

UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
ODDELEK ZA FIZIKO

MATEMATIČNO-FIZIKALNI PRAKTIKUM
7. naloga: Newtonov zakon

Žiga Šinigoj, 28191058

Ljubljana, december 2021

1 Uvod

Gibanje masne točke v polju sil v eni dimenziji opišemo z diferencialno enačbo drugega reda, z Newtonovim zakonom

$$m \frac{d^2x}{dt^2} = F.$$

Enačba je seveda enakovredna sistemu enačb prvega reda

$$m \frac{dx}{dt} = p, \quad \frac{dp}{dt} = F$$

in tako jo tudi rešujemo: kot sistem dveh enačb prvega reda.

Seveda morajo biti na voljo tudi ustrezni začetni pogoji, tipično $x(t=0) = x_0$ in $dx/dt = v(t=0) = v_0$. Splošnejše gre tu za sistem diferencialnih enačb drugega reda:

$$\frac{d^n y}{dx^n} = f(x, y, y', y'', \dots),$$

ki ga lahko prevedemo na sistem enačb prvega reda z uvedbo novih spremenljivk v slogu gibalne količine pri Newtonovi enačbi ($y' = v, y'' = z, \dots$).

Z nekaj truda se da eksplicitno dokazati, mi pa lahko privzamemo, da so metode za reševanje enačb hoda (Runge-Kutta 4. reda, prediktor-korektor...) neposredno uporabne za reševanje takšnih sistemov enačb in torej aplikabilne v poljubno dimenzijah, kar naj bi v principu zadovoljilo večino naših zahtev.

Obstaja še posebna kategorija tako imenovanih *simplektičnih* metod, za enačbe, kjer je f le funkcija koordinat, $f(y)$, ki (približno) ohranjajo tudi Hamiltonian, torej energijo sistema. Najbolj znana metoda je Verlet/Störmer/Encke metoda, ki je globalno natančna do drugega reda in ki točno ohranja tudi vrtilno količino sistema (če je ta v danem problemu smiselna). Rešujemo torej za vsak diskretni korak n velikosti h , $x_n = x_0 + n \cdot h$:

$$\frac{d^2y}{dx^2} = f(y)$$

in pri diskretizaciji dobimo recept za korak y_n in $v_n = y'_n$:

$$\begin{aligned} y_{n+1} &= y_n + h \cdot v_n + \frac{h^2}{2} \cdot f(y_n) \\ v_{n+1} &= v_n + \frac{h}{2} \cdot [f(y_n) + f(y_{n+1})]. \end{aligned}$$

Alternativno lahko to shemo zapišemo tudi s pomočjo dodatnih vmesnih točk in preskakujemo med lego in hitrostjo z zamikom $h/2$ (od tod angleško ime 'leapfrog' za ta zapis):

$$\begin{aligned} y_{n+1} &= y_n + h \cdot v_{n+1/2} \\ v_{n+3/2} &= v_{n+1/2} + h \cdot f(y_{n+1}). \end{aligned}$$

V še enem drugačnem zapisu je metoda poznana tudi kot metoda "Središčne razlike" (Central Difference Method, CDM), če nas hitrost ne zanima:

$$y_{n+1} - 2y_n + y_{n-1} = h^2 \cdot f(y_n),$$

kjer prvo točko y_1 izračunamo po originalni shemi. Metodo CDM lahko uporabljamo tudi za primere, ko je f tudi funkcija 'časa' x , $f(x, y)$, le da tu simplektičnost ni zagotovljena (in tudi verjetno ne relevantna). Za simplektične metode višjih redov je na voljo na primer Forest-Ruth metoda ali Position Extended Forest-Ruth Like (PEFRL) metoda, ki sta obe globalno četrtega reda in enostavni za implementacijo.

2 Naloga

Čim več metod uporabi za izračun nihanja matematičnega nihala z začetnim pogojem $\vartheta(0) = \vartheta_0 = 1, \dot{\vartheta}(0) = 0$. Poišči korak, ki zadošča za natančnost na 3 mesta. Primerjaj tudi periodično stabilnost shem: pusti, naj teče račun čez 10 ali 20 nihajev in poglej, kako se amplitude nihajev sistematično kvarijo. Pomagaš si lahko tudi tako, da občasno izračunaš energijo $E \propto 1 - \cos \vartheta + \frac{\dot{\vartheta}^2}{2\omega_0^2}$. Nariši tudi ustrezne fazne portrete!. Z analitično rešitvijo dobimo za nihajni čas $\frac{4}{\omega_0} K\left(\sin^2 \frac{\vartheta_0}{2}\right)$, kjer je $K(m)$ popolni eliptični integral prve vrste, ki je v SciPy knjižnici in v članku na spletni učilnici podan z:

$$K(m) = \int_0^1 \frac{dz}{\sqrt{(1-z^2)(1-mz^2)}} = \int_0^{\frac{\pi}{2}} \frac{du}{\sqrt{(1-m\sin^2 u)}}$$

Previdno, obstaja tudi definicija z m^2 v integralu - potem je prav $K\left(\sin \frac{\vartheta_0}{2}\right)$, brez kvadrata (npr že v Wikipediji)!

(Dodatno lahko tudi sprogramirate eliptični integral, ki je analitična rešitev dane enačbe ali pa ga vzamete iz ustreznih programskih knjižnic).

Dodatna naloga: Razišči še resonančno krivuljo vzbujenega dušenega matematičnega nihala

$$\frac{d^2x}{dt^2} + \beta \frac{dx}{dt} + \sin x = v \cos \omega_0 t,$$

kjer je β koeficient dušenja, v in ω_0 pa amplituda in frekvenca vzbujanja. Opazuj obnašanje odklonov in hitrosti nihala pri dušenju $\beta = 0.5$, vzbujevalni frekvenci $\omega_0 = 2/3$ in amplitudo vzbujanja na območju $0.5 < v < 1.5$. Poskusi opaziti histerezno obnašanje resonančne krivulje pri velikih amplitudah vzbujanja (Landau, Lifšic, CTP, Vol. 1, *Mechanics*).

Dodatna dodatna naloga: če ti gre delo dobro od rok, si oglej še odmike in hitrosti (fazne portrete) van der Polovega oscilatorja

$$\frac{d^2x}{dt^2} - \lambda \frac{dx}{dt} (1 - x^2) + x = v \cos \omega_0 t$$

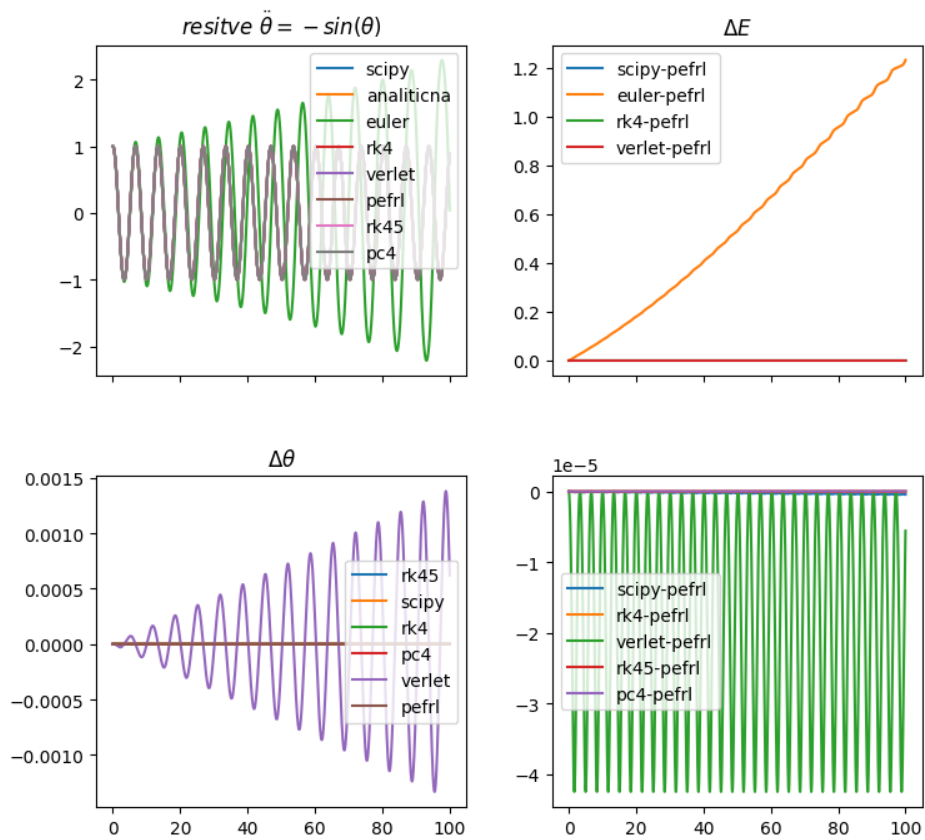
s parametri $\omega_0 = 1$, $v = 10$ ter $\lambda = 1$ ali 100. Tu se ne trudi s preprostimi diferenčnimi shemami: problem je nelinearen in tog, zato uporabi neko preverjeno metodo (na primer iz družine Runge-Kutta ali ekstrapolacijsko metodo) s prilagajanjem velikosti koraka.

3 Rezultati

3.1 Matematično nihalo

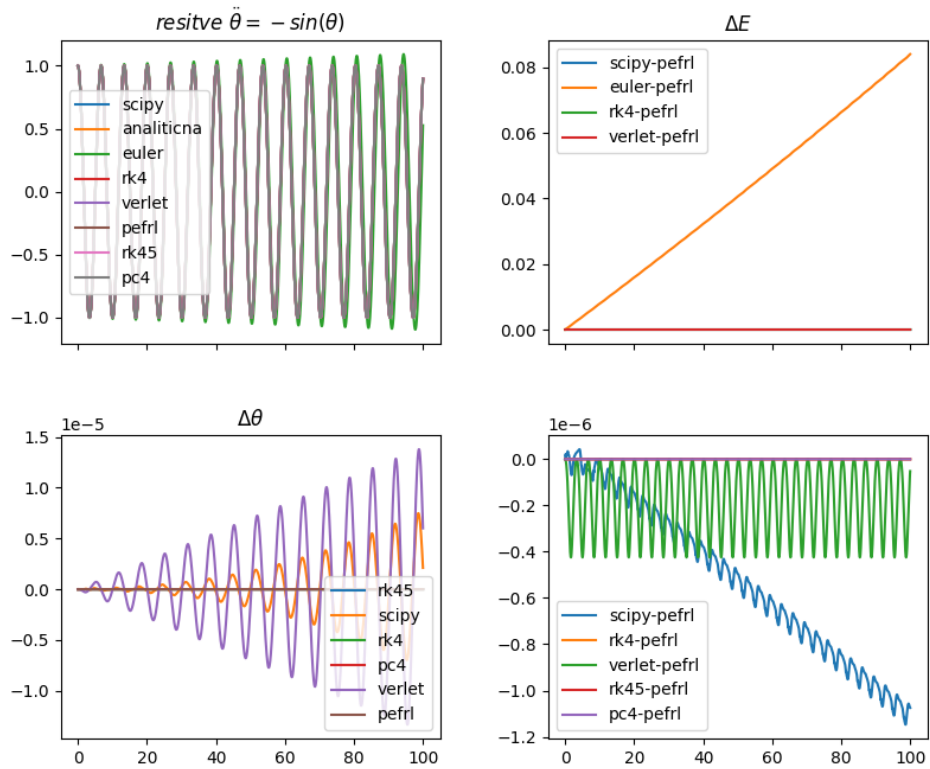
Za izračun gibanja nihala sem uporabil več metod. Začetne pogoje označuje vektor $\vec{\vartheta}$. V splošnem je natančnost odvisna od velikosti koraka in dolžine intervala na katerem iščemo rešitev. Na sliki 1 in 2, so prikazane rešitve, ki jih podajo različne metode in energije sistema, pri dveh različnih korakih. Pri energijah sem vzel za pravo energijo, energijo simplektične metode PEFRL, ki jo ohranja. To ni bilo nujno potrebno, saj lahko gledam energijo, ki jo poda dana metoda in morebitno spreminjanje s časom. To sem kasneje tudi naredil. Gre pa za eno in isto stvar, zanima nas če metoda ohranja energijo in razlika ne spremeni funkcijske odvisnosti v času. Z zmanjševanjem koraka se natančnost metod povečuje in energija sistema se počasneje spreminja. Zanimivo je, da se pri koraku 0.002 večja napaka SciPy metode *odeint*, zato sem hotel pogledati natančnost rešitev metod v odvisnosti od velikosti koraka (slika 13). Za oceno napake na intervalu sem vzel 2. normo vektorja odstopanj leg nihala. Težava, na katero sem naletel je, da če vzamem vedno enak interval in drobim korak, ima vektor razlik vedno večjo dolžino in se napaka metode z manjšim korakom povečuje. Moral sem vzeti vedno enako število točk, oz. konstantno velikost vektorja. To povzroči, da sta velikost in dolžina koraka povezani. Z velikostjo koraka se povečuje interval in posledično tudi napaka, ki se z dolžino intervala večja. To pomeni, da se lahko krivulje na sliki 13 premikajo v vertikalni smeri, odvisno od izbrane dolžine intervala. Vseeno je videti, da se na neki točki natančnost Verlet in Odeint metode obrne.

dt=0.02

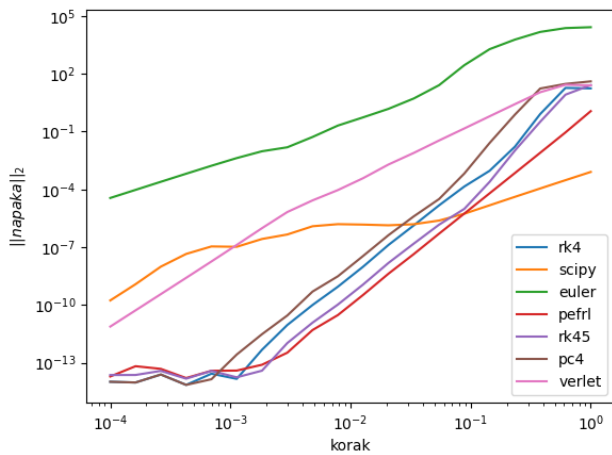


Slika 1: Rešitev nihala, energija in odstopanja. $\vec{\vartheta}(0) = [1, 0]$

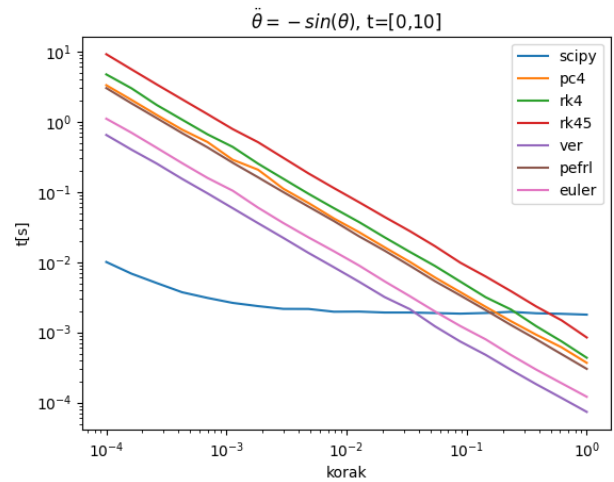
dt=0.002



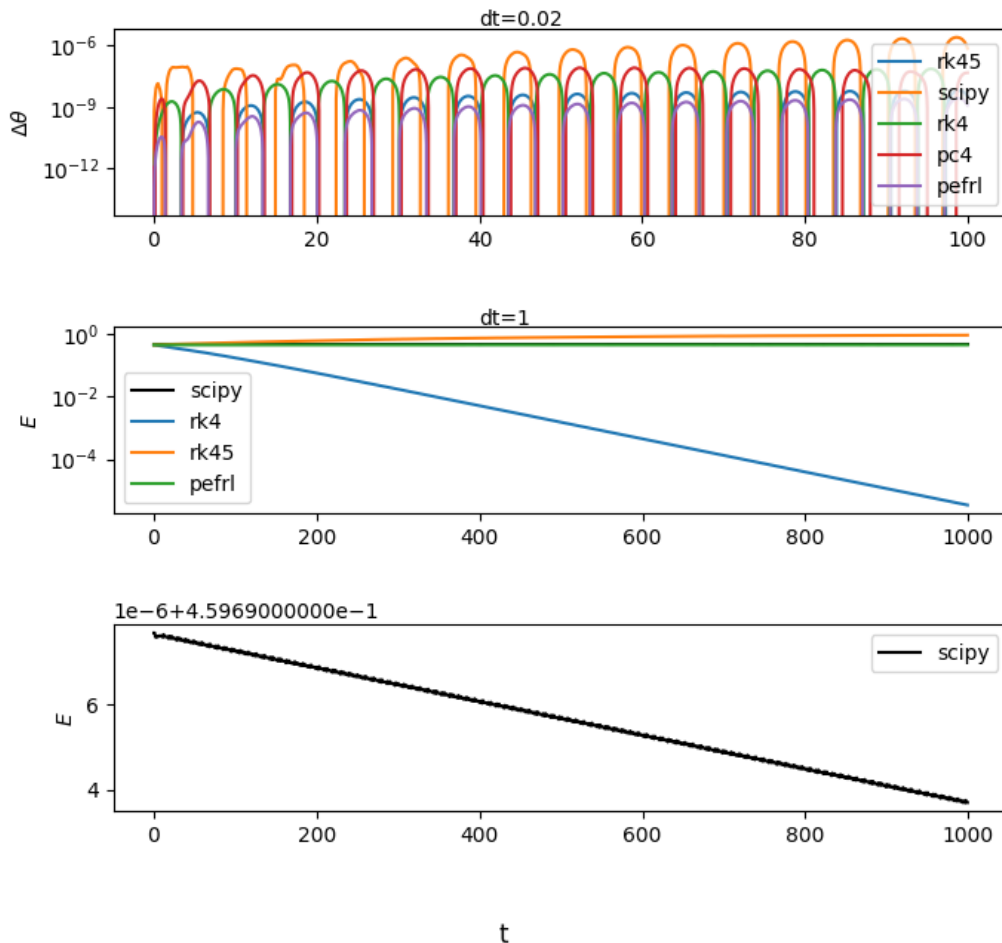
Slika 2: Rešitev nihala, energija in odstopanja. $\vec{\vartheta}(0) = [1, 0]$



Slika 3: Odstopanje metode v odvisnosti od velikosti koraka.

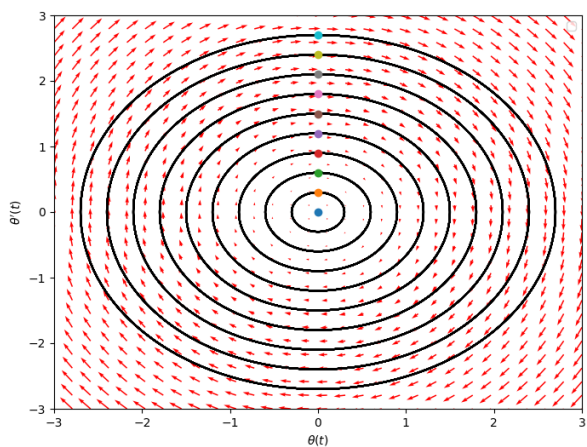


Slika 4: Časovna zahtevnost

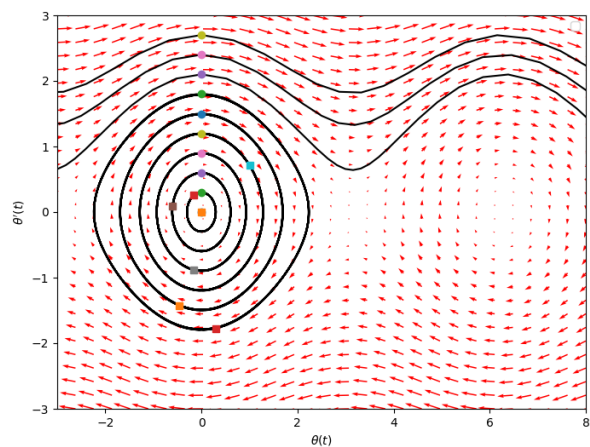


Slika 5: Maksimalna absolutna napaka pri dani velikosti koraka.

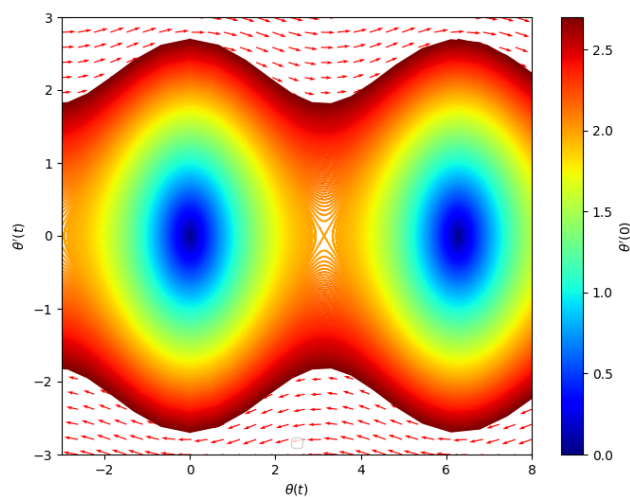
Časovna zahtevnost (slika 4) je eksponentna razen pri SciPy metodi, ki je najhitrejša. Natančnost metod še v logaritemski skali prikazuje slika 5. Energijo izgubljajo Runge-Kutta metode in SciPyjeva metoda, čeprav jo slednja zelo počasi. Simplektične metode (PEFRL, Verlet) so edine, ki ohranjajo energijo, ponavadi nihajo okrog oz. ob energiji sistema. Na faznih diagramih (slika 14, 8) predstavljajo okogle točke začetne vrednosti, kvadratne pa končne.



Slika 6: Fazni diagram linearnega nihala.



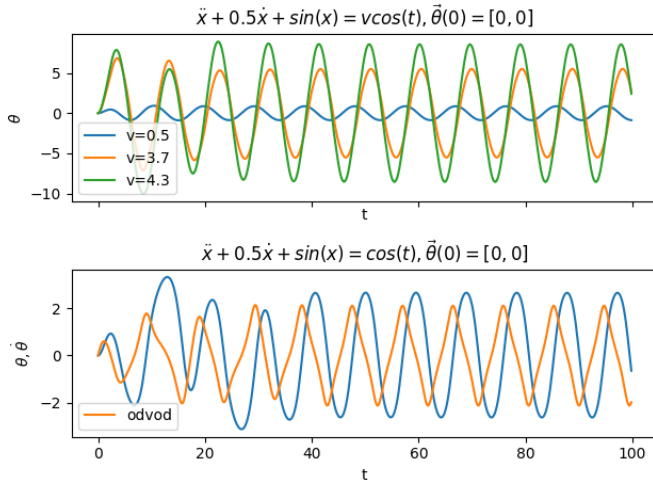
Slika 7: Fazni diagram matematičnega nihala.



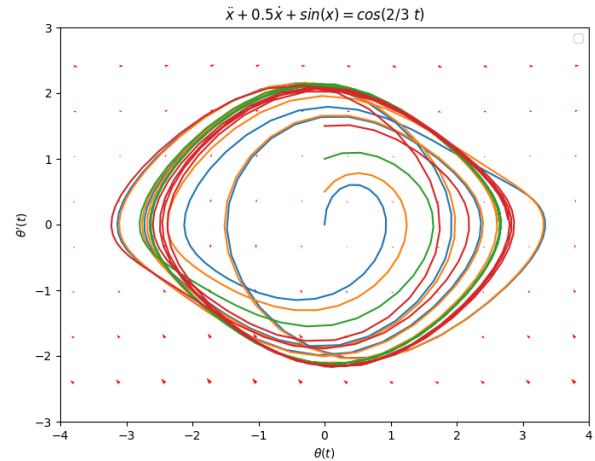
Slika 8: Fazni diagram matematičnega nihala z umetniškim navdihom.

3.2 Dušeno matematično nihalo z vzbujanjem

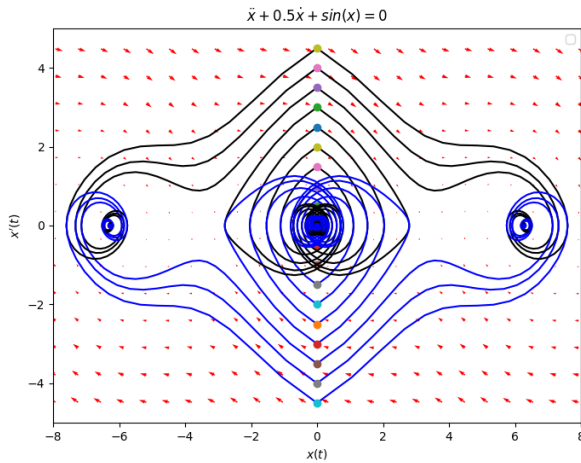
Za izračun sistema sem uporabil SciPyjevo funkcijo *integrate.odeint()*. Ker metoda ne ohranja energije sem moral pri faznih diagramih paziti, da nisem vzel veliko period nihanja. Če bi vzel veliko period bi dobil fazni diagram, ki bi izgubljal energijo in krivulje bi se spreminjale. To lahko privede do napačne interpretacije rezultatov. Fazni diagrami so periodični s periodo 2π .



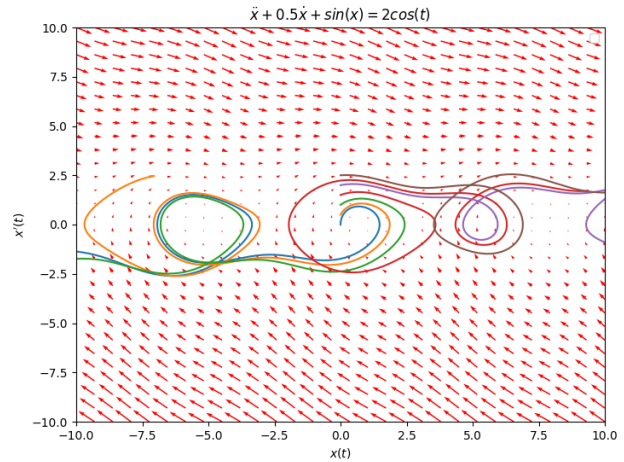
Slika 9: Rešitev matematičnega nihala z vzbujanjem in nekaj resonančnih krivulj.



Slika 10: Fazni diagram dušenega matematičnega nihala z vzbujanjem, pri različnih začetnih pogojih.



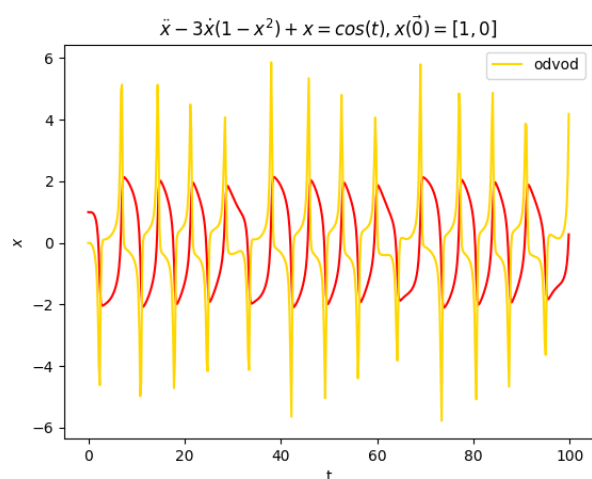
Slika 11: Fazni diagram dušenega matematičnega nihala, pri različnih začetnih pogojih.



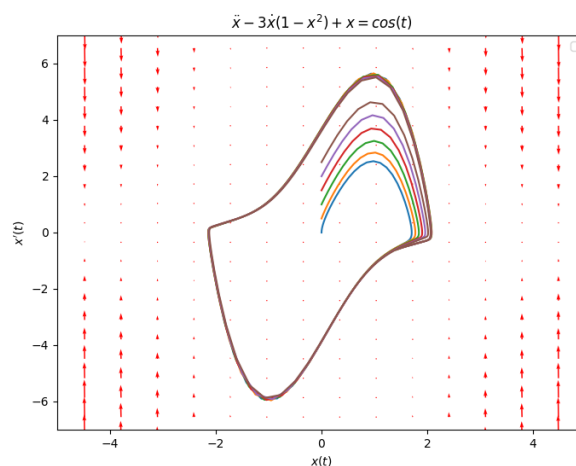
Slika 12: Fazni diagram dušenega matematičnega nihala s prevelikim vzbujanjem, pri različnih začetnih pogojih.

3.3 Van der Polov oscillator

Za izračun sistema sem uporabil SciPyjevo funkcijo `integrate.odeint()`.



Slika 13: Rešitev Van der Polovega nihala.



Slika 14: Fazni diagram, pri različnih začetnih vrednostih.

4 Zaključek

Pri reševanju diferencialnih enačb z začetnimi pogoji so obravnavane metode, razen Eulerjeve, precej natančne. Vedno pa je treba preverjati ali so rezultati smiselni za dani fizikalni problem. Pri novih fizikalnih problemih, kjer rezultata ne poznamo, je dobro uporabiti tudi simplektične metode, ki ohranjajo energijo sistema. S tem se izognemo morebitni napačni interpretaciji rešitve.