

Univerzita Pavla Jozefa Šafárika v Košiciach
Prírodovedecká fakulta

ALGORITMIZÁCIA A IMPLEMENTÁCIA SPLAJNOV NA BÁZE EXPLICITNÝCH VZORCOV

BAKALÁRSKA PRÁCA

Študijný odbor:	Informatika
Školiace pracovisko:	Ústav informatiky
Vedúci záverečnej práce:	doc. RNDr. Csaba Török, CSc..
Konzultant:	Mgr. Lukáš Miňo

Košice 2014

Viliam Kačala

Podakovanie

Rád by som poďakoval vedúcemu bakalárskej práce doc. RNDr. Csabovi Törökovi, CSc. za cenné pripomienky, odborné vedenie a obetavosť počas tvorby práce.

Taktiež by som sa rád poďakoval Mgr. Lukášovi Miňovi za cenné rady a odbornú pomoc počas tvorby aplikačnej časti tejto práce.

Zadanie práce

Abstrakt

Splajny sú dôležitá súčasť počítačovej grafiky. Jedná sa o matematický model krivky, ktorá slúži na čo „najlepšie spojenie“ konečnej množiny bodov. Termín „najlepšie spojenie“ v našom prípade znamená hladká, matematicky ľahko vyjadriteľná krivka s čo najmenším zakrivením. Využitie splajnov v grafike je veľmi široké od rôznych CAD aplikácií, v štatistike, alebo v analýze dát. Splajny existujú v mnohých formách, či už vo forme krivky v rovine, rôznych trojrozmerných telies, atď.. Táto práca si dáva za cieľ analyzovať, implementovať a porovnať rôzne postupy pri počítaní a generovaní splajnov v dvojrozmernom priestore, teda rovine.

Abstract

Splines are important part of computer graphics. It is a mathematical model of the curve, which is for the "best connection" of any finite set of points. The term "best connection" in this case means smooth, easily calculable mathematical curve with minimal curvature. Use of splines in graphics varies from large variety of CAD applications, statistics or data analysing. Splines exist in many forms, whether in the form of curves in the plane, a variety of three-dimensional bodies, etc. .. This work aims to analyze, implement and compare different procedures for counting and generating splines in two-dimensional space, ie in the plane.

Obsah

I	Návrh riešenia	8
1	Splajn krivky	9
1.0.1	Ilustračný príklad	9
1.0.2	Formálna definícia splajnov	10
1.0.3	Delenie splajnov	11
1.1	B-splajny (Bázové splajny)	12
1.1.1	Bell funkcie	12
1.1.2	B-splajn	15
1.2	Globálny B-splajn	19
1.2.1	Pomocné uzly	20
1.2.2	Generovanie kontrolných bodov B-splajnu	21
1.3	Hermitove splajny	25
1.3.1	Bázové funkcie	25
1.3.2	Hermitova krivka	27
1.4	Clamped splajny	30
2	Nový postup generovania clamped splajnov	32
2.1	Zostrojenie	32
II	Implementácia	35
3	Microsoft Silverlight	37
3.1	Architektúra	38
3.1.1	XAML	38
3.2	Grafika v Silverlight	38
3.2.1	Transformácia súradníc	39
4	Vývoj aplikácie	42
4.1	Vytvorenie aplikácie	42
4.2	Definície splajnov ako dátových typov	44

4.2.1	Spline	44
4.2.2	BSpline	45
4.2.3	GlobalBSpline	45
4.2.4	HermiteSpline	46
4.2.5	ClampedSpline	46
4.3	Generovanie a vykreslenie splajnov	47

III Záver 52

Úvod

Témou našej bakalárskej práce sú splajn krivky, kde naším cieľom je preskúmať nové poznatky o splajnoch, na ktorých istý čas pracuje vedúci tejto práce doc. RNDr. Csaba Török, CSc. Výsledkom tejto práce je okrem návrhu tejto novej metódy aj porovnanie so známymi metódami a implementácia aplikácie na vizuálne porovnávanie jednotlivých metód. Technológia v ktorej budeme tieto poznatky implementovať je Microsoft Silverlight. Jedná o veľmi schopný nástroj na tvorbu webových aplikácií s plnou podporou hardvérovej akcelerácie užívateľského prostredia. Výhodou tohto frameworku je, keďže beží na platforme Microsoft .NET, možnosť jednoduchej portácie na desktopovú prípadne mobilnú aplikáciu.

Štruktúra práce je nasledovná.

- **Úvod**

V úvode si vysvetlíme technológiu Silverlight a základy práce grafiky.

- **Splajn krivky**

V tejto stati si povieme základnú definíciu splajnu ako reálnu funkciu reálnej premennej a následne prejdeme k podrobným rozborom konkrétnych kriviek a sítě B-splajnov, globálnych B-splajnov, Hermitovských splajnov a clamped splajnov.

- **Nový postup generovania clamped splajnov**

V tejto kapitole rozoberieme novo vyvinutý postup konštrukcie clamped splajnov.

- **Implementácia**

Podrobnosti implementácie splajnov v MS Silverlight.

Prehľad súčasného stavu

Teória splajnov je veľmi rozsiahla. Okrem klasických výsledkov budeme vychádzať aj z výsledkov výskumu vedúceho tejto práce doc. Töröka. Tejto

problematike sa však venujú aj iní, napríklad Nikolaj Demjanovic Dikudar z Ústavu jadrového výskumu v Dubne. Splajnami sa celkovo zaoberajú napr. David Solomon, alebo Carl de Boor z University of Wisconsin-Madison.

Časť I

Návrh riešenia

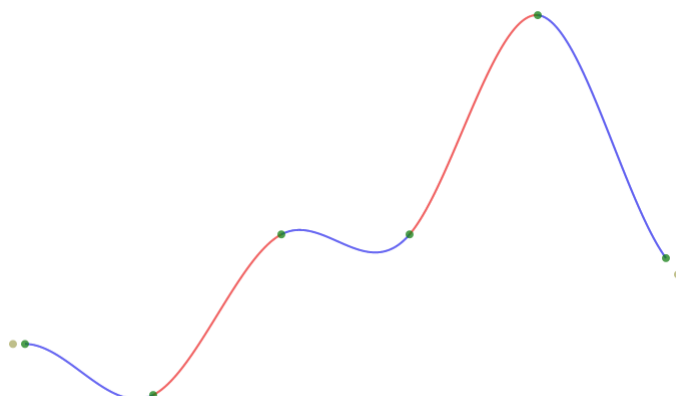
Kapitola 1

Splajn krivky

V tejto stati si povieme základnú definíciu splajnu ako reálnu funkciu reálnej premennej a následne prejdeme k podrobným rozborom konkrétnych kriviek a sítě B-splajnov, globálnych B-splajnov, Hermitovských splajnov a clamped splajnov. V ďalšej kapitole rozoberieme nový postup konštrukcie globálnych B-splajnov pomocou clamped splajnov.

1.0.1 Ilustračný príklad

Predtým, ako si povieme presnú definíciu, si názorne ukážme čo je splajn, ako vyzerá a z čoho sa skladá.



Obr. 1.1: Uniformný globálny B-splajn stupňa 3

Ako vidieť z obrázka, globálny B-splajn je určený pomocou bodov v rovine (zelenou farbou), ktoré nám dávajú (jednoznačné) vyjadrenie krivky. Všimnime si, že tieto body prechádzajú krivkou, jedná sa teda o interpolačnú

krivku. Podrobnejšie detaily zatiaľ vynecháme, budeme sa im totiž venovať v nasledujúcich statiach, kde si presnejšie definujeme niektoré druhy splajnov, konkrétne Hermitovské splajne, Clamped splajny a samozrejme B-splajny.

1.0.2 Formálna definícia splajnov

Než začneme pracovať s jednotlivými krivkami zavedieme všeobecný pojem, z ktorého budeme odvádzať jednotlivé splajny.

Definícia 1.1 Nech n, k sú prirodzené čísla, $V = \langle v_0, v_1, \dots, v_{k-1} \rangle$ a $C = \langle c_0, c_1, \dots, c_{k-1} \rangle$ sú k -tice reálnych čísel. Nech C_0, \dots, C_{k-1} sú tice, ktorých prvky sú z k -tice C . Nech platí:

- $n \geq 3$,
- $k \geq 1$,
- Postupnosť V je rastúca.
- Pre každé i z $\{1, 2, \dots, k-1\}$ je $P_i^{C_i}$ polynóm stupňa n na intervale $[v_{i-1}, v_i]$.

Definujme funkciu $\text{Spline}_{V,C}$ z intervalu $[v_0, v_{k-1}]$ do \mathbb{R} vzťahom:

$$\text{Spline}_{V,C}(x) = \begin{cases} P_1^{C_1}(x) & \text{pre } v_0 \leq x < v_1, \\ P_2^{C_2}(x) & \text{pre } v_1 \leq x < v_2, \\ \vdots & \\ P_{k-1}^{C_{k-1}}(x) & \text{pre } v_{k-2} \leq x \leq v_{k-1}. \end{cases} \quad (1.1)$$

pričom existuje j z $\{0, \dots, n-1\}$ také, že pre každé i z $\{1, \dots, k-1\}$ platí $P_i^{C_i(j)}(v_i) = P_{i+1}^{C_{i+1}(j)}(v_i)$, kde $P_i^{C_i(j)}$ je j -ta derivácia polynomu $P_i^{C_i}$

Označenie 1.2 Funkciu $\text{Spline}_{V,C}$ z predchádzajúcej definície budeme nazývať *splajn*.

Označenie 1.3 Hodnotu n z predchádzajúcej definície budeme nazývať *stupeň splajnu*.

Označenie 1.4 Hodnoty v_0, v_1, \dots, v_{k-1} z predchádzajúcej definície budeme nazývať *uzly splajnu*, alebo skrátene *uzly*.

Označenie 1.5 Hodnoty c_0, c_1, \dots, c_{k-1} z predchádzajúcej definície budeme nazývať *kontrolné body*.

Označenie 1.6 Ak hodnoty v_0, v_1, \dots, v_{k-1} sú uzly tak pre každé i z $\{0, \dots, k-2\}$ budeme interval $[v_i, v_{i+1}]$ nazývať *segment splajnu*, alebo skrátene *segment*.

Z tejto definície vyplýva, že splajn je polynomická funkcia zložená z niekoľkých polynómov. Pravda, tieto polynómy nemôžu byť ľubovoľné, ale mali by spĺňať určité vlastnosti, teda (napríklad) rovnosť derivácii do určitého rádu. Nutno dodať, že sa jedná o všeobecnú definíciu. Pri konštrukcii jednotlivých segmentov splajnu teda potrebujeme ako vstupné parametre rastúcu postupnosť uzlov V a postupnosť C tzv. *kontrolných bodov* (pri aproximačných krivkách), resp. *funkčných hodnôt* (pri interpolačných krivkách), teda postupnosť čísel, ktoré spolu s uzlami predstavujú body v rovine, pričom v_i predstavuje súradnicu na osi x a príslušný kontrolný bod (resp. funkčná hodnota) c_i predstavuje súradnicu na osi y , pre každé i z $\{0, \dots, k-1\}$. Tieto body sú vo väčšine typov splajnov parametre pre polynómy P_i .

Poznámka 1.7 V Computed Aided Graphics systémoch (skrátene CAD) sa používajú tzv. *parametrické splajny*, kde vstupný atribút x a funkčná hodnota y závisia od nejakého reálneho parametra, pričom výsledná krivka nemusí predstavovať funkciu, stručne povedané v krivke je viac bodov „nad sebou“. My však budeme pracovať so splajnami, ktoré nie sú parametrické, teda s takými ktoré sú vždy funkciami.

1.0.3 Delenie splajnov

Teraz si zavedieme niekoľko pojmov, ktoré definujú vlastnosti splajnov.

Definícia 1.8 Funkciu $\text{Spline}_{V,C}$ nazveme *uniformná*, ak pre každé i a j z $\{1, 2, \dots, k-1\}$ a dvojice podintervalov $[v_{i-1}, v_i]$ a $[v_{j-1}, v_j]$ intervalu $[v_0, v_{k-1}]$ z definície funkcie Spline platí, že dĺžka intervalu $[v_{i-1}, v_i]$ je rovná dĺžke intervalu $[v_{j-1}, v_j]$.

Inak funkciu Spline na intervale $[v_0, v_{k-1}]$ nazveme *neuniformná*.

Základné delenie splajnov môžeme definovať na *aproximačné* krivky a *interpolačné* krivky. Interpolačné krivky sú také krivky, ktoré prechádzajú cez všetky body, ktorými sú definované. To je prípad Hermitovských splajnov, naturálnych splajnov, clamped splajnov a globálnych B-splajnov. V prípade aproximačných kriviek toto neplatí a funkcia nemusí prechádzať bodmi, ktorými je definovaná, napríklad v prípade B-splajnov.

Zavedme si teda aj formálne definície:

Definícia 1.9 Nech $V = (v_0, v_1, \dots, v_{k-1})$ sú uzly a $C = (c_0, c_1, \dots, c_{k-1})$ sú kontrolné body. Funkciu $\text{Spline}_{V,C}$ nazveme *interpolačná* ak pre každé i

z $\{0, \dots, k-1\}$ platí $\text{Spline}_{V,C}(v_i) = c_i$. Inak funkciu $\text{Spline}_{V,C}$ nazveme *aproximačná*.

Označenie 1.10 V prípade interpolačných splajnov budeme kontrolné body nazývať *funkčné hodnoty v uzloch*, alebo skráteno len *funkčné hodnoty*.

Ďalej sa splajny delia na *lokálne* a *globálne*. Rozdiel spočíva v správaní sa krivky pri zmene jedného parametra, či už uzla, alebo kontrolného bodu. Pri lokálnych splajnoch sa pri zmene jedného parametra zmení len časť krivky, zatiaľ čo v prípade globálnych splajnov dôjde k zmene celej krivky.

1.1 B-splajny (Bázové splajny)

Tento typ krivky bol definovaný v štyridsiatych rokoch dvadsiateho storočia, avšak výraznejšej pozornosti sa im dostalo až v sedemdesiatych rokoch [1]. V súčasnosti je zaužívaný skrátený názov *B-Splajn*. Najviac sa týmto krivkám podobajú *Bézierove krivky*, presnejšie povedané Bézierove krivky sú špeciálnym prípadom B-splajnov. B-splajny však majú výhodu v tom, že počet riadiacich bodov nie je závislý od stupňa polynómu. Jedná sa o aproximačnú krivku, teda neprechádza svojimi kontrolnými bodmi. Neskôr si ukážeme aj interpolačný variant zvaný globálny B-splajn.

V nasledujúcej stati si postupne zostrojíme B-splajn, pričom najprv začneme konštrukciou bázových funkcií pre jednotlivé segmenty krivky, až nakoniec z nich získame chcenú krivku.

1.1.1 Bell funkcie

Ako sme si povedali, splajn sa skladá zo segmentov, pričom každý z nich je polynomická funkcia. Tieto polynómy sú (nielen u B-splajnov) tvorené súčtom súčinov kontrolných bodov a takzvaných bázových funkcií. V prípade B-splajnu sa tieto funkcie nazývajú aj *bell* (*zvonové*) funkcie. Definujme si teda bázovú funkciu B-splajnu.

Definícia 1.11 Nech n, k a p sú prirodzené čísla, pričom $p \leq n+1$, $k \geq n+2$ a v_0, v_1, \dots, v_{k-1} sú reálne čísla také, že pre každé i z $\{1, 2, \dots, k-1\}$ platí $v_{i-1} < v_i$. Nech j je prirodzené číslo také, že $0 \leq j \leq k-5$ Definujme funkciu $\text{Bs}_{v_j, \dots, v_{j+n+1}}^{n,p}$ z intervalu $[v_{j+n}, v_{j+n+1}]$ do \mathbb{R} vzťahom

$$\text{Bs}_{v_j, \dots, v_{j+n+1}}^{n,p}(x) = (x - v_j)^n - \sum_{k=j+1}^{j+p} (x - v_k)^n \cdot \prod_{m=j+1, m \neq k}^{n+1} \frac{v_j - v_m}{v_k - v_m} \quad (1.2)$$

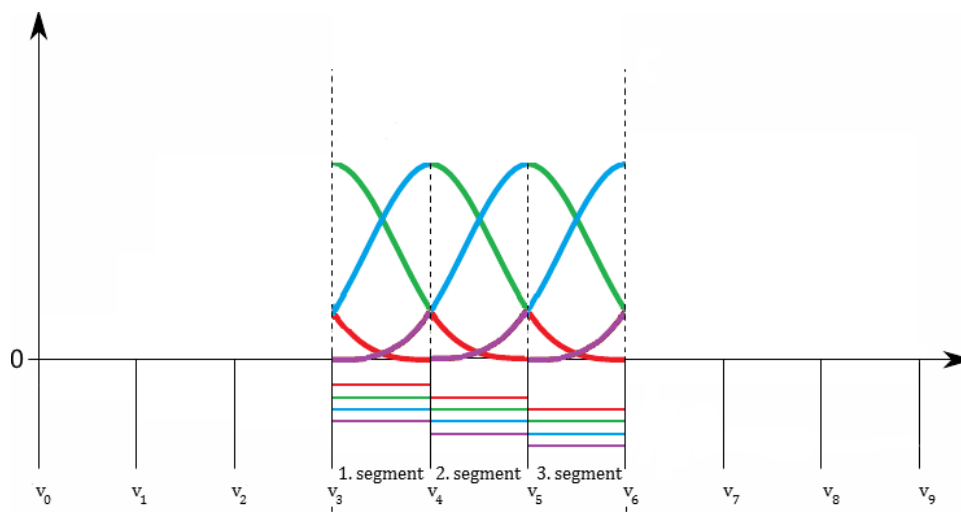
Označenie 1.12 Funkciu $Bs_{v_j, \dots, v_{j+n+1}}^{n,p}$ budeme všeobecne označovať pojmom *bázová funkcia*, alebo aj *bell funkcia*.

Parameter n znamená stupeň polynomickej funkcie $Bs_{v_0, \dots, v_{n+1}}^{n,p}$, dokonca sa jedná sa o stupeň celého splajnu. Pre jeden segment B-splajnu stupňa n potrebujeme $n + 1$ bázových funkcií, ktoré sa budú líšiť parametrom p a vstupnými uzlami. Tieto funkcie sú lineárne nezávislé a teda generujú nejaký priestor funkcií (preto názov *bázový splajn*). Vynásobením týchto funkcií $n + 1$ kontrolnými bodmi dostaneme konkrétnu lineárnu kombináciu (teda vstupnými súradnicami na osi y , pričom hodnoty $v_0, v_1, \dots, v_{2n+1}$ – uzly predstavujú súradnice na osi x), teda jeden polynóm, ktorý tvorí práve jeden segment B-splajnu.

Teda na jeden segment B-splajnu potrebujeme vedieť stupeň bázových funkcií n (ten bude pre celý splajn rovnaký), $2n + 2$ uzlov – súradníc na osi x a $n + 1$ kontrolných bodov – súradníc na osi y . Dôvod prečo potrebujeme $2n + 2$ uzlov je jednoduchý. Keďže pre jeden segment potrebujeme $n + 1$ bell funkcií a každá potrebuje $n + 2$ uzlov, pričom pre prvú bell funkciu vezmeme uzly $v_0, \dots, v_{(n+1)+0} = v_{n+1}$, pre druhú vezmeme uzly $v_1, \dots, v_{(n+1)+1} = v_{n+2}$, až nakoniec pre $(n + 1)$. funkciu vezmeme uzly $v_n, \dots, v_{(n+1)+n} = v_{2n+1}$. Teda pre jeden segment potrebujeme uzly v_0 až v_{2n+1} . Uzly indexujeme od čísla 0, preto dohromady máme $2n + 2$ uzlov. Ak kontrolné body označíme $c_0, c_1, \dots, c_n + 1$, tak na plátne (či už v Silverlight-e, alebo obecné) budú body znázornené dvojicami $\langle v_1, c_0 \rangle, \langle v_2, c_1 \rangle, \dots, \langle v_n + 2, c_n + 1 \rangle$. Samotná krivka bude vykreslená na intervale $[v_n, v_{n+1}]$. V prípade uzlov $v_0, v_{n+2}, v_{n+3}, \dots, v_{2n+1}$ nezáleží na hodnote kontrolných bodov, ich hodnoty určujú hodnotu derivácie splajnu v bodoch v_n a v_{n+1} . To je dôležité pri následnom napájaní ďalších segmentov.

V následujúcom príklade si ukážeme zostrojenie bázových funkcií pre B-splajn stupňa 3 s desiatimi uzlami a teda s tromi segmentami. Kontrolné body teraz nepotrebujeme, keďže zostrojíme iba bázové funkcie. V ďalšej stati si ukážeme, že výsledný splajn bude lineárna kombinácia kontrolných bodov a bázových funkcií.

Príklad 1.13 Nech $n = 3$ je stupeň splajnu a $\langle v_0, v_1, \dots, v_9 \rangle = (0, 1, \dots, 9)$ sú uzly. Na jeden segment B-splajnu teda potrebujeme 4 bázové funkcie a 8 uzlov. Keďže máme 10 uzlov tak počet segmentov v tomto príklade bude 3, pričom pre konštrukciu bázových funkcií pre prvý segment použijeme uzly 0, 1, ..., 7, pre druhý segment uzly 1, 2, ..., 8 a pre posledný segment použijeme uzly 2, 3, ..., 9. Pozrime sa na ilustračný obrázok výsledku a nižšie si ho vysvetlíme. Na obrázku máme grafy dvanástich bázových funkcií pre tri segmenty B-splajnu. Grafy týchto funkcií majú následujúci zápis, pričom si



Obr. 1.2: Bázové funkcie trojsegmentového B-splajnu stupňa 3

ich kvôli prehľadnosti zapíšeme farebne podľa jednotlivých segmentov:

- 1. segment
 - $BS_{0,1,2,3,4}^{3,1}$
 - $BS_{1,2,3,4,5}^{3,2}$
 - $BS_{2,3,4,5,6}^{3,3}$
 - $BS_{3,4,5,6,7}^{3,4}$
- 2. segment
 - $BS_{1,2,3,4,5}^{3,1}$
 - $BS_{2,3,4,5,6}^{3,2}$
 - $BS_{3,4,5,6,7}^{3,3}$
 - $BS_{4,5,6,7,8}^{3,4}$
- 3. segment
 - $BS_{2,3,4,5,6}^{3,1}$
 - $BS_{3,4,5,6,7}^{3,2}$
 - $BS_{4,5,6,7,8}^{3,3}$
 - $BS_{5,6,7,8,9}^{3,4}$

Všimnime si, že grafy funkcií tej istej farby sú, až na posunutý definičný obor, identické aj keď majú rôzny zápis. Je to spôsobené tým, že v príklade uvažujeme uniformný splajn, v prípade neuniformného splajnu by tieto funkcie mali rozličné grafy.

V tejto stati sme si definovali bell funkcie, ktoré sú základným konštrukčným prvkom pri generovaní B-splajnu. V nasledujúcej stati si ukážeme samotnú konštrukciu B-splajnu.

1.1.2 B-splajn

Definovali sme si teda funkciu z ktorej vieme vygenerovať B-splajn, ale stále netušíme ako na to. V tejto stati si povieme ako skonštruovať bázový splajn stupňa n všeobecne.

Zostrojenie B-splajnu Nech n je prirodzené číslo a nech sú dané reálne čísla v_0, \dots, v_k – uzly a c_0, \dots, c_m – kontrolné body, pričom k a m sú prirodzené čísla také, že platí:

- $k > n + 1$
- $m > n$

Našou úlohou je zostrojiť B-splajn stupňa n na intervale $[v_0, v_k]$. Konštrukcia krivky bude prebiehať po stadiach v $s = \min\{k - 1 - 2n, m - n\}$ krokoch, pričom v prvom kroku použijeme uzly v_0, \dots, v_{n+1} a kontrolné body c_0, \dots, c_n . Vďaka podmienkam máme zaručené, že aspoň jeden segment môžeme zkonštruovať. V prípade druhého segmentu použijeme uzly v_1, \dots, v_{n+2} a kontrolné body c_1, \dots, c_{n+1} atď. až kým nevyčerpáme buď uzly, alebo kontrolné body.

Ako sme povedali v predchádzajúcej stati, rovnicu výslednej krivky získame vynásobením $n + 1$ bell funkcií a $n + 1$ kontrolných bodov. Označme P_1, \dots, P_s polynomicke funkcie, ktoré tvoria jednotlivé segmenty B-Splajnu. Ich rovnice pre každé i z $\{1, \dots, s\}$ vyzerajú takto:

$$P_i(x) = \sum_{j=0}^n (c_{i+j-1} \cdot \text{Bs}_{v_{i+j-1}, \dots, v_{i+j+n}}^{n, j+1}(x)). \quad (1.3)$$

Môžeme teda pristúpiť k formálnej definícii B-splajnu.

Definícia 1.14 Nech n je stupeň splajnu, k, m, s a o sú prirodzené čísla, $V = \langle v_0, v_1, \dots, v_{k-1} \rangle$ sú uzly a $C = \langle c_0, c_1, \dots, c_m \rangle$ sú kontrolné body. Nech platí:

- $n \geq 3$,
- $k \geq n + 2$,
- $m \geq n + 1$,
- $s = \min\{k - 1 - 2n, m - n\}$
- $o = \min\{k - 1 - n, m + 1\}$

Definujme funkciu $\text{BSpline}_{V,C}$ z intervalu $[v_n, v_o]$ do \mathbb{R} vzťahom:

$$\text{BSpline}_{V,C}(x) = \begin{cases} \sum_{j=0}^n (c_j \cdot \text{Bs}_{v_j, \dots, v_{j+n+1}}^{n,j+1}(x)) & \text{pre } v_0 \leq x < v_1, \\ \sum_{j=0}^n (c_{1+j} \cdot \text{Bs}_{v_{1+j}, \dots, v_{j+n+2}}^{n,j+1}(x)) & \text{pre } v_1 \leq x < v_2, \\ \vdots & \\ \sum_{j=0}^n (c_{s-1+j} \cdot \text{Bs}_{v_{s-1+j}, \dots, v_{i+j+n+s}}^{n,j+1}(x)) & \text{pre } v_{o-1} \leq x \leq v_o. \end{cases} \quad (1.4)$$

V nasledujúcom príklade si vygenerujeme B-splajn pomocu báзовých funkcií z príkladu 1.13

Príklad 1.15 Uvažujme uzly a báзовé funkcie z príkladu 1.13, teda uzly $V = \langle 0, \dots, 9 \rangle$ a báзовé funkcie:

- 1. segment
 - $\text{Bs}_{0,1,2,3,4}^{3,1}$
 - $\text{Bs}_{1,2,3,4,5}^{3,2}$
 - $\text{Bs}_{2,3,4,5,6}^{3,3}$
 - $\text{Bs}_{3,4,5,6,7}^{3,4}$
- 2. segment
 - $\text{Bs}_{1,2,3,4,5}^{3,1}$
 - $\text{Bs}_{2,3,4,5,6}^{3,2}$
 - $\text{Bs}_{3,4,5,6,7}^{3,3}$
 - $\text{Bs}_{4,5,6,7,8}^{3,4}$
- 3. segment
 - $\text{Bs}_{2,3,4,5,6}^{3,1}$

- $\text{Bs}_{3,4,5,6,7}^{3,2}$
- $\text{Bs}_{4,5,6,7,8}^{3,3}$
- $\text{Bs}_{5,6,7,8,9}^{3,4}$

Nech $\langle c_0, c_1, c_2, c_3, c_4 v_5 \rangle = \langle -\frac{1}{3}, \frac{1}{5}, \frac{1}{4}, -\frac{1}{4}, \frac{1}{3}, -\frac{3}{4} \rangle$ sú kontrolné body. Rovnice pre jednotlivé segmenty P_1 , P_2 a P_3 budú podľa rovnice 1.3 vyzerat takto:

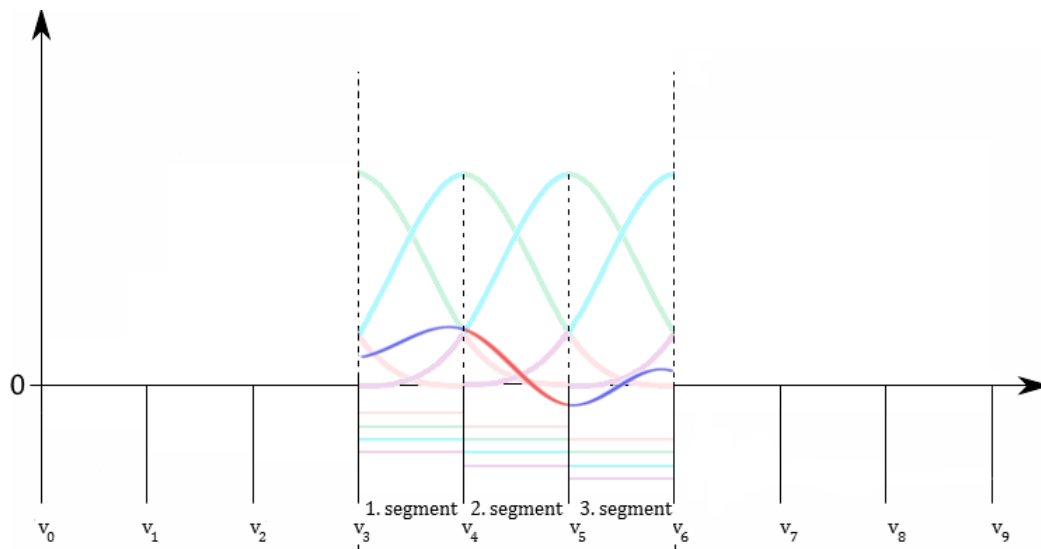
$$P_1(x) = \sum_{j=0}^3 (c_{0+j} \cdot \text{Bs}_{v_{0+j}, \dots, v_{0+j+3+1}}^{3,j+1}(x)),$$

$$P_2(x) = \sum_{j=0}^3 (c_{1+j} \cdot \text{Bs}_{v_{1+j}, \dots, v_{1+j+3+1}}^{3,j+1}(x)),$$

$$P_3(x) = \sum_{j=0}^3 (c_{2+j} \cdot \text{Bs}_{v_{2+j}, \dots, v_{2+j+3+1}}^{3,j+1}(x)).$$

Po rozpísaní súm a dosadení hodnôt dostávame:

$$\begin{aligned}
P_1(x) &= -\frac{1}{3} \cdot \text{Bs}_{0,1,2,3,4}^{3,1}(x) \\
&\quad + \frac{1}{5} \cdot \text{Bs}_{1,2,3,4,5}^{3,2}(x) \\
&\quad + \frac{1}{4} \cdot \text{Bs}_{2,3,4,5,6}^{3,3}(x) \\
&\quad - \frac{1}{4} \cdot \text{Bs}_{3,4,5,6,7}^{3,4}(x), \\
P_2(x) &= \frac{1}{5} \cdot \text{Bs}_{1,2,3,4,5}^{3,1}(x) \\
&\quad + \frac{1}{4} \cdot \text{Bs}_{2,3,4,5,6}^{3,2}(x) \\
&\quad - \frac{1}{4} \cdot \text{Bs}_{3,4,5,6,7}^{3,3}(x) \\
&\quad + \frac{1}{3} \cdot \text{Bs}_{4,5,6,7,8}^{3,4}(x), \\
P_3(x) &= \frac{1}{4} \cdot \text{Bs}_{2,3,4,5,6}^{3,1}(x) \\
&\quad - \frac{1}{4} \cdot \text{Bs}_{3,4,5,6,7}^{3,2}(x) \\
&\quad + \frac{1}{3} \cdot \text{Bs}_{4,5,6,7,8}^{3,3}(x) \\
&\quad - \frac{4}{4} \cdot \text{Bs}_{5,6,7,8,9}^{3,4}(x),
\end{aligned}$$



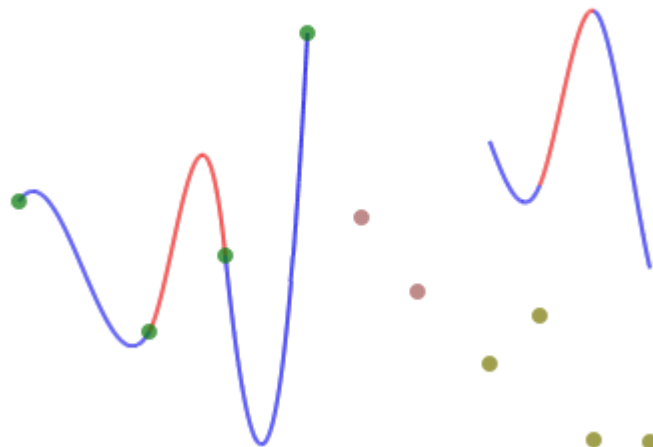
Obr. 1.3: Trojsegmentový B-splajn stupňa 3 (červenou a modrou), v pozadí bázové funkcie

1.2 Globálny B-splajn

B-splajn je aproximačná krivka, teda funkčná hodnota splajnu v nejakom uzle sa nerovná hodnote príslušného kontrolného bodu. Pre používateľov však nie je pohodlné, ak vykresľovaná krivka stručne povedané „lieta kadiaľ-tadiaľ“. Existuje však postup, ako pomocou zadaných bodov, pričom ich x -ové súradnice predstavujú uzly a y -súradnice predstavujú funkčné hodnoty B-splajnu, jednoznačne vypočítať kontrolné body B-splajnu a takto zkonštruovať celú krivku.

Poznámka 1.16 Globálny B-splajn je možné samozrejme zkonštruovať aj z B-splajnov vyšších stupňov, pre potreby tejto práce budeme uvažovať iba splajny 3. stupňa.

Vstupné atribúty globálneho B-splajnu sú odlišné od „obyčajného“ B-splajnu. Na jeho konštrukciu potrebujeme k uzlov, pričom $k \geq 2$ a rovnaký počet funkčných hodnôt žiadaného B-splajnu krivky v týchto uzloch. Ďalej potrebujeme hodnoty prvých derivácií v prvom a poslednom uzle.



Obr. 1.4: Interpoláčné krivky sú aj na prvý pohľad intuitívnejšie než aproximačné

1.2.1 Pomocné uzly

Globálny B-splajn je síce interpolačná a globálna krivka, avšak stále sa jedná iba o modifikovaný aproximačný a lokálny B-splajn stupňa 3. Keďže je to interpolačná krivka, tak musí začínať v prvom zadanom uzle a končiť v poslednom uzle. My ale vieme, že B-splajn krivka stupňa n sa začína vykresľovať až od n . uzla (nezabúdajme, že uzly indexujeme od čísla 0). Preto ak chceme aby B-splajn krivka bola vykreslená nad všetkými zadanými uzlami potrebujeme vypočítať $2n$ pomocných uzlov, t.j. uzlov, nad ktorými B-splajn krivka nie je vykreslená. V našej práci budeme používať nasledujúci spôsob.

Definícia 1.17 Nech $n \geq 3$ je stupeň splajnu a $k \geq 2$ je prirodzené číslo a v_0, \dots, v_{k-1} sú uzly. Označme $h = \frac{(u_{k-1}-u_0)}{(k-1)}$. Nech u_0, \dots, u_{k+5} sú reálne čísla také, že pre každé i z $\{0, \dots, k+2n-1\}$ platí:

$$u_i = \begin{cases} v_0 - (n-i) \cdot h & \text{ak } i \leq n-1, \\ v_{i-n} & \text{ak } n \leq i \leq k-1+n, \\ v_{k-1} + (i-n-k+1) \cdot h & \text{inak.} \end{cases} \quad (1.5)$$

Potom $(k+2n-1)$ -ticu $\langle u_0, \dots, u_{k+2n-1} \rangle$ budeme nazývať *pomocné uzly*.

Pre potreby tejto práce je však vhodnejšie pomocné uzly definovať ako funkciu, k premenných, ktorej vstupom budú uzly a výstupom pomocné uzly. Zavedme si preto aj alternatívnu definíciu.

Definícia 1.18 Nech $n \geq 3$ a $k \geq 2$ sú prirodzené čísla a v_0, \dots, v_{k-1} sú uzly. Označme $h = \frac{(u_{k-1}-u_0)}{(k-1)}$. Definujme funkciu $\text{SKnots}_{n,k}$ z \mathbb{R}^k do \mathbb{R}^{k+2n} vzťahom:

$$\begin{aligned} \text{SKnots}_{n,k}(v_0, \dots, v_{k-1}) = \langle &v_0 - n \cdot h, v_0 - (n-1) \cdot h, \\ &v_0 - (n-2), \dots, v_0 - (n-(n-1)) \cdot h, \\ &v_0, v_1, \dots, v_{k-1}, \\ &v_{k-1} + 1 \cdot h, v_{k-1} + 2 \cdot h, \dots, v_{k-1} + n \cdot h \rangle \end{aligned} \quad (1.6)$$

Poznámka 1.19 Hodnoty u_0, \dots, u_{k+2n-1} teda predstavujú uzly, pomocou ktorých už môžeme zostrojiť B-splajn tak, aby krivka začínala v bode $u_2 = v_0$ a končila v bode $u_{k+2} = v_{k-1}$.

V nasledujúcom príklade si ukážeme generovanie pomocných uzlov, pričom v nasledujúcej stati si ukážeme výpočet kontrolných bodov B-splajnu zo zadaných funkčných hodnôt globálneho B-splajnu a prvých derivácií v prvom a poslednom uzle.

Príklad 1.20 Nech $\langle v_0, v_1, v_2 \rangle = \langle 2, \frac{15}{4}, 5 \rangle$ sú uzly. Položme číslo h a $\langle u_0, u_1, \dots, u_7, u_8 \rangle$ tak, že platí (podľa 1.17):

- $h = \frac{(5-2)}{2} = \frac{3}{2}$
- Pre každé i z $\{0, 1, \dots, 8\}$ platí:

$$u_i = \begin{cases} v_0 - (3-i) \cdot \frac{3}{2} & \text{ak } i \leq 2, \\ v_{i-3} & \text{ak } 3 \leq i \leq 5, \\ v_2 + (i-3-3+1) \cdot \frac{3}{2} & \text{inak.} \end{cases}$$

Teda pomocné uzly budú mať hodnoty $\langle -\frac{5}{2}, -1, \frac{1}{2}, 2, \frac{15}{4}, 5, \frac{13}{2}, 8, \frac{19}{2} \rangle$.

1.2.2 Generovanie kontrolných bodov B-splajnu

V prípade lokálneho B-splajnu sme na jeho konštrukciu potrebovali na vstupe tzv. kontrolné body (hodnoty na osi y). Keďže globálny B-splajn je interpolačná krivka, tak zadané hodnoty na osi y musia byť funkčnými hodnotami globálneho B-splajnu v príslušnom uzle, inak povedané, krivka musí prechádzať týmito bodmi. Existuje však spôsob ako pomocou známych funkčných hodnôt B-splajnu v uzloch získať hodnoty kontrolných bodov a jednoznačne zostrojiť B-splajn krivku.

Definícia 1.21 Nech k je z \mathbb{N} , $k \geq 2$ a hodnoty $V = \langle v_0, \dots, v_{k-1} \rangle$ sú vstupné uzly, hodnoty $\langle c_0, \dots, c_{k-1} \rangle$ sú kontrolné body, číslo d_{left} je hodnota derivácie v bode u_0 a číslo d_{right} je hodnota derivácie v bode u_{k-1} . Nech $(k+6)$ -tica $\langle u_0, \dots, u_{k+5} \rangle = \text{SKnots}_{3,k}(V)$ predstavuje pomocné uzly. Označme prvú deriváciu funkcie $\text{Bs}_{v_a}^{b,c}$ ako $\text{Bd}_{v_a}^{b,c}$ pre ľubovoľné a, b, c prirodzené. Ďalej definujme maticu \mathbf{L} vzťahom:

$$\mathbf{L} = \begin{pmatrix} \text{Bs}_{u_0}^{3,1}(v_0) & \text{Bs}_{u_1}^{3,2}(v_0) & \text{Bs}_{u_2}^{3,3}(v_0) & \text{Bs}_{u_3}^{3,4}(v_0) & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & \text{Bs}_{u_1}^{3,1}(v_1) & \text{Bs}_{u_2}^{3,2}(v_1) & \text{Bs}_{u_3}^{3,3}(v_1) & \text{Bs}_{u_4}^{3,4}(v_1) & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & \text{Bs}_{u_2}^{3,1}(v_2) & \text{Bs}_{u_3}^{3,2}(v_2) & \text{Bs}_{u_4}^{3,3}(v_2) & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \text{Bs}_{u_3}^{3,1}(v_3) & \text{Bs}_{u_4}^{3,2}(v_3) & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{Bs}_{u_4}^{3,1}(v_4) & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & \text{Bs}_{u_{k-3}}^{3,1}(v_{k-2}) & \text{Bs}_{u_{k-2}}^{3,2}(v_{k-2}) & \text{Bs}_{u_{k-1}}^{3,3}(v_{k-2}) & \text{Bs}_{u_k}^{3,4}(v_{k-2}) \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & \text{Bs}_{u_{k-2}}^{3,1}(v_{k-1}) & \text{Bs}_{u_{k-1}}^{3,2}(v_{k-1}) & \text{Bs}_{u_k}^{3,3}(v_{k-1}) \\ \text{Bd}_{u_0}^{3,1}(v_0) & \text{Bd}_{u_1}^{3,2}(v_0) & \text{Bd}_{u_2}^{3,3}(v_0) & \text{Bd}_{u_3}^{3,4}(v_0) & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & \text{Bd}_{u_{k-2}}^{3,1}(v_{k-1}) & \text{Bd}_{u_{k-1}}^{3,2}(v_{k-1}) & \text{Bd}_{u_k}^{3,3}(v_{k-1}) \end{pmatrix} \quad (1.7)$$

Nech vektor (b_0, \dots, b_{k+1}) je riešením následujúcej sústavy rovníc:

$$\mathbf{L} \cdot \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{k+1} \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ \vdots \\ c_{k-2} \\ c_{k-1} \\ d_{\text{left}} \\ d_{\text{right}} \end{pmatrix} \quad (1.8)$$

Potom $k+1$ -ticu $\langle b_0, \dots, b_{k+1} \rangle$ potom nazveme *pomocné kontrolné body globálneho B-splajnu*, laebo skrátené *pomocné kontrolné body*.

Podobne ako pri pomocných uzloch, aj tu si zavedieme alternatívnu definíciu v podobe funkcie.

Definícia 1.22 Nech k je z \mathbb{N} , $k \geq 2$ a hodnoty $V = \langle v_0, \dots, v_{k-1} \rangle$ sú vstupné uzly, hodnoty $\langle c_0, \dots, c_{k-1} \rangle$ sú kontrolné body, číslo d_{left} je hodnota derivácie v bode u_0 a číslo d_{right} je hodnota derivácie v bode u_{k-1} . Nech $(k+6)$ -tica $U = \langle u_0, \dots, u_{k+5} \rangle = \text{SKnots}_{3,k}(V)$ predstavuje pomocné uzly. Označme

prvú deriváciu funkcie $\text{Bs}_{v_a}^{b,c}$ ako $\text{Bd}_{v_a}^{b,c}$ pre ľubovoľné a, b, c prirodzené. Položme \mathbf{L} z rovnice 1.7. Definujme funkciu $\text{SCPoints}_{V,U}^k$ z \mathbb{R}^{k+2} do \mathbb{R}^{k+2} vzťahom:

$$\text{SCPoints}_{V,U}^k(c_0, \dots, c_{k-1}, d_{\text{left}}, d_{\text{right}}) = \mathbf{L}^{-1} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ \vdots \\ c_{k-2} \\ c_{k-1} \\ d_{\text{left}} \\ d_{\text{right}} \end{pmatrix} \quad (1.9)$$

Teda ak máme zadané uzly v_0, \dots, v_{k-1} , hodnoty c_0, \dots, c_{k-1} , derivácie d_{left} a d_{right} , tak pomocou vzťahov 1.5 a 1.11 získame uzly u_0, \dots, u_{k+5} a kontrolné body b_0, \dots, b_{k+1} také, že z nich zostrojený B-splajn stupňa 3 bude prechádzať všetkými bodmi so súradnicami $\langle u_0, b_0 \rangle$ až $\langle u_{k-1}, b_{k-1} \rangle$. Týmto spôsobom zostrojenú B-splajn krivku budeme nazývať globálny B-splajn. Pre lepšiu predstavu si to ukážme na ilustračnom príklade.

Príklad 1.23 Uvažujme uzly $V = \langle 2, \frac{15}{4}, 5 \rangle$ a pomocné uzly $U = \langle -\frac{5}{2}, -1, \frac{1}{2}, 2, \frac{15}{4}, 5, \frac{13}{2}, 8, \frac{19}{2} \rangle$ (z príkladu 1.20), funkčné hodnoty $C = \langle -\frac{1}{2}, \frac{11}{10}, \frac{3}{4} \rangle$ a dvojicu prvých derivácií v uzloch v_0 a v_2 $\langle d_{\text{left}}, d_{\text{right}} \rangle = \langle -2, \frac{4}{5} \rangle$. Označme prvú deriváciu funkcie $\text{Bs}_{v_a}^{b,c}$ ako $\text{Bd}_{v_a}^{b,c}$ pre ľubovoľné a, b, c prirodzené. Položme maticu \mathbf{L} vzťahom:

$$\mathbf{L} = \begin{pmatrix} \text{Bs}_{u_0}^{3,1}(v_0) & \text{Bs}_{u_1}^{3,2}(v_0) & \text{Bs}_{u_2}^{3,3}(v_0) & \text{Bs}_{u_3}^{3,4}(v_0) & 0 \\ 0 & \text{Bs}_{u_1}^{3,1}(v_1) & \text{Bs}_{u_2}^{3,2}(v_1) & \text{Bs}_{u_3}^{3,3}(v_1) & \text{Bs}_{u_4}^{3,4}(v_1) \\ 0 & 0 & \text{Bs}_{u_2}^{3,1}(v_2) & \text{Bs}_{u_3}^{3,2}(v_2) & \text{Bs}_{u_4}^{3,3}(v_2) \\ \text{Bd}_{u_0}^{3,1}(v_0) & \text{Bd}_{u_1}^{3,2}(v_0) & \text{Bd}_{u_2}^{3,3}(v_0) & \text{Bd}_{u_3}^{3,4}(v_0) & 0 \\ 0 & 0 & \text{Bd}_{u_{k-2}}^{3,1}(v_{k-1}) & \text{Bd}_{u_{k-1}}^{3,2}(v_{k-1}) & \text{Bd}_{u_k}^{3,3}(v_{k-1}) \end{pmatrix}$$

Po dosadení a vypočítaní báзовých funkcií, pričom pre prehľadnosť sú získané hodnoty zaokrúhlené na dve desatinné miesta (a zapísané v tvare zlomkov) dostávame:

$$\mathbf{L} = \begin{pmatrix} \frac{134}{25} & \frac{1093}{100} & \frac{57}{20} & 0 & 0 \\ 0 & \frac{39}{20} & \frac{61}{5} & \frac{217}{50} & 0 \\ 0 & 0 & \frac{169}{50} & \frac{1309}{100} & \frac{271}{100} \\ -\frac{919}{100} & \frac{11}{20} & \frac{571}{100} & 0 & 0 \\ 0 & 0 & -\frac{27}{4} & \frac{41}{50} & \frac{13}{2} \end{pmatrix}$$

Pozornému oku iste neuniklo, že hodnoty funkcií $Bs_a^{3,4}(b)$ a $Bd_a^{3,4}(b)$, kde a je pomocný uzol a b je uzol sú v tomto príklade rovné nule. Hodnoty týchto funkcií sa totiž vždy pohybujú vo veľmi malom okolí čísla 0, takže po zaokrúhlení na dve desatinné miesta majú túto hodnotu. Nech vektor $(b_0, b_1, b_2, b_3, b_4)$ je riešením nasledujúcej sústavy rovníc:

$$\mathbf{L} \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ d_{\text{left}} \\ d_{\text{right}} \end{pmatrix} \quad (1.10)$$

Teda po dosadení máme:

$$\mathbf{L} \cdot \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ \frac{11}{10} \\ \frac{3}{4} \\ -2 \\ \frac{4}{5} \end{pmatrix} \quad (1.11)$$

Prvky vektora $(b_0, b_1, b_2, b_3, b_4)$, ktorý predstavuje kontrolné body pre B-splajn bude teda mať hodnoty (zaokrúhlené na dve desatinné miesta):

$$\left(\frac{29}{100}, -\frac{11}{50}, \frac{7}{50}, -\frac{3}{100}, \frac{27}{100} \right)$$

So získanými (pomocnými) uzlami U a prvkami vektora $(b_0, b_1, b_2, b_3, b_4)$ ako kontrolnými bodmi môžeme zostrojiť B-splajn stupňa 3 tak, aby krivka prechádzala zadanými funkčnými hodnotami C .

Na koniec teda môžeme pristúpiť k formálnej definícii globálneho B-splajnu.

Definícia 1.24 Nech k je prirodzené číslo, $V = \langle v_0, v_1, \dots, v_{k-1} \rangle$ sú uzly, $C = \langle c_0, c_1, \dots, c_{k-1} \rangle$ sú kontrolné body a $D = \langle d_{\text{left}}, d_{\text{right}} \rangle$ sú prvé derivácie v bodoch v_0 a v_{k-1} . Nech platí:

- $k \geq 2$,
- k -tica V je rastúca.

Definujme funkciu $\text{GBSpline}_{V,C,D}^k$ z intervalu $[v_0, v_{k-1}]$ do \mathbb{R} vzťahom:

$$\text{GBSpline}_{V,C,D}^k(x) = \text{BSpline}_{\text{SKnots}_{3,k}(V), \text{SCPoints}_{V, \text{SKnots}_{3,k}(C,D)}^k}(x) \quad (1.12)$$

1.3 Hermitove splajny

Narozdiel od B-splajnu predstavuje Hermitov splajn interpolačnú krivku. Hermitova krivka sa (pochopiteľne) skladá zo segmentov, kde každý z nich je definovaný dvomi bodmi a ich prvými deriváciami, ktoré zabezpečia spojitosť kriviek v jednotlivých segmentov.

1.3.1 Bázové funkcie

Podobne ako pri B-splajne aj tu sú potrebné bázové funkcie z ktorých vygenerujeme krivku. Tieto funkcie sú značne jednoduchšie než v prípade B-splajnu. V tomto prípade budeme potrebovať dvojicu funkcií.

Definícia 1.25 Nech a a b sú reálne čísla. Definujme funkciu $H_{s,a,b}$ z intervalu $[a, b]$ do \mathbb{R} vzťahom

$$H_{s,a,b}(x) = \left(1 - \frac{2 \cdot (x - a)}{a - b}\right) \cdot \frac{(x - a)^2}{(a - b)^2}.$$

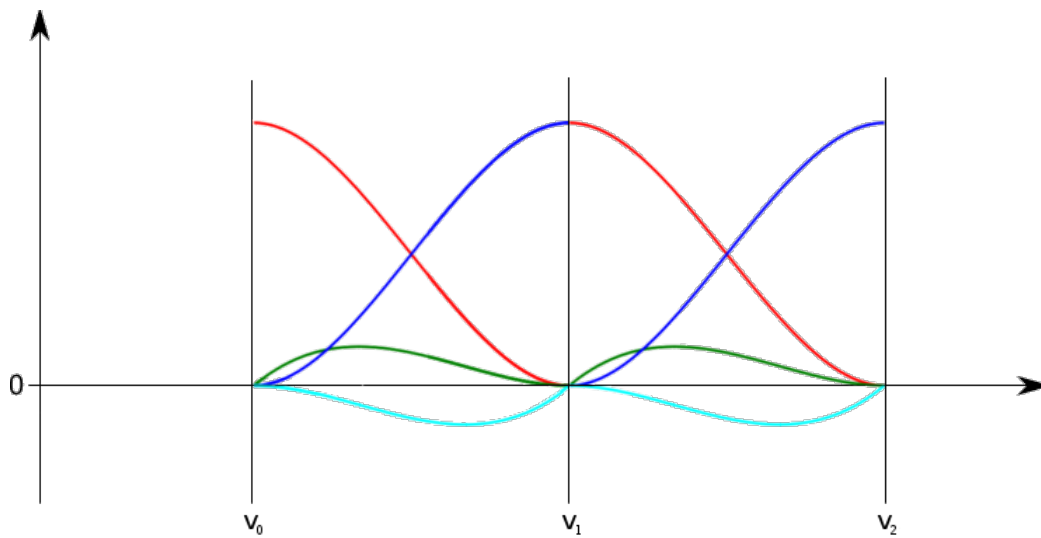
Definícia 1.26 Nech a a b sú reálne čísla. Definujme funkciu $H_{d,s,a,b}$ z intervalu $[a, b]$ do \mathbb{R} vzťahom

$$H_{d,s,a,b}(x) = \frac{(x - b)^2 \cdot (x - a)}{(b - a)^2}.$$

Poznámka 1.27 Môžeme si všimnúť, že bázové funkcie sú polynómy tretieho stupňa. Teda všetky hermitovské krivky budú práve stupňa 3.

Podobne ako v predchádzajúcich statiach aj tu si urobíme ilustračný príklad pre bázové funkcie dvojsegmentového uniformného Hermitovho splajnu.

Príklad 1.28 Nech $\langle v_0, v_1, v_2 \rangle = \langle 1, 2, 3 \rangle$ sú uzly. Keď Na jeden segment Hermitovho splajnu potrebujeme 4 bázové funkcie a 2 uzly. Keďže máme 3 uzly tak počet segmentov v tomto príklade bude 2, pričom pre konštrukciu bázových funkcií pre prvý segment použijeme uzly $v_0 = 1$ a $v_1 = 2$ a pre zostrojenie druhého segmentu použijeme uzly $v_1 = 2$ a $v_2 = 3$. Pozrime sa na ilustračný obrázok výsledku a nižšie si ho vysvetlíme.



Obr. 1.5: Bázové funkcie pre dvojsegmentovú uniformnú Hermitovu krivku

Na obrázku máme grafy ôsmich bázových funkcií pre dva segmenty Hermitovho splajnu. Grafy týchto funkcií majú nasledujúci zápis, pričom si ich kvôli prehľadnosti zapíšeme farebne podľa jednotlivých segmentov:

- 1. segment

- $Hs_{1,2}$
- $Hs_{2,1}$
- $Hds_{1,2}$
- $Hds_{2,1}$

- 2. segment

- $Hs_{2,3}$
- $Hs_{3,2}$
- $Hds_{2,3}$
- $Hds_{3,2}$

Keďže uvažujeme uniformný splajn, funkcií tej istej farby sú, až na posunutý definičný obor, identické, v prípade neuniformného splajnu by bázové funkcie v rôznych segmentoch mali rozličné grafy.

V tejto stati sme si definovali dvojicu funkcií, ktoré budú generovať priestor funkcií nad jedným segmentom Hermitovského splajnu. Výslednú funkciu dostaneme ako lineárnu kombináciu dvoch kontrolných bodov a dvoch prvých derivácií v príslušných uzloch so štyrmi bazovými funkciami. V nasledujúcej stati si ukážeme samotnú konštrukciu Hermitovskej krivky a jej formálnu definíciu.

1.3.2 Hermitova krivka

Zostrojenie Hermitovho splajnu je značne jednoduchšie než zostrojenie B-splajnu. Na jeden segment nám stačia dva uzly, ďalej funkčné hodnoty splajnu v týchto uzloch (teda kontrolné body) a ich prvé derivácie.

Konštrukcia Hermitovho splajnu Nech sú dané reálne čísla v_0, \dots, v_{k-1} – uzly, c_0, \dots, c_{k-1} – funkčné hodnoty žiadanej krivky v uzloch, a d_0, \dots, d_{k-1} – hodnoty derivácií splajnu v uzloch, pričom k je prirodzené číslo, také že $k \geq 2$.

Našou úlohou je zostrojiť Hermitov splajn stupňa intervalu $[v_0, v_{k-1}]$. Konštrukcia krivky bude prebiehať po častiach v k krokoch, pričom v prvom kroku použijeme uzly v_0 a v_1 , funkčné hodnoty c_0 a c_1 a derivácie d_0 a d_1 . Vďaka podmienke $k \geq 2$ máme zaručené, že aspoň jeden segment môžeme zkonštruovať. V prípade druhého segmentu použijeme uzly v_1 a v_2 , funkčné hodnoty c_1 a c_2 a derivácie d_1 a d_2 atď..

Rovnicu výslednej krivky získame vynásobením dvoch funkčných hodnôt a dvoch derivácií s bazovými funkciami. Označme P_1, \dots, P_{k-1} polynomicke funkcie, ktoré tvoria jednotlivé segmenty Hermitovho splajnu. Ich rovnice vyzerajú takto:

$$P_i(x) = c_{i-1} \cdot \text{Hs}_{v_{i-1}, v_i}(x) + c_i \cdot \text{Hs}_{v_i, v_{i-1}}(x) + d_{i-1} \cdot \text{Hds}_{v_{i-1}, v_i}(x) + d_i \cdot \text{Hds}_{v_i, v_{i-1}}(x). \quad (1.13)$$

Môžeme pristúpiť k formálnej definícii Hermitovho splajnu.

Definícia 1.29 Nech k je prirodzené číslo, $k \geq 2$, $V = \langle v_0, v_1, \dots, v_{k-1} \rangle$ sú uzly, $C = \langle c_0, c_1, \dots, c_{k-1} \rangle$ sú funkčné hodnoty a $D = \langle d_0, d_1, \dots, d_{k-1} \rangle$ sú prvé derivácie v uzloch. Definujme funkciu $\text{HSpline}_{V,C,D}^k$ z intervalu $[v_0, v_{k-1}]$ do \mathbb{R}

vzťahom:

$$\text{HSpline}_{V,C,D}^k(x) = \begin{cases} c_0 \cdot \text{Hs}_{v_0,v_1}(x) + c_1 \cdot \text{Hs}_{v_1,v_0}(x) & \text{pre } v_0 \leq x < v_1, \\ +d_0 \cdot \text{Hds}_{v_0,v_1}(x) + d_1 \cdot \text{Hds}_{v_1,v_0}(x) & \\ c_1 \cdot \text{Hs}_{v_1,v_2}(x) + c_2 \cdot \text{Hs}_{v_2,v_1}(x) & \text{pre } v_1 \leq x < v_2, \\ +d_1 \cdot \text{Hds}_{v_1,v_2}(x) + d_2 \cdot \text{Hds}_{v_2,v_1}(x) & \\ \vdots & \\ c_{k-2} \cdot \text{Hs}_{v_{k-2},v_{k-1}}(x) + c_{k-1} \cdot \text{Hs}_{v_{k-1},v_{k-2}}(x) & \text{pre } v_{k-2} \leq x \leq v_{k-1}. \\ +d_{k-2} \cdot \text{Hds}_{v_{k-2},v_{k-1}}(x) + d_{k-1} \cdot \text{Hds}_{v_{k-1},v_{k-2}}(x) & \end{cases} \quad (1.14)$$

Príklad 1.30 Uvažujme uzly a bázové funkcie z príkladu 1.28, teda uzly $V = \langle 1, 2, 2 \rangle$ a bázové funkcie:

- 1. segment

- $\text{Hs}_{1,2}$
- $\text{Hs}_{2,1}$
- $\text{Hds}_{1,2}$
- $\text{Hds}_{2,1}$

- 2. segment

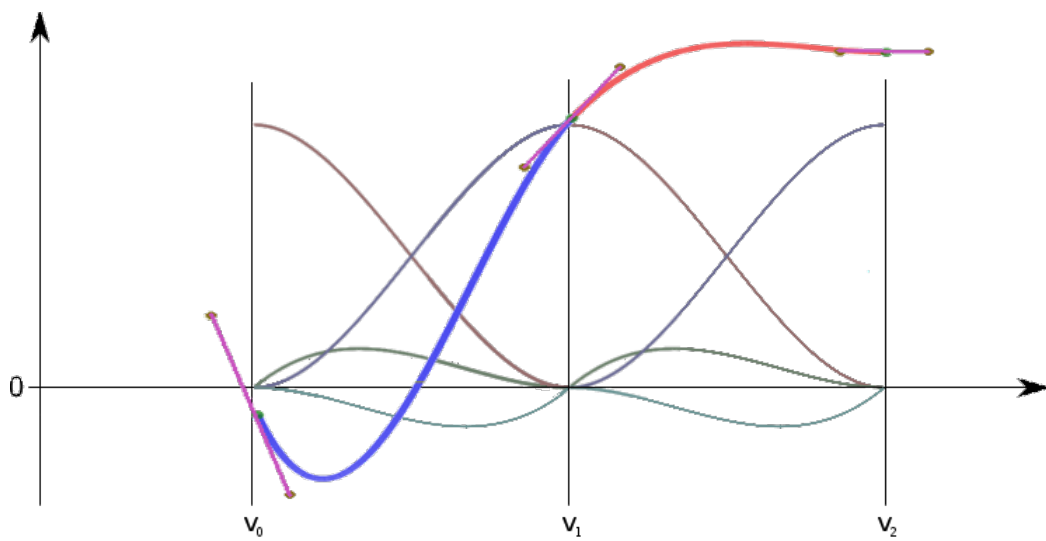
- $\text{Hs}_{2,3}$
- $\text{Hs}_{3,2}$
- $\text{Hds}_{2,3}$
- $\text{Hds}_{3,2}$

Nech $\langle c_0, c_1, c_2 \rangle = \langle -\frac{1}{10}, 1, \frac{11}{10} \rangle$ sú funkčné hodnoty žiadaného splajnu v uzloch u_0 , u_1 a u_2 a $\langle d_0, d_1, d_2 \rangle = \langle -2, 1, 0 \rangle$ sú funkčné hodnoty žiadaného splajnu v uzloch u_0 , u_1 a u_2 . Rovnice pre jednotlivé segmenty P_1 a P_2 budú podľa rovnice 1.13 vyzerat takto:

$$\begin{aligned} P_1(x) &= c_0 \cdot \text{Hs}_{v_0,v_1}(x) + c_1 \cdot \text{Hs}_{v_1,v_0}(x) \\ &\quad + d_0 \cdot \text{Hds}_{v_0,v_1}(x) + d_1 \cdot \text{Hds}_{v_1,v_0}(x), \\ P_2(x) &= c_1 \cdot \text{Hs}_{v_1,v_2}(x) + c_2 \cdot \text{Hs}_{v_2,v_1}(x) \\ &\quad + d_1 \cdot \text{Hds}_{v_1,v_2}(x) + d_2 \cdot \text{Hds}_{v_2,v_1}(x). \end{aligned}$$

Po dosadení hodnôt dostávame:

$$\begin{aligned}
P_1(x) &= -\frac{1}{10} \cdot \text{Hs}_{1,2}(x) + \text{Hs}_{2,1}(x) \\
&\quad - 2 \cdot \text{Hds}_{1,2}(x) + \text{Hds}_{2,1}(x), \\
P_2(x) &= \text{Hs}_{2,3}(x) + \frac{11}{10} \cdot \text{Hs}_{3,2}(x) \\
&\quad + \text{Hds}_{v_1,v_2}(x).
\end{aligned}$$



Obr. 1.6: Krivka v každom segmente (modrou alebo červenou) je lineárna kombinácia bázových funkcií, ružové úsečky znázorňujú smer derivácii v uzloch

Poznámka 1.31 V prípade Hermitovských kriviek máme síce zadané prvé derivácie v uzloch, čo zabezpečí spojitosť jednotlivých segmentov, ale nemáme zaručené rovnosti druhých derivácií segmentov v uzloch, inými slovami nemáme zaručenú aj hladkosť krivky v okolí uzlov. Jedná sa o takzvaný splajn triedy $C1$, na rozdiel od B-splajnov, pri ktorých sa druhé derivácie segmentov v uzloch rovnajú. V nasledujúcej kapitole prejdeme ku clamped splajnom, ktoré tento problém riešia.

1.4 Clamped splajny

Clamped splajn (vnorený splajn) je interpolačná a globálna krivka. Vychádza z Hermitovských splajnov s tým, že na rozdiel od nich nezadáme prvé derivácie pre každý uzol, ale len pre prvý a posledný uzol. Zbytok derivácii sa dodatočne dopočíta pomocou sústavy lineárnych rovníc. Z takto získaných derivácii vypočítame Hermitovský splajn. Zaujímavosťou je, že ak budeme brať do úvahy len uniformné splajny, tak že výsledná krivka je totožná z globálnym B-splajnom.

Existuje aj variant zvaný *naturálny splajn* v ktorom navyše predpokladáme, že druhé derivácie v uzloch u_0 a u_{k-1} sú rovné číslu 0.

Konštrukcia clamped splajnu (1) Nech k je z \mathbb{N} , $k \geq 2$, $V = \langle v_0, \dots, v_{k-1} \rangle$ sú vstupné uzly, hodnoty $C = \langle c_0, \dots, c_{k-1} \rangle$ sú funkčné hodnoty žiadaného clamped splajnu v uzloch, číslo d_0 je hodnota derivácie v bode u_0 a číslo d_{k-1} je hodnota derivácie v bode v_{k-1} . Položme číslo h vzťahom:

$$h = \frac{\sum_{i=1}^{k-1} (v_i - v_{i-1})}{k} \quad (1.15)$$

Označme maticu \mathbf{L} vzťahom:

$$\mathbf{L} = \begin{pmatrix} 4 & 1 & 0 & \dots & 0 & 0 \\ 1 & 4 & 1 & \dots & 0 & 0 \\ 0 & 1 & 4 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 4 & 1 \\ 0 & 0 & 0 & \dots & 1 & 4 \end{pmatrix} \quad (1.16)$$

Nech vektor (d_1, \dots, d_{k-2}) je riešením nasledujúcej sústavy rovníc:

$$\mathbf{L} \cdot \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{k-3} \\ d_{k-2} \end{pmatrix} = \begin{pmatrix} \frac{3 \cdot (c_2 - c_0)}{h} - d_0 \\ \frac{3 \cdot (c_3 - c_1)}{h} \\ \frac{3 \cdot (c_4 - c_2)}{h} \\ \vdots \\ \frac{3 \cdot (c_{k-2} - c_{k-4})}{h} \\ \frac{3 \cdot (c_{k-1} - c_{k-3})}{h} - d_{k-1} \end{pmatrix} \quad (1.17)$$

Hodnoty d_0, d_1, \dots, d_{k-1} predstavujú prvé derivácie hermitovského splajnu v uzloch v_0, v_1, \dots, v_{k-1} . Na základe zadaných uzlov, funkčných hodnôt splajnu

v uzloch, dvoch derivácií a následne získaných derivácií v uzloch tak vieme vygenerovať Hermitovu krivku.

Poznámka 1.32 Všimnime si, že sústava rovníc je trojdiagonálna. To znamená že ak máme k rovníc, tak ju vieme riešiť v čase $O(k)$ narozdiel od sústavy rovníc v prípade globálneho B-splajnu, ktorú vieme riešiť len v čase $O(k^3)$. Výpočet clamped splajnov je teda výrazne rýchlejší, než výpočet globálneho B-splajnu.

Formálne zapísaný postup bude mať teda takúto podobu.

Definícia 1.33 Nech k je z \mathbb{N} , $k \geq 2$, $V = \langle v_0, \dots, v_{k-1} \rangle$ sú vstupné uzly, hodnoty $C = \langle c_0, \dots, c_{k-1} \rangle$ sú funkčné hodnoty, číslo d_0 je hodnota derivácie v bode u_0 a číslo d_{k-1} je hodnota derivácie v bode v_{k-1} . Položme číslo h vzťahom:

$$h = \frac{\sum_{i=1}^{k-1} (v_i - v_{i-1})}{k}$$

Nech vektor $(d_1, \dots, d_{k-2})^T$ je riešením sústavy rovníc 1.17. Definujme funkciu CDer_V^k z \mathbb{R}^{k+2} do \mathbb{R}^k vzťahom:

$$\text{CDer}_V^k(c_0, \dots, c_{k-1}, d_0, d_{k-1}) = \langle d_0, d_1, \dots, d_{k-2}, d_{k-1} \rangle \quad (1.18)$$

Definícia 1.34 Nech k je prirodzené číslo, $k \geq 2$, $V = \langle v_0, v_1, \dots, v_{k-1} \rangle$ sú uzly, $C = \langle c_0, c_1, \dots, c_{k-1} \rangle$ sú funkčné hodnoty a $D = \langle d_0, d_{k-1} \rangle$ sú prvé derivácie v uzloch v_0 a v_{k-1} . Definujme funkciu $\text{CSpline}_{V,C,d_0,d_{k-1}}$ z intervalu $[v_0, v_{k-1}]$ do \mathbb{R} vzťahom:

$$\text{CSpline}_{V,C,d_0,d_{k-1}}^k(x) = \text{HSpline}_{V,C,\text{CDer}_V^k(C,d_0,d_{k-1})}^k(x) \quad (1.19)$$

Kapitola 2

Nový postup generovania clamped splajnov

Ako som spomínal na začiatku článku, cieľom práce je preskúmať a implementovať iný spôsob generovania globálneho B-splajnu použitím Hermitovských splajnov s ktorým pracuje doc. Török. Postup je podobný ako pri konštruovaní clamped splajnov, teda pre zadané uzly, funkčné hodnoty a dve derivácie vypočítať derivácie v každom uzle a vypočítať Hermitovskú krivku. Rozdiel oproti predchádzajúcemu postupu spočíva v rozdielnom postupe pri generovaní derivácií. Podobne ako pri clamped splajnoch, uniformné krivky konštruované týmto postupom sú identické s globálnym B-splajnom.

Pred tým než začneme s konštrukciou krivky si definujeme pomocnú funkciu, ktorú použijeme pri hľadaní prvých derivácií v uzloch.

2.1 Zostrojenie

Definícia 2.1 Definujme funkciu CDN_h z \mathbb{R}^4 do \mathbb{R} vzťahom

$$CDN_h(u_{\text{left}}, u_{\text{right}}, d_{\text{left}}, d_{\text{right}}) = \frac{\frac{3 \cdot (u_{\text{right}} - u_{\text{left}})}{h} - d_{\text{left}} - d_{\text{right}}}{4} \quad (2.1)$$

Konštrukcia clamped splajnu Nech k je z \mathbb{N} , $k \geq 2$ a hodnoty u_0, \dots, u_{k-1} sú vstupné uzly, hodnoty y_0, \dots, y_{k-1} sú funkčné hodnoty žiadaného clamped splajnu v uzloch, číslo d_0 je hodnota derivácie v bode u_0 a číslo d_{k-1} je hodnota derivácie v bode u_{k-1} .

Označme reálne číslo h vzťahom:

$$h = \frac{\sum_{i=1}^{k-1} (u_i - u_{i-1})}{k}$$

Konštrukcia clamped splajnu touto metódou sa líši podľa parametra k , teda podľa počtu zadanych uzlov a funkčných hodnôt. Rozlíšme jednotlivé prípady:

- Nech $k = 2$.

V tom prípade zostrojíme Hermitovský splajn podľa 1.29

- Nech $k = 3$.

Potom hodnota $d_1 = \text{CDN}_h(u_0, u_2, d_0, d_2)$ je derivácia v uzle u_1 a teda môžeme zostrojiť Hermitovský splajn podľa postupu 1.29.

- Nech $k = 4$.

Potom hodnota $d_2 = (d_0 + 3 \cdot u_0 - 3 \cdot u_2 - 12 \cdot u_1 + 12 \cdot u_3)/15$ je derivácia v uzle u_2 a hodnota $d_1 = \text{CDN}_h(u_0, u_2, d_0, d_2)$ je derivácia v uzle u_1 a teda môžeme zostrojiť Hermitovský splajn podľa 1.29.

- Nech $k \geq 5$.

Definujme premenné J , K , L a M vzťahmi:

$$\begin{aligned} - L &= \begin{cases} \frac{15}{2} & \text{ak } k \text{ je párne,} \\ 7 & \text{inak.} \end{cases} \\ - M &= \begin{cases} \frac{3}{2} + (k-5)/2 & \text{ak } k \text{ je párne,} \\ 1 + (k-5)/2 & \text{inak.} \end{cases} \\ - K &= \begin{cases} k-2 & \text{ak } k \text{ je párne,} \\ k-3 & \text{inak.} \end{cases} \end{aligned}$$

Hodnota K je index posledného párneho uzla, ktorý zároveň nie je posledný.

$$- J = \begin{cases} k-3 & \text{ak } k \text{ je párne,} \\ k-2 & \text{inak.} \end{cases}$$

Hodnota J je index posledného nepárneho uzla, ktorý zároveň nie je posledný.

Pri zostrojení clamped splajnu v predchádzajúcej kapitole sme vypočítali prvé derivácie v uzloch pomocou sústavy rovníc. Podobný prístup použijeme aj v tomto prípade s tým rozdielom, že tu vypočítame hodnoty len párnych derivácií (nezabudnime, že indexujeme od 0), t.j. hodnoty d_2, d_4, \dots, d_K a zvyšné dopočítame pomocou funkcie CDN_h .

Definujme maticu \mathbf{A} vzťahom:

$$\mathbf{A} = \begin{pmatrix} 7 & -\frac{1}{2} & 0 & \dots & 0 & 0 \\ -\frac{1}{2} & 7 & -\frac{1}{2} & \dots & 0 & 0 \\ 0 & -\frac{1}{2} & 7 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 7 & -\frac{1}{2} \\ 0 & 0 & 0 & \dots & -\frac{1}{2} & L \end{pmatrix}$$

Nech vektor $(d_2, d_4, d_6, \dots, d_K)$ je riešením nasledujúcej sústavy rovníc:

$$\mathbf{A} \cdot \begin{pmatrix} d_2 \\ d_4 \\ d_6 \\ \vdots \\ d_{K-2} \\ d_K \end{pmatrix} = \begin{pmatrix} \frac{3 \cdot (y_0 - y_4) - 12 \cdot (y_1 - y_3)}{2 \cdot h} \\ \frac{3 \cdot (y_2 - y_6) - 12 \cdot (y_3 - y_5)}{2 \cdot h} \\ \frac{3 \cdot (y_4 - y_8) - 12 \cdot (y_5 - y_7)}{2 \cdot h} \\ \vdots \\ \frac{3 \cdot (y_{2(M-3)} - y_{(2(M-3)+4)}) - 12 \cdot (y_{(2(M-3)+1)} - y_{(2(M-3)+3)})}{2 \cdot h} \\ \frac{3 \cdot (y_{2(M-2)} - y_{(2(M-2)+4)}) - 12 \cdot (y_{(2(M-2)+1)} - y_{(2(M-2)+3)})}{2 \cdot h} \end{pmatrix} \quad (2.2)$$

Ostáva ešte dopočítať hodnoty derivácií $d_1, d_3, d_5, \dots, d_J$. Nech pre každé i z $\{1, 3, 5, \dots, J\}$ platí:

$$d_i = \text{CDN}_h(u_{i-1}, u_{i+1}, d_{i-1}, d_{i+1}).$$

Hodnoty d_0, d_1, \dots, d_{k-1} predstavujú prvé derivácie hermitovského splajnu v uzloch u_0, u_1, \dots, u_{k-1} . Na základe zadaných uzlov, funkčných hodnôt splajnu v uzloch, dvoch derivácií a následne získaných derivácií v uzloch tak vieme vygenerovať Hermitovskú krivku na základe postupu 1.29.

Ako vidieť, rozdiel oproti predchádzajúcemu spôsobu spočíva v rozdielnom počítaní derivácií, kedy stále máme trojdiagonálnu sústavu rovníc, ale zredukovanú na polovičnú veľkosť, získali sme tým teda mierne zrýchlenie.

Časť II

Implementácia

V tejto časti, ako už názov napovedá prejdeme k implementácii navrhnutých postupov. Najprv si ale povieme niečo o technológii, v ktorej je práca implementovaná a v nasledujúcej kapitole si priblížime detaily implementovania aplikácie.

Kapitola 3

Microsoft Silverlight

Microsoft Silverlight (ďalej len Silverlight) je bezplatný nástroj na tvorbu webových aplikácií spustiteľných vo webovom prehliadači. V súčasnosti je k dispozícii vo verzii 5, pričom prvá verzia bola vydaná v roku 2007 ako konkurencia k technológii Adobe Flash. Samotný vývoj aplikácií môže prebiehať v ľubovoľnom interpretovanom programovacom jazyku bežiacom pod virtuálnym strojom Common Language Runtime, najčastejšie používané (a oficiálne podporované) sú však programovacie jazyky C# a Visual Basic. V našom prípade sme siahli po jazyku C#.



Silverlight je kompatibilný s väčšinou moderných webových prehliadačov a operačných systémov vrátane Microsoft Windows, Apple OS X a vďaka technológii Mono – open-source implementácii platformy .NET, je možné Silverlight aplikácie používať aj na väčšine Linuxových distribúcií. Framework je možné použiť aj na vývoj off-line aplikácií v operačnom systéme Windows Phone, Windows 8 a Windows RT.

Oficiálne vývojové prostredie (skrátene IDE) je Microsoft Visual Studio, ktoré existuje vo viacerých variantách. Najpoužívanejší je variant Express, ktorý je dostupný zdarma avšak prichádza s mnohými obmedzeniami (napríklad nepodpora prídavných modulov) a nie je vhodný na tvorbu rozsiahlych projektov. Pri vývoji aplikačnej časti tejto práce bol použitý plnohodnotný variant Professional.

3.1 Architektúra

V tejto stati si si ukážeme základ architektúry Silverlight-u. Ten sa skladá z dvoch hlavných súčastí, *Core presentation framework*, *.NET Framework for Silverlight*. Ich funkcie si v krátkosti zhrnieme.

Core presentation framework Obsahuje komponenty a služby pre užívateľské rozhranie, nástroje pre vektorovú grafiku, animácie a podobne. Tak tiež obsahuje jazyk *Extensible Application Markup Language* (skrát. *XAML*) na definovanie grafického používateľského rozhrania. Účel jazyka XAML si vysvetlíme neskôr.

.NET Framework for Silverlight Jedná sa o podmnožinu .NET Framework zabezpečujúca fyzický beh aplikácie na počítači. Obsahuje nástroje na správu pamäte (garbage collection), rôzne knižnice a v neposlednom rade kompilátory .NET kompatibilných jazykov, ako napr. Visual Basic, C#, JavaScript do CLR kódu a následne do strojového kódu.

3.1.1 XAML

Extensible Application Markup Language (XAML) je deklaratívny jazyk, ktorého účel a syntax sú podobné jazykom XML a HTML. Účelom XAML v Silverlight je deklarovať prvky grafického používateľského rozhrania (GUI) aplikácie, ktoré sa zobrazia vo webovom prehliadači. Samotná logika aplikácie je naprogramovaná v nejakom CLR jazyku, v našom prípade v jazyku C#.

Ukážka XAML kódu:

```
<UserControl x:Class="MySilverlight.Page"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  >
  <!--Prvky rozhrania našej aplikácie-->
  <Canvas Background="White" Height="600" Width="800">
    <!--Prvky rozhrania v rámci Canvas-->
  </Canvas>
</UserControl>
```

3.2 Grafika v Silverlight

Silverlight prichádza s pomerne pokročilými nástrojmi na tvorbu dvojrozmernej grafiky, pričom použitím vhodných doplnkov je možné tvoriť aj grafiku v

troch rozmeroch. V tejto kapitole si stručne rozoberieme možnosti a nástroje, ktoré nám Silverlight ponúka. Podrobné detaily implementácie si predstavíme v kapitole 3.

Na zobrazenie ľubovoľných objektov v rovine slúži komponent zvaný *Canvas* (plátno). Ten má mnoho parametrov na prispôsobenie vizuálnej stránky a správania sa plátna, včítane objektov ktoré obsahuje. Nás však teraz budú zaujímať dva parametre: *Width* (ďalej len *šírka*) a *Height* (ďalej len *výška*). Tieto parametre, ako to vyplýva z ich názvu, slúžia na nastavenie rozmerov plátna v pixeloch obrazovky počítača.

Poznámka 3.1 Canvas má ďalší veľmi dôležitý parameter, a to je *Children* (deti). Jedná sa o kolekciu referencií na všetky objekty, ktoré sú vykreslené na plátne.

3.2.1 Transformácia súradníc

Každý objekt na plátne má svoju pozíciu určenú v ortogonálnej súradnicovej sústave, kde počiatočný bod sa nachádza v „ľavom-hornom“ rohu plátna, jedná sa teda o ľavotočivú sústavu. Túto súradnicovú sústavu budeme označovať pojmom *PlotArea*. Pre užívateľov je však značne pohodlnejšie používať súradnicovú sústavu na ktorú su roky zvyknutí, a síce ortogonálnu súradnicovú sústavu takú, kde počiatočný bod leží v priesečníku uhlopriečok obdĺžnikového plátna. Túto súradnicovú sústavu budeme označovať pojmom *WorldArea*. Teda užívateľské vstupy, podobne aj výstupy programu transformovať je potrebné prepočítavať medzi dvoma súradnicovými sústavami.

Takže vieme, že Canvas v Silverlight-e používa ortogonálnu sústavu. Hranice plátna sú určené dvomi bodmi, ktoré označujú počiatočný bod v „ľavom-hornom“ rohu plátna a maximálny bod v „pravom-dolnom“ rohu plátna.

Väčšinou platí, že počiatočný bod má hodnotu $\langle 0, 0 \rangle$ a maximálny bod má hodnotu $\langle w, h \rangle$, kde w a h sú kladné čísla. Čísla w a h teda určujú veľkosť plátna v pixeloch, pravda iba vtedy ak počiatočný bod má hodnotu $\langle 0, 0 \rangle$.

Označme si konštanty, ktoré určujú hranice plátna v ortogonálnej a v karteziánskej sústave.

- Súradnicu na osi x počiatočného bodu plátna v *PlotArea* označíme pax_{\min} .
- Súradnicu na osi y počiatočného bodu plátna v *PlotArea* označíme pay_{\min} .
- Súradnicu na osi x maximálneho bodu plátna v *PlotArea* označíme pax_{\max} .

- Súradnicu na osi y maximálneho bodu plátna v PlotArea označíme pax_{\max} .
- Najmenšiu súradnicu na osi x plátna vo WorldArea označíme wax_{\min} .
- Najmenšiu súradnicu na osi y plátna v WorldArea označíme way_{\min} .
- Najväčšiu súradnicu na osi x plátna v WorldArea označíme wax_{\max} .
- Najväčšiu súradnicu na osi y plátna v WorldArea označíme way_{\max} .

Usporiadaná dvojica $\langle \text{pax}_{\min}, \text{pay}_{\min} \rangle$ teda predstavuje hodnotu počiatočného bodu v PlotArea sústave. Ako bolo spomenuté, tento bod má väčšinou hodnotu $\langle 0, 0 \rangle$.

Transformačné vzťahy

Teraz si zavedme vzťahy pre transformáciu súradníc medzi dvoma súradnicovými sústavami plátna.

Transformácia z WorldArea do PlotArea Nech x a y sú ľubovoľné reálne čísla a usporiadaná dvojica $\langle x, y \rangle$ je bod vo WorldArea sústave. Ak u a v sú reálne čísla také, že platí:

$$u = \text{pax}_{\min} + \frac{(x - \text{wax}_{\min}) \cdot (\text{pax}_{\max} - \text{pax}_{\min})}{\text{wax}_{\max} - \text{wax}_{\min}}, \quad (3.1)$$

$$v = \text{pay}_{\max} - \frac{(y - \text{way}_{\min}) \cdot (\text{pay}_{\max} - \text{pay}_{\min})}{\text{way}_{\max} - \text{way}_{\min}}, \quad (3.2)$$

tak usporiadaná dvojica $\langle u, v \rangle$ predstavuje súradnice bodu $\langle x, y \rangle$ v PlotArea sústave.

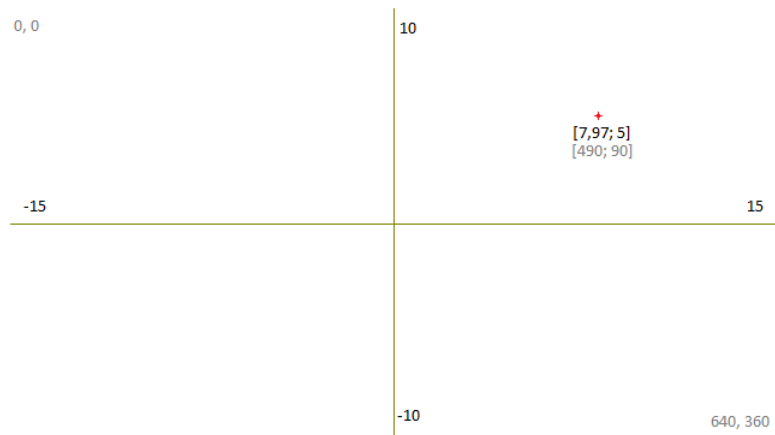
Transformácia z PlotArea do WorldArea Nech u a v sú ľubovoľné reálne čísla a usporiadaná dvojica $\langle u, v \rangle$ je bod v PlotArea sústave. Ak x a y sú reálne čísla také, že platí:

$$x = \text{wax}_{\min} + \frac{(u - \text{pax}_{\min}) \cdot (\text{wax}_{\max} - \text{wax}_{\min})}{\text{pax}_{\max} - \text{pax}_{\min}}, \quad (3.3)$$

$$y = \text{way}_{\min} - \frac{(v - \text{pay}_{\max}) \cdot (\text{way}_{\max} - \text{way}_{\min})}{\text{pay}_{\max} - \text{pay}_{\min}}, \quad (3.4)$$

tak usporiadaná dvojica $\langle x, y \rangle$ predstavuje súradnice bodu $\langle u, v \rangle$ vo WorldArea sústave.

Pre lepšiu predstavu uvedieme ilustračný obrázok.



Obr. 3.7: Bod (červenou) a jeho súradnice vo WorldArea (čiernou) a v PlotArea (sivou)

Na obrázku máme príklad plátna s počiatočným bodom $\langle 0, 0 \rangle$ a maximálnym bodom $\langle 640, 360 \rangle$, pričom súradnice oboch bodov sú v PlotArea súradnicovej sústave. Teda $pax_{\min} = 0$, $pay_{\min} = 0$, $pax_{\max} = 640$ a $pay_{\max} = 360$. Hranice plátna vo WorldArea sústave sú v tomto prípade $wax_{\min} = -15$, $way_{\min} = -10$, $wax_{\max} = 15$ a $way_{\max} = 10$. Užívateľom zadany bod (na obrázku červenou) s WorldArea súradnicami $\langle 7, 97, 5 \rangle$ teda bude mať po transformácii v plátne súradnice $\langle 490, 90 \rangle$.

Kapitola 4

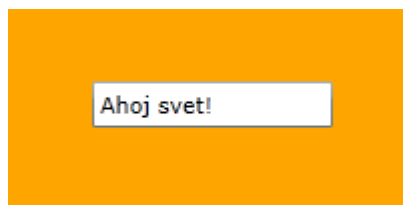
Vývoj aplikácie

V tejto kapitole si pomerne podrobne predstavíme implementačnú časť práce a síce webovú aplikáciu vyvinutú vo frameworku Silverlight. Prejdeme si implementovanie všetkých postupov pri zostrojovaní splajnov v prvej časti práce.

4.1 Vytvorenie aplikácie

Ako bolo spomemuté v minulej kapitole, grafické užívateľské rozhranie (GUI) je zadeklarované v jazyku XAML, pričom logika programu je naprogramovaná v jednom z CLR jazykov. Vytvorme ukážkovú aplikáciu s názvom *AhojSvet*, ktorá nám vypíše reťazec „*Ahoj svet!*“.

Zdrojové kódy sa implicitne nachádzajú v dvoch súboroch. Jeden sa nazýva *MainPage.xaml* a obsahuje deklaráciu GUI v XAML. Druhý súbor – *MainPage.xaml.cs* obsahuje logiku programu, v tomto prípade, v jazyku C#. Pozrime sa na obrázok našej aplikácie a nižšie si ukážeme, ako sme k takému výsledku dospeli.



Obr. 4.8: Triviálna aplikácia *AhojSvet*

Deklarácia GUI v súbore MainPage.xaml vyzerá takto.

```
<!-- Deklarácia aplikácie, ako inštancie triedy UserControl -->
<UserControl
  x:Class="AhojSvet.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
>
  <!-- Deklarácia plátna (Canvas) s názvom OrangeCanvas -->
  <Canvas x:Name="OrangeCanvas"
    <!-- Nastavenie pozadia na oranžovú
      a rozmerov plátna v pixeloch-->
    Background="Orange" Width="200" Height="100">

    <!-- Deklarácia textového poľa -->
    <TextBox Height="23"
      TextWrapping="Wrap"
      <!-- Nastavenie zobrazovaného textu -->
      Text="Ahoj svet!"
      <!-- Poloha v plátna vzhľadom na počiatočný bod
        <0,0> -->
      Canvas.Left="42" Canvas.Top="36"
      Width="120"/>

    <!-- Všetko medzi deklaráciou plátna
      a nasledujúcim riadkom je v plátna-->
  </Canvas>

</UserControl>
```

Pridružený súbor MainPage.xaml.cs bude obsahovať tento kód.

```
using System.Windows.Controls;

namespace AhojSvet
{
  // Celá naša aplikácia je rozšírením triedy UserControl
  public partial class MainPage : UserControl
  {
    public MainPage()
    {
      // Táto metóda inicializuje všetky objekty deklarované
      v XAML
      InitializeComponent();
    }
  }
}
```

```

    }
}
}

```

Zdrojový kód v C# toho pravda veľa neobsahuje, keďže sme našej aplikácii nenaprogramovali žiadnu funkcionálnosť.

4.2 Definície splajnov ako dátových typov

Keďže jazyk C# je objektovo orientovaný a Silverlight tento princíp dodržiava, je potrebné aby sme zvolili vhodnú dátovú štruktúru v ktorej si budeme ukladať referencie na zostrojené splajny.

4.2.1 Spline

Ako v prvej časti práce aj tu si zavedieme všeobecnú triedu *Spline*, ktorej atribúty budú spoločné pre všetky ostatné splajny, s tým, že tieto budú dediť práve od triedy *Spline*.

```

public abstract class Spline
{
    public List<Ellipse> DragEllipses { get; set; }
    public List<Line> LinesOfSpline { get; set; }
    public List<double> ControlPoints { get; set; }
    public List<double> Knots { get; set; }

    ...
}

```

Prejdime si jednotlivé atribúty, ktoré budú spoločné pre všetky typy splajnov:

- *Knots* Uzly splajnu.
- *ControlPoints* Kontrolné body splajnu.
- *LinesOfSpline* Obsahuje samotnú krivku tvorenú úsečkami.
- *DragEllipses* Predstavuje vizuálnu reprezentáciu uzlov a kontrolných bodov na plátne.

Poznámka 4.1 Modifikátor *abstract* znamená, že táto trieda je abstraktná, teda nie je možné vytvárať inštancie tejto triedy, iba inštancie dcérskych tried, v našom prípade tried *BSpline*, *GlobalBSpline*, atď..

4.2.2 BSpline

Trieda *BSpline* bude reprezentovať B-splajn a bude dediť od triedy *Spline*, pričom navyše obsahuje len stupeň splajnu.

```
public class GlobalBSpline : BSpline
{
    public int Degree { get; set; }

    ... // konštruktory
}
```

4.2.3 GlobalBSpline

Trieda *GlobalBSpline* bude reprezentovať globálny B-splajn a bude dediť od triedy *BSpline*.

Keďže dedíme od triedy *BSpline*, tak *GlobalBSpline* obsahuje všetky jej atribúty pričom navyše obsahuje:

- *FunctionValues* Užívateľom zadané funkčné hodnoty B-splajnu.
- *LeftDerivation* Prvá derivácia v uzle u_0 .
- *LeftDerivationEllipse* Predstavuje vizuálnu reprezentáciu atribútu *LeftDerivation*.
- *RightDerivationEllipse* Predstavuje vizuálnu reprezentáciu atribútu *RightDerivation*.

Implementácia vyzerá nasledovne:

```
public class GlobalBSpline : BSpline
{
    public List<double> FunctionValues { get; set; }
    public double LeftDerivation { get; set; }
    public double RightDerivation { get; set; }
    public Ellipse LeftDerivationEllipse { get; set; }
    public Ellipse RightDerivationEllipse { get; set; }

    ... // konštruktory
}
```

4.2.4 HermiteSpline

Trieda *HermiteSpline* bude reprezentovať Hermitov splajn a bude dediť od triedy *Spline*, pričom navyše obsahuje tieto atribúty:

- *Derivations* Užívateľom zadané prvé derivácie v uzloch.
- *DerivationEllipses* Vizuálna podoba derivácii na plátne Canvas.

Implementácia vyzerá nasledovne:

```
public class HermiteSpline : Spline
{
    public List<double> Derivations {get; set; }
    public List<Ellipse> DerivationEllipses {get; set; }

    ... // konštruktory
}
```

4.2.5 ClampedSpline

Nakoniec trieda *ClampedSpline* bude reprezentovať Hermitov splajn a bude dediť od triedy *Spline*, pričom navyše obsahuje tieto atribúty:

- *Derivations* Užívateľom zadané prvé derivácie v uzloch.
- *LeftDerivationEllipse* Vizuálna podoba prvej derivácie v prvom uzle na plátne Canvas.
- *RightDerivationEllipse* Vizuálna podoba prvej derivácie v poslednom uzle na plátne Canvas.

Implementácia vyzerá nasledovne:

```
public class ClampedSpline : Spline
{
    public List<double> Derivations { get; set; }
    public Ellipse LeftDerivationEllipse { get; set; }
    public Ellipse RightDerivationEllipse { get; set; }

    ... // konštruktory
}
```

Runge	Bell	Global B-spline
Basis	B-Spline	
H-Spline	C-Spline	

Function values:

3 2 -1 0,5

Interval:

Derivations:

☐ Regular curve

Obr. 4.9: Zadávací panel pre clamped splajn.

4.3 Generovanie a vykreslenie splajnov

Dátové štruktúry pre splajny máme definované, ale stále metódy, ktoré nám z užívateľom zadaných hodnôt vypočítajú a vykreslia žiadanú krivku.

Na obrázku vyššie vidíme zadávací panel pre clamped splajn. Na základe zadaných hodnôt v textových poliach sa po stlačení tlačidla *Draw* zavolá metóda, ktorá vypočíta a vykreslí clamped splajn, pričom nám vráti referenciu na objekt triedy *ClampedSpline*, s ktorými môžeme manipulovať, meniť hodnoty riadacích bodov, derivácie a podobne.

V nasledujúcom texte si ukážeme vygenerovanie clamped novou metódou splajnu krok po kroku. Vykreslenie iných kriviek prebieha analogickým spôsobom, postupy sa líšia pri počítaní básových funkcií, alebo pri počítaní derivácií a podobne.

Vykreslenie clamped splajnu Po tom čo užívateľ zadá vstup, ako na obrázku vyššie, je zavolaná metóda *InteractiveClampedSpline* ktorej úlohou je vykresliť užívateľsky interaktívnu clamped krivku. Jej stručná a skrátaná, implementácia vyzerá takto:


```
public ClampedSpline InteractiveClampedSpline(double[] knots,
    double[] controlPoints, double leftDerivation, double
    rightDerivation){
    ...
}
```

Metóda *InteractiveClampedSpline* dostane na vstup pole zadaných uzlov (atribút *knots*), pole zadaných funkčných hodnôt žiadaného splajnu v uzloch (atribút *controlPoints*) a prvé derivácie v prvom a poslednom uzle (atribúty *leftDerivation* a *rightDerivation*). V tele metódy nasleduje deklarovanie novej inštalácie triedy *ClampedSplajn*.

```
var clampedSpline = new ClampedSpline();
```

Vykreslenie uzlov a kontrolných bodov Než začneme s konštrukciou krivky, najprv vykreslíme samotné body, ktorými je splajn definovaný. Tieto budú vykreslené v podobe malých elíps v Silverlight-e reprezentovaných triedou *Ellipse*. Potrebujeme toľko elíps, koľko máme uzlov (a kontrolných bodov), teda ich uložíme ako zoznam (*List*).

```
var ellipses = new List<Ellipse>(knots.Length);
```

Taktiež si zadeklarujeme elipsy, ktoré budú reprezentovať zadané derivácie.

```
Ellipse leftDerEll, rightDerEll;
```

Ako už vieme, Canvas pracuje so súradnicovou sústavou, ktorú sme označili *PlotArea*. Všetky užívateľom zadané vstupy sú ale v *prírodzenej* súradnicovej sústave, ktorú sme označili pojmom *WorldArea*. Zadeklarujeme si pomocné premenné *pax* a *pay*, ktorých hodnoty budú predstavovať hodnoty uzlov a kontrolných bodov v *PlotArea* sústave.

```
double pax, pay;
```

Následne pre každý uzol a kontrolný bod vypočítame jeho *PlotArea* súradnice, vytvoríme novú elipsu s týmito súradnicami a pridáme ju do zoznamu elíps *ellipses*.

```
for (int i = 0; i < knots.Length; i++)
{
    // vypočítanie x-ovej súradnice
    pax = TransformCoordinates.WorldAreaToPlotAreaX(knots[i],
        _plotArea, _worldArea);

    // vypočítanie y-ovej súradnice
    pay = TransformCoordinates.WorldAreaToPlotAreaY(controlPoints[i],
        _plotArea, _worldArea);
```

```

// vytvorenie novej elipsy so zelenou farbou a pridanie do nášho
// zoznamu
ellipses.Add(DrawEllipse(pax, pay, COLOR_GREEN));
}

```

Poznámka 4.2 Metóda *DrawEllipse* vykreslí do Canvas, novú elipsu a vráti jej referenciu.

Potrebuje vypočítať elipsy pre derivácie. Vypočítajme súradnice elipsy pre deriváciu v bode $knots[0]$. V prípade druhej elipsy je postup analogický. Súradnice pre tieto elipsy nemáme užívateľom zadané, preto ich vypočítame tak, aby smernica spojnice medzi súradnicami elipsy a bodom $\langle knots[0], controlPoints[0] \rangle$ mala hodnotu *leftDerivation*.

x -ová súradnica *leftDerEll* môže mať ľubovoľnú hodnotu. My sme si zvolili $knots[0] - 0.15$.

```

var wax = knots[0] - DERIVATIVE_AUX_KNOT_DISTANCE; //
DERIVATIVE_AUX_KNOT_DISTANCE = 0.15

```

Keďže hľadaná spojnicu s môžeme zapísať v tvare lineárnej funkcie $s(x) = (x - x_0) \cdot k + y_0$, kde $\langle x_0, y_0 \rangle$ je známy bod spojnice s (v našom prípade $\langle knots[0], controlPoints[0] \rangle$), k je smernica spojnice (v našom prípade hodnota *leftDerivation*), tak hľadaná súradnica bude mať nasledujúcu hodnotu:

```

var way = (wax - knots[0]) * leftDerivation + controlPoints[0];

```

Následuje vyrátanie a vykreslenie elipsy *leftDerEll*.

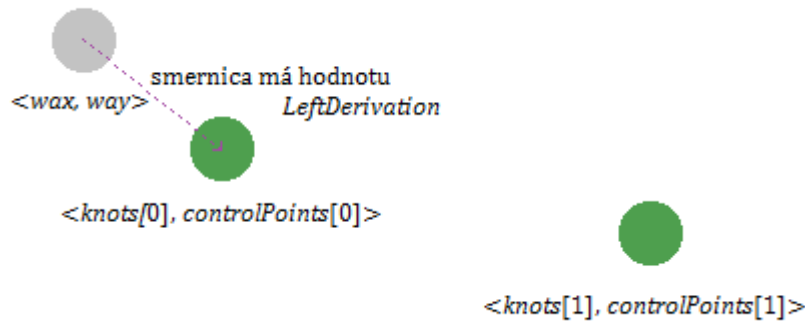
```

pax = TransformCoordinates.WorldAreaToPlotAreaX(wax, _plotArea,
    _worldArea);
pay = TransformCoordinates.WorldAreaToPlotAreaY(pax, _plotArea,
    _worldArea);
leftDerEll = _auxiliaryEngine.DrawEllipse(pax - W_DIV_2, pay -
    H_DIV_2, COLOR_GREY);

```

Poznámka 4.3 Tento prístup vizualizácie derivácie má dôvod pri interaktivite krivky. Aby užívateľ mohol interaktívne meniť hodnotu derivácie v uzle stačí aby kurzorom myši „ťahal“ za elipsu, pričom zo zmeny jej súradníc sa vyráta nová hodnota smernice spojnice medzi súradnicami ťahanej elipsy a bodom $\langle knots[0], controlPoints[0] \rangle$, pričom táto smernica predstavuje novú hodnotu prvej derivácie v danom uzle. Následne tak budeme počas doby za ktorú užívateľ ťahá elipsu celú prekreslovať niekoľko desiatok krát za sekundu, čím vznikne dojem interaktivity.

Pre lepšiu predstavu sa pozrime na ilustračný obrázok.



Obr. 4.10: Zelené elipsy predstavujú uzly a funkčné hodnoty splajnu v Canvas, sivá predstavuje deriváciu.

```
pax = TransformCoordinates.WorldAreaToPlotAreaX(wax, _plotArea,
    _worldArea);
pay = TransformCoordinates.WorldAreaToPlotAreaY(pax, _plotArea,
    _worldArea);
leftDerEll = _auxiliaryEngine.DrawEllipse(pax - W_DIV_2, pay -
    H_DIV_2, COLOR_GREY);
```

Uzly, kontrolné body a derivácie už máme vizualizované, na rade je vypočítanie samotnej krivky. Najprv vyrátame prvé derivácie pre zvyšné uzly podľa postupu v stati 2.1 a potom vykreslíme krivku.

```
double[] derivations = ClampedDerivations(knots);
```

Postup v stati 2.1 rozlišoval prípad pre dvojsegmentové (3 uzly), trojsegmentové (4 uzly), päťsegmentové (5 uzlov) a $k - 1$ -segmentové splajny (k uzlov). Implementáciu tohto postupu v metóde (ClampedDerivations) preskočíme, tá nám je známa zo stati 2.1.

Ostáva nám už iba vykreslenie krivky. Vieme, že clamped splajn je modifikovaný Hermitov splajn. Silverlight nemá vstavané funkcie na vykreslenie Hermitovej krivky, preto si musíme pomôcť sami. Algoritmus vykreslenia krivky môžeme zhrnúť takto. Nech interval $[a, b]$ je interval, nad ktorým má byť vykreslená krivka. Položme kladnú konštantu r blízku číslu 0 (napr. $d = 0.01$) a premenné $x = a$, y , $x_p = a$, y_p . Potom:

1. Vypočítaj funkčnú hodnotu splajnu v bode x_p a prirad' ju do y_p .
2. Zvýš hodnotu x o hodnotu d .

3. Ak $x \leq b$ prejdí na krok 4, inak prejdí na krok 7.
4. Vypočítaj funkčnú hodnotu splajnu v bode x a prirad' ju do y .
5. Vykresli úsečku $\langle x_p, y_p \rangle \langle x, y \rangle$
6. Prejdí na krok 3.
7. Vypočítaj funkčnú hodnotu splajnu v bode b a prirad' ju do y .
8. Vykresli úsečku $\langle x_p, y_p \rangle \langle b, y \rangle$

V prípade Clamped splajnu implementácia bude vyzerat' nasledovne.

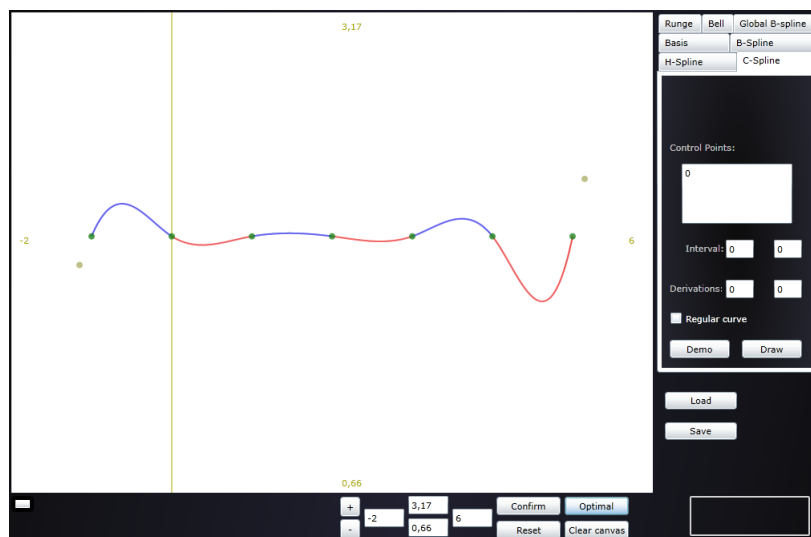
```
var curve = DrawHermiteCurve(knots, controlPoints, derivations);
```

Metóda *DrawHermiteCurve* vykreslí Hermitovu krivku ako kolekciu úsečiek podľa vyššie uvedeného algoritmu pričom sa budú vypočítavat' funkčné hodnoty funkcie $HSpline_{knots, controlPoints, derivations}^{knots.Length}(x)$. Celý postup vykresľovania sa dá s úpravami aplikovat' na všetky nami používané splajny.

Časť III

Záver

Ciele tejto práce sme spolu so školiteľom splnili, nový postup sa ukázal funkčný a rýchlejší ako predchádzajúce spôsoby. V súčasnosti je webová aplikácia takmer hotová, implementovali sme všetky postupy z tejto práce pričom sa doladujú rôzne detaily a vylepšuje ovládanie užívateľského rozhrania. Predpokladáme, že finálna verzia by mala byť hotová v priebehu mája 2014.



Obr. 4.11: Ukážka z aplikácie

Zoznam použitej literatúry

- [1] David Salomon, Curves and Surfaces for Computer Graphics, Springer, 2006
- [2] M. MacDonald, Pro Silverlight 5 in C#, press, 2012
- [3] <http://www.brnt.eu/phd/node11.html>
- [4] <http://msdn.microsoft.com/en-us/library/bb404713%28v=vs.95%29.aspx>