

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Použití L-systémů pro křivky počítané rekurzivním dělením**

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne

.....  
Michal Campr

## **Abstrakt**

Bakalářská práce se zabývá problematikou využívání L-systémů pro křivky počítané dělením řídicího polygonu, modelováním fyzikálních jevů v L-systémech a aproximací B-spline křivky pomocí Bézierovy křivky.

## **Abstrakt**

Bachelor thesis deals with problems connected with the use of L-systems for creating subdivision curves, with simulating physical effects in L-systems and with approximation of B-spline curve with the use of Bézier curve.

# Obsah

1 Úvod.....	5
2 Křivky počítané dělením řídicího polygonu.....	6
3 L-systémy.....	8
3.1 Základy L-systémů.....	8
3.2 Křivky počítané dělením pomocí L-systémů.....	10
3.3 Fyzika v L-systémech.....	12
4 Bézierovy a B-spline křivky.....	13
4.1 Bézierova křivka.....	13
4.1.1 Bézierova kubika.....	14
4.2 B-spline křivky.....	15
4.2.1 Coonsova kubika.....	15
4.2.2 Uniformní kubický B-spline.....	16
5 Aproximace uniformního kubického B-spline pomocí Bézierovy křivky.....	17
6 Aproximace Bézierovy křivky pomocí L-systémů.....	19
6.1 Aproximace Bézierovy kubiky.....	19
7 Větvení křivky.....	19
7.1 Větvení po segmentech.....	20
7.2 Větvení v rámci dělení.....	21
8 Implementace.....	22
8.1 Řídící body .....	23
8.1.1 Ruční zadávání.....	23
8.1.2 Využití L-systémů.....	23
8.2 Aplikace Bézierovy křivky vypočtené pomocí L-systémů.....	24
8.2.1 Bézierova kubika jako segment křivky.....	24
8.2.2 Globální dělení.....	24
8.3 Možnosti začlenění fyziky.....	24
8.3.1 Globální vlivy.....	24
8.3.2 Lokální vlivy.....	24
9 Závěr.....	24
Literatura.....	25

# 1 Úvod

Tato práce má za úkol shrnout některé poznatky z oblasti generování křivek ve dvojrozměrném prostoru - konkrétně jsou to křivky počítané dělením řídicího polygonu – a z oblasti využití L- systémů právě ve spojení s generováním křivek. Dalším tématem práce je vyzkoušet algoritmus pro aproximaci B-spline křivky pomocí Bézierovy křivky a prozkoumat jeho možnosti ve spojení s křivkami generovanými pomocí L- systémů.

Celá práce je rozdělena na dvě části: teoretickou a praktickou.

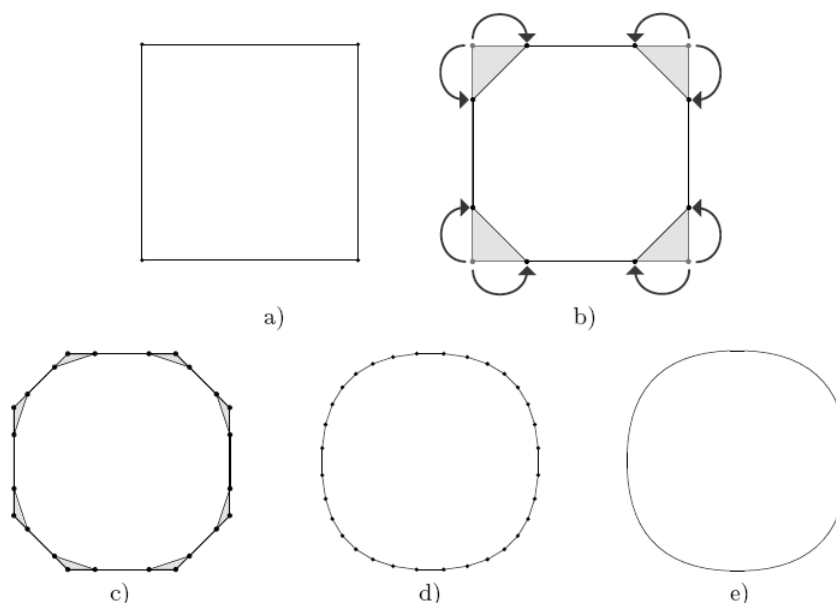
Teoretická část má objasnit základní pojmy týkající se generování křivek, dále vysvětlit základy L-systémů a popsat některé základní křivky a jejich vlastnosti. Cílem je tedy objasnit teoretické pojmy, jež budou využity v praktické části. Tato část se skládá z kapitol 2, 3 a 4.

Praktická část pojednává o implementaci problémů, kterými se tato práce zabývá. Především je to generování křivek pomocí L-systémů a možnosti modifikací použitého algoritmu tak, aby bylo možné tyto křivky větvit a také na ně aplikovat změny v závislosti na vnějším prostředí, tedy fyzikální vlivy. Praktická část obsahuje kapitoly 5 až 8.

## 2 Křivky počítané dělením řídicího polygonu

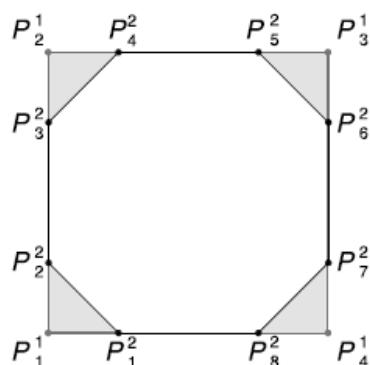
Algoritmům pro generování křivek pomocí dělení řídicího polygonu se v posledních letech věnuje stále více pozornosti. V souvislosti s rozvojem grafických procesorů se totiž dostává do popředí reprezentace geometrických objektů pomocí mnohoúhelníků – polygonů. Ovšem jen málo objektů v reálném světě má ostré hrany a při počítačovém zpracování nastávají problémy s úrovní detailů. Například při matematickém vyjádření kružnice nenastává žádný problém, ale při polygonální reprezentaci už je situace mnohem složitější. Po zvolení měřítka zobrazení je sice možné nadefinovat kružnici vhodným počtem bodů, ale po následném zvětšení se opět objeví jasně viditelné hrany. Použití dělicích algoritmů tento problém dokáže vyřešit. Základním principem těchto algoritmů je převádění množiny několika bodů, které představují hrubou aproximaci daného objektu (v tomto textu se budu zabývat pouze modelováním křivek) na množinu jiných bodů, jež ho definují s větší přesností. Koncept těchto algoritmů pro generování křivek přiblížím na jednoduchém Chaikinově algoritmu.

Prvotní aproximace křivky je dána množinou kontrolních bodů, které jsou předem definovány uživatelem. Na obrázku (*obr. 2.1a*) je uveden příklad takové základní aproximace ve formě bodů spojených do polygonu. Provedením jednoho kroku dělicího algoritmu dostaneme další přesnější aproximaci. Jednoduše řečeno se jedná o “useknutí” rohů původního polygonu (*obr.2.1b*). Každý bod původního polygonu je tedy nahrazen párem nových bodů, kde každý z nich je umístěn v jedné čtvrtině vzdálenosti mezi původním bodem a jedním z jeho sousedů. Stejným způsobem je možné pokračovat dále (*obr. 2.1c*), přičemž vzniká stále přesnější aproximace křivky (*obr. 2.1d, obr. 2.1e*).



Obr. 2.1: Generování křivky pomocí Chaikinova algoritmu

Ve standardním zápisu dělicího algoritmu jsou všechny body očíslovány a jsou jim přiřazeny unikátní označení - indexy. Jedna možnost takového značení je znázorněna na obrázku (obr. 2.2).



Obr. 2.2: Číslování bodů v prvním kroku Chaikinova algoritmu

Horní index značí, v kolikáté iteraci byl daný bod vytvořen. Dolní index je pak pořadové označení bodu v rámci množiny bodů, které byly vytvořeny v jedné iteraci. Každý nový bod vzniká v závislosti na souřadnicích bodů z předchozí iterace jako skalární kombinace souřadnic těchto bodů. Skalární kombinaci  $n$  bodů  $P_1, P_2, \dots, P_n$  lze vyjádřit jako

$$\alpha_1 P_1 + \alpha_2 P_2 + \dots + \alpha_n P_n, \quad (2.1)$$

kde skalární koeficienty  $\alpha_i$  dávají v součtu 1 a představují „váhu“ daného bodu při vytváření bodu nového. V případě bodu  $P_1^2$  na obrázku (obr. 2.2) dostaneme

$$P_1^2 = \frac{3}{4} P_1^1 + \frac{1}{4} P_4^1. \quad (2.2)$$

Skalární kombinaci  $n$  bodů lze dále upravit do formy

$$P = P_1 + \alpha_2(P_2 - P_1) + \dots + \alpha_n(P_n - P_1). \quad (2.3)$$

Speciálně pro dva body

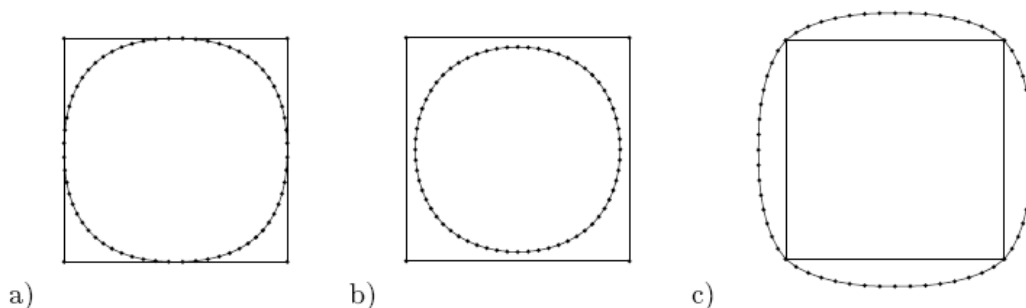
$$P = \alpha_1 P_1 + \alpha_2 P_2 = P_1 + \alpha_2(P_2 - P_1) = P_2 + \alpha_1(P_1 - P_2), \quad (2.4)$$

z čehož plyne, že bod  $P$  dělí úsečku  $P_1 P_2$  v poměru  $\alpha_2 : \alpha_1$  a nové body mají tedy souřadnice

$$P_1^2 = \frac{3}{4} P_1^1 + \frac{1}{4} P_4^1 \quad (2.5)$$

$$P_2^2 = \frac{3}{4} P_1^1 + \frac{1}{4} P_2^1 \quad (2.6)$$

Všechna ostatní schémata dělení lze vyjádřit podobným způsobem s tím rozdílem, že se použijí jiné koeficienty  $\alpha$  a v některých případech se také odvozují souřadnice z více než dvou bodů (viz. obr. 2.3b a obr. 2.3c).



Obr. 2.3: Křivky vygenerované pomocí Chaikinova dělení(a), B-spline dělení(b) a Dyn-Levyn-Gregory algoritmu(c)

### 3 L-systémy

L-systémy (Lindenmayerovy systémy) jsou matematický formalismus vycházející ze systémů paralelního přepisování řetězců (parallel string rewriting systems). Podstatou tohoto mechanismu je paralelní přepisování řetězců – posloupnosti symbolů – podle určitých pravidel. Po několika přepsáních je posloupnost symbolů možné geometricky interpretovat, přičemž každý ze symbolů má přiřazen jistý geometrický význam (např. transformaci či generování objektu). Nejznámějším uplatněním L-systémů je simulace růstu rostlin, ovšem lze jich využít například i při modelování říčních toků, modelování mořských mušlí, generování silničních sítí ve městě a také pro generování křivek definovaných dělicími schémata. V tomto textu se budu zabývat právě tou poslední možností - využitím L- systémů pro generování křivek definovaných dělicími schémata. Jejich pomocí je totiž možné zformovat dělicí algoritmy popsané v předešlé kapitole do stručné a jednoduché formy, v níž je patrný lokální charakter výpočtů v dělicích algoritmech. Techniku využití L-systému pro generování křivek budu ilustrovat opět pomocí Chaikinova algoritmu uvedeného v kapitole 2.

#### 3.1 Základy L-systémů

Nejjednodušší variantou L-systému je tzv. dL-systém (někdy označován d0L-systém). Deterministický bezkontextový L-systém (d0L-systém) je uspořádaná trojice

$$G = \langle V, P, S \rangle, \quad (3.1.1)$$



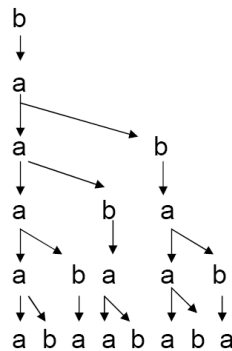
kde  $V$  je konečná abeceda symbolů ( $A, B, \dots$ ),  $P$  je konečná množina pravidel ve tvaru

$$A \rightarrow B, \quad (3.1.2)$$

kde  $A$  je přepisovaný symbol (předchůdce, rodič) a  $B$  je symbol, kterým bude  $A$  přepsán, tedy tzv. potomek nebo následník. Množina  $S$  je pak axiom - počáteční posloupnost symbolů. Nula v názvu d0L- systému znamená, že se jedná o tzv. bezkontextový přepis. To znamená, že výsledek aplikace přepisovacího pravidla na symbol  $A$  nezávisí na symbolech, jež se nacházejí po jeho levé a pravé straně (na jeho kontextu). Determinismus d0L-systému pramení z podmínky, že v množině  $P$  nesmějí být dvě pravidla se stejnou levou stranou. Pro ilustraci uvedu příklad. Necht' je dán L-systém

$$G = \langle \{a, b\}, \{a \rightarrow ab, b \rightarrow a\}, \{b\} \rangle. \quad (3.1.3)$$

V první iteraci - tedy aplikaci přepisovacího pravidla  $b \rightarrow a$  na počáteční symbol  $b$  - získám symbol  $a$ . V další iteraci už dostanu řetězec  $ab$ , v další iteraci  $aba$ , v další  $abaab$  atd. Prvních pět iterací tohoto d0L-systému je znázorněno na obrázku (obr. 3.1.1).



Obr. 3.1.1: Prvních pět iterací d0L-systému 3.1.3

Podobně jako L-systém 3.1.3 je možné nadefinovat jiný, který je již možné geometricky interpretovat. Například

$$G = \langle \{A, F, +, -\}, \{A \rightarrow F- -F, F \rightarrow F+F-F, + \rightarrow +, - \rightarrow -\}, \{A\} \rangle, \quad (3.1.4)$$

jehož první dvě iterace mají podobu

$$A \rightarrow F--F \rightarrow F+F-F- -F+F-F.$$

Takovýto typ řetězců už je možné interpretovat geometricky – tzv. želví grafikou (více viz. [2]). Ovšem pro potřeby generování dělicích křivek je nutné využít o něco složitější typ L-systémů. Takovým typem je parametrický kontextový L-systém. Fakt, že se při aplikaci

přepisovacího pravidla na daný symbol berou v úvahu i symboly na pravé a levé straně, tedy kontext symbolu, lze zapsat pomocí upraveného pravidla 3.1.2:

$$lk \langle A \rangle pk \rightarrow B, \quad (3.1.5)$$

kde  $lk$  značí levý kontext a  $pk$  pravý kontext. Znamená to, že symbol  $A$  bude přepsán symbolem  $B$  jen v případě, stojí-li uprostřed požadovaného kontextu.

To, že je L-systém parametrický znamená, že pracuje s tzv. parametrickými slovy. Parametrická slova jsou posloupnosti modulů, což jsou písmena (symboly) abecedy  $V$  (viz 3.1.1), rozšířená o parametry. Modul má syntaktickou podobu  $A(x_1, \dots, x_m)$ . Množina parametrů modulu může být prázdná, ale musí být konečná. Parametry nabývají hodnot z množiny reálných čísel. Parametrický L-systém je tedy uspořádaná čtveřice

$$G = \langle V, \Sigma, \omega, P \rangle, \quad (3.1.6)$$

kde  $V$  je abeceda,  $\Sigma$  je množina formálních parametrů,  $\omega$  je axiom (počáteční řetězec modulů) a  $P$  je konečná množina přepisovacích pravidel. Pravidla  $p \in P$  v parametrických kontextových L-systémech mají stejný tvar jako v 3.1.5 s tím rozdílem, že  $A$  a  $B$  jsou parametrizované moduly. Příkladem takového L-systému může být:

$$\begin{aligned} G &= \langle V, \Sigma, \omega, P \rangle \\ V &= \{A(x)\}, \\ \Sigma &\in R, \\ \omega &= \{A(2)\}, \\ P &= \{A(x) \rightarrow A(x+1)A(x-2)\}, \end{aligned}$$

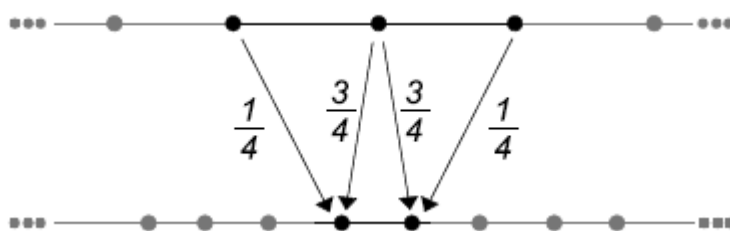
jehož první dvě iterace vypadají:  $A(2) \rightarrow A(3)A(0) \rightarrow A(4)A(1)A(1)A(-2)$ .

### 3.2 Křivky počítané dělením pomocí L-systémů

Nedávná rozšíření L-systémů umožňují místo číselných parametrů modulů použít také různé datové struktury. Takovou datovou strukturou může být například bod v prostoru. Díky tomu pak lze L-systémy využít i pro generování dělicích křivek. V první řadě je ovšem třeba si uvědomit, jak dělicí algoritmy fungují a jak je možné je implementovat s pomocí L-systémů.

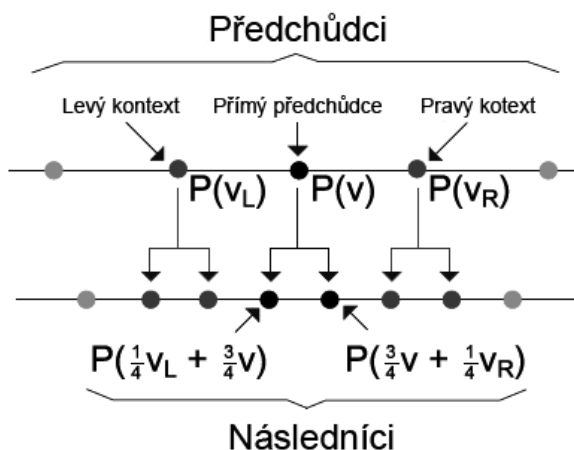
Jednou z možností formalizace dělicích algoritmů je jejich reprezentace pomocí šablon. Šablony jsou často prezentovány jako grafy, které znázorňují části z množiny rodičovských bodů a množiny jejich potomků. Tyto body bývají propojeny šipkami

ohodnocenými koeficienty  $\alpha \in \langle 0, 1 \rangle$ . Šablona pro Chaikinův algoritmus je naznačena na následujícím obrázku (obr. 3.2.1).



Obr. 3.2.1: Šablona pro Chaikinův algoritmus.

Šablony poskytují intuitivní reprezentaci dělicích algoritmů bez použití indexů. Bohužel toto není úplně nejpřesnější reprezentace. Na obrázku (obr. 3.2.2) je například řečeno, že tři body se podílejí na vytvoření dvou nových. Ale nelze se z něho dozvědět, zda se některý z těchto tří bodů podílí na tvorbě nějakých jiných bodů než těch zobrazených dvou. Není tedy zřejmý rozsah překrývání šablon při její aplikaci na různé části množiny všech bodů.



Obr. 3.2.2: Chaikinův algoritmus ve formě přepisovacího pravidla.

Nejasností s překrýváním šablon se lze zbavit tak, že se pojem „šablona“ upraví do podoby L-systému. Na sekvenci bodů lze tedy nahlížet jako na slovo nějaké abecedy, tedy jako na řetězec symbolů (modulů). Šablonu je pak možné považovat za jistou formu přepisovacího pravidla. V takovém pravidlu (ve formě jako 3.1.5) bude jeho levá strana reprezentovat sekvenci rodičovských bodů a pravá strana pak body, které mají být nově vytvořeny. Celá levá strana je rozdělena na tři části. Uprostřed (mezi značkami „<“ a „>“) se nachází symbol reprezentující rodičovský bod a vlevo a vpravo je pak kontext tohoto bodu. Při aplikaci přepisovacího pravidla na jeden konkrétní symbol dojde k nahrazení levé části pravidla jeho pravou částí, rodičovský bod je tedy nahrazen několika novými body. Souřadnice těchto bodů jsou závislé na poloze rodičovského bodu a jeho kontextu. Bod, který už byl jednou takto přepsán, ovšem už nemůže být znovu použit jako rodičovský bod.

V případě Chaikinova algoritmu (na obr. 2.2) je např. bod  $P_1^1$  považován za rodičovský bod. Body  $P_2^1$  a  $P_4^1$  představují jeho levý resp. pravý kontext a body  $P_1^2$  a  $P_2^2$  pak jeho následníky. Takže přepisovací pravidlo L-systému, jež reprezentuje Chaikinův algoritmus, bude vypadat následovně:

$$P(v_l) < P(v) > P(v_r) \rightarrow P\left(\frac{1}{4}v_l + \frac{3}{4}v\right)P\left(\frac{3}{4}v + \frac{1}{4}v_r\right), \quad (3.2.1)$$

kde  $P(v)$  je rodičovský bod,  $P(v_l)$  je bod nalevo od něj, a  $P(v_r)$  bod napravo. Z pravé strany přepisovacího pravidla je zřejmé, že jeho aplikací vzniknou dva nové body. První v závislosti na souřadnicích rodičovského bodu a jeho levého kontextu, druhý pak v závislosti na rodiči a jeho pravém kontextu.

Na stejném principu, na jakém funguje L-systém pro Chaikinův algoritmus, pracují i L-systémy pro jiná dělicí schémata. V kapitole 5 je popsán L-systém odvozený právě z Chaikinova algoritmu, který generuje aproximaci Bézierovy kubiky.

### 3.3 Fyzika v L-systémech

Fyzikou se rozumí působení nějakých vnějších vlivů na objekty a také působení objektu na sebe samého (například v závislosti na jeho hmotnosti). Aby bylo možné zahrnout fyzikální působení do L-systémů, je nutný nějaký nástroj, s jehož pomocí lze předat dodatečné informace o působení vnějšího prostředí do přepisovacího pravidla L-systému. Takový nástroj nám poskytnou tzv. otevřené parametrické kontextové L-systémy. Tyto L-systémy jsou rozšířené o tzv. komunikační moduly ve tvaru

$$E(x_1, \dots, x_m), \quad (3.3.1)$$

které slouží k přenášení informací mezi moduly a okolím objektů, který L-systém reprezentuje.

Před vlastním procesem přepsání modulů se provede mezikrok, ve kterém se získávají hodnoty skutečných parametrů komunikačních modulů  $E(x_1, \dots, x_m)$ . Na začátku tohoto kroku jsou parametry komunikačních modulů neznámé. Sekvence modulů se nejprve projde zleva doprava a zjišťuje se stav objektu s tím, že není vytvářen geometrický model. Při nalezení komunikačního modulu je vytvořena zpráva s jeho parametry. Vlastnosti modulů, kterých se tato zpráva týká jsou následně upraveny (stav modulu je znám). Po přenastavení parametrů modulu se dále pokračuje v prohlížení sekvence modulů. Jakmile jsou všechny moduly prohlédnuty a jejich parametry případně přenastaveny, začne fáze přepisování modulů, která je již shodná s přepisováním v parametrických L-systémech.

Pravidla následujícího L-systému například zabraňují tomu, aby objekt rostl ve směru osy  $y$  dále, než do vzdálenosti dva.

$$\begin{aligned} \omega: & F(0, 0) A E(0) \\ p1: & A > E(y) \quad : y < 2 \rightarrow F(x, y + 1) A \\ p2: & A > E(y) \quad : y \geq 2 \rightarrow \varepsilon. \end{aligned}$$

Modul  $F(x, y)$  kreslí úsečku z aktuální pozice do bodu o souřadnicích  $[x, y]$ . První pravidlo je aplikováno, pokud vrchol  $A$  nepřekročil hranici  $y = 2$ . Pokud k tomu došlo, je aplikováno tzv. e-pravidlo  $p2$  (pravidlo s prázdnou stranou), které vrchol  $A$  smaže z posloupnosti modulů. Iterace tedy mají tvar

$$F(0, 0) A E(0) \rightarrow F(0, 0) F(0, 1) A E(1) \rightarrow F(0, 0) F(0, 1) F(0, 2) A E(2) \rightarrow F(0, 0) F(0, 1) F(0, 2) E(3).$$

Tímto způsobem je možné zahrnout působení vlivů vnějšího prostředí (jako je např. gravitace, teplota nebo vlhkost) do L-systémů.

## 4 Bézierovy a B-spline křivky

### 4.1 Bézierova křivka

Teoretický základ těchto křivek vytvořil na přelomu 50. a 60. let P. E. Bézier, když vyvíjel programový nástroj pro návrh křivek a ploch u francouzské firmy Renault. Jedná se patrně o nejpopulárnější aproximační křivky používané pro modelování ve dvou rozměrech, ale i pro definici trojrozměrných objektů. Tyto křivky se často používají například při návrhu počítačových písem (fontů). Nejprve uvedu obecné Bézierovy křivky  $n$ -tého stupně a pak jejich nejčastěji používanou variantu - Bézierovy kubiky.

Bézierova křivka  $n$ -tého stupně je určena  $n + 1$  body  $P_i$ , které tvoří tzv. řídicí polygon, a vztahem

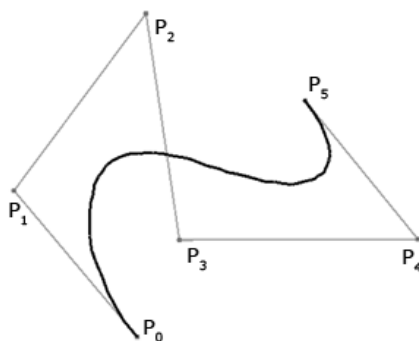
$$Q(t) = \sum_{i=0}^n P_i B_i^n(t), \quad (4.1.1)$$

kde  $B_i^n(t)$  jsou Bernsteinovy polynomy  $n$ -tého stupně definované vztahem

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{(n-i)}; t \in \langle 0, 1 \rangle; i = 0, 1, \dots, n. \quad (4.1.2)$$

Pokud do vztahu 4.1.1 dosadíme hodnotu  $t = 0$ , resp.  $t = 1$ , zjistíme, že křivka začíná a končí v krajních bodech řídicího polygonu. Dosazením hodnot  $t = 0$ , resp.  $t = 1$  do derivace stejného vztahu dostaneme výrazy pro tečné vektory v krajních bodech křivky:

$$\begin{aligned}\vec{q}_0 &= n(P_1 - P_0), \\ \vec{q}_1 &= n(P_n - P_{(n-1)}).\end{aligned}\tag{4.1.3}$$



Obr. 4.1.1: Ukázka Bézierovy křivky pátého stupně

Jednou z vlastností Bézierových křivek je to, že při změně polohy jednoho řídicího bodu  $P_i$  dojde ke změně tvaru celé křivky. To může být pro křivky vyšších stupňů nepříjemné, proto jsou při modelování složitější křivky děleny na segmenty nižších stupňů (v naprosté většině případů na kubiky), a tyto segmenty se pak postupně navazují. Při navazování segmentů složených z Bézierových křivek zaručíme její spojitost identitou posledního bodu řídicího polygonu  $Q_1$  a prvního bodu segmentu  $Q_2$ .

#### 4.1.1 Bézierova kubika

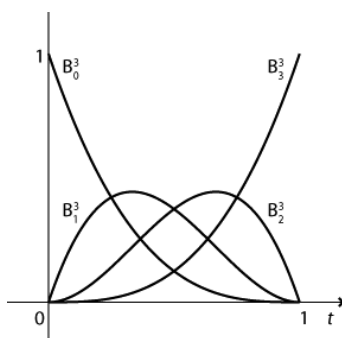
Jedná se o nejčastěji používanou variantu Bézierových křivek v počítačové grafice. Bézierova kubika je zadána čtyřmi body  $P_0$ ,  $P_1$ ,  $P_2$  a  $P_3$ . Vychází z prvního řídicího bodu, končí v posledním a je určena vztahem

$$Q(t) = \sum_{i=0}^3 P_i B_i^3(t),\tag{4.1.1.1}$$

přičemž Bernsteinovy polynomy třetího stupně jsou

$$B_0^3(t) = (1-t)^3, B_1^3(t) = 3t(1-t)^2, B_2^3(t) = 3t^2(1-t), B_3^3(t) = t^3.\tag{4.1.1.2}$$

a jejich průběhy jsou na následujícím obrázku (obr. 4.1.1.1):



Obr. 4.1.1.1: Bernsteinovy polynomy třetího stupně

Bézierovu kubiku můžeme zapsat v maticovém tvaru

$$Q(t) = [t^3 \ t^2 \ t \ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (4.1.1.3)$$

a lze také vyjádřit tečné vektory v prvním a posledním bodě:

$$\vec{p}'(0) = 3(P_1 - P_0), \quad \vec{p}'(1) = 3(P_3 - P_2) \quad (4.1.1.4)$$

## 4.2 B-spline křivky

### 4.2.1 Coonsova kubika

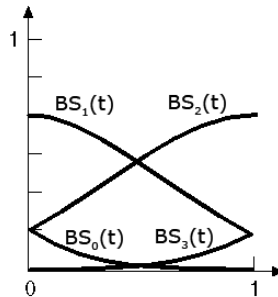
Coonsova kubika má některé odlišné vlastnosti oproti Bézierově a díky nim se často používá při konstrukci po částech skládaných polynomiálních křivek. Coonsova kubika neboli uniformní neracionální B-spline je opět určena čtyřmi řídicími body  $P_0, P_1, P_2, P_3$  a vztahem

$$Q(t) = \frac{1}{6} [t^3 \ t^2 \ t \ 1] \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (4.2.1)$$

Bázové polynomy  $BS_0(t), BS_1(t), BS_2(t)$  a  $BS_3(t)$  Coonsových kubik s průběhy na obrázku (obr.4.2.1) jsou voleny tak, že segmenty křivky na rozdíl od Bézierových křivek

obecně neprocházejí krajními body svého řídicího polygonu. Dosazením  $t = 0$  a  $t = 1$  do vztahu 4.2.1 zjistíme, že křivka začíná a končí v bodech

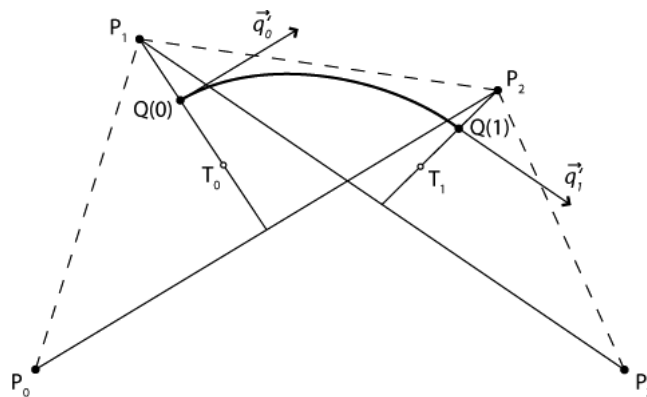
$$Q(0) = \frac{(P_0 + 4P_1 + P_2)}{6}, \quad Q(1) = \frac{(P_1 + 4P_2 + P_3)}{6}. \quad (4.2.2)$$



Obr. 4.2.1: Bázové polynomy B-spline kubiky

Bod  $Q(0)$  (viz obr.4.2.2) leží v první třetině té těžnice trojúhelníku tvořeného body  $P_0$ ,  $P_1$  a  $P_2$ , která začíná v bodě  $P_1$ . Tento bod se nazývá antitěžišť. Podobně bod  $Q(1)$  leží v první třetině těžnice trojúhelníku tvořeného body  $P_1$ ,  $P_2$  a  $P_3$  začínající v bodě  $P_2$ .

Platí, že tečný vektor  $q_0'$  je rovnoběžný s úsečkou  $P_0P_2$  a vektor  $q_1'$  je rovnoběžný s úsečkou  $P_1P_3$ .

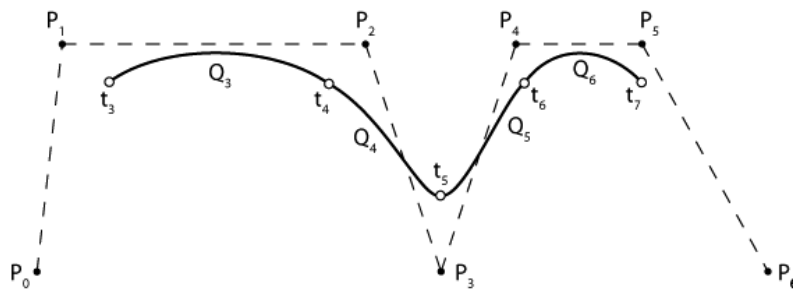


Obr. 4.2.2: Coonsova kubika

## 4.2.2 Uniformní kubický B-spline

Jiný název pro tuto křivku zní Coonsův kubický B-spline. Vzniká navazováním Coonsových kubik (obr.4.3.1) takto: segment  $Q_i$  je určen body  $P_{i-3}$ ,  $P_{i-2}$ ,  $P_{i-1}$  a  $P_i$ . Následující segment  $Q_{i+1}$  je definován body  $P_{i-2}$ ,  $P_{i-1}$ ,  $P_i$  a  $P_{i+1}$ , tedy třemi posledními body segmentu  $Q_i$  a jedním bodem segmentu následujícího. Zatímco Coonsova křivka je určena právě čtyřmi body, Coonsův B-spline je určen  $n \geq 4$  body a skládá se z  $n - 3$  segmentů.

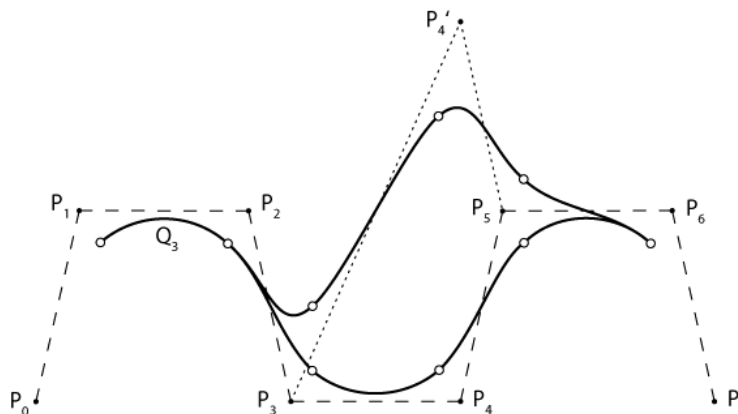




Obr. 4.3.1: Coonsův kubický B-spline

Lze dokázat, že koncový bod křivky v jednom segmentu a počáteční bod křivky v následujícím segmentu jsou totožné a zároveň i jejich první derivace jsou v tomto místě navázání stejné. Další vlastnosti B-spline křivek jsou:

- invariance vůči otáčení, posunutí a změně měřítka,
- každý segment leží v konvexní obálce bodů svého řídicího polygonu a celá křivka pak v konvexní obálce celého řídicího polygonu,
- změna polohy řídicích bodů je lokální, konkrétně se promítne jen do těch segmentů křivky, které ho mají ve svém řídicím polygonu (tzn. změna tvaru nanejvýš čtyř segmentů křivky a tří odpovídajících uzlových bodů, viz obrázek (obr.4.3.2) ).



Obr. 4.3.2: Vliv změny polohy řídicího bodu na tvar křivky

## 5 Aproximace uniformního kubického B-spline pomocí Bézierovy křivky

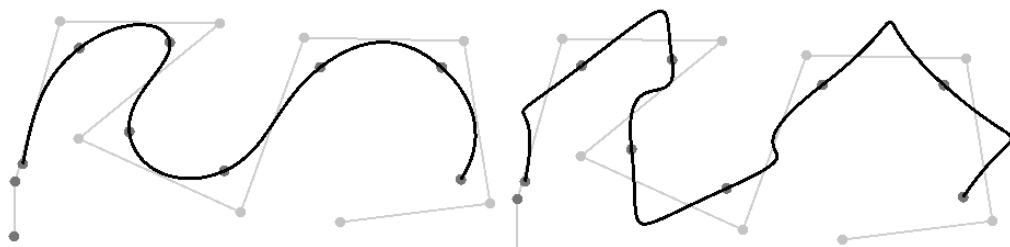
Při aproximaci uniformního kubického B-spline (dále jen ukB-spline) pomocí Bézierovy křivky je potřeba zvážit jejich vlastnosti a v čem se obě křivky navzájem liší.

Zásadním rozdílem je vliv jednotlivých bodů řídicího polygonu na tvar celé křivky. V případě ukB-spline má jeden řídicí bod pouze lokální vliv, tzn. že změna polohy bodu se projeví jen do těch segmentů, které ho mají ve svém řídicím polygonu (maximálně tedy do čtyř segmentů). Ovšem změna polohy jednoho řídicího bodu Bézierovy křivky má za následek deformaci celé křivky. Dále je potřeba zvážit stupeň Bézierovy křivky, kterým bude ukB-spline aproximována. Vzhledem ke stupňům segmentů ukB-spline křivek je výhodné využít pro aproximaci Bézierovu kubiku, tedy křivku třetího stupně (řídicí polygon má čtyři body, stejně jako jeden segment ukB-spline).

Vzhledem k vlastnostem ukB-spline a jeho segmentované struktuře není velkým problémem nahrazení jednotlivých segmentů Bézierovou kubikou. Počáteční a koncový bod je znám (kapitola 4.2), stejně jako směry tečných vektorů v těchto bodech. Velikost těchto vektorů lze odvodit z délky příslušných úseček. Na obrázku (obr.4.2.2) je vektor  $q_0'$  rovnoběžný s úsečkou  $P_0P_2$  a jeho délku lze odvodit z té samé úsečky. Zavedu tedy parametr  $b \in \langle 0,1 \rangle$ , kterým vynásobím velikost této úsečky, a získám tak velikost tečného vektoru. Když už jsou známy počáteční body i tečné vektory v nich, lze zkonstruovat Bézierovu kubiku (obr. 5.1). Takto vypočtenými křivkami je pak možné nahradit segmenty ukB-spline (obr. 5.2).



Obr. 5.1: Tvary Bézierovy křivky při různých hodnotách parametru  $b$   
(vlevo  $b = 0,2$ ; vpravo  $b = 0,8$ )



Obr. 5.2: Křivka složená ze segmentů tvořených Bézierovými kubikami  
(vlevo  $b = 0,2$ ; vpravo  $b = 0,5$ )

## 6 Aproximace Bézierovy křivky pomocí L-systémů

Hlavním problémem při aproximaci Bézierovy křivky pomocí L-systémů je vliv jednotlivých řídicích bodů na tvar celé křivky. Protože dělicí schémata obecně a tedy i L-systémy z nich odvozené mají lokální vlastnosti při dělení, není možné odvodit jeden univerzální L-systém pro aproximaci Bézierovy křivky různých řádů. Protože jsem pro aproximaci ukB-spline využil Bézierovu kubiku, zabýval jsem se pouze L-systémem generujícím aproximaci tohoto specifického druhu Bézierovy křivky.

### 6.1 Aproximace Bézierovy kubiky

V kapitole 3 je popsán velmi jednoduchý L-systém generující křivky podle Chaikinova dělicího algoritmu (dělicí pravidlo 3.2.1). Tento L-systém jsem rozšířil a upravil tak, abych ho mohl využít pro generování křivky aproximující Bézierovu kubiku.

Na obrázku (*obr. 3.2a*) je vidět, že při Chaikinově dělení se výsledná křivka dotýká hran řídicího polygonu. Je to způsobeno tím, že při výpočtu nového bodu se uvažují pozice pouze dvou bodů a výsledný bod je tedy umístěn na úsečce, která tyto body spojuje. I po více krocích dělení je tím způsobeno, že vždy přesně dva body budou ležet na dané hraně. Toto ovšem pro Bézierovu křivku neplatí. Je tedy nutné upravit výpočet tak, aby se křivka „vzdálila“ od příslušné hrany. Toho lze dosáhnout zahrnutím dalšího bodu do výpočtu. Nový bod budu tedy počítat ze souřadnic tří bodů - rodičovského bodu a jeho levého a pravého souseda.

Vzhledem k tomu, že se v této práci věnuji generování otevřených křivek, je nutné brát při výpočtu v úvahu i fakt, že existuje jejich počáteční a koncový bod a výpočet pro ně bude vypadat jinak. V případě krajních bodů je výpočet nového bodu závislý pouze na dvou bodech (počátečním resp. koncovém a jeho pravém resp. levém sousedovi). Otázkou zůstává pouze míra, se kterou se oba body budou podílet na souřadnicích bodu nového, tj. jejich koeficienty  $\alpha_i$  (kapitola 2). Tyto koeficienty je potřeba určit také pro „vnitřní“ body křivky.

V programu „Bezier approximation“ je možné upravovat jak koeficienty pro krajní body, tak koeficienty pro body „uvnitř“ křivky. Je možné je přímo zadávat a okamžitě vidět výslednou křivku. Tento program také dokáže zhodnotit přesnost aproximace dané křivky. Výpočet ohodnocení funguje na jednoduchém principu. Nejprve je nutné, aby jak Bézierova křivka, tak křivka vygenerovaná L-systémem byla reprezentována stejným počtem bodů. Postupně jsou pak vypočteny vzdálenosti mezi jednotlivými body obou křivek. Jedná se vždy o dvojici bodů se stejným indexem v rámci jedné křivky (například se počítá vzdálenost prvního bodu Bézierovy křivky a prvního bodu její aproximace). Výsledný součet všech vzdáleností je pak vydělen počtem bodů křivky. Konečný výsledek udává aritmetický průměr jednotlivých vzdáleností. Výpočet je tedy podle vztahu (6.1.1)

$$e = \left(\frac{1}{n}\right) * \sum_{i=1}^n |Pb_i - Pl_i|, \quad (6.1.1)$$

kde  $e$  značí průměrnou vzdálenost bodů, tedy naše ohodnocení. Symbol  $n$  je počet bodů jedné křivky,  $Pb_i$  je  $i$ -tý bod Bézierovy křivky a  $Pl_i$   $i$ -tý bod křivky vygenerované L-systémem.

Kdyby bylo možné pomocí použitého L-systému vygenerovat přesně body Bézierovy křivky, byl by tento průměr roven nule. Mojí snahou tedy bylo určit takové koeficienty, aby bylo příslušné ohodnocení co nejnižší. V upraveném programu jsem provedl experiment na 1 000 různých křivkách (každá reprezentována 32 body) pro parametry v intervalu  $\langle 0, 100 \rangle$  a dostal jsem hodnotu nejlepšího ohodnocení rovnu 1,783. To ale platí jen pro jeden určitý tvar křivky a pro jednu kombinaci koeficientů. Problémem je, že pro různé křivky mohou mít její koeficienty zaručující nejlepší ohodnocení jiné hodnoty. Při určení univerzálních koeficientů jsem nepoužil nejpřesnější možný výsledek, ale o něco méně přesný, který je ovšem použitelný pro různé křivky. Výsledný L- systém má tedy tvar:

$$G = \langle V, \Sigma, \omega, T \rangle \quad (6.1.2)$$

$$V: \{P(v)\},$$

$$\Sigma: v \in R^2,$$

$$\omega: P(v_1) P(v_2) P(v_3) P(v_4),$$

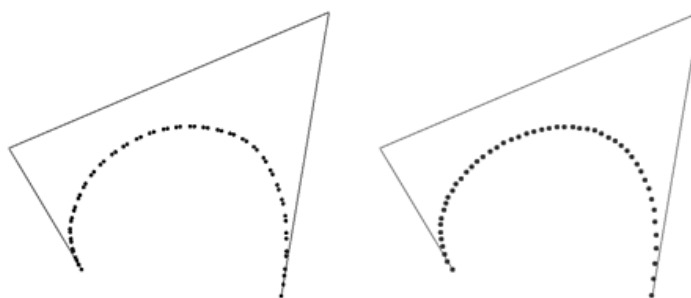
$T:$

$$T_1: \varepsilon < P(v) > P(v_r) \rightarrow E P(v) P\left(\frac{11}{20} v + \frac{9}{20} v_r\right),$$

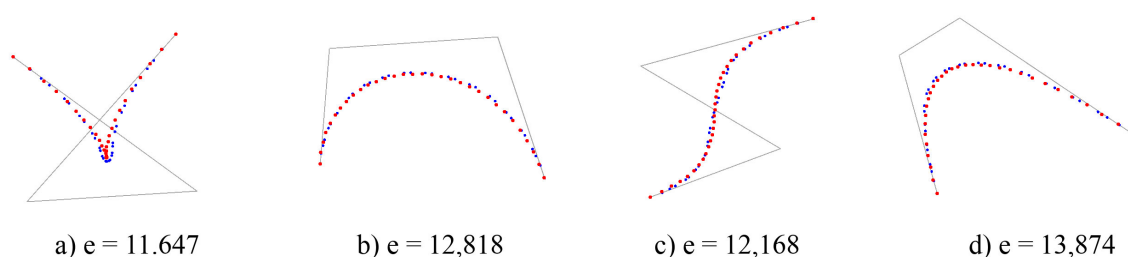
$$T_2: P(v_l) < P(v) > P(v_r) \rightarrow EP\left(\frac{7}{20} v_l + \frac{9}{20} v + \frac{4}{20} v_r\right) P\left(\frac{4}{20} v_l + \frac{9}{20} v + \frac{7}{20} v_r\right),$$

$$T_3: P(v_l) < P(v) > \varepsilon \rightarrow E P\left(\frac{9}{20} v_l + \frac{11}{20} v\right) P(v).$$

Množina  $V$  představuje množinu prvků, nad kterými bude probíhat výpočet, v tomto případě se jedná o body  $P(v)$  ve dvojrozměrném prostoru. Množina  $\Sigma$  je množina parametrů bodů, tedy jejich souřadnic v prostoru. Počáteční slovo  $\omega$  představuje čtveřice řídicích bodů. Množina  $T$  je pak trojice přepisovacích pravidel. Pravidla  $T_1$  a  $T_3$  jsou aplikována na krajní body křivky a pravidlo  $T_2$  na „vnitřní“ body. Moduly  $E$  jsou komunikační moduly (kapitola 3.3).



Obr. 6.1.1: Bézierova kubika vygenerovaná pomocí L-systému (vlevo)  
a vypočtená podle vztahu 4.1.1.1 (vpravo)

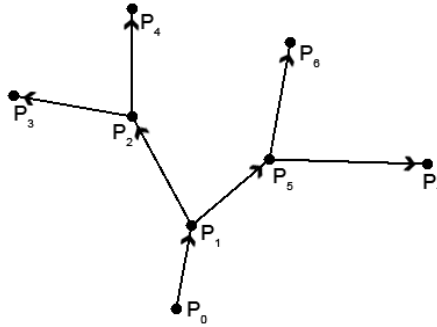


Obr. 6.1.2: Bézierovy křivky (červeně) a jejich aproximace (modře)  
s příslušnými ohodnoceními pro koeficienty uvedené v definici L-systému (6.1.2)

## 7 Větvení křivky

Výpočet aproximační křivky zadané lineární posloupností řídicích bodů lze upravit tak, aby bylo možné křivku větvit. Řídící body lze stále uchovávat ve formě seznamu, ale je nutné ke každému z nich připojit informace, které umožní vytvořit stromovou strukturu bodů. Spolu se souřadnicemi daného bodu tedy budu ukládat i ukazatel na jeho rodičovský bod a na jeho potomky (pro jednoduchost jsem zvolil počet potomků pouze 2). Takto získám návaznost bodů, která je ve formě binárního stromu. Na obrázku (obr.7.1) je taková struktura naznačena.

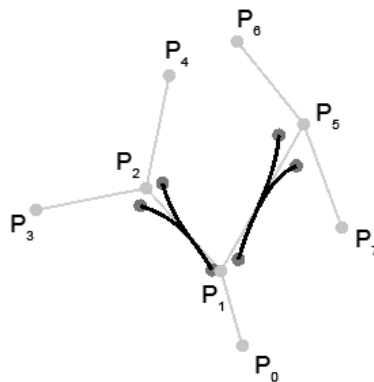
Díky této hierarchii řídicích bodů lze upravit výpočet křivky tak, aby šla větvit, a to dvěma způsoby. První možností je větvení po segmentech, kdy budu mezi řídicími body vyhledávat čtveřice bodů, jež poslouží jako řídicí body jednoho segmentu křivky (viz. kap. 6). Druhou možností je využít upravené dělicí schéma (resp. L-systém), které je schopné vypořádat se i s větvením. To znamená, že bere v potaz větší počet sousedů na jedné straně bodu než jen jednoho, jako je tomu v případě nevětvené křivky. Obě možnosti jsou blíže popsány v následujících kapitolách.



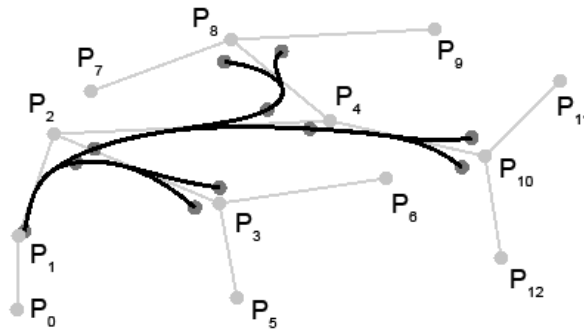
Obr. 7.1: Hierarchická struktura bodů ve formě binárního stromu  
(šipky znázorňují vztah rodič -> potomek)

### 7.1 Větvení po segmentech

Při větvení po segmentech se vyhledávají v zadané struktuře řídicích bodů čtveřice bodů, které poslouží jako řídicí polygony jednotlivých segmentů křivky. Při jejich vyhledávání lze výhodně využít vztah mezi body, tj. vztah rodič-potomek. Vyberu jeden řídicí bod a prozkoumám jeho rodičovské body (jde o procházení binárního stromu směrem od listů ke kořeni). Pokud má zvolený bod alespoň tři, získám čtveřici bodů, jež tvoří řídicí polygon jednoho segmentu křivky. Tímto způsobem prozkoumám všechny zadané řídicí body. Na obrázku (obr.7.1.1) splňují tuto podmínku pouze čtyři body:  $P_3$ ,  $P_4$ ,  $P_6$  a  $P_7$ . S pomocí bodů  $P_3$  a  $P_4$  získám dva segmenty, které vycházejí z jednoho bodu, stejně tak segmenty pro body  $P_6$  a  $P_7$  vycházejí ze stejného bodu, došlo tedy k rozvětvení křivky. Takto lze získat segmenty křivky zadané jakkoliv složitým binárním stromem (obr.7.1.2).



Obr. 7.1.1: Segmenty pro body  $P_3$ ,  $P_4$ ,  $P_6$  a  $P_7$

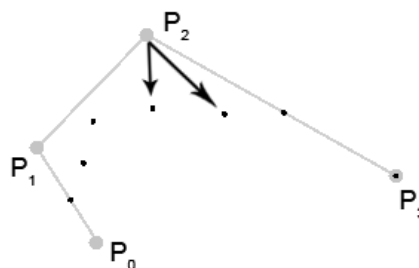


Obr. 7.1.2: Příklad složitější struktury řídicích bodů a pro ně vypočtená křivka

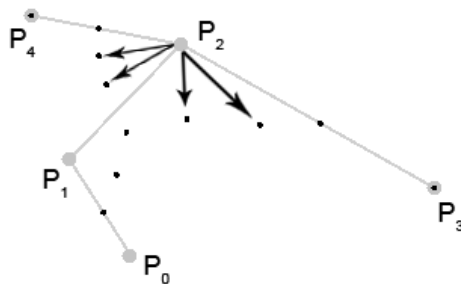
## 7.2 Větvení v rámci dělení

Jelikož se v této práci zabývám aproximací Bézierovy křivky pomocí L-systémů, je možné její větvení realizovat ještě jiným způsobem než po segmentech jako v předešlé kapitole. Tento způsob obnáší pouze upravení L-systému tak, aby byl schopný brát v úvahu rozvětvenou stromovou strukturu řídicích bodů.

Původní L-systém pro aproximaci Bézierovy křivky pracuje s jedním bodem a jeho levým a pravým sousedem. Na obrázku (obr. 7.2.1) je znázorněno dělení bodu  $P_2$  s levým sousedem  $P_1$  a pravým sousedem  $P_3$ . Problém nastane, jsou-li na jedné straně body dva. Na obrázku (obr. 7.2.2) je dělen opět bod  $P_2$ . Jeho levý soused je  $P_1$  a na pravé straně jsou body  $P_3$  a  $P_4$ . Jak je vidět na obrázku, problém lze vyřešit jednoduše tak, že bod  $P_2$  necháme dělit dvakrát. Při prvním dělení vzniknou dva body se souřadnicemi závislými na bodech  $P_1, P_2$  a  $P_3$  a při druhém dělení na bodech  $P_1, P_2$  a  $P_4$ . Tento případ je nutné řešit jen v případě bodu, ve kterém dochází k větvení, pro zbytek bodů zůstává L-systém nezměněn.

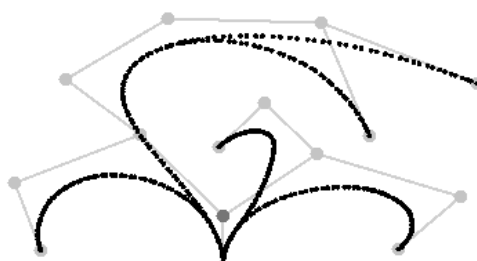


Obr. 7.2.1: Dělení bodu  $P_2$  s levým sousedem  $P_1$  a pravým  $P_3$



Obr. 7.2.2: Dělení bodu v místě větvení

S takto upraveným L-systémem lze vypočítat křivku zadanou libovolným množstvím řídicích bodů v binárním stromě. Příklad je na obrázku (obr. 7.2.3).



Obr. 7.2.3: Příklad křivky rozvětvené pomocí dělení

## 8 Implementace

V následujících kapitolách jsem popsal různé problémy, se kterými jsem se potýkal v průběhu vytváření programového vybavení k bakalářské práci. Celkem jsem vytvořil tři programy, ve kterých jsem implementoval problémy popsané v předešlých kapitolách.

První program je „Bezier aproximation“, ve kterém je možné srovnat Bézierovu křivku vypočtenou pomocí vztahu 4.1.1 s křivkou vypočtenou pomocí L-systému (kapitola 6.1). Parametry L- systému je možné za běhu měnit.

Program „Global L-system“ umožní uživateli ručně zadat stromovou strukturu řídicích bodů křivky, s nimiž je možné pohybovat. Nad takto zadanými body je pak vypočtena rozvětvená křivka způsobem popsaným v kapitole 7.2. Na tuto křivku je možné po zaškrtnutí políčka „Local force“ aplikovat působení lokální síly.

Třetím programem je „B-spline aproximation“. Řídící body jsou v něm generovány pomocí L-systému a výsledná křivka je počítaná způsobem popsaným v kapitole 7.1. Na tuto křivku lze pak aplikovat globální fyzikální vlivy.



## 8.1 Řídící body

Řídící body křivky je možné zadávat několika způsoby: náhodně, ručně nebo je generovat pomocí nějakého algoritmu (například pomocí L-systému). Protože náhodně vygenerované body by tvořily nepřehledné struktury a zmatené křivky, zabýval jsem se ručním zadáváním bodů a generováním pomocí L-systému. Každá z těchto možností má své výhody i nevýhody, které budou popsány dále.

Pro účely výpočtu křivek, navíc s možností větvení, bylo nutné vytvořit speciální datovou strukturu, která by dokázala uchovat potřebné informace. Takovou strukturu jsem nazval *myNode* a její základní struktura je:

```
public class myNode
{
    public PointF point;
    public myNode l_branch;
    public myNode r_branch;
    public myNode parent;

    ...
}
```

V proměnné *point* je uložena pozice daného bodu v prostoru, jehož souřadnice jsou typu float. *L\_branch* a *r\_branch* jsou ukazatele na levého a pravého potomka a *parent* je ukazatel na rodičovský bod. Tato základní struktura je implementována ve všech programech, ovšem v každém z nich bylo nutné v závislosti na použitých algoritmech ještě nějaké informace přidat (například antitěžiště vypočtená k příslušnému bodu).

### 8.1.1 Ruční zadávání

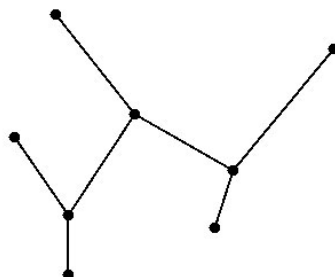
V programu „Bezier approximation“ je možné řídicí body zadávat jednoduše klikáním levého tlačítka na bílé ploše. Křivku lze v tomto případě vykreslit, jakmile jsou k dispozici více než dva body.

Program „Global L-system“ už umožní uživateli vytvořit stromovou strukturu řídicích bodů. Problém při tomto zadávání je způsob, jak získat informace o návaznosti bodů ve stromové struktuře, tj. který bod je rodičem nebo naopak potomkem. Postup zadávání jsem tedy vyřešil následovně:

- První dva body jsou již zadané, přičemž teprve druhý bod v pořadí je považován za kořenový bod stromu.
- Stisknutím pravého tlačítka nad kterýmkoliv bodem vznikne nový bod a tažením myši se stále stisknutým pravým tlačítkem je tento bod posunut na nové souřadnice.

Umístování bodu skončí po uvolnění pravého tlačítka myši. K tomuto novému bodu je automaticky uložena informace, že bod, nad kterým bylo kliknuto, je jeho předkem a zároveň k původnímu bodu je přidána informace o tom, že nově vytvořený bod je jeho potomek.

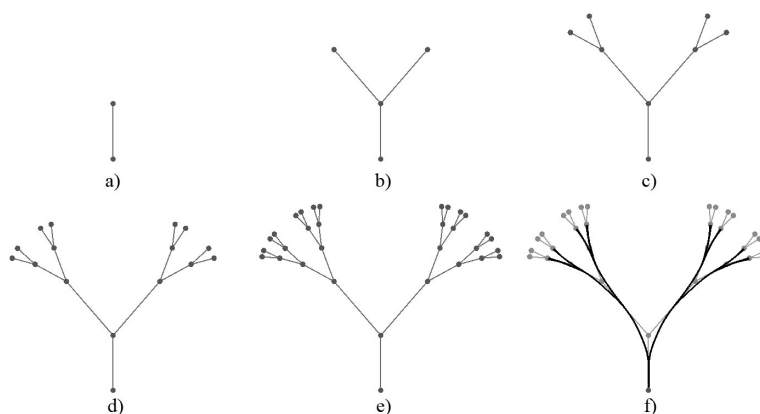
Každému bodu je takto možné přidat dva body, jež budou jeho potomky. Vznikne tak hierarchická struktura bodů ve formě binárního stromu. Příklad je na obrázku (obr. 8.1.1.1).



Obr. 8.1.1.1: Příklad stromové struktury zadané ručně

### 8.1.2 Využití L-systému

Řídící body v programu „B-spline aproximace“ jsou generovány pomocí jednoduchého L- systému, který vytváří stromovou strukturu ve tvaru rostliny. Uživatel může pouze zadat počet kroků, který ovlivní hloubku stromové struktury, a řídicí body křivky jsou pak automaticky vygenerovány. Na obrázku (obr. 8.1.1.2a-e) je vidět, jak použitý L-systém funguje. V každém kroku jsou vytvořeny dva nové body, které vycházejí ze svého rodičovského bodu v určitém úhlu a vzdálenosti.



Obr. 8.1.1.2: Vývoj L-systému řídicích bodů a vygenerovaná křivka

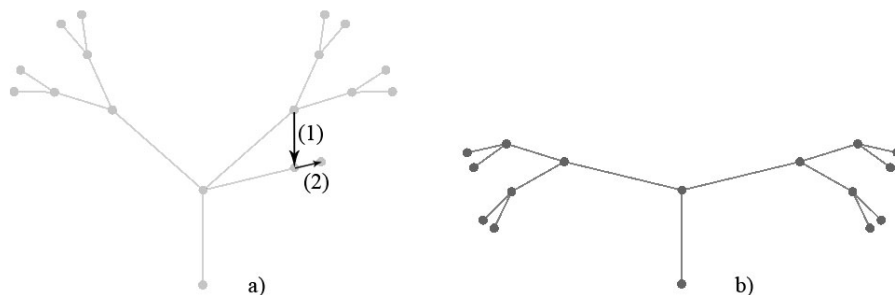
## 8.2 Možnosti začlenění fyziky

Fyzikální vlivy jsem rozdělil na dva druhy. První možností jsou vlivy globální, tedy ty, které mají vliv na celý objekt. Mezi ně patří například teplota nebo vlhkost okolního prostředí a také síla působící na celý objekt, tedy gravitace. Druhou možností jsou lokální vlivy jako například jedna síla působící v určitém bodě objektu. Každou z možností jsem implementoval v jiném programu.

### 8.2.1 Globální vlivy

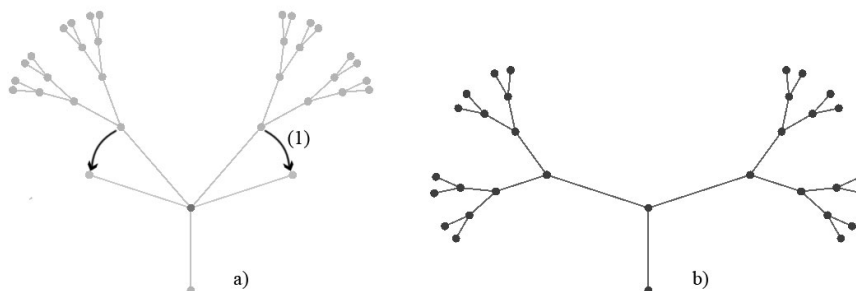
Existují dva způsoby, jak na celou křivku aplikovat vliv vnějšího prostředí. První možností je brát v úvahu toto působení již při generování řídicích bodů (kapitola 8.1.2). Druhou možností je využít komunikačních modulů L-systému. Při použití komunikačních modulů jsem ale narazil na jeden problém a tím je neschopnost L-systému přizpůsobit se při dělení bodu vlastnostem vzdálenějších částí křivky. Příkladem může být působení gravitace na křivku. Body reprezentující určitou část křivky (hmotného objektu) mají svou hmotnost a reagují tak daným způsobem na gravitaci, ale měl by se brát i ohled na ty navazující části, které nejsou nijak upevněny a také působí na tyto body. Jako možné řešení jsem vyzkoušel zavedení dalších proměnných pro každý bod, jejichž hodnoty by byly závislé na vzdálenosti od kořenového bodu. Podle těchto hodnot by pak bylo možné zjišťovat vlastnosti i v jiných částech křivky. Tyto proměnné by ale bylo nutné už při zadávání řídicích bodů přepočítávat a především by pak nastal problém při samotném dělení. Každý bod při dělení vytváří dva nové a jejich parametry by pak bylo nutné složitě přepočítávat. Kvůli těmto problémům jsem se rozhodl brát globální fyzikální vlivy v úvahu už při generování řídicích bodů. Vyhnul jsem se tím také mnoha výpočtům a urychlil tím chod programu. Pro otestování možností globálních vlivů jsem implementoval vliv gravitace a teploty.

L-systém generující řídicí body reaguje na gravitaci tak, že nově vygenerovaný bod posune ve směru gravitace (obr. 8.2.1.1a krok 1). Aby byla zachována původní vzdálenost mezi body, je ještě potřeba upravit velikost nově vzniklého vektoru na velikost, kterou měl před změnou. Nový vektor je tedy znormalizován a vynásoben původní velikostí (obr. 8.2.1.1a krok 2). Tento postup je aplikován na každý bod při jeho vytváření. Finální soustava řídicích bodů je na obrázku (obr. 8.2.1.1b).



Obr. 8.2.1.1: Implementace gravitace

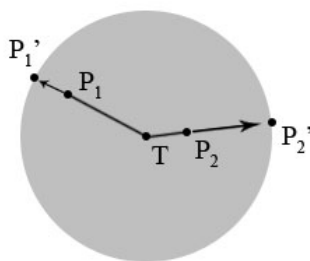
Abych soustavu řídicích bodů mohl měnit v závislosti na teplotě, musel jsem si nejprve uvědomit, jak se různé předměty a materiály s rostoucí teplotou mění. Homogenní objekty tvořené jedním materiálem se pouze roztahují nebo smršťují. To by se při generování řídicích bodů projevilo pouze změnou velikosti celé soustavy bodů. Ovšem tzv. bimetalický pás, tzn. pás složený ze dvou různých kovů, se s rostoucí teplotou ohýbá. Ten samý jev jsem se pokusil aplikovat i na řídicí body křivky. Při jejich generování tedy v závislosti na zadané teplotě dochází ke změně úhlu, ve kterém vycházejí ze svého rodičovského bodu (obr. 8.2.1.2a krok 1). Aplikace teploty na všechny řídicí body pak může vypadat jako na obrázku (obr. 8.2.1.2b).



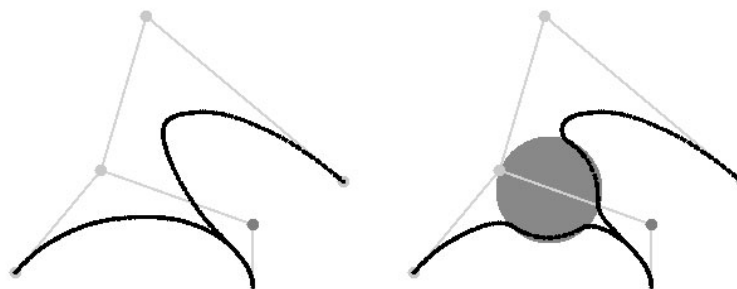
Obr. 8.2.1.2: Implementace teploty

## 8.2.2 Lokální vlivy

Program „Global L-system“ generuje křivku pomocí L-systému, který umožňuje působení síly pouze na určitou část křivky. Místo působení síly jako vektoru v určitém bodě jsem se kvůli jednoduššímu ovládání rozhodl pro implementaci objektu (ve tvaru kruhu), jehož prostřednictvím je možné křivku přímo měnit. Aplikace síly je zavedena do L-systému přes komunikační moduly (kapitola 3.3). Při dělení každého bodu se zjišťuje, zda se tento bod nenachází uvnitř zmíněného objektu. Pokud ano, je tento bod ještě před dělením posunut ve směru působící síly. Směr a velikost síly je odvozena ze vzájemné polohy bodu  $P_i$  a těžiště  $T$  působícího objektu (obr. 8.2.2.1).



Obr. 8.2.2.1: Objekt působící na body



Obr. 8.2.2.2: Objekt působící na křivku

## 9 Závěr

V bakalářské práci jsem popsal základy dělicích schémat pro generování křivek a základy L- systémů a prozkoumal možnosti propojení obou oblastí tak, abych dokázal vygenerovat aproximaci Bézierovy křivky právě pomocí L-systému.

Tuto křivku jsem dále využil při testování algoritmu pro aproximaci uniformní kubické B-spline křivky pomocí Bézierovy křivky. Vytvořený algoritmus umožňuje vytvářet zajímavé křivky, které se od klasické B- spline podstatně liší.

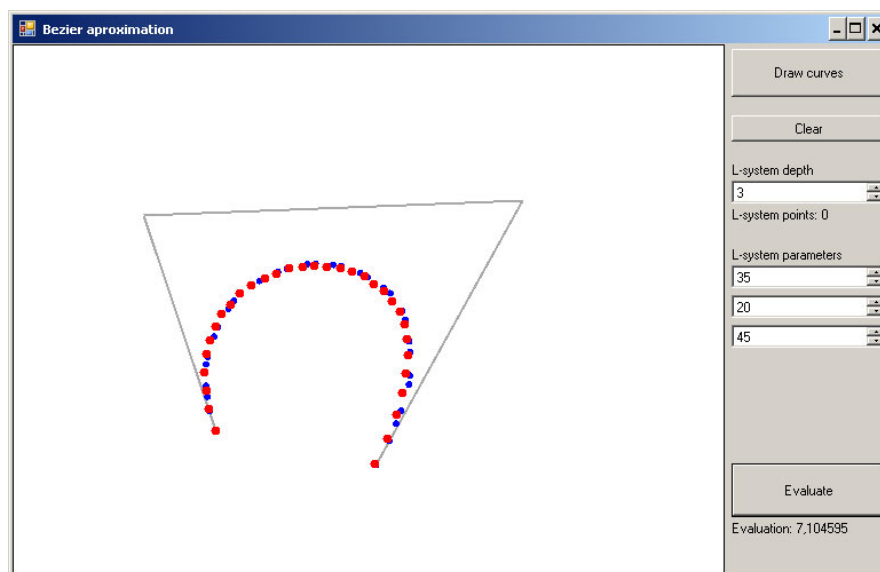
Dále jsem prozkoumal možnosti větvení křivek. Vyzkoušel jsem dvě metody, jak k větvení přistupovat, a zjistil jsem, že oběma způsoby je možné vytvářet zajímavé rozvětvené křivky.

Na konec jsem zjišťoval, jak lze do křivek vygenerovaných pomocí L-systému zavést působení vnějších vlivů, jako je například gravitace, teplota nebo síla působící v jednom bodě. Zjistil jsem, že existují různé způsoby, jak lze modifikovat křivku v závislosti na okolním prostředí a tyto způsoby implementoval v přiložených programech.

## 10 Uživatelská příručka

### 10.1 Program Bezier aproximation

Tento program umožňuje porovnávání křivky vygenerované L-systémem s Bézierovou křivkou.



Obr. 10.1.1: Okno programu

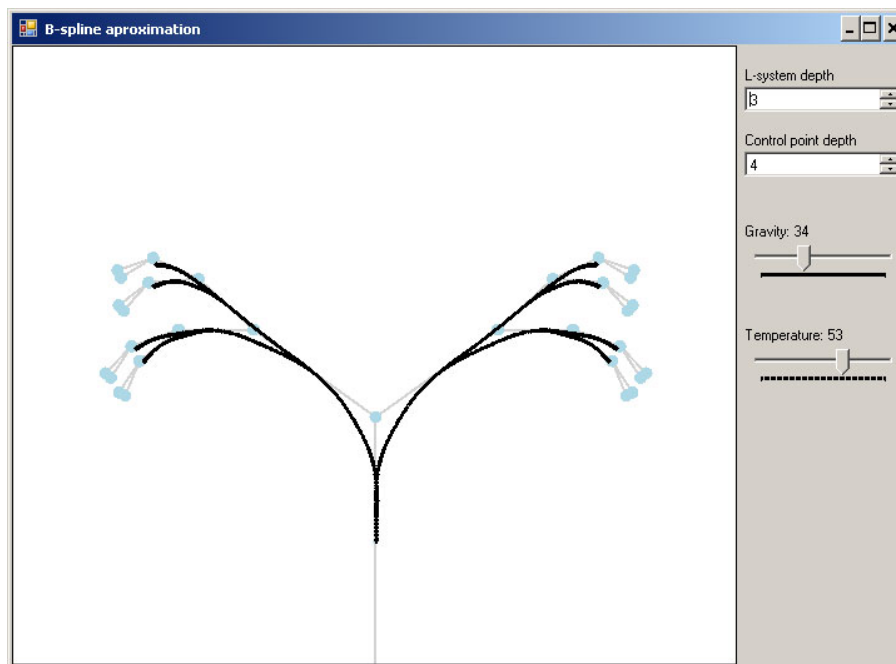
Po spuštění programu je potřeba klikáním levého tlačítka myši na bílé ploše zadat čtyři řídicí body křivky. Jakmile jsou tyto body zadány, stisknutím tlačítka „Draw curves“ program vykreslí body dvou křivek. Červenou barvou jsou zobrazeny body Bézierovy křivky a modrou barvou pak body křivky vygenerované pomocí L-systému. Pro zadání nové křivky stačí stisknout tlačítko „Clear“ nebo pravé tlačítko myši.

Pod popiskem „L-system parameters“ se nacházejí tři pole pro zadávání číselných hodnot. První dvě hodnoty představují koeficienty pro dělení bodů uvnitř křivky a třetí hodnota je koeficient pro dělení krajních bodů. Pod popiskem „L-system depth“ je pole pro zadání počtu iterací L-systému, během kterých jsou vygenerovány body křivky. Pro informaci je pod tímto polem vypisován počet bodů, ze kterých se vygenerovaná křivka skládá.

Posledním ovládacím prvkem je tlačítko „Evaluate“. Po jeho stisknutí jsou opět vypočteny obě křivky, a následně vypočteno ohodnocení křivky vypočtené L-systémem (podle vztahu 6.1.1).

## 10.2 Program B-spline approximation

V tomto programu jsou křivky počítané podle řídicích bodů, jež jsou předem vygenerovány pomocí jednoduchého L-systému. Tyto křivky odpovídají aproximaci B-spline křivek z kapitoly 5. Vygenerovanou křivku je možné deformovat podle síly gravitace a podle teploty.



Obr. 10.2.1: Okno programu

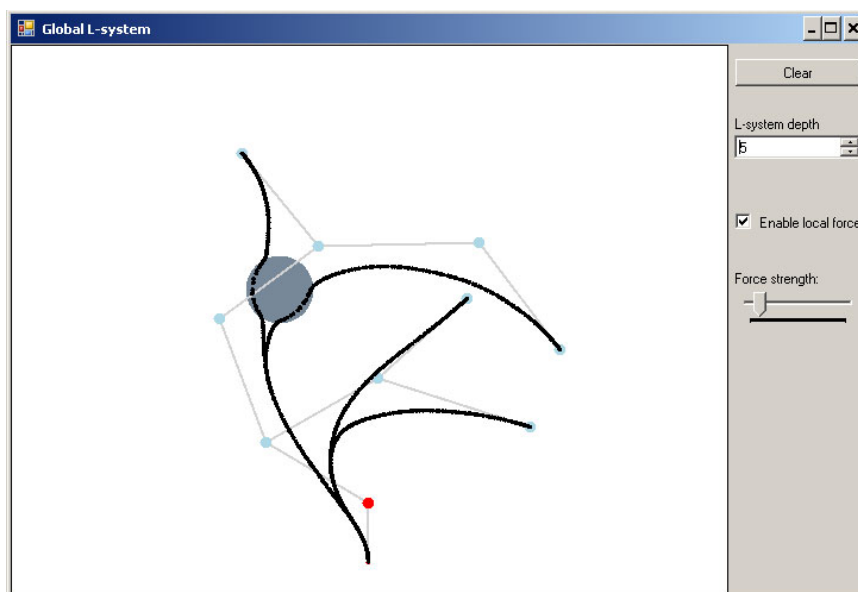
Ihned po spuštění programu je vidět křivka vygenerovaná podle základní konfigurace řídicích bodů. Tuto konfiguraci lze měnit třemi ovládacími prvky. Prvním je pole pro zadávání číselných hodnot pod popiskem „control point depth“. Tato hodnota udává počet iterací L- systému, který generuje řídicí body.

Posuvník s nadpisem „Gravity“ určuje sílu gravitace, která působí na křivku. Stejně tak posuvník s nadpisem „Temperature“ definuje hodnotu okolní teploty. Ihned po přenastavení kteréhokoliv prvku se křivka přizpůsobí novým hodnotám.

Posledním ovládacím prvkem je pole pro zadávání hodnoty „L-system depth“, která určuje počet iterací L-systému, jež generuje finální křivku (podle kapitol 5 a 6).

### 10.3 Program Global L-system

Program „Global L-system“ generuje křivky podle řídicích bodů zadaných uživatelem. Křivky nejsou aproximací B-spline, ale jsou přímo generovány L-systémem, který umožňuje také dělení křivek. Je zde také možnost deformovat křivku lokální silou.



Obr. 10.3.1: Okno programu

Po spuštění programu jsou vygenerovány dva základní body křivky, které jsou označeny červenou barvou. Bod vykreslený větší značkou slouží jako kořenový bod a z něj je možné vést další části křivky. Postup zadávání nových bodů byl popsán v kapitole 8.1.1.

Program má několik ovládacích prvků. Prvním z nich je tlačítko „Clear“, které po jeho stisknutí smaže právě vykreslenou křivku a nastaví řídicí body do základní konfigurace.

Pole s nadpisem „L-system depth“ slouží k zadávání počtu iterací pro L-systém, který generuje body křivky.

Posledními ovládacími prvky je zaškrtnuté pole „Enable local force“, po jehož aktivaci je možné deformovat křivku pomocí objektu, který reprezentuje působící lokální sílu. Velikost tohoto objektu lze měnit pomocí posuvníku „Force strength“.



## Literatura

- [1] PRUSINKIEWICZ, P. - SAMAVATI, F. - SMITH, C. - KARWOWSKI, R.  
L-system description of subdivision curves  
Department of Computer Science, The University of Calgary
  
- [2] ŽÁRA, J. - BENEŠ, B. - SOCHOR, J. - FELKEL, P.  
Moderní počítačová grafika
  
- [3] <http://cs.wikipedia.org>
  
- [4] <http://herakles.zcu.cz>
  
- [5] <http://iason.zcu.cz>