STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 2: Fyzika

Počítačové modelování dynamických magnetických systémů

Ondřej Sedláček

Hlavní město Praha **Praha, 2023**

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 2: Fyzika

Počítačové modelování dynamických magnetických systémů

Computer modelling of dynamic magnetic systems

Jméno: Ondřej Sedláček
Škola: Gymnázium Christiana Dopplera, Zborovská 621, 150 00 Malá Strana
Kraj: Hlavní město Praha
Konzultant: RNDr. Pavel Josef, CSc.

Praha, 2023

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Praze dne 9. září 2023 ....

Ondřej Sedláček

Poděkování

Chtěl bych poděkovat ...

Anotace

Sem napíšeš svůj abstrakt. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Klíčová slova

Šablona, LaTeX, SOČ, ...

Annotation

Write your abstract here! Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Keywords

Template, LaTeX, High school proffessional activity, ...

Obsah

1 Úvod

Motivací pro tuto práci byla úloha mezinárodní fyzikální soutěže zvané *"International Young Physicists Tournament"*, neboli IYPT. U nás je avšak tato soutěž známější pod zkratkou TMF vycházející z překladu původního názvu - *"Turnaj mladých fyziků"*. Soutěž se v České republice kéná pod záštitout Fakulty jaderné a fyzikálně inženýrské ČVUT, FZU AV ČR, MŠMT a JČMT a z podstaty soutěže je cílem úloh dovést středoškolské studenty k vědeckému sledování nejrůznějších jevů ze všech částí fyziky.

Úloha, kterou jsem se zabýval, je v pořadí desáta úloha letošního, tedy 37., ročníku. Zadání je následovné [7]:

10. Magnetický převod

"Vezměte několik identických prstových točítek [1] *a připevněte k jejich koncům neodymové magnety. Pokud umístíte točítka v rovině vedle sebe a točíte jedním z nich, ostatní se začnou otáčet jen vlivem magnetického pole. Prozkoumejte a vysvětlete tento jev."*

Zadání úlohy je, jak je pro TMF tradiční, velmi otevřené a je tedy na řešiteli, aby si vymezil přesnou oblast svého zkoumání. Tato práce se bude zabívat:

1. Určením vlastností *prstových točítek* (dále "fidget spinner" či pouze "spinner")
2. Popisem třecích sil působících na spinner
3. Vývojem simulace chování systémů více fidget spinnerů a porovnáním této simulace s realitou
4. Přenosem úhlové rychlosti
5. Přenosem momentu síly
6. Možným využím získaných poznatků k vývoji efektivnějších magnetických převodů

[1] Z překladu anglického "Fidget spinner".

1.1 Nomenklatura

V tabulce 1 definujeme symboly, které budeme používat v průběhu celé práce, společně s jejich významem:

Tab. 1: Nomenklatura

Vlastnosti spinneru

Symbol	Jednotka	Popis	Poznámka
n		Celkový počet ramen	Ekvivaletní počtu magnetů
r	m	Poloměr spinneru	Ekvivaletní vzdálenosti osy otáčení od magnetů
S	$(m, m, 0)$	Střed spinneru v rovině	(tzn. pozice osy otáčení)
$P(i)$	$(m, m, 0)$	Pozice i. magnetu spinneru	Funkce indexu magnetu
φ	rad	Úhel rotace spinneru	
ω	rad $\cdot\, s^{-1}$	Úhlová rychlost spinneru	
I	$kg \cdot m^2$	Moment setrvačnosti spinneru	
α	rad $\cdot\, s^{-2}$	Koeficient rychlostně nezávislého brzdného úhlového zrychlení	viz Kap. 4
β	s^{-1}	Koeficient lineárně závislého brzdného úhlového zrychlení	viz Kap. 4
γ	rad^{-1}	Koeficient kvadraticky závislého brzdného úhlového zrychlení	viz Kap. 4
c_1, t_{max}	s	Celková délka otáčení spinneru	viz Kap. 4

Vlastnosti magnetu

Symbol	Jednotka	Popis	Poznámka
$\vec{m}$	$A \cdot m^2$	Magnetický moment	
$\vec{B}_r$	T	Remanentní magentizace	(tzv. remanence)
V	m^3	Objem magnetu	
$F_m(\vec{r}, \vec{m}_1, \vec{m}_2)$	N	Silová interakce mezi magnetickými momenty $\vec{m}_1$ a $\vec{m}_2$ vzdálenými o $\vec{r}$	[1]
$B(\vec{r}, \vec{m})$	T	Magnetická indukce tvořená magnetickými momentem $\vec{m}$ ve vzdálenosti $\vec{r}$	[1, 2]
τ_F, τ_{mag}	Nm	Momenty sil působící na spinner vycházející ze silové a magnetické interakce	[2, 3], viz Rov. 31

Dále stojí za zmínku, že pro vyjádření chyby měření je v textu používána tzv. *shorthand error notation* [5] (pro naše účely zkráceno na SEN). Pro jasnost uvedeme příklad, kde zápis pomocí SEN vypadá takto: 11.5(12), a ekvivalentní přepis do standardní notace je: 11.5 ± 1.2. Tímto zjednodušíme zápis: $11.5 \pm 1.2 = 11.5(12)$ [6].

2 Úvodní sledování

Prvním krokem v řešení této úlohy bylo kvalitativní sledování jejich chování v co největším rozpětí konfigurací, abychom mohli určit relevantní parametry a odstranit nezajímavé konfigurace.

Hlavním poznatkem je změna chování systému dvou pinnerů v závislosti na jejich relativních rychlostech. Máme-li na stole 2 spinnery, ze kterých je jeden nehybný, a druhý roztočíme na nízké otáčky, dojde po krátké chvíli k silné, ale chaotické, interakci (viz příloha 1). Naopak, roztočíme-li druhý spinner znatelně rychleji, nedochází téměř k žádné interakci (viz příloha 2). Druhý spinner se prvnímu spinneru efektivně jeví jako permanentí magnet - druhý magnet se tedy nanejvýše umístí do energeticky nejvýhodnější polohy a dále zůstává nehybný.

Obr. 1: Spinner osazeným neodymovými magnety

Dalším cenným poznatkem je, že při interakci více spinnerů se systém chová chaoticky téměř vždy (viz příloha 3). Toto pro nás dělá měření interakcí více jak dvou spinnerů nepříznivé a k získání použitelných výsledků je důležité omezit naše bádání pouze na jeden či dva spinnery. Poté, co kvalitně popíšeme menší počet spinnerů, se můžeme pomocí simulace pokusit o extrapolování našeho modelu na více spinnerů.

Obr. 2: Tři interagující spinnery

Z přesného zadání můžeme také vyčíst nějaké důležité předpoklady. Jmenovitě se jedná o umístění všech spinnerů v jedné *"rovině vedle sebe"*, což z velké míry usnadní budoucí výpočty, točením pouze *"jedním z nich"* a omezení interakcí mezi spinnery pouze na *"vlivy magnetického pole"*. Neměli bychom opomenout ani skutečnost, že všechny spinnery mají být *"identické"*.

2.1 Určení parametrů

Z úvodního sledování není těžké určit relevantní parametry a vybrat, které z nich je možné s naším vybavením měřit.

2.1.1 Parametry týkající se konfigurace spinnerů

Jakožto nejdůležitější bychom určitě označili relativní pozice všech spinnerů, které popíšeme pomocí jejich středů $S_1, S_2, \ldots$ (každý střed je braný jako vektor ve spinnerové rovině) a jejich poloměrů $r_1, r_2, \ldots$. Dále bude k přesnému určení pozicí magnetů nezbytné znát okamžité úhly rotace spinnerů, které ozačíme $\varphi_1, \varphi_2, \ldots$. Nakonec k popsání spinneru musíme určit počet ramen, neboli počet připevněných magnetů. Tento počet označíme n a pro naše spinnery platí $n = 3$.

Pomocí těchto údajů je triviální vyjádřit pozici $P(i)$ libovolného magnetu pomocí jeho indexu i (kde $0 \leq i < n$):

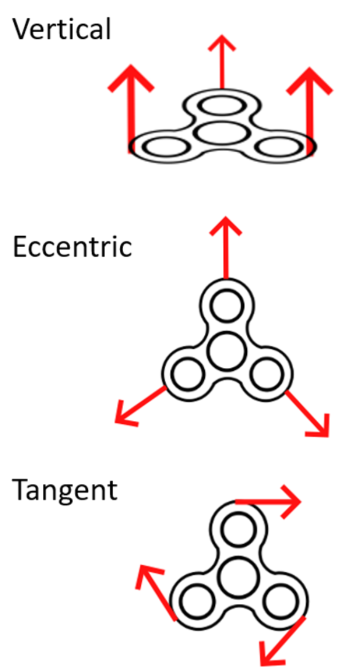
$$P(i) = S + \left(r\cos\left(\varphi + \frac{2\pi i}{n}\right), r\sin\left(\varphi + \frac{2\pi i}{n}\right), 0\right) \tag{1}$$

Kromě pozice magnetu hraje také klíčovou roli nasměrování jeho pólů. Zde definujeme 3 důležité orientace pólů (viz Obr. 3):

1. **Vertikální (*vertical*)**
2. Odstředivá (*eccentric*)
3. Tečná (*tangent*)

V našich experimentech jsme používali převážně vertikální konfiguraci, jelikož takto bylo uchycení magnetů nejjednodušší. Magnety se totiž samy připevnily ke kovovému závaží v každém z ramen spinneru (viz Obr. 1), které je zde z důvodu zvýšení momentu setrvačnosti.

Tyto konfigurace jsme také schopni popsat a to opět jakožto funkci indexu magnetu. Nejdříve vyjádříme směrový vektor $\vec{u}(i)$ pro i. magnet v každé konfiguraci:



Obr. 3: Tři námi vyhranění orientace magnetů

Tab. 2: Směrové vektory pro různé konfigurace

Konfigurace	**Směrový vektor magnetu**
Vertikální	$\vec{u}(i) = (0, 0, 1)$
Odstředivá	$\vec{u}(i) = \widehat{P(i) - S}$
Tečná	$\vec{u}(i) = (0, 0, 1) \times \widehat{(P(i) - S)}$

Když nyní zvolíme velikost magnetického momentu našich magnetů $|\vec{m}_0|$, jsme schopni popsat magnetický moment včetně jeho velikosti [2]:

$$\vec{m} = |\vec{m}_0| \cdot \vec{u}(i) \tag{2}$$

Velikost magnetických momentů bude záviset na teplotě, velikosti a materiálových vlastnostech magnetů (např. jejich chemickém složení a kvalitě), ale to, jaká je pravá velikost magnetických momentů $|\vec{m}_0|$, je momentálně nepodstatné a určíme ji později (viz Kap. 3.3).

Posledním parametrem, který zmíníme, ale nebudeme se jím zabývat, je přitahování, či odpuzování magnetů. V našem případě jsem se zaměřili převážně na systémy, kde se všechny magnety odpuzují.

2.1.2 Parametry týkající se pohybu spinnerů

Druhou, složitější, částí popisu našeho systému je jeho pohyb a chování v čase. Zde se nevyhneme úhlovým rychlostem jednotlivých spinnerů, které budeme značit $\omega_1, \omega_2, ...$ Poté by nás přirozeně napadlo úhlové zrychlení $\alpha_1, \alpha_2, ...$, ale pro náš případ bude šikovnější využít toho, že $\tau = I\alpha$, kde τ značí moment síly a I značí moment setrvačnosti[3] spinneru. Moment setrvačnosti určíme později ve své vlastní kapitole (viz Kap. 3.2).

Posledním parametrem, který zmíníme, je tření v ložiscích spinnerů a jiné odporové síly. Těm se budeme do hloubky věnovat v kapitole 4.

[2] Všimněme si, že $u(i) = \hat{u}(i)$.

[3] Někdy také značeno J.

2.2 Modelování magnetů

V průběhu našich experimentů používáme neodymové (NdFeB) magnety krychlového tvaru o hraně 5mm a jakosti N35. [4] Toto označení jakosti neodymových magnetů popisuje jejich chemické složení, tepelnou odolonost a hlavně sílu [8], která je popsána pomocí tzv. remanentní magnetizace, neboli remanence.

Jelikož jsou magnety poměrně malé, můžeme je ve větších vzdálenostech aproximovat jakožto magnetické dipóly. Zároveň existuje velmi elegantní způsob, jak vypočítat velikost magnetického dipólu z jeho remanence [2]:

$$|\vec{m}_0| = \frac{1}{\mu_0}|\vec{B}_r|V \tag{3}$$

Tabulkové hodnoty pro remanenci NdFeB magnetů jsou sice známé, ale v našem případě budeme přesnou hodnotu $|\vec{B}_r|$ našich magnetů měřit později, v kapitole 3.3.

2.3 Popis magnetických interakcí

Jelikož k popisu magnetů používáme idealizaci pomocí magnetických dipólů, můžeme popsat interakce mezi nimi pomocí následujících rovnic. Hlavní interakce mezi dvěma magnetickými dipóly jsou:

1. Silové interakce [1] mezi dvěma momenty $\vec{m}_1$ a $\vec{m}_2$, které jsou od sebe vzdáleny $\vec{r}$ [5]:

$$F_m(r, m_1, m_2) = \frac{3\mu_0}{4\pi||r||^5}\left[(m_1 \cdot r)m_2 + (m_2 \cdot r)m_1 + (m_1 \cdot m_2)r - \frac{5(m_1 \cdot r)(m_2 \cdot r)}{||r||^2}r\right] \tag{4}$$

2. Magnetické interakce, neboli působení momentu síly [2] na $\vec{m}_2$ z důvodu vytvoření magnetické indukce $B(r, m_1)$ momentem $\vec{m}_1$ [1]:

$$\begin{gathered} B(r, m) = \frac{\mu_0}{4\pi}\frac{3\hat{r}(\hat{r} \cdot m) - m}{|r|^3} \\ \tau = m_2 \times B(r, m_1) \end{gathered} \tag{5}$$

[4] Jakosti neodymových magnetů se pohybují od N35 do N55 [8].

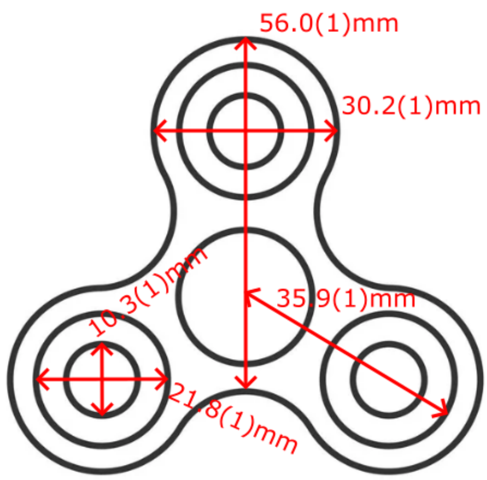
[5] $\vec{r} = P_1 - P_2$

3 Měření parametrů

3.1 Rozměry spinneru

K popsání spinneru nejdříve určíme jeho rozměry (viz Obr. 4). Nejdůležitější rozměr pro nás bude vzdálenost osy otáčení od místa, kde budou umístěny magnety. To je v našem případě $r = 35.9(1)mm$.

Vzhledem k tomu, že na sebe spinnery mohou vzájemně působit pouze magnetickými silami, jak je řečeno v zadání, není třeba řešit zbytek geometrie spinneru v budoucích modelech. Spinner budeme dále modelovat pouze jako n magnetických dipólů, které se otáčejí kolem středu S a pevně si udržují své relativní pozice. Tomuto celkovému systému definujeme moment setrvačnosti podle následujícího měření.



Obr. 4: Ilustrace spinneru společně s vyznačenými rozměry

3.2 Moment setrvačnosti

K určení momentu setrvačnosti spinneru bylo využito idealizace spinneru jakožto fyzického kyvadla, pro které platí: [9]

$$\omega = \sqrt{\frac{mgr}{I}} \tag{6}$$

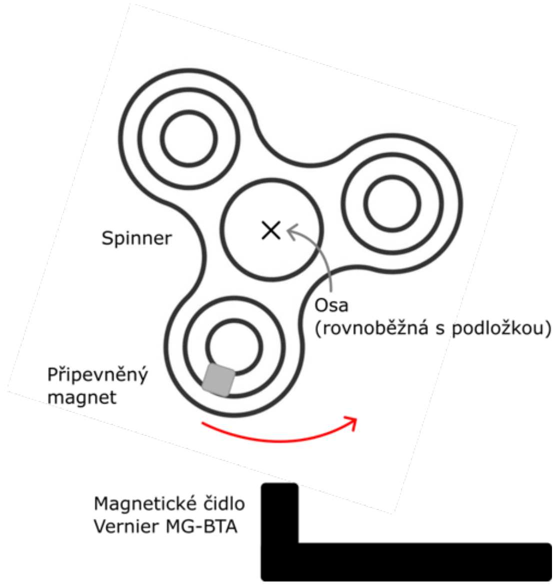
A po vyjádření z rovnice 6 tedy získáváme:

$$I = \frac{mgr}{\omega^2} = \frac{mgr}{4\pi^2 f^2} \tag{7}$$

Nyní stačí změřit frekvenci kmitů zavěšeného spinneru s několika magnety na jednom z jeho ramen.

3.2.1 Aparatura

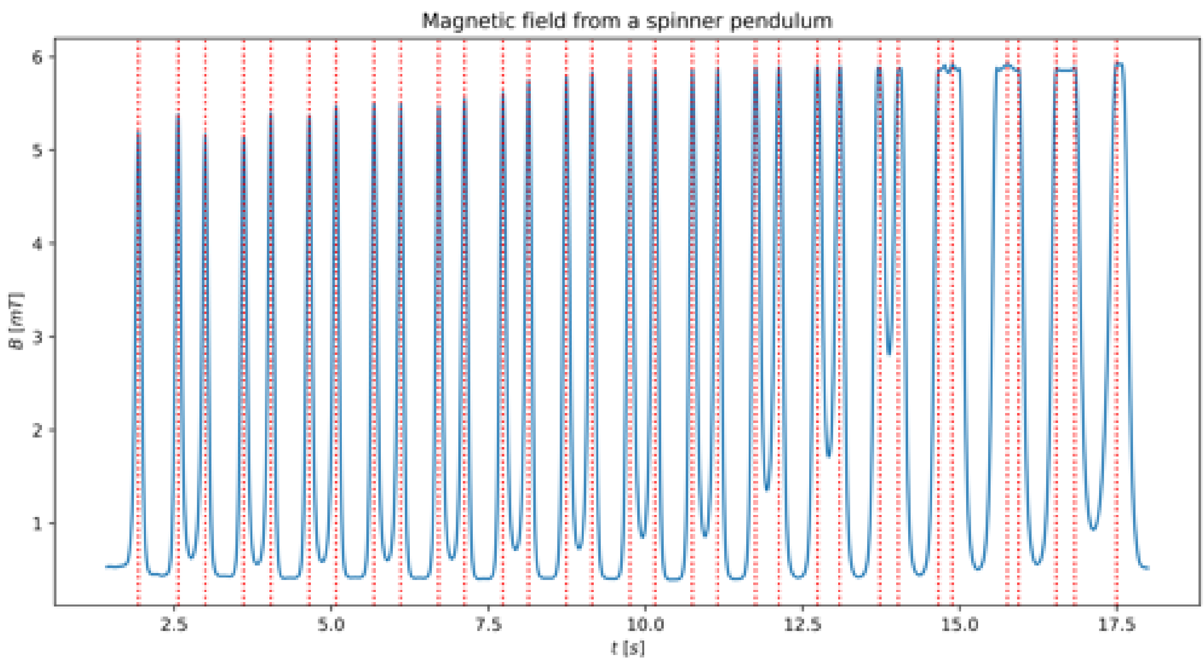
Po připevnění šesti neodymových magnetů o jednotkové hmotnosti 0.92(1)g byl spinner připevněn pevně ve své ose otáčení. Pod takto zavěšený spinner bylo umístěno Magnetické čidlo Vernier MG-BTA, které zaznamenávalo, jak blízko se magnety nacházejí. Po umístění spinneru mimo rovnovážnou polohu byl ponechán oscilovat a záznam průběhu magnetické indukce v čase byl čidlem nasnímán (viz Obr. 6).



Obr. 5: Ilustrace aparatury pro měření frekvence kmitů spinneru

3.2.2 Analýza měření a výsledky

Průběh následně analyzujeme pomocí programovacího jazyka `Python` [10] a knihovny `scipy` [11]. Námi kýžená maxima vyhledáme použitím funkce `scipy.signal.find_peaks` a z jejich počtu jsme schopni určit počet period, kterými kyvadlo prošlo za určený čas.



Obr. 6: Nasnímaný průběh magnetické indukce v čase (modře) pro spinnerové kyvadlo. Peaky, označující uplynutí jedné půlperiody, jsou vyznačeny červeně.

Máme-li tedy pole p[6] všech časů nalezených peaků, je výpočet frekvence následující:

$$f = \frac{|p| - 1}{2(p_{max} - p_{min})} = 1.00(1)Hz \tag{8}$$

Dalším dosazením do předchozí rovnice 7 získáváme výslednou hodnotu[7]:

$$I = 4.80(50) \cdot 10^{-5} kg \cdot m^2 \tag{9}$$

3.2.3 Popis skriptu

V níže připnutém ústřižku kódu 1 nejdříve načteme použité knihovny: `numpy` [12] pro provedení konvoluce, `matplotlib` pro vytvoření grafu 6 [13] a nakonec `scipy`, kde využijeme funkce `find_peaks`, jak již bylo zmíněno.

Prvním krokem je načtení dat ze souboru `kyvadlo.csv` a jejich převedení z textového fomrátu do 2D pole pojmenovaného `data`. Dále izolujeme data jednotlivých os do separátních proměnných `xdata`, `ydata`.

`ydata` vyhladíme, abychom se zbavili šumu a to pomocí konvoluce skrze funkci `numpy.convolve`. Délku 1D konvoluční matice jsme zvolili $m = 15$ a konvoluční matice délky m v našem případě vypadá obecně takto:

$$M_{conv} = \underbrace{\begin{pmatrix} \frac{1}{m} & \frac{1}{m} & \dots & \frac{1}{m} \end{pmatrix}}_{m \text{ krát}} \tag{10}$$

Tato konvoluční matice přesněji provádí aritmetický průměr m za sebou jdoucích hodnot.

Nyní provedeme hledání peaků na vyhlazenách datech pomocí funkce `find_peaks`, která akceptuje další dobrovolné argumenty:

1. height: určuje minimální absolutní velikost peaku
2. threshold: minimální rozdíl dvou sousedních hodnot, aby bod mohl být peakem
3. distance: minimální indeová vzdálensot dvou peaků

Poté určíme časový interval, kde se peaky nacházejí (a přidáme na každou stranu 0.5 sekundy pro přehlednost grafu), a vygrafujeme půdovní naměřená data (modře) společně s pozicemi peaků (červeně). Dále ozačíme osy.

Nakonec dopočítáme frekvenci kmitů kyvadla podle vzorce 8 a vypíšeme ji.

```
# imports
```

[6] $|p|$ označuje počet prvků pole p; p_{min} označuje čas prvního peaku; p_{max} označuje čas posledního peaku.

```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.signal as sig

# helper function to convert the map and filter iterators to a list
lmap = lambda f,x : list(map(f, x))
lfilt = lambda f,x : list(filter(f, x))

with open("./kyvadlo.csv", "r") as read:
    data = read.readlines()[1:] # load all the value rows
    # convert all rows into value arrays
    data = lmap(lambda row: row.rstrip().split(","), data)

    xdata = lmap(lambda row: float(row[0]), data) # x axis data
    ydata = lmap(lambda row: float(row[1]), data) # y axis data
    mov = 15 # the amount of smoothing
    # smooths out the data using convolution
    ydata_clean = np.convolve(ydata, np.ones(mov)/mov, "same")

    # find the peaks from the data using scipy.signal.find_peaks
    peaks, heights = sig.find_peaks(
        ydata_clean,
        height=1,
        threshold=0.0001,
        distance=35
    )
    # get the sampling frequency
    samples_per_sec = (len(xdata) / max(xdata))
    # find the start and end time of where the peaks are
    peaks_s = peaks[0] - int(samples_per_sec / 2)
    peaks_e = peaks[-1] + int(samples_per_sec / 2)

    # plot the smoothed out data
    plt.plot(xdata[peaks_s: peaks_e], ydata_clean[peaks_s: peaks_e])
    # plot all the peak lines
    for p in peaks:
        plt.axvline(p / samples_per_sec, color="r", linestyle=":")

    # set all other styling of the plot
    plt.title("Magnetic field from a spinner pendulum")
    plt.ylabel("$B$ $[mT]$")
    plt.xlabel("$t$ $[s]$")

    # calcualte and print out the resulting frequency
    f = 1/2 * ((len(peaks)-1) / ((max(peaks) - min(peaks)) /
   samples_per_sec))
    print(f"{f:.2f} Hz")
```

Ústřižek kódu 1: Kód k analyzování spinnerového kyvadla

3.3 Remanence $\vec{B}_r$

K určení remanence našich magnetů využijeme dvou různých metod. Jedna z nich bude založena na silové interakci magnetů a jedna na magnetem vytvořeném poli.

3.3.1 Určení remanence přes magnetické pole

Prvním způsobem, jak určit přesnou hodnotu remanentní magentizace našich magnetů, je měření magnetického pole pomocí magnetometru mobilního telefonu. Aplikace `Phyphox` umožňuje čtení těchto dat, která následně analyzujeme.

Měření provedeme pro sloupce 2 a 3 magnetů, abychom ověřili, zda je náš způsob měření konsistentní. Dále také měříme větší neodymový magnet, jehož parametry nám nejsou známé, a který budeme používat v některých budoucích experimentech. Abychom odstranili pozadí tvořené magnetickým polem země, provedeme měření magnetického pole jednou a poté magnet otočíme. K získání výsledku jednoduše odečteme rozdíl naměřených hodnot, což bude dvojnásobek velikosti magnetického pole tvořeného magnetem. Je zřejmé, že tímto odstraníme z meření jiné, neměnné, zdroje magnetických polí.

Kromě provádění měření pro různé počty a velikosti magnetů, budeme také meřit pole v různých vzdálenostech od magnetu, abychom dále potvrdili, zda je měření a náš model konzistetní. Celkem provedeme tedy měření 2 magnetů, 3 magnetů a velkého magnetu pro vzdálenosti 5 cm, 10 cm, 20 cm a 30 cm.

Analýza měření

Výstupem aplikace `Phyphox` jsou sobory typu `.csv`, neboli *"Comma separated values"* [16]. Toto je poměrně čitelný a přívětivý formát ukládání dat, která v `Pythonovém` skriptu načteme pomocí funkce `open`, stejně jako v kódu 1. Jinak budeme používat tyto knihovny: `numpy`, `matplotlib` pro vytvoření grafu 7, `scipy`, kde využijeme funkce `curve_fit`, a `math` [14]. Poté ručně určíme názvy všech datových souborů a časov é intervaly, na kterých jsou magnety orientovány k snímači a od snímače.

Po načtení dat vyjmeme pouze hodnoty ve dříve zmíněných intervalech a najdeme jejich průměrnou hodnotu a odchylky. Z rozdílu průměrných hodnot magnetického pole v obou otočeních magnetů určíme velikost magnetického pole generovaného pouze magnetem (očištěného od pozadí). Z odchylek také určíme celkovou odchylku výsledné hodnoty.

Následně pro každé měření vytvoříme graf (příkladem je Obr. 7).

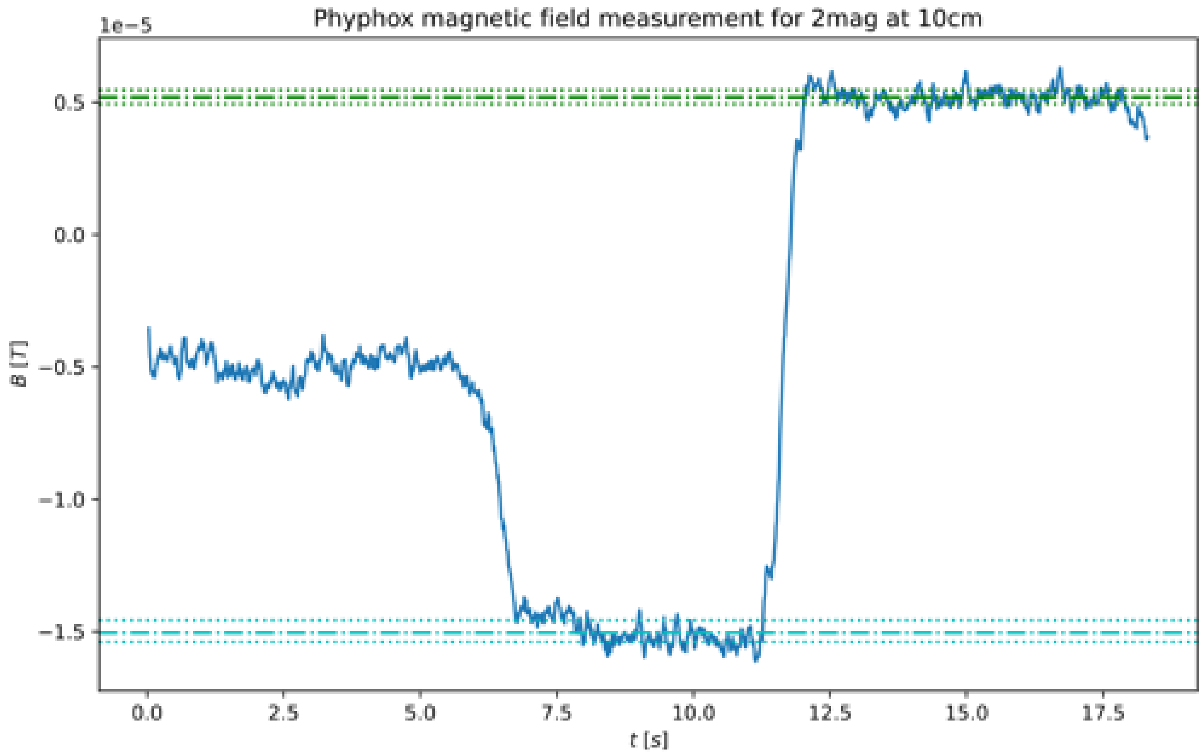
```
down_vals = lmap(lambda j: ydata[j], lfilt(lambda j: down_intervals[i
    ][0] < xdata[j] < down_intervals[i][1], range(len(ydata))))
up_vals = lmap(lambda j: ydata[j], lfilt(lambda j: up_intervals[i][0] <
    xdata[j] < up_intervals[i][1], range(len(ydata))))

down_avg = np.average(down_vals)
up_avg = np.average(up_vals)

amp = (up_avg - down_avg) / 2

down_error = [
    np.average(lmap(lambda x: down_avg - x, lfilt(lambda x: x <=
    down_avg, down_vals))),
    np.average(lmap(lambda x: x - down_avg, lfilt(lambda x: x > down_avg
    , down_vals)))
]
up_error = [
    np.average(lmap(lambda x: up_avg - x, lfilt(lambda x: x <= up_avg,
    up_vals))),
    np.average(lmap(lambda x: x - up_avg, lfilt(lambda x: x > up_avg ,
    up_vals)))
]
```

Ústřižek kódu 2: Kód k určení průměrné hodnoty magnetického pole a odchylek



Obr. 7: Nasnímaný průběh magnetické indukce v čase (modře) pro určení remanence. Průměrné hodnoty horní orientace (zeleně čerchovaně) a dolní orientace (modře čerchovaně). Odpovídající odchylky jsou znázorněny tečkovaně. Výsledná hodnota magnetické indukce tohoto příkladu, tedy dvou magnetů ve vzdálenosti 10 cm, je: $B = 1.01(8) \cdot 10^{-5} T$

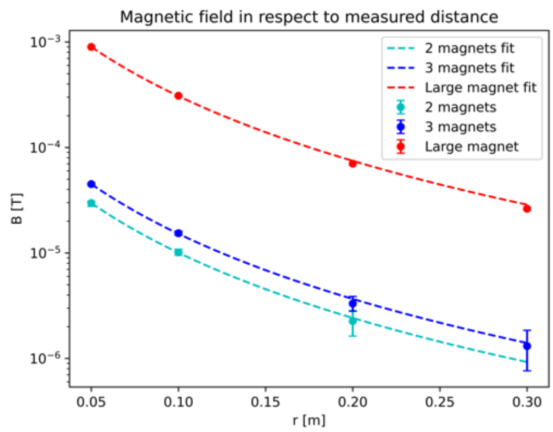
Po zanalyzování všech měření můžeme vytvořit graf závislosti B na vzdálenosti (viz Obr. 8):

Pro určení B_r z naměřených hodnot provedeme zjednodušení vzorce 5, kdy nahradíme vektory pouze jejich velikostmi:

$$|B| = \frac{\mu_0 |m|}{2\pi r^3} \tag{11}$$

Zde si všímáme úměrnosti $B \propto r^{-3}$, kterou také sledujeme v našich naměřených datech. Následným dosazením rovnice 3 a vyjádřením remanence získáváme:

$$|B_r| = \frac{2\pi r^3 |B|}{V} \tag{12}$$



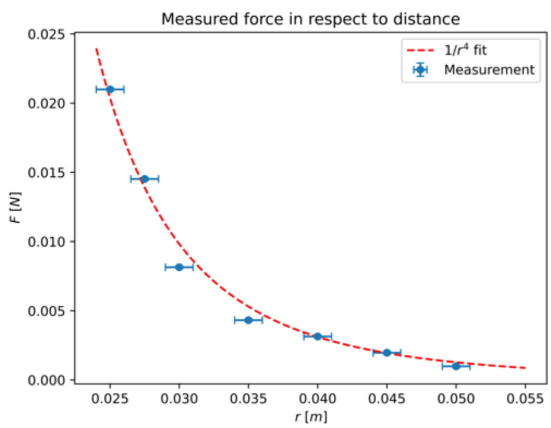
Obr. 8: Závislost magnetické indukce na vzdálenosti. Sledujeme, že $B \propto r^{-3}$

Pro každou naměřenou hodnotu B dopočítáme odpovídající B_r a výsledky zobrazíme v jednotném grafu 10.

3.3.2 Určení remanence přes sílu

Druhý způsob, kterým jsme schopni určit remanenci, je skrze přitahování (resp. odpuzování) jednoho magnetu k (resp. od) druhého magnetu. Posazením jednoho magnetu na váhu a sledováním změny měřené hmotnosti jsme z gravitačního zrychlení schopni dopočítat působící sílu druhého magnetu, který je umístěn v určité výšce. Zároveň první magnet podložíme lehkou, tlustou a nemagnetickou vrstvou (např. blokem polystyrenu), abychom zabránili interakci magnetu s kovovými částmi váhy.

Výhodou této metody je, že jsme schopni přesněji měřit slabší magnety, jelikož je můžeme dát blíže. V předchozí kapitole jsme nebyli schopni přiblížit magnety dostatečně blízko ke snímači telefonu kvůli jeho rozměrům. Toto měření provádíme pouze pro 1 magnet, jelikož pro ten nám chybí výsledky z měření pomocí magnetického pole (ze dříve zmíněných důvodů). Výsledky můžeme vidět na grafu 9.



Obr. 9: Závislost magnetické síly na vzdálenosti. Sledujeme, že $F \propto r^{-4}$

Výpočet remanence provedeme podobně jako u rovnice 11 a to náhradou vektorů za jejich velikosti v rovnici 4. Tím jsme ji schopni zjednodušit na následující podobu:

$$|F| = \frac{3\mu_0}{2\pi r^4}|m|^2 \tag{13}$$

Po dosazení rovnice 3 a vyjádřením remanence získáváme:

$$B_r = \sqrt{\frac{2\pi\mu_0 r^4 F}{3V^2}} \tag{14}$$

Výsledné hodnoty B_r zobrazíme v jednotném grafu 10.

3.3.3 Výsledky

Výsledky obou způsobů měření sjednotíme do grafu závislosti B_r na vzdálenosti (viz Obr. 10). Očekávali bychom, že se B_r v závislosti na vzdálenosti nebude měnit, a pro každé měření určíme pomocí optimalizace fitu konstantní funkcí souhrnnou hodnotu. Výsledkem všech měření bude aritmetický průměr těchto souhrnných hodnot.

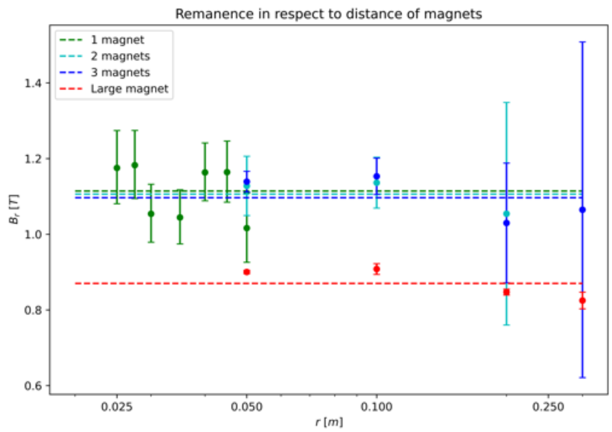
Výsledná hodnota remanence našich krychlových magnetů je [8]:

$$B_r = 1.10(13)T$$

a našeho velkého magnetu je:

$$B_{r_{xl}} = 0.87(5)T$$

Důvodem, proč je remanence většího magentu znatelně nižší, může být skutečnost, že takto velký magnet náš model není schopen popsat jako jediný magnetický dipól, nebo například jeho stáří.



Obr. 10: Souhrnný graf všech měření remanence. Ve výpočtu výsledné hodnoty B_r není započítávána remanence velkého magnetu, jelikož není součástí stejné várky magnetů a budeme s ním nakládat později jinak.

[8] Námi naměřené hodnoty jsou srovnatelné s externími zdroji (viz. [8]).

4 Tření

O třecích jevech při otáčení spinneru jsme si již byli schopni udělat představu při úvodním kvalitativním sledování. Tření bude hrát značnou roli v našem systému, jelikož to bude hlavní limitující faktor průběhu chaotického chování. Dalším využitím, kromě našeho úhlavního záměru - zpřesnění popisu chování, bude také například možnost odhadnutí maximálního času, než se spinnery zastaví a systém ztratí všechnu svou energii. Zajímavým, a lehko měřitelným, pro nás bude průběh úhlové rychlosti ω (resp. změna úhlové rychlosti [9]) v čase, že kterého tkví i náš popis těchto třecích složek.

Zajímá-li nás tedy změna rychlosti, dovolíme si odhadnout, že hlavními zdroji tření bude ložisko a odpor vzduchu. Provedeme idealizaci ložiska a řekneme, že jeho brzným moment síly, a tím pádem i úhlové zpomalení, bude neměnné v čase a nezávislé na rychlosti. Toto nemusí být vždy pravda, např. objeví-li se v našem, přirozeně neideálním, ložisku nějaké nechtěnné oscilace (které jsme sledovali) nebo je-li na ložisko působeno nějakou externí silou mířící kolmo na osu otáčením (jako například když se od sebe velmi silně odpuzují magnety ze dvou blízkých spinnerů). Oba tyto příklady jsou příliš složité k popsání a nejsme je tedy schopni zkoumat. Toto brzdné úhlové zrychlení (resp. zpomalení) nezávislé na otáčkách spinneru označíme α.

Další výraznou složkou bude již zmíněný odpor vzduchu. Z látky středoškolské fyziky známe přibližnou úměrnost odporové síly na kvadrátu rychlosti ($F \propto v^2$) při turbulentním proudění (které v našem a případě očekáváme kvůli složité geometrii spinneru). V našem rotačním případě bychom tedy analogicky očekávali $\dot{\omega} \propto \omega^2$. Míru této závislosti označíme γ.

Poslední složkou, pro kterou nejsme schopni najít fyzikální podstatu, ale kterou přesto budeme zkoumat je lineární složka β. Existují-li tedy jevy, které nám unikly, a platí pro ně závislost $\dot{\omega} \propto \omega$, můžeme pro ně kompenzovat touto složkou. Zároveň budeme podrobně srovnávat výkony našich předpovědí s, i bez, této složky, abychom určili, zda dává smysl tuto složku využívat, či vůbec uvádět.

[9] $I\dot{\omega} = \tau$

4.1 Analytický popis

Všech tři dříve zmíněné složky dohromady tedy popisuje řídící závislost systému: [10]:

$$\dot{\omega}(t) = -\alpha - \beta\omega(t) - \gamma\omega^2(t) \tag{15}$$

Pokud bychom odstranili lineární komponent, získáváme:

$$\dot{\omega}(t) = -\alpha - \gamma\omega^2(t) \tag{16}$$

Obě tyto diferencialní rovnice jsou analyticky řešitelné a jejich řešení pro ω jsou v tomtéž pořadí [11]:

$$\omega(t) = -\frac{\sqrt{4\alpha\gamma - \beta^2}\tan\left(\frac{1}{2}\sqrt{4\alpha\gamma - \beta^2}(t - c_1)\right) - \beta}{2\gamma} \tag{17}$$

a

$$\omega(t) = -\sqrt{\frac{\alpha}{\gamma}}\tan\left(\sqrt{\alpha\gamma}(t - c_1)\right) \tag{18}$$

4.1.1 Maximální doba otáčení

Objevující se parametr c_1 [12] je hodnota v sekundách, která odpovídá době, po které se spinner zastaví. Vyjádřením c_1 (např. z druhé rovnice) jsme schopni určit poměrně přesně, jak dlouho se bude spinner točit, známe-li jeho parametry α, (β) a γ a jeho počáteční úhlovou rychlost ω_0:

$$c_1(\omega_0) = \frac{1}{\sqrt{\alpha\gamma}}\arctan\left(\sqrt{\frac{\gamma}{\alpha}}\omega_0\right) \tag{19}$$

Přáli bychom si vyjádřit celkovou dobu otáčení z počáteční kinetické energie, můžeme dosadit $E_k = \frac{1}{2}I\omega^2$, kde je nám I již známe (viz Kap. 3.2), a jednoduchou úpravou získáváme:

$$c_1(E) = \frac{1}{\sqrt{\alpha\gamma}}\arctan\left(\sqrt{\frac{2\gamma E}{\alpha I}}\right) \tag{20}$$

Jakmile dojde k tomu, že $t > c_1$, nepopisuje již funkce náš spinner, protože by začal opět nabývat energie, k čemuž nemůže dojít.

[10] Záporná znaménka používáme čistě z toho důvodu, aby nám koeficienty α, β a γ vycházely kladně, jelikož očekáváme, že systém bude zpomalovat.

[11] Všimněme si, že rovnice 17 je ekvivalentní rovnici 18, když $\beta = 0$.

[12] Někdy také označujeme t_{max}.

Horní limit doby otáčení n spinnerů

Další zajímavou vlastností, kterou jsme schopni pro tento idealizovaný systém dokázat je, že máme-li libovolné (přirozené) množství spinnerů o počátečních energiích $E_1, E_2 ... E_n$, které spolu neinteragují, je maximální dobou, po kterou se může některý z nich otáčet $t_{max} \leq c_1(\sum_{i=1}^{n} E_i)$. Tento fakt se může zdát triviálním a zbytečným, ale jsme tímto způsobem schopni např. určit horní časovou hranici, po kterou by musela běžet simulace, než bychom si byli poměrně [13] jistí, že již došlo k zastavení všech spinnerů.

Chtěli bychom-li zlepšit tento odhad i pro interagující spinnery, bylo by potřeba počítat s tím, že spinnery se mohou zrychlovat, zpomalovat a často budou oscilovat. Pokud bychom sledovali mnoho přibližně harmonických oscilací spinnerů, kdy se po relativně dlouhou dobu nepohybují svou nejvyšší rychlostí, mohli bychom se inspirovat efektivní hodnotou sinusoidy a udělat tak hrubý odhad, že $t_{max} \leq \sqrt{2} c_1(\sum_{i=1}^{n} E_i)$. Nad tím, jaká konstanta by byla dobrá můžeme dlouze polemizovat, ale to není zájmem této práce, proto nebudeme myšlenku dále ozvádět.

Tuto vložku zakončíme důkazem dříve vyslovené věty, že $t_{max} \leq c_1(\sum_{i=1}^{n} E_i)$:

Nejdříve zdůrazníme několik zřejmých faktů o naší funkci $c_1(\omega_0)$:

$$
\begin{gathered}
c_1(E) = \frac{1}{\sqrt{\alpha\gamma}} \arctan\left(\sqrt{\frac{2\gamma E}{\alpha I}}\right) \\
c_1(E) \text{ je rostoucí a prostá funkce pro } \alpha, \gamma \in \mathbb{R}^+ \\
E_1, E_2, ..., E_n \in \mathbb{R}^+ \implies c_1(E_1), ..., c_1(E_n) \in \mathbb{R}^+
\end{gathered} \tag{21}
$$

Bez újmy na obecnosti poté můžeme říci:

$$
E_1 \leq E_2 \leq ... \leq E_n \in \mathbb{R}^+ \implies c_1(E_1) \leq ... \leq c_1(E_n) \in \mathbb{R}^+ \tag{22}
$$

A obecně vybereme energie E_u a E_v takové, že $E_u \geq E_v$ $(u \neq v \land 1 \leq u, v \leq n)$, pro které dokážeme následující:

$$
c_1(E_u + E_v) \overset{?}{\geq} \max(c_1(E_u), c_1(E_v)) \tag{23}
$$

[13] Nemůžeme si být kompletně jistí kvůli interakcím spinnerů a nepřestnostem simulace.

Důkaz

$$c_1(E_u + E_v) \geq \max(c_1(E_u), c_1(E_v))$$

$$c_1(E_u + E_v) \geq \max(c_1(E_u), c_1(E_v)) \equiv c_1(E_u + E_v) \geq c_1(E_u) \because (c_1(E_u) \geq c_1(E_v) \because E_u \geq E_v)$$

$$\begin{gathered}
c_1(E_u + E_v) \geq c_1(E_u) \\
\frac{1}{\sqrt{\alpha\gamma}} \arctan\left(\sqrt{\frac{2\gamma(E_u + E_v)}{\alpha I}}\right) \geq \frac{1}{\sqrt{\alpha\gamma}} \arctan\left(\sqrt{\frac{2\gamma E_u}{\alpha I}}\right) \\
\arctan\left(\sqrt{\frac{2\gamma(E_u + E_v)}{\alpha I}}\right) \geq \arctan\left(\sqrt{\frac{2\gamma E_u}{\alpha I}}\right) \\
\sqrt{\frac{2\gamma(E_u + E_v)}{\alpha I}} \geq \sqrt{\frac{2\gamma E_u}{\alpha I}} \\
\sqrt{\frac{2\gamma}{\alpha I}}\sqrt{E_u + E_v} \geq \sqrt{\frac{2\gamma}{\alpha I}}\sqrt{E_u} \\
\sqrt{E_u + E_v} \geq \sqrt{E_u} \\
E_u + E_v \geq E_u \\
E_v \geq 0 \\
\text{QED}
\end{gathered} \tag{24}$$

Dále pouhou analogií pro libovolné dvojice, větší k-tice a konečně celou n-tici energií je zřejmé, že vyplývá:

$$\begin{gathered}
\forall E_j; (1 \leq j \leq n) : c_1(E_j) \leq c_1\left(\sum_{i=1}^{n} E_i\right) \\
\Updownarrow \\
t_{max} \leq c_1\left(\sum_{i=1}^{n} E_i\right) \\
\text{QED}
\end{gathered} \tag{25}$$

4.1.2 Určení jednotek parametrů α, β a γ

Přepsáním rovnice 15 pouhými jednotkami jsme jednoduše schopni určit jednotky koeficientů α, β a γ:

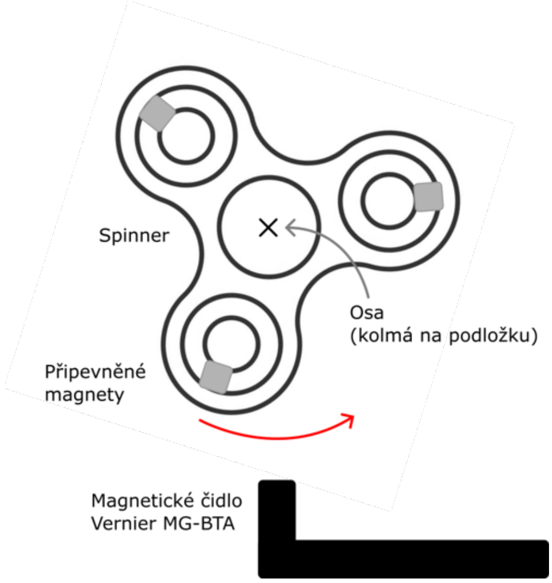
$$\begin{gathered}
[rad \cdot s^{-2}] = -[\alpha] - [\beta][rad \cdot s{-}1] - [\gamma][rad \cdot s]^2 \\
[rad \cdot s^{-2}] = -[rad \cdot s^{-2}] - [s^{-1}][rad \cdot s{-}1] - [rad^{-1}][rad^2 \cdot s^{-2}] \\
\Longrightarrow [\alpha] = [rad \cdot s^{-2}]; [\beta] = [s^{-1}]; [\gamma] = [rad^{-1}]
\end{gathered} \tag{26}$$

4.1.3 Experiment pro potvrzení analytického řešení

Abychom ověřili, zda výsledky našich diferencialních rovnic odpovídají realitě, budeme sledovat jak se snižuje úhlová rychlost každého z našich spinnerů. K tomu využijeme opět magnetického čidla Vernier MG-BTA, které bude sledovat momenty, kdy se kolem něj mihne jeden z magnetů připevněných na ramenou spinneru (velmi podobně jako u Obr. 5). Časová hustota těchto vrcholů je úměrná úhlové rychlosti - přesněji 1 vrchol za sekundu se rovná $\frac{2}{3}\pi\ rad/s$.

Aparatura

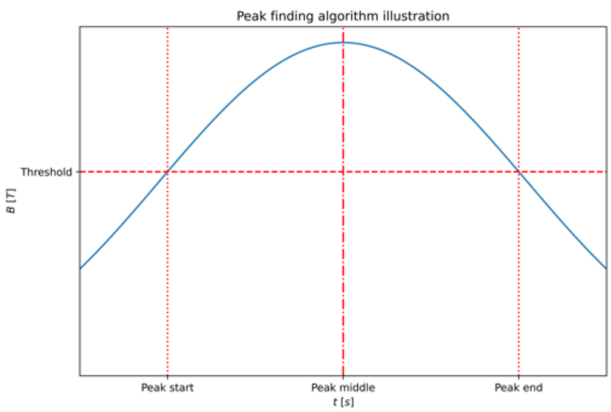
Aparatura tohoto experimetu (vyobrazená na Obr. 11) je velmi podobná té na Obr. 5. Jediným rozdílem je, že nyní má spinner osazená všechna ramena a to pouze jedním magentem. Zároveň není zavěšen ve své ose, ale leží na podložce a je tedy jeho osa otáčení kolmá na prodložku (resp. rovnoběžná se směrem gravitačního zrychlení). V této orientaci budou prováděna i všechna následující měření v dalších kapitolách.



Obr. 11: Ilustrace aparatury pro měření přibližné úhlové rychlosti spinneru

Zpracování dat

Ke zpracování dat opět použijeme jazyka `Python` a stejných knihoven jako doposud. Prvním úkolem je převedení průběhů kmitů magnetického pole (způsobených mihnutím magnetu před čidlem) do průběhu úhlové rychlosti v čase. K tomto bude potřeba určit peaky a jejich pozici v čase a následně konvolucí počítat množství peaků na určitých intervalech. Pozice peaků určíme tak, že budeme sledovat kdy naměřená hodnota překročí nějaký práh. Toto bude začátek našeho peaku. Když půjde zpět pod hodnotu prahu, ukončíme peak a pozici maxima řekneme, že je ve středu tohoto časového intervalu (viz Obr. 12).



Obr. 12: Ilustrace fungování algoritmu pro hledání peaků

```python
# imports
import math
def lmap(f,x):
    return list(map(f,x))

# function that converts the B/t relation to omega/t relation
def process_raw_decay_data(
        id,
        startTime = -1,
        threshold = 0.7):
    # the times of the found peaks
    peaks = []
    with open(f"./inputs/Spinner_{id}_decay_rate.csv") as inp:
        # parse in the data lines
        lines = inp.read().split("\n")
        # find the starting index from which the data points are after startTime
        startInd = 1
        while (float(lines[startInd].split(",")[0]) < startTime):
            startInd += 1

        hitPeak = False # keeps track of if we found the sttart of a peak
        startPeak, endPeak = 0 # bounding times of the peak
        for i in range(
            startInd,
            len(lines)-1):
            # going through each line
            row = lines[i].split(",")

            # if we go over the threshold, the peak has started
            if (float(row[1]) >= threshold and not hitPeak):
                # we start the peak
                hitPeak = True
                startPeak = float(row[0])

            # if we go back under and have found a start
            if (float(row[1]) < threshold and hitPeak):
                # we end the peak
                hitPeak = False
                endPeak = float(row[0])
                # and save it's middle value
                peaks.append((endPeak+startPeak)/2)
```

Ústřižek kódu 3: Kód k hledání pozic peaků

Druhým krokem je projít celý čas měření a rozdělit ho na menší intervaly. V těchto intervalech spočítáme počet peaků a přepočtem 1 $peak = \frac{2\pi}{3}\ rad/s$ určíme přibližnou průměrnou rychlost v tomto okolí.

```
    results = {}
    interval = 1
    step = 0.1
    curTime = startTime
    x = 0
    while x < len(peaks):
        if (peaks[x] > curTime+interval):
            # if the peaks are positioned outside the interval stop
            # and reposition the interval by step
            curTime += step
            x = 0
        elif (peaks[x] > curTime and peaks[x] < curTime+interval):
            # a peak is positioned in the time interval
            if (not curTime in results): results[curTime] = 0
            results[curTime] += 1 # increase the count amount
        x += 1

    with open(f"./outputs/Spinner_{id}_decay_rate_PROCESSED_FLOATING_AVERAGE.csv", 'w') as res:
        res.write("t, peaks/s, omega \n")
        # go through all the results
        for x in range(4, len(results.keys())):
            # create a 5 element floating average
            flAverage = 0
            for y in range(5):
                flAverage += results[list(results.keys())[x-y]]
            flAverage = flAverage/5

            # write the time, avg. peak count and avg. omega to file
            res.write(str(round(list(results.keys())[x-y], 1)) + "," + str(flAverage) + "," + str(flAverage*2*math.pi/3) + "\n")
    # return the output file location
    return f"./outputs/Spinner_{id}_decay_rate_PROCESSED_FLOATING_AVERAGE.csv"
```

Ústřižek kódu 4: Pokračování kódu 3 - výpočet úhlové rychlosti pomocí konvoluce

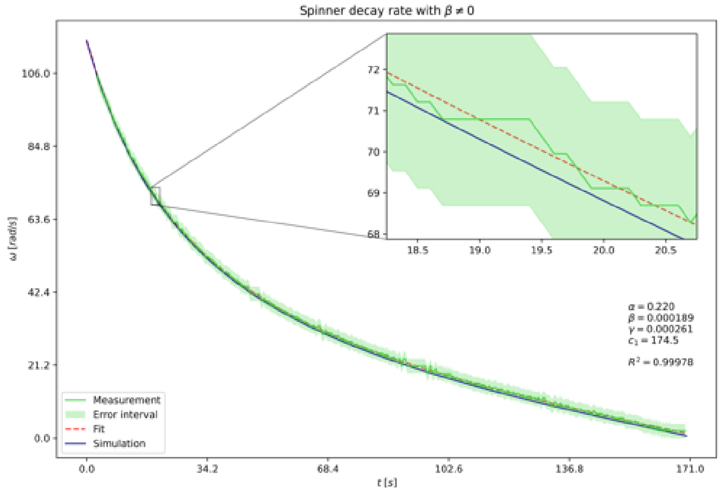
4.1.4 Výsledky a porovnání se simulací

Po analýze všech naměřených dat výše uvedeným algorimem (celkem 7 měření, protože máme 7 různých spinnerů) a provedením fitu funkcí dle rovnice 17 získáváme koeficienty v následucících rozmezích:

$$\begin{aligned} 0.220 &\leq \alpha \leq 0.927 \\ 0.0 \cdot 10^{-15} &\leq \beta \leq 2.5 \cdot 10^{-3} \\ 2.22 \cdot 10^{-4} &\leq \gamma \leq 3.17 \cdot 10^{-4} \end{aligned} \tag{27}$$

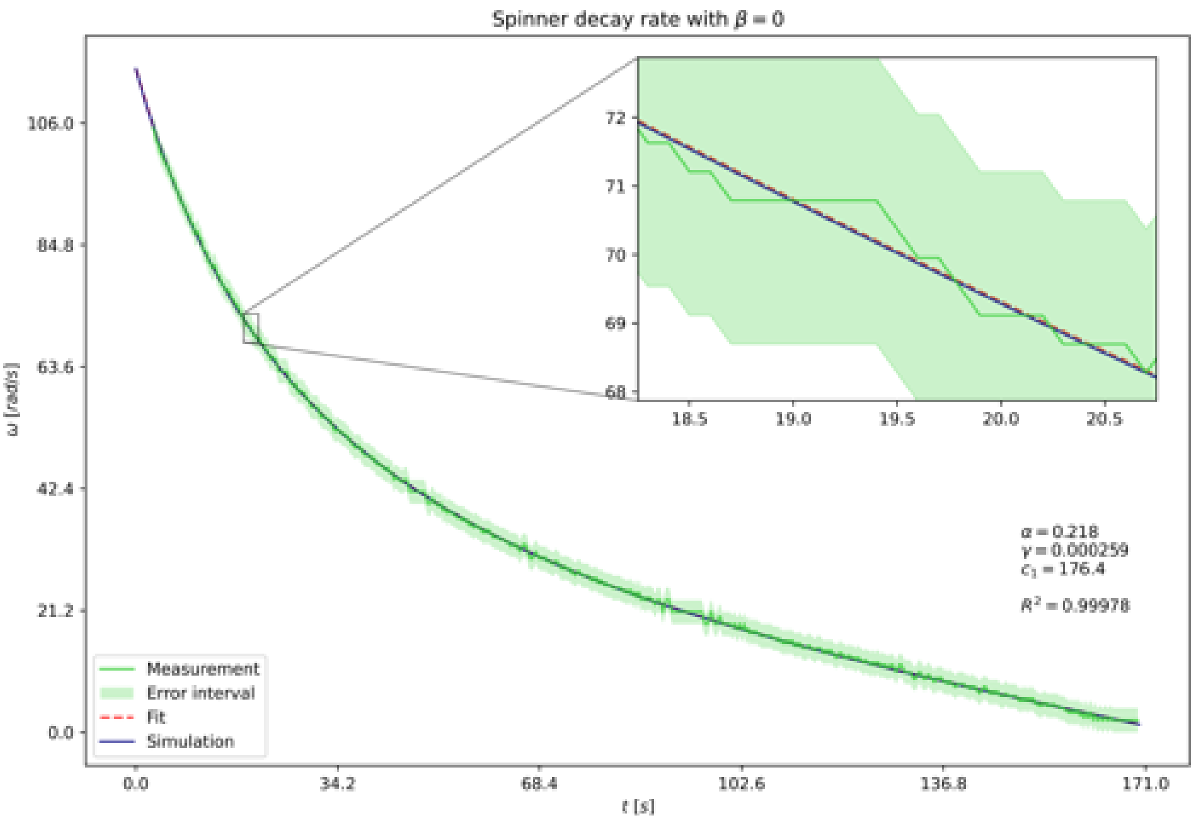
4.1.5 Zhodnocení použití lineárního koeficientu

Nyní je na řadě prozkoumání významu a užitečnosti lineárního brzdného koeficientu. Získané výsledky pro β z předchozích sedmi měření se pohybují velmi nahodile a mezi měřeními zaznamenáváme řádové rozdíly. Toto je prvním indikátorem toho, že lineární složka nebude součástí dobrého popisu tření (alespoň pro tento případ).



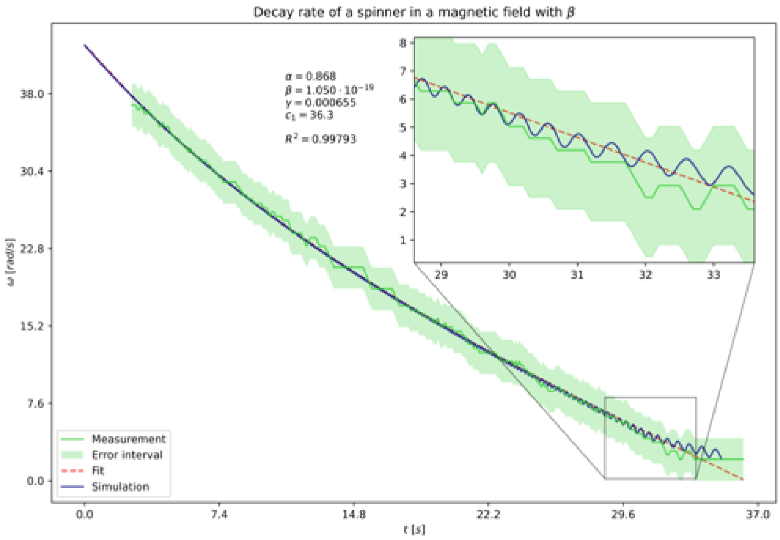
Obr. 13: Příklad grafu měřeného průběhu úhlové rychlosti v čase (zeleně) v porovnání s fitem užívajícím všech tří brzdných složek (červeně). Simulace modře.

Fitujeme-li stejná data funkcí bez lineární složky získáváme graf 14. Vidíme, že simulace, jejíž implementací se budeme zabývat v Kap. 5, v tomto případě viditelně lépe kopíruje naměřené hodnoty.



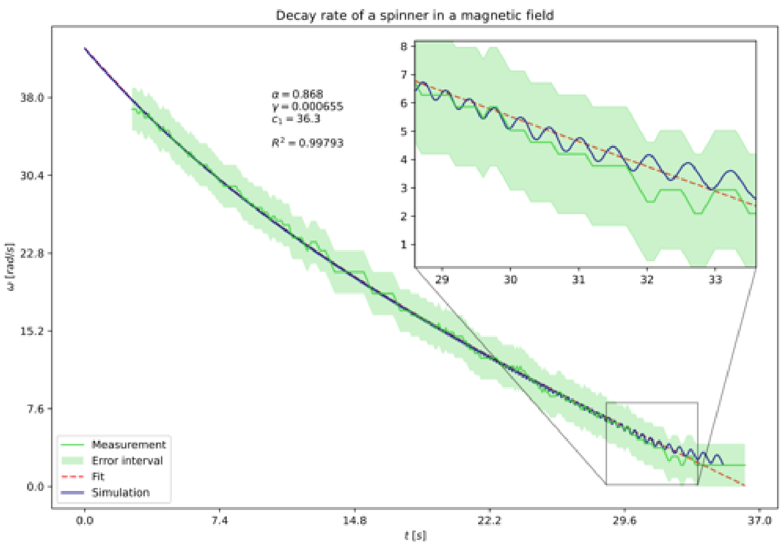
Obr. 14: Příklad grafu měřeného průběhu úhlové rychlosti v čase (zeleně) v porovnání s fitem užívajícím pouze složek α a γ (červeně). Simulace modře.

Poslední měření, které jsme k účelům porovnání těchto dvou metod provedli, bylo sledování úhlové rychlosti, když jsme spinner umístili do blízkosti našeho velkého magnetu. Velký magnet jsme upevnili 8cm od středu spinneru a stejnou aparaturou jako v minulém experimentu jsme sledovali jeho chování. Zde jsme prováděli pouze jedno měření a fitovali jsme ho opět oběma způsoby, které porovnáváme níže:



Obr. 15: Graf měřeného úhlové rychlosti v čase v porovnání s fitem užívajícím všech tří složek. (barevné schéma jako v Obr. 13)

Jak můžeme vidět na Obr. 15, koeficient β nám vychází efektivně nulový a průběhy fitů, ani simulací se od sebe téměř neliší. Z těchto měření tedy ověřujeme naši předchozí představu, že lineární komponent tření není fyzikálně signifikantní.



Obr. 16: Graf měřeného úhlové rychlosti v čase v porovnání s fitem užívajícím složek α a γ. (barevné schéma jako v Obr. 14)

5 Simulace

Jak již bylo zmíněno v předchozí kapitole a jak vychází z názvu této práce, značnou částí našeho zkoumání bylo také počítačové simulování těchto systémů.

5.1 Popis způsobu simulování

Naším cílem bude vytvoření tzv. *"Discrete event simulation"* neboli *DES* [18]. Jedná se o druh simulační metody, která popisuje náš stav pomocí diskrétních stavů, které existují v daném čase. Z jednoho určitého stavu poté pomocí určitých pravidel popisujících náš systém simulační algoritmus provádí odhad toho, jak bude vypadat další stav po uplynutí Δt sekund, kde Δt je velmi malé. Toto nazveme nejdím simulačním krokem. Poté stačí opakovat simulační kroky dle libosti.

5.1.1 Runge-Kutta metody

Runge-Kutta metody, pojmenované po Carlu Davidu Tolmé Rungeovi a Martinu Wilhelmu Kuttovi, jsou rodinou metod používaných k řešení implicitních a explicitních diferenciálních rovnic. Implicitní metody popisující budoucí stav systému z předchozího stavu, stejně jak budeme dělat my v naší *DES*. Runge-Kutta metody (také ornačované pouze RK metody) jsou tedy ideálním nástrojem pro simulování našeho problému. nejjednodušším příkladem RK metody je například Eulerova metoda. Těchto metod je nespočet, ale jejich obecný předpis je následující [19]:

$$
\begin{aligned}
y_{n+1} &= y_n + \Delta t \sum_{i=1}^{s} b_i k_i \\
k_m &= f\left(t_n + c_m \Delta t, y_n + \left(\sum_{j=1}^{s-1} k_j a_{m,j}\right)\Delta t\right)
\end{aligned}
\tag{28}
$$

V tomto předpisu jednotlivé znamenají: t_n je čas stavu y_n a Δt je krátký časový krok; s je stupeň RK metody; $a_{m,j}$, b_i a c_m jsou specifické koeficinty dané RK metody; y_n je stávající stav a y_{n+1} je budoucí, předpovídaný, stav; k_m je pomocný sub-stav; $f(t, y) = \dot{y}$.

Specifick0 koeficienty se nejčastěji zapisují pomocí tzv. Butcherových tabulek [20] (pojmenovaných po Johnu Charlesovi Butcherovi). Zápis obecné Butcherovy tabulky pro RK metodu s. stupně je:

Tab. 3: Zápis obecné Butcherovy tabulky pro RK metodu s. stupně

$$\begin{array}{c|ccccc}
c_1 = 0 & & & & & \\
c_2 & a_{2,1} & & & & \\
c_3 & a_{3,1} & a_{3,2} & & & \\
\vdots & \vdots & & \ddots & & \\
c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

Některé z podmínek, které sice nezaručují konzistentnost a stabilitu metody, ale dobrými základními pravidly jsou [20]:

$$\begin{aligned}
&1) \qquad \sum_{i=1}^{s} b_i = 1 \\
&2) \qquad \sum_{j=1}^{i-1} a_{i,j} = c_i \quad (i \in \{2, ..., s\})
\end{aligned} \tag{29}$$

Níže vypíšeme Butcherovy tabulky několika základních metod, které byly později implementovány v simulaci [21]:

Tab. 4: **Eulerova metoda**

$$\begin{array}{c|c}
0 & \\
\hline
 & 1
\end{array}$$

Tab. 5: Metoda středního bodu

$$\begin{array}{c|cc}
0 & & \\
1/2 & 1/2 & \\
\hline
 & 0 & 1
\end{array}$$

Tab. 6: Heunova metoda

$$\begin{array}{c|cc}
0 & & \\
1 & 1 & \\
\hline
 & 1/2 & 1/2
\end{array}$$

Tab. 7: Ralstonova metoda

$$\begin{array}{c|cc}
0 & & \\
2/3 & 2/3 & \\
\hline
 & 1/4 & 3/4
\end{array}$$

Tab. 8: Obecná metoda 2. stupně

$$\begin{array}{c|cc}
0 & & \\
\alpha & \alpha & \\
\hline
 & 1 - \frac{1}{2\alpha} & \frac{1}{2\alpha}
\end{array}$$

Tab. 9: Simpsonovo pravidlo 3/8

0				
1/3	1/3			
2/3	-1/3	1		
1	1	-1	1	
	1/8	3/8	3/8	1/8

Tab. 10: **Standardní Runge-Kutta metoda 4. stupně (neboli RK4)**

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	2/6	2/6	1/6

Tab. 11: Ralstonova metoda 4. stupně

0				
0.4	0.4			
0.45573725	0.29697761	0.15875964		
1	0.21810040	-3.05096516	3.83286476	
	0.17476028	-0.55148066	1.20553560	0.17118478

5.2 Popisující interakce

Řídícími interakcemi našeho simulovaného modelu budou opět popsány rovnicemi 4 a 5. V simulaci se však zaměříme na interakce celých spinnerů, ne pouze 2 dipólů. Pro popsání spinneru musíme určit několik vlastností týkajících se:

1. **Konfigurace**: pozice, velikosti, počet ramen (viz Kap. 2.1.1)
2. **Pohybu**: úhlová rychlost, moment setrvačnosti (viz Kap. 3.2), třecí koeficienty (viz Kap. 4)
3. **Magnetů**: velikost momentů, remanence (viz Kap. 3.3), orientace (viz Obr. 3)

Naším cílem je popisovat rotaci jednotlivých spinnerů v čase. Toto provedeme určením celkového momentu sílu jakožto součtu silových a magnetických složek přes všechny externí magnetické momenty. Popisující rovnice spinneru jsou tedy [14]:

$$\begin{gathered} I\dot{\omega} = \sum_{external} (\tau_F + \tau_{mag}) \\ \dot{\varphi} = \omega \end{gathered} \tag{30}$$

Ve výše uvedených rovnicí τ_F popisuje moment síly pocházející ze silové interakce, kterým na spinner (neboli na všechny jeho magnety) působí nějaký externí magnetický moment $m_{external}$ umístěný v prostoru na pozici $P_{external}$. Moment síly určíme pomocí ramene r, přes které síla F na spinner působí: $\vec{\tau} = \vec{F} \times \vec{r}$.

[14]První rovnice vychází z definice momentu síly: $\dot{L} = I\dot{\omega} = \tau$

τ_{mag} popisuje moment síly pocházející z toho, že na každý magnet spinneru působí externí magnetické pole $B(r, m_{external})$ tvořené externím magnetickým momentem $m_{external}$ umístěný v prostoru na pozici $P_{external}$. Pomocí momentu magentu umístěného na spinneru jednoduše dopočítáme moment síly podle rovnice 5: $\tau = m \times B(r, m_{external})$.

Předpisy τ_F a τ_{mag} jsou:

$$\begin{aligned}\tau_F &= \sum_{j=0}^{n} F_m(P_{external} - P(j), m(j), m_{external}) \times (P(j) - S) \\ \tau_{mag} &= \sum_{j=0}^{n} m(j) \times B(P(j) - P_{external}, m_{external})\end{aligned} \tag{31}$$

5.3 Implementace

K implementaci naší simulace použijeme sotwarový systém zvaný `Node.js`, který používá programovací jazyk `JavaScript (JS)` [22]. K tomuto rozhodnutí jsme došli, jelikož `Node.js` je rychlejší než `Python`, ale stále nabízí větší množství programátorsky nápomocných prvků. Pro nás to bude ideální kompromis mezi jednoduchostí implementace a rychlostí. V průběhu implementace také používáme datového formátu `JSON` [23] a nativní knihovnu `Node.js` zvanou `fs (File System)`, pomocí které budeme naše výsledky exportovat do `.csv` souborů pro další zpracování. Použití jazyka `JavaScript` také nabízí jednoduché znovupoužití již napsaného kódu k vytvoření webového rozhraní s vizualizací simulovaného systému. Toto rozhraní je skvělé k odstraňování chyb v průběhu vývoje a zároveň zjednodušuje interpretaci generovaných výsledků.

5.3.1 Použité datové struktury

3D vektory

První datovou strukturou, kterou budeme používat v našich výpočtech jsou 3D vektory. Jejich implementace společně s užitečnými funkcemi (sčítání, odčítání, vektorov a skalární násobení, apod.) je triviální.

3D vektory v našem kódu budou instancí třídy pojmenované `v3` a jejich jediné vlastnosti jsou jejich x, y a z komponenty vyjádřené jako 64-bitová čísla podle standardu `IEEE 754` [25].

Spinnery

Další strukturou jsou jednotlivé spinnery, které budeme označovat symbolem $\mathbb{S}$ a případně odpovídajícím číselným indexem, či jiným identifikátorem v pravém dolním indexu. Je-li z kontextu zřejmé, o jaký spinner se jedná, můžeme k jeho vlastnosnostem referovat čistě podle znaků určených v nomenklatuře (viz Kap. 1.1). Je-li potřeba oddělovat jednotlivé spinnery, budeme k jejich vlastnostem referovat exaktně a to skrze horní index; například, chtěli bychom-li určit počet ramen (značeno písmenem n) nějakého externího spinneru, použili bychom $\mathbb{S}^n_{external}$. Pokud bychom chtěli získat pozici 1. magnetu 3. spinneru, použili bychom $\mathbb{S}^P_3(1)$, jelikož $P(i)$ je funkce. Pomocí tohoto způsobu zápisu také v průběhu následujících kapitol konkretizujeme rovnice 30 a 31.

V kódu jsou spinnery instancí třídy `spinner`, jejíž vlastnosti jsou vypsány v tabulce 12.

Tab. 12: Výpis vlastností třídy `spinner`:

Symbol	Datový typ	Iniciální hodnota
r [15]	`float64`	0.0359
S [16]	`v3`	žádná
I [17]	`float64`	$4.80 \cdot 10^{-5}$
φ_0 [18]	`float64`	žádná
φ [19]	`float64`	žádná
ω_0 [20]	`float64`	žádná
ω [21]	`float64`	žádná
α [22]	`float64`	0.868
β [23]	`float64`	0.0
γ [24]	`float64`	0.00068
n [25]	`int`	3
m_0 [26]	`float64`	$\frac{1.1049 \cdot 0.005^3}{4\pi \cdot 10^{-7}} = 0.1099$
`magnet_orientation` [27]	`"vertical" / "radial" / "tangent"`	`"vertical"`
`constant_`ω [28]	`bool`	false
$m(i)$ [29]	`funkce (int` $\rightarrow$ `v3)`	nelze měnit
$P(i)$ [30]	`funkce (int` $\rightarrow$ `v3)`	nelze měnit
`copy ()` [31]	`funkce (`$\varnothing$ $\rightarrow$ `spinner)`	nelze měnit

[15] Poloměr (viz Obr. 4)
[16] Střed
[17] Moment setrvačnosti (viz Kap. 3.2)
[18] Počáteční úhel
[19] Okamžitý úhel
[20] Počáteční úhlová rychlost
[21] Okamžitý úhlová rychlost
[22] Konstantní koeficient tření (viz Kap. 4)
[23] Lineární koeficient tření (viz Kap. 4)
[24] Kvadratický koeficient tření (viz Kap. 4)
[25] Počet ramen/magnetů
[26] Velikost magnetického momentu (viz Rov. 3)
[27] Orientace magnetů (viz Obr. 3)
[28] Určuje, zda má úhlová rychlost spinneru zůstat konstantní (simuluje hnaný spinner)
[29] Funkce vracející orientaci magnetu podle jeho indexu
[30] Funkce vracející pozici magnetu podle jeho indexu
[31] Funkce vracející novou instanci spinneru se stejným stavem

Ω-stavy

Další důležitou datovou strukturou je něco, co nazveme Ω-stav. Tato datová struktura má za úkol uložit celkový stav systému (tedy stavy všech spinnerů - řekněme, že jejich počet bude p) a provádět na nich operace potřebné k řešení diferenciálních rovnic pomocí RK metod. Druhým úkolem Ω-stavů je výpočet všech momentů sil pro všechny spinnery - toto v popisu RK metody (viz Rov. 28) představuje naši derivaci, neboli funkci f.

Hlavní vlastností Ω-stavů je tedy pole spinnerů označované $\mathbb{P} = \{\mathbb{S}_1, \mathbb{S}_2, ., \mathbb{S}_p\}$ o velikoti $|\Omega^{\mathbb{P}}| = \Omega^p$. Na obecný k. prvek tohoto pole se budeme odkazovat následovně: $\Omega^{\mathbb{P}}(k) = \mathbb{S}_k$.

Jelikož úkolem Ω-stavu je i určení všech momentů sil, bude mít i vlastnost značenou A, což je pole dopočítaných úhlových zrychlení jednotlivých spinnerů. Pole A tedy ukládá všechna vypočtená úhlová zrychlení všech spinnerů, která získáme z rovnice 30. Jednotlivá zrychlení označíme symbolem α a tím pádem $A = \{\alpha_1, \alpha_2, ., \alpha_p\}$. Na jednotlivé prvky se opět odkazujeme takto: $\Omega^A(k) = \alpha_k = \frac{\mathbb{S}_k^\tau}{\mathbb{S}_k^I}$.

Nyní je na řadě provést formalizaci řídících rovnic 30 a 31 pomocí tohoto způsobu zápisu. Začneme rovnicemi pro výpočet silové a magnetické interakce magnetů, které nyní popíšeme jakožto funkce tří parametrů. Prvním parametrem v pořadí bude spinner $\mathbb{S}$, pro který budeme počítat momenty sil, kterými na něj působí externí magnetický moment $m_{external}$ nacházející se na pozici $P_{external}$:

$$\overbrace{\tau_F(\mathbb{S}, m_{external}, P_{external})}^{\substack{\text{Moment síly z} \\ \text{silové interakce}}} = \underbrace{\sum_{j=1}^{\mathbb{S}^n}}_{\text{Iterace přes všechny magnety spinneru}} \overbrace{F_m(P_{external} - \mathbb{S}^P(j), \mathbb{S}^m(j), m_{external})}^{\substack{\text{Síla mezi } j. \text{ magnetem spinneru} \\ \text{a externím magnetickým momentem}}} \times \underbrace{(\mathbb{S}^P(j) - \mathbb{S}^S)}_{\text{Rameno síly}}$$

$$\overbrace{\tau_{mag}(\mathbb{S}, m_{external}, P_{external})}^{\substack{\text{Moment síly z} \\ \text{magnetické interakce}}} = \underbrace{\sum_{j=1}^{\mathbb{S}^n}}_{\text{Iterace přes všechny magnety spinneru}} \mathbb{S}^m(j) \times \overbrace{B(\mathbb{S}^P(j) - P_{external}, m_{external})}^{\substack{\text{Magnetické pole tvořené externím} \\ \text{momentem na pozici } j. \text{ magnetu}}} \tag{32}$$

Nyní, když jsme formalizovali silové a magnetické interakce z jednoho externího magnetického dipólu, stačí určit celkový moment síly τ_{tot}, kterým na spinner působí všechny magnety dohromady. τ_{tot} tedy bude součet obou složek ze všech magnetů všech ostatních spinnerů. Z τ_{tot} konečně dopočítám úhlové zrychlení α_k obecného k. spinneru $\mathbb{S}_k$ takto:

$$\alpha_k = \frac{\tau_{tot}}{I} = \frac{1}{\mathbb{S}_k^I} \underbrace{\sum_{j=1;j\neq k}^{|\Omega^{\mathbb{P}}|}}_{\substack{\text{Iterace přes všechny} \\ \text{ostatní spinnery } \mathbb{S}_j}} \overbrace{\sum_{i=1}^{\mathbb{S}_j^n}}^{\substack{\text{Iterace přes všechny} \\ \text{magnety spinneru } \mathbb{S}_j}} \Big(\underbrace{\tau_F(\mathbb{S}_k, \mathbb{S}_j^m(i), \mathbb{S}_j^P(i))}_{\text{Silová složka}} + \underbrace{\tau_{mag}(\mathbb{S}_k, \mathbb{S}_j^m(i), \mathbb{S}_j^P(i))}_{\text{Magnetická složka}} \Big) \tag{33}$$

Pro další výpočty ještě zavedeme, jak se chová sčítání stavů a násobení Ω^A časem. Jednoduše řečeno dochází k sčítání hodnot pro každý index, nebo k násobení hodnoty na každém indexu daným časem[32]. Matematicky bychom tedy tyto operace formulovali takto[33]:

$$\begin{gathered} \text{Sčítání:} \quad \Omega_{sum} = \Omega_a + \Omega_b \\ \Updownarrow \\ \Omega_{sum}^{\mathbb{P}}(k)^{\omega} = \Omega_a^{\mathbb{P}}(k)^{\omega} + \Omega_b^{\mathbb{P}}(k)^{\omega} \quad \forall k \in 1, \ldots, \Omega^p \end{gathered} \tag{34}$$

$$\begin{gathered} \text{Násobení:} \quad \Omega_b = \Omega_a^A \cdot t \\ \Updownarrow \\ \Omega_b^{\mathbb{P}}(k)^{\omega} = \Omega_a^A(k) \cdot t \quad \forall k \in 1, \ldots, \Omega^p \end{gathered} \tag{35}$$

S tímto jsme konečně schopni definovat poslední vlastnost Ω-stavu, kterou je krátky časový úsek δ. Z výpočtu zrychlení je pak provedena predikce budoucího stavu pod uplynutí δ, který označíme $^*\Omega$. Jeho definicí je:

$$^*\Omega = \Omega + \Omega^A \cdot \Omega^{\delta} \tag{36}$$

V kódu jsou tyto struktury instancemi třídy `omega_state`.

[32] V rovnici 35 jsou jednotky: $[rad \cdot s^{-1}] = [rad \cdot s^{-2}] \cdot [s]$.

[33] Podmínkou sčítání je, že: $\Omega_{sum}^p = \Omega_a^p = \Omega_b^p$

Φ-stavy

Poslední datovou strukturou jsou tzv. Φ-stavy, které jsou velmi podobné Ω-stavům svou funkcí. Jejich účelem je opět umožnit použití RK metod. V průběhu jednoho simulačního kroku totiž dochází k dvojitému numerickému integrování - nejdříve, abychom získali z úhlového zrychlení úhlovou rychlost, a poté podruhé, abychom z úhlové rychlosti získali úhel ($\alpha_k \xRightarrow{RK} \mathbb{S}_k^\omega \xRightarrow{RK} \mathbb{S}_k^\varphi$).

Vlastnosti Φ-stavů jsou stejné jako u Ω-stavů, kromě pole A, které je nahrazeno polem O. Pole O, narozdíl od pole A, které ukládá úhlová zrychlení, obsahuje úhlové rychlosti každého spinneru. Pole O zapisujeme takto: $\Phi^O = \{\omega_1, \omega_2, \ldots, \omega_p\}$. Na libovoný k. člen se opět odkazujeme takto: $\Phi^O(k) = \omega_k$. Ω-stavy a Φ-stavy na sebe navazují následovně:

$$\Phi^O(k) = \Omega^{\mathbb{P}}(k)^\omega \quad \forall k \in 1, \ldots, \Phi^p \tag{37}$$

Poté je predikce dalšího Φ-stavu analogická predikci Ω-stavu:

$$^*\Phi = \Phi + \Phi^O \cdot \Phi^\delta \tag{38}$$

V kódu jsou tyto struktury instancemi třídy `phi_state`.

5.3.2 Popis iteračního kroku

Jak jsme již avizovali, pomocí Ω-stavů a Φ-stavů provedeme v každém iteračním kroku dvě numerické integrace: $\alpha_k \xRightarrow{RK} \mathbb{S}_k^\omega \xRightarrow{RK} \mathbb{S}_k^\varphi$.

Průběh výpočtu budoucího Ω-stavu (a analogicky Φ-stavu) bude probíhat následovně:

Máme začáteční stav Ω_1 a určíme $k_1 = \Omega_1^A$. Poté vytvoříme nový stav, jehož spinnery (resp. jejichž úhlové rychlosti) jsou zároveň částečně ovlivněny úhlovými zrychleními z k_1. To, jakou mírou jsou ovlivněny, udává $a_{2,1} \cdot \Delta t$. Toto odpovídá tomu, jako kdyby libovolný k. spinner $\mathbb{S}_k$ byl zrychlován z jeho původní úhlové rychlosti ω po dobu $a_{2,1} \cdot \Delta t$ úhlovým zrychlením $\Omega_1^A(k)$ na nějakou novou úhlovou rychlost ω'. Tento výpočet tedy můžeme zapsat následovně:

$$\begin{gathered} {\Omega'_1}^{\mathbb{P}}(k)^\omega = {\Omega_1}^{\mathbb{P}}(k)^\omega + k_1 \cdot a_{2,1} \cdot \Delta t \\ \Omega'_1 = \Omega_1 + k_1 \cdot a_{2,1} \cdot \Delta t \end{gathered} \tag{39}$$

Takto vytvořené spinnery necháme zrychlovat a zároveň rotovat po dobu ${\Omega'}_1^\delta = c_2 \cdot \Delta t$, čímž získáme:

$$^*\Omega'_1 = {\Omega_1}^{\mathbb{P}}(k)^\omega + {\Omega'_1}^A(k) \cdot {\Omega'}_1^\delta \tag{40}$$

Nakonec získáme k_2 jako úhlové zrychlení těchto zrychlených a pootočených spinnerů:

$$k_2 = {^*\Omega'_1}^A \tag{41}$$

Obecně můžeme tedy vyjádři k_n takto:

$$k_n = \left(\Omega_n + \left(\sum_{i=1}^{n}(k_i \cdot a_{n,i} \cdot \Delta t)\right) \cdot c_n \cdot \Delta t\right)^A \tag{42}$$

Po určení všech $k_1, \ldots, k_s$ konečně získáváme nový, odhadovaný stav:

$$\Omega_{n+1} = \Omega_n + \left(\sum_{i=1}^{s}(b_i k_i)\right) \cdot \Delta t \tag{43}$$

Identicky funguje postup i pro provádění numerického integrování Φ-stavů, kde pouze místo k_i budeme používat q_i pro přehlednost do budoucna.

5.3.3 Simulační instance

Simulační instance poté opakovaně provádí tyto dvě numerické integrace v každém kroku a získává nové stavy, s kterými pokračuje v dalším kroku. Každý simulační kontext je instancí třídy `sim_instance`, jejíž vstupními parametry jsou:

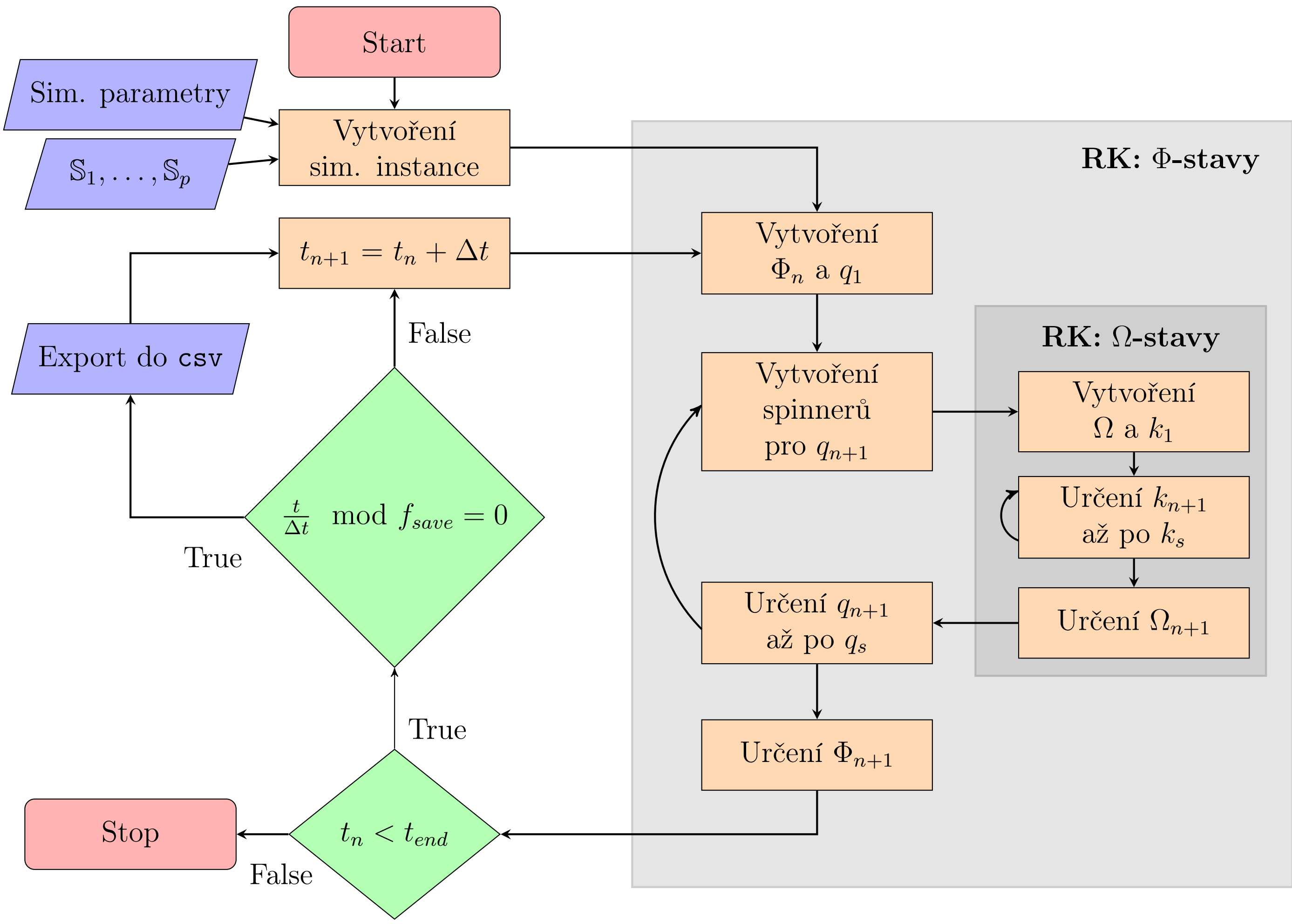
1. **Simulační parametry:** JSON objekt přepisující vybrané výchozí simulační parametry [34], jejiž hodnoty jsou v tabulce 13.
2. **Butcherova tabulka RK metody:** instance třídy `RK_matrix` (implementace je triviální podle pramene [20])
3. **Výchozí hodnoty spinnerů:** JSON objekt přepisující výchozí parametry, které se automaticky nastavují při vytvoření každého nového spinneru v tomto kontextu (viz Tab. 12).

[34] Pomocí tzv. objektové destrukturalizace, což je objektová operace v jazyce JavaScript (viz [26]).

Tab. 13: Výpis simulačních parametrů třídy `sim_instance`:

Symbol	**Datový typ**	**Iniciální hodnota**
Δt	`float64`	10^{-3}
`start_time` (t_{start})	`float64`	0
`end_time` (t_{end})	`float64`	1
`save_freq` (f_{save}) [35]	`get float64`	$\left\lceil \frac{10^{-3}}{\Delta t} + 1 \right\rceil - 1$
`out_path`	`string`	`out.csv`
`exports` [36]	`string[]`	["s[0].ω", "s[0].φ"]

Po vytvoření kontextu jsou do něj přidány jednotlivé spinnery (číslovány od 0) pomocí funkce `add_spinner`, jejíž povinné parametry jsou S, φ_0 a ω_0 (viz Tab. 12). Zbylé parametry jsou dobrovolné - jejich výchozí hodnoty závisí na kontextu (viz bod 3 Tab. 13). Spimulace je poté spuštěna funkcí `run`. Vývojový diagram celé simulace je znázorněn níže:



Obr. 17: Vývojový diagram simulace

[35] Určuje, kolikátý každý krok se uloží do výstupního souboru.
[36] Určuje, jaké vlastnosti kterých spinnerů se ukládají. Každá vlastnost má vlastní sloupec.

5.3.4 Časová komplexita

Pro dokončení simulace trvající čas T s časovým intervalem τ je potřeba udělat T/τ iteračních kroků. V každém kroku počítáme interakce každého magnetu s každým - máme-li tedy P spinnerů a každý z nich má N ramen, získáváme $(NP)^2$ interagujících dvojic. Zároveň provádíme více výpočtů, abychom zpřesnili výsledky, tudíž, použijeme-li RK metodu K. stupně a potřebujeme v každém kroku provést U numerických integrací (pro nás je $U = 2$), získáváme finální časovou komplexitu:

$$\mathcal{O}\left(\left(\frac{T}{\tau}\right) \cdot (NP)^2 \cdot K^U\right) \tag{44}$$

5.3.5 Webové rozhraní

Literatura

[1] YUNG, Kar W.; LANDECKER, Peter B. a VILLANI, Daniel D. An Analytic Solution for the Force Between Two Magnetic Dipoles. [Online]. *Magnetic and Electrical Separation.* 1998, roč. 9, č. 1, s. 39-52. ISSN 1055-6915. Dostupné z: `https://doi.org/10.1155/1998/79537`. [cit. 2023-12-16].

[2] CULLITY, B. D. a GRAHAM, C. D. *Introduction to magnetic materials.* Second edition. Hoboken: IEEE Press, [2009]. ISBN 978-0-471-47741-9.

[3] SERWAY, Raymond A. a JEWETT, John W. Jr. *Physics for scientists and engineers.* 6th ed. Belmont: Thomson-Brooks/Cole, 2004. ISBN 0-534-40842-7.

[4] YE, Jianhe; ZHAN, Pengfei; ZENG, Jincheng; KUANG, Honglin; DENG, Yongfang et al. Concise magnetic force model for Halbach-type magnet arrays and its application in permanent magnetic guideway optimization. [Online.] *Journal of Magnetism and Magnetic Materials.* 2023, roč. 587. ISSN 03048853. Dostupné z: `https://doi.org/10.1016/j.jmmm.2023.171301`. [cit. 2023-12-16].

[5] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *Standard Uncertainty and Relative Standard Uncertainty* [online]. [cit. 2023-12-16]. Dostupné z: `https://physics.nist.gov/cgi-bin/cuu/Info/Constants/definitions.html`

[6] Jasper. *Shorthand error notation (with brackets) accros decimal point [duplicate]* [online]. [cit. 2023-12-16]. Dostupné z: `https://physics.stackexchange.com/questions/445141/shorthand-error-notation-with-brackets-accros-decimal-point`

[7] *Turnaj mladých fyziků* [online]. [cit. 2023-12-16]. Dostupné z: `https://tmf.fzu.cz/tasks.php?y`

[8] *Neodymium Magnet Grades* [online]. [cit. 2023-12-16]. Dostupné z: `https://totalelement.com/blogs/about-neodymium-magnets/neodymium-rare-earth-magnet-grades`

[9] REICHL, Jaroslav a Martin VŠETIČKA. *Fyzické kyvadlo* [online]. [cit. 2023-12-16]. Dostupné z: `http://fyzika.jreichl.com/main.article/view/206-fyzicke-kyvadlo`

[10] VAN ROSSUM, Guido, Drake VAN ROSSUM a L. FRED. *Python 3 Reference Manual.* Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.

[11] VIRTANEN, Pauli. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods.* 2020(17), 261-272.

[12] HARRIS, Charles R. Array programming with NumPy. *Nature.* 2020(585), 357–362.

[13] HUNTER, John D. Matplotlib: A 2D graphics environment. *Computing in science & engineering.* 2007, 9(3), 90-95.

[14] VAN ROSSUM, Guido. *The Python Library Reference, release 3.8.2.* Python Software Foundation, 2020.

[15] STAACKS, S; HÜTZ, S; HEINKE, H a STAMPFER, C. Advanced tools for smartphone-based experiments: phyphox. Online. *Physics Education.* 2018, roč. 53, č. 4. ISSN 0031-9120. Dostupné z: `https://doi.org/10.1088/1361-6552/aac05e`. [cit. 2023-12-17].

[16] SHAFRANOVICH, Yakov. Common format and MIME type for comma-separated values (CSV) files. *RFC 4180, October.* 2005.

[17] ALLAIN, Rhett. Let's Explore the Physics of Rotational Motion With a Fidget Spinner. *WIRED* [online]. 2017, **2017**, 1-1 [cit. 2023-12-19]. Dostupné z: `https://www.wired.com/2017/05/physics-of-a-fidget-spinner/`

[18] STEWART, Robinson. *Simulation – The practice of model development and use.* 1. John Wiley, 2004. ISBN 978-0-470-84772-5.

[19] PRESS, William H. *Numerical recipes: the art of scientific computing.* 3rd ed. Cambridge: Cambridge University Press, 2007. ISBN 978-0-521-88068-8.

[20] ISERLES, A. *A first course in the numerical analysis of differential equations.* Cambridge: Cambridge University Press, 1996. ISBN 978-0-521-55655-2.

[21] PCC, Ben. WIKIMEDIA FOUNDATION. *List of Runge–Kutta methods* [online]. 2007, 2023-12-20 [cit. 2023-12-22]. Dostupné z: `https://en.wikipedia.org/wiki/RungeâĂŞKutta_methods`

[22] FLANAGAN. *JavaScript: the definitive guide.* O'Reilly Media, 2006.

[23] PEZOA, Felipe; REUTTER, Juan L.; SUAREZ, Fernando; UGARTE, Martín a VRGOČ, Domagoj. Foundations of JSON Schema. Online. In: *Proceedings of the 25th International Conference on World Wide Web.* Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2016, s. 263-273. ISBN 9781450341431. Dostupné z: `https://doi.org/10.1145/2872427.2883029`. [cit. 2023-12-22].

[24] PEREIRA, Caio Ribeiro a PEREIRA, Caio Ribeiro. Introduction to Node.js. Online. In: *Building APIs with Node.js*. Berkeley, CA: Apress, 2016, s. 1-3. ISBN 978-1-4842-2441-0. Dostupné z: https://doi.org/10.1007/978-1-4842-2442-7_1. [cit. 2023-12-22].

[25] GANSSLE, Jack. IEEE 754 Floating Point Numbers. Online. In: *The Firmware Handbook*. Elsevier, 2004, s. 203-205. ISBN 9780750676069. Dostupné z: https://doi.org/10.1016/B978-075067606-9/50019-9. [cit. 2023-12-22].

[26] MOZILLA. Mozilla Developer Network. *Destructuring assignment* [online]. [cit. 2023-12-27]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment

Seznam obrázků

Seznam tabulek

Seznam ústřižků

SEZNAM ÚSTŘIŽKŮ

Přílohy