Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Modelování říčního toku

Bc. Jan Beránek

Diplomová práce

srpen 2019

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2018/2019

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Jan Beránek**
Osobní číslo:	**I16205**
Studijní program:	**N2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Modelování říčního toku**
Zadávající katedra:	**Katedra softwarových technologií**

Zásady pro vypracování:

Práce se bude zabývat popisem rozložení rychlosti proudění v úseku koryta. Autor práce zpracuje literární rešerši k problematice statistického modelování hydrologických jevů s cílem sestavit vhodný model pro zpracování změřených rychlostí a popis chování toku v říčním korytu. Následným úkolem je navrhnout algoritmy a vytvořit programový kód, který zpracuje naměřená data a vizualizuje je do podoby gridu. Vytvořený model umožní aproximovat průběh koryta řeky a najít odhad hodnoty rychlosti proudění v místech, kde nebyla rychlost naměřena. Aplikace pomocí vhodných vizualizačních metod a grafických vyjádření dat a matematických modelů poskytne komplexní informaci o rozložení rychlostí v říčním korytu. Zejména bude nalezen průběh proudnice, na které je rychlost řeky největší. V práci budou popsány veškeré aplikované metody a bude vysvětleno uživatelské prostředí aplikace.

Rozsah grafických prací: **10**

Rozsah pracovní zprávy: **cca 40–50 stran**

Forma zpracování diplomové práce: **tištěná**

Seznam odborné literatury:

TUČEK Pavel a Jaroslav MAREK. The Velocity of the river flow. Tatra Mountains Mathematical Publications. 2006, no. 14, s. 241-248.
DOUGLAS, J. F. a J. M. GASIOREK, J. A. SWAFFIELD: Fluid Mechanics. London: Pitman publishing inc., 1979. 721 s. ISBN 0-273-00462-X.
NETOPIL, Rostislav. Základy hydrologie povrchových a podpovrchových vod. Praha 1 : Státní pedagogické nakladatelství n. p., 1970. 220 s.

Vedoucí diplomové práce: **Mgr. Jaroslav Marek, Ph.D.**
Katedra matematiky a fyziky

Datum zadání diplomové práce: **22. října 2018**

Termín odevzdání diplomové práce: **18. května 2019**

Ing. Zdeněk Němec, Ph.D.
děkan

Univerzita Pardubice
Česká republika

prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 17. listopadu 2018

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 21. 8. 2019

Jan Beránek

PODĚKOVÁNÍ

Děkuji vedoucímu mé diplomové práce, panu Mgr. Jaroslavu Markovi, Ph.D., za cenné rady, trpělivost a ochotu během psaní této práce. Dále bych rád poděkoval své rodině, která mi byla obrovskou oporou během celého studia.

ANOTACE

Cílem práce je prostudovat matematické nástroje pro modelování laminárního proudění a vytvořit aplikaci umožňující grafické znázornění vodního toku. V práci jsou uvedeny základní principy a technické prostředky pro měření rychlosti proudění. Matematické modely popisující laminární proudění tekutin umožňují nalézt odhady rychlostí vodního toku i v místech, kde měření rychlosti nebylo realizováno. Vytvořená softwarová aplikace graficky prezentuje rychlosti proudění na libovolných souřadnicích vodního toku, zobrazuje říční čáru a poskytuje numerické výsledky konkrétní studie. Pro vytvoření aplikace je využit objektově orientovaný programovací jazyk JAVA a jako vývojové prostředí jsou použity NetBeans. K některým matematickým výpočtům jsou použity knihovny třetích stran.

KLÍČOVÁ SLOVA

laminární proudění, říční čára, regresní model s podmínkou, odhad rychlosti, Java, NetBeans

TITLE

River flow modeling

ANNOTATION

The goal of this work is to study mathematical tools for laminar flow modelling and to create an application providing graphical representation of water flow. The thesis describes the mathematical modelling of laminar fluid flow, the possibility of measuring the velocity of this flow and using this knowledge to implement the application visualizing the behaviour of the watercourse. The application visualizes the flow velocities at any river surface coordinates, displays the river line, and provides numerical results for a particular study. The object-oriented programming language JAVA is used to create the application and NetBeans is used as a development environment. Third-party libraries are used for some mathematical calculations.

KEYWORDS

laminární proudění, říční čára, regresní model s podmínkou, odhad rychlosti, Java, NetBeans

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM ZKRATEK

PDF	Portable Document Format
GPS	Global Positioning System
HSV	Hue Saturation Vlaue
RGB	Red Green Blue
BLUE	Best linear unbiased estimator
Var	varianční matice

ÚVOD

Hydrologie se zabývá koloběhem vody na Zemi ve všech skupenstvích. Chováním tekutin se věnuje část fyziky, která se nazývá mechanika a kinematika tekutin. Základy těchto vědních disciplín vznikly v 18. století, kdy například David Bernoulli formuluje některé zákony. V současnosti se v inženýrském modelování pro popis chování tekutin používají softwary založené na aplikaci metody konečných prvků (Finite Element Method), výpočetní dynamiku tekutin (Computational Fluid Dynamics), soustavy mnoha těles (Multibody System) a optimalizaci. Pro popis pohybu vody ve vodních tocích používáme různé modely např. turbulentního (pro horní části toku) a laminárního proudění (dolní část toku). U laminárního proudění nedochází k míchání vrstev vody a nevznikají víry. Geoinformatika pro modelování proudění, resp. jiných zeměpisných jevů, používá velmi často metodu 3D Krigingu založenou na tzv. variogramu.

První kapitola bude věnována matematickému modelování laminárního proudění. Ve studované úloze budou k dispozici změřené rychlosti toku v několika místech, u kterých byla také změřena i jejich poloha. Dále budou k dispozici souřadnice bodů na březích vodního toku. Část této kapitoly bude věnována metodám a přístrojům pro měření rychlostí kapalin, např. Pitotově trubici.

Ve druhé kapitole bude popsána implementace aplikace pro vizualizaci chování vodního toku, který splňuje předpoklady laminárního proudění. Pro vytvoření aplikace bylo zvoleno vývojové prostředí NetBeans. V této kapitole budou rozebrány jednotlivé třídy programu a využití knihoven třetích stran, které slouží k řešení některých matematických problému v této práci.

Ve třetí kapitole budou prezentovány grafické a numerické výsledky konkrétní studie, ve které byly k dispozici změřené prostorové souřadnice bodů a rychlostí proudění.

Poslední čtvrtá kapitola Apendix obsahuje uživatelský manuál pro vytvořenou aplikaci. Čtenář se zde může stručně seznámit s ovládacími prvky aplikace a s možnostmi načítání dat, exportu výsledků, nastavením vstupních parametrů naprogramovaných algoritmů, nastavením parametrů pro grafické zobrazení výstupů.

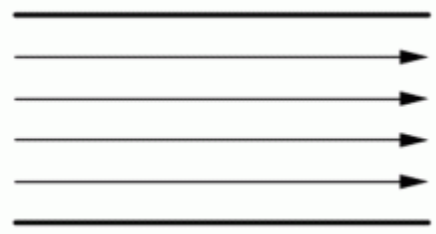
Při čtení této práce se předpokládají znalosti matematické statistiky v rozsahu předmětů vyučovaných v navazujícím oboru Informační technologie na Fakultě elektrotechniky a informatiky Univerzity Pardubice.

1 MATEMATICKÉ MODELOVÁNÍ PROUDĚNÍ

1.1 Laminární proudění

Kapalina patří společně s plyny mezi tekutiny a můžeme ji definovat jako jedno ze skupenství látek, při kterém jsou částice látky vzájemně relativně málo vzdálené, nejsou umístěny v pevných pozicích a mohou se pohybovat v celém objemu látky.

Jednosměrný pohyb kapalin se nazývá proudění, kapalina vždy proudí z místa vyššího tlaku (vyšší tlakové potenciální energie) do místa nižšího tlaku (nižší tlakové potenciální energie). Trajektorie pohybu jednotlivých částic kapaliny při proudění je nazývána proudnicí a rychlost částice v libovolném místě proudu je tečnou k proudnici. Každým bodem proudící kapaliny prochází v jednom okamžiku právě jedna proudnice. Proudění kapaliny, při kterém se proudnice nekříží a jsou rovnoběžné, se nazývá proudění laminární, částice se pohybují ve vrstvách, které se vzájemně nemíchají a nevznikají v kapalině víry. V reálné kapalině mezi těmito vrstvami vzniká vnitřní tření.



Obrázek 1: Ukázka laminárního proudění [1]

Proudění obsahující víry se nazývá proudění turbulentní, tímto typem proudění se však v této práci dále nebudeme zabývat.

Pokud se uvažuje ideální kapalina, předpokládáme konstantní tlak i rychlost. V žádném místě se nehromadí a průřezem za konstantní dobu ustáleně proudí stejný objem kapaliny.

Uvažujme vodorovnou trubici s ideální kapalinou, kde průřezem *s* proudí částice rychlostí *v*. Pak za jednu sekundu proteče kapalina o objemu *V*. Definujeme objemový průtok $Q_v = \frac{V}{t}$.

Objemový tok Q_v bude součinem obsahu průřezu *S* a rychlosti tekutiny *v* s jednotkou m^3/s. Vložíme-li do potrubí lopatkové kolo a jeho pohyb převedeme na počítadlo, můžeme objemový průtok měřit. Tento přístroj se nazývá Pitotova trubice.

V numerické studii je představena úloha věnována analýze konkrétních rychlosti proudění takto naměřených v reálném vodním toku.

Pitotova trubice funguje na principu spojení dvou trubic, každé s vodorovným a svislým ramenem. Vodorovné rameno první trubice má otvory do stran, takže hladina ve svislém rameni ukazuje výšku volné hladiny. Vodorovné rameno druhé trubice má otvor proti proudu, takže hladina ve svislém rameni ukazuje hydrodynamickou výšku. Rychlost proudící kapaliny (plynu) se určuje na základě rozdílu tlaků.

V práci bude dále uvažováno proudění, které není ideální. Budeme se snažit matematickými prostředky popsat reálny tok, který není ideální kapalinou. K dispozici budeme mít měření rychlostí v různých místech a hloubkách koryta, získané měřením pomocí Pitotovy trubice a GPS přístroje. Na základě těchto změřených údajů určíme proudnici s největší rychlostí, matematický model nám poté umožní odhadnout rychlosti proudění v bodech, kde jsme je neměřili. Získané poznatky nám umožní například odhadnout průtok skrz libovolný řez koryta.

Zdroje [1], [3], [4]

1.2 Základní pilíře matematického modelu proudění

Rychlost proudění se v říčním korytu mění. Důležitým milníkem je nalezení proudnice – křivky, na které je tečný vektor rychlostí největší. Po normále od této křivky směrem ke břehu rychlosti proudění postupně klesají až k nule. Proto je pro nás stěžejní matematicky popsat břehy vodního toku a proudnici. V další podkapitole 1.2.1 budou uvedeny vztahy pro popis břehů a proudnice. V podkapitole 1.2.2 bude tvar proudnice korigován na základě změřených rychlostí v různých bodech říčního koryta. V podkapitole 1.2.3 bude popsán model pro výpočet rychlosti toku v libovolném bodě koryta.

V dalším textu bude použito značení uvedené v Tab. 1.

Tabulka 1: Označení proměnných a jejich grafické značení

Proměnná	Význam	Indexy
$(\mathbf{x}^{\circ}, \boldsymbol{y}^{\circ}, \boldsymbol{z}^{\circ})$	Bod břehu řeky	$i = 1, \dots, N$
$(\mathbf{x}^{\times}, \boldsymbol{y}^{\times}, \boldsymbol{z}^{\times})$	Bod měření rychlosti toku	$j = 1, \dots, I$
$Y_d = f_d(\boldsymbol{\alpha}, x_D) = \sum_{i=0}^{4} \alpha_i \cdot x^i$	Dolní břeh	i = 1, ..., D
$Y_h = f_h(\boldsymbol{\beta}, x_H)$	Horní břeh	i = 1, ..., H
$Y_r = f_r(\boldsymbol{\gamma}, x_R)$	Proudnice	i = 1, ..., R

Proměnná	Význam	Indexy
$(\mathbf{x}^{\circ}, \boldsymbol{y}^{\circ}, \boldsymbol{z}^{\circ})$	Bod břehu řeky	$i = 1, \dots, N$
$(\mathbf{x}^{\times}, \boldsymbol{y}^{\times}, \boldsymbol{z}^{\times})$	Bod měření rychlosti toku	$j = 1, \dots, I$
$(\mathbf{x}^{\triangle}, \boldsymbol{y}^{\triangle}, \boldsymbol{z}^{\triangle})$	Projekce měřených bodů na proudnici	$j = 1, \dots, I$
$\boldsymbol{\alpha}$	Neznámé paramery ve funkci f_D	$k = 1, \dots, 5$
$\boldsymbol{\beta}$	Neznámé paramery ve funkci f_H	$k = 1, \dots, 5$
$\boldsymbol{\gamma}$	Neznámé paramery ve funkci f_R	$k = 1, \dots, 5$

1.2.1 Aproximace břehů polynomem

Postupně budeme zpracovávat změřené údaje. Budeme předpokládat, že říční břehy lze popsat polynomem čtvrtého stupně. To znamená, že dolní břeh lze popsat funkcí

$$P_d(x) = \sum_{i=0}^{4} \alpha_i \cdot x^i. \quad (1)$$

Obdobně můžeme horní břeh modelovat funkcí

$$P_h(x) = \sum_{i=0}^{4} \beta_i \cdot x^i. \quad (2)$$

Předpokládáme tedy, že řeka je orientována tak, že břehy lze popsat funkcí jedné nezávislé proměnné.

Nejprve metodou nejmenších čtverců pro funkci $P_d(x)$ odhadneme neznámé parametry $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4)^T$ respektive $\boldsymbol{\beta} = (\beta_0, \dots, \beta_4)^T$ pro funkci $P_h(x)$. K dispozici máme souřadnice několika změřených bodů na dolním břehu $D_j = (x_j, y_j), j = 1, \dots d$ a na horním břehu $H_j = (x_j, y_j), j = 1, \dots h$. Z těchto dat sestavíme matici plánu

$$\mathbb{X}_D = \begin{pmatrix} 1 & x_1^1 & x_1^2 & x_1^3 & x_1^4 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_d^1 & x_d^2 & x_d^3 & x_d^4 \end{pmatrix}. \quad (3)$$

Odhad neznámého vektorového parametru $\boldsymbol{\alpha}$ získáme ze známého maticového vztahu

$$\widehat{\boldsymbol{\alpha}} = (\mathbb{X}_D' \mathbb{X}_D)^{-1} \mathbb{X}_D' \mathbb{Y}_D, \quad (4)$$

kde $\mathbb{Y}_D = (y_{1,\dots,} y_d)^T$. Obdobně sestavíme matici $\mathbb{X}_H$ pro horní břeh a odhadneme $\widehat{\boldsymbol{\beta}}$.

Pro tyto odhady budeme potřebovat určit také jejich kovarianční matice

$$\mathrm{Var}(\widehat{\boldsymbol{\alpha}}) = (\mathbb{X}_D'\mathbb{X}_D)^{-1}. \tag{5}$$

1.2.2 Odhad říční čáry

Nyní se budeme snažit získat počáteční odhad funkce popisující říční čáru (proudnice s nejvyšší rychlostí).

Uvažujme, že jsme řeku rozdělili s využitím vysvětlující proměnné x na úseky $\langle a, a+h\rangle, \langle a+h, a+2h\rangle$ atd.

Obsah 1. úseku lze s využitím integrálního počtu vyjádřit pomocí určitého integrálu takto.

$$S = \int_a^{a+h} \sum_{i=0}^{4} (\hat{\beta}_i - \hat{\alpha}_i) x^i dx = \left[\sum (\hat{\beta}_i - \hat{\alpha}_i) \cdot \frac{x^{i+1}}{i+1} \right]_a^{a+h}. \tag{6}$$

Uvažujme, že bychom tento úsek nahradili obdélníkem o vodorovné straně $a = h$ a svislé straně b.

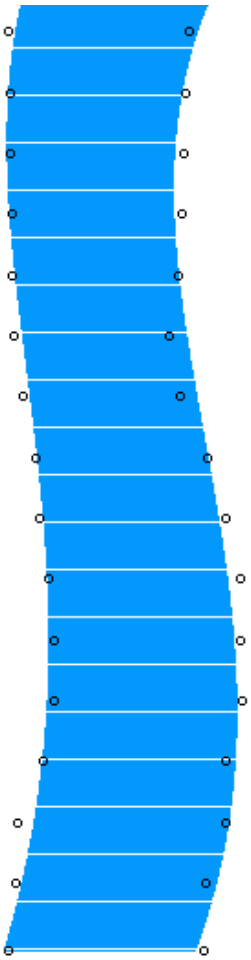
Pak můžeme z identity $S = H \cdot b$ snadno vyjádřit b $= \frac{S}{h}$. Středem našeho obdélníku je bod

$$R_1 = \left[a + \frac{h}{2}, P_d\left(a + \frac{h}{2}\right) + \frac{S}{h} \right]. \tag{7}$$

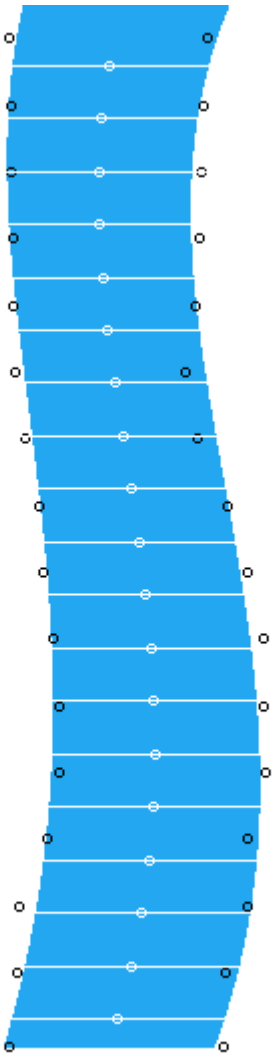
Tento bod budeme uvažovat, že se nachází na říční čáře. Obdobně získáme další body $R_2, R_3, \ldots$ Z těchto bodů opět pomocí metody nejmenších čtverců odhadneme průběh říční čáry

$$P_R = \sum_{i=0}^{4} \gamma_i x^i. \tag{8}$$

Říční koryto rozdělíme například na 20 úseků, proměnnou h určíme jako $h = \frac{x_{max} - x_{min}}{20}$.



Obrázek 2: Krájení na úseky



Obrázek 3: body na úsecích

Z těchto 20 bodů určíme

$$\hat{\gamma} = (\mathbb{X}_R'\mathbb{X}_R)^{-1}\mathbb{X}_R'\mathbb{Y}_R, \tag{9}$$

kde $\mathbb{Y}_R = (y_{1,\dots,}y_r)^T$.

Pro tento odhad budeme potřebovat určit také jeho kovarianční matici

$$\mathrm{Var}(\widehat{\gamma}) = (\mathbb{X}_R'\mathbb{X}_R)^{-1}. \tag{10}$$

1.2.3 Projekce známého bodu na proudnici

Velmi užitečná pro další výpočty bude znalost souřadnic nebližšího bodu na říční čáře (proudnici) od měřeného bodu. Tento nejbližší bod najdeme s využitím analytické geometrie a diferenciálního počtu. K říční čáře popsané polynomem (8) můžeme najít kromě tečny také normálu, která prochází měřeným bodem. Cílem je tedy najít projekci libovolného měřeného bodu $(x^{\times}; y^{\times})$ do bodu proudnice, který budeme označovat $(x^{\triangle}; y^{\triangle})$.

Určíme tečnu k proudnici, která je určena explicitní funkcí ve formě $y_p = f(x_p)$.

Směrnici této tečny a její normály určíme derivováním takto:

$$f'(x_p) = \frac{y^{\triangle} - f(x_p)}{x^{\triangle} - x_p}, \tag{11}$$

$$\frac{1}{f'(x_p)} = \frac{y^{\times} - y^{\triangle}}{x^{\triangle} - x^{\triangle}}. \tag{12}$$

Algebraickými úpravami těchto rovnic lze dospět ke vztahům

$$f'(x_p) \cdot x^{\triangle} - y^{\triangle} = f'(x_p) \cdot x_p - f(x_p)$$
$$x^{\triangle} + f'(x_p) \cdot y^{\triangle} = y^{\times} + f'(x_p) \cdot y^{\times}.$$

Soustavu můžeme přepsat do maticového tvaru $\mathbf{A_x} = \mathbf{b}$ v podobě

$$\begin{pmatrix} f'(x_p) & -1 \\ 1 & f'(x_p) \end{pmatrix} \begin{pmatrix} x^{\triangle} \\ y^{\triangle} \end{pmatrix} = \begin{pmatrix} f'(x_p) \cdot x_p - f(x_p) \\ x^{\times} + f'(x_p) \cdot y^{\times} \end{pmatrix},$$

kde $\mathbf{A} = \begin{pmatrix} f'(x_p) & -1 \\ 1 & f'(x_p) \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} f'(x_p) \cdot x_p - f(x_p) \\ x^{\times} + f'(x_p) \cdot y^{\times} \end{pmatrix}$.

Neznámé souřadnice projekce získáme řešením této soustavy dvou lineárních rovnic.

Řešení můžeme určit pomocí inverzní matici $\boldsymbol{A}^{-1}$, kterou určíme ve tvaru

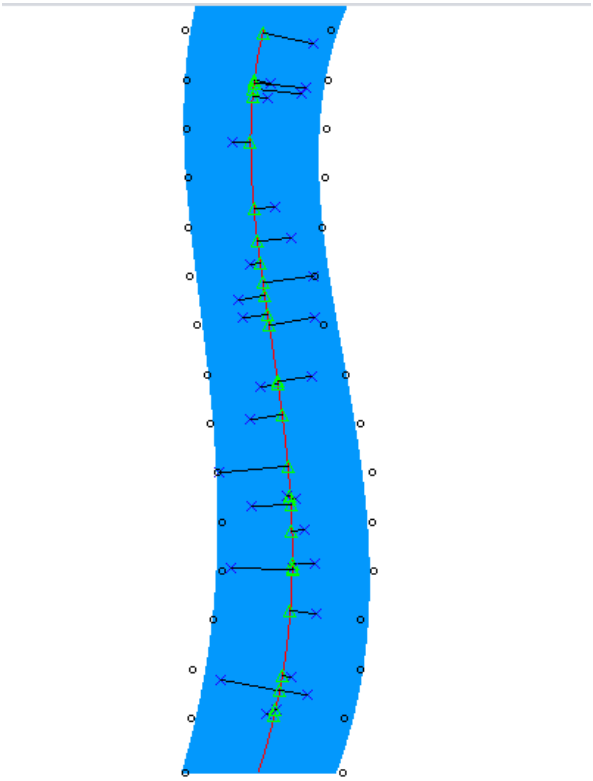
$$\mathbf{A^{-1}} = \frac{1}{\left(f'(x_p)\right)^2 + 1} \cdot \begin{pmatrix} f'(x_p) & 1 \\ -1 & f'(x_p) \end{pmatrix}.$$

Řešení soustavy ve tvaru $\mathbf{A^{-1}} \cdot \mathbf{b}$ je

$$\mathbf{A}^{-1} \cdot \mathbf{b} = \frac{1}{\left(f'(x_p)\right)^2 + 1} \begin{pmatrix} \left(f'(x_p)\right)^2 \cdot x_p - f'(x_p) \cdot f(x_p) + x^{\times} + f'(x_p) \cdot y^{\times} \\ -f'(x_p) \cdot x_p - f(x_p) + f'(x_p) \cdot x^{\times} + \left(f'(x_p)\right)^2 \cdot y^{\times} \end{pmatrix}.$$

Po úpravě lze souřadnice bodů $\begin{pmatrix} x^{\triangle} \\ y^{\triangle} \end{pmatrix}$ ortogonální projekce měřeného bodu na proudnici vyjádřit takto:

$$\begin{pmatrix} \mathrm{x}^{\triangle} \\ \mathrm{y}^{\triangle} \end{pmatrix} = \frac{f'(x_p)}{\left(f'(x_p)\right)^2 + 1} \begin{pmatrix} f'(x_p) \cdot x_p - f(x_p) + \dfrac{x^{\times}}{f'(x_p)} + y^{\times} \\ -x_p + \dfrac{f(x_p)}{f'(x_p)} + x^{\times} + f'(x_p) \cdot y^{\times} \end{pmatrix}. \qquad (13)$$



Obrázek 4: Projekce

1.3 Regresní model s podmínkou

Budeme uvažovat model měření viz zdroje [7] a [8]

$$\begin{pmatrix} \boldsymbol{Y}_1 \\ \boldsymbol{Y}_2 \end{pmatrix} \sim \left[\begin{pmatrix} \boldsymbol{X}_1 & 0 \\ 0 & \boldsymbol{X}_2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta} \\ \boldsymbol{\beta} \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma}_{1,1} & 0 \\ 0 & \boldsymbol{\Sigma}_{2,2} \end{pmatrix} \right], \tag{14}$$

kde složky vektorového parametru $\boldsymbol{\theta}$ definují průběh proudnice v prostoru takto

$$\begin{pmatrix} x \\ \beta_1(x) \\ \beta_2(x) \end{pmatrix} = \begin{pmatrix} x \\ \gamma_1 + \gamma_2 x + \gamma_3 x^2 + \gamma_4 x^3 + \gamma_5 x^4 \\ \delta_1 + \delta_2 x + \delta_3 x^2 + \delta_4 x^3 + \delta_5 x^4 \end{pmatrix} \tag{15}$$

a rychlost toku v libovolném bodě koryta je vázána s rychlostí toku na proudnici vztahem

$$V_i^{\times} = V_0 - (X_i^{\times} - X_i^{\triangle})^2\kappa - (Y_i^{\times} - Y_i^{\triangle})^2\kappa - (Z - Z_i^{\triangle})^2\kappa \tag{16}$$

V tomto modelu parametry $\boldsymbol{\theta}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ označují následující vektory skutečných hodnot

$$\boldsymbol{\theta} = \left[\gamma_0,\gamma_1,\gamma_2,\gamma_3,\gamma_4,\delta_0\delta_1\delta_2\delta_3\delta_4\right]^T, \tag{17}$$

$$\boldsymbol{\beta} = [X_1^{\times}, Y_1^{\times}, Z_1^{\times}, V_1^{\times}, \dots, X_n^{\times}, Y_n^{\times}, Z_n^{\times}, V_n^{\times}], \tag{18}$$

$$\boldsymbol{\gamma} = \left[X_1^{\triangle}, Y_1^{\triangle}, Z_1^{\triangle}, \dots, X_n^{\triangle}, Y_n^{\triangle}, X_n^{\triangle}, V_0, \kappa \quad\right]. \tag{19}$$

Skutečné vektorové parametry musí splnit tyto podmínky

$$g_1^i = V_i^{\times} - V_0 + \left(X_i^{\times} - X_i^{\triangle}\right)^2\kappa + \left(Y_i^{\times} - Y_i^{\triangle}\right)^2\kappa + \left(Z - Z_i^{\triangle}\right)^2\kappa = 0, \tag{20}$$

$$\begin{aligned} g_2^i = {} & X_i^{\times} - X_i^{\triangle}\left(Y_i^{\times} - Y_i^{\triangle}\right)\left[\gamma_1 + 2\gamma_2 X_i^{\triangle} + 3\gamma_3\left(X_i^{\triangle}\right)^2 + 4\gamma_4\left(X_i^{\triangle}\right)^3\right] \\ & + \left(Y_i^{\times} - Y_i^{\triangle}\right)\left[\delta_1 + 2\delta_2 X_i^{\triangle} + 3\delta_3\left(X_i^{\triangle}\right)^2 + 4\delta_4\left(X_i^{\triangle}\right)^3\right] = 0 \end{aligned} \tag{21}$$

$$g_3^i = \gamma_0 + \gamma_1 X_i^{\triangle} + \gamma_2 (X_i^{\triangle})^2 + \gamma_3 (X_i^{\triangle})^3 + \gamma_4 (X_i^{\triangle})^4 - Y_i^{\triangle} = 0, \tag{22}$$

$$g_4^i = \delta_0 + \delta_1 X_i^{\triangle} + \delta_2 (X_i^{\triangle})^2 + \delta_3 (X_i^{\triangle})^3 + \delta_4 (X_i^{\triangle})^4 - Z_i^{\triangle} = 0, \tag{23}$$

Tyto nelineární podmínky můžeme převést pomocí Taylorova rozvoje na systém lineárních podmínek ve tvaru

$$\boldsymbol{b} + \boldsymbol{B\beta} + \boldsymbol{C\theta} + \boldsymbol{G\gamma} = \boldsymbol{0}. \tag{24}$$

Derivováním nelineárních vazeb (20) až (23) získáme matice

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{C_1} \\ \boldsymbol{C_2} \\ \cdots \\ \boldsymbol{C_I} \end{pmatrix},$$

kde

$$\boldsymbol{C_i} = \left(\boldsymbol{C}_{i,1}, \boldsymbol{C}_{i,2}\right),$$

$$\boldsymbol{C'}_{i,1} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & (Y_i^{\times} - Y_i^{\triangle}) & X_i^{\triangle} & 0 \\ 0 & 2(Y_i^{\times} - Y_i^{\triangle})X_i^{\triangle} & (X_i^{\triangle})^2 & 0 \\ 0 & 3(Y_i^{\times} - Y_i^{\triangle})(X_i^{\triangle})^2 & (X_i^{\triangle})^3 & 0 \\ 0 & 4(Y_i^{\times} - Y_i^{\triangle})(X_i^{\triangle})^3 & (X_i^{\triangle})^4 & 0 \\ 0 & 5(Y_i^{\times} - Y_i^{\triangle})(X_i^{\triangle})^4 & (X_i^{\triangle})^5 & 0 \end{pmatrix},$$

a

$$\boldsymbol{C'}_{i,2} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & (Z_i^{\times} - Z_i^{\triangle}) & X_i^{\triangle} & 0 \\ 0 & 2(Z_i^{\times} - Z_i^{\triangle})X_i^{\triangle} & (X_i^{\triangle})^2 & 0 \\ 0 & 3(Z_i^{\times} - Z_i^{\triangle})(X_i^{\triangle})^2 & (X_i^{\triangle})^3 & 0 \\ 0 & 4(Z_i^{\times} - Z_i^{\triangle})(X_i^{\triangle})^3 & (X_i^{\triangle})^4 & 0 \\ 0 & 5(Z_i^{\times} - Z_i^{\triangle})(X_i^{\triangle})^4 & (X_i^{\triangle})^5 & 0 \end{pmatrix}.$$

Dále matice **B** má strukturu

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{B_1} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B_2} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ & & \cdots & & \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{B_I} \end{pmatrix},$$

kde

$$\boldsymbol{B_i} = \begin{pmatrix} 2(X_i^{\times} - X_i^{\triangle})\kappa & 2(Y_i^{\times} - Y_i^{\triangle})\kappa & 2(Z_i^{\times} - Z_i^{\triangle})\kappa & 1 \\ 1 & b_{i,1} & b_{i,2} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Přičemž $b_{i,1} = \gamma_1 + 2\gamma_2 X_i^{\triangle} + 3\gamma_3 (X_i^{\triangle})^2 + 4\gamma_4 (X_i^{\triangle})^3$ a $b_{i,2} = \delta_1 + 2\delta_2 X_i^{\triangle} + 3\delta_3 (X_i^{\triangle})^2 + 4\delta_4 (X_i^{\triangle})^3$.

Konečně matice **G** má v případě dvouetapového modelu rychlosti prouděné vody v 3D říčním korytě strukturu

$$\boldsymbol{G} = \begin{pmatrix} \boldsymbol{G_1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & L^{(1)} \\ \mathbf{0} & \boldsymbol{G_2} & \cdots & \mathbf{0} & \mathbf{0} & L^{(2)} \\ & & \cdots & & & \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{G_I} & L^{(I)} \end{pmatrix},$$

kde

$$\boldsymbol{G} = \left(\boldsymbol{G}_{i,1}, \boldsymbol{G}_{i,2}, \boldsymbol{G}_{i,3}\right),$$

přičemž

$$\boldsymbol{G}_{i,1} = \begin{pmatrix} -2(X_i^{\times} - X_i^{\triangle})\kappa_1 \\ g_i \\ \gamma_1 + 2\gamma_2 X_i^{\triangle} + 3\gamma_3 (X_i^{\triangle})^2 + 4\gamma_4 (X_i^{\triangle})^3 \\ \delta_1 + 2\delta_2 X_i^{\triangle} + 3\delta_3 (X_i^{\triangle})^2 + 4\delta_4 (X_i^{\triangle})^3 \end{pmatrix},$$

kde

$$g_i = -1 + \left(Y_i^{\times} - Y_i^{\triangle}\right)\left[2\gamma_2 + 6\gamma_3 X_i^{\triangle} + 12\gamma_4 \left(X_i^{\triangle}\right)^2\right] +$$
$$+\left(Z_i^{\times} - Z_i^{\triangle}\right)\left[2\delta_2 + 6\delta_3 X_i^{\triangle} + 12\delta_4 \left(X_i^{\triangle}\right)^2\right].$$

Dále

$$G_{i,2} = \begin{pmatrix} -2(Y_i^{\times} - Y_i^{\triangle})\kappa \\ -[\gamma_2 + 2\gamma_3 X_i^{\triangle} + 3\gamma_4 (X_i^{\triangle})^2 + 4\gamma_5 (X_i^{\triangle})^3] \\ -1 \\ 0 \end{pmatrix}$$

a konečně

$$G_{i,3} = \begin{pmatrix} -2(Z_i^{\times} - Z_i^{\triangle})\kappa \\ -[\delta_2 + 2\delta_3 X_i^{\triangle} + 3\delta_4 (X_i^{\triangle})^2 + 4\delta_5 (X_i^{\triangle})^3] \\ 0 \\ -1 \end{pmatrix}.$$

Poslední submatice $L^{(i)}$pak mají tvar

$$L^{(i)} = \begin{pmatrix} -1 & (X_i^{\times} - X_i^{\triangle})^2 + (Y_i^{\times} - Y_i^{\triangle})^2 & (Z_i^{\times} - Z_i^{\triangle})^2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Nejlepší nevychýlené odhady neznámých parametrů (BLUE) určíme v regresním modelu pomocí vztahů, viz zdroje [7] a [8]

$$\widehat{\boldsymbol{\theta}} = \left(\boldsymbol{X_1'\Sigma_{1,1}^{-1}X_1}\right)^{-1}\boldsymbol{X_1'\Sigma_{1,1}^{-1}Y_1}, \tag{25}$$

$$\begin{aligned}\widetilde{\widetilde{\boldsymbol{\beta}}} = \boldsymbol{I} - \ \boldsymbol{\Sigma_{22}}\ \boldsymbol{B'}\{\boldsymbol{U} - \boldsymbol{UG}\{\boldsymbol{G'UG}\}^{-1}\boldsymbol{G'U}\}\boldsymbol{B}\mathbb{Y} - \\ -\boldsymbol{\Sigma_{22}B'}\{\boldsymbol{U} - \boldsymbol{UG}\{\boldsymbol{G'UG}\}^{-1}\boldsymbol{G'U}\}(\boldsymbol{b} + \boldsymbol{C\theta}),\end{aligned} \tag{26}$$

$$\widetilde{\widetilde{\boldsymbol{\gamma}}} = \{\boldsymbol{G'UG}\}^{-1}\boldsymbol{G'U}[\boldsymbol{B}\mathbb{Y}_2 + \boldsymbol{b} + \boldsymbol{C\theta}]\ , \tag{27}$$

kde $\boldsymbol{U} = (\boldsymbol{B\Sigma_{22}B'} + \boldsymbol{GG'})^{-1}$.

kovarianční matice odhadu $\widetilde{\boldsymbol{\gamma}}$ je

$$\boldsymbol{Var}(\widetilde{\boldsymbol{\gamma}}) = (\boldsymbol{G'UG})^{-1} - \boldsymbol{I}\ , \tag{28}$$

Zdroje [7].

2 APLIKACE PRO VIZUALIZACI ŘÍČNÍHO TOKU

Jádrem této diplomové práce bylo vytvoření softwaru, který umožní provedení výpočtu polohy říční čáry, vypočtu rychlostí v bodech, kde rychlost nebyla změřena a vizualizaci těchto výpočtů pro zadaná měření konkrétního říčního koryta.

2.1 Použité technologie

Aplikace byla vyvinuta v programovacím jazyce Java. K vlastnostem tohoto programovacího jazyka patří automatická správa paměti, což znamená, že jsou automaticky odstraňovány nepotřebné objekty z operační paměti pomocí takzvaného Garbage Collectoru. Jazyk je vysokoúrovňový, tudíž dobře čitelný pro člověka a je objektově orientovaný.

Objektově orientovaný programovací jazyk pracuje s objekty, které obsahují atributy, což jsou data popisující stav objektu. Objekty dále disponují metodami, které určují chování objektu. Příkladem objektu v této práci může být třída Shores, která uchovává data o březích řeky a obsahuje metody například pro výpočet koeficientů polynomu jednoho z břehů. *Třída je množina objektů s určitými vlastnostmi. Přitom samotná třída nedefinuje nějaké konkrétní objekty té třídy, jen udává, jaké vlastnosti bude mít každý objekt té třídy (podobně představa o tom co je telefon může existovat i kdyby všechny existující telefony zanikly).*

Programovací Jazyk Java má výhodu v tom, že je přenositelný, to znamená, že je možné ho spustit na kterékoliv platformě, na které je nainstalovaný Java Virtual Mchine, ten poté interpretuje byte code.

Pro vývoj softwaru bylo zvolené vývojové prostředí Netbeans obsahující vše potřebné pro vývoj této aplikace, zejména editor pro grafické rozhraní programu.

Zdroje [5], [6]

2.2 Použitá knihovna třetích stran

Pro vývoj aplikace byla použita knihovna třetích stran obsahující komponenty pro řešení matematických problémů. Vlastnosti této knihovny budou popsány v této podkapitole.

2.2.1 Knihovna Commons Math

Commons Math je matematická knihovna z Apache Commons. Obsahuje matematické a statistické komponenty pro řešení problémů, jako je řešení diferenciálních funkcí, řešení soustav lineárních rovnic, proložení křivek, strojové učení a další. Tato knihovna byla využita v této práci zejména pro výpočet koeficientů polynomu představujícího křivku břehů či říční čáry, dále byla knihovna využita pro maticové výpočty využívané pro odhad neznámých parametrů.

2.3 Struktura aplikace

V této kapitole budou popsány třídy aplikace a stěžejní standardní knihovny jazyka Java pro tuto aplikaci.

2.3.1 Třída RiverCurve

Třída *RiverCurve* je využita k výpočtu koeficientů polynomu břehů a říční čáry. Třída přebírá do konstruktoru souřadnice uložené ve třídě *WeightedObservedPoints*, tyto body jsou dále využívány v metodě *calculateCurveCoefs*, která vypočítá koeficienty polynomu a vrátí je jako pole.

Pro zjištění začátku a konce křivky jsou zde metody getStartPoint a getEndPoint, dále je zde obsažena metoda pro výpočet projekcí měřených bodu na říční čáru, jenž jsou nezbytné pro další výpočty. Ukázka kódu této třídy je k nalezení v příloze A.

2.3.2 Třída River

Třída River slouží k uchování bodů naměřených na břehu řeky a k provádění operací nad těmito body. Tato třída inicializuje tři instance typu *RiverCurve*, dvě instance obsahují polynom pro horní a dolní břeh a třetí instance obsahuje odhadovanou říční čáru.

Pro odhad říční čáry je zde využita funkce *estimateStreamLine*, která využívá výpočty z kapitoly 1.2.2 a vrací instanci třídy *RiverCurve*.

Pro zjištění, zdali se vykreslovaný pixel nachází mezi křivkami aproximujícími břehy řeky, je zde funkce *isBetweenShores*, která přijímá parametry zkoumané souřadnice a vrací booleovskou hodnotu podle toho, kde se souřadnice nachází.

2.3.3 Třída MatrixCBG

Kapitola 1.3 pojednává o výpočtu neznámých parametrů, pro jejichž odhad jsou nezbytné matice $\boldsymbol{C}$ $\boldsymbol{B}$ a $\boldsymbol{G}$, jejichž výpočet má na starost právě třída MatrixCBG. Třída MatrixCBG přebírá do konstruktoru měřené body a třídu *River*, která obsahuje koeficienty polynomu využité pro další výpočty. Konstruktor této třídy volá metody pro výpočet jednotlivých matic, které jsou poté k dispozici pomocí jednotlivých *Getterů*.

Třída *RealMatrix* z výše zmiňované knihovny třetích stran *Common Math* je zde využita pro návrat jednotlivých matic a usnadňuje další práci s maticemi ve třídě zmíněné v následující podkapitole.

2.3.4 Třída EstimatedParameters

Výpočet neznámých parametrů $\tilde{\tilde{\boldsymbol{\beta}}}$ a $\tilde{\tilde{\boldsymbol{\gamma}}}$ je realizován ve třídě *EstimatedParameters,* tato třída přebírá do konstruktoru potřebná měření a také instanci výše popsané třídy *MatrixCBG,* která poskytuje matice pro výpočty neznámých parametrů.

Je zde využita již zmíněná třída *RealMatrix*, jejíž metody výrazně usnadňují maticové operace a jejíž instance s maticemi jsou získávány právě z instance třídy *MatrixCBG*.

Třída stejně jako třída *MatrixCBG* poskytuje výsledné výpočty pomoci *Getterů*.

Výsledné parametry jsou vektory a jejich typem je instance tříd *RealVector* z knihovny *Common Math*.

2.3.5 Třída RiverGraphics

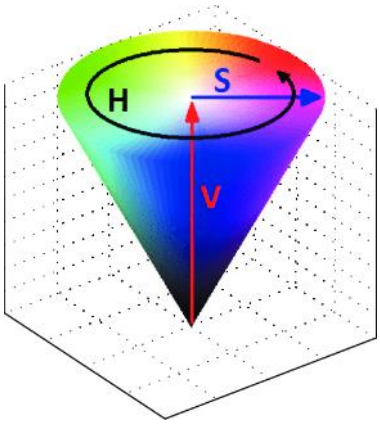
Tato třída je nejvíce komplexní a slouží k vykreslování modelu řeky. Obsahuje instance tříd River, Matrix CBG a EstimatedParameters pro získání potřebných hodnot pro výpočet jednotlivých rychlostí ve všech souřadnicích na hladině řeky.

Vykreslování je realizováno pomocí procházení jednotlivých řádků pixelů instance třídy BufferedImage, jenž reprezentuje pixelovou mřížku plátna. Samotnou řeku je možno vykreslit obarvením pixelů, které se nacházejí mezi polynomy břehů, což umožňuje zjistit atribut typu River. Pokud se aktuální zkoumaná souřadnice nachází mezi, tak je pomocí vzorce (18) vypočtena hodnota rychlosti v metrech. Tuto hodnotu je poté nutno převést na hodnotu barvy, kterou bude aktuální pixel obarven, pro mapování intervalu rychlostí byl použit barevný model

HSV. Metoda *paintComponent*, která se stará o toto vykreslování, je v práci k prohlédnutí v příloze B.

Na rozdíl od barevného modelu RGB (Red Green Blue), který využívá k reprezentaci konkrétní barvy míchání červené, zelené a modré barvy, využívá model HSV (Hue Saturation Value) parametry pro odstín, sytost a jas barvy.

Výhodou tohoto modelu vzhledem pro potřeby této práce je možnost vyjádření odstínu barvy pomocí úhlu ve stupních. Lze v něm lépe namapovat odstín barvy na rychlostní interval protékající vody a získat tak odstín barvy pro jednotlivé pixely podle rychlosti průtoku na konkrétní souřadnici. Výpočet barvy z hodnoty rychlosti je realizován metodou *getColorOfPixel*.



Obrázek 5: Barevný model HSV [2]

Inicializace všech potřebných atributu je realizována metodou *load*, která přebírá jako parametry soubory s naměřenými daty, se souřadnicemi břehů a kok pro výpočet odhadu říční čáry. Pro tuto metodu jsou nezbytné metody *loadData* a *parseData*. Metoda *loadData* načte ze souboru všechny řádky a vrátí je jako *ArrayList*, tento *ArrayList* pak využije metoda *parseData*, která pak přiřadí hodnoty do jejích vstupně výstupních parametrů.

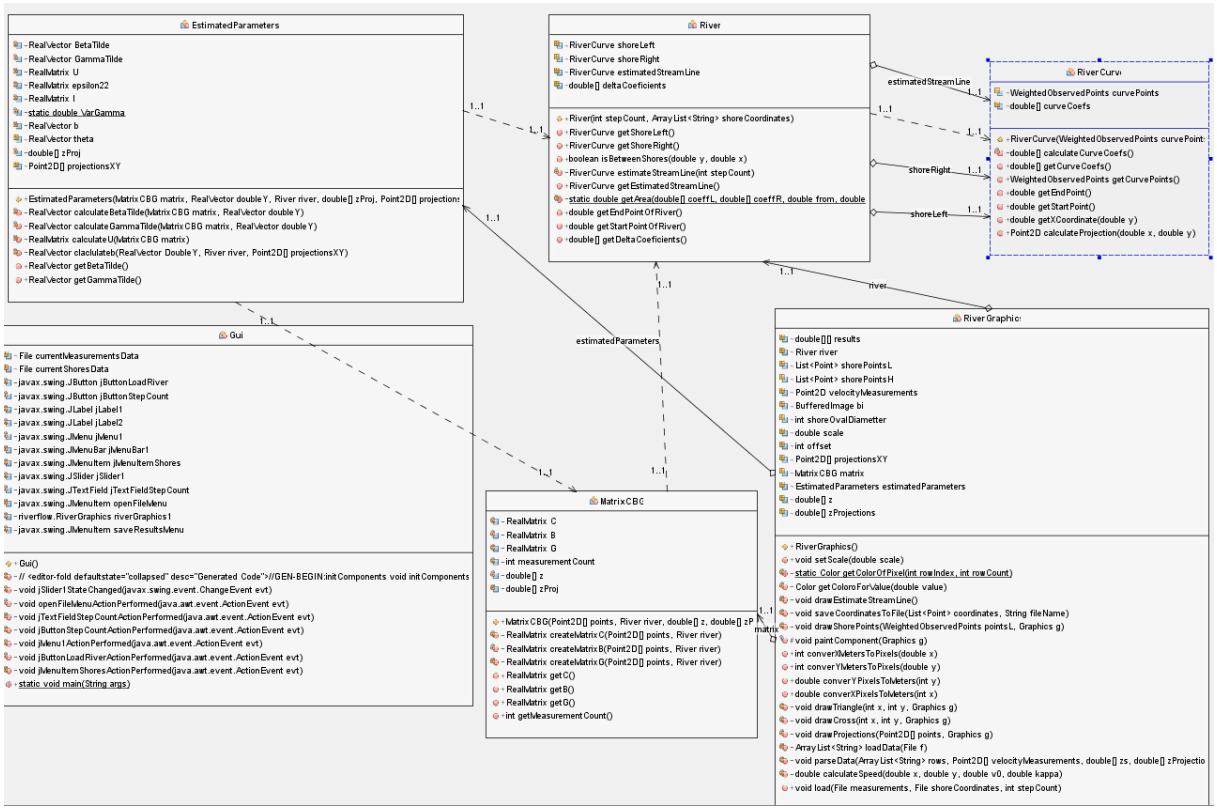
Třída během vývoje aplikace obsahovala pomocné a testovacích třídy, jako je například třída pro uložení souřadnic břehů v pixelech do souboru.

2.3.6 Třída Gui

Tato třída představuje třídu obsahující komponenty grafického rozhraní aplikace, jako jsou tlačítka či posuvníky pro změnu měřítka vykreslovaného modelu řeky. Obsahuje také instanci třídy *RiverFlow* pro zobrazení výsledného modelu řeky s barevnou vizualizací rychlostí.

Jsou zde uloženy i načtené soubory, které jsou předávány metodě *load* ze třídy *RiverGraphics*. Před předáním parametrů je zajištěna kontrola souborů.

2.4 Grafické znázornění



Obrázek 6: UML Diagram tříd

3 NUMERICKÁ STUDIE

V této části budeme používat symboliku uvedenou v Tab. 1 na str. 13.

Mějme k dispozici měření bodů na dolním břehu uvedené v Tabulce 2. Údaje jsou v metrech.

Tabulka 2: Souřadnice bodů dolního břehu

X	0,5040	1,5040	2,5040	3,5040	4,5040	5,5040	6,5040	7,5040
Y	0,0860	0,1020	0,1190	0,1390	0,1470	0,1720	0,3020	0,5300

8,5040	9,5040	10,5040	11,5040	12,5040	13,5040	14,5040	15,6040
0,5980	0,7320	0,8320	0,8320	0,6320	0,2320	0,2020	0,0820

Tabulka 3: Koeficienty dolního břehu

Koeficienty dolního břehu				
0,2657	-0,2324	0,0661	-0,0047	0,0001

Pro data z Tab. 2 jsme vypočetli odhad $\widehat{\boldsymbol{\beta}}$ pomocí vzorce (4). Tento odhad, viz Tab. 3, určil polynom vykreslený na Obr. 6. Při vykreslení jsme pracovali se standardním nastavením souřadnicového systému v programovacím jazyku Java. Počátek použité souřadnicové soustavy (0,0) je v levém horním rohu. Proměnná *x* je znázorněna na vodorovné ose. Orientace svislé osy je směrem dolů a znázorňuje proměnnou *y*. Měřítka na obou osách jsou shodná. Aplikace načítá údaje o souřadnicích v metrech a neuvažuje se převod souřadnic na jiné jednotky, vypočtené souřadnice jsou samozřejmě také v metrech. Při grafické reprezentaci aplikace provede transformaci z SI na pixely pomocí vztahů: $x = \left(\frac{x}{scale}\right) + \text{offset}$ a $y = \left(\frac{y}{scale}\right) + \text{offset}$, které jsou naprogramované v proceduře *converYMetersToPixels* a v proceduře *converXMetersToPixels*.

Obdobně pro horní břeh máme měření uvedená v tabulce 4.

Tabulka 4: Body horního břehu

X	0,5040	1,5040	2,5040	3,5040	4,5040	5,5040	6,5040	7,5040
Y	3,0380	2,9720	2,9650	2,9390	2,8720	2,7120	2,9020	3,3320

15,6040	8,5040	9,5040	10,5040	11,5040	12,5040	13,5040	14,5040
3,2820	3,6520	3,8920	3,8720	3,9020	3,6320	3,6320	3,3060

Metodou nejmenších čtverců dostáváme odhad $\widehat{\boldsymbol{\beta}}$ neznámých parametrů horního břehu $\boldsymbol{\beta}$.

Tabulka 5: Koeficienty horního břehu

Koeficienty horního břehu				
3,3555	-0,4226	0,0922	-0,0053	0,0001

Nyní máme popsané oba břehy odhadnutými polynomy P_d a P_h.

Dalším naším cílem je popsat říční čáru, kterou odhadneme pomocí postupu uvedeného v sekci 1.2.2.

V aplikaci byl pro zvolený počet úseků k = 19 dělících řeku vypočten krok $h = 0{,}795\ m$. Určená plocha prvního úseku $S_1 = 4{,}371\ m^2$.

Vypočtené těžiště prvního úseku je R_1=[1,259; 1.5161]. Iteračně procedura *getArea* určila středy úseků uvedené v dalších sloupcích Tab. 6.

Tabulka 6: Těžiště 19 úseků vodního toku

X	1,259	2,769	4,279	5,789	7,299	8,809	10,32	11,83
Y	1,516	1,411	1,493	1,683	1,909	2,11	2,235	2,241

13,34	14,85	1,259	2,769	4,279	5,789	7,299	8,809	10,32
2,098	1,782	1,516	1,411	1,493	1,683	1,909	2,11	2,235

11,83	13,34	14,85
2,241	2,098	1,782

Tyto body jsme také aproximovali polynomem 4. řádu dle vzorců (10) až (13) a dostali řešení $\boldsymbol{\beta}$.

Tabulka 7: Odhady koeficienty polynomu říční čáry

Koeficienty polynomu říční čáry				
1,8131	-0,3277	0,0791	-0,0050	0,0001

Odhadnutý průběh říční čáry pro odhad z Tab. 6 je znázorněn červeně na Obr. 6. Toto řešení ale budeme ještě vylepšovat pomocí postupu uvedeného v sekci 1.3. V navrženém matematickém modelu náš počáteční odhad zpřesníme, když využijeme měření rychlosti toku ve změřených bodech.

Uvažujme následující úlohu: v korytě vodního toku byly změřeny souřadnice a rychlosti ve 30 bodech. Výsledky měření jsou uvedeny v Tab. 8.

Tabulka 8: Výsledky měření rychlostí

$\mathbf{X}^{\times}$	$\mathbf{Y}^{\times}$	$\mathbf{Z}^{\times}$	$\mathbf{X}^{\Delta}$	$\mathbf{Y}^{\Delta}$	$\mathbf{Z}^{\Delta}$	$\mathbf{V_0}$
10,6771	2,5011	251,4929	10,6843	2,2488	249,7368	0,3705
4,7112	2,232	250,1244	4,7912	1,5482	249,916	0,8965
4,0983	1,9114	251,2512	4,1373	1,4785	249,9358	0,6162
6,3575	2,7101	250,2847	6,4968	1,7877	249,8643	0,7906
13,6564	2,2336	250,7827	13,6249	2,0527	249,6494	0,7364

11,3482	2,7237	250,9833	11,3429	2,2547	249,7169	0,6352
11,4493	2,2694	250,2393	11,449	2,2531	249,7137	0,9447
2,7676	1,0444	250,591	2,7669	1,4092	249,977	0,898
1,8661	1,7585	251,3495	1,8404	1,4461	250,0048	0,6187
13,7168	0,8026	250,0004	13,9307	1,9983	249,6405	0,679
10,032	2,3331	251,5913	10,0391	2,2196	249,7563	0,324
5,9812	1,1495	250,699	5,9	1,6984	249,8824	0,8051
7,7552	1,6036	250,0499	7,7039	1,9674	249,8275	0,9631
7,542	2,6487	250,6545	7,6418	1,9586	249,8294	0,7666
1,5662	1,8026	251,3596	1,5288	1,4779	250,0141	0,6166
14,4174	1,7146	251,1003	14,4568	1,8827	249,6254	0,559
0,7721	2,6741	250,0926	0,5528	1,6502	250,0432	0,7802
9,4925	0,7772	250,1128	9,3715	2,1682	249,7766	0,5875
9,9725	2,1415	250,6571	9,9677	2,2151	249,7585	0,8374
12,3661	2,7549	250,713	12,3234	2,2128	249,6876	0,7306
10,1914	1,4181	250,0917	10,1475	2,2261	249,7531	0,8461
1,7926	2,4532	250,1336	1,7027	1,4585	250,0089	0,7974
1,6756	2,5364	250,003	1,5659	1,473	250,013	0,7714
14,0397	2,5713	250,2787	13,9228	1,9986	249,6407	0,8503
5,4946	2,6734	250,0071	5,6363	1,66	249,8904	0,7879
8,4082	1,4077	250,0847	8,3272	2,0519	249,8085	0,9004
6,3492	1,2318	250,9216	6,2704	1,7535	249,8712	0,7237
5,276	1,3909	251,0999	5,247	1,6063	249,9022	0,7037
11,4529	1,0026	250,001	11,4752	2,2527	249,713	0,6708
14,3334	1,9317	251,6027	14,329	1,9123	249,629	0,2208

Tyto vstupní hodnoty nám již umožňují určit všechny matice potřebné ve vztahu $\tilde{\tilde{\boldsymbol{\gamma}}}$.

Získali jsme matici B rozměru 120 × 120, matici C rozměru 120 × 10 a matici G rozměru 120 × 92, vzhledem k velikosti těchto matic jsou uvedeny pouze v příloze na CD.

Dále jsme podle vzorce (26) odhadli $\tilde{\tilde{\boldsymbol{\beta}}}$ a podle vzorce (27) odhadli $\tilde{\tilde{\boldsymbol{\gamma}}}$.

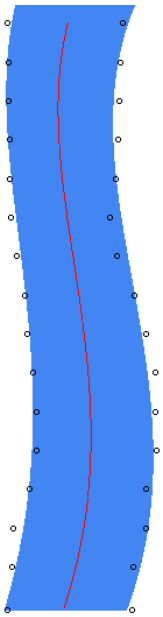
Z odhadnutého vektoru $\tilde{\tilde{\boldsymbol{\gamma}}}$ je pro nás nejzajímavější předposlední a poslední prvek, které obsahují odhad rychlosti v a koeficientu κ:

$$\hat{v} = 1,$$
$$\hat{\kappa} = 0{,}234.$$

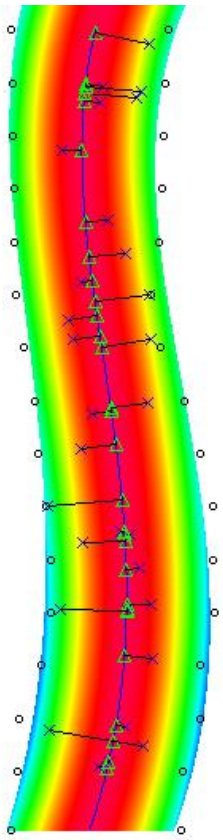
Pro odhad $\tilde{\tilde{\boldsymbol{\gamma}}}$ jsme určili jeho kovarianční matici $\mathrm{var}(\tilde{\tilde{\gamma}})$, rozměru 92 × 92 podle vzorce (28). Přesnost odhadnuté rychlosti $\hat{v}$ a koeficientu $\hat{\kappa}$ udává pravý dolní roh této matice

$$\mathrm{var}(\hat{v}, \hat{\kappa}) = \begin{pmatrix} 0{,}00007329 & 0{,}00002162 \\ 0{,}00002162 & 0{,}00001825 \end{pmatrix}.$$

Odmocněním diagonály této matice získáme směrodatné odchylky $s_{\hat{v}} = 0{,}0086$ a $s_{\hat{\kappa}} = 0{,}0043$.



Obrázek 7: Odhad říční čáry



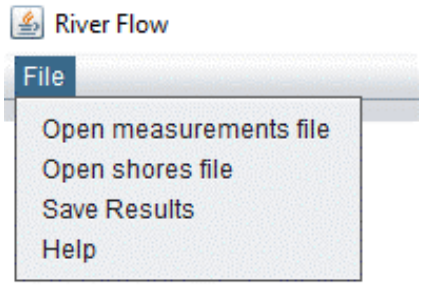
Obrázek 8: Grafické rozložení rychlostí v korytě vodního toku

4 MANUÁL PRO VYTVOŘENOU APLIKACI

Tato kapitola se bude věnovat stručnému popisu ovládání aplikace a možnostem jejího nastavení.

Pro spuštění aplikace je nutno mít na systému nainstalovanou Javu. Aplikaci lze spustit pomocí souboru *RirverFlow.jar*.

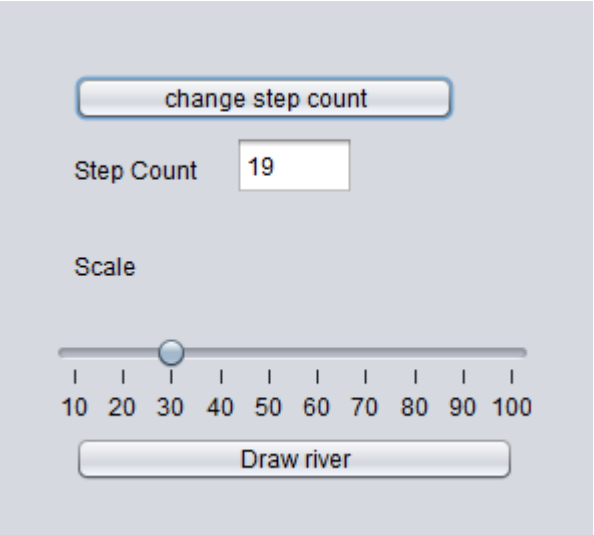
Před vykreslením řeky je nutno načíst soubory s naměřenými hodnotami rychlostí a s naměřenými souřadnicemi břehu. Načtení souborů je realizováno pomocí menu *File* v levém horním rohu programu. Po rozbalení tohoto menu jsou zobrazeny tři možnosti, pro načtení souboru s měřenými rychlostmi, vytvoření výstupu do souboru a pro načtení souřadnic břehů. Menu je znázorněno na obr. 4.



Obrázek 9: File menu

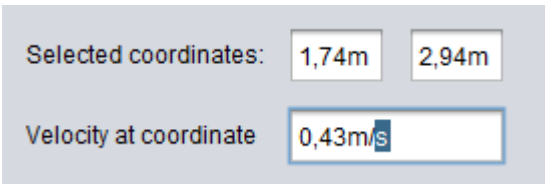
Po vybrání jedné z *Open* možností je otevřeno okno s prohlížečem souborů, kde je možno požadovaný soubor vybrat a načíst. Po otevření obou potřebných souborů už stačí použít tlačítko *Draw river*, pokud soubory obsahují nezbytná data, je nyní vykreslena řeka.

Výpočtu říční čáry z kapitoly 1.2.2 je možno nastavit krok pro výpočet pomocí hodnoty v *text Filedu Step Count,* a potvrzení změny pomocí tlačítka *change step count*. Změny při defaultním nastavení škálování nemusí být patrné, proto je možno řeku přiblížit změnou škálování o což se stará slider *Scale*. Ovládací prvky pro změnu škálování, kroku pro výpočet říční čáry a vykreslení řeky jsou znázorněny na obr. 5.



Obrázek 10: Ovládací prvky programu

Odečítání rychlostí v konkrétní souřadnici je realizováno pomocí výběru souřadnice kurzorem, kurzor musí být umístěn na říční ploše a poté se odečtení provede levým tlačítkem myši, výsledná rychlost je poté zobrazena v textovém poli *Velocity at coordinate*. Vybrané souřadnice jsou pak zobrazeny v metrech v text boxech *Selected coordinates*. Text boxy s naměřenou rychlostí v souřadnicích jsou zobrazeny na obr. 10.



Obrázek 11: Text boxy se souřadnicemi a naměřenou rychlostí

Odečtené rychlosti se souřadnicemi je možno uložit do csv souboru. Uživatel aplikace nejdříve vybere na říční ploše kurzorem libovolný počet měření a poté vybere z *File* menu možnost *Save Results*, v dialogovém okně vybere umístění souboru, zadá jméno souboru a pomocí tlačítka *Save* uloží. Pro vytvoření nové sady měření je možno předešlá měření odstranit tlačítkem *Clear coordinates to save*.

ZÁVĚR

Cílem práce bylo nastudování matematických nástrojů pro modelování laminárního proudění a vytvoření aplikace umožňující grafické znázornění vodního toku.

Nejprve byl čtenář seznámen s pojmem laminární proudění a poté s genezí postupné tvorby matematického modelu. Poté byly uvedeny vztahy pro výpočet odhadů říční čáry a rychlostí vody v souřadnicích říčního koryta.

Pro vytvoření aplikace byl využit objektově orientovaný programovací jazyk JAVA a jako vývojové prostředí byly použity NetBeans. K některým matematickým výpočtům byla použita knihovna třetích stran, zejména k výpočtu parametrů polynomu a maticovým operacím.

Pro potřeby práce byl využit nelineární statistický model s podmínkou typu II a neznámé parametry byly odhadnuty metodou nejmenších čtverců. Tento model je možno do budoucna obohatit o další parametry, které napomohou k dalšímu přiblížení modelu realitě. I když výsledky modelu nejsou zcela přesné, velice se přibližují výsledkům, které byly naměřeny. Akceptovatelnost modelu by mohla být posouzena při aplikaci na data naměřená u různých vodních toků.

Softwarová aplikace, která byla vyvinuta v rámci této práce pomohla k lepšímu znázornění výsledků a uživatel tak může sledovat chování tekoucí vody v různých korytech, i přesto, že byl proveden pouze omezený počet měření.

Tato aplikace dokáže pracovat se soubory obsahujícími různý počet naměřených hodnot rychlostí v libovolných bodech vodního toku.

Díky této práci jsem si mohl rozšířit poznatky z oblasti matematické statistiky, fyzikálního modelu proudění kapalin a poznatky z oblasti počítačové grafiky.

Práci by bylo možno obohatit o vizualizaci průřezů vodního koryta, což by bylo možno využít například jako pomocný nástroj pro výpočet tlaku vody při projektování přehrad. Další vylepšení mohou směřovat k rozšíření možností vizualizace vodního toku.

POUŽITÁ LITERATURA

[1] KRÁLOVÁ, Magda. PROUDĚNÍ TEKUTIN. In: *Eduportál | Eduportál Techmania* [online]. © Techmania Science Center [cit. 2019-08-21]. Dostupné z: https://edu.techmania.cz/cs/encyklopedie/fyzika/tekutiny/proudeni-tekutin

[2] KAU, Lih-Jen a Tien-Lin LEE. The commonly used HSV color model. In: KAU, Lih-Jen a Tien-Lin LEE: An Efficient and Self-Adapted Approach to the Sharpening of Color Images. *Scientific World Journal* [online]. 2013, vol. 2013 [cit. 2019-08-21]. ISSN 1537-744X. Dostupné z: https://www.hindawi.com/journals/tswj/2013/105945/

[3] Pitotova trubice. In: *FYZMATIK* [online]. ©fyzmatik, 15. června 2009 [cit. 2019-08-21]. Dostupné z: http://fyzmatik.pise.cz/930-pitotova-trubice.html

[4] MASSEY, Bernard a John WARD-SMITH. *Mechanics of Fluids*. University College London: CRC Press, 1998. ISBN 0748740430.

[5] Základní principy a pojmy objektově orientovaného programování. *Pehapko* [online]. [cit. 2019-08-21]. Dostupné z: http://pehapko.cz/oop/uvod

[6] BOSÁK, Tomáš a Martin ŽÁK. Úvod do programovacieho jazyka Java. In: *Programujte* [online]. © 2003–2019 Programujte.com, 24. dubna 2006 [cit. 2019-08-21]. Dostupné z: http://programujte.com/clanek/2006041804-uvod-do-programovacieho-jazyka-java/

[7] TUČEK Pavel a Jaroslav MAREK. *The Velocity of the river flow*. Tatra Mountains Mathematical Publications. 2006, no. 14, s. 241-248.

[8] KUBÁČEK, Lubomír. *Statistical theory of geodetic networks*. Vydání první. Zdiby: Výzkumný ústav geodetický, topografický a kartografický, Odvětvové informační středisko, 2013. ISBN 80-858-8131-4.

PŘÍLOHY

PŘÍLOHA A – TŘÍDA RIVERCURVE

```
public class RiverCurve {

    final WeightedObservedPoints curvePoints;// body brehu pro vypocet
polynomu
    final double[] curveCoefs;// pole koeficientu do polynomu

    public RiverCurve(WeightedObservedPoints curvePoints) {
        this.curvePoints = curvePoints;
        curveCoefs = calculateCurveCoefs();
    }

    //spocita koeficienty polynomu
    private double[] calculateCurveCoefs() {
        final  PolynomialCurveFitter  fitter  =  PolynomialCurveFit-
ter.create(4);
        return fitter.fit(curvePoints.toList());
    }

    public double[] getCurveCoefs() {
        return curveCoefs;
    }

    public WeightedObservedPoints getCurvePoints() {
        return curvePoints;
    }

    public double getEndPoint() {
        return  (curvePoints.toList().get(curvePoints.toList().size()
- 1).getX());
    }

    public double getStartPoint() {
```

```
        return (curvePoints.toList().get(0).getX());
    }

    public double getXCoordinate(double y) {
        //vraci xovou souradnici krivky v metrech
        double shoreLPoint = y * y * y * y * curveCoefs[4] + y * y *
y
                * curveCoefs[3] + y * y * curveCoefs[2] + y * curve-
Coefs[1] + curveCoefs[0];
        return shoreLPoint;
    }

    public Point2D calculateProjection(double x, double y) {
        double lineCoef[] = curveCoefs;
        double fXp = x * x * x * x * lineCoef[4] + x * x * x *
lineCoef[3]
                + x * x * lineCoef[2] + x * lineCoef[1] + lineCoef[0];
        double fXpDer = lineCoef[1] + 2 * lineCoef[2] * x + 3 *
lineCoef[3] * x * x
                + 4 * lineCoef[4] * x * x * x;
        double coef = fXpDer / (fXpDer * fXpDer + 1);

        double xDelta = coef * ((fXpDer * x) - fXp + (x / fXpDer) +
y);
        double yDelta = coef * (-x + (fXp / fXpDer) + x + fXpDer * y);

        return new Point2D.Double(xDelta, yDelta);
    }
}
```

PŘÍLOHA B – METODA PAINTCOMPONENT

```
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        if (z == null) {
            return;
        }

        for (int y = 0; y < bi.getHeight(); y++) {// prochazeni radku pixelu
            double yMeters = converYPixelsToMeters(y);

            for (int x = 0; x < bi.getWidth(); x++) {// prochazeni sloupcu pixelu
                double xMeters = converXPixelsToMeters(x);

                if (river.isBetweenShores(yMeters, xMeters) && yMeters <= river.getEndPointOfRiver()) {
                    try {
                        RealVector vect = estimatedParameters.getGammaTilde();
                        double velocity = calculateSpeed(yMeters, xMeters, vect.getEntry(vect.getDimension() - 3), vect.getEntry(vect.getDimension() - 2));
                        velocityMap.put(new Point2D.Double(x, y), velocity);
```

```
                        bi.setRGB(x,        y,        getColoroForVa-
lue(velocity).getRGB());// kdyz je mezi brehy vybarvi pixel


                        double  V0  =  estimatedParameters.getGamma-
Tilde().getEntry(estimatedParameters.getGammaTilde().getDimension()
- 1);


                    } catch (Exception e) {
                        System.out.println("" + e.toString());
                    }
                } else {
                    bi.setRGB(x, y, 0xffffff);// kdyz neni mezi brehy
                }
            }
        }
        drawEstimateStreamLine();
        g.drawImage(bi, 0, 0, null);// nejdriv vykresli vykresli reku
a pak teprva body aby se body rekou neprekreslily a nebyly videt


        drawProjections(velocityMeasurements, g);
        drawShorePoints(river.shoreLeft.getCurvePoints(), g);
        drawShorePoints(river.shoreRight.getCurvePoints(), g);
    }
```