

# Matematický software

Zápočtový dokument

<b>Jméno:</b>	Bao Kieu Quang
<b>Kontaktní email:</b>	bao.kieuquang@gmail.com
<b>Datum odevzdání:</b>	27.09.2023
<b>Odkaz na repozitář:</b>	<a href="https://github.com/ChaoZavitkar/msw_prace">https://github.com/ChaoZavitkar/msw_prace</a>

# 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 tato 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 obraťte na vyučujícího.

# 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:**

doplňte

## 2. Vizualizace dat

### Zadání:

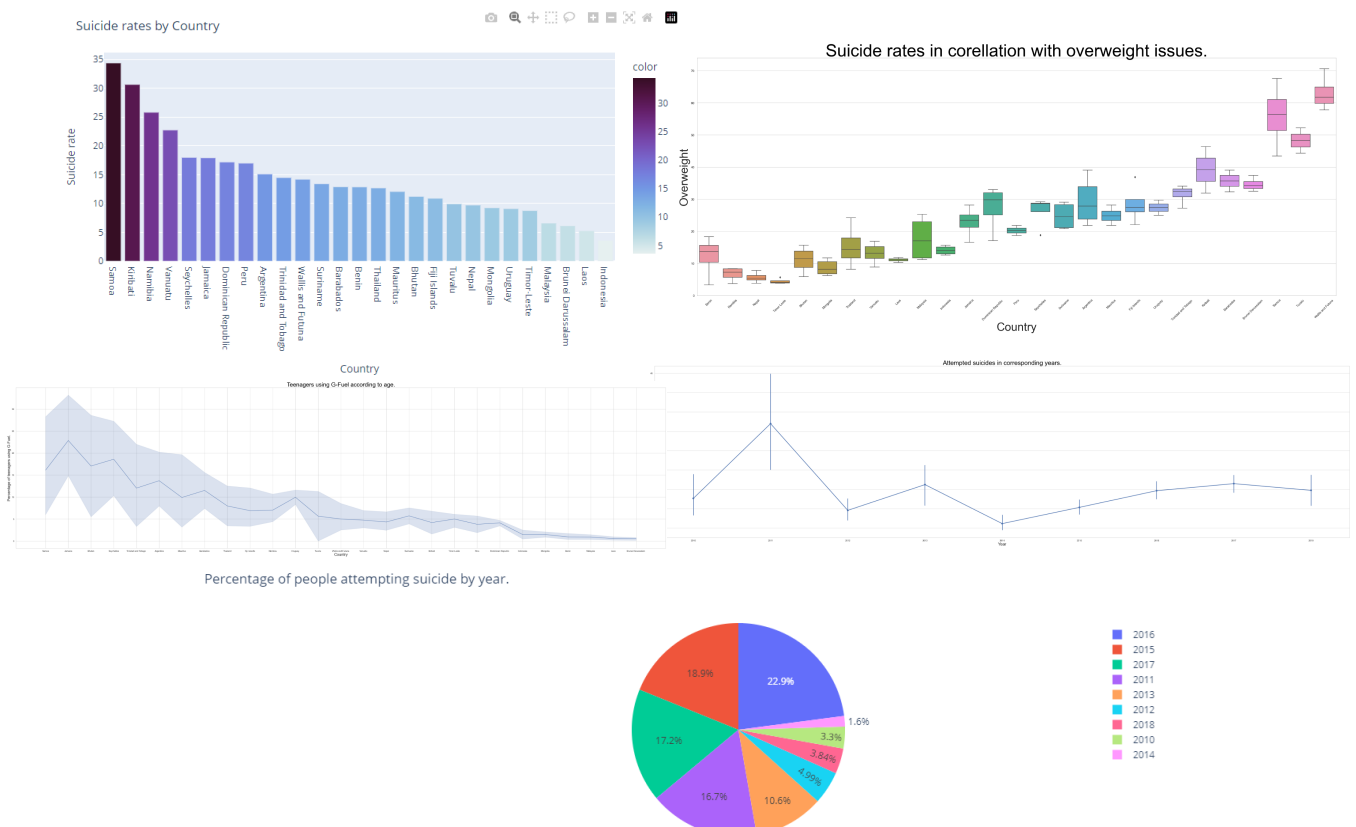
V jednom ze cvičení jste probírali práci s moduly pro vizualizaci dat. Mezi nejznámější moduly patří matplotlib (a jeho nadstavby jako seaborn), pillow, opencv, aj. Vyberte si nějakou zajímavou datovou sadu na webovém portále Kaggle a proveďte datovou analýzu datové sady. Využijte k tomu různé typy grafů a interpretujte je (minimálně alespoň 5 zajímavých grafů). Příklad interpretace: z datové sady pro počasí vyplynulo z liniového grafu, že v létě je vyšší rozptyl mezi minimální a maximální hodnotou teploty. Z jiného grafu vyplývá, že v létě je vyšší průměrná vlhkost vzduchu. Důvodem vyššího rozptýlu může být absorpce záření vzduchem, který má v létě vyšší tepelnou kapacitu.

### Řešení:

Pro tuto úlohu se používají následující knihovny pro manipulaci s daty a vizualizaci. Pandas pro práci s datovými strukturami, matplotlib pro tvorbu grafů, seaborn pro estetickou vizualizaci, plotly.express pro interaktivní vizualizaci a make\_subplots pro tvorbu podgrafů.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from plotly.subplots import make_subplots
```

Kód generuje celkem pět grafů: za prvé sloupcový graf zobrazující průměrnou míru pokusů o sebevraždu podle zemí s příslušnými úpravami pro přehlednost; za druhé krabicový a šikmý graf zobrazující vztah mezi mírou pokusů o sebevraždu a problémy s hmotností podle zemí; za třetí spojnicový graf zobrazující procento dospívajících v zemích, které používají G-Fuel. čtvrtý je bodový graf znázorňující změnu míry sebevražedných pokusů v čase a pátý je koláčový graf znázorňující podíl osob, které se pokusily o sebevraždu.



### 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:**

doplňte

## 4. Interpolace a aproximace funkce jedné proměnné

### **Zadání:**

Během měření v laboratoři získáte diskrétní sadu dat. Často potřebujete data i mezi těmito diskrétními hodnotami a to takové, které by nejpřesněji odpovídaly reálnému naměření. Proto je důležité využít vhodnou interpolační metodu. Cílem tohoto zadání je vybrat si 3 rozdílné funkce (např. polynom, harmonická funkce, logaritmus), přidat do nich šum (trošku je v každém z bodů rozkmitajte), a vyberte náhodně některé body. Poté proveďte interpolaci nebo aproximaci funkce pomocí alespoň 3 rozdílných metod a porovnejte, jak jsou přesné. Přesnost porovnáte s daty, které měly původně vyjít. Vhodnou metrikou pro porovnání přesnosti je součet čtverců (rozptylů), které vzniknou ze směrodatné odchylky mezi odhadnutou hodnotou a skutečnou hodnotou.

### **Řešení:**

doplňte

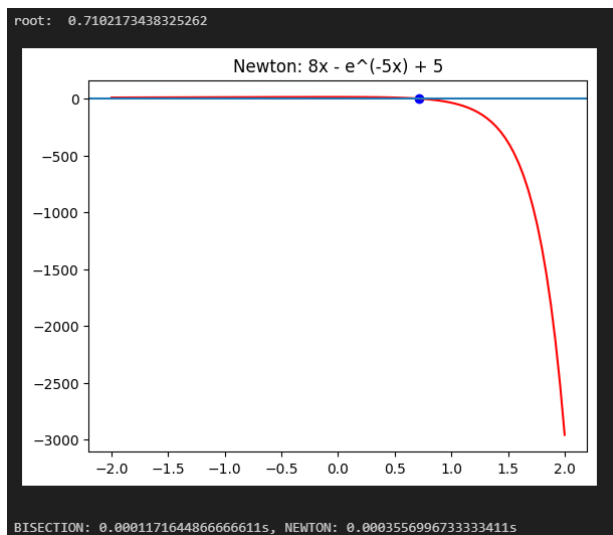
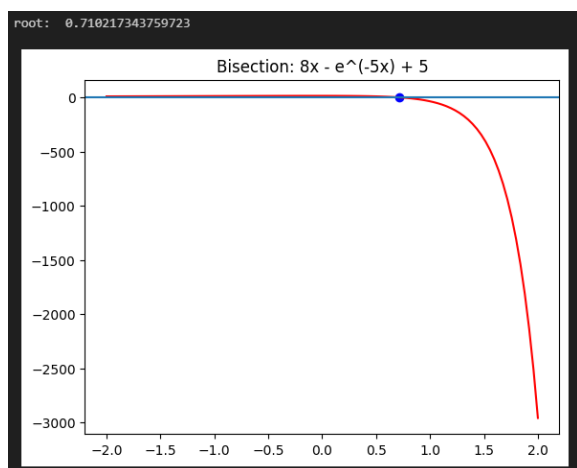
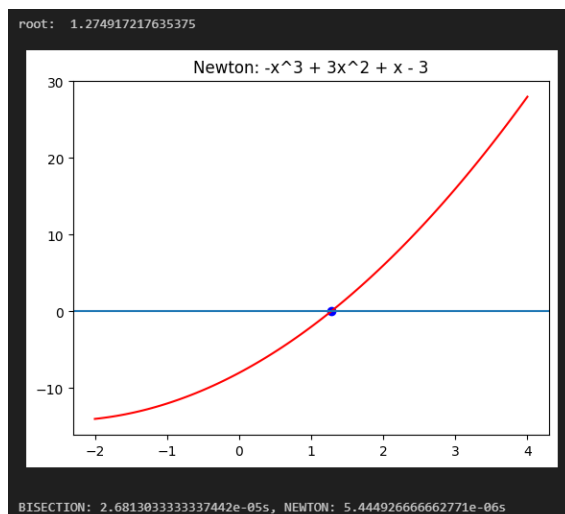
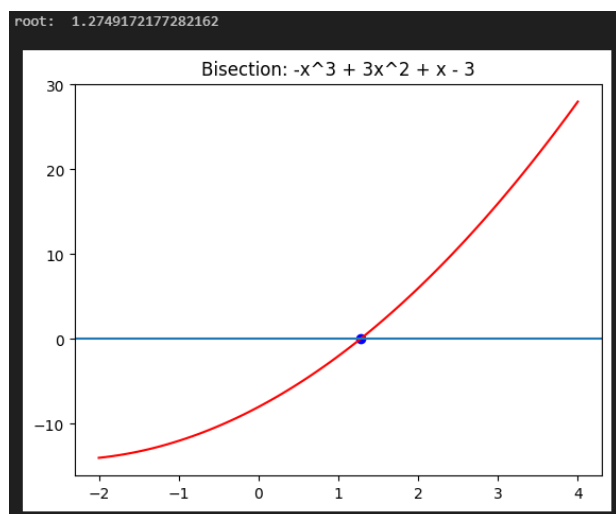
## 5. Hledání kořenů rovnice

### Zadání:

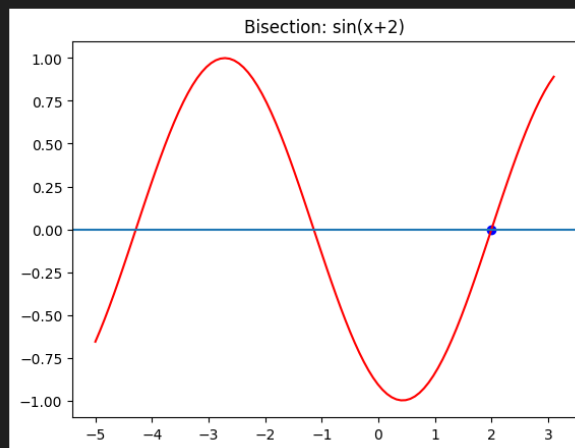
Vyhledávání hodnot, při kterých dosáhne zkoumaný signál vybrané hodnoty je důležitou součástí analýzy časových řad. Pro tento účel existuje spousta zajímavých metod. Jeden typ metod se nazývá ohraničené (například metoda půlení intervalu), při kterých je zaručeno nalezení kořenu, avšak metody typicky konvergují pomalu. Druhý typ metod se nazývá neohraničené, které konvergují rychle, avšak svojí povahou nemusí nalézt řešení (metody využívající derivace). Vaším úkolem je vybrat tři různorodé funkce (například polynomiální, exponenciální/logaritmickou, harmonickou se směrnicí, aj.), které mají alespoň jeden kořen a nalézt ho jednou uzavřenou a jednou otevřenou metodou. Porovnejte časovou náročnost nalezení kořene a přesnost nalezení.

### Řešení:

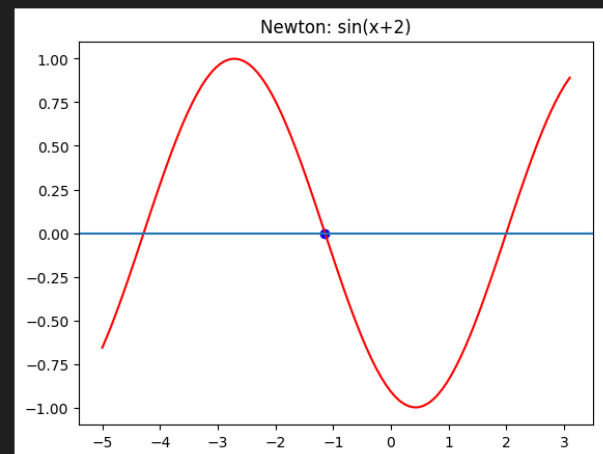
Kód implementuje dvě numerické metody (bisekci a Newtonovu metodu) pro hledání kořenů funkce. Dále jsou definovány tři různé matematické funkce ( $fce1 = \lambda x: x^2 + 5x - 8$ ,  $fce2 = \lambda x: 3x - np.exp(4x) + 15$ ,  $fce3 = \lambda x: np.sin(x-2)$ ) a hledají se kořeny těchto funkcí. Výsledky nalezení kořenů jednotlivých funkcí pomocí obou metod jsou znázorněny graficky. Kromě toho je také změřen čas potřebný k nalezení kořenů každé funkce. Celý postup je podrobně ukázán pro tři různé matematické funkce.



root: 1.999999999374274



root: -1.1415926535897933



BISECTION: 0.00010327399999999519s, NEWTON: 1.991092666655845e-05s



## 6. Generování náhodných čísel a testování generátorů

### Zadání:

Tento úkol bude poněkud kreativnější charakteru. Vaším úkolem je vytvořit vlastní generátor semínka do pseudonáhodných algoritmů. Jazyk Python umí sbírat přes ovladače hardwarových zařízení různá fyzická a fyzikální data. Můžete i sbírat data z historie prohlížeče, snímání pohybu myši, vyzvání uživatele zadat náhodné úhozy do klávesnice a jiná unikátní data uživatelů.

### Řešení:

Kód generuje náhodná čísla jako výsledek pohybu myši na obrazovce. K získání polohy myši používá knihovnu turtle a ke sledování pohybu myši knihovnu pyautogui; ukládá polohu myši po dobu 5 sekund a na základě této polohy generuje náhodná čísla. Semeno pro generování náhodných čísel se vypočítá jako faktoriál součtu souřadnic x a y každé pozice myši. Nakonec je vypsán výsledek semínka náhodných čísel.

```
Mouse position 798 162 saved.  
Mouse position 1047 191 saved.  
Mouse position 1047 191 saved.  
Mouse position 1047 191 saved.  
Mouse position 1047 191 saved.  
Mouse position 1047 191 saved.  
Mouse position 1047 191 saved.  
Mouse position 1049 197 saved.  
Mouse position 1049 197 saved.  
Mouse position 1051 200 saved.  
Mouse position 1051 200 saved.  
Mouse position 1051 202 saved.  
Mouse position 1051 202 saved.  
Mouse position 1051 202 saved.  
Mouse position 1051 202 saved.  
Mouse position 1051 202 saved.  
Mouse position 1051 202 saved.  
Mouse position 1051 202 saved.  
Mouse position 1051 202 saved.  
Mouse position 1051 202 saved.  
Mouse position 1051 202 saved.  
Mouse position 1051 202 saved.  
Mouse position 1051 202 saved.  
Your randomly generated seed is: 25228718021934010162619073100770614309566263392587257296956366720793610675378650101186
```

## 7. Metoda Monte Carlo

### **Zadání:**

Metoda Monte Carlo představuje rodinu metod a filozofický přístup k modelování jevů, který využívá vzorkování prostoru (například prostor čísel na herní kostce, které mohou padnout) pomocí pseudonáhodného generátoru čísel. Jelikož se jedná spíše o filozofii řešení problému, tak využití je téměř neomezené. Na hodinách jste viděli několik aplikací (optimalizace portfolia aktiv, řešení Monty Hall problému, integrace funkce, aj.). Nalezněte nějaký zajímavý problém, který nebyl na hodině řešen, a získejte o jeho řešení informace pomocí metody Monte Carlo. Můžete využít kódy ze sešitu z hodin, ale kontext úlohy se musí lišit.

### **Řešení:**

doplňte

## 8. Derivace funkce jedné proměnné

### Zadání:

Numerická derivace je velice krátké téma. V hodinách jste se dozvěděli o nejvyžívanějších 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. Provedte srovnání se statickým krokem a analytickým řešením.

### Řešení:

Kód definuje funkce pro numerické diferenciální výpočty pomocí různých metod. (dopředné, zpětné, centrální difference). Dále je definována matematická funkce "func" jako  $(x^5 - 3x^{30} + 5)$ . Jsou zde nastaveny hodnoty pro velikost intervalu ( $x_0$ ) a krok ( $h$ ). Uživatel si také může zvolit adaptační krok. Pro dané hodnoty  $x_0$  a  $h$  jsou vypočteny derivace pro danou funkci. Derivace se pak rovněž vypočítají pomocí adaptačního kroku. Výsledky jsou vyneseny do grafu pro srovnání.

```
Bez adaptivního kroku: -3.796227794048001e+19, s adaptivním krokem: -2.2996152696719855e+20  
Bez adaptivního kroku: -1.8404629063581348e+19, s adaptivním krokem: -6.79158064906311e+18  
Bez adaptivního kroku: -2.8183453502030676e+19, s adaptivním krokem: -1.1837655380813083e+20
```

## 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 vypočítat 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:

Tento kód používá různé metody knihovny `scipy.integrate` k výpočtu integrálů tří matematických funkcí. Nejprve jsou definovány tři matematické funkce `fce1`, `fce2` a `fce3`. Následně jsou nastaveny hodnoty intervalu  $[a, b]$  a kroku `dx`. Jsou také definovány různé metody pro výpočet integrálů: `scipy_trapezoid`, `scipy_romberg`, `scipy_simpson` a `scipy_gaussian`.

Pro každou matematickou funkci se vypočítá integrál pomocí těchto metod a přesného řešení. U každé funkce je pak uveden rozdíl mezi přesným řešením a výpočtem pomocí jednotlivých metod. Cílem celého kódu je porovnat přesnost a efektivitu různých numerických metod pro výpočet integrálů matematických funkcí.

```
The difference is: 5.379749999999973
The difference is: 0.0
The difference is: 5.026049999999977
The difference is: 1.4210854715202004e-14
```

```
The difference is: -4.503834924962547
The difference is: 9.765699360286817e-11
The difference is: -4.535592372583267
The difference is: -3.97988912936853e-08
```

```
The difference is: -0.07762319856924571
The difference is: -7.416289804496046e-14
The difference is: -0.06901208931237535
The difference is: -1.1605161276406761e-11
```

## 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. Provedte také slovní okomentování toho, co lze z grafu o modelovaném procesu vyčíst.

### **Řešení:**

doplňte