Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Interaktívne vyšetrovanie priebehu elementárnych funkcií

Bakalárska práca

2020
Juraj Vetrák

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Interaktívne vyšetrovanie priebehu elementárnych funkcií

Bakalárska práca

Študijný program:	Aplikovaná informatika
Študijný odbor:	2511 Aplikovaná informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	Ing. Ján Komara, PhD.

Bratislava, 2020
Juraj Vetrák

23814387

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Juraj Vetrák
Študijný program:	aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor:	informatika
Typ záverečnej práce:	bakalárska
Jazyk záverečnej práce:	slovenský
Sekundárny jazyk:	anglický

Názov: Interaktívne vyšetrovanie priebehu elementárnych funkcií
Properties of Elementary Functions Interactively

Anotácia: Návrh, vývoj a implementácia editora pre interaktívne vyšetrovanie priebehu elementárnych funkcií vo výpočtovom prostredí IPython/Jupyter.

Vedúci:	Ing. Ján Komara, PhD.
Katedra:	FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry:	prof. Ing. Igor Farkaš, Dr.
Dátum zadania:	20.09.2019
Dátum schválenia:	07.10.2019

doc. RNDr. Damas Gruska, PhD.
garant študijného programu

....
študent vedúci práce

Poďakovanie: ...

Abstrakt

...

Kľúčové slová: ...

Abstract

...

Keywords:

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

TODO

1 Vyšetrovanie priebehu funkcií

V tejto kapitole bude uvod do problematiky, akysi matematicky popis toho, co editor umoznuje graficky vizualizovat.

2 Východiská

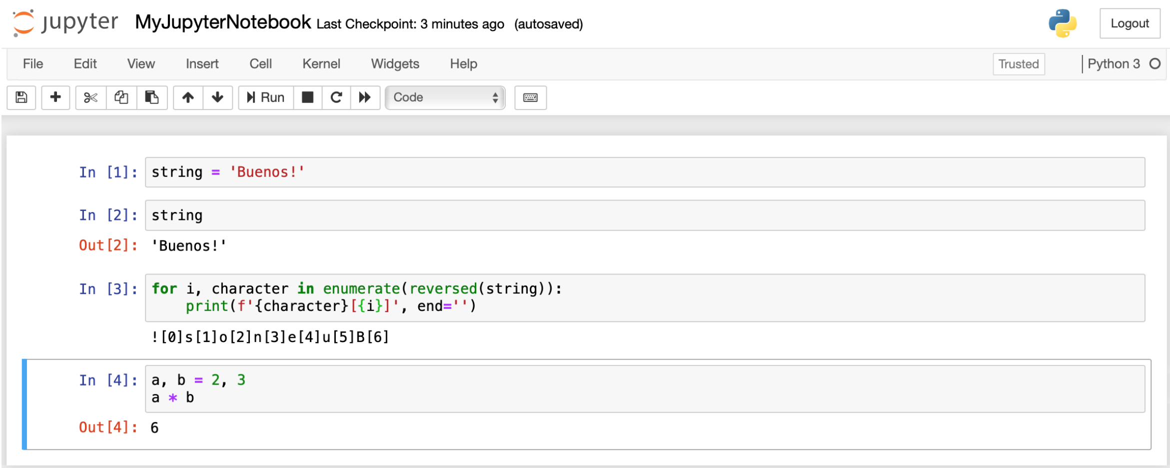
Pred navrhnutím a implementáciou riešenia popisovaného touto bakalárskou prácou je dôležité objasniť východiskový stav pred začatím práce, teda popísať uvažované technológie, programy a knižnice, v ktorých bude práca vyvíjaná (sekcia 2.1), zároveň popísať základnú, v tomto prípade najmä matematickú teóriu, ktorá objasní funkcionalitu niekoľkých algoritmov použitých pre celkovú funkčnosť riešenia (sekcia 2.2). V sekcii 2.3 je uvedených niekoľko existujúcich riešení problematiky v podobe verejne dostupných open-source programov. V závere tejto kapitoly je popísaná podobná práca, ktorá bola obhájená v roku 2017, pričom je uvedená aj krátka komparácia s touto bakalárskou prácou (sekcia 2.4).

2.1 Technologické východiská

Editor, všetky jeho funkcionality a výpočtové procesy sú naprogramované v programovacom jazyku Python. Jednou z výhod tohto jazyka je, že ho používa početná komunita vývojárov v rôznych vedeckých oblastiach, ako matematika, dátová veda, či umelá inteligencia. To podporilo vznik mnohých nástrojov a knižníc, ktoré uľahčujú každodennú prácu v spomínaných oblastiach. Niektoré z týchto nástrojov a knižníc tvoria práve jadro tejto bakalárskej práce.

2.1.1 Jupyter Notebook

Okrem programovacieho jazyka Python túto prácu charakterizuje prostredie Jupyter Notebook (obr. 2.1), do ktorého je implementovaná a v ktorom sa používa. Jupyter Notebook autorka praktickej cvičebnice *IPython Cookbook* [11] charakterizuje ako webové interaktívne prostredie, ktoré kombinuje kód, textové prvky, obrázky, videá, animácie, matematické rovnice, grafy, mapy, interaktívne prvky, widgety a grafické užívateľské rozhranie do jediného dokumentu. Spustenie tohto prostredia je realizované príkazom `jupyter notebook` v príkazovom riadku. Práca s týmto nástrojom však nie je limitovaná použitím programovacieho jazyka Python, rôzne numerické výpočty môžu byť totiž realizované aj v iných populárnych jazykoch dátovej vedy, ako Julia, či R. O spúšťanie užívateľského kódu a rôzne numerické výpočty sa starajú programy zvané IPython kernely. IPython je projekt, ktorý vznikol s cieľom poskytnúť nie len značne vylepšený Python shell, ale aj možnosti pre interaktívne, distribuované a paralelné výpočty. Na podobné účely slúžia známe komerčné prostredia ako MATLAB, The Interactive Data Language pre numerické výpočty, Mathematica a Maple pre manipuláciu so symbolmi, či ich otvorené alternatívy GNU Data Language, Octave, Maxima alebo Sage [10].



Obr. 2.1: Interaktívne webové prostredie Jupyter Notebook

2.1.2 NumPy, SciPy a matplotlib

Prostredie Jupyter Notebook umožňuje využiť výpočtové operácie jazyka Python, no v mnohých prípadoch, ako aj v prípade tejto bakalárskej práce, je zoznam týchto operácii nedostatočný a je potrebné ho rozšíriť o ďalšie numerické operácie. Knižnica NumPy, menej známa ako Numerical Python, je jedným z najdôležitejších balíkov v rámci numerického počítania v jazyku Python.

Medzi príklady funkcionalít knižnice NumPy, ktoré uvádza autor knihy *Python for Data Analysis* [9] a sú kľúčové pre túto prácu, patrí:

- ndarray, efektívne viacrozmerné pole umožňujúce na ňom vykonávať rýchle aritmetické (vektorizované) operácie
- Matematické funkcie pre rýchle operácie nad poliami dát bez potreby písania cyklov
- Podmienený výber dát priamo ako argument polí, namiesto vetiev if-elif-else
- Algoritmy na týchto poliach ako triedenie, výber unikátnych hodnôt, či operácie s množinami

Autor ďalej vysvetľuje, prečo je knižnica taká dôležitá pre prácu s numerickými výpočtami. Knižnica NumPy je totiž navrhnutá na efektívne zvládanie veľmi veľkých polí dát. Aby to dosiahla, ukladá dáta v súvislých blokoch pamäte, nezávislé od vstavaných objektov jazyka Python. Algoritmy v NumPy sú taktiež napísané v jazyku C a vedia pracovať so spomínanou pamäťou bez vyčerpávajúcej kontroly typov údajov. Ďalším dôvodom, hoci už naznačeným je, že NumPy vo väčšine prípadov obchádza štandardné for cykly jazyka Python a namiesto nich vykonáva komplexné výpočty na celých poliach naraz pomocou paralelizmu.

Špecifickým dôvodom na použitie knižnice NumPy je aj využitie ďalšej knižnice, SciPy, ktorá je na NumPy priamo postavená, ako uvádza jej oficiálna dokumentácia [12]. SciPy, alebo aj Scientific Python, je knižnica, alebo aj kolekcia matematických algoritmov a určených funkcií. Pre potreby tejto práce poskytuje algoritmy na výpočet derivácií, či nulových bodov. S knižnicou SciPy sa Python a prostredie Jupyter Notebook stávajú plnohodnotným konkurentom spomínaných riešení ako MATLAB, The Interactive Data Language, či Octave. V nasledovnom príklade kódu ukážme jednoduchý výpočet prvej derivácie v bode použitím knižnice SciPy.

Jednou z najpodstatnejších funkcií tejto bakalárskej práce je vizuálna interakcia s užívateľom. Funkcie a jej vlastnosti sa vizualizujú užívateľovi v podobe grafu a v rámci možností v čo najlepšej kvalite. Takéto vizualizačné funkcionality dodáva knižnica matplotlib. matplotlib je Python knižnica pre vytváranie a vizualizáciu najmä 2D grafov v produkčnej kvalite. Podporuje interaktívnu, aj neinteraktívnu vizualizáciu a umožňuje ukladať tieto vizualizácie vo forme obrázkov rozličných formátov (vektorových aj rastrových) typu JPG, PNG, PDF, PS a iných [9, 14].

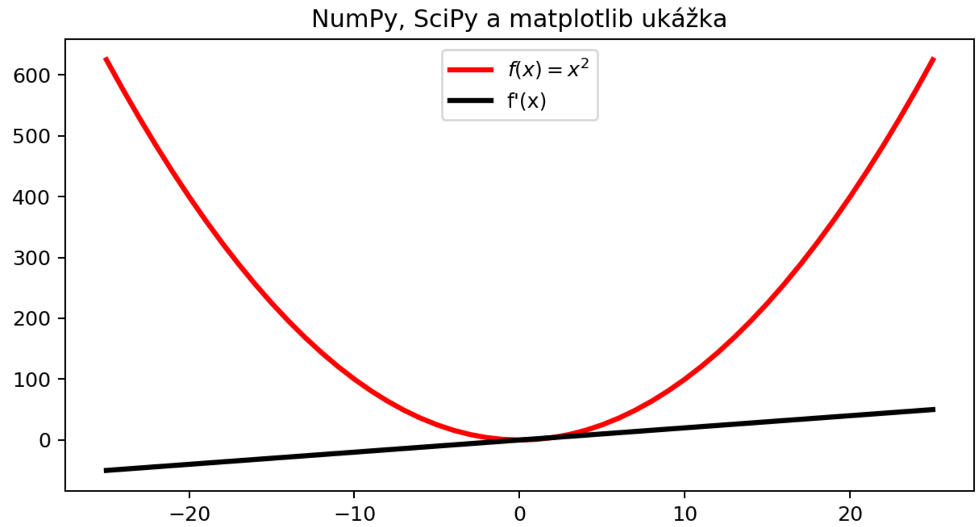
Nasledovný kód slúži na demonštráciu práce s knižnicami NumPy, SciPy a matplotlib. V kóde sa zadefinuje funkcia $f(x) = x^2$ a horizontálne koordináty $x \in \langle -25, 25 \rangle$ v podobe premennej `X`. Funkčné hodnoty (`Y`) a derivácie (`primes`) na celom intervale sa vypočítajú pomocou vektorizovaných funkcií knižníc NumPy, resp. SciPy. Cez modul pyplot knižnice matplotlib sa vypočítané hodnoty vykreslia do podoby grafu. (obr. 2.2).

```
In [1]: %matplotlib notebook
        import matplotlib.pyplot as plt
        import numpy as np
        from scipy.misc import derivative

        def f(X): return X ** 2

        X = np.arange(-25, 25+1)
        Y, primes = f(X), derivative(f, X)
        fig, ax = plt.subplots()
```

```
fig.set_size_inches(8, 4)
ax.plot(X, Y, c='red', linewidth=2.5, label=r"$f(x) = x^2$")
ax.plot(X, primes, c='black', linewidth=2.5, label=r"f'(x)")
ax.set_title('NumPy, SciPy a matplotlib ukazka')
ax.legend()
fig.show()
```



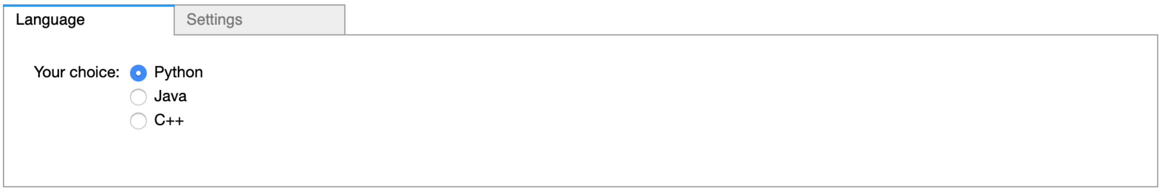
Obr. 2.2: Demonštrácia použitých knižníc - výstup

2.1.3 ipywidgets

Editor ako softvér umožňuje interakciu užívateľa so zvyšnými časťami programu prostredníctvom myši, rôznych tlačidiel, textových vstupov, či iných ovládacích prvkov. Editor funguje striktne v prostredí Jupyter Notebook, a preto sú možnosti prispôsobenia grafického užívateľského prostredia obmedzené. Potrebná časť upraviteľných ovládacích prvkov pre Jupyter Notebook je však dostupná pomocou rozšírenia ipywidgets. Ide o sadu HTML elementov ako tlačidlá, textové vstupy, checkboxy, obrázky, či animácie. Tieto elementy sa môžu združovať v rôznych rozloženiach ako okná, taby, či mriežky. V bunke Jupyter Notebooku je tak možné vytvoriť a spustiť vlastnú grafickú interaktívnu aplikáciu (obr. 2.3).

2.1.4 Anaconda

Dostupnosť a manažment všetkých potrebných knižníc pre túto prácu, vrátane distribúcie samotného jazyka Python a prostredia Jupyter Notebook, zariaďuje programový balík Anaconda. Programový balík Anaconda je zadarmo a po nainštalovaní slúži okrem už uvedeného aj na spravovanie virtuálnych prostredí, poskytuje



Obr. 2.3: Príklad interaktívnej aplikácie v prostredí Jupyter Notebook

programy a nástroje pre distribúciu programovacieho jazyka pre štatistiku R a obsahuje vyše 7500 ďalších open source riešení najmä pre dátovú vedu [1].

2.2 Teoretické východiská

Pri tvorbe tejto bakalárskej práce je potrebné prispôsobiť zložitosť funkcionalít úrovni kurzu *Matematická analýza I*, v ktorom je uvažované využívanie jej programovej časti. Od toho sa odvíja úroveň matematickej teórie potrebnej k pochopeniu výpočtov na pozadí práce. Kurz v rámci vyšetrovania priebehu elementárnych funkcií postupne prechádza od definície reálnej funkcie o reálnej premennej, cez definíciu monotónnosti, derivácie, extrémov, až po konvexnosť, konkávnosť a inflexné body. Je potrebné presne zadefinovať tieto, ako aj iné pojmy, poprípade doplniť pojmy, ktoré sa v osnovách predmetu priamo nenachádzajú, ako napr. Newtonová metóda. Všetky tieto pojmy sú v práci využívané, či už priamo, v rámci definovania funkcií, alebo na zabezpečenie algoritmickej funkcionality. Ak nie je uvedené inak, definície a vety pochádzajú z vysokoškolských skrípt *Cvičenia z matematickej analýzy I* od autorov Kubáček a Valášek [6].

2.2.1 Funkcia, graf funkcie a elementárne funkcie

V tejto časti sú okrem pojmov funkcia, funkčná hodnota, definičný obor funkcie a graf funkcie definované aj elementárne funkcie. Uvedené sú jednotlivé druhy základných elementárnych funkcií, s ktorými sa môže študent stretnúť na kurze *Matematická analýza I* a ktoré slúžili ako predloha, či vstupy pri tvorbe tejto bakalárskej práce.

Definícia. *Nech $A \subset \mathbb{R}$ je neprázdna množina. Ak je každému číslu $x \in A$ priradené práve jedno číslo $y \in \mathbb{R}$, ktoré označíme $f(x)$, hovoríme, že f je* ***funkcia (funkcia definovaná na množine A)****. Číslo $f(x)$ sa nazýva* ***funkčná hodnota (v bode x)*** *a množina A* ***definičný obor funkcie f****, označujeme aj $D(f)$.*

Definícia. *Nech je v rovine daná pravouhlá súradnicová sústava, pričom jednotky dĺžky na súradnicových osiach Ox a Oy sú rovnaké. Množina $\{(x, f(x)) \mid x \in A\}$ bodov roviny, kde*

$f : A \to \mathbb{R}$ *je daná funkcia, sa nazýva* ***graf funkcie f****, pričom* $(x, f(x))$ *je zápis bodu roviny pomocou jeho súradníc v danej súradnicovej sústave.*

Definícia. *Funkcie, ktoré vzniknú zo základných elementárnych funkcií len použitím operácií súčtu, rozdielu, súčinu, podielu a superpozície funkcií, sa nazývajú* ***elementárne funkcie****. Medzi základné elementárne funkcie patria funkcie konštantné, mocninové, exponenciálne, logaritmické, goniometrické a cyklometrické.*

2.2.2 Rastúca, klesajúca a rýdzomonotónna funkcia

Definícia. *Nech* $A, B \subset \mathbb{R}$ *sú neprázdne množiny, kde* $B \subset A$*. Funkcia* $f : A \to \mathbb{R}$ *sa nazýva* ***rastúca na množine B****, ak platí*

$$\forall\, x, y \in B\,;\, x < y \Rightarrow f(x) < f(y).$$

Definícia. *Nech* $A, B \subset \mathbb{R}$ *sú neprázdne množiny, kde* $B \subset A$*. Funkcia* $f : A \to \mathbb{R}$ *sa nazýva* ***klesajúca na množine B****, ak platí*

$$\forall\, x, y \in B\,;\, x < y \Rightarrow f(x) > f(y).$$

Definícia. *Funkcia, ktorá je rastúca na množine B alebo klesajúca na množine B sa nazýva* ***rýdzomonotónna*** *na množine B.*

Definícia. *Funkcia rastúca (klesajúca, rýdzomonotónna) na svojom definičnom obore sa nazýva* ***rastúca (klesajúca, rýdzomonotónna)****.*

2.2.3 Derivácia funkcie

Pojem derivácie funkcie zastupuje kľúčovú funkcionalitu v rámci tejto bakalárskej práce. Okrem toho, že patrí do osnov predmetu *Matematická analýza I*, je derivácia využívaná na výpočet kľúčových hodnôt pri vyšetrovaní vlastností funkcie pomocou diferenciálneho počtu.

Definícia. *Hovoríme, že funkcia f definovaná v okolí vnútorného bodu* $a \in \mathbb{R}$*, t.j. pre niektoré* $\epsilon > 0$ *platí* $(a - \epsilon, a + \epsilon) \subset D(f)$*,* ***má deriváciu v bode a****, ak existuje konečná* $\lim_{x \to a} \frac{f(x) - f(a)}{x - a}$*. Hodnotu* $\lim_{x \to a} \frac{f(x) - f(a)}{x - a}$ *označujeme* $f'(a)$*.*

Definícia. *Nech M je množina všetkých bodov definičného oboru* $D(f)$*, v ktorých má funkcia f deriváciu. Funkcia* $f' : M \to \mathbb{R}$*, ktorá každému bodu* $a \in M$ *priradí hodnotu* $f'(a)$ *derivácie funkcie f v bode a, sa nazýva* ***derivácia funkcie f****.*

2.2.4 Derivácie vyšších rádov

V nasledujúcej sekcii 2.2.5 je uvedených niekoľko viet, ktoré slúžili ako podklad pre tvorbu algoritmov na vyšetrovanie niektorých vlastností funkcie pomocou diferenciálneho počtu. V niektorých z týchto viet sa spomínajú derivácie vyšších rádov, preto je dôležité si ich korektne zadefinovať.

Definícia. *Definície vyšších rádov definujeme rekurentne: Nech funkcia f je definovaná v okolí vnútorného bodu $a \in \mathbb{R}$, označme $f^{(0)} := f$. Hovoríme, že* ***funkcia f má n-tú deriváciu v bode a****, ak existuje derivácia funkcie $f^{(n-1)}$ v bode a. n-tú deriváciu v bode a označujeme $f^{(n)}(a)$.*

Definícia. *Derivácia funkcie $f^{(n-1)}$ sa nazýva* ***n-tá derivácia funkcie f*** *a označuje sa $f^{(n)}$. Okrem označení $f^{(1)}, f^{(2)}, f^{(3)}, f^{(4)}, f^{(5)}, \ldots$ sa používajú aj označenia f', f'', f''', f'''' (alebo f^{IV}), $f^{V}, \ldots$.*

2.2.5 Vyšetrovanie monotónnosti funkcie

Veta. *Nech funkcia $f : I \to \mathbb{R}$ je spojitá na intervale I a má deriváciu v každom jeho vnútornom bode. Potom*

1. *ak $f' > 0$ vo vnútri intervalu I (t.j. pre každý vnútorný bod x z intervalu I platí, že $f'(x) > 0$), tak f je* ***rastúca na I****;*
2. *ak $f' < 0$ vo vnútri intervalu I (t.j. pre každý vnútorný bod x z intervalu I platí, že $f'(x) < 0$), tak f je* ***klesajúca na I****.*

2.2.6 Vyšetrovanie konvexnosti, konkávnosti a inflexných bodov funkcie

Veta. *Nech funkcia $f : I \to \mathbb{R}$ je spojitá na intervale I a dvakrát diferencovateľná v každom jeho vnútornom bode. Potom*

1. *ak $f'' > 0$ vo vnútri intervalu I, tak f je* ***rýdzo konvexná na I****;*
2. *ak $f'' < 0$ vo vnútri intervalu I, tak f je* ***rýdzo konkávna na I****.*

Definícia. *Vnútorný bod a množiny $D(f)$ sa nazýva* ***inflexný bod funkcie f****, ak má v bode a deriváciu a existuje $\epsilon > 0$ tak, že funkcia f je rýdzo konvexná na jednej z množín $(a-\epsilon, a\rangle$, $\langle a, a+\epsilon)$ a rýdzo konkávna na druhej z nich.*

Veta. *Nech funkcia f je trikrát diferencovateľná v bode a a dvakrát diferencovateľná v niektorom jeho okolí. Ak $f''(a) = 0$ a $f'''(a) \neq 0$, tak a je* ***inflexný bod funkcie f****.*

2.2.7 Vyšetrovanie extrémov funkcie

Definícia. *Nech definičným oborom funkcie f je interval I. Hovoríme, že funkcia f má v bode $a \in I$* ***ostré lokálne maximum (ostré lokálne minimum)****, ak existuje okolie $O(a)$ bodu a tak, že platí*

$$\forall x \in (O(a) \setminus \{a\}) \cap I : f(x) < f(a), \text{ resp. } \forall x \in (O(a) \setminus \{a\}) \cap I : f(x) > f(a).$$

Ostré lokálne maximá a ostré lokálne minimá sa súhrnne nazývajú ***ostré lokálne extrémy****.*

Veta. *Ak funkcia f je dvakrát diferencovateľná vo vnútornom bode a množiny $D(f)$ a platí*

$$f'(a) = 0 \text{ a } f''(a) > 0, \text{ resp. } f'(a) = 0 \text{ a } f''(a) < 0,$$

tak f má v bode a ostré lokálne minimum (ostré lokálne maximum).

Veta. *Nech funkcia f je n-krát $(n \geq 2)$ diferencovateľná vo vnútornom bode a množiny $D(f)$, nech $f'(a) = ... = f^{(n-1)}(a) = 0$ a $f^{(n)}(a) \neq 0$.*

1. *Ak n je párne a $f^{(n)}(a) > 0$, resp. $f^{(n)}(a) < 0$, tak funkcia f má v bode a ostré lokálne minimum (ostré lokálne maximum).*

2. *Ak n je nepárne, nemá funkcia f v bode a lokálny extrém.*

2.2.8 Absolútna a relatívna chyba približnej hodnoty čísla

...

2.2.9 Zaokrúhľovanie približných hodnôt čísel

...

2.2.10 Hľadanie nulových bodov funkcie

Súčasťou práce sú aj tri metódy na aproximáciu nulových bodov funkcie. Metódy sa rozlišujú podľa využitia prvej a druhej derivácie funkcie, a to nasledovne:

- ak sa nevyužíva prvá ani druhá derivácia funkcie, nulové body funkcie sú aproximované **metódou sečníc**

- ak sa využíva len prvá derivácia funkcie, nulové body funkcie sú aproximované **Newtonovou metódou**

- ak sa využíva prvá aj druhá derivácia funkcie, nulové body funkcie sú aproximované **Halleyho metódou**

Nasledujúce odseky vysvetľujú princípy spomenutých metód hľadania (aproximovania) nulových bodov funkcie.

Newtonova metóda

Newtonová metóda, inak známa aj ako metóda dotyčníc, je podľa [2] najznámejšou procedúrou pre nájdenie koreňov rovnice. Nie je vždy najvhodnejšou metódou na riešenie určeného problému, no jej formálna jednoduchosť a pomerne veľká rýchlosť znamená, že si ju vyberie väčšina ľudí pri pokuse vyriešiť nejaký nelineárny problém.

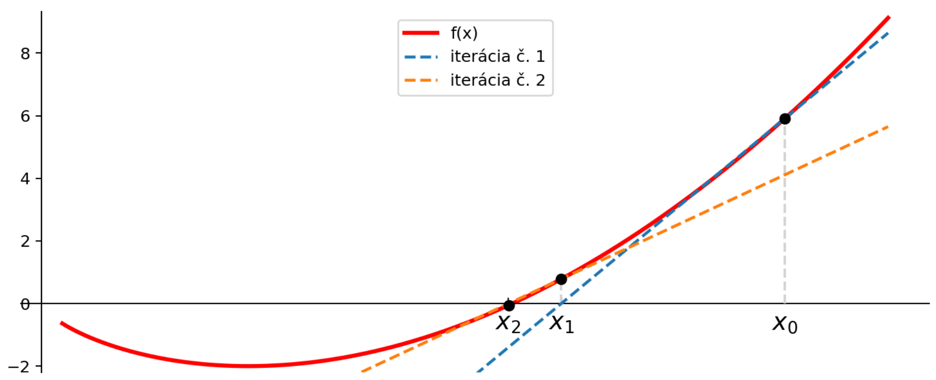
Nech x_0 je počiatočný odhad pre uvažovaný koreň α rovnice $f(x) = 0$. Newtonová metóda má za úlohu nájsť postupnosť čísel $\{x_n : n \geq 1\}$, ktorá by mala postupne konvergovať k bodu α. Hľadanie tejto postupnosti prebieha nasledovne: pri predpoklade, že x_0 je dostatočne blízko α, metóda aproximuje graf funkcie $y = f(x)$ v okolí bodu α, a to tak, že v bode $(x_0, f(x_0))$ skonštruuje dotyčnicu ku grafu funkcie. Koreňom tejto dotyčnice je potom nová aproximácia α, označená x_1. Postupným opakovaním tohto postupu pre

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad n \geq 1$$

sa metóda dopracuje k spomínanej postupnosti čísel x_n. Postup prvých dvoch iterácií na príklade funkcie

$$f(x) = x^2 - 3\sqrt[3]{x^2}$$

je znázornený obrázkom 2.4 s aproximovanými bodmi x_1 a x_2. Tabuľka 2.1 zase ukazuje namerané hodnoty pri rovnakej funkcii, a to pri štyroch iteráciách.



Obr. 2.4: Prvé dve iterácie Newtonovej metódy

Zo štvrtého stĺpca tabuľky 2.1 je názorne vidieť, ako sa s pribúdajúcimi iteráciami n zmenšuje odchýlka x_n od uvažovaného koreňa α, dopočítaného programom WolframAlpha (`www.wolframalpha.com`) a rovnako odchýlka jednotlivých

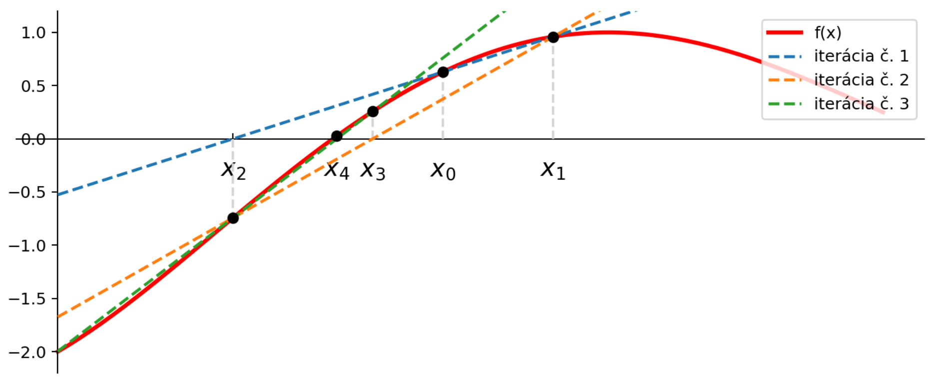
n	x_n	$f(x_n)$	$\alpha - x_n$	$x_{n+1} - x_n$
0	3.6	5.91323825	-1.32049294	-1.082999
1	2.517001	0.78422221	-0.23749394	-0.2551462
2	2.2618548	-0.05330492	0.01765226	0.0213885
3	2.2832433	0.01137122	-0.00373624	-0.00447446
4	2.27876884	-0.00224309	0.00073822	

Tabuľka 2.1: Vypočítané hodnoty prvých štyroch iterácií Newtonovej metódy

aproximácií x_n v poslednom stĺpci. Newtonova metóda na príklade konverguje naozaj veľmi rýchlo, a to hneď, ako sa aproximovaný bod x_n blíži ku koreňu. Väčšie odchýlky pri prvej, resp. druhej iterácií sú dôsledkom zvolenia nie príliš presného počiatočného odhadu bodu x_0. Z toho vyplýva, že posúvaním tohto počiatočného odhadu bodu x_0 ku skutočnému koreňu je možné ešte viac urýchliť konvergenciu Newtonovej metódy.

Metóda sečníc

...



Obr. 2.5: Prvé tri iterácie metódy sečníc

...

Halleyho metóda

...

2.3 Existujúce riešenia

Editor, ako výstup tejto práce, využíva vety a definície zo sekcie 2.2. Zaobaľuje ich do formy funkcií, či algoritmov, ktoré riadia funkcionalitu programu. Tieto funkcionality, spolu s vizualizáciou grafov je možné v rôznych podobách implementovať aj v iných, verejne dostupných nástrojoch. V tejto sekcii je popísaný jeden komerčný a dva open-source nástroje, ktoré môžu slúžiť ako prípadná alternatíva pre študenta analyzujúceho priebeh elementárnych funkcií.

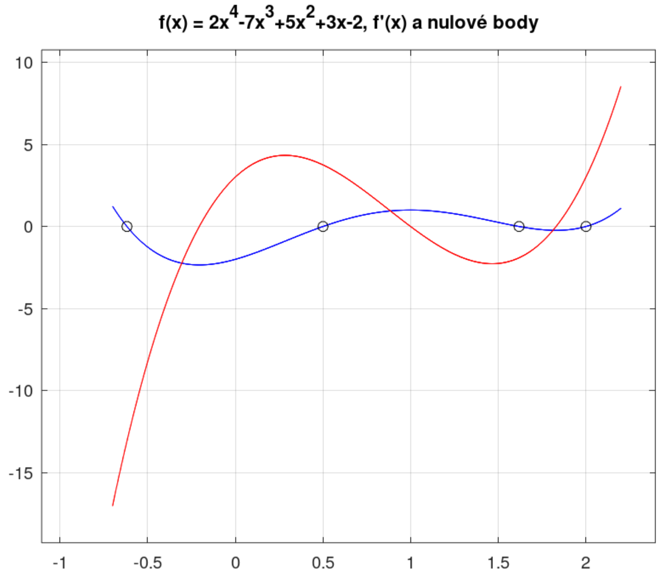
2.3.1 MATLAB

MATLAB je vysoko-úrovňový jazyk a interaktívne prostredie pre numerické výpočty, vizualizáciu a programovanie. Umožňuje analyzovať dáta, vyvíjať algoritmy, či vytvárať modely a aplikácie. Jazyk, jeho nástroje a zabudované matematické funkcie umožňujú zvoliť mnoho prístupov pri riešení problému a dosiahnuť toto riešenie rýchlejšie v porovnaní s hárkami tabuľkového procesora, či programovacími jazykmi C/C++ alebo Java. MATLAB je využívaný v mnohých aplikáciách ako spracovanie signálu, obrazu, či videa, kontrolné systémy, systémy pre testovanie a meranie, finančníctvo alebo biológia [8]. Používanie softvéru MATLAB vyžaduje platenú licencie.

2.3.2 GNU Octave

GNU Octave je tiež vysoko-úrovňový jazyk a interaktívne prostredie pre numerické výpočty, pričom v zmysle účelu tejto práce poskytuje rovnaké možnosti ako MATLAB, no je distribuované pod GNU licenciou. Licencia GNU používateľovi umožňuje využívať tento softvér zadarmo v plnom rozsahu, šíriť ho a dokonca aj prípadne upravovať. Obidva jazyky sú numerické, nie symbolické a ich základným typom je matica, pričom softvér je plne optimalizovaný pre vektorové operácie [7]. Nasledovný príklad demonštruje hľadanie nulových bodov a derivácie polynomiálu $f(x) = 2x^4 - 7x^3 + 5x^2 + 3x - 2$ v prostredí GNU Octave, pričom kód je aplikovateľný aj v prostredíi MATLAB.

```
> x = [-0.7:0.001:2.2];
> p = [2, -7, 5, 3, -2];
> y = polyval(p, x);
> dydx = diff(y) ./ diff(x);
> r = roots(p);
  r = 2.00000, 1.61803, -0.61803, 0.50000
> plot(x, y, 'b', x(1:end-1), dydx, 'r', r, 0, 'ko')
> title("f(x) = 2x^4-7x^3+5x^2+3x-2, f'(x) a nulove body")
```



Obr. 2.6: Demonštrácia MATLAB/Octave - výstup

Práca s nástrojmi popísanými v tejto časti je najmä z hľadiska funkcionalít porovnateľných s touto bakalárskou prácou podobná a vzhľadom na ich robustnosť aj flexibilnejšia. To však predurčuje užívateľa k väčšej znalosti jednotlivých jazykov a ovládacích procesov, čím sa komplikuje prístup k jednoduchým prvkom matematickej analýzy. Na druhej strane editor, ako predmet tejto bakalárskej práce, obsahuje nevyhnutné minimum funkcionalít a grafických výstupov potrebných k splneniu edukačných potrieb študenta, pričom mu k nim umožňuje pristupovať interaktívne, cez grafické rozhranie.

2.4 Podobné práce

Táto bakalárska práca sa dá považovať ako priama alternatíva pre diplomovú prácu Jakuba Trubača, *Vývoj interaktívnych dokumentov pre výučbu matematickej analýzy*, ktorá bola vypracovaná pod vedením rovnakého školiteľa. Práca tak vychádza z pomerne rovnakých základov. Autor vytvoril a popísal rozšírenie funkcionalít interaktívneho prostredia Jupyter Notebook vo forme editora. Tento editor bol vytvorený za účelom efektívnejšej analýzy priebehu funkcií na cvičeniach k predmetu Matematická analýza. Editor na vstupe spracuje užívateľom zadané hodnoty, vykreslí graf funkcie a umožní nájsť nulové body, extrémy funkcie, inflexné body, či s tým spojené vlastnosti ako monotónnosť, konkávnosť, alebo konvexnosť. Jedným z najdôležitejších parametrov je metóda na hľadanie nulových bodov. Zvolil Newtonovu metódu, ktorá iteratívnym procesom vyberie vhodného kandidáta na nulový bod. Na vstupe očakáva jeden štartový bod, akýsi počiatočný tip, ktorý si môže užívateľ interaktívne meniť a dopracovať sa tak k prípadným presnejším výsledkom.

Z hľadiska používateľského rozhrania Trubač v jeho práci využíva knižnicu ipywidgets, ktorá ma na starosti interaktívne ovládacie prvky v prostredí Jupyter Notebook. Po spracovaní užívateľských hodnôt a vykreslení grafu sa pod ním zobrazia ovládacie prvky v podobe posuvných tlačidiel na výber hodnôt, poprípade označovacích boxov. Editor má viacero okien, medzi nimi okno na samotnú analýzu, okno na prispôsobenie Newtonovej metódy a okno na výstupné informácie.

Hlavný rozdiel medzi prácami je práve v prístupe k Newtonovej metóde, zatiaľ čo *Trubač* túto metódu naprogramoval, v tejto práci sa využije metóda z už naprogramovanej knižnice scipy. To umožňuje využiť silu knižnice na numerickú matematiku numpy a prijať ako argumenty celé vektory hodnôt, nie len jediný bod - skalár. Práca s vypočítanými údajmi a ich spracovanie je tak odlišné vzhľadom na túto vlastnosť. Rozdiely v grafickom prevedení a ovládaní sú v princípe minimálne. V tejto práci sa graf funkcie nachádza priamo v editore, čím tvorí jeho značnú časť a všetky hlavné ovládacie prvky sa nachádzajú vedľa neho. Doplnené sú taktiež výstupy, avšak aj v podobe zaznamenávania užívateľských krokov, čím je užívateľ sprevádzaný celou analýzou funkcie. Rovnako je vynechaná možnosť dodatočne špecifikovať interval analýzy, v tomto prípade si interval merania určí užívateľ jeho definovaním ešte pred samotným spustením editora.

3 Návrh riešenia

...

4 Implementácia

...

5 Testovanie a používateľská príručka

...

Záver

TODO

Literatúra

[1] *Anaconda Distribution Documentation*. URL: `docs.anaconda.com/anaconda` (citované: 11.2.2020).

[2] ATKINSON, Kendall E. *An Introduction to Numerical Analysis*. Wiley, New York, second edition, 1989.

[3] ČERNÝ, Ilja. *Úvod do inteligentního kalkulu (1000 příkladů z elementární analýzy)*. Akademie věd České republiky, 2002.

[4] ELIÁŠ, Jozef. *Matematika (Úvod do numerickej analýzy)*. Slovenská vysoká škola technická v Bratislave, 1974.

[5] *Jupyter Widgets Developer Docs*. URL: `ipywidgets.readthedocs.io` (citované: 11.2.2020).

[6] KUBÁČEK, Zbyněk, VALÁŠEK, Ján. *Cvičenia z matematickej analýzy I*. Univerzita Komenského v Bratislave, 1989.

[7] LACHNIET, Jason. *Introduction to GNU Octave, Second Edition*. lulu.com, Inc, 2019.

[8] *MATLAB® Primer*. The MathWorks, Inc, 2014. URL: `mathworks.com/help/releases/R2014b/pdf_doc/matlab/getstart.pdf` (citované: 10.2.2020).

[9] McKINNEY, Wes. *Python for Data Analysis, Second Edition*. O'Reilly Media, Inc, 2017.

[10] PÉREZ, Fernando, GRANGER, Brian E. *IPython: A System for Interactive Scientific Computing, Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29*. 2007. URL: `ipython.org` (citované: 10.2.2020).

[11] ROSSANT, Cyrille. *IPython Cookbook, Second Edition*. 2018. URL: `ipython-books.github.io` (citované: 11.2.2020).

[12] *SciPy v1.4.1 Reference Guide*. URL: `docs.scipy.org/doc/scipy/reference` (citované: 11.2.2020).

[13] ŠVEC, Marko, KLUVÁNEK, Igor. *Matematika I pre štúdium technických vied*. Slovenské vydavateľstvo technickej literatúry, 1959.

[14] TOSI, Sandro. *Matplotlib for Python Developers*. Packt Publishing Ltd., 2009.

[15] VENCKO, Jozef, NEUBRUNN, Tibor. *Matematická analýza I, vysokoškolské skriptá*. Matematicko-fyzikálna fakulta Univerzity Komenského, 1992.