Poročilo 3. domače naloge

Nejc Hirci, 63180335

6. 8. 2024

1 Matematično nihalo

Kotni odmik $\theta(t)$ (v radianih) pri nedušenem nihanju nitnega nihala opišemo z diferencialno enačbo drugega reda:

$$\frac{g}{l}\sin(\theta(t)) + \theta''(t) = 0 \quad \theta(0) = \theta_0, \quad \theta'(0) = 0$$

Preprosto jo lahko prevedemo na sistem dveh diferencialnih enačb prvega reda z vpeljavo nove spremenljivke $\omega(t) = \theta'(t)$:

$$\begin{aligned} \theta'(t) &= \omega(t) && \text{z začetnim pogojem } \omega(0) = \omega_0 \\ \omega'(t) &= -\frac{g}{l}\sin(\theta(t)) && \text{z začetnim pogojem } \theta(0) = \theta_0 \end{aligned}$$

Za reševanje bomo uporabili metodo Runge-Kutta 4. reda, ki aproksimira odvod v točki t z 4 tangentami:

$$\begin{aligned} k_1 &= hf(t, y) \\ k_2 &= hf(t + \frac{h}{2}, y + \frac{k_1}{2}) \\ k_3 &= hf(t + \frac{h}{2}, y + \frac{k_2}{2}) \\ k_4 &= hf(t + h, y + k_3) \end{aligned}$$

[68]:
```
# Dodajanje implementacije v Jupyter notebook
include("../src/main.jl");
```

1.1 Testiranje implementacije

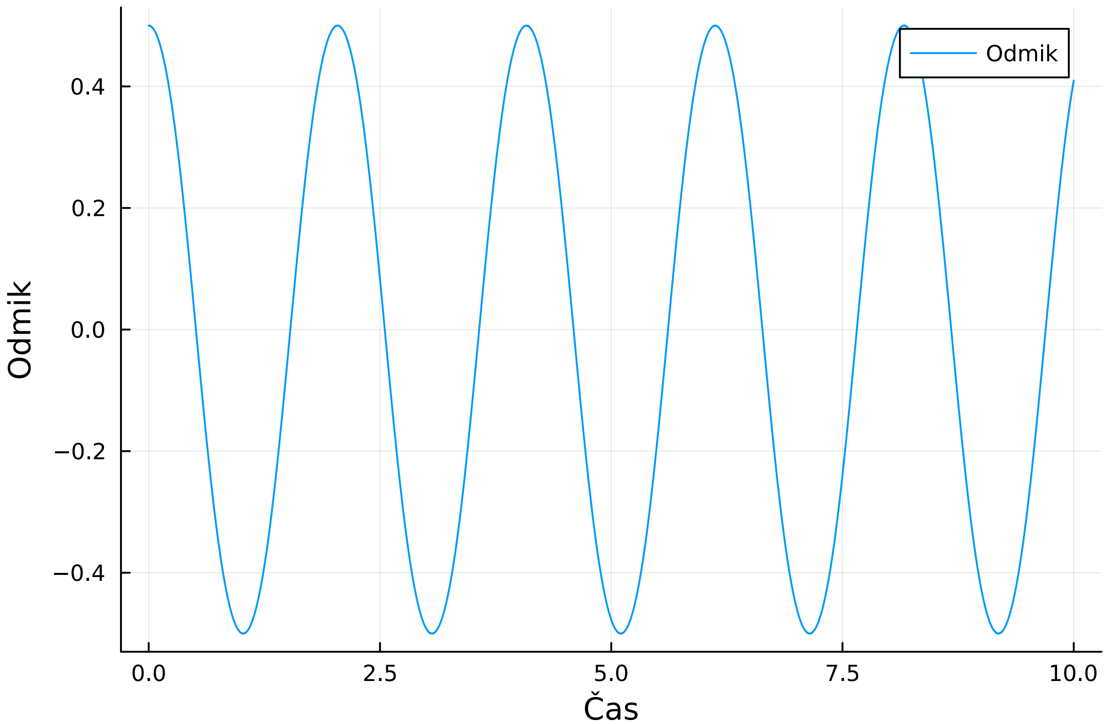
Najprej bomo preverili pravilno implementacijo metode Runge-Kutta 4. reda.

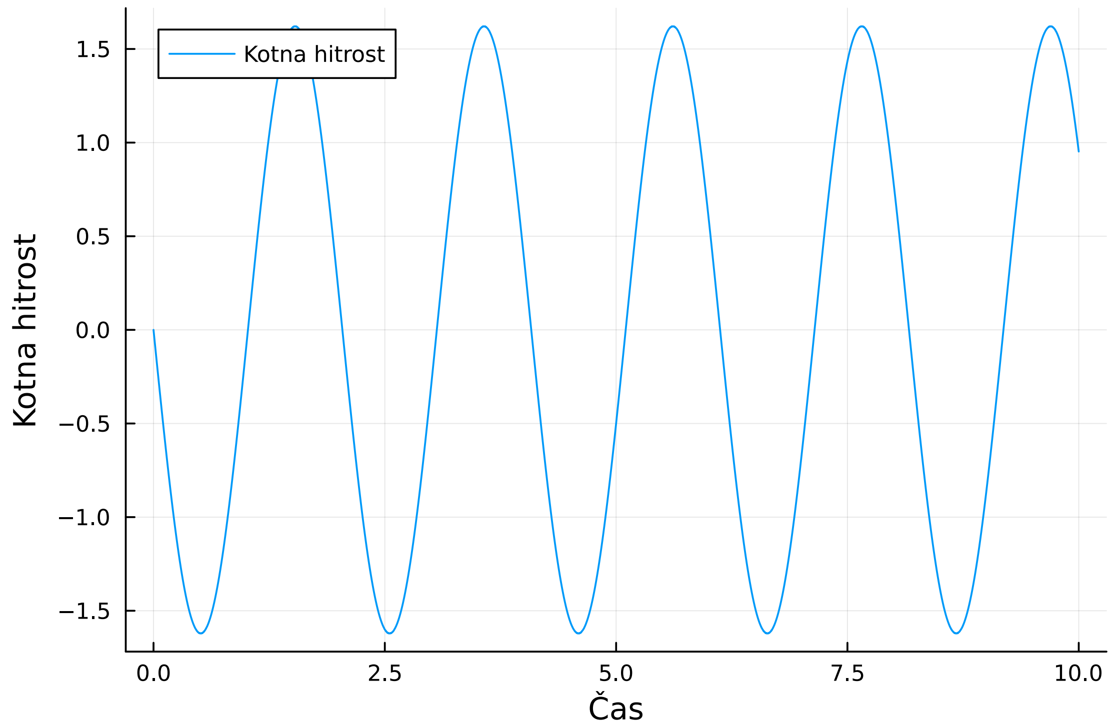
[69]:
```
# Testiranje implementacije
include("../tests/runtests.jl");
```

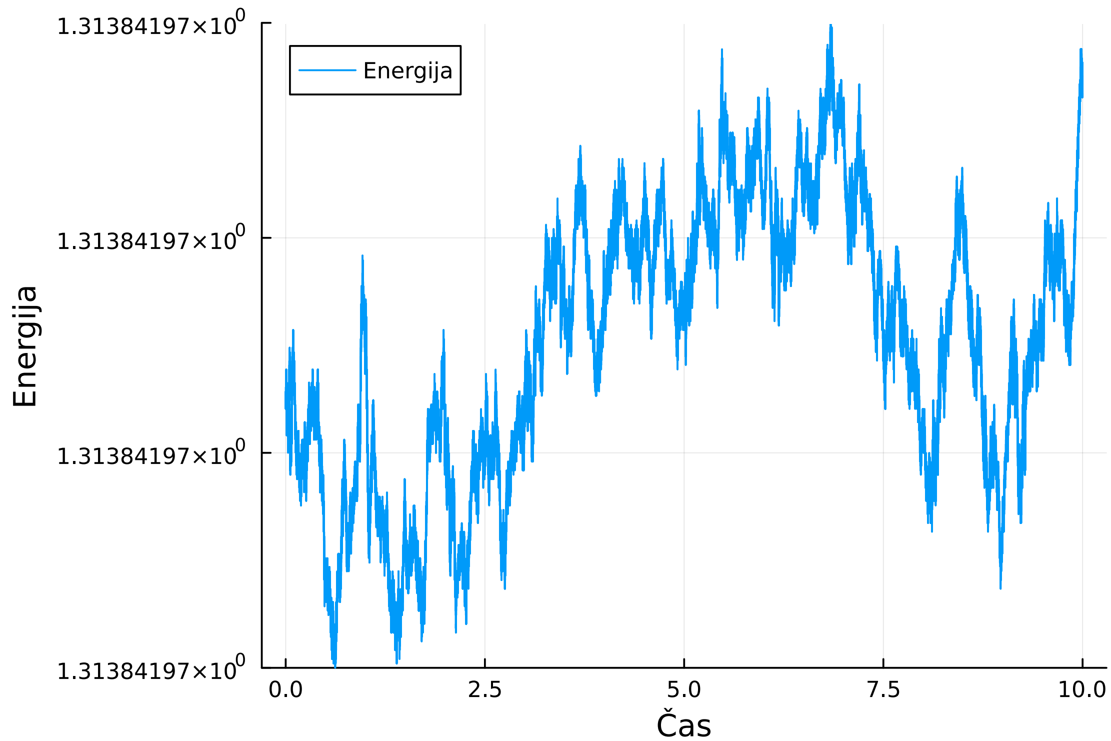
```
Test Summary:                              | Pass

Total  Time
Testiranje Runge-Kutta metode 4. reda      |    2      2

0.4s
```

```
  globalna napaka na primeru dy=-y+1, y(0) = 2 |     1       1
0.3s
  ocena lokalne napake na primeru nihala       |     1       1
0.1s
```

[70]:
```
# Prikaz grafov implementacije
l = 1.0
t = 10.0
theta0 = pi/6
dtheta0 = 0.0
n = 100000
out_nihalo = nihalo(l, t, theta0, dtheta0, n)
plot_pendulum(out_nihalo, t, l)
```




Kotna hitrost
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.0
2.5
5.0
7.5
10.0
Čas
Kotna hitrost


Energija
1.31384197×10^0
1.31384197×10^0
1.31384197×10^0
1.31384197×10^0
0.0
2.5
5.0
7.5
10.0
Čas
Energija

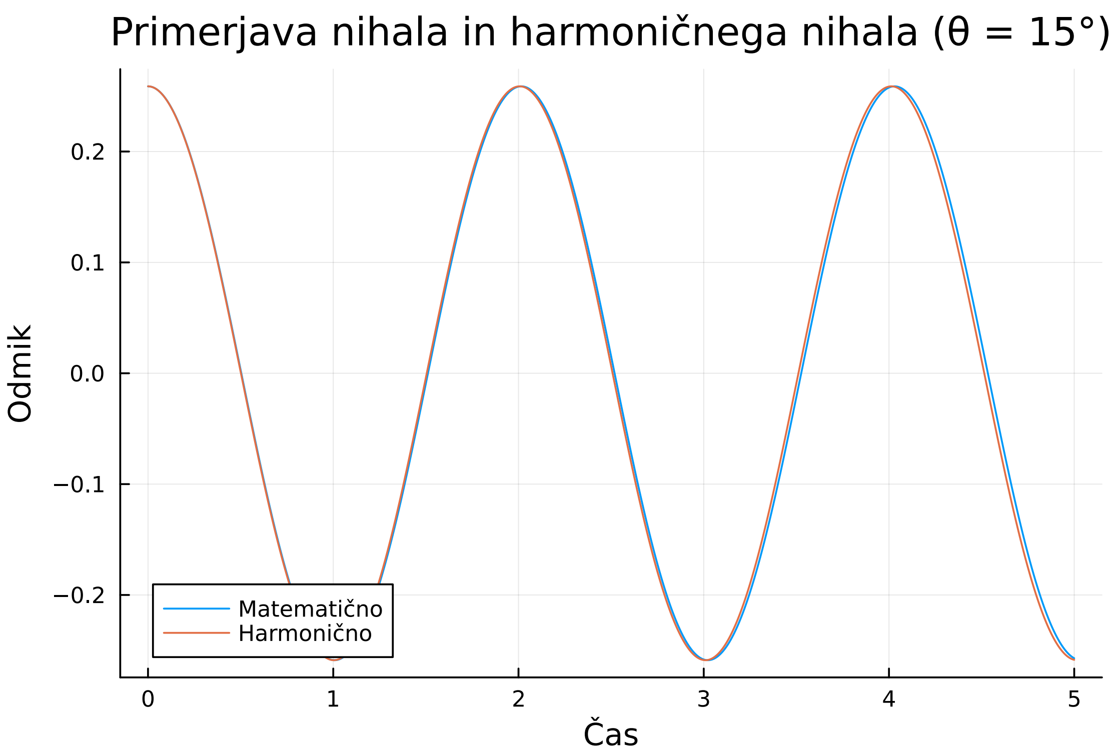
```
Relativna napaka ohranitve energije: 2.1632517451367652e-14
```

1.2 Primerjava implementacije z nihanjem harmoničnega nihala

Pri harmoničnem nihalo je odmična sila sorazmerna z odmikom in nasprotno usmerjena. Praviloma je to res za majhne odmike, ko je razlika med $\sin(\theta)$ in θ zanemarljiva (približno 1% za $\theta < 15°$).

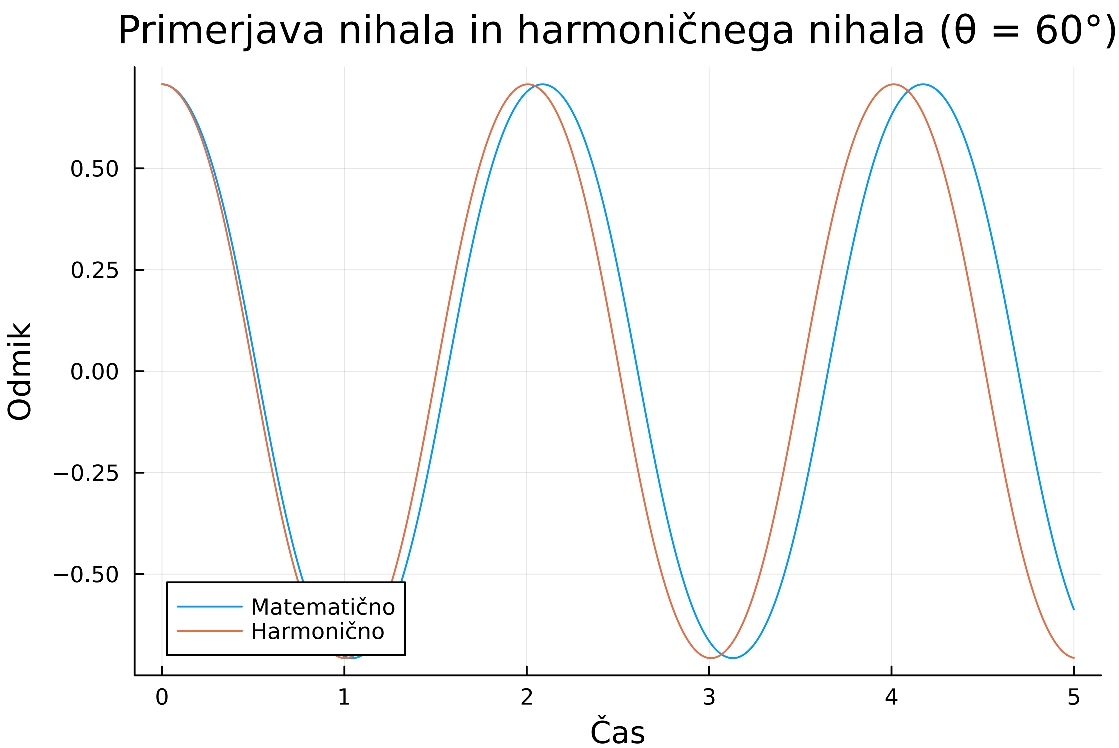
Primerjamo nihajni čas za majhne in velike odmike, ter grafe odmika in hitrosti v odvisnosti od časa.

[31]:
```
t = 5.0
l = 1.0
n = 1000
theta0 = pi / 12.0
out_nihalo = nihalo(l,t,theta0, 0.0, n)
out_harmonic = harmonicno_nihalo(l,t,theta0,dtheta0,n)
plot_compare(out_nihalo, out_harmonic, t, l, "Primerjava nihala in harmoničnega␣
 ↪nihala (θ = 15°)")
```



[41]:
```
g = 9.80665
t = 5.0
l = 1.0
n = 1000
theta0 = pi / 4.0
```

```
dtheta0 = 0.0
out_nihalo = nihalo(l,t,theta0, dtheta0, n)
out_harmonic = harmonicno_nihalo(l,t,theta0,dtheta0,n)
plot_compare(out_nihalo, out_harmonic, t, l, "Primerjava nihala in harmoničnega␣
↪nihala (θ = 45°)")
```



Primerjamo nihajne čase matematičnega in harmoničnega nihala za $\theta_0 = 15°$ in $\omega_0 = 0$. Pri tem za periodo harmonično nihalo uporabimo enačbo:

$$T = 2\pi\sqrt{\frac{l}{g}}$$

Periodo matematičnega nihala izračunamo kot čas med dvema prehodoma skozi ničelni odmik, pri čemer uporabimo rezultat Runge-Kutta metode skupaj z bisekcijo med dvema prehodoma skozi ničelni odmik za natančnejši rezultat.

Implementacijo za izračun periode primerjamo z analitično rešitvijo matematičnega nihala izračunano preko popolnega eliptičnega integrala in knjižnice `Elliptic.jl`. Analitična enačba periode nelinearnega nihala povzeta iz Exact solution for the nonlinear pendulum je:

$$T = 4\sqrt{\frac{l}{g}}K(\sin^2(\frac{\theta_0}{2}))$$

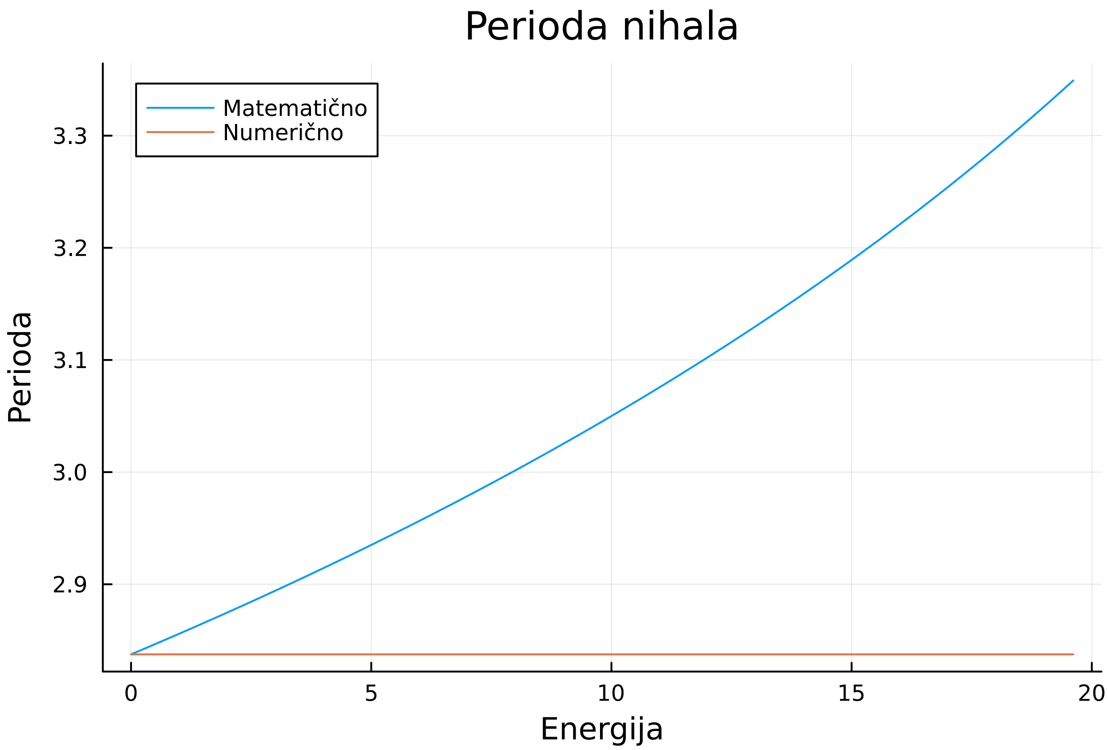
[45]:
```
using Elliptic

l = 1.0
theta0 = pi / 12.0
dtheta0 = 0.0

# Izračunamo nihajni čas harmoničnega nihala in matematičnega nihala
t_harmonic = 2*pi*sqrt(l/g)

# Izračunamo nihajni čas matematičnega nihala z bisekcijo
t_nihalo = nihajni_cas(l, theta0, dtheta0, 100000)

println("Nihajni čas harmoničnega nihala: ", t_harmonic)
println("Nihajni čas matematičnega nihala z bisekcijo Runge-Kutte: ", t_nihalo)

t_nihalo_exact = 4 * sqrt(l / g) * Elliptic.K(sin(theta0 / 2)^2)
println("Relativna numerična napaka periode: ", abs(t_nihalo - t_nihalo_exact) /␣
↪t_nihalo_exact)
```

```
Nihajni čas harmoničnega nihala: 2.0064092925890407
Nihajni čas matematičnega nihala z bisekcijo Runge-Kutte: 2.015038014606129
Relativna numerična napaka periode: 3.3719288973482885e-14
```

1.3 Energijska analiza nihanja

Energijska analiza nihanja pokaže, da je perioda matematičnega nihala odvisna od energije sistema za razliko od harmoničnega nihala, kjer je perioda neodvisna od energije.

[40]:
```
l = 2.0
m = 1.0
thetas0 = range(0.01, stop=pi / 2.0, length=1000)
energies = zeros(1000)
periods_math = zeros(1000)
periods_harmonic = zeros(1000)
for i in 1:1000
    theta0 = thetas0[i]
    energies[i] = m * g * l * (1 - cos(thetas0[i]))
    periods_math[i] = nihajni_cas(l, thetas0[i], 0.0, 10000)
    periods_harmonic[i] = 2 * pi * sqrt(l / g)
end
plot(energies, periods_math, label="Matematično", xlabel="Energija",␣
↪ylabel="Perioda", title="Perioda nihala")
plot!(energies, periods_harmonic, label="Numerično")
```


Perioda nihala
Matematično
Numerično
Perioda
3.3
3.2
3.1
3.0
2.9
0
5
10
15
20
Energija