

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

INTERAKTÍVNE VYŠETROVANIE PRIEBEHU
ELEMENTÁRNYCH FUNKCIÍ
BAKALÁRSKA PRÁCA

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.



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

Pod'akovanie: Ďakujem...

Abstrakt

TODO

Klíčové slová: ...

Abstract

TODO

Keywords: ...

Obsah

Úvod	1
1 Východiská	2
1.1 Technologické východiská	2
1.2 Teoretické východiská	6
1.3 Existujúce riešenia	8
1.4 Podobné práce	10
2 Návrh riešenia	11
3 Implementácia	12
4 Testovanie a používateľská príručka	13
Záver	14

Úvod

TODO

1 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 1.1), zároveň popísať základnú, v tomto prípade najmä matematickú teóriu, ktorá objasní funkcionality niekoľkých algoritmov použitých pre celkovú funkčnosť riešenia (sekcia 1.2). V sekcii 1.3 je uvedených niekoľko existujúcich riešení problematiky v podobe verejne dostupných open-source programov. Kapitola o východiskách končí popisom podobnej práce obhájenej v roku 2017 a krátkou komparáciou s touto prácou (sekcia 1.4).

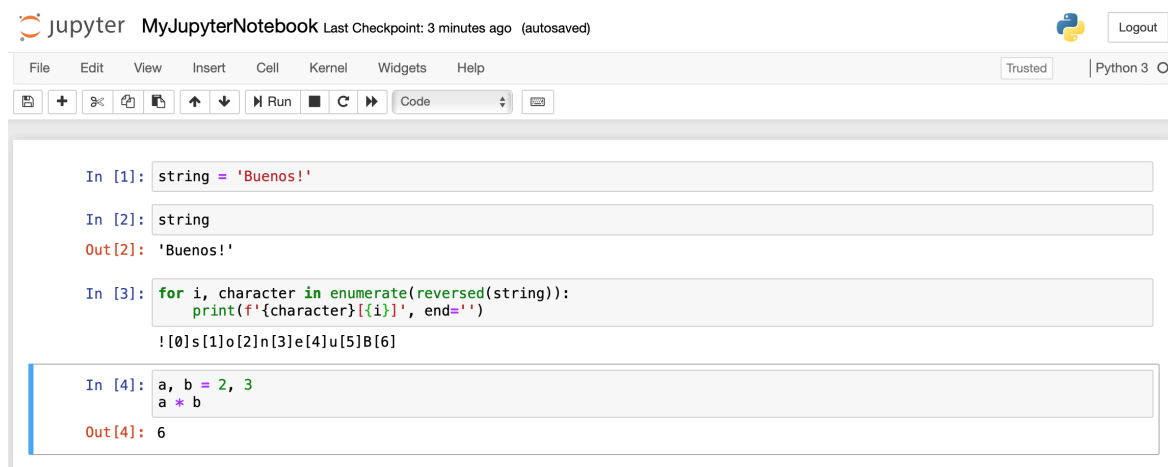
1.1 Technologické východiská

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.

Jupyter Notebook

Okrem programovacieho jazyka Python túto prácu charakterizuje prostredie Jupyter Notebook (obr. 1.1), do ktorého je implementovaná a v ktorom sa používa. Jupyter Notebook autorka praktickej cvičebnice *IPython Cookbook* [10] 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. Ďalej uvádza, že tento nástroj je ideálnou bránou pre náročné numerické výpočty aj v iných populárnych jazykoch dátovej vedy, ako Julia, alebo R. Výpočty v Jupyter Notebookoch, niekedy nazývaných aj IPython Notebooky, majú na starosti 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, mimo jazyka Python 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 a Sage [9]. Po



Obr. 1.1: Interaktívne webové prostredie Jupyter Notebook

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 operácií 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 funkcionality knižnice NumPy, ktoré uvádza autor knihy *Python for Data Analysis* [8] 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 [11]. SciPy, alebo aj Scientific Python, je knižnica, alebo aj kolekcia matematických algoritmov a určitých 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 [8, 13].

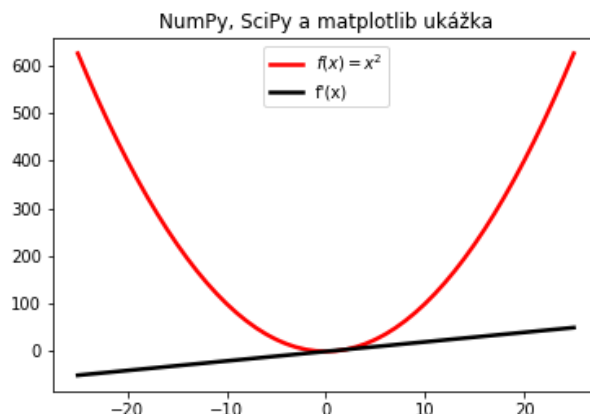
Nasledovný kód slúži na demonštráciu práce s knižnicami NumPy, SciPy a matplotlib. V kóde sa zdefinuje funkcia $f(x) = x^2$ a horizontálne koordináty $x \in \langle -25, 25 \rangle$ v podobe premennej X. Funkčné hodnoty (Y) a derivácie (primes) na celom intervale sa vypočítajú pomocou vektorizovaných operácií NumPy, resp. SciPy. Cez modul pyplot knižnice matplotlib sa vypočítané hodnoty vykreslia do podoby grafu. (obr. 1.2).

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

        def f(X): return X ** 2

        X = np.arange(-25, 25+1)
        Y, primes = f(X), derivative(f, X)

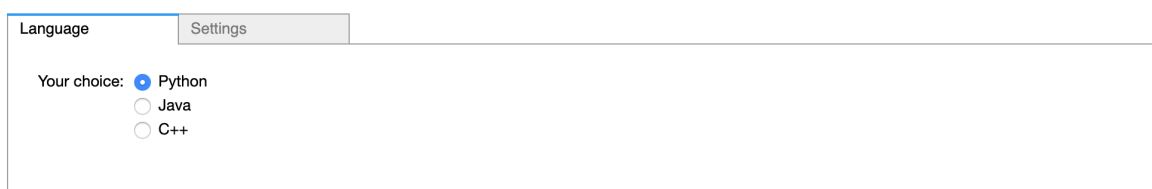
        plt.plot(X, Y, c='red', linewidth=2.5, label=r"$f(x) = x^2$")
        plt.plot(X, primes, c='black', linewidth=2.5, label=r"$f'(x)$")
        plt.title('NumPy, SciPy a matplotlib ukazka')
        plt.legend()
        plt.show()
```



Obr. 1.2: Demonštrácia použitých knižníc - výstup

ipywidgets

Editor ako softvér umožňuje interakciu užívateľ a 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. 1.3).



Obr. 1.3: Príklad interaktívnej aplikácie v prostredí Jupyter Notebook

Anaconda

Dostupnosť a manažment všetkých potrebných knižníc pre túto prácu, vrátane distribúcie samotného jazyka Python a prostredia Jupyter Notebook, zariaduje programový balík Anaconda. Programový balík Anaconda je zadarmo a po nainštalovaní slúži okrem už uvedeného aj na spravovanie virtuálnych prostredí, poskytuje programy a nástroje pre distribúciu programovacieho jazyka pre štatistiku R a obsahuje vyše 7500 ďalších open source riešení najmä pre dátovú vedu [1].

1.2 Teoretické východiská

Pri tvorbe tejto bakalárskej práce je potrebné prispôbiť zložitost' funkcionalít úrovni kurzu *Matematická analýza I*, na ktorom je uvažované jej využívanie. 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. Definície a vety pochádzajú z vysokoškolských skrípt *Cvičenia z matematickej analýzy I* od autorov Kubáček a Valášek [5].

Funkcia, elementárne funkcie

V tejto časti sú okrem pojmov funkcia, funkčná hodnota, definičný obor funkcie, 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. Sú to najmä konštantné, lineárne, kvadratické, polynomiálne, goniometrické, či cyklometrické funkcie.

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 , alebo 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 $f : A \rightarrow \mathbb{R}$ 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.

Monotónnosť funkcie

Definícia. *Nech $A, B \subset \mathbb{R}$ sú neprázdne množiny, kde $B \subset A$. Funkcia $f : A \rightarrow \mathbb{R}$ sa nazýva rastúca na množine B , resp. neklesajúca na množine B , ak platí $\forall x, y \in B ; x < y \Rightarrow f(x) < f(y)$, resp. $\forall x, y \in B ; x < y \Rightarrow f(x) \leq f(y)$.*

Definícia. *Nech $A, B \subset \mathbb{R}$ sú neprázdne množiny, kde $B \subset A$. Funkcia $f : A \rightarrow \mathbb{R}$ sa nazýva klesajúca na množine B , resp. nerastúca na množine B , ak platí $\forall x, y \in B ; x < y \Rightarrow f(x) > f(y)$, resp. $\forall x, y \in B ; x < y \Rightarrow f(x) \geq f(y)$.*

Definícia. *Nech $A, B \subset \mathbb{R}$ sú neprázdne množiny, kde $B \subset A$. Ak má funkcia f niektorú z vlastností uvedených v definíciách 4 a 5, nazýva sa monotónna na množine B . Funkcia, ktorá je rastúca na množine B alebo klesajúca na množine B sa nazýva rýdzomonotónna na množine B . Funkcia rastúca (klesajúca, nerastúca, neklesajúca, monotónna, rýdzomonotónna) na svojom definičnom obore sa nazýva rastúca (klesajúca, nerastúca, neklesajúca, monotónna, rýdzomonotónna).*

Spojitosť funkcie

V niekoľkých prípadoch počas semestra, sa študent stretne s funkciami ako lineárne a racionálne lomené funkcie, ktorých definičné obory sú rozdelené na intervaly. Preto je potrebné zdefinovať pojem spojitá funkcia pomocou nasledovných definícií a viet.

Definícia. *Hovoríme, že funkcia f je spojitá v bode $a \in D(f)$, ak platí, že $\forall \varepsilon > 0 \exists \delta > 0 \forall x \in D(f) : |x - a| < \delta \Rightarrow |f(x) - f(a)| < \varepsilon$.*

Veta. *Ak funkcie f, g sú spojité v bode a , tak aj funkcia $c \cdot f$, kde c je reálna konštanta, $f + g$, $f - g$ a $f \cdot g$ sú spojité v bode a . Ak navyše $g(a) \neq 0$, tak aj funkcia $\frac{f}{g}$ je spojitá v bode a .*

Veta. *Ak funkcia f je spojitá v bode a a funkcia g je spojitá v bode $f(a)$, tak funkcia $g \circ f$ je spojitá v bode a .*

Definícia. *Hovoríme, že funkcia f je spojitá, ak je spojitá v každom bode svojho definičného oboru. Hovoríme, že funkcia f je spojitá na množine A , ak je spojitá funkcia $f|_A$.*

Veta. *Každá základná elementárna funkcia je spojitá.*

Derivácia funkcie

Pojem derivácie funkcie zastupuje kľúčovú funkcionálnu v rámci tejto bakalárskej práce. Okrem faktu, že patrí do osnov predmetu *Matematická analýza I*, je derivácia využívaná na výpočet nulových bodov funkcie, čo tvorí základ pre vyšetrovanie priebehu funkcií. Preto je potrebné definovať pojmy vlastná a nevlastná derivácia a uviesť všeobecnú definíciu derivácie podľa [5].

Definícia. Hovoríme, že funkcia f má deriváciu v bode a ($a \in \mathbb{R}$), ak existuje konečná $\lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a}$. Ak existuje nevlastná $\lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a}$, hovoríme, že funkcia f má nevlastnú (alebo nekonečnú) deriváciu v bode a . Hodnotu $\lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a}$ v obidvoch týchto prípadoch označujeme $f'(a)$. Ak nás nezaujíma, či je $\lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a}$ konečná, alebo nekonečná, používame spoločný názov vlastná alebo nevlastná derivácia v bode a .

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

Vyšetrovanie vlastností funkcie pomocou diferenciálneho počtu

Definícia. ...

Newtonova metóda

...

Metóda sečníc

...

Halleyho metóda

...

1.3 Existujúce riešenia

Editor, ako výstup tejto práce, využíva vety a definície zo sekcie 1.2. Zaobalí uje ich do formy funkcií, či algoritmov, ktoré riadia funkcionality 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.

MATLAB

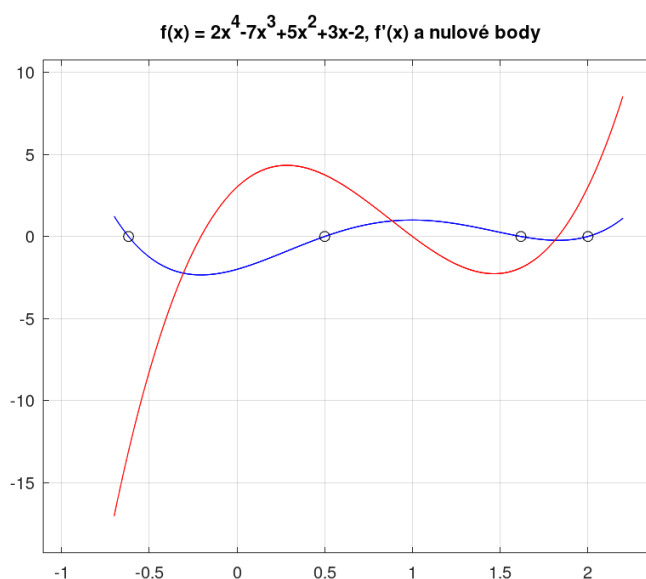
MATLAB je vysoko-úrovňový jazyk a interaktívne prostredie pre numerické výpočty, vizualizáciu a programovanie. Umožňuje analyzovať dáta, vyvíjať algoritmy, či vytvárať modely a aplikácie. Jazyk, jeho nástroje a zabudované matematické funkcie umožňujú zvoliť

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 [7]. Používanie softvéru MATLAB vyžaduje platenú licenciu.

GNU Octave

GNU Octave je tiež vysoko-úrovňový jazyk a interaktívne prostredie pre numerické výpočty, ktoré v zmysle účelu tejto práce poskytuje rovnaké možnosti ako MATLAB, pričom je distribuované pod GNU licenciou, a teda ho je možné využívať zadarmo v plnom rozsahu. Obidva jazyky sú numerické, nie symbolické a ich základným typom je matica, pričom softvér je plne optimalizovaný pre vektorové operácie [6]. 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 prostredí 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 = 2.000000, 1.61803, -0.61803, 0.500000
> 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 nulové body")
```



Obr. 1.4: Demonštrácia MATLAB/Octave - výstup

Práca s nástrojmi popísanými v tejto časti je najmä z hľadiska funkcionalít porovnateľ-

ných 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.

1.4 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, extrémny 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.

2 Návrh riešenia

...

3 Implementácia

...

4 Testovanie a používateľská príručka

...

Záver

TODO

Zoznam obrázkov

1.1	Interaktívne webové prostredie Jupyter Notebook	3
1.2	Demonštrácia použitých knižníc - výstup	5
1.3	Príklad interaktívnej aplikácie v prostredí Jupyter Notebook	5
1.4	Demonštrácia MATLAB/Octave - výstup	9

Zoznam tabuliek

Literatúra

- [1] *Anaconda Distribution Documentation*. URL: docs.anaconda.com/anaconda (citované: 11.2.2020).
- [2] ČERNÝ, Ilja. *Úvod do inteligentního kalkulu (1000 příkladů z elementární analýzy)*. Akademie věd České republiky, 2002.
- [3] ELIÁŠ, Jozef. *Matematika (Úvod do numerickej analýzy)*. Slovenská vysoká škola technická v Bratislave, 1974.
- [4] *Jupyter Widgets Developer Docs*. URL: ipywidgets.readthedocs.io (citované: 11.2.2020).
- [5] KUBÁČEK, Zbyněk, VALÁŠEK, Ján. *Cvičenia z matematickej analýzy I*. Univerzita Komenského v Bratislave, 1989.
- [6] LACHNIET, Jason. *Introduction to GNU Octave, Second Edition*. lulu.com, Inc, 2019.
- [7] *MATLAB® Primer*. The MathWorks, Inc, 2014. URL: mathworks.com/help/releases/R2014b/pdf_doc/matlab/getstart.pdf (citované: 10.2.2020).
- [8] McKINNEY, Wes. *Python for Data Analysis, Second Edition*. O'Reilly Media, Inc, 2017.
- [9] 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).
- [10] ROSSANT, Cyrille. *IPython Cookbook, Second Edition*. 2018. URL: ipython-books.github.io (citované: 11.2.2020).
- [11] *SciPy v1.4.1 Reference Guide*. URL: docs.scipy.org/doc/scipy/reference (citované: 11.2.2020).
- [12] ŠVEC, Marko, KLUVÁNEK, Igor. *Matematika I pre štúdium technických vied*. Slovenské vydavateľstvo technickej literatúry, 1959.

- [13] TOSI, Sandro. *Matplotlib for Python Developers*. Packt Publishing Ltd., 2009.
- [14] VENCKO, Jozef, NEUBRUNN, Tibor. *Matematická analýza I, vysokoškolské skriptá*. Matematicko-fyzikálna fakulta Univerzity Komenského, 1992.