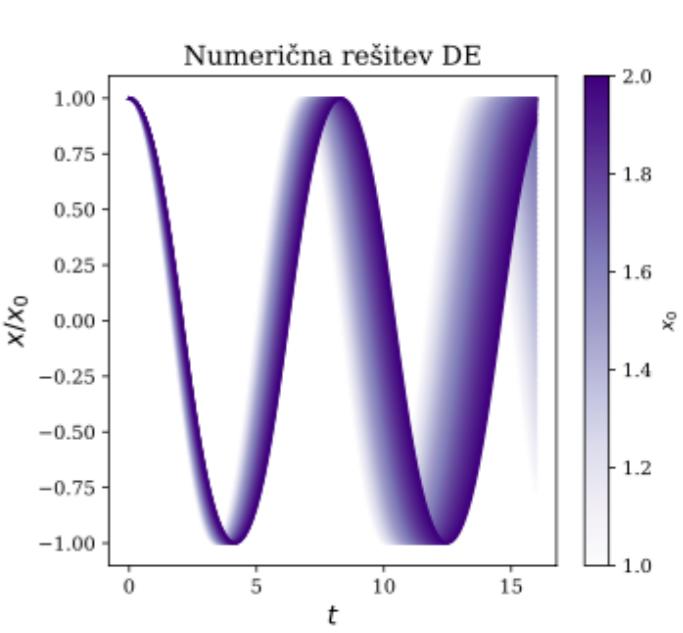
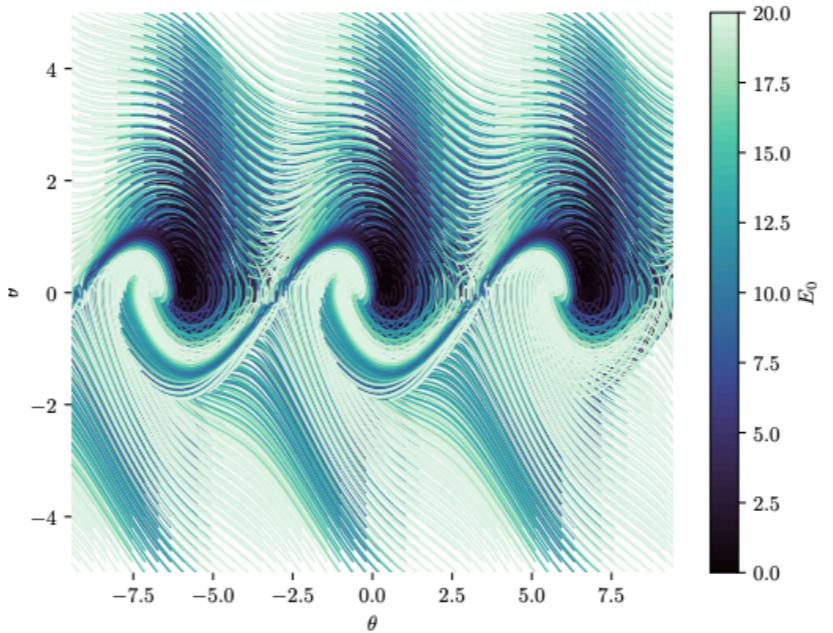
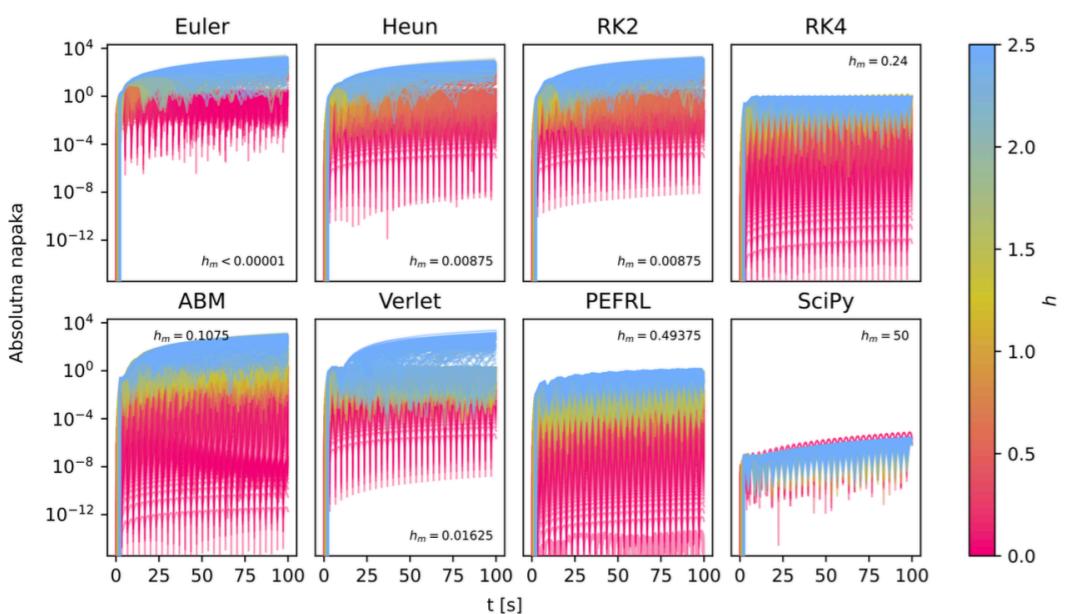




7. naloga

2022/23

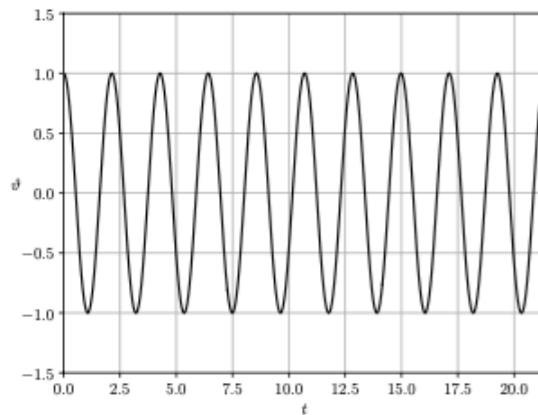




Začetni koraki

- Smiselno je preveriti metodo s preprostimi testi!
 - Oglej si analitično rešitev...

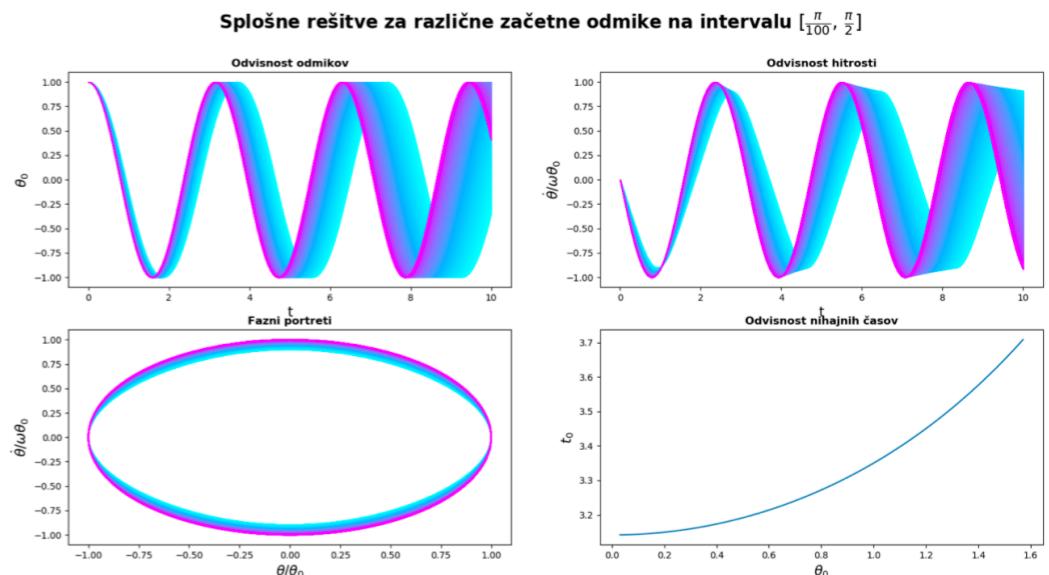
BRAVO!



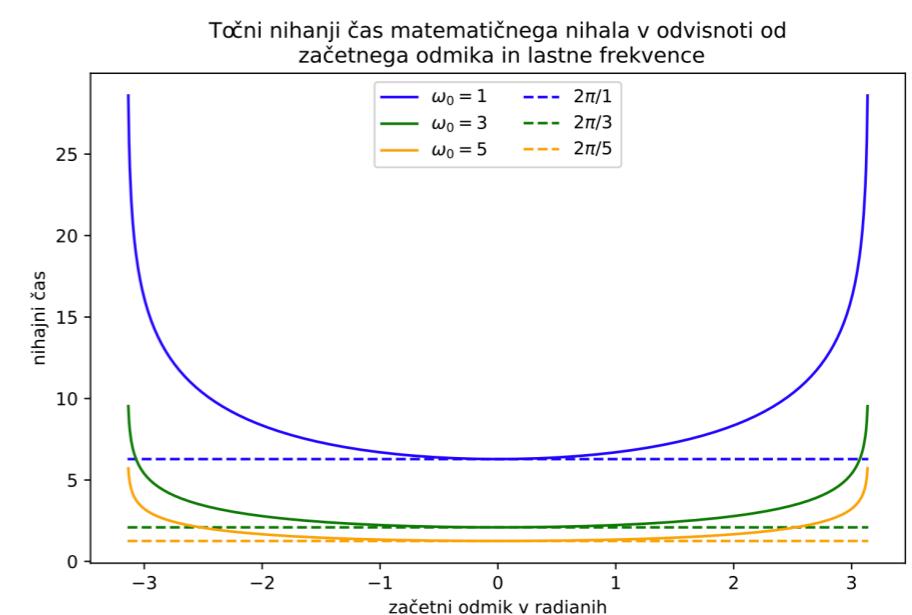
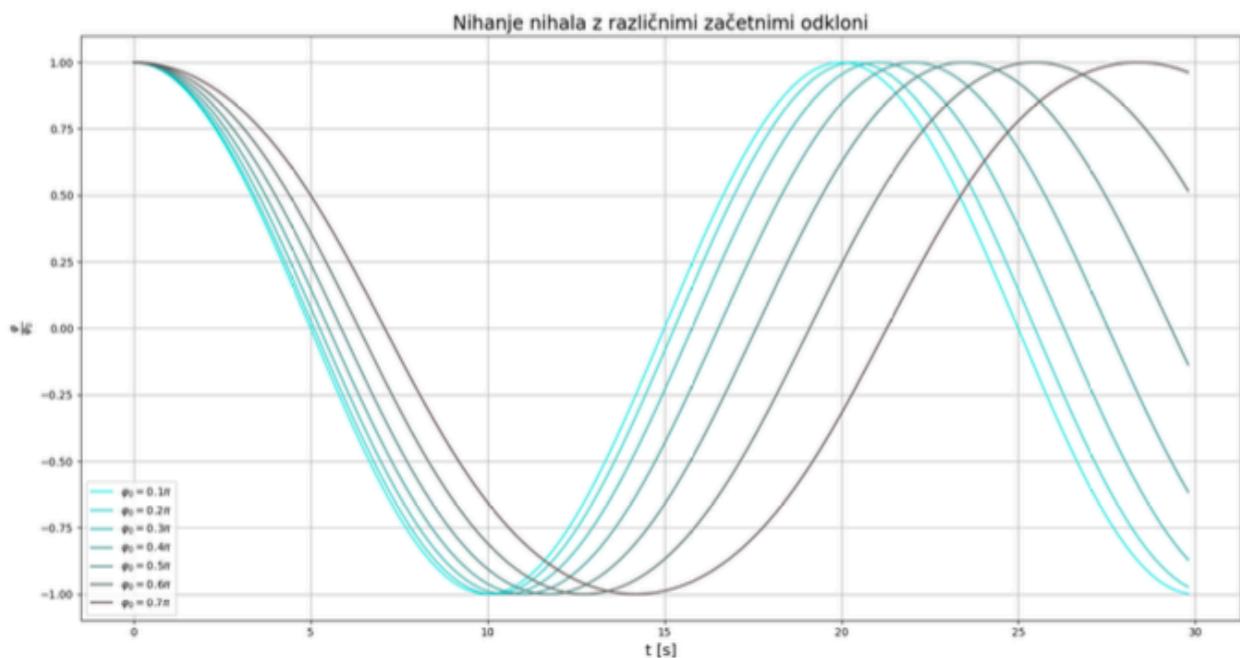
(a) Časovna odvisnost kota za enostavno nihalo
(b) Fourierova transformacija časovne odvisnosti kota za enostavno nihalo

Slika 2: Analitična rešitev matematičnega nihala

Lani ...



Slika 1: Analične rešitve odmikov, hitosti in nihajnih časov za nekaj začetnih odmikov. Z roza barvo se začnejo manjši začetni odmiki, ki se z večanjem prelivajo v modro. $\omega = 2$



Slika 1: Na grafu je prikazana odvisnost nihajnega časa od začetnega odmika in lastne frekvene. Vidimo lahko kako zelo različen je glede na vrednost, ki jo uporabljammo pri majhnih odmikih. Divergiranje nihajnega časa pri $\pm\pi$ je zaradi labilne legi, ko je nihalo točno nad mirovno lego.



Natančnost/zahtevnost

- Bolj natančen pogled v kodiranje iste metode...

Lani... (Henrici)

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 $v_n = y'_n$:

$$y_{n+1} = y_n + h \cdot v_n + \frac{h^2}{2} \cdot f(y_n) \quad (1)$$

$$v_{n+1} = v_n + \frac{h}{2} \cdot [f(y_n) + f(y_{n+1})]. \quad (2)$$

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):

$$y_{n+1} = y_n + h \cdot v_{n+1/2} \quad (3)$$

$$v_{n+3/2} = v_{n+1/2} + h \cdot f(y_{n+1}). \quad (4)$$

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

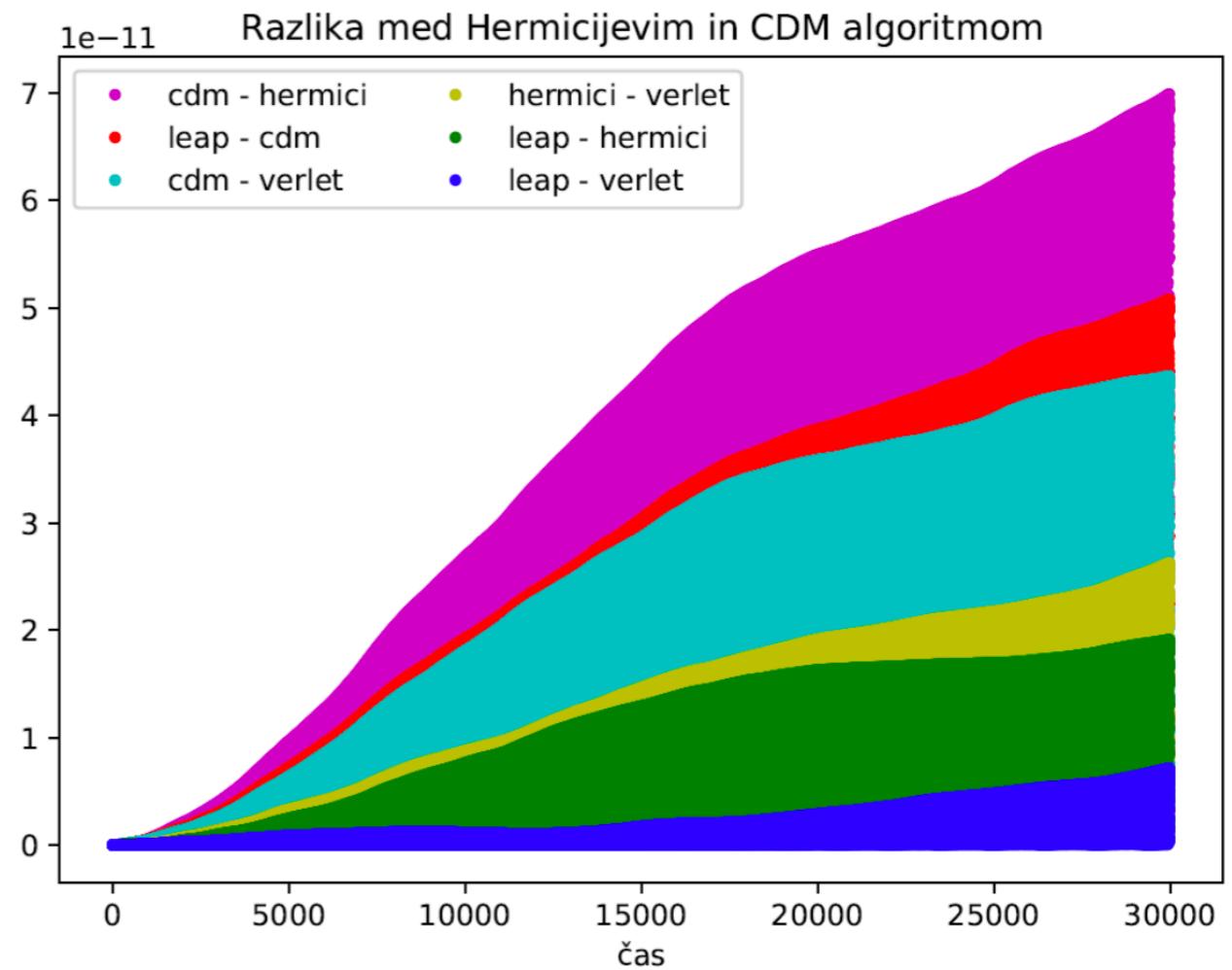
$$y_{n+1} - 2y_n + y_{n-1} = h^2 \cdot f(y_n), \quad (5)$$

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). Obstaja še četrti zapis Verletove metode, ki jo je vpeljal Hermici in je le drugačen zapis CDM metode:

$$\Delta_n = y_n - y_{n-1} \quad (6)$$

$$\Delta_{n+1} = \Delta_n + h^2 f(y_n) \quad (7)$$

$$y_{n+1} = y_n + \Delta_{n+1} \quad (8)$$



Slika 7: Na grafu so prikazane absolutne vrednosti razlik med pari 4 različnih interpretacij Verletove metode opisanih v poglavju 1. Na grafu ni nič polnjeno z barvami, prikazanih je le toliko točk, da se jih ne loči več med sabo. Razlike so seveda precej majhne glede na ostale napake, ki sem jih računal, a so zaznavne in niso posledica končne natančnosti računalnika, saj so za 3-4 rede večje kot natančnost računanja računalnika.

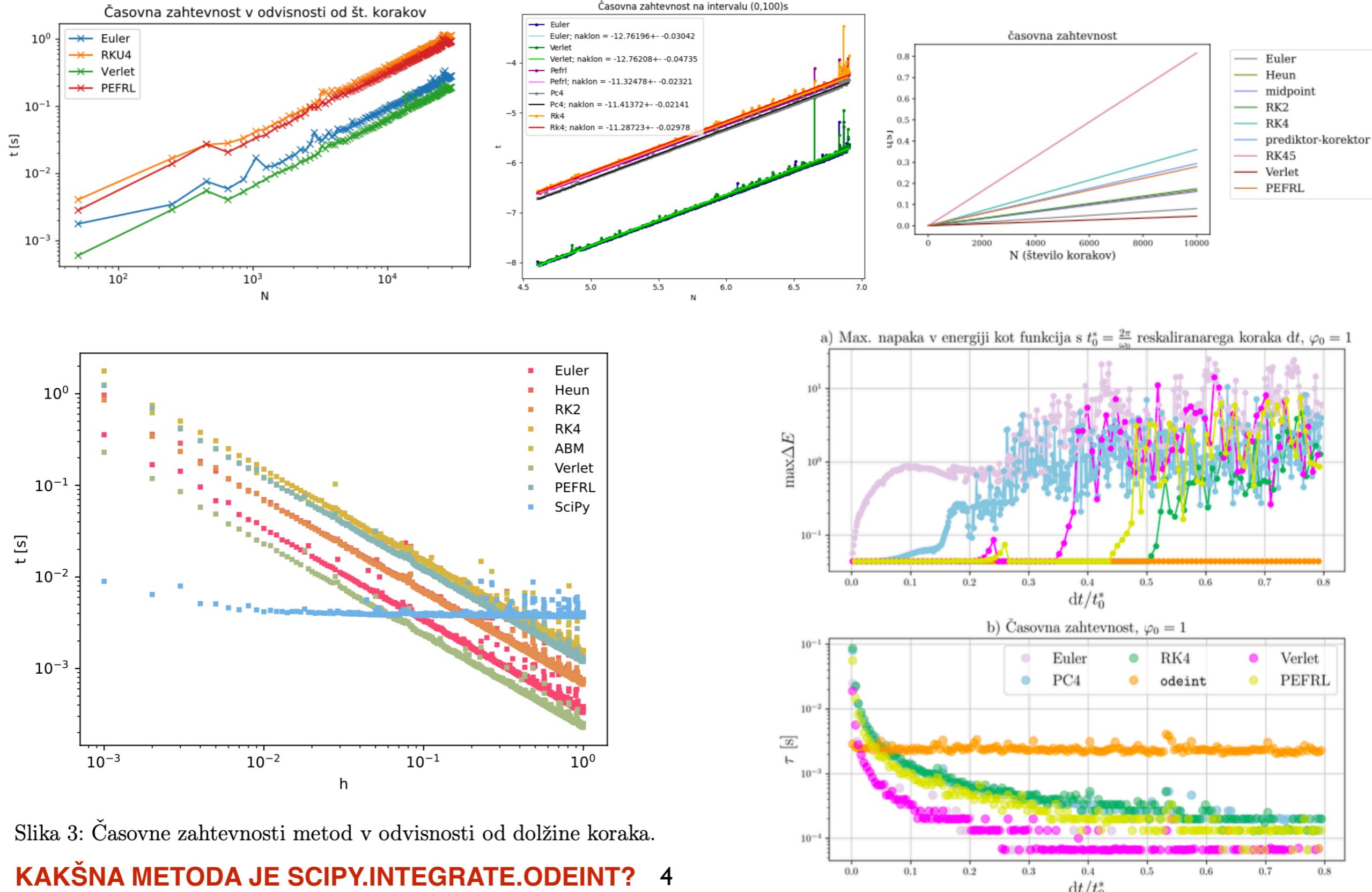
Na grafu je prikazana razlika med pari 4 različnih interpretacij Verletove metode. 'cdm' metoda je po enačbi (5), 'leap' metoda je po enačbi (3) in (4), 'hermici' je po enačbah (6)-(8) in verlet je po enačbah (1) in (2).

Na grafu (7) lahko vidimo, da izračuni po teh interpretacijah niso identični. Ko sem skušal ugotoviti, katera metoda je najbolj natančena glede na analitično rešitev, je pri različnih začetnih pogojih in časovnem intervalu bila najboljša drugačna metoda. Tudi to me je zelo presenetilo. Skladnost metode in analitične rešitve sem gledal kot vsoto absolutnih vrednosti razlik po točkah na določenem intervalu.



Računska požrešnost...

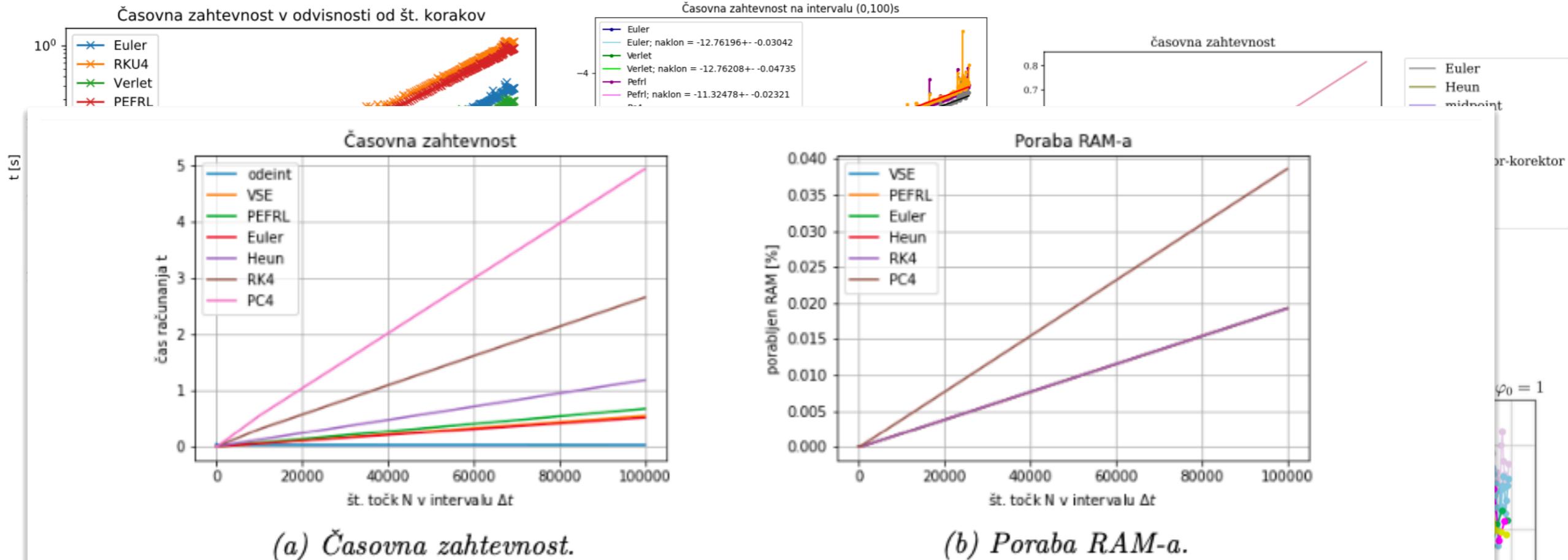
- Še vedno je dobrodošla tudi ocena računske zahtevnosti!



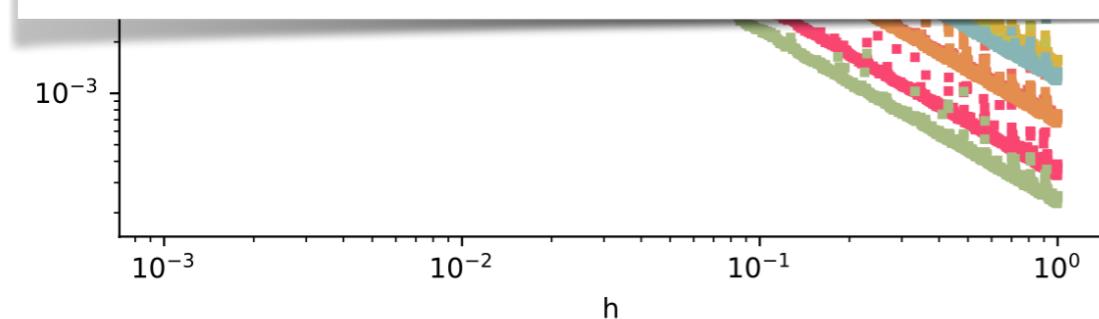


Računska požrešnost...

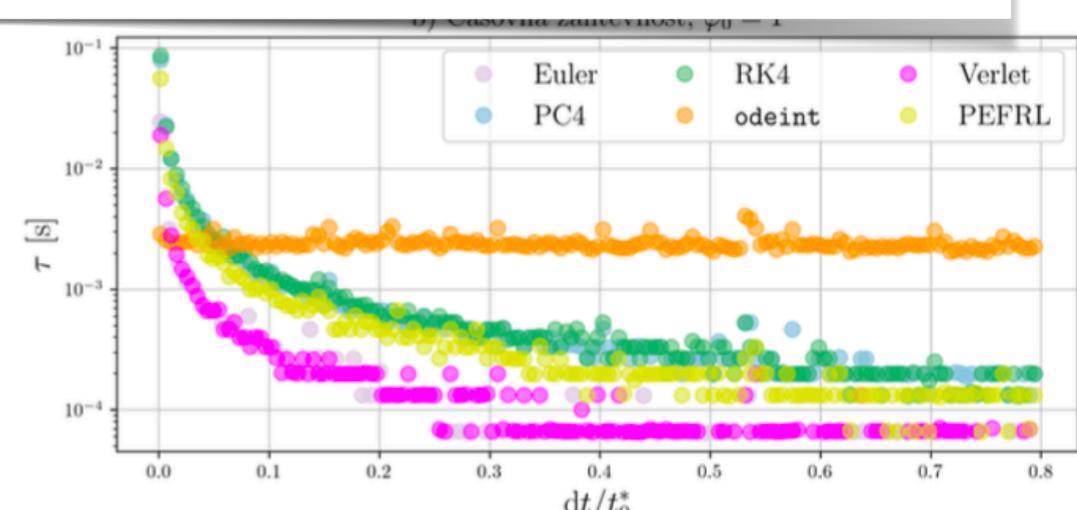
- Še vedno je dobrodošla tudi ocena računske zahtevnosti!



Slika 1: Iz slike (a) je razvidno, da je časovno nabolj potratna PC4 metoda, ki prav tako porabi največ RAM-a. Vse ostale metode imajo nato manjšo časovno zahtevnost in porabo RAM-a.



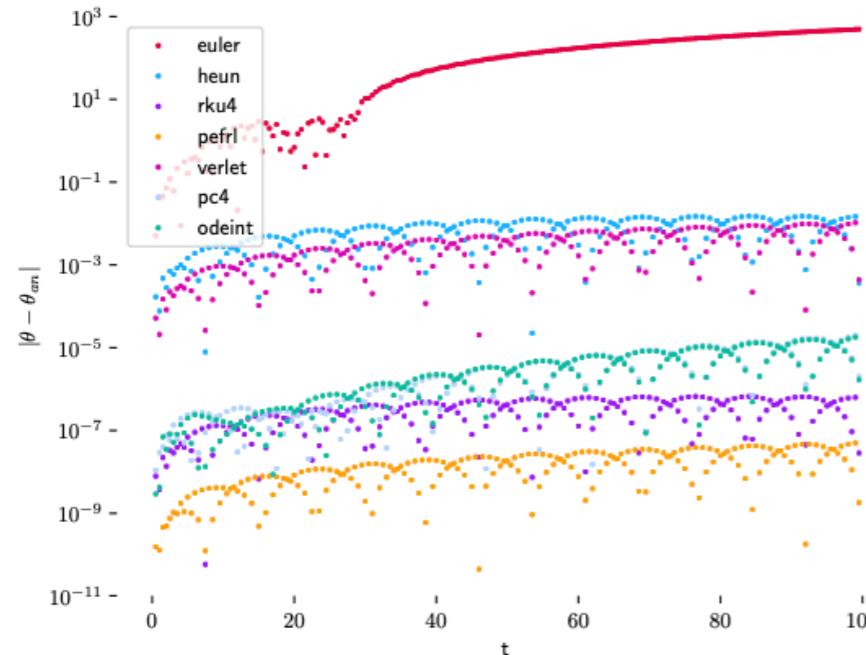
Slika 3: Časovne zahtevnosti metod v odvisnosti od dolžine koraka.



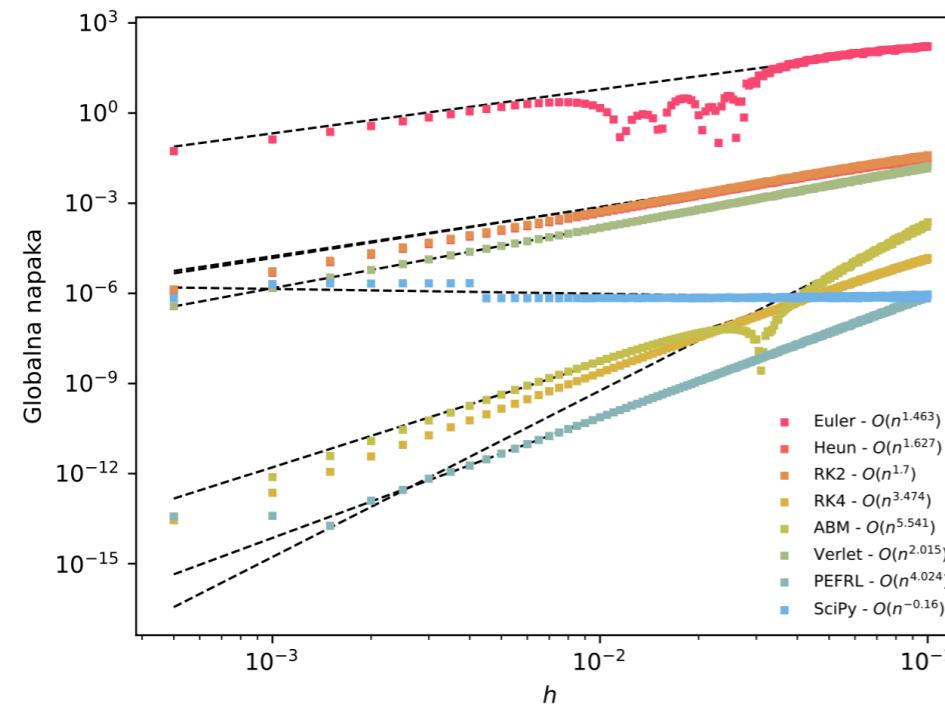


Natančnost/zahtevnost

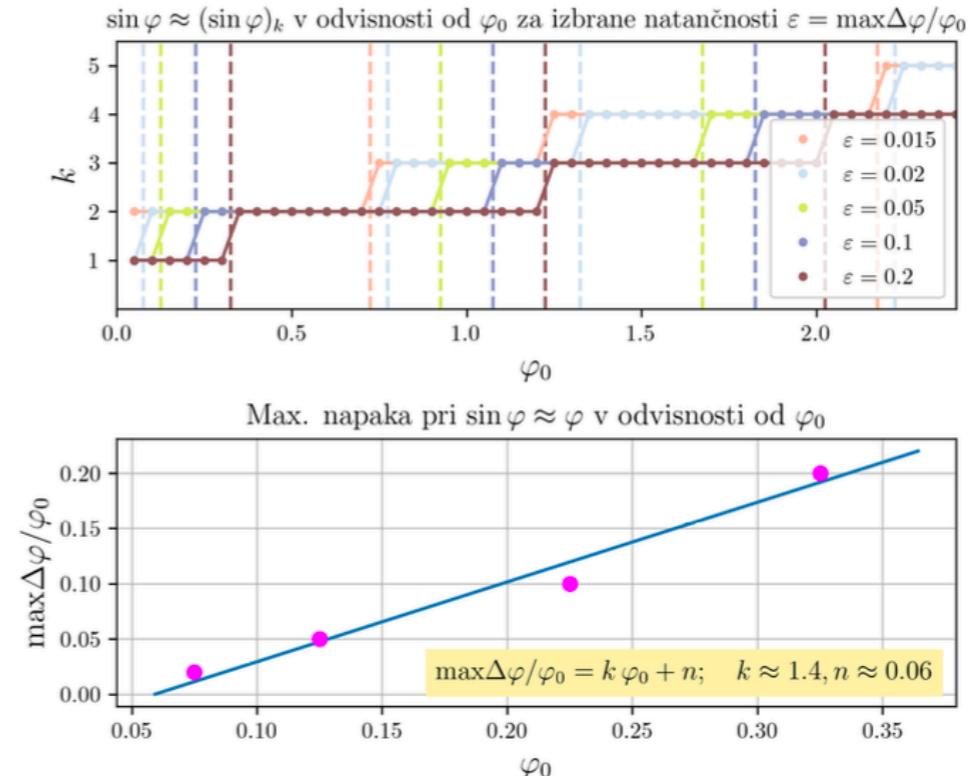
- Ocene natančnosti v različnih oblikah...



Slika 1: Globalna napaka v odvisnosti od časa za vsako metodo



Slika 5: Globalna napaka metod v odvisnosti od velikosti koraka ($h = [0, 0.1]$). Obe osi sta v logaritemski skali.



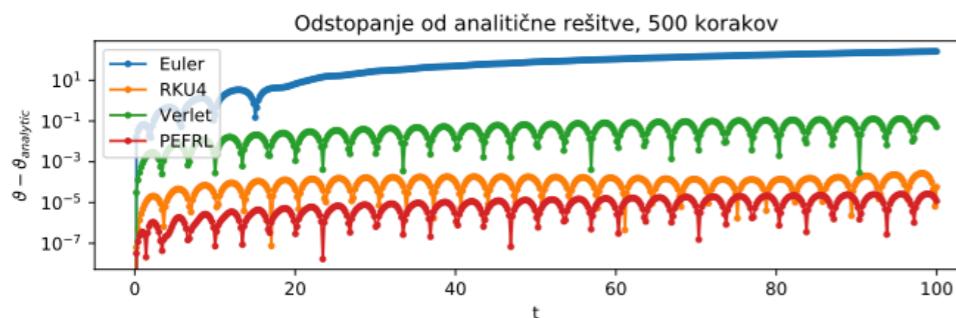
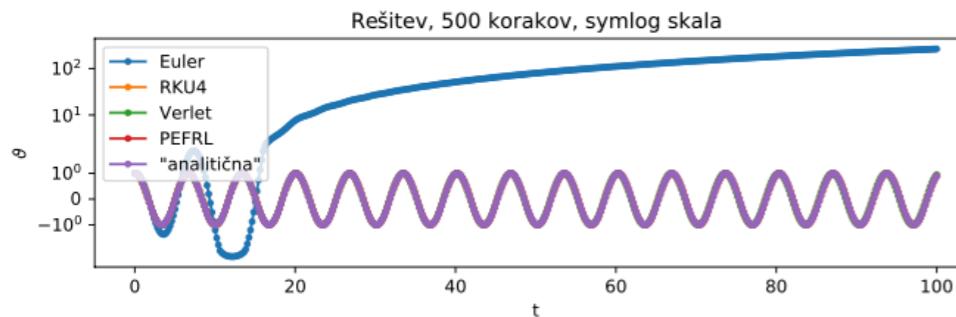
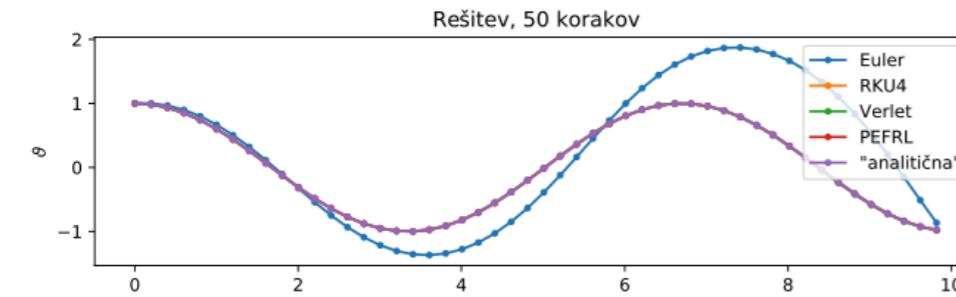
Slika 3: Zgornji graf pokaže, koliko členov v Taylorjev razvoju sin moramo pri Verletovi metodi vzeti pri $dt = 0.01$ s za izbrano natančnost $\varepsilon = \max\Delta\varphi/\varphi_0$ za različne začetne odmike φ_0 . Spodnji graf se osredotoča na mejo med $k = 1$ in $k = 2$, saj pri nihalih in oscilatorjih večinoma vzamemo $\sin \varphi \approx \varphi$. Relativna napaka raste linearno z φ_0 .

Metoda	Globalna napaka Izračunana	$a, O(n^a)$ Teoretična
Euler	1.463 ± 0.0315	2
Heun	1.627 ± 0.0119	3
RK2	1.7 ± 0.0124	3
RK4	3.474 ± 0.0189	5
ABM	5.541 ± 0.0802	5
Verlet	2.015 ± 0.0166	2
PEFRL	4.024 ± 0.0325	4

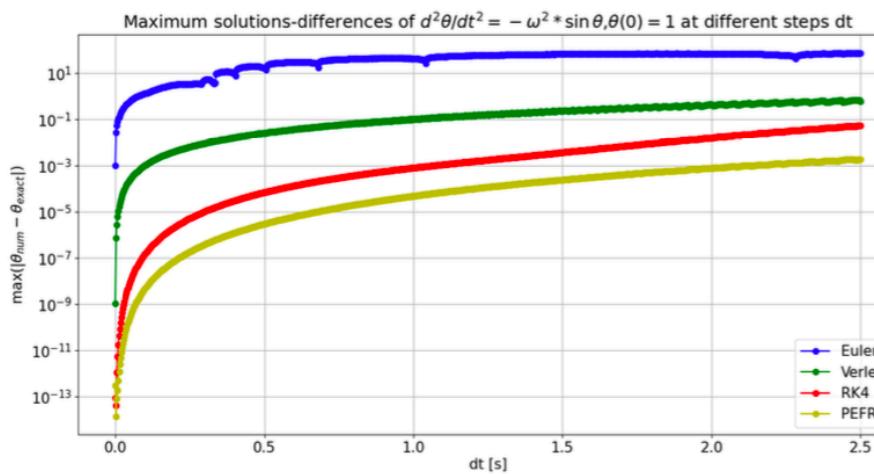
Tabela 2: Tabela izračunanih in podanih vrednosti.

Natančnost/zahtevnost

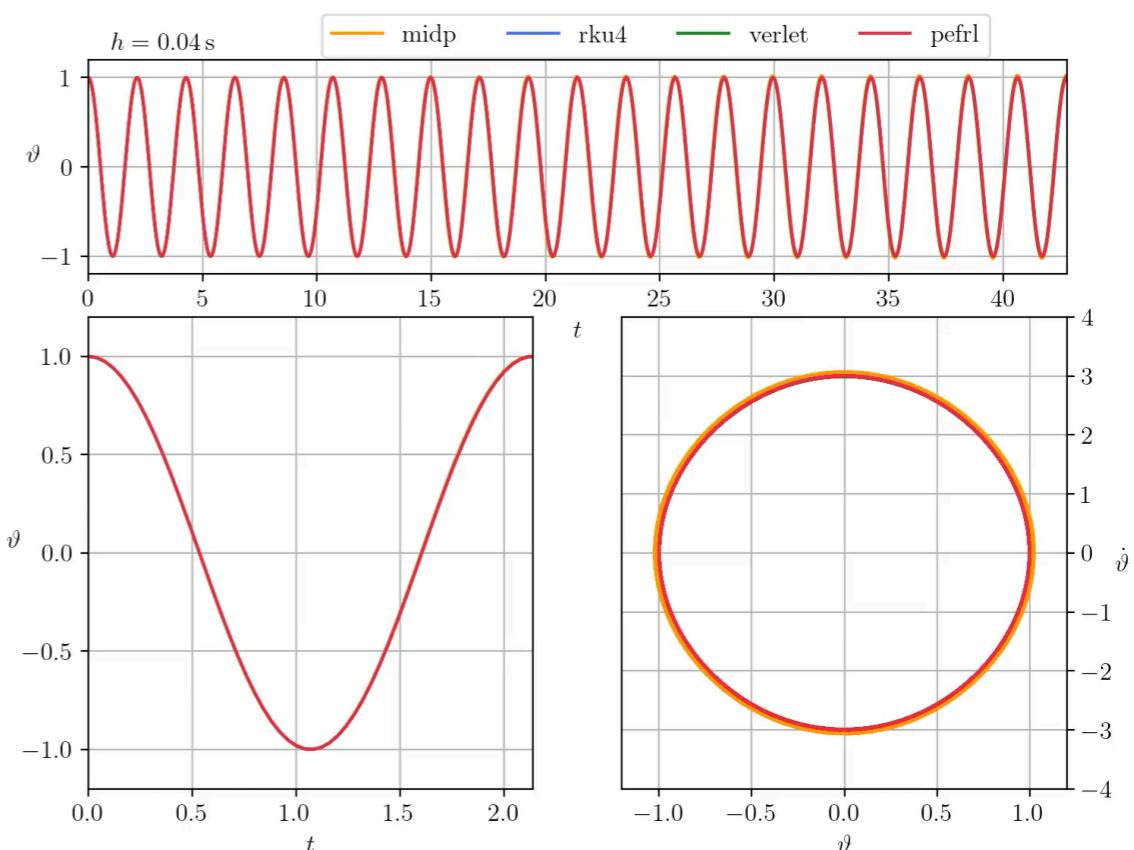
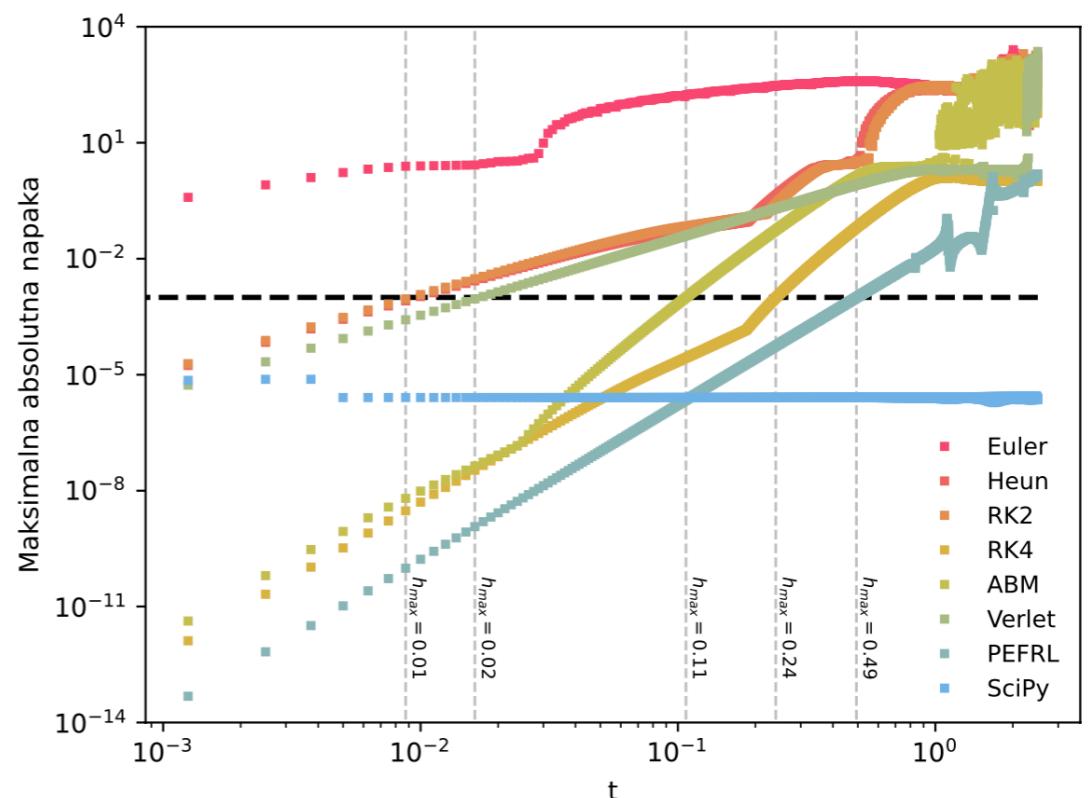
- Ocene natančnosti v različnih oblikah:
 - odvisnost od koraka...



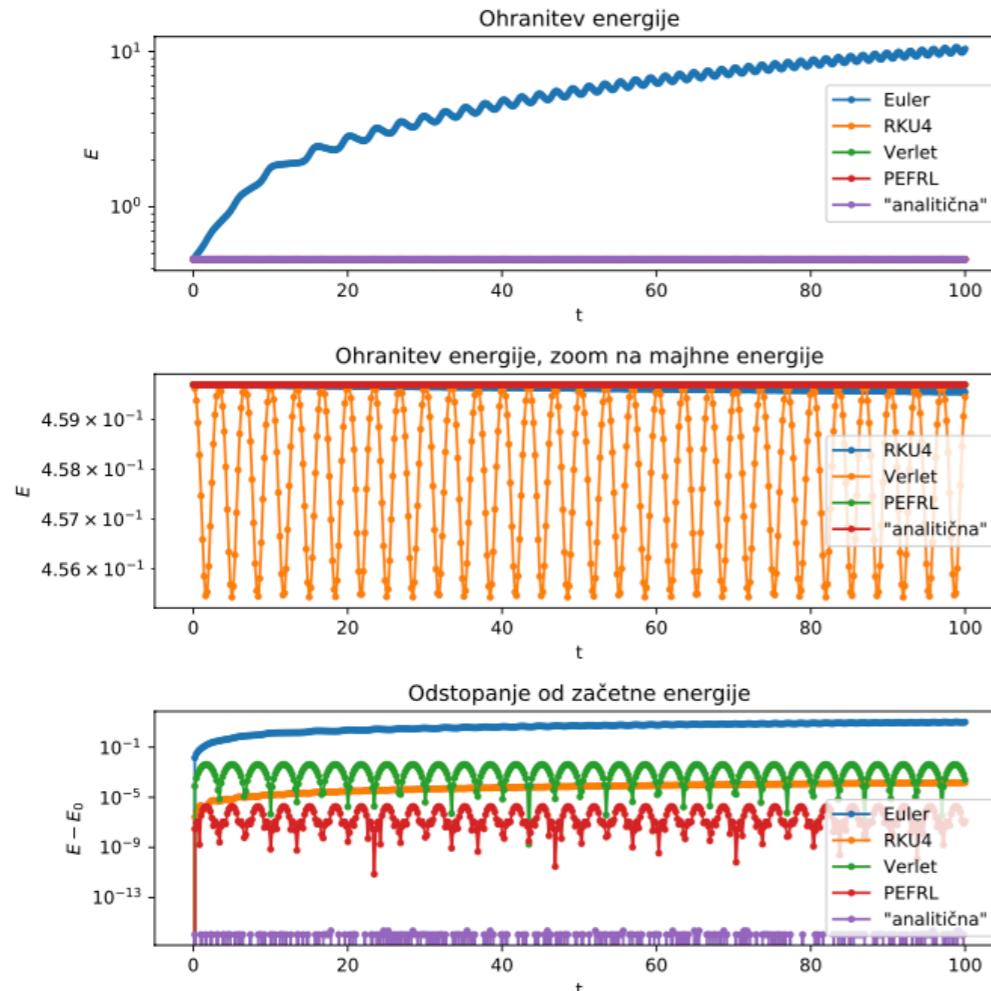
Slika 1: Rešitev problema s 50 in 500 koraki na intervalu.



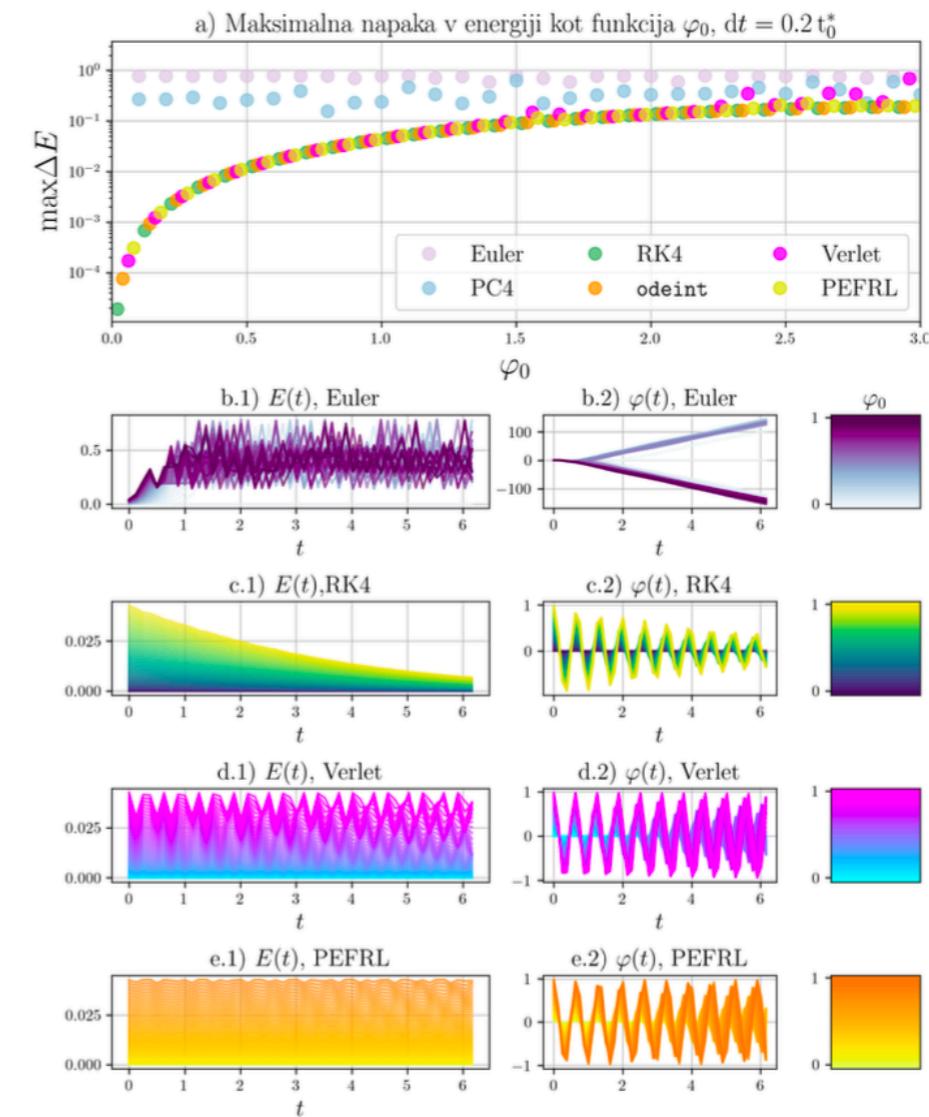
Slika 12: Maksimalne napake matod v odvisnosti od velikosti koraka



Študije napak in ohranitve energije



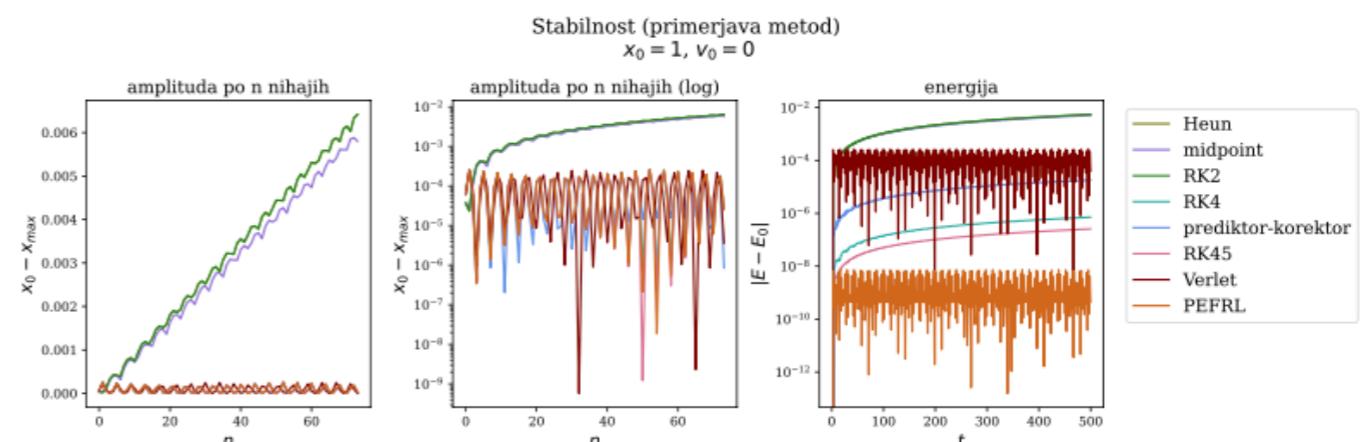
Slika 4: Ohranjanje začetne energije.



Slika 2: Pri fiksniem koraku $= 0.24t_0^*$ je za različne metode na grafu a) prikazana maksimalna napaka v energiji sistema kot funkcija φ_0 . Na spodnjih grafih pa so za metode Euler, RK4, Verlet in PEFRL časovni poteki energije in odmika pri taistih φ_0 .

Sedaj je potrebno še poiskati optimalno dolžino koraka za posamezno metodo. Uporabim bisekcijo in dobim:

Metoda	Dolžina koraka
Euler	0.000101
Midpoint metoda	0.0125
Runge-Kutta 4. reda	0.137
Verletova 2. reda	0.0237
Position Extended Forest-Ruth Like 4th order symplectic method	0.291

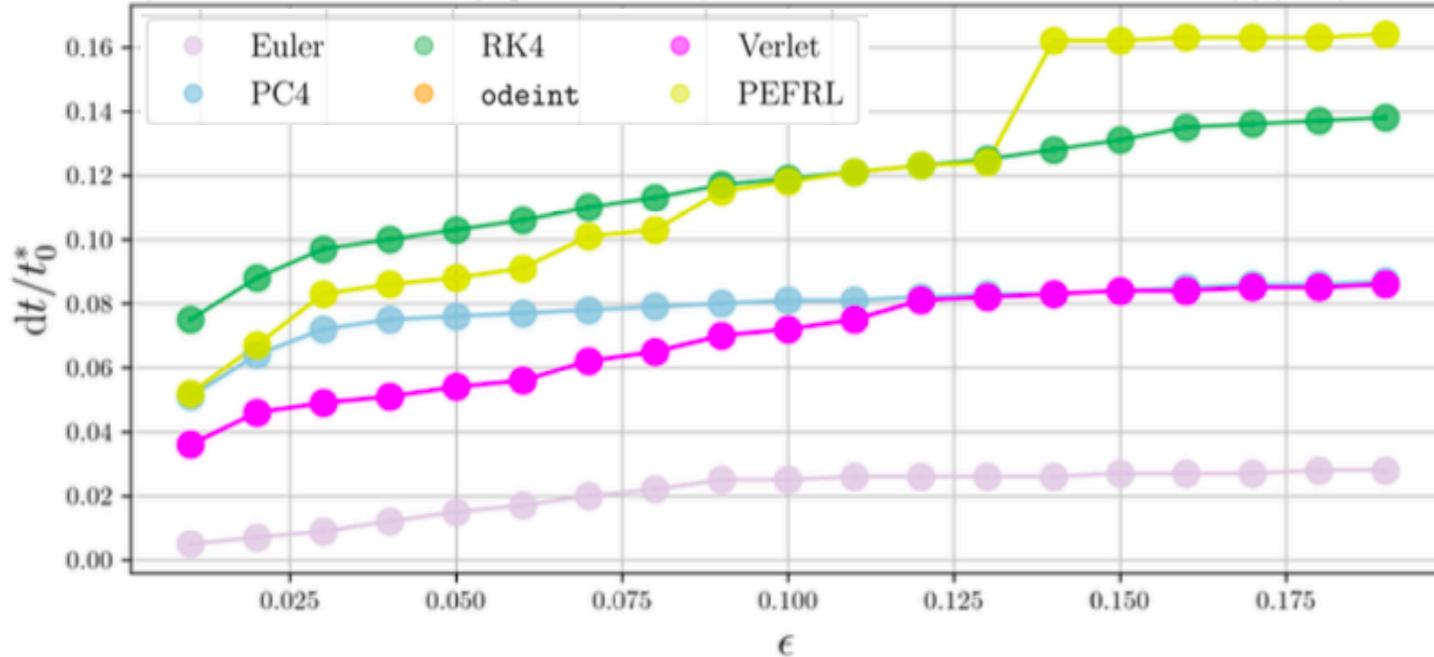




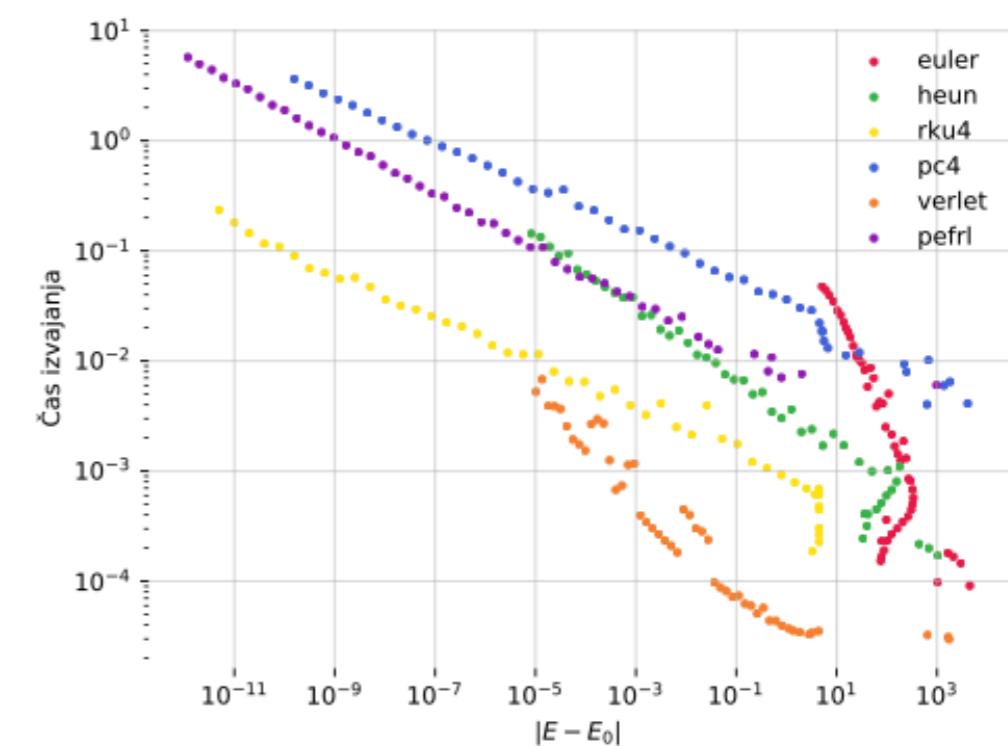
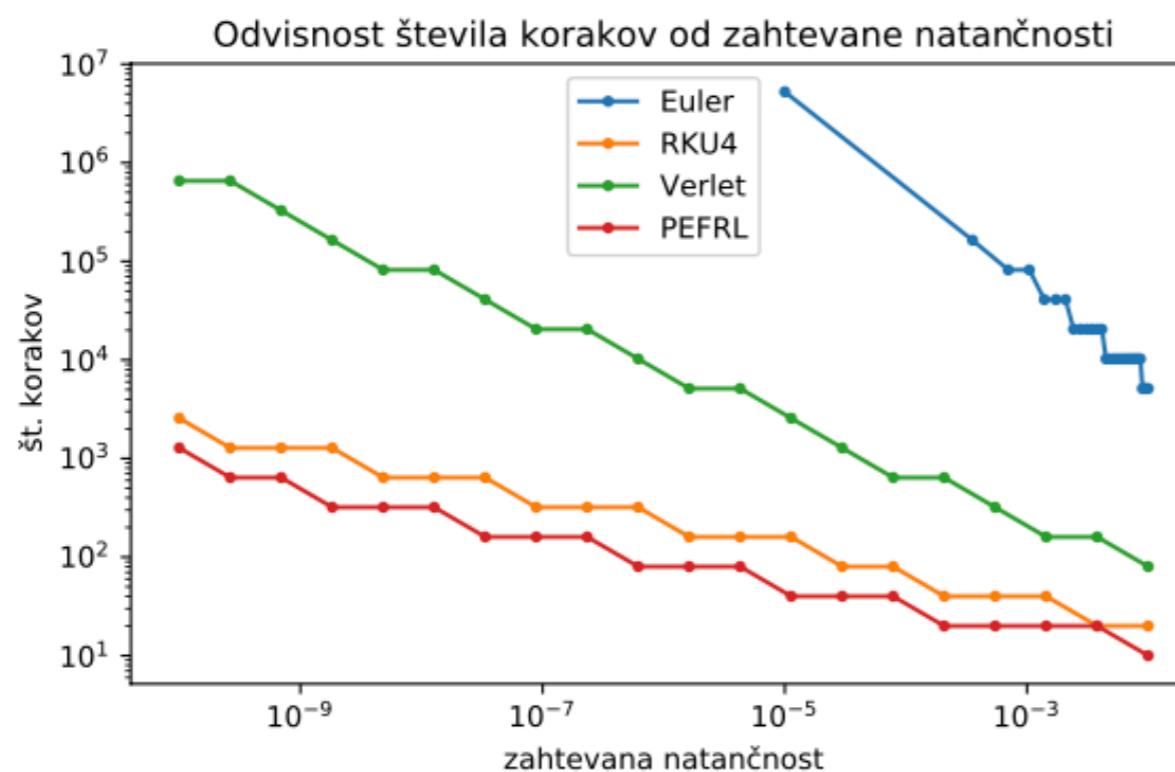
Grafi ‘tretje generacije’

BEST-OF-THE-DAY

c) Potreben korak dt/t_0^* kot funkcija želene natančnosti $\epsilon = \max \Delta\varphi / \varphi_0$, $\varphi_0 = 1$



(Verlet, PEFRL) sta na grafih a) in b). Z grafa c) je vidno, kolikšen korak dt moramo vzeti pri posamezni metodi za predpisano natančnost $\epsilon = \max \Delta\varphi / \varphi$ na $0 < t < 10 t_0^*$.



Slika 5: Časovna zahtevnost metod

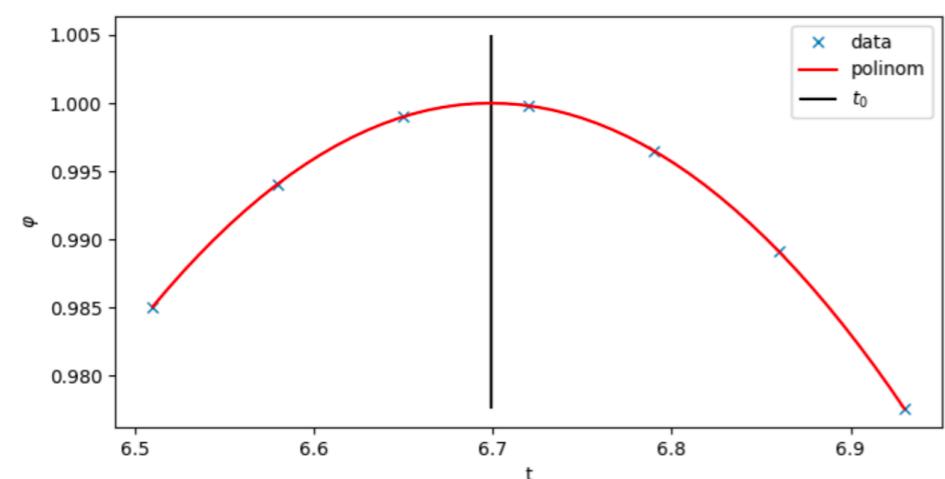


Nihajni čas

- kot dobra mera za natančnost služi tudi merjenje nihajnega časa med dvema točkama v fazi (lega=0 ipd..).

Lani...

Kot lahko vidimo se nihajni čas z povečanjem amplitude podaljša, kar nam pokaže tudi eksaktna formula za izračun nihajnega časa. Numeričen izračun nihajnega časa sem opravil s pomočjo interpolacije. Ker sem imel na voljo le numerično rešitev v določenih točkah, sem nihajni čas določil tako, da sem poiskal drugi lokalni maksimum funkcije in izbral nekaj točk z evaluirano rešitvijo v njegovi okolini. Nato sem na njih izvedel interpolacijo kvadratnega polinoma in iz koeficientov izračunal, ob katerem času nihalo spet zopet preide začetno lego, kar nam da formulo $t_0 = -b/2a$, kjer sta a in b koeficiente polinoma druge stopnje $p(x) = ax^2 + bx + c$. Predpostavljal sem, da bi moral ta postopek dokaj natančno izračunati nihajni čas nihala.



Slika 2: Določitev nihajnega časa matematičnega nihala z interpolacijo rešitve v določenih točkah.
Črna črta predstavlja maksimum nihala.

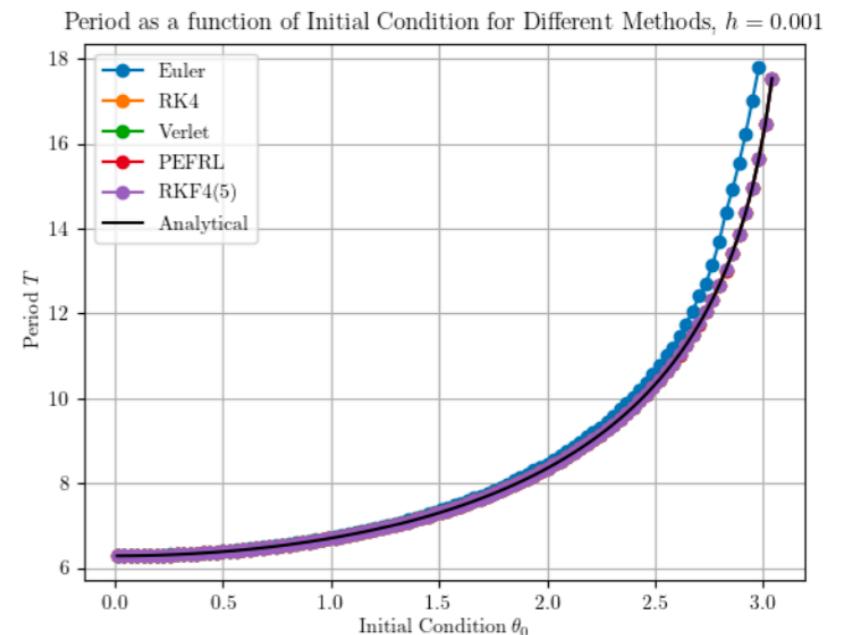
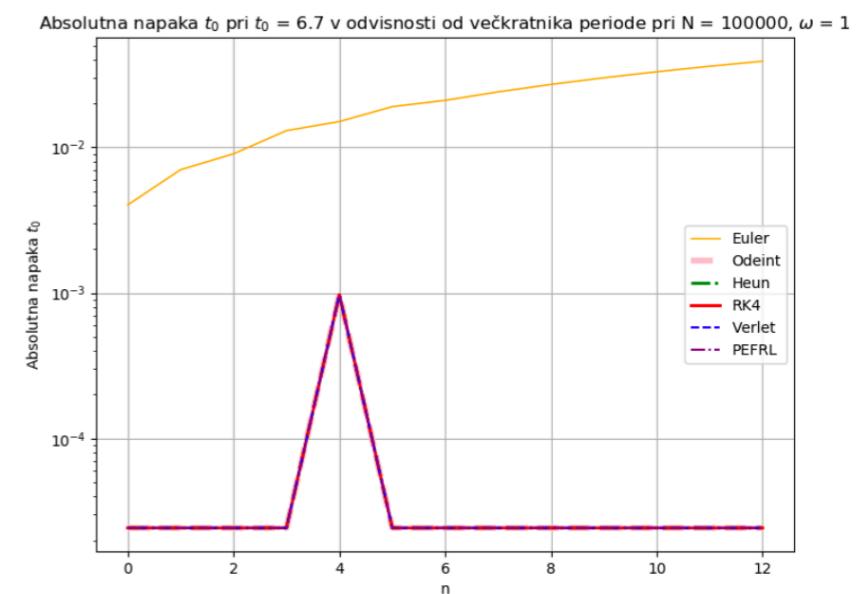
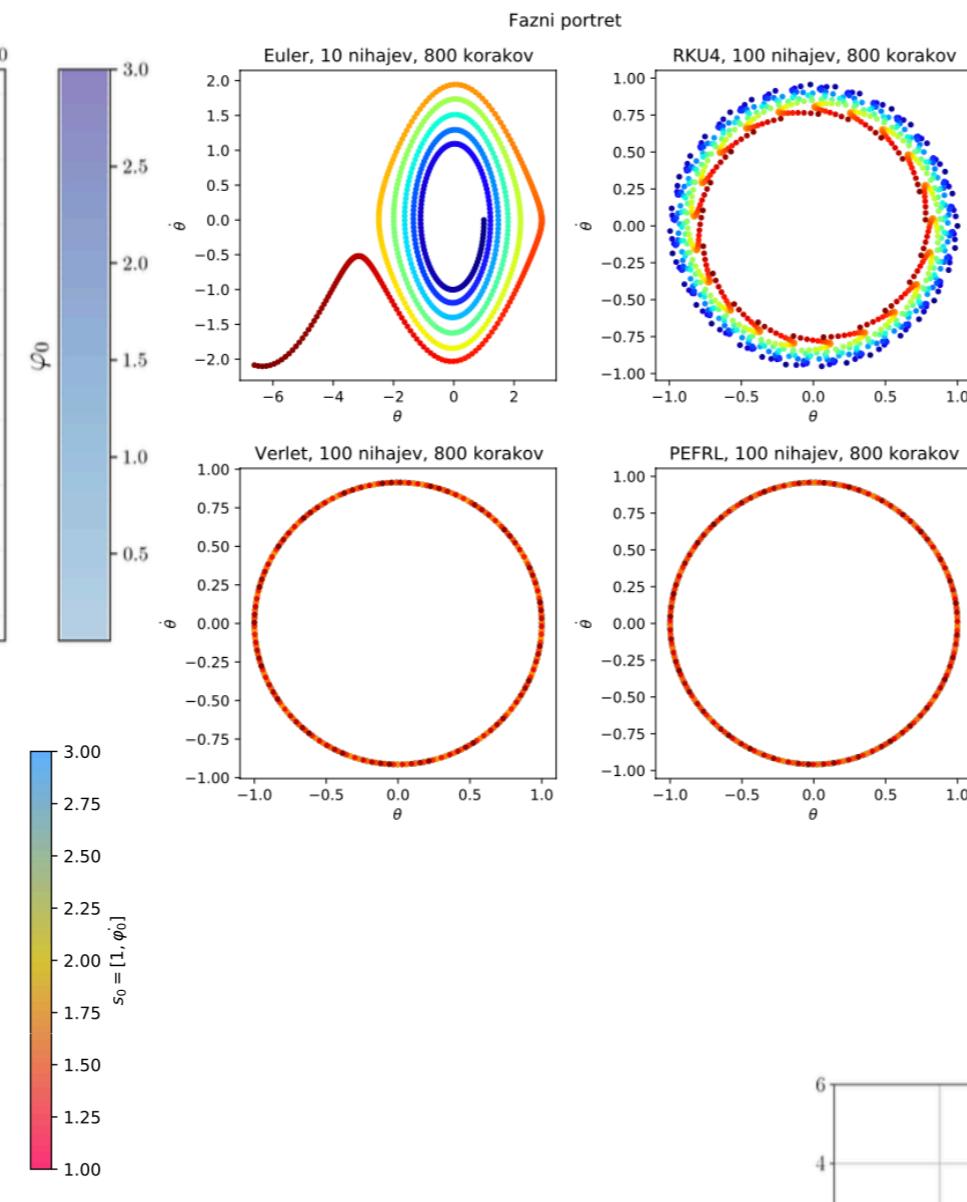
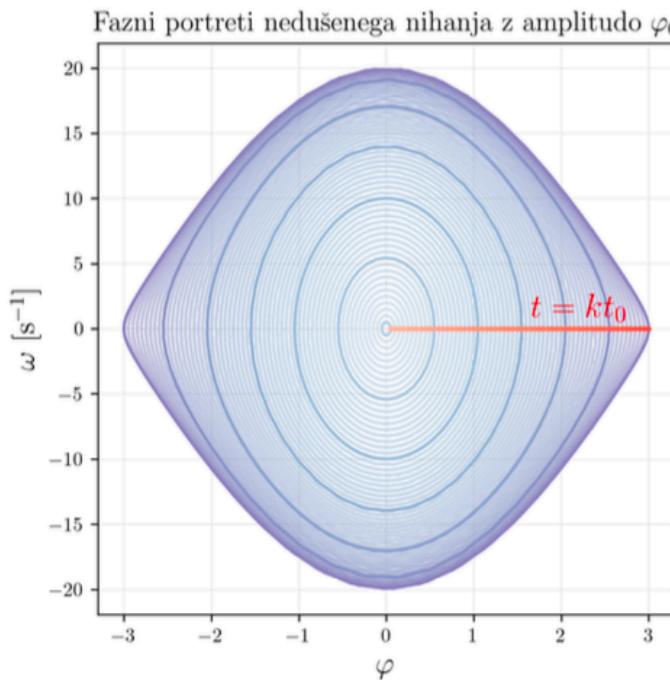


Figure 13: Period as a function of initial condition θ_0 .

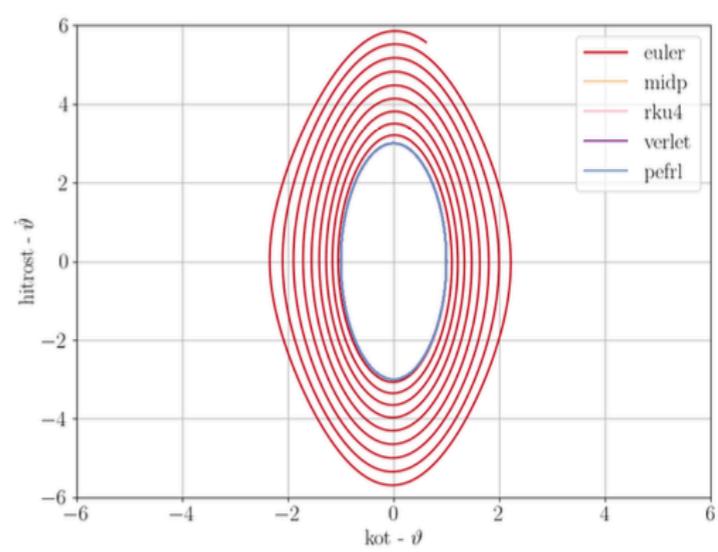
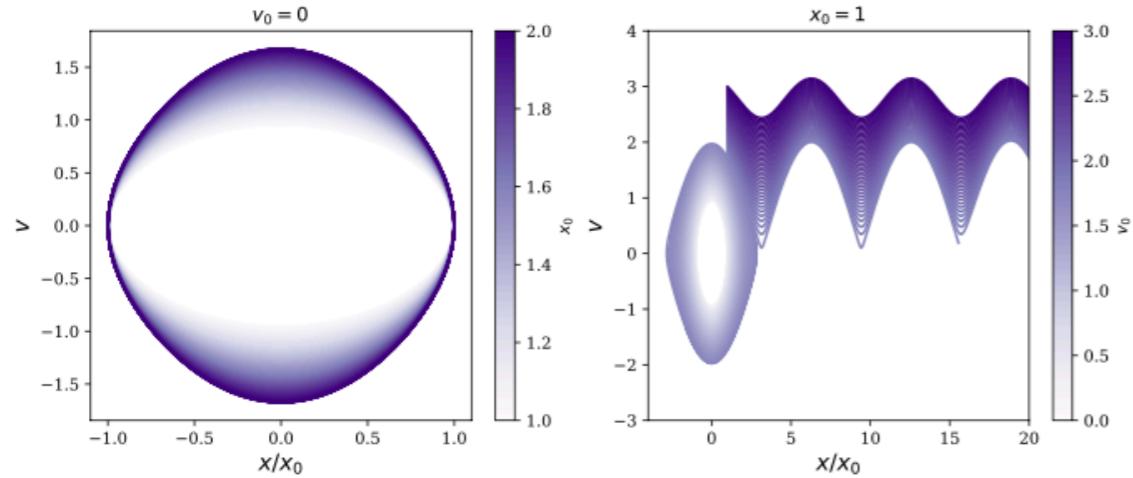


Slika 8: Absolutna napaka nihanjega časa glede na teoretično vrednost

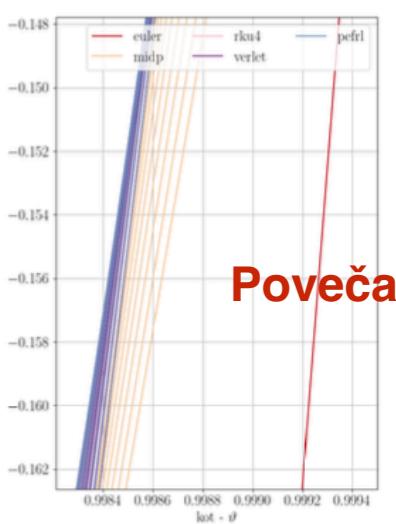
Fazni portreti



Slika 8: Fazni portreti v odvisnosti od začetne hitrosti.

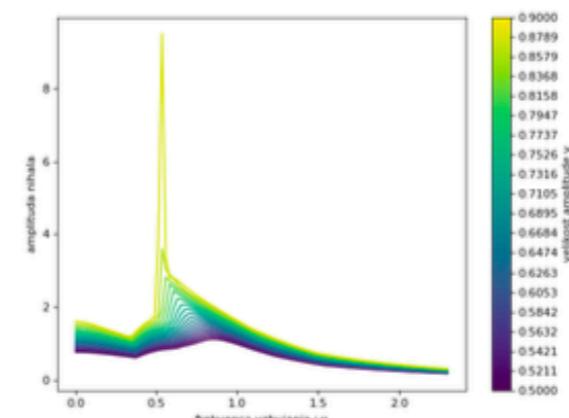
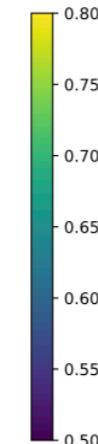
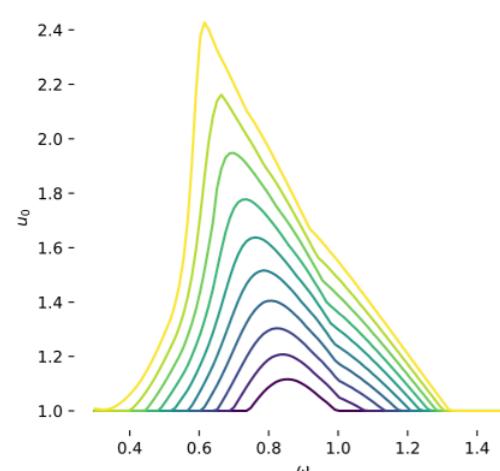
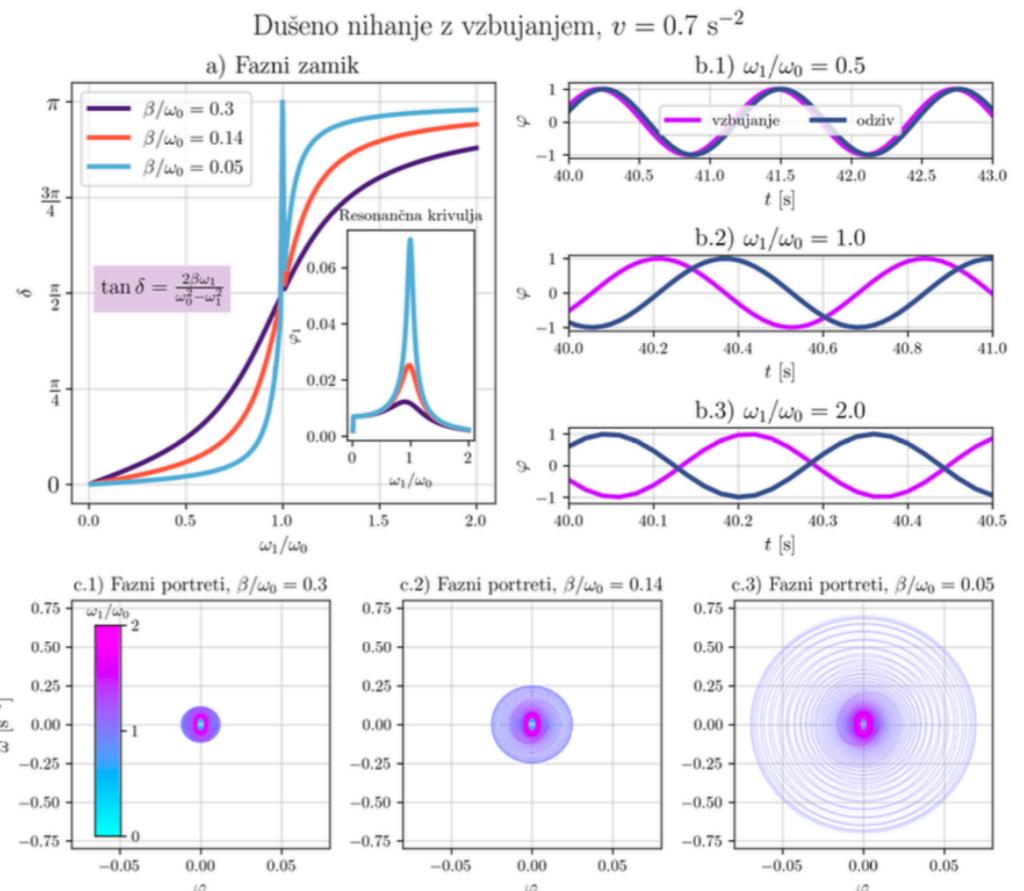
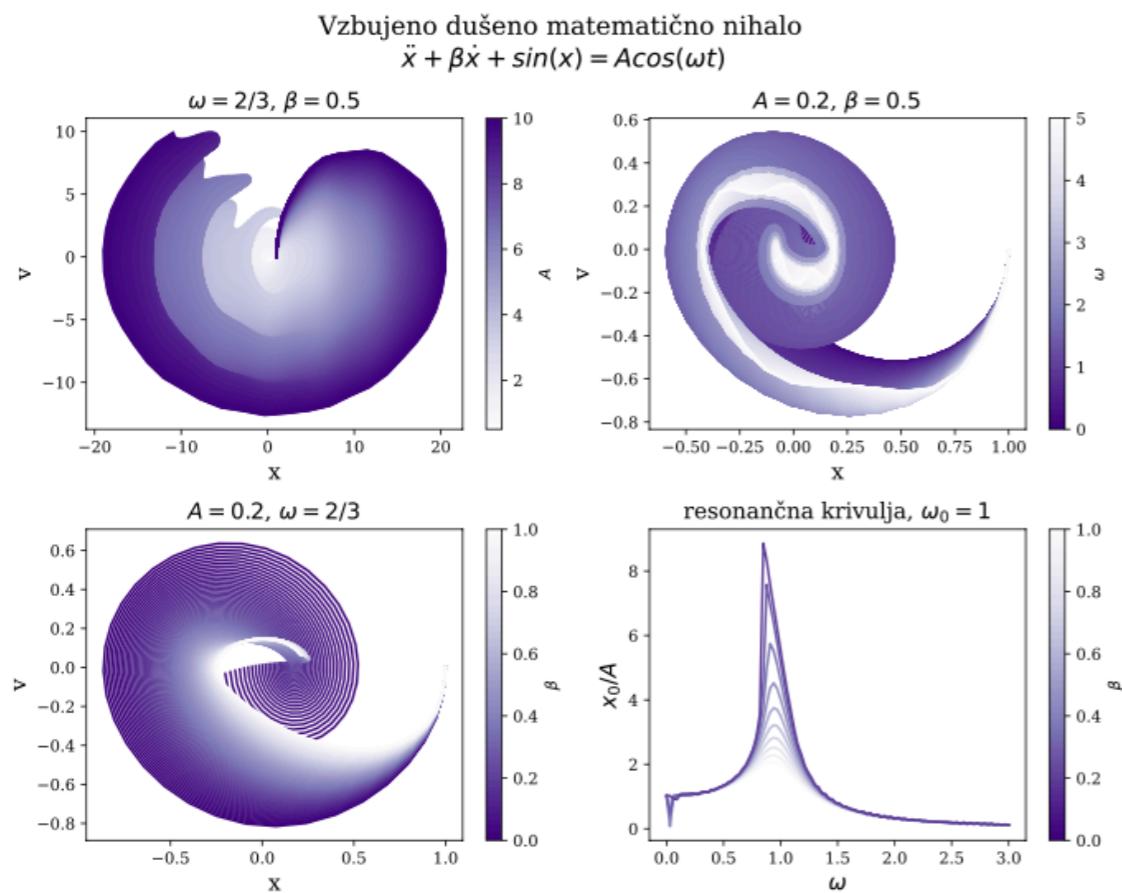


(a) Fazni portret nihala - vrednosti so si dovolj blizu, da se prekrivajo

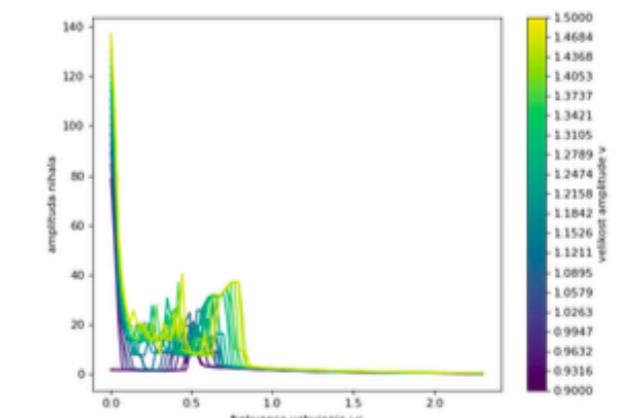


(b) Pri približanju vidimo, da se vrednosti v posameznih krogih tudi razlikujejo med sabo, čeprav gre za isto metodo

Resonance...



(a) $v=[0.5, 0.9]$

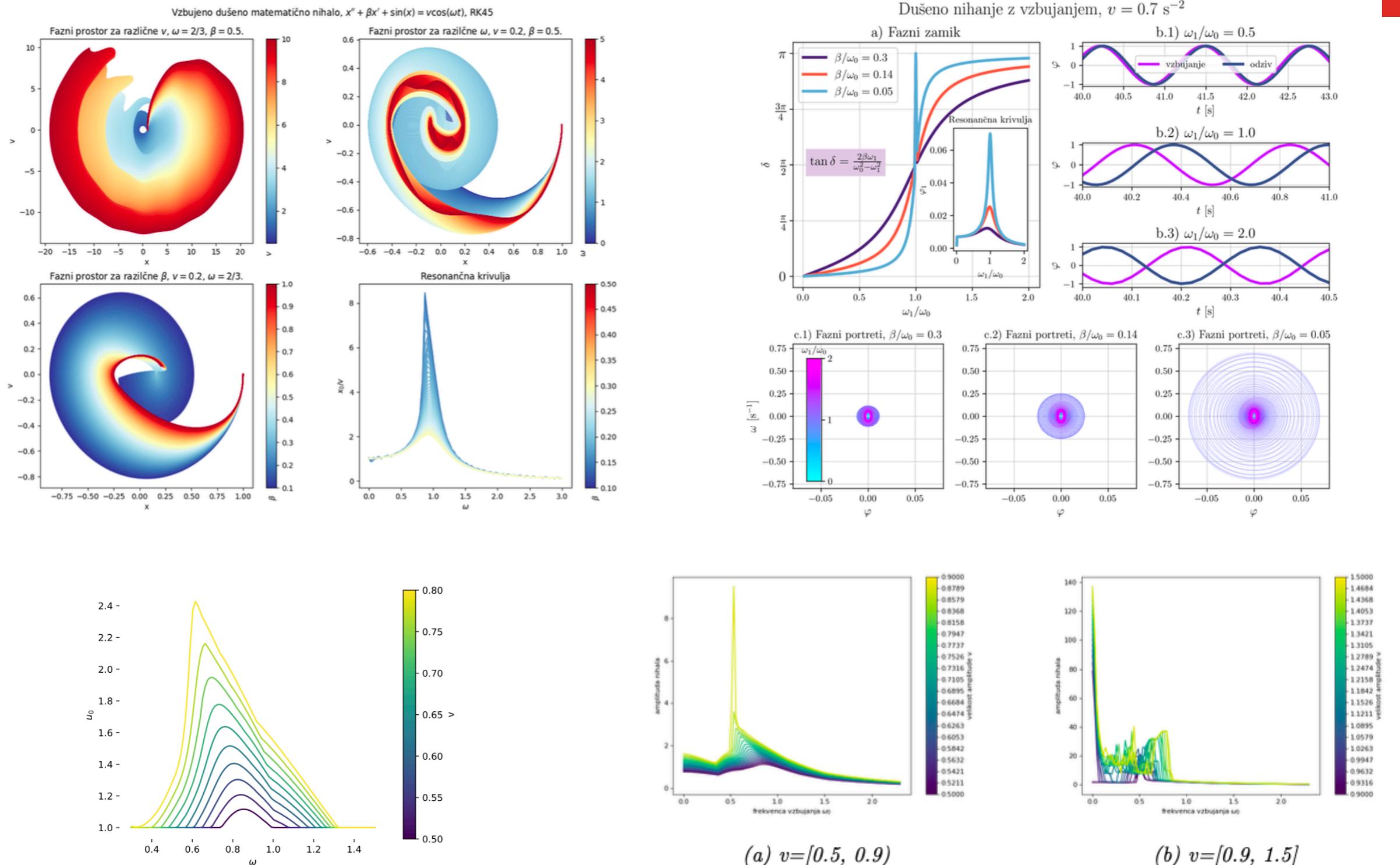


(b) $v=[0.9, 1.5]$

Slika 7: Resonančna krivulja dušenega nihala z različnimi amplitudami vzbujanja

Slika 10: Resonančna krivulja za različne parametre amplitute v .

Resonance...



Slika 7: Resonančna krivulja dušenega nihala z različnimi amplitudami vzbujanja

Slika 10: Resonančna krivulja za različne parametre amplitute v .

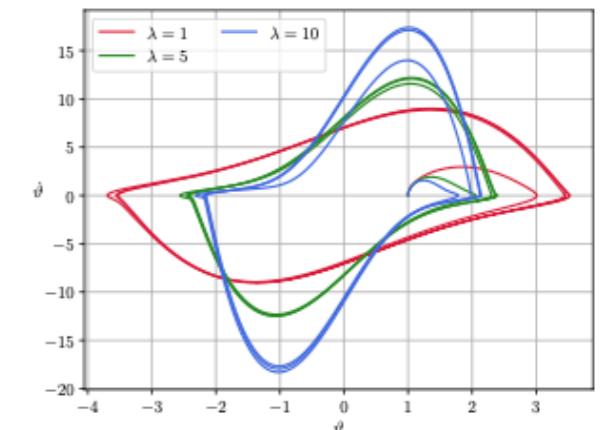
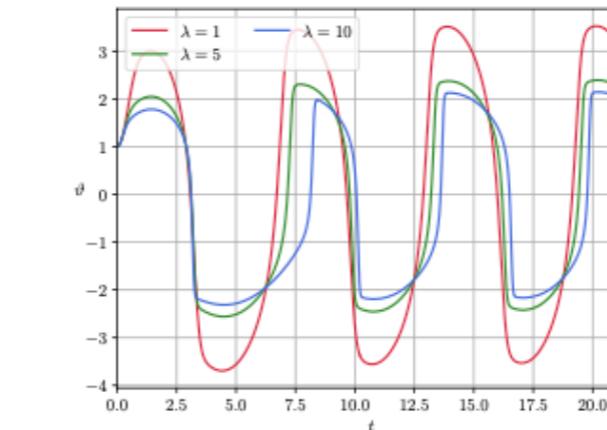
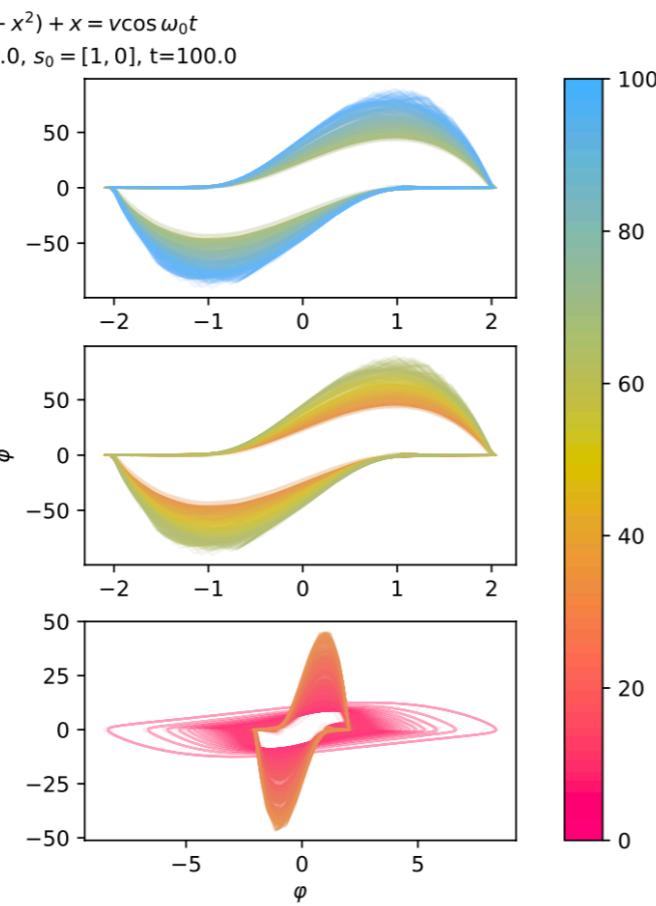
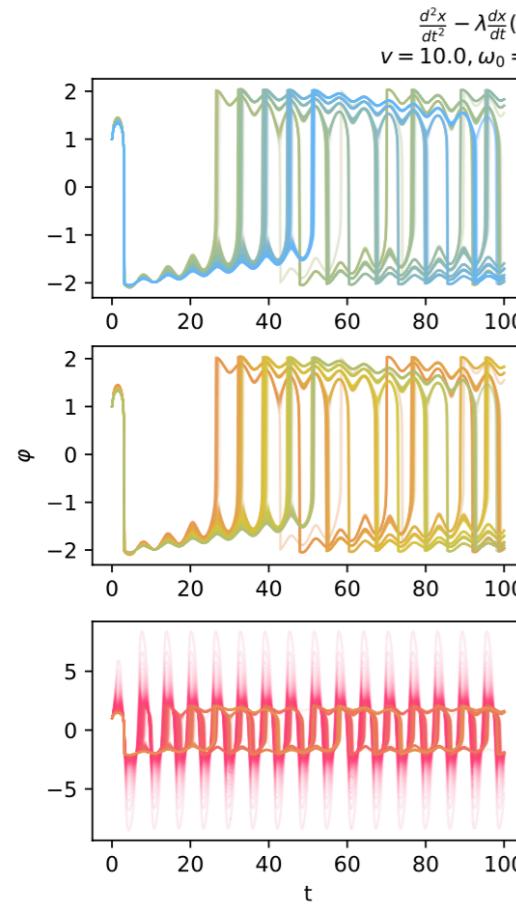


Van der Polov oscilator

BRAVO!

- Lepi primeri faznih portretov...
- Narišite še časovni potek!

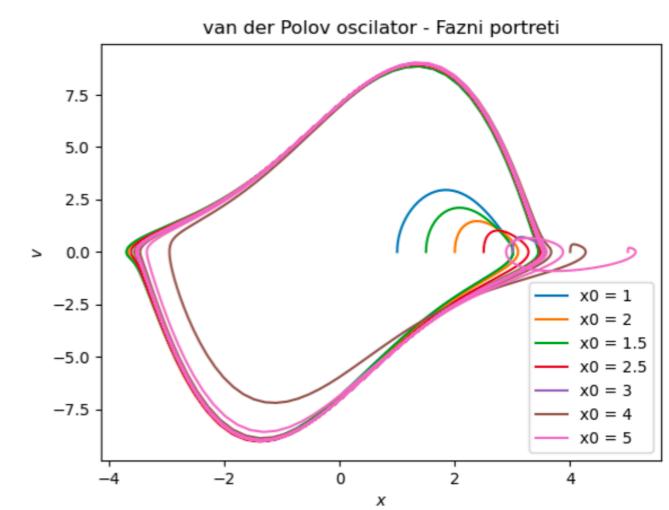
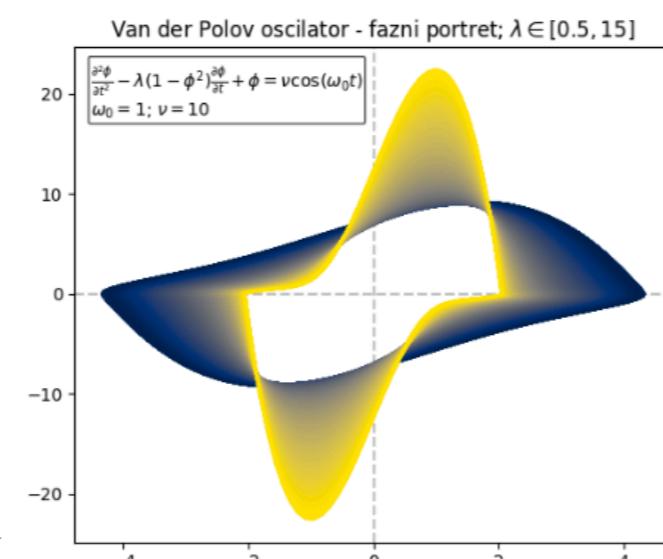
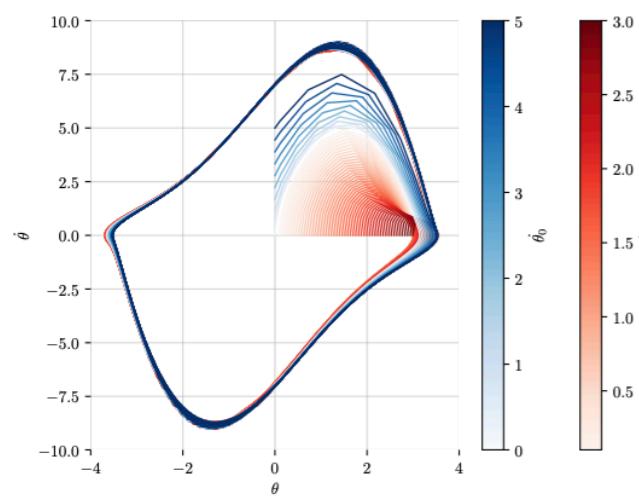
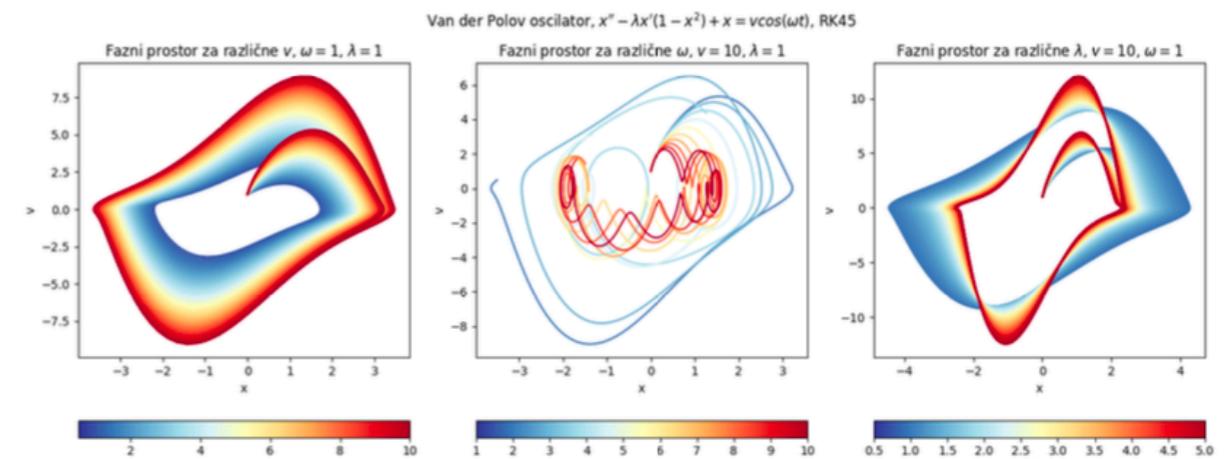
$$\frac{d^2x}{dt^2} - \lambda \frac{dx}{dt} (1 - x^2) + x = v \cos \omega_0 t.$$



(a) Časovna odvisnost kota

(b) Fazni portret

Slika 6: Van der Polov oscilator



Slika 9: Fazni portret van der Polovega oscilatorja pri $\omega_0 = 1, v = 10, \lambda = 1$

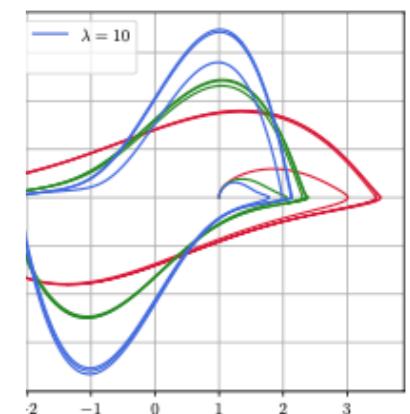
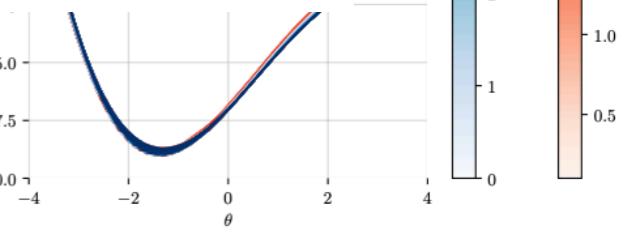
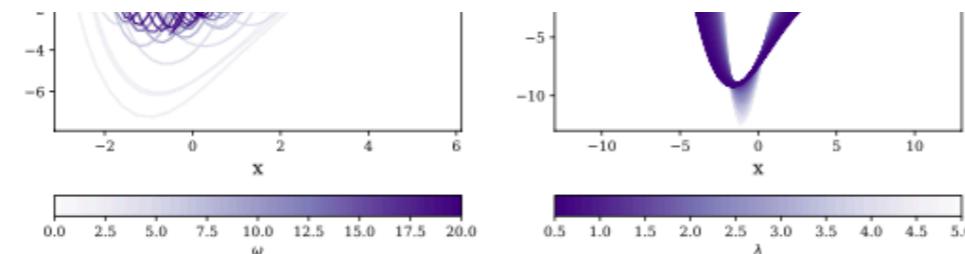
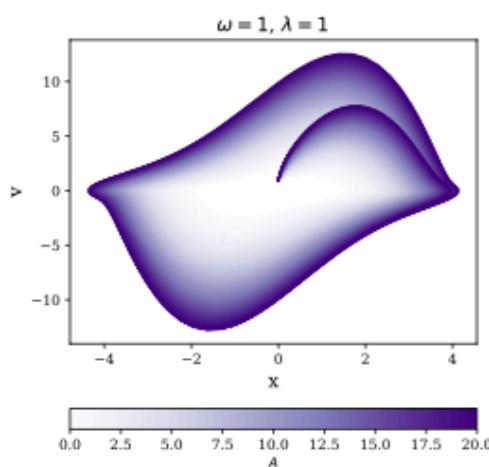
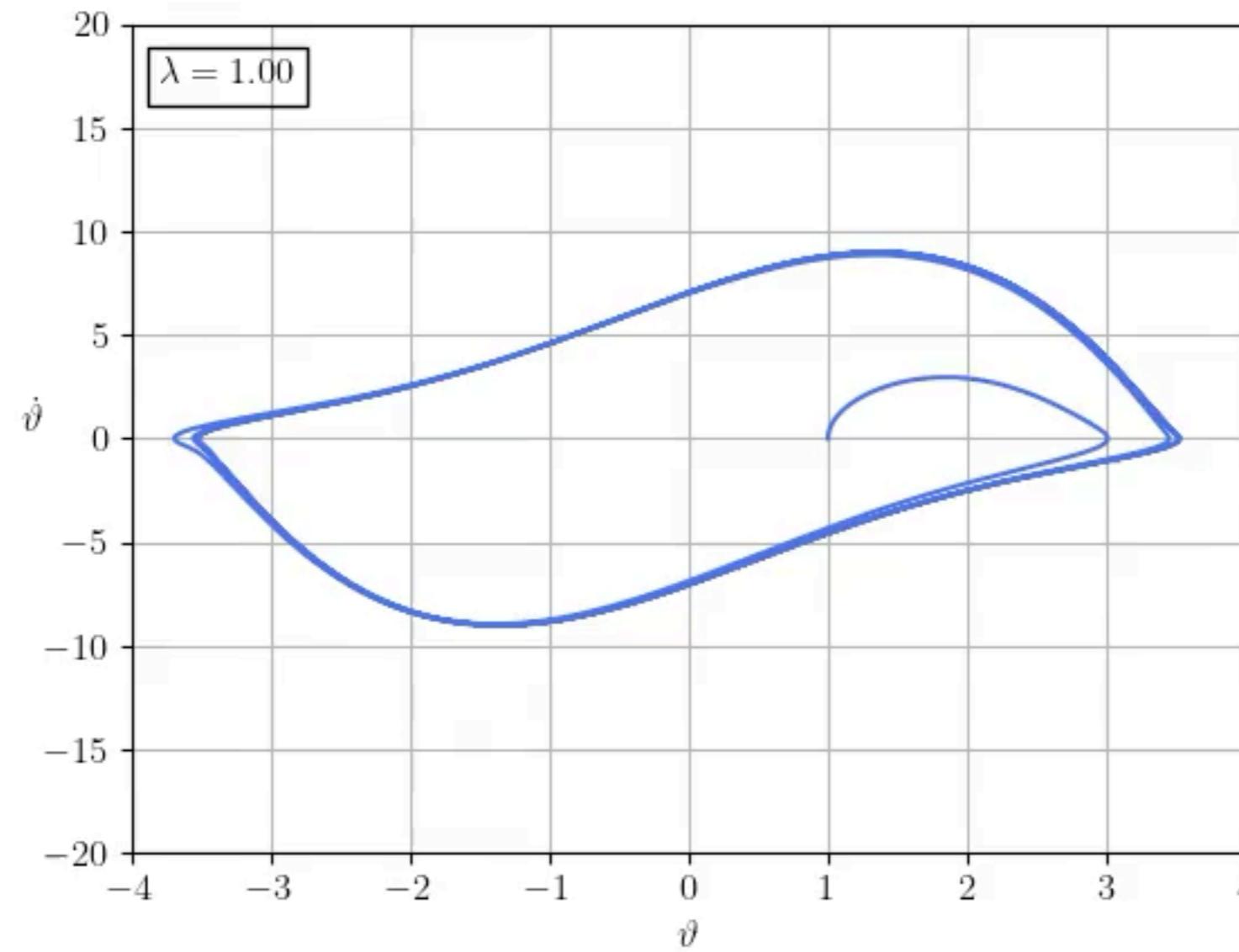
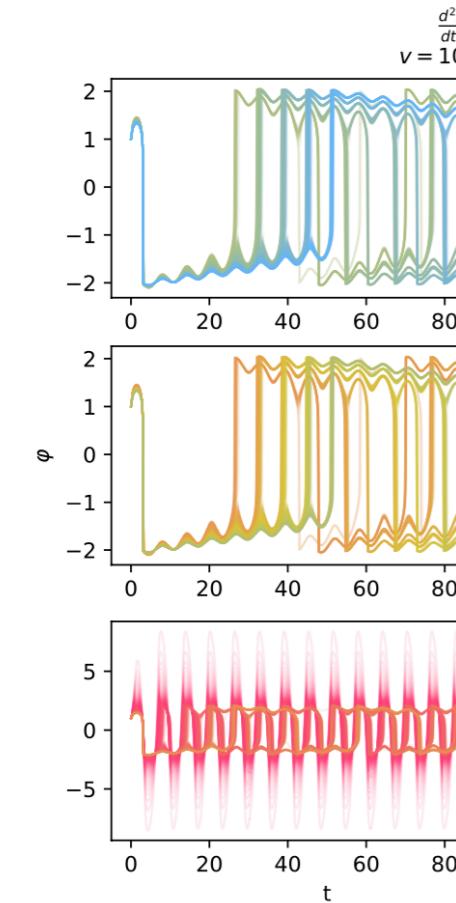


Van der Polov oscilator

BRAVO!

- Lepi primeri faznih portretov...
 - Narišite še časovni potek!

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



b) Fazni portret
tor

Slika 9: Fazni portret van der Polovega oscilatorja pri $\omega_0 = 1, v = 10, \lambda = 1$

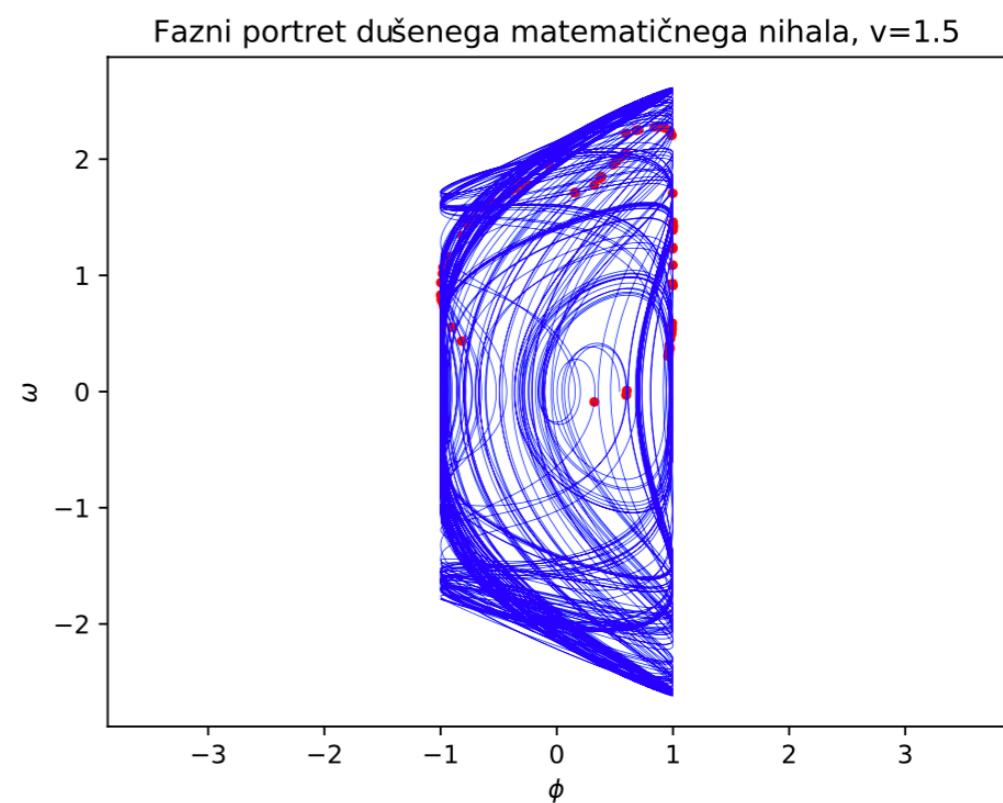
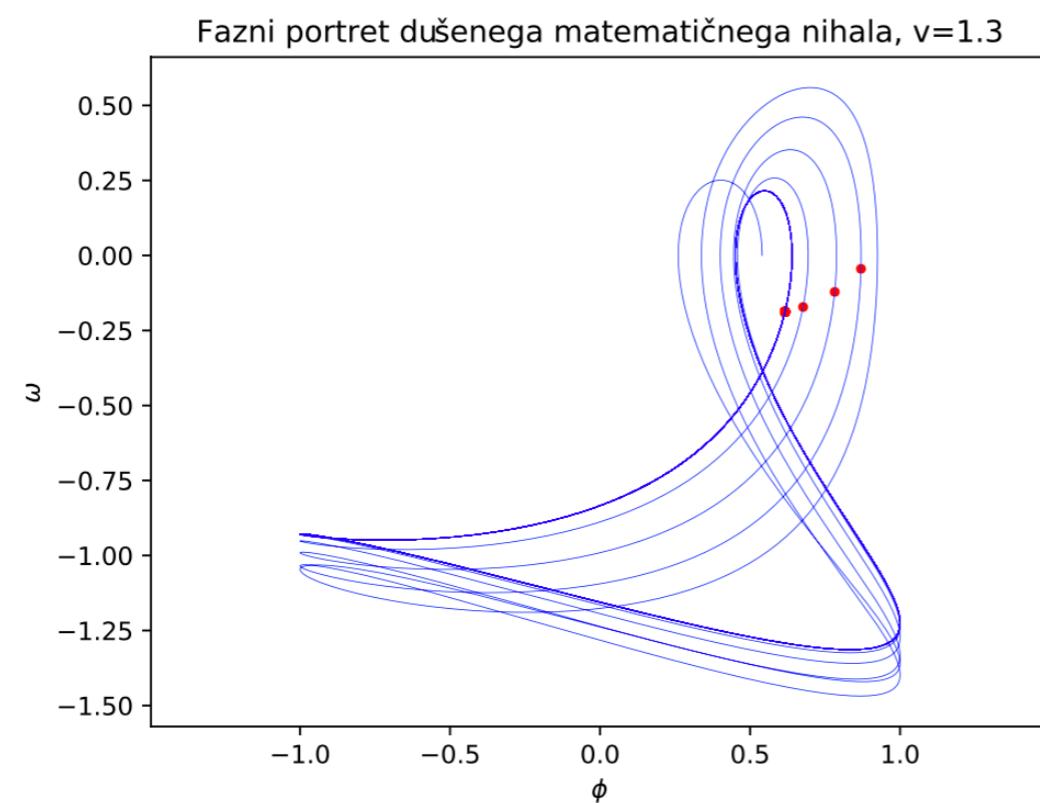
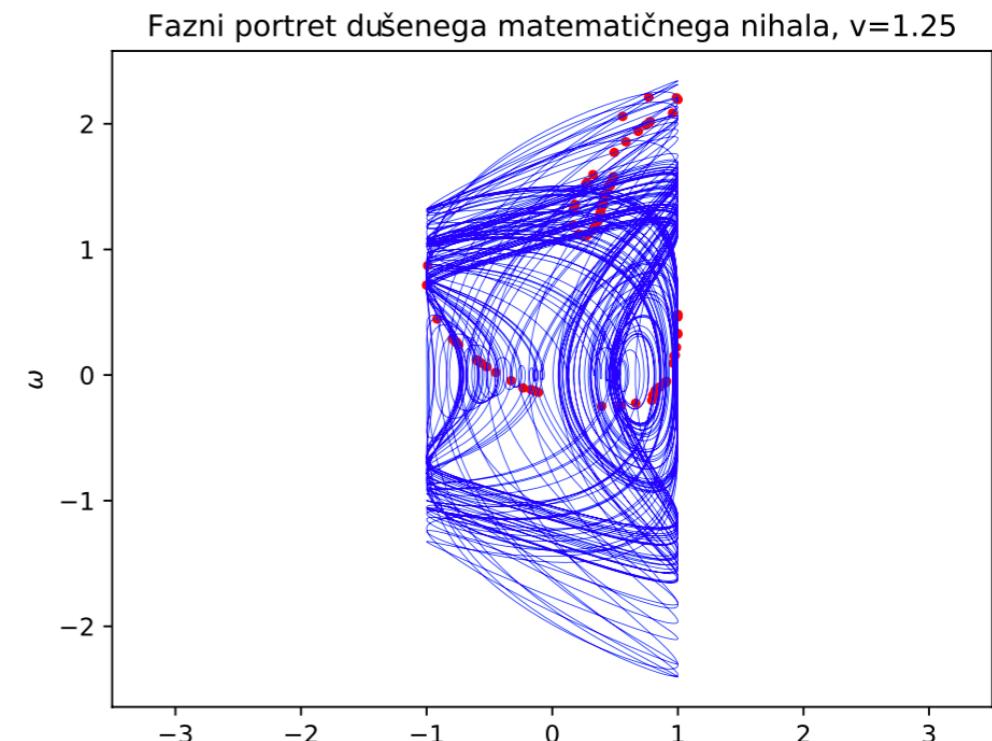
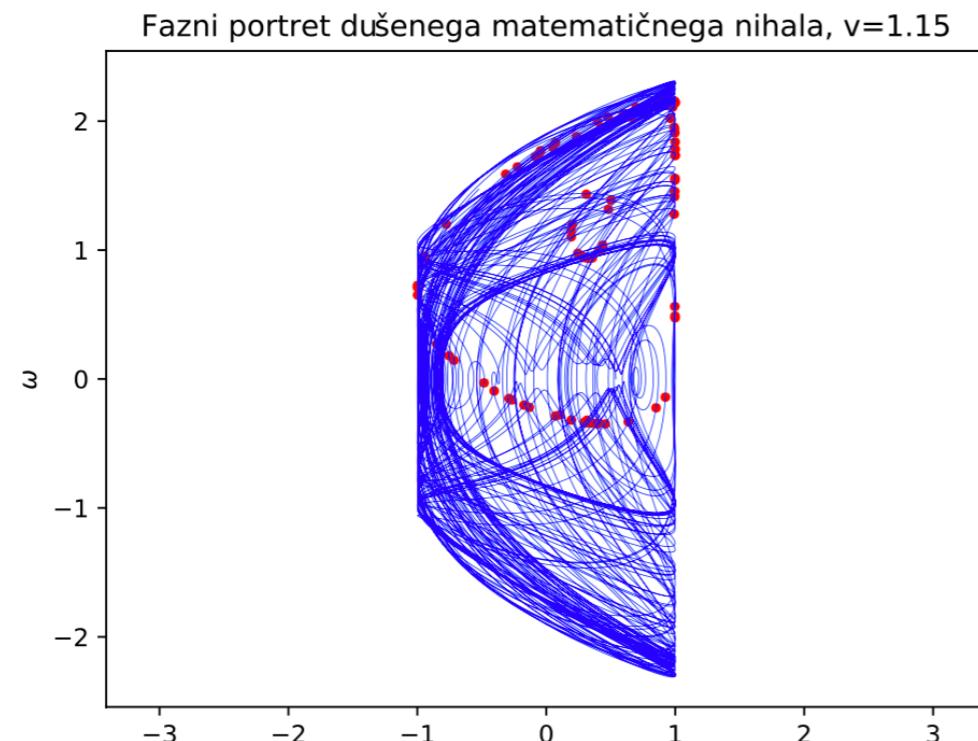


Na poti h kaosu...

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

Slika 9: Fazni portreti pri različnih jakostih vzbujanja s Pointcarejevimi preseki - rdeče točke

Lani...

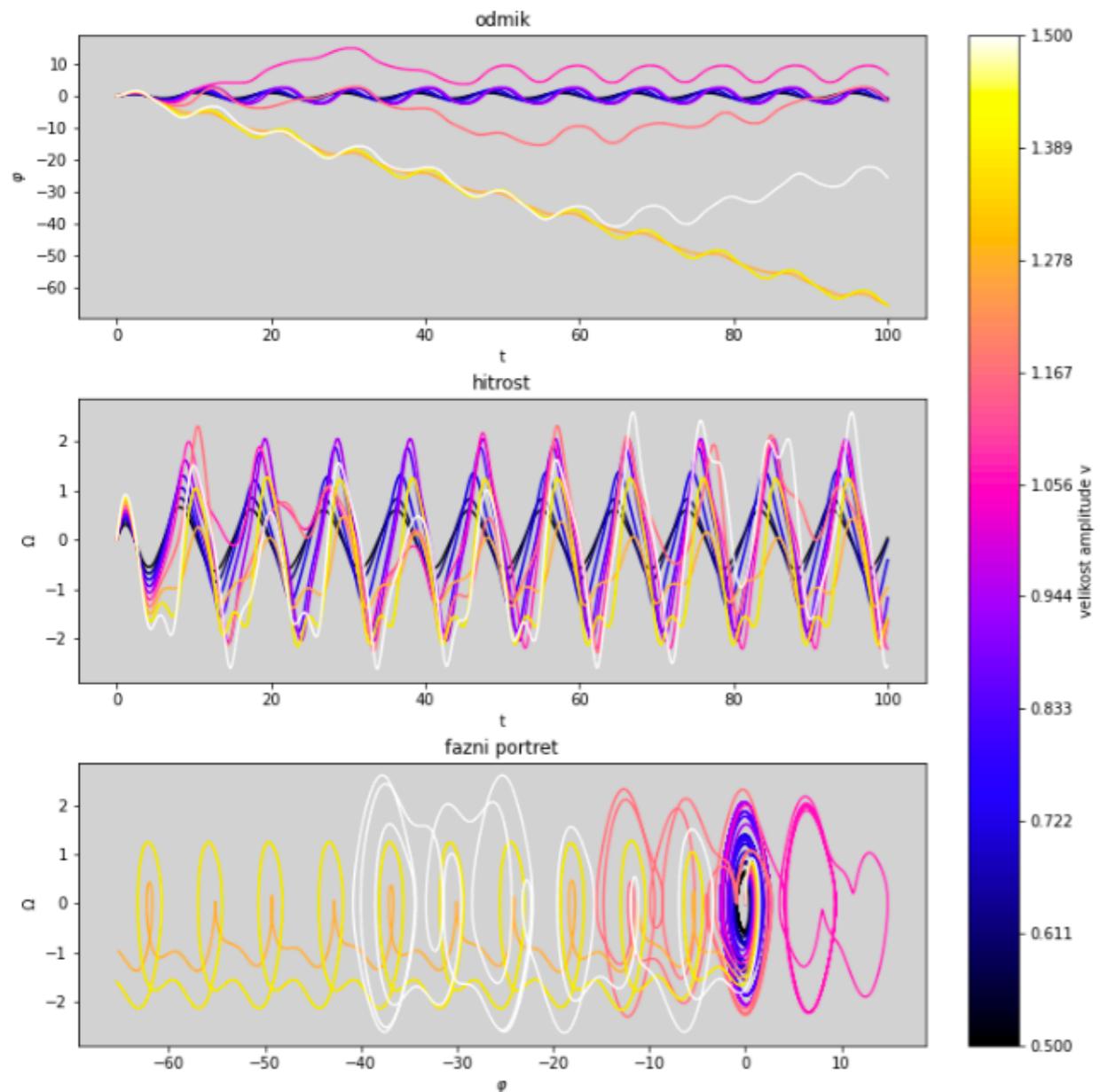




Na poti h kaosu...

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

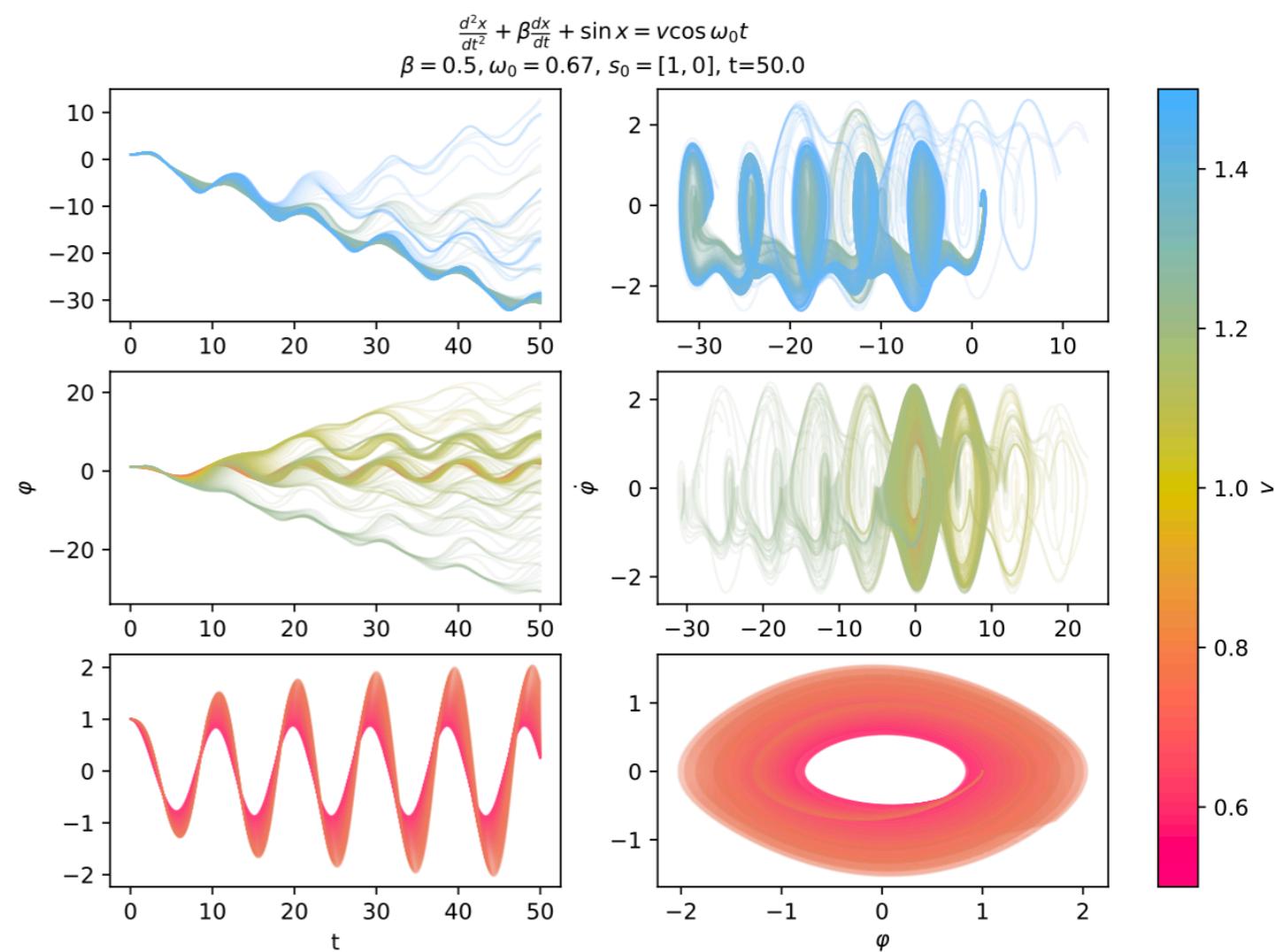
Bravo!



Slika 11: Odmiki, hitrost in fazni portret za vzbujeno nihanje. Iz grafov je moč razbrati, da je metoda stabilna za korake pod približno 0.5 - takrat tako odmiki kot hitrost uniformno nihajo okoli začetnih leg.

100000.

Nato sem se lotil dodatnih dveh primerov. Najprej sem sliki 9 izrisal fazne portrete dušenega nihala. Razdelil sem jih na tri dele, da se lepše vidi pri katerih vrednostih v nihanje pobezlja.



Slika 9: Fazni portreti dušenega nihala.



Na poti h kaosu...

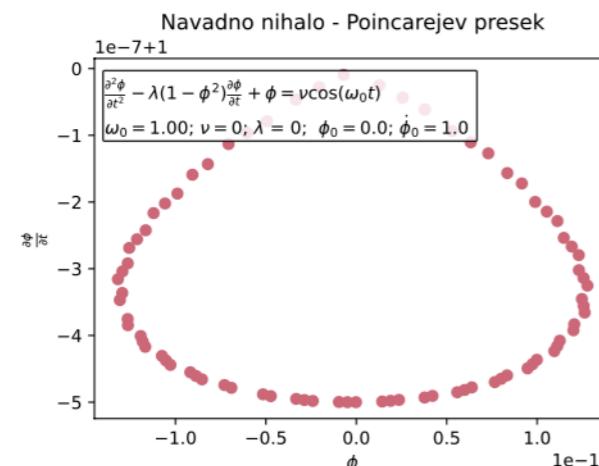
$$\frac{d^2x}{dt^2} + \beta \frac{dx}{dt} + \sin x = v \cos \omega_0 t,$$

Bravo!

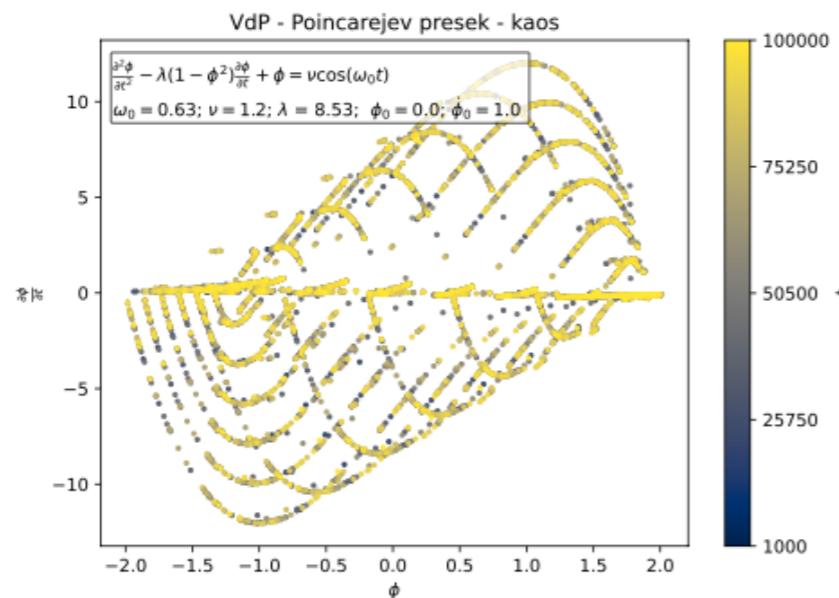
1.1.1 Kaotično obnašanje

Za nekatere vrednosti parametrov λ , v , ω_0 in določene začetne pogoje lahko pri VdP oscilatorju opazimo kaotično (neperiodično oz. več-periodično) obnašanje.

Primer tri-periodičnega osciliranja vidimo na grafih 3.

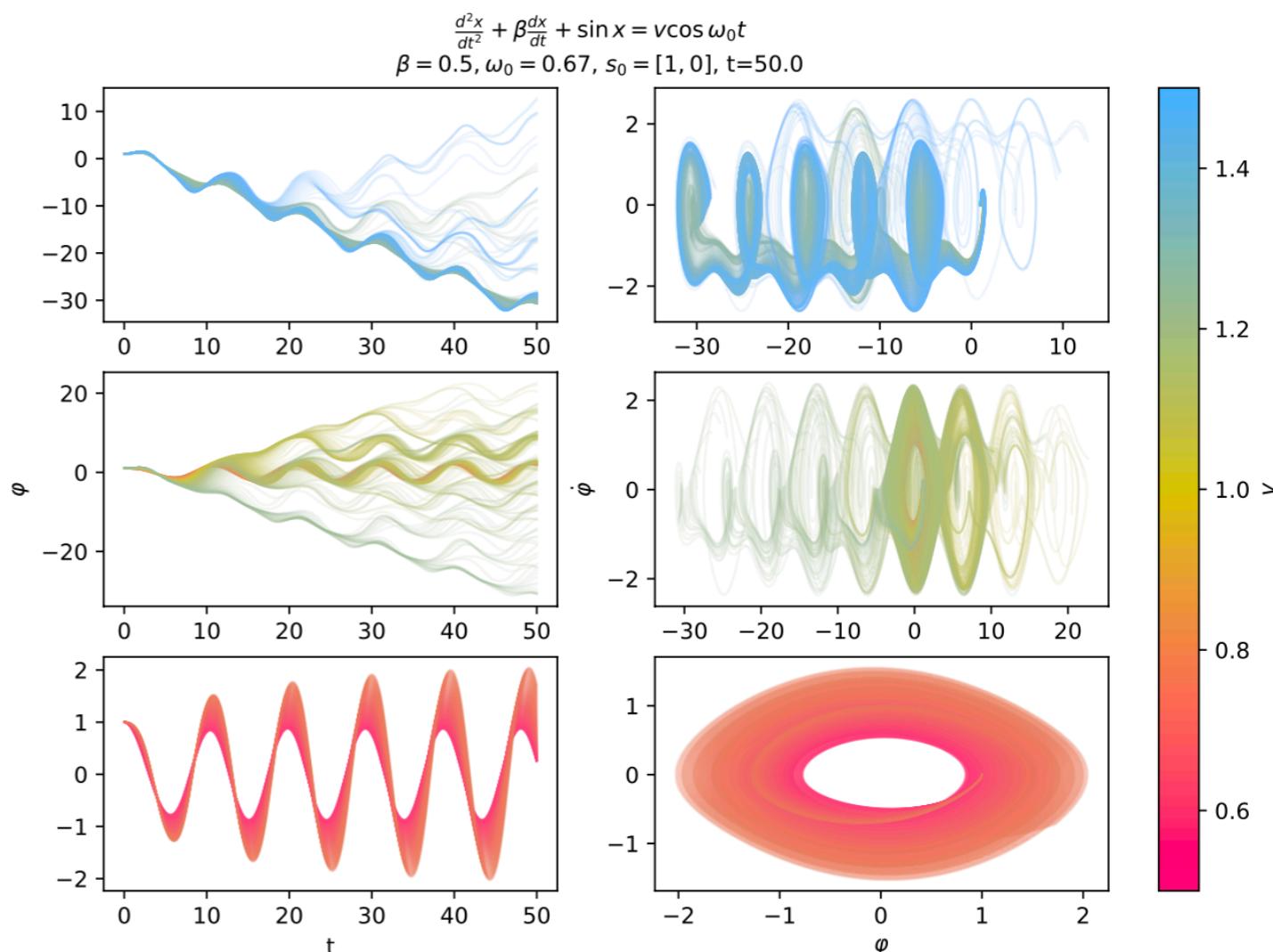


Slika 4: Poincarjev presek za navadno nihalo. Vse točke so zelo blizu ničle (SKALA!), torej gledamo bolj kot ne le numerično napako.



Slika 6: Poincarjev presek za VdP. Poincarjev presek je vzorčen z $T_0 = 2\pi/\omega_0$. Simulacija je tekla na intervalu časa $[0, 100000]$. Barva prikazuje časovno komponento - vidimo, da se točke po faznem prostoru rišejo bolj kot ne naključno/neodvisno; ne opazimo nobenega (časovnega) vzorca.

Nato sem se lotil dodatnih dveh primerov. Najprej sem sliki 9 izrisal fazne portrete dušenega nihala. Razdelil sem jih na tri dele, da se lepše vidi pri katerih vrednostih v nihanje pobezlja.



Slika 9: Fazni portreti dušenega nihala.



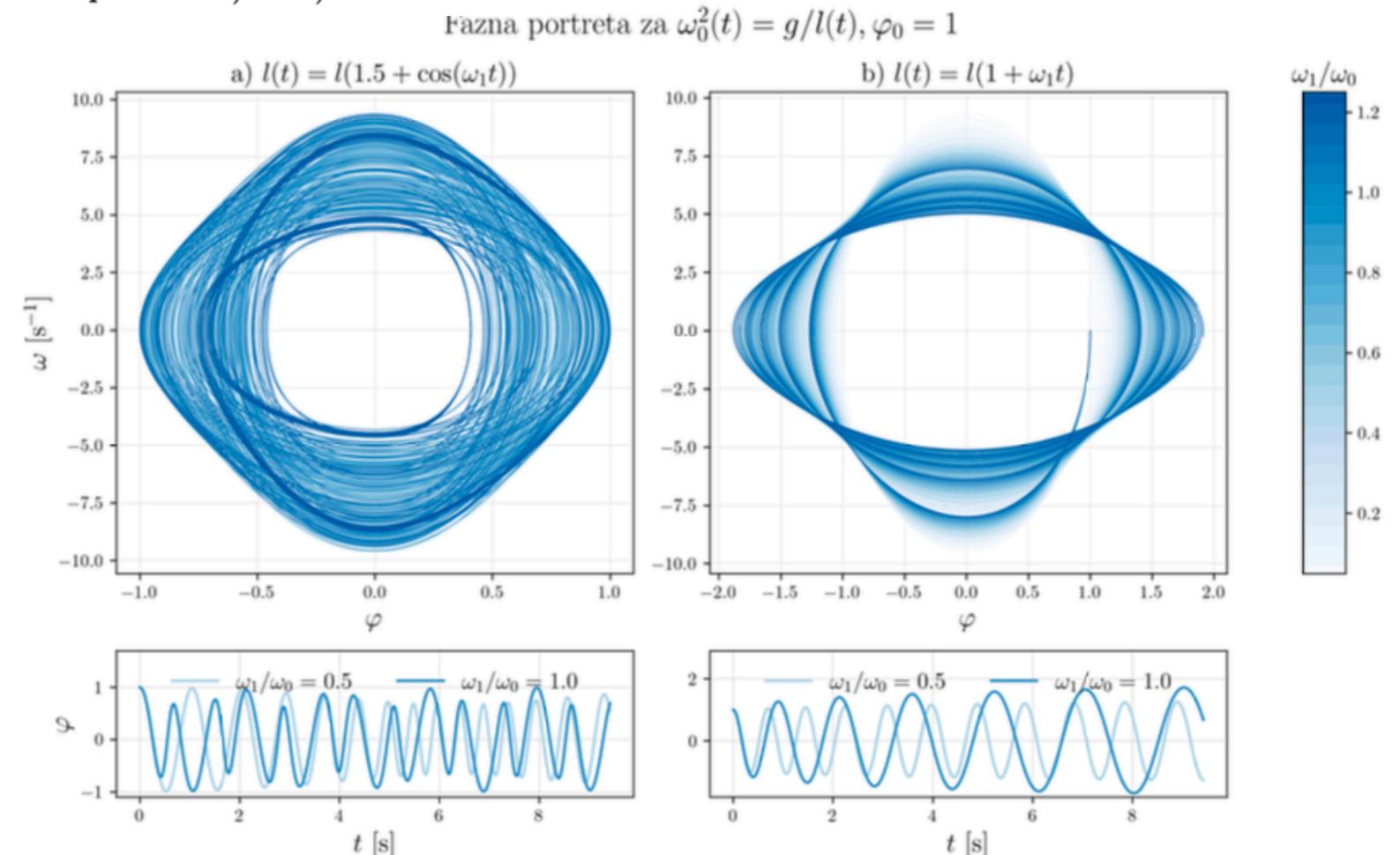
Dodatna samoiniciativnost..

- Parametrično vzbujanje ...

3.6 Časovno spremenljiva lastna frekvenca

V nalogi sem že lela obravnavati primer s časovno spremenljivo lastno frekvenco. Praktično rečeno to pomeni, da v strop zvrtamo luknjico in od zgoraj vlečemo vrvico, tako da se njena dolžina spodaj spreminja. Ogledala sem si primera $l_1(t) = l(1 + \cos(\omega_1 t))$ in $l_1(t) = l(1 + \omega_1 t)$ ter rešitvi takega problema v faznem prostoru. Fazne slike za različne $0.05\omega_0 < \omega_1 < 1.24\omega_0$ so prikazane na sliki 7. Rešitve sem poiskala s `scipy.odeint` za čase do $T = 10t_0^*$. Vizualen vtis pravi, da je rešitev z l_2 lepša ... Za izbrana primera $\omega_1/\omega_0 = 0.5, 1.0$ sta na sliki 7 še časovna poteka rešitev za primera a) in b).

BRAVO!



Slika 7: Gibanje matematičnega nihala s spremenljivo dolžino $l = l(t)$ in $\varphi_0 = 1$ za različne parametre ω_1 v faznem prostoru za $\varphi_0 = 1$ in $0 < t < 15t_0^*$. Množica portretov v primeru a) deluje presenetljivo neurejeno.

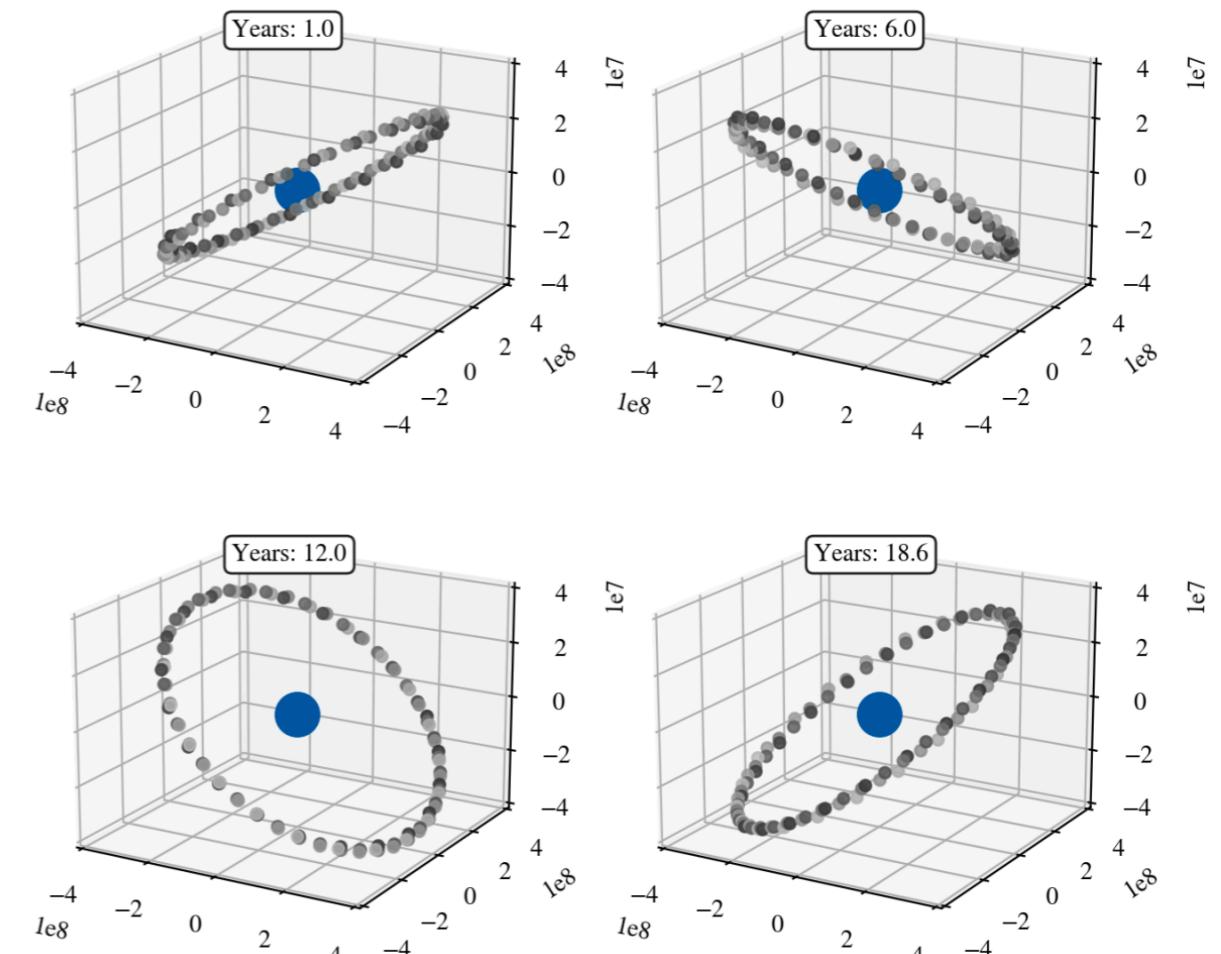
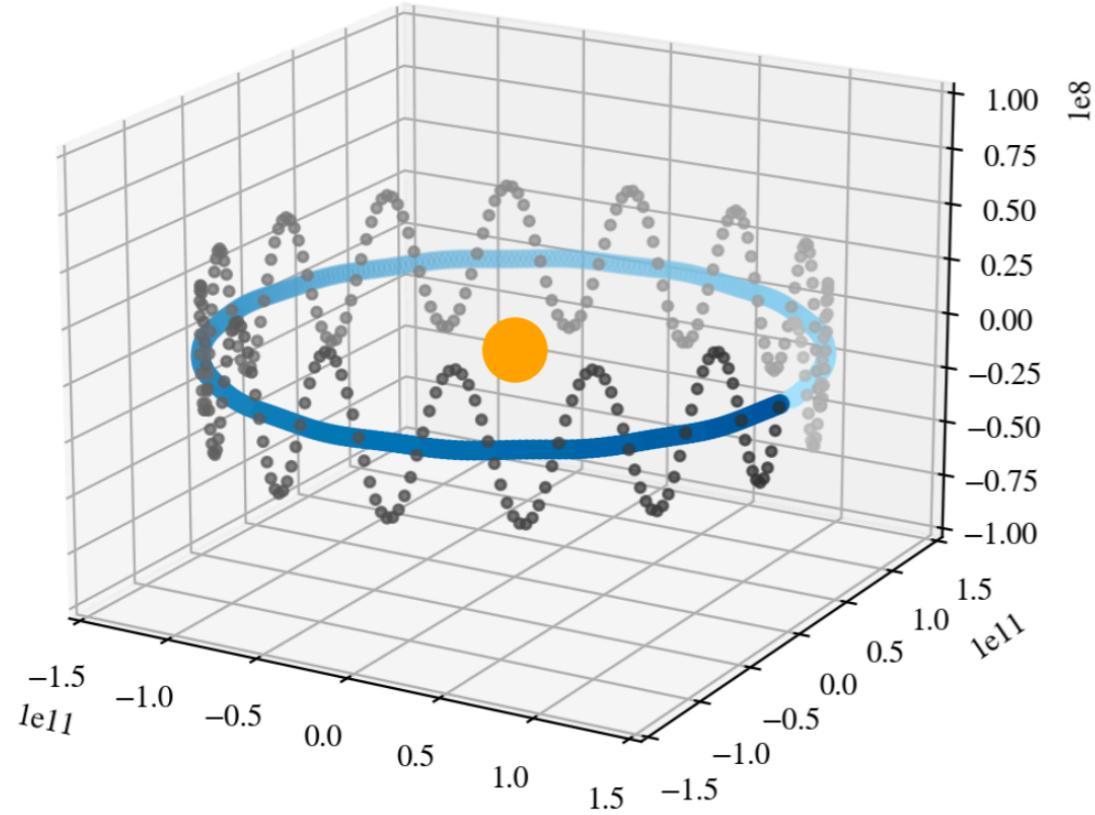


Dodatna

- Samo-iniciativnost....

As an extra, in the spirit of Newton's second law, I tried simulating the Earth-Moon-Sun three-body system using the differential equation methods in this report. As a simplification, I assumed the sun was fixed in inertial space, and solved only for the motion of the earth and moon around the sun, which reduces the problem to a system of 12 equations (2 bodies \times 3 spatial dimensions \times 2nd order differential equation). The corresponding code is included in the file `newton-ems.py`. Figure 14 shows the earth and moon's orbit around the sun over the course of one year with Figure 15 shows the moon's 18.6 year nodal precession period. The attached files `ems-animated-365.mp4` and `moon-precession.mp4` show animated versions of the same graphs.

Lani ...

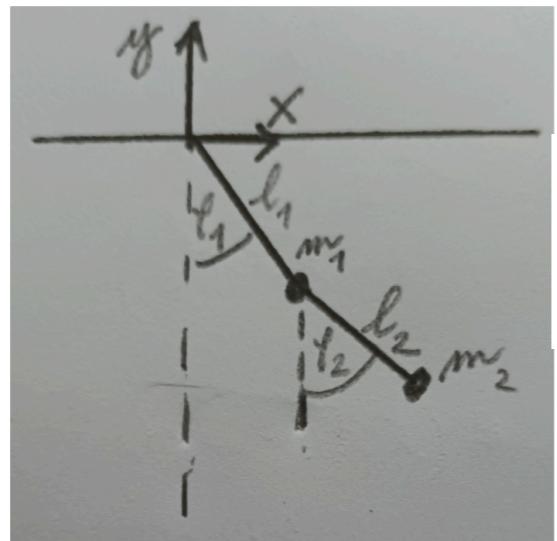


21 Figure 15: Visualizing the moon's 18.6 year nodal precession as it orbits the earth.



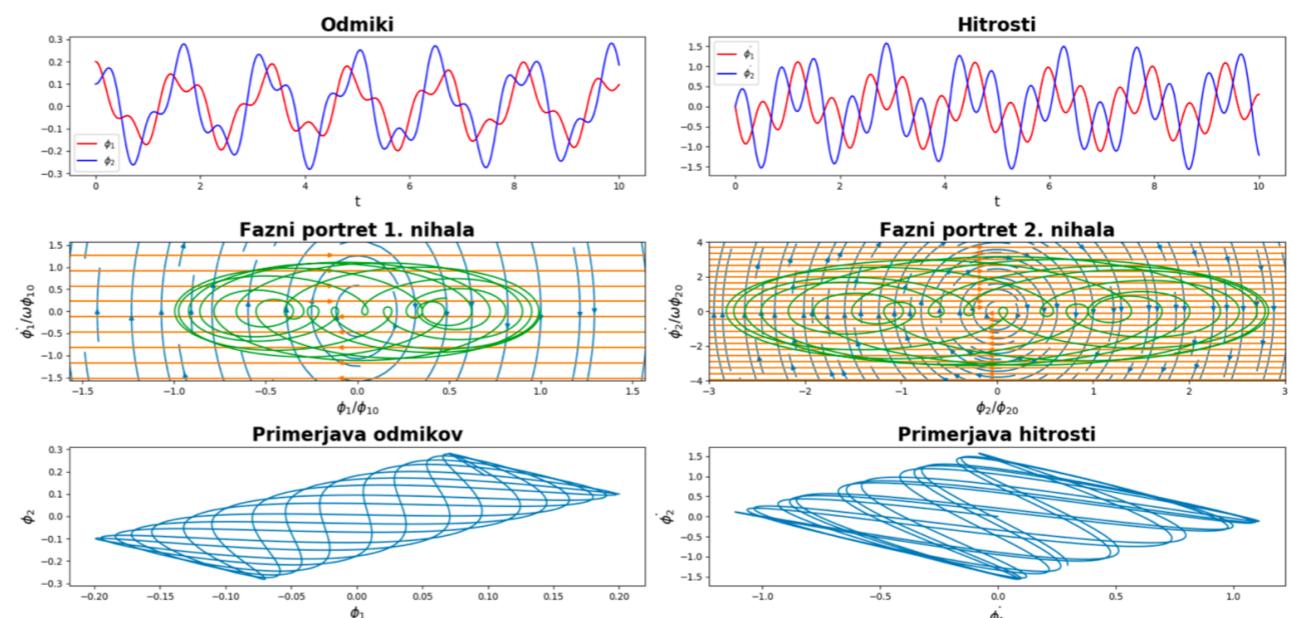
‘Dvojna’ samoiniciativnost..

Lani....



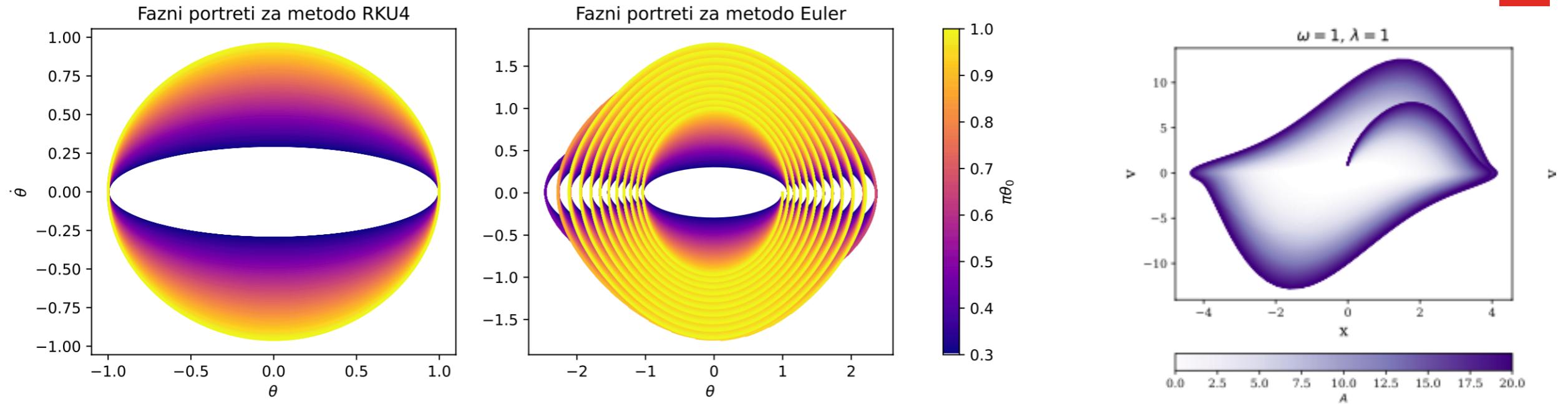
$$\ddot{\varphi}_2 + l \ddot{\varphi}_1 + \omega_2^2 \varphi_2 = 0$$

$$\ddot{\varphi}_1(1 + \mu) + \frac{\mu}{l} \ddot{\varphi}_2 + \omega_1^2 \varphi_1(1 + \mu) = 0$$

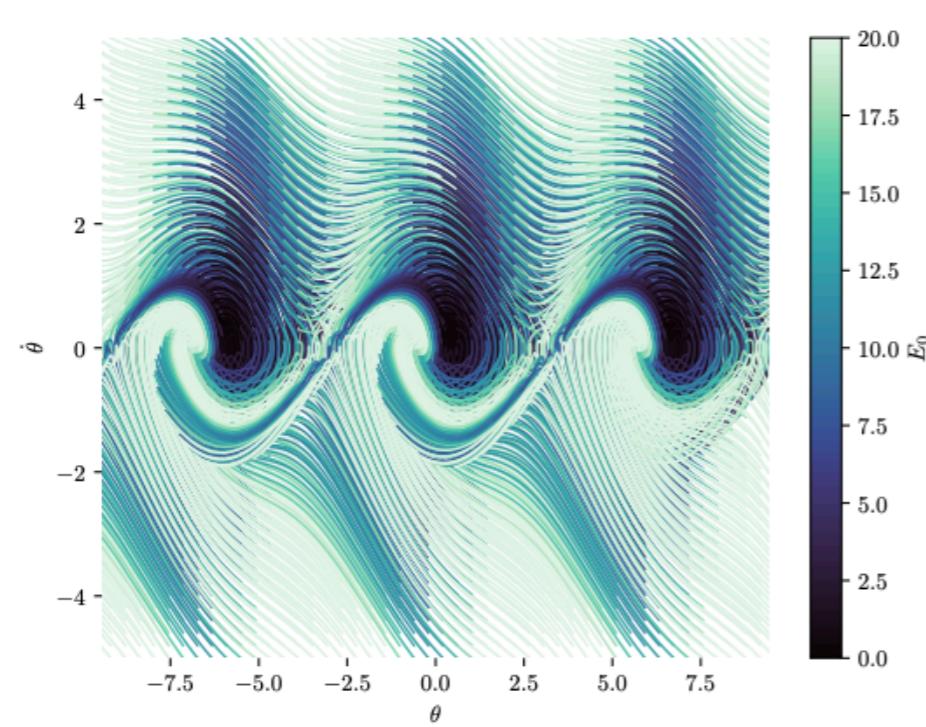


Slika 8: Rešitve dodatne naloge z začetnimi pogoji $\varphi_1 = 0.2$, $\varphi_2 = 0.1$, $\dot{\varphi}_1 = \dot{\varphi}_2 = 0$ in $\omega = 5$.

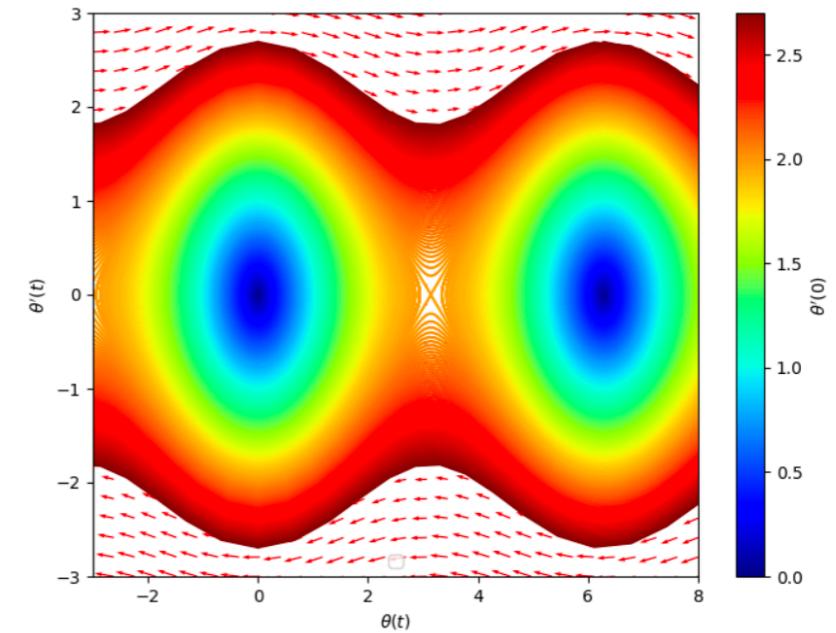
Najlepši fazni portreti skozi čas



Slika 8: Lepi fazni portreti. Trdim, da so Egipčani tukaj dobili navdih za piramide.



Slika 8: Fazni portret dušenega nihala pri $\omega_0 = 2/3, v = 0.8, \beta = 0.5$



Slika 8: Fazni diagram matematičnega nihala z umetniškim navdihom.



Zaključki

Se opravičujem, poglavje 5 mi je pojedel pes.

5 Zaključek

