Problém tří těles

Anna Marie Kyšová, Matouš Volf

Vedoucí projektu: Jakub Dvořák

Soustředění mladých fyziků a matematiků

Plasnice 2024

Obsah

Anotace

V našem projektu jsme se věnovali Problému tří těles a jeho počítačové simulaci v programovacím jazyce Python. Naše implementace dovoluje umístit tělesa na pozice do roviny a nastavit jejich hmotnost a počáteční rychlost. Neomezili jsme se pouze na tři tělesa, a je tak možné nechat simulovat pohyb libovolného počtu těles.

Práce popisuje fyzikální principy, převod rovnic Newtonových zákonů do kódu a jejich využití v simulačních výpočtech.

Klíčová slova

Problém tří těles, Newtonovy zákony, gravitace, planety, simulace.

Poděkování

Děkujeme našemu vedoucímu projektu Jakubu Dvořákovi, který nám vždy ochotně poskytl pomoc a rady během vypracování projektu. Také děkujeme matematicko-fyzikální fakultě Univerzity Karlovy za organizaci Soustředění mladých matematiků a fyziků, díky kterému jsme měli možnost tento projekt uskutečnit.

Úvod

Problém tří těles je problém z oblasti nebeské mechaniky a dynamiky, který se zabývá pohybem tří těles, která se navzájem přitahují gravitační silou. Jedná se o rozšíření Problému dvou těles (neboli Keplerovy úlohy), který vyřešil Isaac Newton (1). Třetí těleso v tomto problému však vnáší chaos a je výrazně složitější předpověď vypočítat.

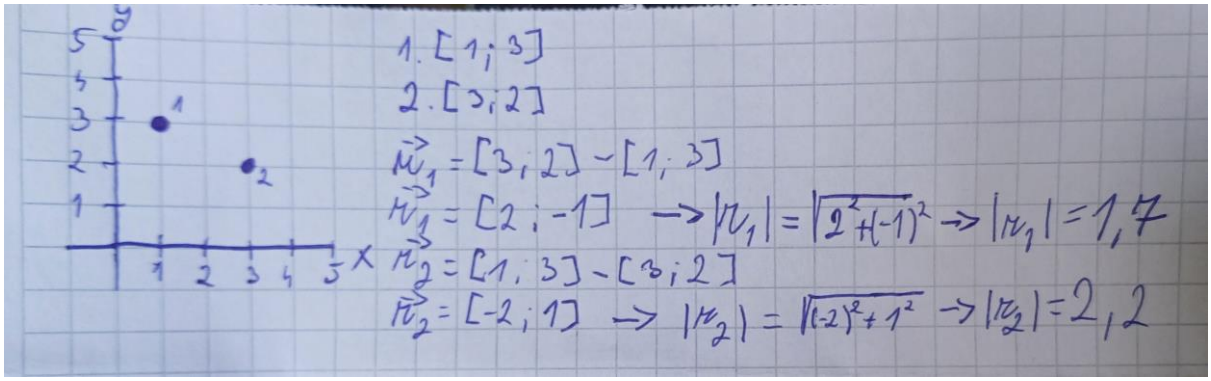
Výpočty

Problém tří těles je analyticky neřešitelný. V momentě, kdy všechny tři tělesa se začnou pohybovat, vzniká chaos. Ten je sice deterministický, učiněné předpovědi se nicméně přesto od chování systému brzy odchýlí (2). Poprvé byl Problém tří těles popsán na vztahu Měsíce, Země a Slunce v Newtonových Principiích (3).

Pro naši simulaci Problému tří těles jsme se rozhodli využít rovnice:

$$F^{\rightarrow} = G\,\frac{m1 \cdot m2 \cdot r^{\rightarrow}}{|\,r\,|^3}$$

$F^{\rightarrow}$ je vektor síly. G je gravitační konstanta rovna $6{,}67 \cdot 10^{-11}\, m^3 \cdot kg^{-1} \cdot s^{-2}$. Hodnoty $m1$ a $m2$ reprezentují hmotnost dvou těles, která jsou v daný okamžik porovnávána. Norma vektoru $r^{\rightarrow}$ je vzdálenost mezi dvěma tělesy, která se dá dopočítat přes Pythagorovu větu, do které jsme dosadili počáteční souřadnice daných těles. Jednotkový vektor $|r|$ jsme dopočítávali pomocí rovnice $|r| = \sqrt{x^2 + y^2}$, do které jsme dosadili souřadnice z vektoru $r^{\rightarrow}$.

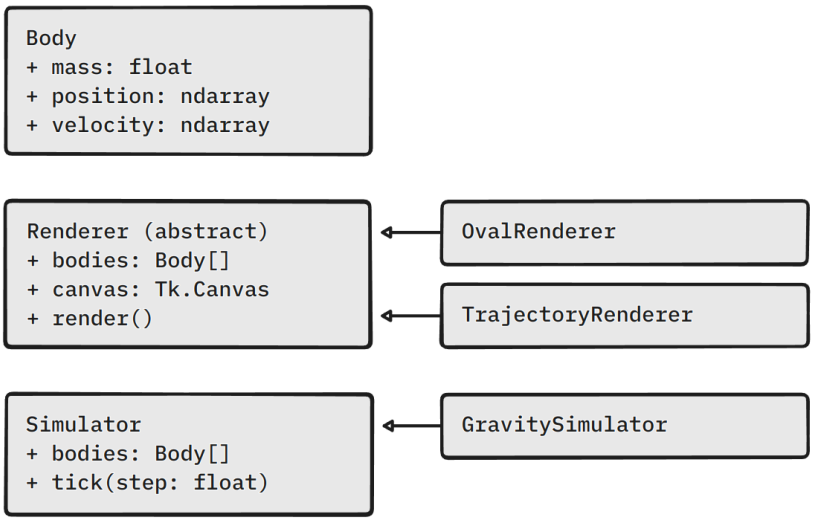


Obrázek 1 – popis výpočtu $r^{\rightarrow}$ a $|\,r\,|$

Následně bylo potřeba vypočítat i jednotlivou akceleraci samotných těles. Pro to jsme využili vzorce $a = \frac{F}{m}$, do kterého jsme dosadili výsledek první rovnice. Posledním krokem bylo vypočítat rychlost z tohoto zrychlení a příslušného časového kroku: $v = a \cdot \Delta t$.

Implementace do programu

Program je napsán v jazyce Python a je rozdělen do tří částí: simulace, vykreslování a grafické rozhraní. První dvě jsou součástí separátního modulu `body_problem`:



Obrázek 2 – diagram modulu `body_problem`

Simulace

Je definována obecná abstraktní třída Simulator s metodou `tick`. Ta má za úkol provést jeden krok simulace. Z abstraktní třídy dědí `GravitySimulator`, který metodu implementuje simulaci přitažlivých sil a pohybů těles. Ta se nevykonává spojitě, nýbrž po malých časových krocích (řádově po jednotkách milisekund). V každém kroku se dle výše uvedených rovnic vypočítají přitažlivé síly mezi tělesy, z nich zrychlení a rychlost. Podle té se tělesům nakonec změní souřadnice. K reprezentaci vektorů je využita knihovna NumPy. Ukázka podstatného kódu:

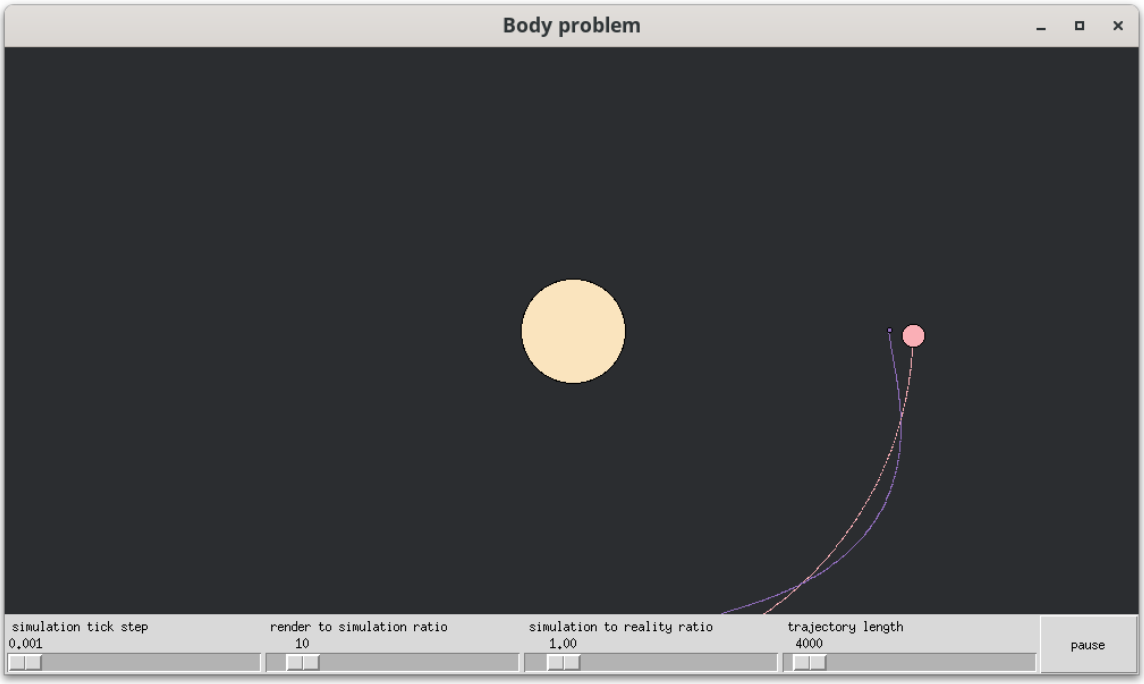
```
def tick(self, step: float = 0.1):
  for body1 in self.bodies:
    force = sum(map(
      lambda body2:
      gravitational_constant * body1.mass * body2.mass
      * (body2.position - body1.position)
      / max(norm(body2.position - body1.position) ** 3, self.DISTANCE_MIN),
      filter(lambda b: b != body1, self.bodies))
    )
    acceleration = force / body1.mass
    body1.velocity = body1.velocity + step * acceleration

  for body in self.bodies:
    body.position = body.position + body.velocity * step
```

Vykreslování

Podobně jako k simulaci je vytvořena i abstraktní třída `Renderer`, z níž dědí `OvalRenderer` a `TrajectoryRenderer`. Jejich účel je vykreslit příslušné části vizuálu – kruhy a trajektorie těles – na plátno.

Grafické rozhraní

Okno je vykresleno knihovnou Tkinter. Obsahuje plátno (objekt typu Canvas) k vykreslení roviny s tělesy a posuvníky pro změnu různých simulačních parametrů.



Obrázek 3 - grafické rozhraní programu

Závěr

Povedlo se nám připravit výpočty a naprogramovat simulaci Problému tří těles. V rámci práce vznikl program v programovacím jazyce Python, ve kterém uživatel může měnit parametry simulace (například rychlost nebo časový krok). Zároveň je možné nechat simulovat nejen tři, nýbrž libovolný počet těles. V rámci dalšího vývoje se nabízí zvýšit přesnost simulace, vypočítávání dalších veličin (např. pohybové a polohové energie) nebo možnost simulovat tělesa ve 3D prostoru.

Zdroje

1. *Problém dvou těles*. Online. 2020. Dostupné z: https://www.walter-fendt.de/html5/phcz/gravity_cz.htm. [cit. 2024-07-10].
2. REICHL, Jaroslav. *Problém tří těles*. Online. Encyklopedie fyziky. 2006. Dostupné z: http://fyzika.jreichl.com/main.article/view/1255-problem-tri-teles. [cit. 2024-07-10].
3. GREGERSEN, Erik. *Principia*. Online. Britannica. Dostupné z: https://www.britannica.com/topic/Principia. [cit. 2024-07-10].
4. VIKTORIN, Petr a HRONČOK, Miro. *NumPy*. Online. Nauč se Python. 2017. Dostupné z: https://naucse.python.cz/lessons/intro/numpy/. [cit. 2024-07-10].