

Projekt 2023 - část 1

předmět Zpracování a vizualizace dat v prostředí Python

Cílem projektu je získat, zpracovat a analyzovat data dostupná na internetu. První část projektu se bude týkat ověření vašeho osvojení technik pro efektivní zpracování, vizualizaci a získání dat.

Orientační struktura projektu

- Část 1 (20 b)
 - efektivní numerické výpočty
 - jednoduchá vizualizace
 - stažení a zpracování dat
- Část 2 (20 b)
 - různé pohledy na data
 - pokročilá vizualizace výsledků
 - zpracování závěrů (porozumění datům)
- Celkový projekt (60 b)
 - znázornění dat na mapě, operace nad těmito daty
 - korelace a predikce
 - automatické vytváření částí zpráv
 - spojení do analytické zprávy

Získání a předzpracování dat (20 bodů)

Vytvořte soubor `part01.py`, který bude implementovat níže uvedené metody. Pro zpracování a vizualizaci dat **není povoleno** použít pokročilých knihoven jako je **Pandas** či **Seaborn**. Kromě vestavěných knihoven (`os`, `sys`, `re`, `gzip`, `pickle`, `csv`, `zipfile`...) byste si měli vystačit s: `numpy`, `matplotlib`, `BeautifulSoup`, `requests`. Další knihovny je možné použít po schválení opravujícím (např ve fóru IS VUT).

Úkol 1: Numerický výpočet integrálu (3 body)

Cílem této funkce je numericky vypočítat integrál nepřesnou tzv. obdélníkovou metodou. Jako vstup dostanete funkci f (kompatibilní s NumPy, za vstup bere jedno NumPy pole a vrací NumPy pole o stejném rozměru, není nutné testovat), počátek a konec intervalu a, b a volitelně počet kroků `steps`. Pokud není nastaven, bude `steps` rovno 1000. Funkce vrátí hodnotu určitého integrálu vypočtenou podle následujícího vzorce, kdy X jsou všechny body na lineárním rozdělení od a po b .

$$\int_{x_0}^{x_{|X|-1}} f(x) dx = \sum_{i=1}^{|X|-1} (x_i - x_{i-1}) \cdot f\left(\frac{x_{i-1} + x_i}{2}\right)$$

Pro získání plného hodnocení je nutné se zaměřit na efektivitu a využít možností knihovny *numpy* - t.j. vyhnout se procházení všech prvků cyklem *for*. Je zakázáno použít knihovní funkci, která přímo vrací určitý integrál.

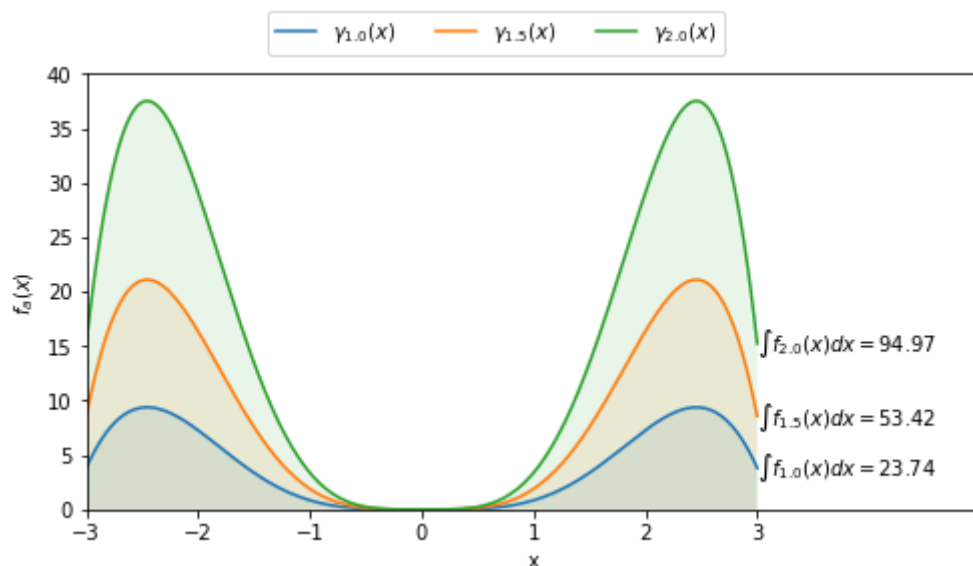
Prototyp funkce:

```
def integrate(f: Callable[[NDArray], NDArray], a: float, b: float, steps=1000)
-> float:
```

Úkol 2: Generování grafu s různými koeficienty (6 bodů)

Navrhněte funkci *generate_graph*, která bude vizualizovat funkci $f_a(x) = a^2 * x^3 * \sin(x)$, definovanou na rozsahu $<-3, 3>$. Na začátku kódu vygenerujte v jednom kroku výsledky funkce *f* pro všechny hodnoty *a* (t.j. bez cyklů, do dvourozměrné matice) zadané ve vstupním argumentu (reprezentované jako seznam čísel s plovoucí desetinnou čárkou). Je tedy nutné (pro plné hodnocení) využít *broadcasting*.

Následně tuto matici po jednotlivých řádcích vizualizujte tak, aby výsledný graf vypadal následovně. Pro nastavení rozsahů zobrazení, umístění popisků a podobně můžete počítat s tím, že $a=[1.0, 1.5, 2.0]$. Je nutné dodržet *Latex* styl sazby popisků os a jednotlivých čar. Hodnoty integrálů vypočtete pomocí funkce *trapez* v *numpy*. Zachovejte následující vzhled:



Funkce *generate_graph* má další dva argumenty - boolean hodnotu *show_figure*, která určuje, zda se má graf zobrazit pomocí funkce *show()* a *save_path*, která (pokud je nastavena), určuje, kam se má graf uložit pomocí funkce *savefig()*.

Prototyp funkce

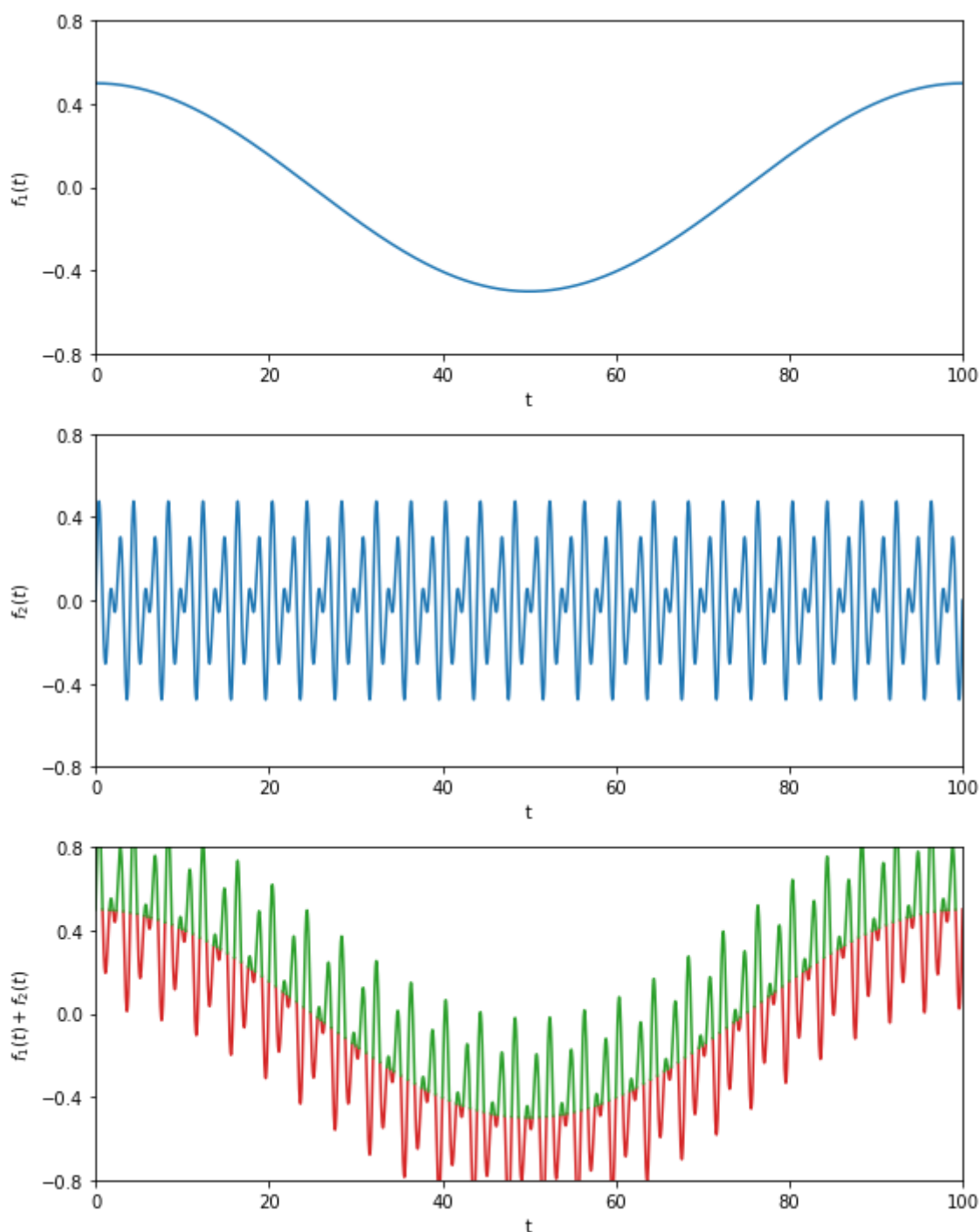
```
def generate_graph(a: List[float],
                  show_figure: bool = False,
                  save_path: str | None = None):
```

Úkol 3: Pokročilá vizualizace sinusového signálu (5 bodů)

Vytvořte graf se třemi podgrafy, zobrazující funkci f_1 , f_2 a součet f_1+f_2 v rozsahu $t \in \langle 0, 100 \rangle$. Funkce jsou definovány následovně:

$$f_1(t) = 0.5 \cdot \cos\left(\frac{1}{50}\pi t\right), f_2(t) = 0.25 \cdot \left(\sin(\pi t) + \sin\left(\frac{3}{2}\pi t\right)\right)$$

V třetím podgrafu bude část, kdy se hodnota součtu obou funkcí dostává nad hodnotu samotné funkce f_1 zeleně, v opačném případě červeně. Je nutné, aby na grafu nevznikly žádné další artefakty (např. nějaké spojovací čáry a podobně). Upravte také body na ose tak, aby vypadaly, jako na tomto vzorovém obrázku. Pro argumenty `show_figure` a `save_path` platí stejné podmínky, jako v druhém úkolu.



Prototyp funkce

```
def generate_sinus(show_figure: bool = False,  
                  save_path: str | None = None)
```

Úkol 4: Stažení tabulky (5 bodů)

Ze stránek <https://ehw.fit.vutbr.cz/izv/stanice.html> stáhněte meteorologických stanic. Jedná se o kopii stránek CHMI a je zakázáno se připojovat na oficiální stránky. Url, ze které stahujete, můžete mít uloženou v kódu. Nemusíte pracovat rovnou s touto stránkou (v prohlížeči např. přes *Network panel* zjistíte, že stránky jsou řešeny poměrně “originálním” stylem). Pro každý řádek tabulky vytvořte záznam typu slovník s následující strukturou:

```
{'position': 'Cheb', 'lat': 50.0683, 'long': 12.3913, 'height': 483.0}
```

Výstupem funkce bude seznam (list) obsahující záznamy (dict) pro jednotlivé řádky. Validita výstupního formátu je základním způsobem testována i v přiloženém *unittestu*. Můžete počítat s tím, že struktura stránek se nezmění, vlastní načítání je však povinné dělat ze stránek <https://ehw.fit.vutbr.cz/izv> a je přísně zakázáno přistupovat na stránky CHMI.

Prototyp funkce

```
def download_data() -> List[Dict[str, Any]]:
```

Testování

K otestování funkčnosti můžete přistoupit dvěma základními způsoby. Buď do části, která se spouští pouze pokud je zavolán celý skript (if `__name__=="__main__"`), vložíte vaši testovací sekvenci, nebo můžete použít předpřipravený *unittest* v souboru *test_part01.py* a knihovnu *pytest*. Pokud máte tuto knihovnu nainstalovanou, můžete spustit příkaz `pytest` či `python3 -m pytest`. Pokud projdete testovacím skriptem, je to podmínka nutná (nikoliv dostačující) k dobrému hodnocení.

Dokumentace všech částí (souborů, funkcí) bude přímo v odevzdaném souboru. Snažte se dodržovat konvenci PEP 257 [<https://www.python.org/dev/peps/pep-0257>] a PEP 8 [<https://www.python.org/dev/peps/pep-0008/>]. Pozor, Python má přímo definovaný kódovací styl (narozdíl od C), a proto kontrola bude součástí hodnocení (1 bod).

Odevzdávání a hodnocení

Do 5. 11. 2022 odevzdejte jeden soubor *part01.py*.

Hodnotit se bude zejména:

- správnost výsledků
- vizuální dojem z grafů
- kvalita kódu
 - efektivitu implementace a reprezentace (i rychlost v porovnání s ostatními řešeními), využívání efektivních funkcí (např. NumPy)
 - přehlednost kódu

- dodržení standardů a zvyklostí pro práci s jazykem Python - dokumentační řetězce, splnění PEP8
- dokumentace funkcí a souborů
- znovupoužitelnost kódů - správná izolace potřebných částí do funkcí

Celkem za první část můžete získat až 20 bodů, přičemž je k zápočtu nutné získat z této části minimálně 1 bod.

Dotazy a připomínky

Na fóru IS VUT případně na mailu mrazek@fit.vutbr.cz.