Programovacia úloha č. 2

(25b)

Téma: Bézierovo orezávanie

Cieľ: Cieľom druhej programovacej úlohy je vytvoriť aplikáciu riešiacu polynomické rovnice pomocou tzv. Bézierovho orezávania. Riešenie bude vyžadovať znalosti Bézierových kriviek a polárnej formy. Aplikácia zároveň vykreslí aj graf funkcie, pričom je možné použiť kód z predchádzajúcej programovacej úlohy.

Zadanie:

Vytvorte aplikáciu, ktorá umožní nájsť korene polynomickej funkciu na danom intervale použitím Bézierovho orezávania (napr. stlačením tlačidla, ktoré spustí algoritmus). Graf polynomickej funkcie môže byť vykreslený použitím metódy v predošlej úlohe. Vykreslený graf nie je potrebné interkatívne meniť. Vypočítané korene však musia byť vyznačené na grafe. Algoritmus pozostáva z nasledovných krokov:

Vstupné parametre: $\boxed{\textbf{7b}}$ Program vyžaduje zadanie polynómu a počiatočného intervalu. Počiatočný interval $\langle a, b\rangle$ je reprezentovaný dvoma reálnymi číslami a a b. Pre zadanie polynómu si môžte zvoliť ľubovoľný spôsob, avšak musíte používateľovi umožniť zadanie polynómu ľubovoľného stupňa.

Jedným z možných spôsobov, ako zadať aplikácii vstupný polynóm, je pomocou reťazca v nasledovnom formáte:

- Počet medzier v texte je nepodstatný (všetky medzery na začiatku spracovania odstráňte).
- Reťazec pozostáva z niekoľkých členov – monómov, oddelených aditívnym operátorom (`+`, `-`). Prvý monóm nemusí mať znamienko. Reťazec možno rozdeliť na jednotlivé členy nasledovným spôsobom. Nájdite prvý zo znakov `+` alebo `-`, ktorý nie je prvým znakom reťazca. Na ľavej strane od tohto znaku sa nachádza monóm, ktorý spracujte a z reťazca odstráňte. Tento postup opakujte, kým nespracujete celý reťazec.
- Monóm pozostáva z koeficientu (reálne číslo), písmena `t` a exponentu v tvare `^x`, kde `x` je prirodzené číslo. V prípade, že chýba koeficient, použije sa hodnota 1. V prípade, že chýba znak `t`, jedná sa o absolútny člen. Ak chýba len exponent, jedná sa o lineárny člen. Vhodný postup spracovania je hľadať znak `t`. Ak sa nenájde, konvertujte celý reťazec na reálne číslo (absolútny člen), inak rozdeľte reťazec na dve časti. Prvá je buď prázdna (koeficient nastavíme na 1), alebo obsahuje reálne číslo. Druhá je buď prázdna (exponent je 1), alebo obsahuje exponent (prirodzené číslo za znakom `^`).

Váš program nemusí byť schopný ošetriť všetky potenciálne chyby spojené s nekorektným vstupom. Korektné vstupy však musia byť správne spracované. Príklady vstupov sa nachádzajú na konci dokumentu.

Riadiace vrcholy: $\boxed{\textbf{5b}}$ Na aktuálnom pracovnom intervale (pri prvej iterácii sa pracuje so štartovacím intervalom $[a, b]$ zadaným používateľom) sa funkcia namodeluje Bézierovou krivkou. Zadaný polynóm reprezentuje y-ovú zložku krivky, x-ovou je rovnomerne parametrizovaný interval $[a,b]$. Nech $\mathbf{p}_y$ je polárna forma k nášmu polynómu. Potom hodnoty polárnej formy $\mathbf{p}_y$ pre rôzne kombinácie koncových bodov aktuálneho intervalu nám dávajú y-ové súradnice riadiacich vrcholov hľadanej krivky. Keďže naša krivka je vlastne funkciou, x-ová zložka je reprezentovaná polynómom t, ku ktorému prislúcha polárna forma $\mathbf{p}_x$.

Príklad: pre polynóm $\mathbf{p}(t) = 3t^3 + 2t^2 - 5t + 1$ máme polárnu formu (3. stupňa)

$$\mathbf{p}_y(t_1,t_2,t_3) = \frac{3}{\binom{3}{3}} t_1 t_2 t_3 + \frac{2}{\binom{3}{2}} \left(t_1 t_2 + t_1 t_3 + t_2 t_3 \right) - \frac{5}{\binom{3}{1}} \left(t_1 + t_2 + t_3 \right) + \frac{1}{\binom{3}{0}} 1.$$

Polárna forma pre x-ovú zložku je

$$\mathbf{p}_x(t_1,t_2,t_3) = \frac{1}{3}\left(t_1 + t_2 + t_3\right)$$

Riadiace vrcholy pre Bézierovu krivku reprezentujúcu zadaný polynóm na intervale $[a,b]$ sú

$$\begin{bmatrix} \mathbf{p}_x(a,\ldots,a,a) \\ \mathbf{p}_y(a,\ldots,a,a) \end{bmatrix}, \begin{bmatrix} \mathbf{p}_x(a,\ldots,a,b) \\ \mathbf{p}_y(a,\ldots,a,b) \end{bmatrix}, \ldots, \begin{bmatrix} \mathbf{p}_x(b,\ldots,b,b) \\ \mathbf{p}_y(b,\ldots,b,b) \end{bmatrix},$$

čo vieme triviálnym určením hodnôt funkcie $\mathbf{p}_x$ prepísať ako:

$$\begin{bmatrix} a + 0\frac{b-a}{n} \\ \mathbf{p}_y(a,\ldots,a,a) \end{bmatrix}, \begin{bmatrix} a + 1\frac{b-a}{n} \\ \mathbf{p}_y(a,\ldots,a,b) \end{bmatrix}, \ldots, \begin{bmatrix} a + n\frac{b-a}{n} \\ \mathbf{p}_y(b,\ldots,b,b) \end{bmatrix}.$$

Konvexný obal: $\boxed{\textbf{5b}}$ Z teórie vieme, že Bézierova krivka sa nachádza v konvexnom obale svojich riadiacich vrcholov. To znamená, že graf funkcie (a teda aj všetky hľadané korene polynómu) sa nachádza v tomto obale. Nájdeme teda prienik konvexného obalu a osi x a zúžime tak prehľadávaný interval.

Keďže dôležitý je len prienik konvexného obalu s osou x, miesto hľadania celého obalu sa stačí zaujímať len o priesečníky hrán konvexného obalu s osou x. Takýto priesečník majú len hrany, ktorých jeden koncový bod je nad osou x a druhý pod osou x. Náš nový interval je potom určený priesečníkmi s minimálnou a maximálnou hodnotou.

Iterácie: $\boxed{\textbf{5b}}$ Aplikovaním predchádzajúcich krokov sa nám zúžil prehľadávaný interval.
V tomto kroku analyzujeme niekoľko stavov:

- Dĺžka intervalu je menšia ako nejaké ε_{stop}. Interval je dostatočne malý a teda ho prehlásime za bod – hľadaný koreň. Funkcia hľadania v tomto bode končí. Aplikácia vypíše nájdený koreň.
- Dĺžka intervalu je väčšia ako ε_{stop} a od poslednej iterácie sa zmenšila o viac ako ε_{split}. V tomto prípade pokračujeme v iteráciach algoritmu s novým (menším) prehľadávaným intervalom.
- Dĺžka intervalu je väčšia ako ε_{stop} a od poslednej iterácie sa zmenšila o menej ako ε_{split}. Pravdepodobne sa v prehľadávanom intervale nachádza viacero koreňov. Interval teda rozdelíme na dva a rekurzívne necháme vyhľadať korene v ľavej aj pravej časti pôvodného intervalu.

Vykreslenie: $\boxed{\textbf{3b}}$ Aby mal používateľ predstavu o približnom tvare funkcie, vykreslíme jej graf. Je možné použiť kód z predchádzajúcej programovacej úlohy. Riadiace vrcholy krivky zistíme aplikovaním postupu z vyššie uvedeného bodu (**Riadiace vrcholy**) pre počiatočný interval. Keďže používateľom zadaný interval nezodpovedá rozmerom komponentu, do ktorého budeme krivku vykresľovať, treba vykonať transformáciu súradníc. Ak zadaný interval je $[a, b]$ a šírka komponentu w, tak bod $p \in [a, b]$ bude mať obraz v bode $p' = \frac{p-a}{b-a} \cdot w$. Pre y-ovú súradnicu platí podobný vzorec. Nájdené korene zvýraznite.

Komentár k implementácii: Výsledná aplikácia môže mať problém nájsť niektoré korene, resp. môže niektoré korene nájsť viackrát. Správanie závisí od použitých konštánt ε a od implementovania operácií. Snažte sa aplikáciu vytvoriť tak, aby pri rovnakých nastaveniach, aké sú vo vzorovej aplikácii, dávala rovnaké výsledky.

Vzorová aplikácia používa nasledovné nastavenia, resp. postupy: $\varepsilon_{stop} = 0.0001$, $\varepsilon_{split} = 0.000001$. Pri každej iterácií algoritmu sa v prípade, že nedôjde k jeho vetveniu ani ukončeniu, upravia nové koncové body intervalu tak, že sa interval na každom konci rozšíri o $\varepsilon_{stop}/4$ (aby nám náhodou pri nepresnostiach s reálnymi číslami nepreskočil koreň). Ak sa interval delí na 2 časti, ľavej zodpovedá interval $[a, (a + b)/2]$, pravej interval $[(a + b)/2 + \varepsilon_{stop}/2, b]$. Korene nájdené viackrát sa na konci dajú odstrániť – prechodom usporiadaného zoznamu koreňov vieme identifikovať (a následne odstrániť) tie, ktoré sú od predchádzajúceho vzdialené o menej ako ε_{split}.

Korektnosť implementácie môžete otestovať (porovnaním so vzorovou aplikáciou) na nasledovných polynómoch, ktoré aproximujú funkcie *sin* a *cos* pomocou Taylorovho radu:

```
sin_7(t) = t - 0.16667t^3 + 0.00833t^5 - 0.00019t^7
```

na intervale $[-3.5, 3.5]$ a

```
cos_8(t) = 1 - 0.5t^2 + 0.04166t^4 - 0.00138t^6 + 0.00002t^8
```

na intervale $[-5.0, 5.0]$.

Výstup:

Kód musí byť **dostatočne** komentovaný a **prehľadne** formátovaný. Nedostatočné komentáre a neprehľadné formátovanie môže byť penalizované stratou bodov.

Použitie výlučne externých knižníc je zakázané.

Vzorová aplikácia je dostupná na MS Teams.