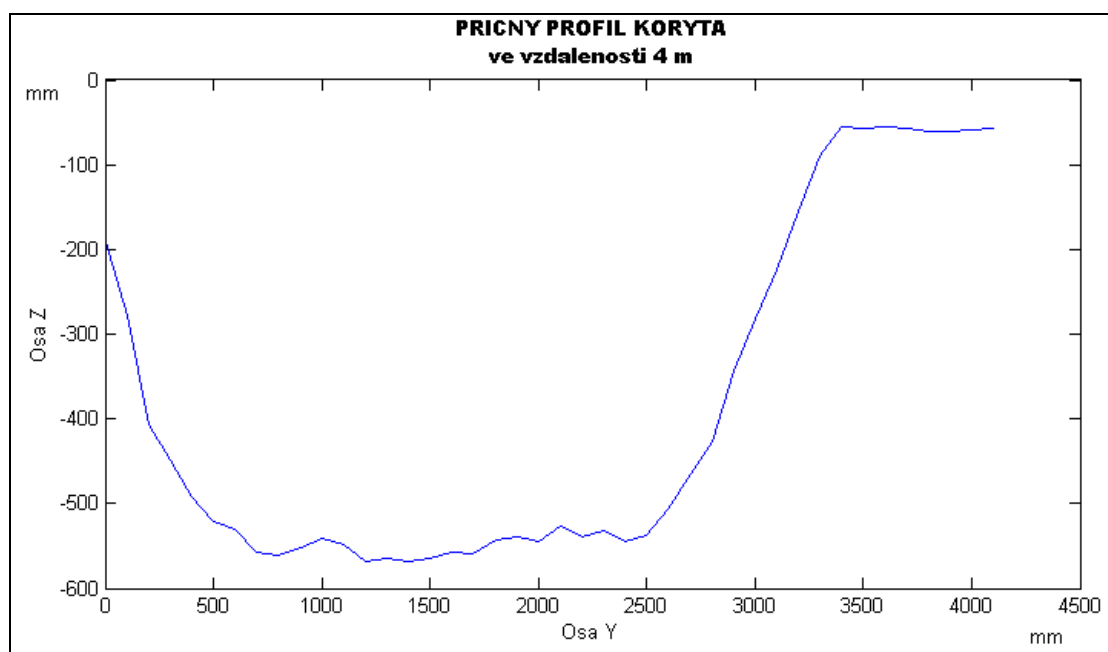


Obr. 18: Použité body měření

proudění je 0.74 ms^{-1} , tuto rychlost by měla hledaná proudnice teoreticky obsahovat. V našem modelu však celé proudnici zadáme konstantní rychlost 0.71 ms^{-1} , která byla

odhadnuta na základě dalších statistických výpočtů Mgr. Pavla Tučka. Dále zjišťujeme, že příčné profily byly naměřeny ve 203 bodech a jsou rozprostřeny po celém úseku koryta. Jednotlivé profily jsou od sebe vzdáleny 1 m, máme tedy k dispozici 15 příčných profilů (viz. obr. 18). Pro odhad proudnice použijeme 15 bodů s nejrychlejší rychlostí proudění.

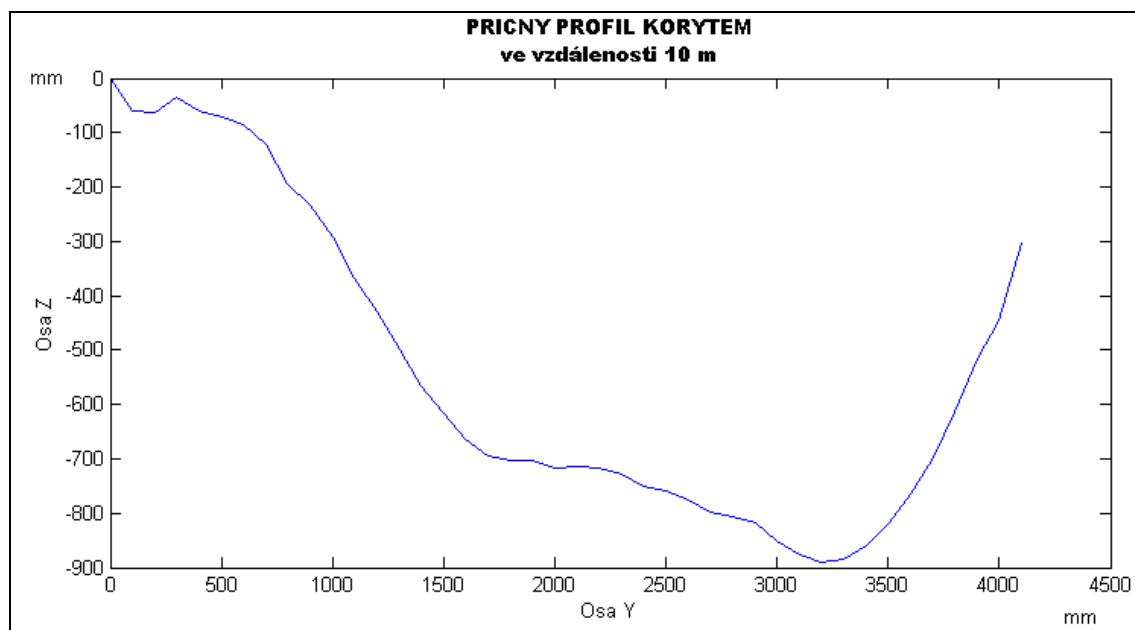
Již podle obrázku 18 je zřejmé, že studované koryto je v horní části spíše rovné a v dolní části se stáčí do oblouku. Příčný profil v horní části koryta (obr. 19) se svým tvarem blíží symetrickému lichoběžníku, což podle teorie z kapitoly 5.2.2 indukuje rovný úsek. V této části koryta tedy předpokládáme rovný průběh proudnice.



Obr. 19: Příčný profil ve vzdálenosti 4 m

Oproti tomu na obrázku 20, který zobrazuje profil ve vzdálenosti 10 m je vidět, že pravý břeh (konvexní) je strmější a hlubší. Oproti tomu levý břeh (konkávní) stoupá pozvolněji, a proto předpokládáme, že proudnice o největší rychlosti bude od středu vychýlena směrem k pravému břehu. Z obou profilů můžeme usuzovat, že tvar proudnice s největší rychlostí proudění se bude v horním úseku koryta blížit středové čáře¹⁹ a přibližně ve střední části se od ní začne vychylovat.

¹⁹ Středová čára je čára spojující všechny středy.



Obr. 20: Příčný profil ve vzdálenosti 10 m

Pokusíme se tedy nyní dokázat předešlá tvrzení a vytvořit 2D rozložení rychlosti proudění v programu VOR. Nejprve musíme odhadnout koeficienty břehů, které se v naší aplikaci vytváří pomocí uživatelem zadaných bodů, z kterých se funkcí *polyfit* (viz. 7.1.3) vypočítají koeficienty polynomu. Body pro výpočet koeficientů se zadávají přes interaktivní mód nástroje *Vytvořit koeficienty břehů* (viz. příloha 4) a měly by být voleny tak, aby obsahovaly body koryta s hloubkou -150 mm. Počet zadávaných bodů závisí na členitosti a rozložení měřených bodů v určité hloubce, pro hloubku -150 jsme použili 30 bodů pro pravý a 35 bodů pro levý břeh. Výstupem nástroje je soubor obsahující koeficienty pravého a levého břehu. Počet koeficientů závisí na zadaném stupni polynomu.

Máme-li soubor s koeficienty břehové linie, můžeme přistoupit k poslednímu kroku. Podle teorie modelu, prezentovaném v kapitole 6.3, potřebujeme pro realizování modelu znát koeficienty úbytku κ_1 a κ_2 . Oba interpolační nástroje pro výpočet modelu v programu VOR jsou navrženy tak, aby řešitel tyto koeficienty spolu s rychlostí proudění celé proudnice (viz. v této kapitole výše) zadával sám. Výpočet těchto hodnot není v současné době zautomatizován, jelikož je jejich početní studie stále ve vývoji (viz. [2]). Pro námi zvolenou situaci mají oba koeficienty hodnotu 0.00035.

Známe všechny potřebné parametry pro odhad rychlosti proudění, a proto aplikujeme nástroj *2D interpolace (interaktivní)* na načtená data. Zadáme cestu k odhadnutým

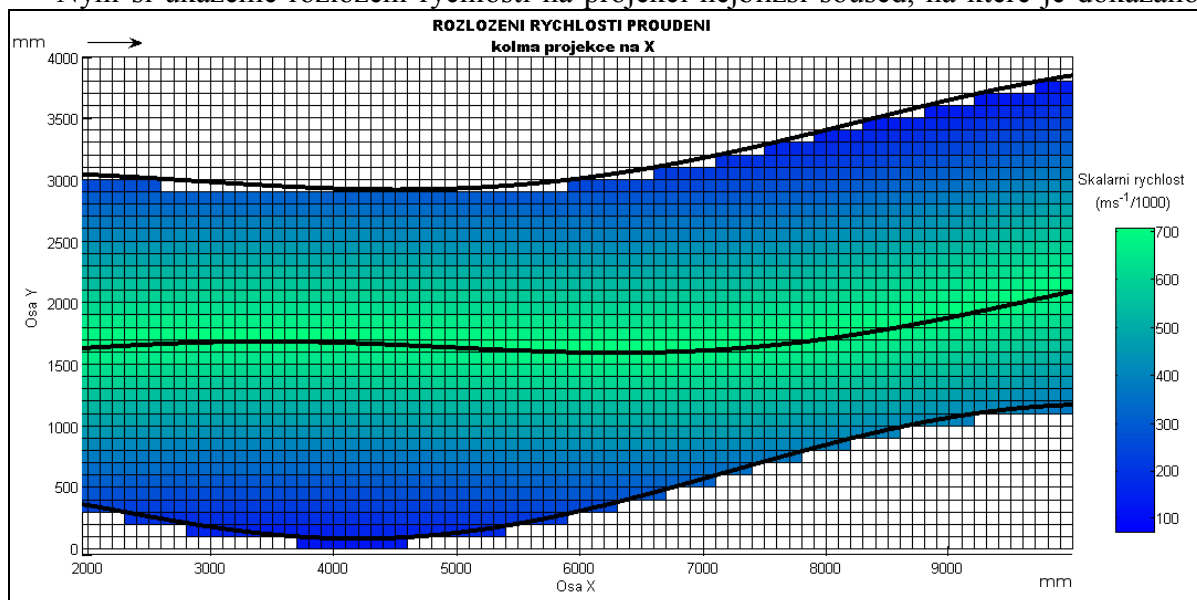
koeficientů břehů, hodnotu rychlosti na proudnici nastavíme na 0.71, koeficienty kappa na 0.00035 a počet bodů proudnice na 15. Vytvoříme kolmou projekci s polynomem 5-tého řádu. Jelikož jsou body měření rozprostřeny po celém korytě, můžeme použít volbu *Přichytávat vše*. Po spuštění tlačítka *Start* se v interaktivním okně (viz. příloha 5) zobrazí body příčného profilu. Tyto body jsou barevně odlišeny podle naměřené rychlosti proudění. Dle legendy označíme v každém profilu jeden bod s největší rychlostí proudění.

Aplikace VOR nyní provede odhad tvaru proudnice na základě zadaných bodů měření. Rozsah 2D koryta je omezen odhadnutými polynomy břehových linií, které byly ve skriptu uzavřeny do polygonu. Dále je na základě délky a šířky koryta vytvořen prázdný grid, který je polygonem břehových linií ořezán pomocí funkce `poly2mask` (viz. 7.1.2).

V této fázi již máme proudnici a ořezaný grid, který má ve svých buňkách hodnoty 1. Nyní je pro každý bod gridu, který má hodnotu 1 nalezena projekce na proudnici na základě zvolené metody hledání (nejbližší soused nebo kolmá projekce). Pro každý bod ořezaného gridu a jeho projekci na proudnici je aplikován vzorec 20 ze stránky 26, podle kterého je odhadnuta hodnota v bodě gridu.

Vizualizované rozložení rychlosti proudění (obr. 21) ukazuje úbytek rychlosti proudění od proudnice s největší rychlostí směrem k břehovým liniím. Protože byl použit pro výpočet odhadu proudnice, vykazuje výsledný grid laminární rozpad rychlostí bez vzniku umělých turbulencí.

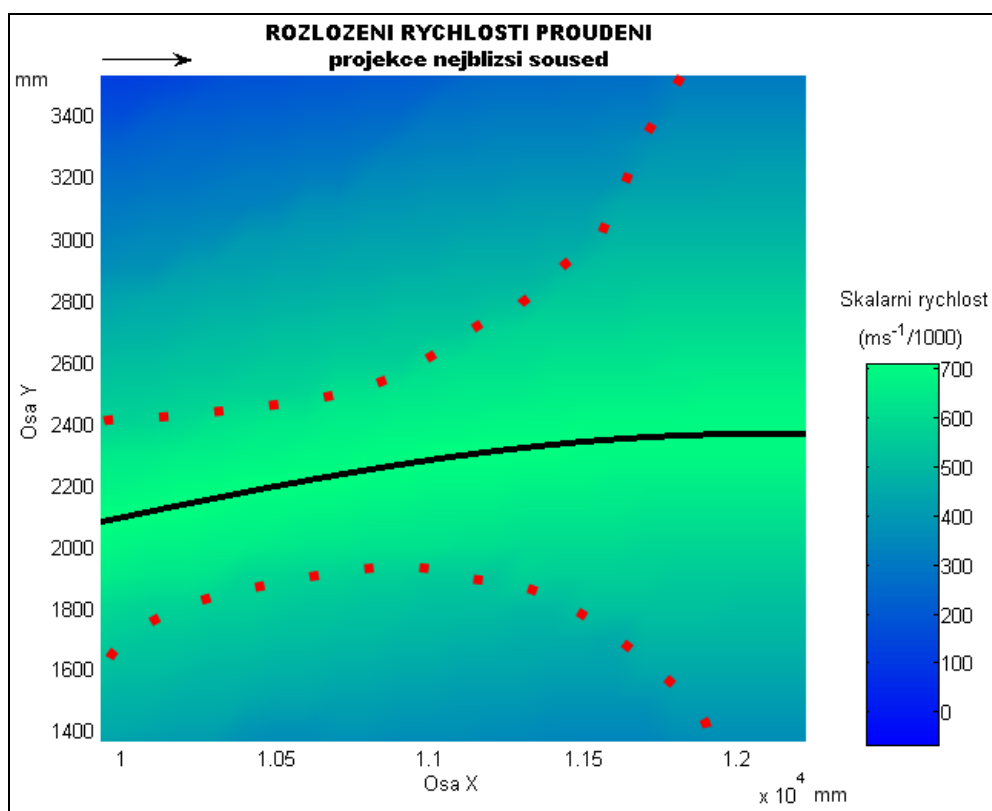
Nyní si ukážeme rozložení rychlosti na projekci nejbližší soused, na které je dokázáno,



Obr. 21: Detail gridu rozložení rychlosti proudění (kolmá projekce X)

Obr. 22 byl vytvořen za použití projekce nejbližší soused, podle stejných parametrů jako předchozí gridy. Abychom lépe odlišili jednotlivé rychlosti proudění, je obrázek zobrazen bez hranic pixelů. Na tomto obrázku vidíte, že se vytvářejí umělá „oka“, která se v podobě soustředných kružnic vyvíjejí od levého horního rohu směrem k pravému dolnímu rohu. Podobný efekt je vidět od hodnoty $x = 1.1 \cdot 10^4$ směrem k proudnici.

Z obrázku 22 vyplývá, že způsob hledání projekce ovlivňuje výsledné rozložení zvláště v nerovných úsecích, kde má proudnice tendenci se „vlnit“. V rovných úsecích jsou tyto umělé útvary zanedbatelné.

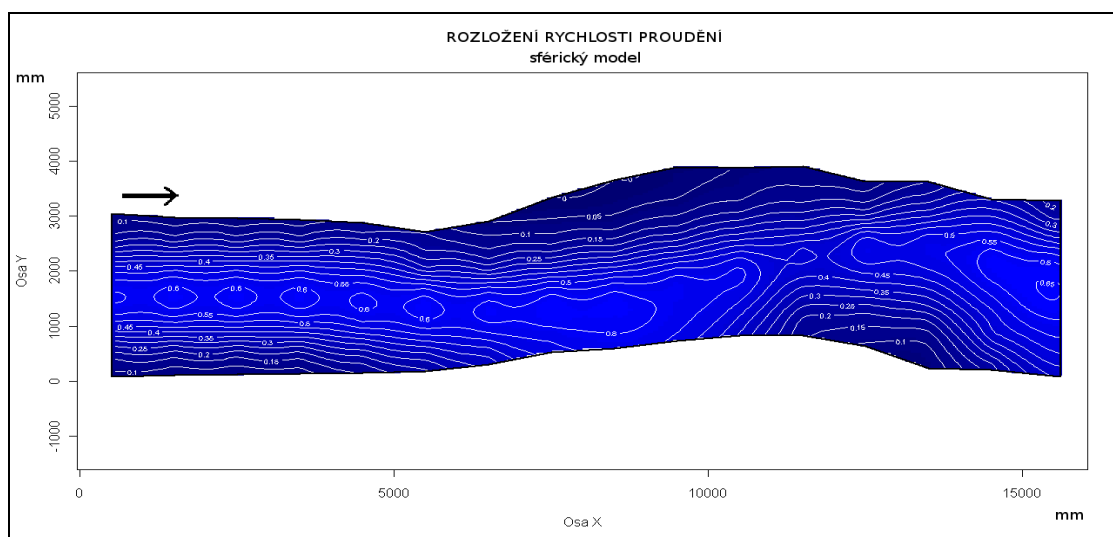


Obr. 22: Detail rozložení rychlosti proudění (červenou tečkovanou čarou jsou zvýrazněna umělá "oka")

10 Výsledky

Výsledkem diplomové práce je výpočetní zpracování dvou statistických modelů, pomocí kterých se autor a vedoucí práce snažili vystihnout rozložení rychlostí proudění v konkrétním říčním korytě. Na přiloženém CD (příloha 1) jsou k dispozici všechny programové kódy, které byly použity pro zpracování statistických modelů a jsou tedy jedním z výsledků diplomové práce.

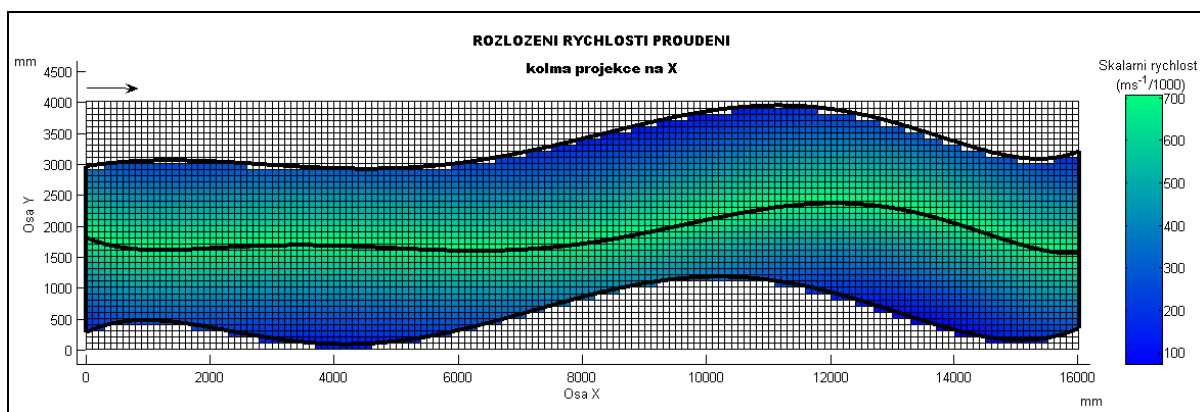
Model Kriging byl zpracován v programu R a jeho výsledkem je grid rozložení rychlostí proudění. Kriging odhadl teoretický průběh proudnice a jeho vizualizace připomíná rozložení rychlostí při laminárním proudění. Jeho chybou je, že vytváří umělé turbulence, které jsou u tohoto typu proudění nepřipustné. Kriging se tedy stal prvotním náhledem na studovaný problém.



Obr. 23: Kriging (sférický model)

Pro účely druhého modelu byla vytvořena aplikace VOR (Velocity Of the River), která obsahuje základní nástroje pro načtení dat koryta a dat rychlostí proudění. VOR je napsán v jazyce C++ s použitím knihovny Qt a jazyka Matlab, umožňuje tak aplikovat statistický model pro rozložení rychlosti proudění (TUČEK, 2009) na reálná data. Jeho předností jsou další doprovodné nástroje, které napomáhají při charakterizování úseku koryta. Aplikace je modulární a je možné ji dále rozvíjet podle konkrétních potřeb modelu. Již v této fázi vývoje může být aplikována na jiné datové sady s požadovanou strukturou měření (viz. kapitola 6.1).

Výsledný grid, který je výstupem z programu VOR představuje úbytek rychlosti proudění směrem od proudnice s největší rychlostí k břehům koryta podle teorie o laminárním proudění. Rychlost proudění se snižuje na základě koeficientů úbytku, vzdálenosti odhadovaného bodu a jeho projekce na proudnici. Na obrázku 24 je zachycen tvar proudnice s největší rychlostí, který se podle očekávání v horní části úseku blíží středové čáře. Dále je vlivem zakřivení koryta průběh proudnice vychýlen od středové čáry a začíná se „vlnit“.



Obr. 24: Grid rozložení rychlosti proudění

Hlavní výsledek této práce je navržený algoritmus zpracování zadaného statistického modelu, který se bude dále vyvíjet. Všechny programové kódy jsou navrženy tak, aby mohly být dále využity a nebyly pouze součástí této diplomové práce.

Nemůžeme říci, že výsledky práce jsou konečným popisem chování jevu v reálné situaci, jelikož zkoumaný model Mgr. Pavla Tučka v současné fázi zanedbává některé podmínky, které ovlivňují rozložení rychlostí v přírodním korytě. Tyto podmínky však mohou být dále zahrnuty v koeficientech úbytku κ_1 a κ_2 . Tyto výsledky mají však pro další modelování stěžejní roli, jelikož jsou první aplikací vyvíjeného statistického modelu na reálná data. Na základě ukázkových postupů této diplomové práce můžeme tedy tento algoritmus aplikovat na jiná koryta s podobným chováním.

11 Diskuze

Současná fáze modelu definovaného v kapitole 6.3 se striktně zabývá laminárním prouděním, což se může jevit jako jeden z klíčových problémů celé práce, avšak sledované koryto bylo velice pečlivě vybíráno tak, aby se chování vody v tomto úseku co možná nejvíce blížilo tomuto druhu proudění. Rychlost proudění, morfologie koryta, sklon koryta a další kritéria pro výběr koryta indikující laminární povahu proudění, nemůžou ale zaručit vznik fluktuací v rychlostech proudění, které jsou způsobeny zakřivením dráhy částic či vznikem drobných vírů. Tyto parametry by ovšem mohly být zahrnuty v dalších fázích modelování.

Velice zajímavým směrem by mohlo být testování různých projekcí (viz. kapitola 6.3.4), jelikož není v současné době úplně jasné, který bod na proudnici nejlépe vystihuje odhad v bodě, který se snažíme zjistit. Bod na proudnici se může nacházet vzhledem k odhadovanému bodu blíže proti proudu, může být v jeho úrovni nebo dokonce dále směrem po proudu. Blížší srovnání těchto projekcí a jejich vliv na modelované rozložení rychlostí by mohlo být vhodným námětem pro další práci.

Jelikož v naší práci operujeme pouze s jednou sadou dat, jsme těmito měřeními do jisté míry omezeni. Bylo by vhodné aplikovat model na další úseky koryta, avšak vzhledem k obtížnému získávání odpovídajícího počtu dat se tento návrh bude asi odsouvat na pozdější fáze modelu, kdy bude blíže specifikováno, jak se mají data pro účel modelu sbírat, jaký má být jejich ideální počet, rozmístění v prostoru atd.

Modely se obecně konstruuji tak, aby se jejich výsledky shodovaly s realitou, což je prakticky nemožné. Abychom se pokud možno blížili reálné situaci, je žádoucí, aby bylo provedeno zpětné porovnání výstupů s reálnými daty. Rozsah této práce již neumožnil, aby byla data dále testována např. bumerangovým testem či jinými metodami pro srovnání přesnosti odhadu s reálnými daty. V rámci dalších testování modelu by mohly být srovnány výsledky z dat nejen v rámci datových sad z více říčních úseků, ale také datové sady pořízené na jednom úseku z různých časových období.

Softwarové hledisko je také diskutabilní. Program VOR je řešen za pomoci zkompileovaných Matlab skriptů, a proto je jeho editovatelnost podmíněna licenčním ujednáním. Existují i další nekomerční výpočetní nástroje, které by se daly použít pro matematické výpočty, ale jejich začlenění do programu VOR je programátorsky značně náročné.

12 Závěr

Diplomová práce Statistické modelování hydrologických jevů se zabývá rozložením rychlosti proudění v úseku koryta. Jejím hlavním cílem bylo odhadnout hodnoty rychlosti proudění v místech, která nebyla naměřena. Následně bylo úkolem navrhnout algoritmy a vytvořit programový kód, který zpracuje naměřená data a vizualizuje je do podoby gridu. Všechny stanovené cíle byly dosaženy.

Součástí diplomové práce jsou programové kódy programu R, Matlab, C++ a knihovny Qt, které ze zadaných statistických modelů vytváří grid rozložení rychlosti proudění. Největší výhodou těchto kódů je jejich aplikovatelnost na jiné datové sady, a proto se předpokládá jejich další využití.

Modely Kriging a rychlost proudění v řece (viz. kapitola 6.3) dokázaly vystihnout trend snižování rychlosti směrem od proudu s největší rychlostí. Kriging odhadl přibližný tvar proudu, avšak zobrazil turbulence v místech, kde se nepředpokládají. Druhý model již tyto nedostatky ošetřuje a vizualizuje rozložení rychlosti laminárního proudění správně, tedy tak, jak je popisováno v dostupné literatuře.

Prezentované výsledky jsou v současné době pouze výpočetním zpracováním statistických modelů a o jejich správnosti se můžeme jen dohadovat, proto nemohou být brány jako obecně platné závěry. Výstupy práce se začlení do probíhajícího modelu, jehož tutorem je Mgr. Pavel Tuček a budou podrobeny dalším analýzám.

13 Summary

The diploma thesis called „Statistical Modeling of the Hydrological Phenomena“ has been solved at the Department of Geoinformatics of Palacky University in Olomouc.

The aim of the thesis is to find estimation of velocity of the river flow in the locations where values were not measured. The algorithms were suggested and the program code was designed, which elaborated the measured data and visualized them to the form of a GRID.

In the theoretical part, it is mainly dealt with the theory of river flow, but also the other fields of hydrology are commented. The concept of the developing program, which is written in the program languages C++, Qt and Matlab, is also included.

Two statistical models, which are used in the practical part are defined here. The first model is a generally known geostatistical method used for the estimation of unobserved point from the nearby measured points. The second model tries to estimate the streamline with the highest velocity of flow. This streamline is then used for the calculation of any other point in the river section.

The result of the thesis is the GRID distribution of velocity of flow in the river section, based on the statistical models mentioned above.

The outcomes of the thesis will become a part of a model solved by Mgr. Tuček and will be submitted for further analysis.

Literatura

- [1] TUČEK, Pavel, MAREK, Jaroslav. The Velocity of the river flow. *Tatra Mountains / Mathematical Publications*. 2006, no. 14, s. 241-248.
- [2] TUČEK, Pavel, MAREK, Jaroslav. Statistical Processing of Water Flow Data : Water Flow Modelling. *Acta Universitatis Palackianae Olomucensis : Facultas Rerum Naturalium, Mathematica* [online]. 2009 [cit. 2009-05-01].
- [3] PRATA, Stephen. *Mistrovství v C++*. 3. aktualiz. vyd. Brno : Computer Press, a.s., 2007. 1119 s. ISBN 978-80-251-1749-1.
- [4] ECKEL, Bruce. *Thinking in C++*. 2nd edition. [s.l.] : [s.n.], 2000. 2 sv. (857, 592 s.). Dostupný z WWW: <<http://www.datastore.cz/bruceeckel/>>.
- [5] MATTAS, Daniel. *Praktické cvičení z hydrometrie*. [s.l.] : ČVUT v Praze, Stavební fakulta, 2001. 22 s. Dostupný z WWW: <<http://hydraulika.fsv.cvut.cz/Predmety/HEMM/hydrometrování/hymetr.pdf>>.
- [6] BOITEN, Wubbo. *Hydrometry*. [s.l.] : Taylor & Francis, 2000. 246 s. 2. ISBN 9054104198.
- [7] HUGGETT, Richard John. *Fundamentals of Geomorphology*. London : Routledge, 2003. 408 s. 1.
- [8] *The R Project for Statistical Computing* [online]. [2009] [cit. 2009-03-30]. Dostupný z WWW: <<http://www.r-project.org/>>.
- [9] LEOPOLD, Luna B., WOLMAN, M.Gordon , MILLER, John P. *Fluvial Processes in Geomorphology*. New York : Dover Publications, Inc., 1995. 522 s.
- [10] *The Mathworks* [online]. 1994-2009 [cit. 2009-03-01]. Dostupný z WWW: <<http://www.mathworks.com/>>.
- [11] NOVÁK, Jiří, PULTAROVÁ, Ivana, NOVÁK, Pavel. *Základy informatiky : Počítačové modelování v Matlabu*. [s.l.] : Česká technika - nakladatelství ČVUT, 2005. 170 s. ISBN 80-01-03308-2.
- [12] MATOUŠEK, Václav. *1141 HYA(Hydraulika) : Přednáškové slidy předmětu* [online]. Praha : FSv ČVUT, 2009 [cit. 2009-04-02]. Dostupný z WWW: <http://hydraulika.fsv.cvut.cz/users/matousek/MujWeb_soubory/Page437.htm>.
- [13] *Kinematika : Bodu* [online]. VUT Brno : Ústavu mechaniky těles, mechatroniky a biomechaniky, 2009 [cit. 2009-04-03]. Dostupný z WWW: <<http://www.umt.fme.vutbr.cz/~kpellant/2%20Kinematika%20Bodu.htm>>.
- [14] ROTH, Peter M., WINTER, Martin, JUNG, Bernhard. *Compiling MATLAB M-Files for Usage Within C-Applications*. [s.l.] : [s.n.], 2005. 25 s. Dostupný z WWW: <<http://www.icg.tu-graz.ac.at/Members/pmroth/projects/m2c/HowToCompile.pdf/view>>.

- [15] BRDIČKA, Miroslav, SAMEK, Ladislav, SOPKO, Bruno. *Mechanika kontinua*. 2. opravené vyd. Praha : Academia, 2000. 799 s. ISBN 80-200-0772-5.
- [16] DOUGLAS, J F, GASIOREK, J M, SWAFFIELD, J A. *Fluid Mechanics*. London : Pitman publishing inc., 1979. 721 s. ISBN 0-273-00462-X.
- [17] NETOPIL, Rostislav. *Základy hydrologie povrchových a podpovrchových vod*. Praha 1 : Státní pedagogické nakladatelství n. p., 1970. 220 s.
- [18] *Qt : Online Reference Documentation* [online]. 2008 [cit. 2009-04-09]. Dostupný z WWW: <<http://doc.trolltech.com/>>.
- [19] KŘÍŽ, Vladislav. *Hydrometrie*. 1. vyd. Praha : Státní pedagogické nakladatelství, 1988. 176 s.
- [20] BLANCHETTE, Jasmin, SUMMERFIELD, Mark. *C++ GUI Programming with Qt 4*. 1st edition. Massachusetts, USA : [s.n.], 2006. 537 s. Dostupný z WWW: <<http://www.qtrac.eu/C++-GUI-Programming-with-Qt-4-1st-ed.zip>>. ISBN 0-13-187249-4.
- [21] BARTSCH, Hans-Jochen. *Matematické vzorce*. Spálená 51, Praha : SNTL - Nakladatelství technické literatury, n.p., 1983. 832 s.
- [22] DUTTER, Rudolf. *Geostatistics*. Vienna : University of Technology , 2000. 146 s.
- [23] ARMSTRONG, Margaret. *Basic linear geostatistics*. Germany : Springer, 1998. 149 s. ISBN 3-540-61845-7.
- [24] KARBAN, Pavel. *Výpočty a simulace v programech Matlab a Simulink*. [s.l.] : [s.n.], 2006. 214 s. ISBN 978-80-251-1448-3.
- [25] RIBEIRO, Paulo J., DIGGLE, Peter J. *GeoR : Package for Geostatistical Data Analysis : An illustrative session*. [s.l.] : [s.n.], 2006. 24 s. Dostupný z WWW: <<http://leg.ufpr.br/geoR/#introduction>>.

Seznam příloh

Příloha 1.: CD/DVD

Příloha 2.: Grid Krigingu pro exponenciální model

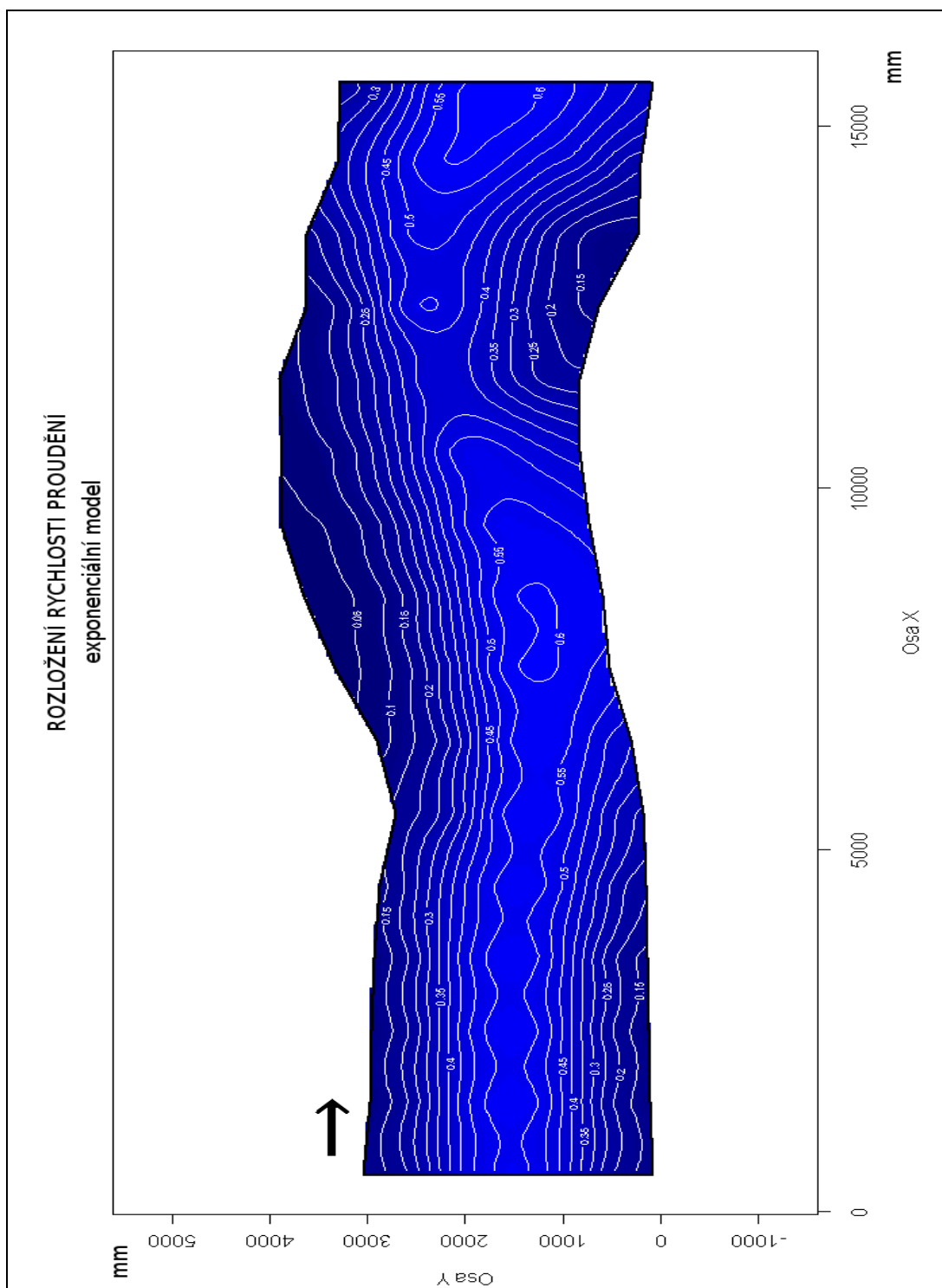
Příloha 3.: Grid Krigingu pro sférický model

Příloha 4.: Interaktivní okno pro vytvořený proudnice

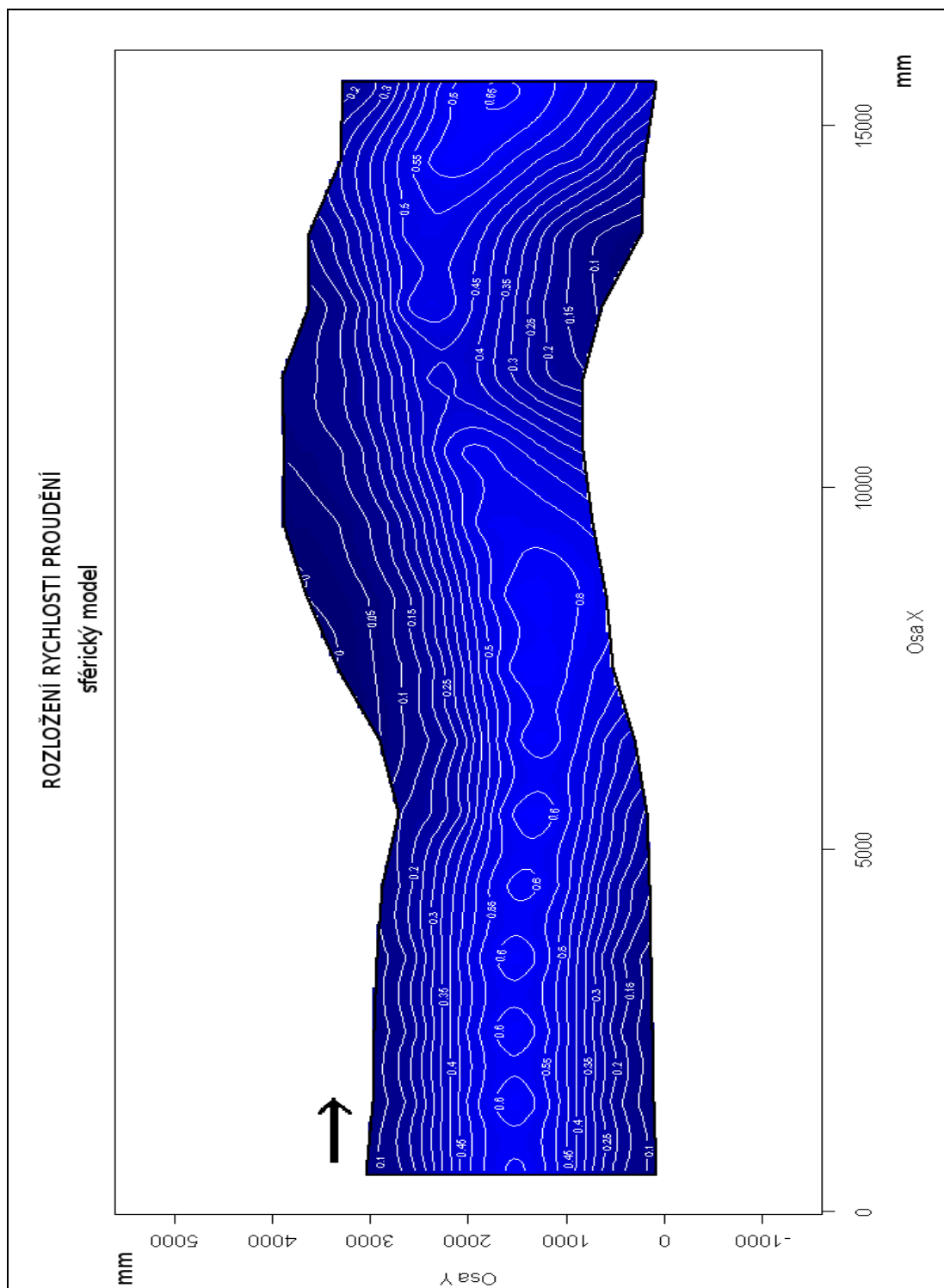
Příloha 5.: Interaktivní okno pro vytvořený břehové linie

Příloha 6.: Grid rozložení rychlosti proudění na základě statistického modelu Mgr. Pavla
Tučka

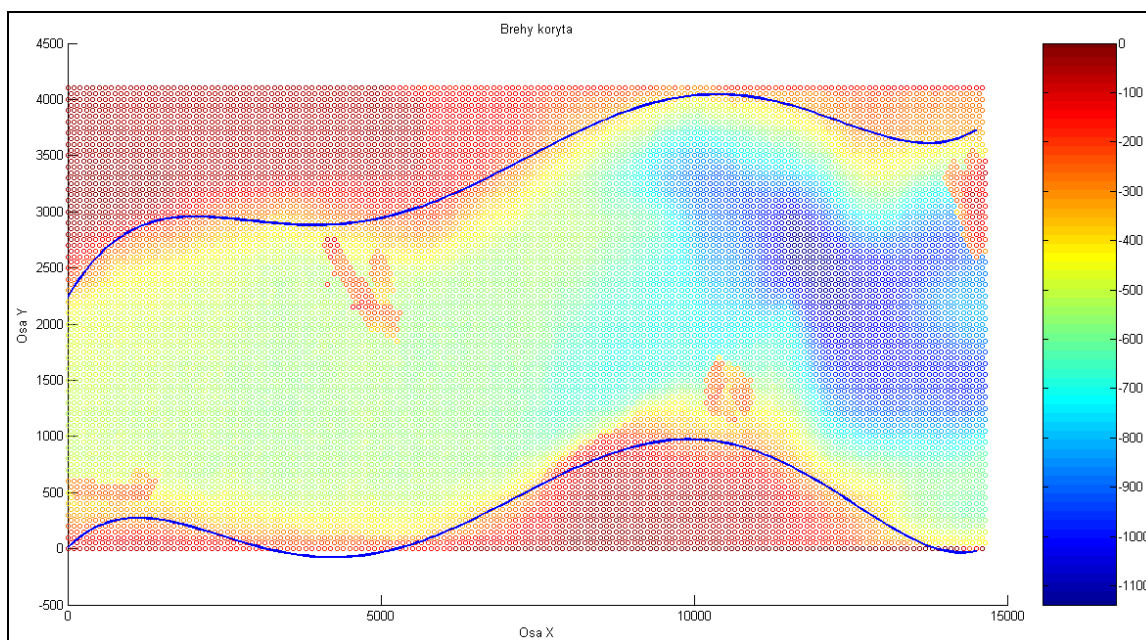
Příloha 2.: Grid Krigingu pro exponenciální model (čím tmavší modrá barva, tím menší rychlost proudění; bílá barva – izotachy)



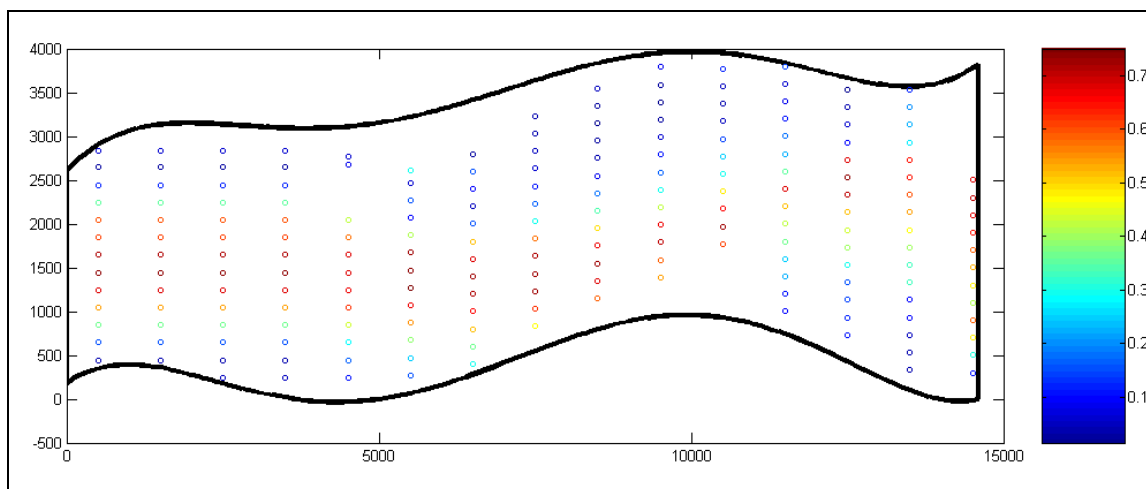
Příloha 3.: Grid Krigingu pro sférický model (čím tmavší modrá barva, tím menší rychlost proudění; bílá barva – izotachy)



Příloha 4.: Interaktivní okno pro vytvořený břehové linie



Příloha 5.: Interaktivní okno pro vytvořený proudnice



Příloha 6.: Grid rozložení rychlosti proudění na základě statistického modelu Mgr. Pavla Tučka

