

Univerzita J. E. Purkyně
Přírodovědecká fakulta

Katedra Informatiky
Matematický software
LS 2022/2023

Matematický software

Zápočtový dokument

Jméno: Jiří Stibor
Informatika – Historie pro vzdělávání
Kontaktní email: stibac01@gmail.com
Datum odevzdání: 30.06.2022
Odkaz na repozitář: https://github.com/stibi1/MSW_2022_Jiri_Stibor_Zapocet

Formální požadavky

Cíl předmětu:

Cílem předmětu je ovládnout vybrané moduly a jejich metody pro jazyk Python, které vám mohou být užitečné jak v dalších semestrech vašeho studia, závěrečné práci (semestrální, bakalářské) nebo technické a výzkumné praxi.

Získání zápočtu:

Pro získání zápočtu je nutné částečně ovládnout alespoň polovinu z probraných témat. To prokážete vyřešením vybraných úkolů. V tomto dokumentu naleznete celkem 10 zadání, která odpovídají probíraným tématům. Vyberte si 5 zadání, vypracujte je a odevzdejte. Pokud bude všech 5 prací korektně vypracováno, pak získáváte zápočet. Pokud si nejste jisti korektností vypracování konkrétního zadání, pak je doporučeno vypracovat více zadání a budou se započítávat také, pokud budou korektně vypracované.

Korektnost vypracovaného zadání:

Konkrétní zadání je považováno za korektně zpracované, pokud splňuje tyto kritéria:

1. Použili jste numerický modul pro vypracování zadání místo obyčejného pythonu
2. Kód neobsahuje syntaktické chyby a je interpretovatelný (spustitelný)
3. Kód je čistý (vygooglete termín clean code) s tím, že je akceptovatelné mít ho rozdělen do Jupyter notebook buněk (s tímhle clean code nepočítá)

Forma odevzdání:

Výsledný produkt odevzdáte ve dvou podobách:

1. Zápočtový dokument
2. Repozitář s kódem

Zápočtový dokument (vyplněný tento dokument, který čtete) bude v PDF formátu. V řešení úloh uveďte důležité fragmenty kódu a grafy/obrázky/textový výpis pro ověření funkčnosti. Stačí tedy uvést jen ty fragmenty kódu, které přispívají k jádru řešení zadání. Kód nahrajte na veřejně přístupný repozitář (github, gitlab) a uveďte v práci na něj odkaz v titulní straně dokumentu. Strukturujte repozitář tak, aby bylo pro nás hodnotitele intuitivní se vyznat v souborech (doporučuji každou úlohu dát zvlášť do adresáře).

Podezření na plagiátorství:

Při podezření na plagiátorství (významná podoba myšlenek a kódu, která je za hranicí pravděpodobnosti shody dvou lidí) budete vyzváni k fyzickému dostavení se na zápočet do prostor univerzity, kde dojde k vysvětlení podezřelých partií, nebo vykonání zápočtového testu na místě z matematického softwaru v jazyce Python.

Kontakt:

Při nejasnostech ohledně zadání nebo formě odevzdání se můžete obrátit nejlépe na tvůrce těchto zadání na Discord serveru Pavla Beránka (měli byste mít odkaz) nebo na emailu: *pavelberanek91@gmail.com*.

9. Integrace funkce jedné proměnné

Zadání:

V oblasti přírodních a sociálních věd je velice důležitým pojmem integrál, který představuje funkci součtů malých změn (počet nakažených covidem za čas, hustota monomerů daného typu při posouvání se v řetízku polymeru, aj.). Integraci lze provádět pro velmi jednoduché funkce prostou Riemannovým součtem, avšak pro složitější funkce je nutné využít pokročilé metody. Vaším úkolem je vybrat si 3 různorodé funkce (polynom, harmonická funkce, logaritmus/exponenciála) a vypočíst určitý integrál na dané funkci od nějakého počátku do nějakého konečného bodu. Porovnejte, jak si každá z metod poradila s vámi vybranou funkcí na základě přesnosti vůči analytickému řešení.

Řešení:

Využité funkce: Harmonická: $2 * \sin(8 * x)$ | Logaritmická: $\log(7 * x) + (2 / 7)$ |

Polynomická: $x ** 2 - 3 * x + 9$

Využité výpočty: Reimannova metoda `integrate.quadrature(function, x, y)` |

Simpsonova metoda `integrate.simpson(function(np.arange(x, y + z, z)), np.arange(x, y + z, z))` | Rombergova metoda `integrate.romberg(function, x, y)`

Výsledky funkcí:

| | | |
|------------------------|---------------------|--|
| Harmonic function | | |
| Fun | Results | |
| Rie | 0.20303986157367274 | |
| Simp | 0.1877390315322114 | |
| Romb | 0.20303986162869195 | |
| Logarithmical function | | |
| Fun | Results | |
| Rie | 2.6179187959692247 | |
| Simp | 2.6765030223172497 | |
| Romb | 2.617918795889222 | |
| Polynomic function | | |
| Fun | Results | |
| Rie | 6.833333333333332 | |
| Simp | 6.9735720000000006 | |
| Romb | 6.833333333333333 | |

1. Knihovny a moduly pro matematické výpočty

Zadání:

V tomto kurzu jste se učili s některými vybranými knihovnami. Některé sloužily pro rychlé vektorové operace jako NumPy, některé mají naprogramovány symbolické manipulace, které lze převést na numerické reprezentace (sympy), některé mají v sobě funkce pro numerickou integraci (scipy). Některé slouží i pro rychlé základní operace s čísly (numba).

Vaším úkolem je změřit potřebný čas pro vyřešení nějakého problému (např.: provést skalární součin, vypočítat určitý integrál) pomocí standardního pythonu a pomocí specializované knihovny. Toto měření proveďte alespoň pro 5 různých úloh (ne pouze jiná čísla, ale úplně jiné téma) a minimálně porovnejte rychlost jednoho modulu se standardním pythonem. Ideálně proveďte porovnání ještě s dalším modulem a snažte se, ať je kód ve standardním pythonu napsán efektivně.

Řešení:

Využité funkce: Derivace | Integrace | Skalární součet | Faktoriál | Násobení Maticí

```
| Sympy Library |
| Derivation = 3540 | Time to reach result = 0.046875(s)
-----

| Basic Python |
| Derivation = 3540.5040240020753 | Time to reach result = 0.0(s)

-----

| Scipy Library |
| Integration = (1.3333333333333335, 1.4802973661668755e-14) | Time to reach result = 0.0(s)
-----

| Basic Python |
| Integration = 1.3334333400001694 | Time to reach result = 0.015625(s)

-----

| Numpy Library |
| Scalar product = 210 | Time to reach result = 0.0(s)
-----

| Basic Python |
| Scalar product = 210 | Time to reach result = 0.0(s)

-----

| Math Library |
| Factorial of 55200 = 5646115017 | Time to reach result = 0.0625(s)
-----

| Basic Python |
| Factorial of 55200 = 5646115017 | Time to reach result = 0.640625(s)

-----

| Numpy Library |
| Matrix = [[ 825 22380 30015]
[ 9090 22890 29160]
[ 0 29085 30330]] | Time to reach result = 0.0(s)
-----

| Basic Python |
| Matrix = [[825, 22380, 30015], [9090, 22890, 29160], [0, 29085, 30330]] | Time to reach result = 0.0(s)
```

- Derivace vychází rychleji u základního pythonu, Integrace u knihovny Scipy, Skalární součet vychází rychlostně nastejno stejně jako Matice a u faktoriálu je opět rychlejší knihovna.

8. Derivace funkce jedné proměnné

Zadání:

Numerická derivace je velice krátké téma. V hodinách jste se dozvěděli o nejvyužívanějších typech numerické derivace (dopředná, zpětná, centrální). Jedno z neřešených témat na hodinách byl problém volby kroku. V praxi je vhodné mít krok dynamicky nastavitelný. Algoritmům tohoto typu se říká derivace s adaptabilním krokem. Cílem tohoto zadání je napsat program, který provede numerickou derivaci s adaptabilním krokem pro vámi vybranou funkci. Proveďte srovnání se statickým krokem a analytickým řešením.

Řešení:

Využitá funkce: `f = lambda x: x**2 + 3`

Následně provedena dopředná, zpětná a centrální derivace s a bez kroku

- Výsledek pro adaptivní krok 5

```
Input adaptive step(only numbers please): 5

| No Input forward = 8.525999999999993 | Input forward = 13.0 |
| No Input backward = 7.473999999999997 | Input backward = 3.0 |
| No input central = 7.999999999999995 | Input central = 8.0 |

Process finished with exit code 0
```

- Výsledek pro krok 8,9

```
Input adaptive step(only numbers please): 8,9

| No Input forward = 8.525999999999993 | Input forward = 97.0 |
| No Input backward = 7.473999999999997 | Input backward = -81.0 |
| No input central = 7.999999999999995 | Input central = 8.0 |
```

Input je ošetřen pouze na chybovost mezer a “,” místo “.”

10. Řešení obyčejných diferenciálních rovnic

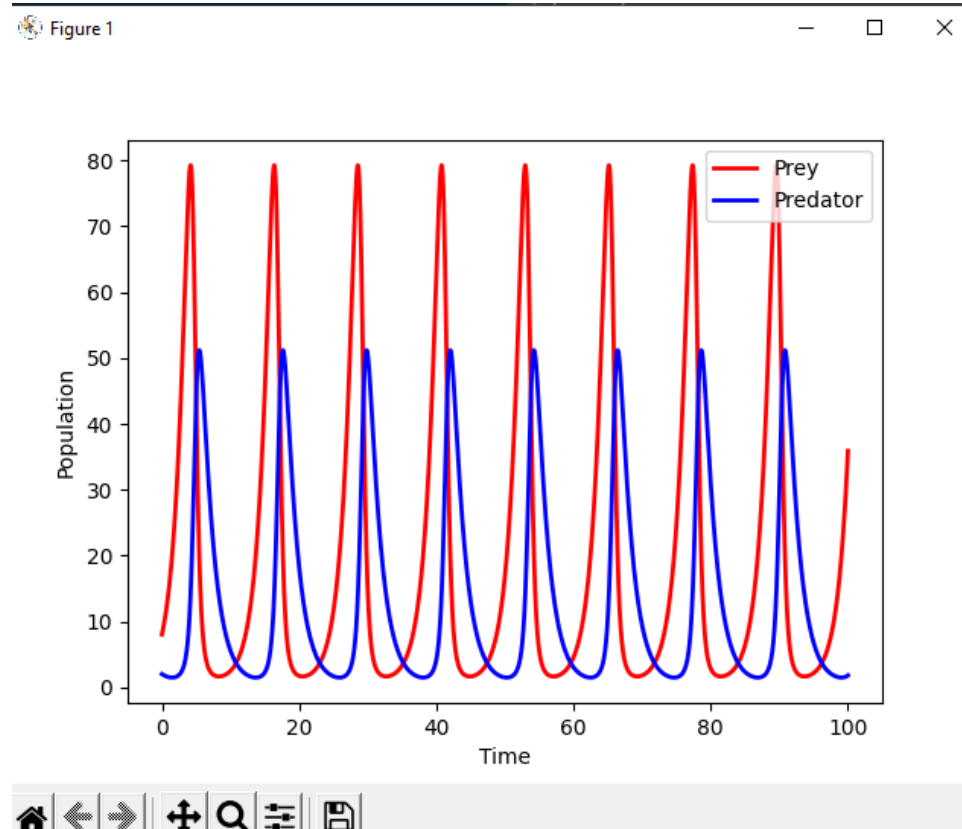
Zadání:

Diferenciální rovnice představují jeden z nejdůležitějších nástrojů každého přírodovědně vzdělaného člověka pro modelování jevů kolem nás. Vaším úkolem je vybrat si nějakou zajímavou soustavu diferenciálních rovnic, která nebyla zmíněna v sešitech z hodin a pomocí vhodné numerické metody je vyřešit. Řešením se rozumí vizualizace jejich průběhu a jiných zajímavých informací, které lze z rovnic odvodit. Proveďte také slovní okomentování toho, co lze z grafu o modelovaném procesu vyčíst.

Řešení:

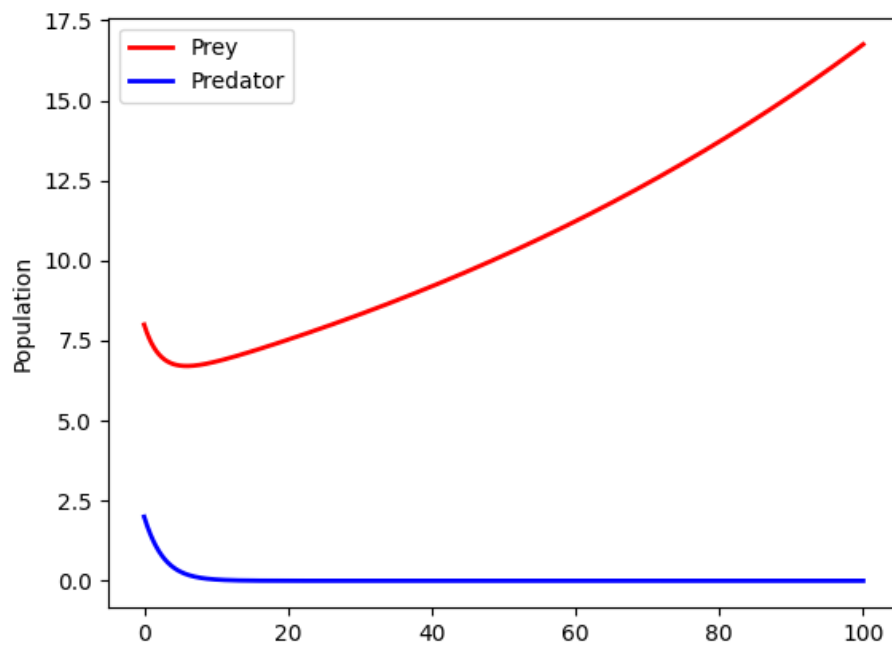
Využité funkce: Lotkúv-Volterrův model Predátora a kořisti dynamiky populace

Graf:

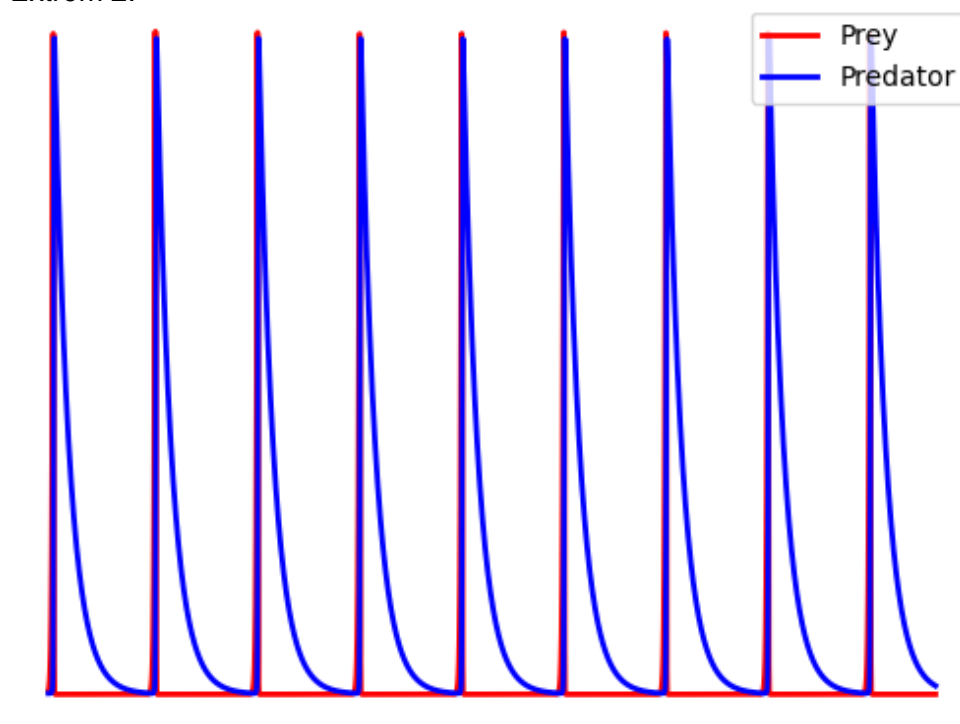


- Hodnoty jsou na sobě závislé aneb v případě nedostatku kořisti vymírá predátor a při nedostatku predátora se kořist přemnoží

Extrém 1:



Extrém 2:



3. Úvod do lineární algebry

Zadání:

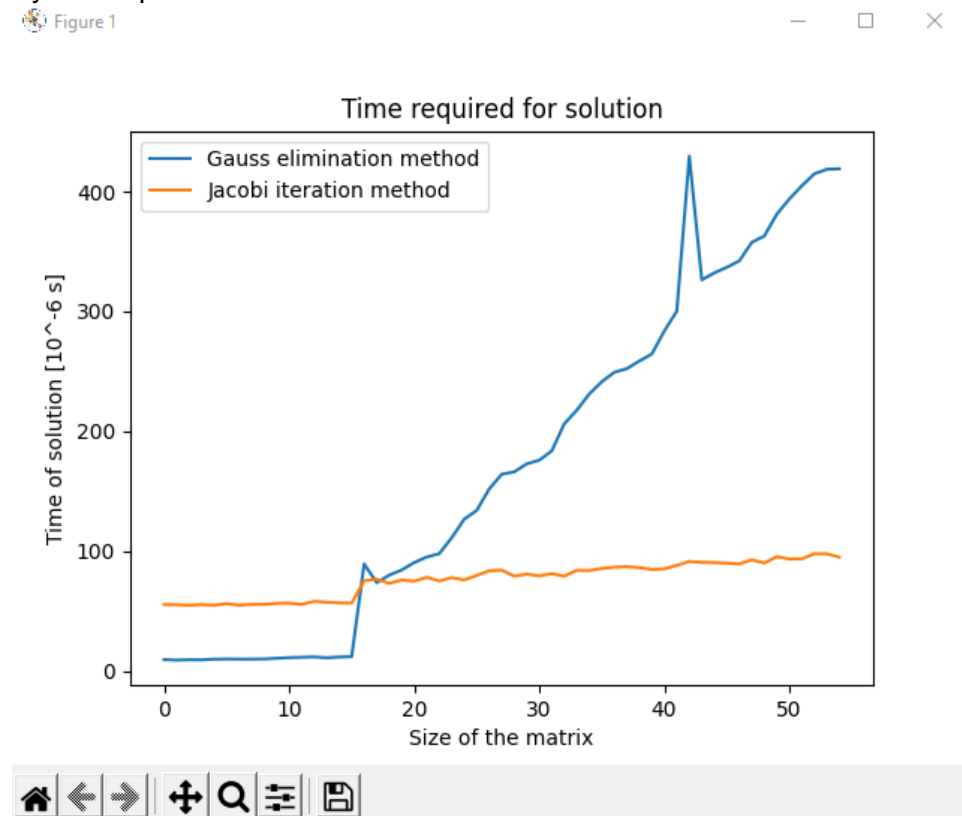
Důležitou částí studia na přírodovědecké fakultě je podobor matematiky zvaný lineární algebra. Poznatky tohoto oboru jsou základem pro oblasti jako zpracování obrazu, strojové učení nebo návrh mechanických soustav s definovanou stabilitou. Základní úlohou v lineární

algebře je nalezení neznámých v soustavě lineárních rovnic. Na hodinách jste byli obeznámeni s přímou a iterační metodou pro řešení soustav lineárních rovnic. Vaším úkolem je vytvořit graf, kde na ose x bude velikost čtvercové matice a na ose y průměrný čas potřebný k nalezení uspokojivého řešení. Cílem je nalézt takovou velikost matice, od které je výhodnější využít iterační metodu.

Řešení:

Využité funkce: Jacobi a Gauss

Výsledek pro velikost matice 54



- Výsledek pro velikost matice 24

Figure 1

