UNIVERZA V LJUBLJANI | FMF Fakulteta za matematiko in fiziko

3. naloga: Simplektična integracija

Poročilo pri predmetu Višje računske metode

2024/25

Avtor:
Rok Mlinar Vahtar
28242004

14. marec 2025

1 Uvod

Tema te naloge je simplektična integracija in širše gledano Trotter-Suzukijev razcep, ki nam jo omogoči. Ta deluje zato, ker enačbe gibanja, ki sledijo iz $H = \frac{\mathbf{p^2}}{2} + V(x)$, lahko prepišemo na unitarno evolucijo z propagatorjem $U(t) = \exp\left(\frac{t}{n}\{\cdot, T\} + \frac{t}{n}\{\cdot, V\}\right)^n$. Tega je ponavadi težko izračunati točno, znamo pa izračunati člene oblike $\exp\left(\frac{t}{n}\{\cdot, T\}\right)$ in $\exp\left(\frac{t}{n}\{\cdot, V\}\right)$. Če čas diskretiziramo na n točk tako, da velja $t = n\tau$, njihovo delovanje lahko zapišemo kot

$$\exp\left(\tau\{\cdot, \mathrm{T}\}\right)(\mathbf{q}, \mathbf{p}) = \left(\mathbf{q} + \nabla_{\mathbf{p}}\mathrm{T}(\mathbf{p})\tau, \mathbf{p}\right),$$

$$\exp\left(\tau\{\cdot, \mathrm{V}\}\right)(\mathbf{q}, \mathbf{p}) = \left(\mathbf{q}, \mathbf{p} - \nabla_{\mathbf{p}}\mathrm{V}(\mathbf{q})\tau\right).$$

Da dobimo iz teh členov dejanski $U(t)$ pa moramo uporabiti enega izmed Trotter-Suzukijevih razcepov. V tej nalogi bomo uporabili razcepe z realnimi koeficienti

$$e^{zA+zB} = e^{zA}e^{zB} + \mathcal{O}(z^2) \qquad k = 1, p = 1$$

$$e^{zA+zB} = e^{\frac{1}{1}zA}e^{zB}e^{\frac{1}{1}zA} + \mathcal{O}(z^3) = S_2(z) + \mathcal{O}(z^3) \qquad k = 2, p = 2$$

$$e^{zA+zB} = S_2(x_1 z)S_2(x_0 z)S_2(x_1 z) + \mathcal{O}(z^5), x_1 = \frac{1}{2 - 2^{1/3}}, x_0 = -2^{1/3}x_1 \qquad k = 4, p = 4$$

in z kompleksnimi koeficienti

$$e^{zA+zB} = e^{zp_1 A}e^{zp_2 B}e^{zp_3 A}e^{zp_4 B}e^{zp_5 A} + \mathcal{O}(z^4) \qquad k = 3, p = 3$$

$$e^{zA+zB} = e^{zp_1 A}e^{zp_2 B}e^{zp_3 A}e^{zp_4 B}e^{2zp_5 A}e^{zp_4 B}e^{zp_3 A}e^{zp_2 B}e^{zp_1 A} + \mathcal{O}(z^5) \qquad k = 5, p = 4$$

$$p_1 = p_5^* = \frac{1}{4}\left(1 + \frac{i}{\sqrt{3}}\right), p_2 = p_4^* = 2p_1, p_3 = \frac{1}{2}$$

kjer z k označimo število členov v razcepu za vsak operator in z p red, do vključno katerega je razcep točen. V nadalnjem bom razcepe imenoval v formatu razcepkp, na primer razcep22.

2 Klasični 2D anharmonski oscilator

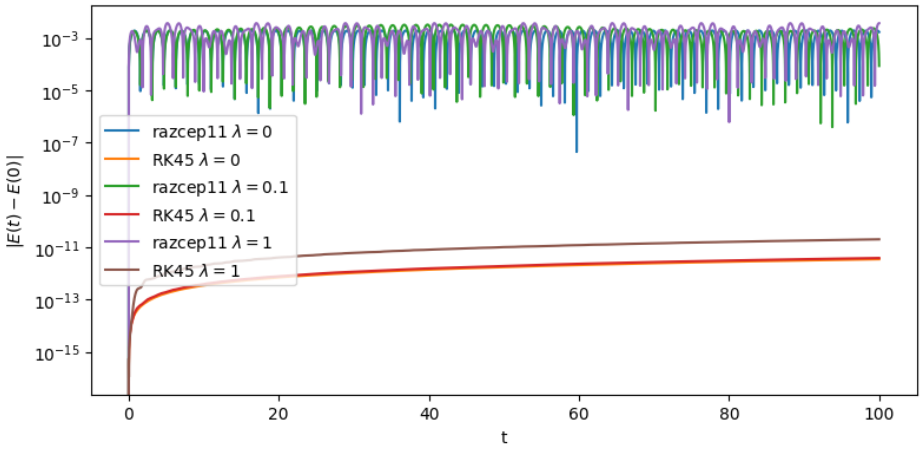
Tokrat nas zanimajo klasične rešitve anharmonskega oscilatorija,

$$H(q_1, q_2, p_1, p_2) = \frac{1}{2}(p_1^2 + p_2^2 + q_1^2 + q_2^2) + \lambda q_1^2 q_2^2$$

torej dejanske tirnice, ki predstavljajo pot delca namesto valovnih funkcij iz prejšnjih nalog.

2.1 Zakaj simplektična integracija

Prvo vprašanje, ki si ga bomo zastavili, je zakaj sploh kompliciramo z simplektičnimi metodami, ko bi lahko veliko enostavneje vzeli kar neko splošno Runge-Kutta metodo, recimo RK45, kjer bi imeli celo možnost spremenljivega koraka. Če izberemo začetni pogoj $q_1(0) = 1/2$, $q_2(0) = 0$, $p_1(0) = 0$, $p_2(0) = 1$, lahko vidimo, kako se različna tipa metod odneseta pri ohranitvi energije.

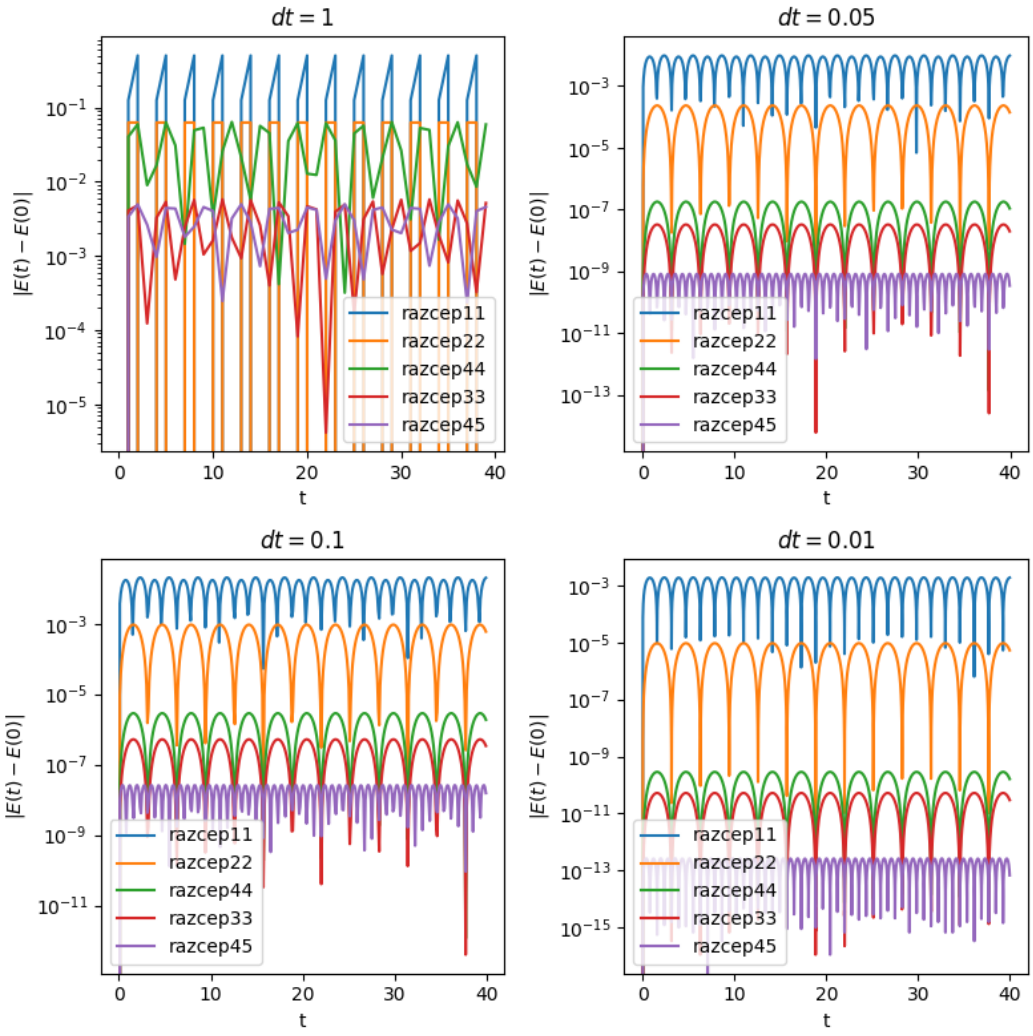


Slika 1: Ohranitev energije pri različnih metodah, λ in $dt = 0.01$

Kot vidimo, je napaka simplektične metode zaradi fiksnega koraka na začetku večja, ampak je v povprečju konstantna in na prvi pogled neodvisna od izbire λ. Tu vidimo glavni razlog za uporabo simplektičnih metod, ker so narejene specifično za hamiltonske sisteme ohranjajo volumen v faznem prostoru in energijo. To pomeni, da če simuliramo obnašanje sistema za daljše čase, metoda Runge-Kutta podivja medtem ko je simplektična metoda veliko bolj stabilna.

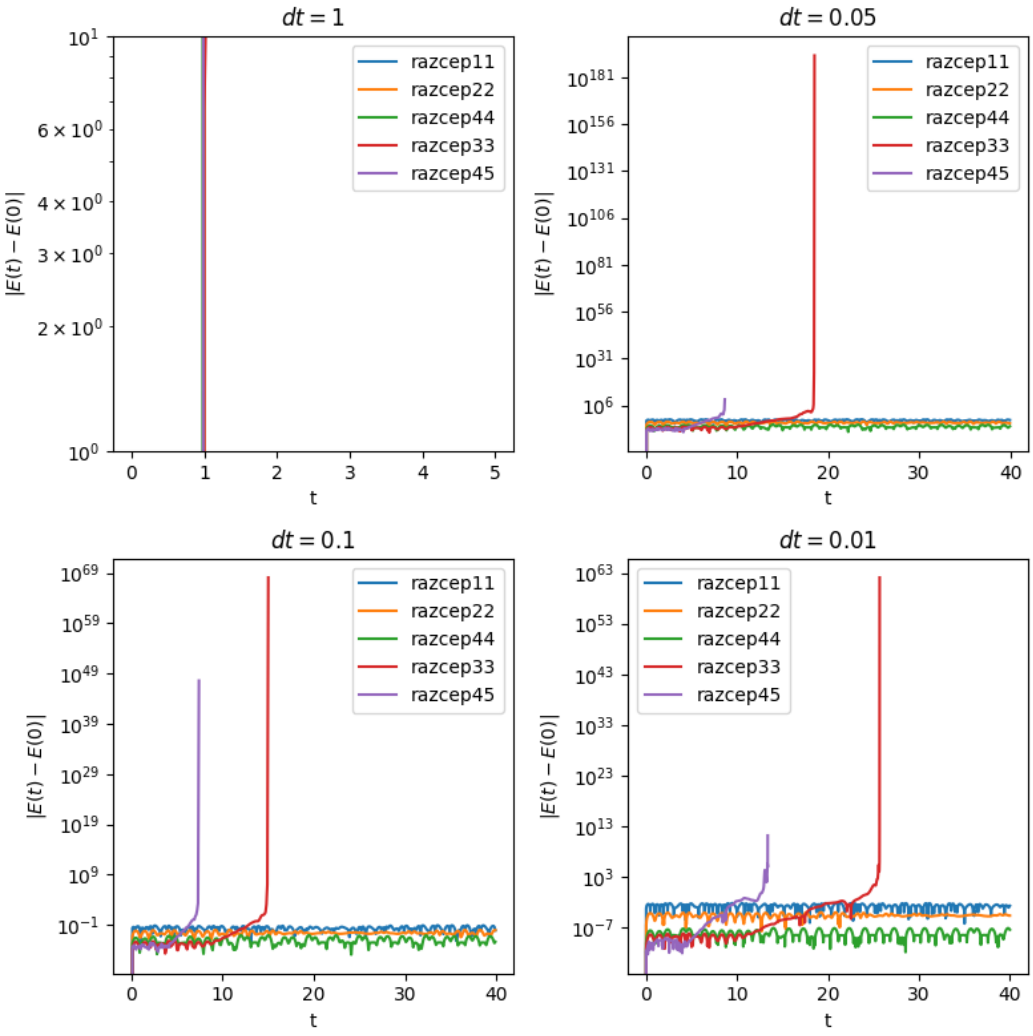
2.2 Ohranitev energije

Zdaj, ko vemo, zakaj imamo opravka z simplektičnimi metodami, pa si jih poglejmo bolj podrobno. Na prejšnjem grafu smo namreč bolj slabo videli podrobnosti simplektičnih metod.



Slika 2: Ohranitev energije pri simplektičnih metodah, $\lambda = 0$

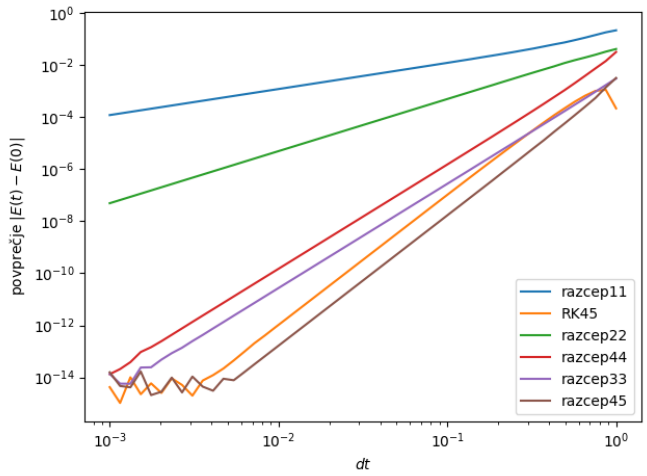
Vidimo, da pri $\lambda = 0$ metode že pri relativno velikem časovnem koraku delujejo kar dobro in presenetljivo oba kompleksna razcepa 33 in 45 premagata celo razcep 44.



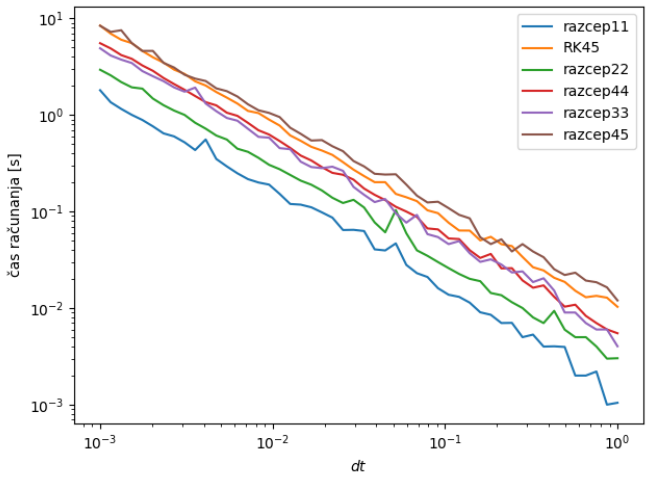
Slika 3: Ohranitev energije pri simplektičnih metodah, $\lambda = 10$

Če pogledamo isti graf še pri $\lambda = 10$, pa vidimo, da uporaba kompleksnih koeficientov metodo naredi nestabilno pri večjih vrednostih λ.

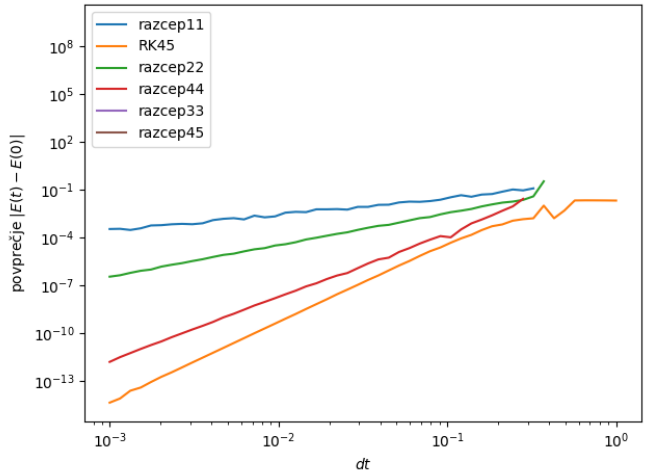
Poglejmo si zdaj še to, kako majhen dt rabimo, oziroma si lahko privoščimo, saj manjšanje tega poveča tudi čas računanja.



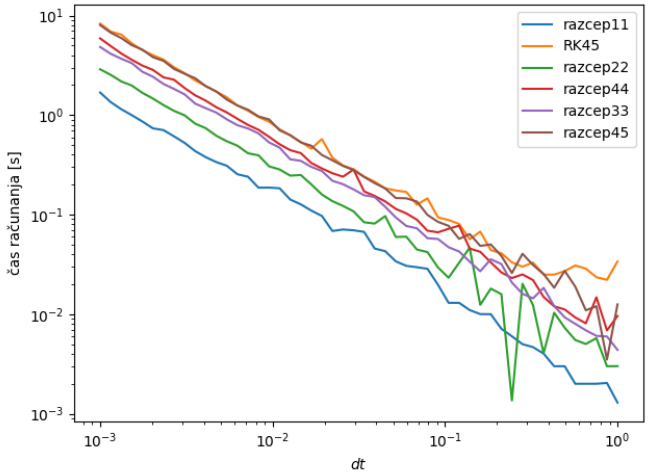
Slika 4: Povprečno odstopanje energije $\lambda = 0$



Slika 5: Čas računanja $\lambda = 0$



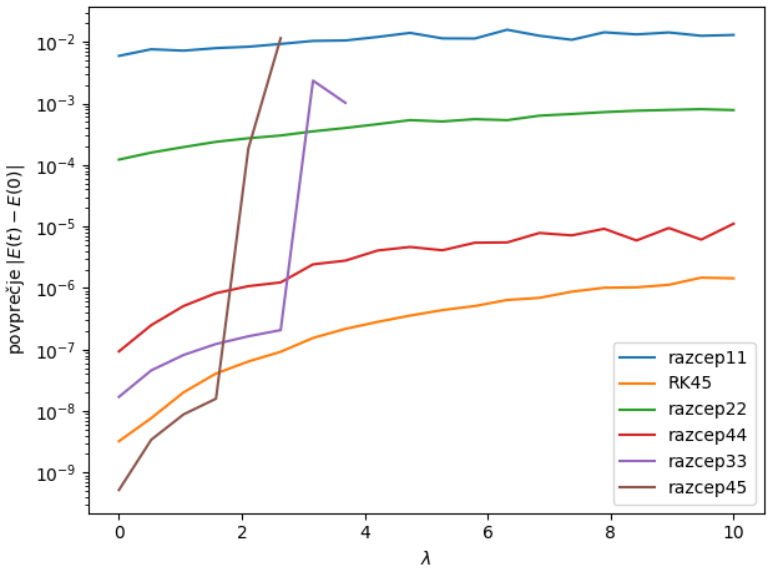
Slika 6: Povprečno odstopanje energije $\lambda = 10$, razcepi ki jih ni so vrnili vrednost ∞



Slika 7: Čas računanja $\lambda = 10$

Na grafih lahko vidimo, da simplektične metode lahko dosežejo zelo dobro natančnost pri natančnost ne glede na λ pri še vedno relativno velikem koraku. Glede ne to, da vidimo do 9 redov velikosti izbolšave v napaki in za to plačamo le z do okrog 10 kratnim podaljšanjem časa računanja, se izplača vzeti višji red razvoja, zagotovo vsaj drugi.

Kot zadnje pa si poglejmo še, kakšen vpliv ima λ na natančnost.



Slika 8: Povprečno odstopanje energije pri različnih metodah in λ pri $dt = 0.01$

Vidimo, da višanje reda načeloma izboljša napako na celo boljšo od RK45. Za metodi z kompleksnimi koeficienti pa to ne velja za velike λ, saj metoda po neki točki eksplodira k neskončni energiji.

2.3 Trajektorije

Do zdaj smo na problem gledali le iz perspektive ohranitve energije. Da bi bolje razumeli kaj se dejansko dogaja, sem se odločil narisati nekaj primerov orbit pri različnih vrednostih λ do časa $t_{max} = 100$. Te so v priloženi animaciji `animacija_1.mp4`. Pri $\lambda = 0$ vidimo, kot bi pričakovali, eliprično gibanje, ki z večanjem λ hitro izgubi začetno obliko. Do okrog $\lambda = 2$ je z izjemo $\lambda = 0.1$ gibanje še vedno izgleda periodično z daljšo periodo, na višjih vrednostih pa ni več opaznega vzorca, ki mu gibanje sledi. Poleg elipričnih orbit je zanimiva še posebna varianta začetnih pogojev, ko je ta krožna. To vidimo na `animacija_2.mp4`, kjer vidimo, da je razlika v začetnem opazna le pri majhnih λ. To, da sistem pozabi začetni pogoj, je še ena tipična lastnost kaotičnih sistemov.

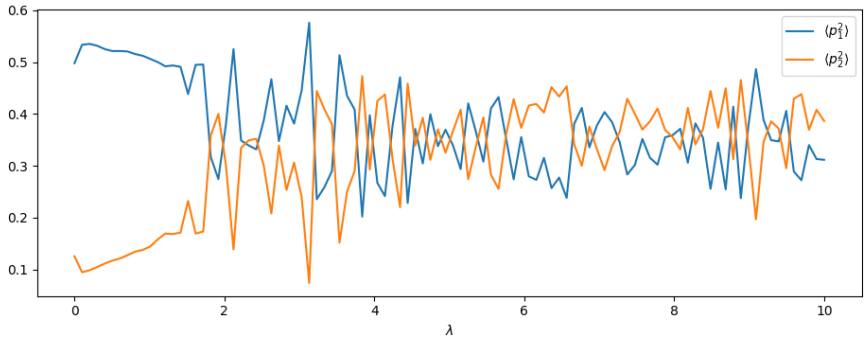
3 Ekviparticijski izrek

Ena izmed lastnosti, ki jih pričakujemo od kaotičnih sistemov, kar 2D anharmonski oscilator tudi je, je upoštevanje ekviparticijskega izreka. Ta pravi, da se v časovnem povprečju kinetična energija enakomerno porazdeli na vse gibalne prostostne stopnje, ko sistem doseže termalni ekvilibrij. Za nas to pomeni, da

pričakujemo, da bosta količini

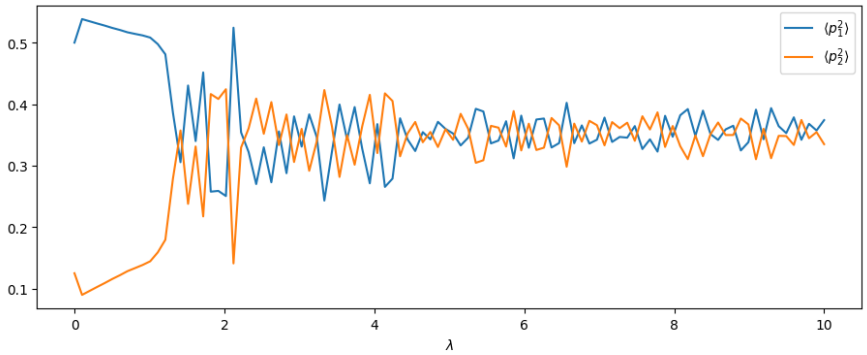
$$\langle p_i^2 \rangle = \frac{1}{t} \int_0^t p_i^2(t')dt' \qquad i = 1, 2,$$

za dovolj velik čas t enaki. Če ju zdaj narišemo pri $t_{max} = 100$, kot pri animaciji, z istim začetnim pogojem kot prej, pri različnih vrednostih λ,

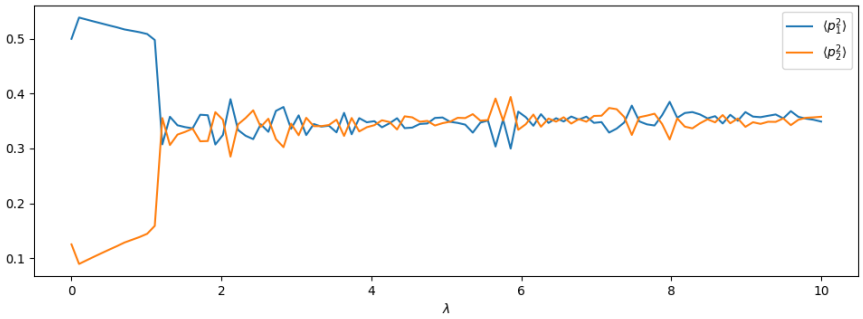


Slika 9: Časovno povprečje kvadratov gibalne količine pri $t = 10^2$

vidimo spet mejo med kaotičnim in nekaotičnim obnašanjem pri $\lambda \approx 2$. Iskaže pa se, da če vzamemo večji t_{max}, vidimo drugo mejo.

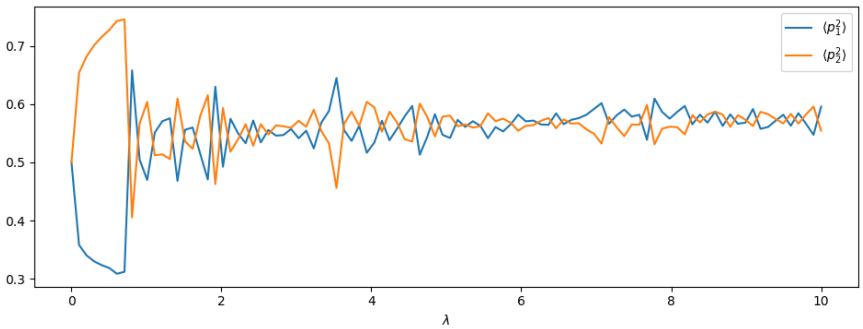


Slika 10: Časovno povprečje kvadratov gibalne količine pri $t = 10^3$



Slika 11: Časovno povprečje kvadratov gibalne količine pri $t = 10^4$

Pri večjih časih tudi dosti bolje vidimo, kako se energija enakomerno razporedi med prostostnima stopnjama. Iz grafov lahko sklepamo, da bi meja, kjer ekviparticija velja z večanjem končnega časa padala, dokler nebi dosegli $\lambda = 0$. Pri $\lambda = 0$ pa ekviparticija ni mogoča, saj imamo stabilno eliptično orbito, razen v primeru, ko je ta krožna.



Slika 12: Časovno povprečje kvadratov gibalne količine pri $t = 10^4$ za začetni pogoj $q_1 = 0, q_2 = 1, p_1 = 1, p_2 = 0$

4 Generacija Trotter-Suzuki shem

Za konec sem poskusil zapisati še python program, ki bi generiral poljubne Trotter-Suzuki sheme. To se je iskazalo za izjemno naporno, a sem vseeno uspel napisati program, ki najde enačbe, ki jih morajo koeficienti v shemi rešiti, za sheme do 4. reda, oziroma do poljubnega, če bi v program na roko dodal več členov Baker-Campbell-Hausdorfovega razvoja. Enačb sam ne reši, ampak to je konec koncev lažji del problema. Programu podamo naravni števili n in p. n je število eksponentov, ki jih želimo v končni shemi ie.

$$e^{a_1 z A} e^{b_1 z B} \rightarrow n = 2$$

$$e^{a_1 z A} e^{b_1 z B} e^{a_2 z A} \rightarrow n = 3$$

p pa red do katerega je razvoj točen. Če enačbe nimajo rešitve takšna shema ne obstaja. Program kličemo iz datoteke `split_step.py` kot `get_equations(n,p)`, kot rezultat pa se v terminal izpišejo enačbe kot

```
[A]:   a1 + a2 = 1
[B]:   b1 = 1
[BA]:   1/2 * b1a2 + -1/2 * a1b1 = 0
```

Slika 13: Enačbe za shemo $n = 3$, $p = 2$, ki nam dajo razvoj $e^{zA} e^{zB} = e^{\frac{1}{2} zA} e^{zB} e^{\frac{1}{2} zA}$