

Gimnazija Kranj

# Optimizacija trebušeja z enojno ročico

**Tehnika**

**Raziskovalna naloga**

Avtor: Urh Mur

Mentorica: Barbara Hudovernik, prof.

Kranj, 2024

**Zahvala**

Rad bi se zahvalil moji mentorici za stalno podporo pri pisanju raziskovalne naloge. Rad bi se zahvalil tudi staršema in starima staršema, ki so mi namenili prostor in orodje za moje raziskovanje.

## Kazalo vsebine

<b>Zahvala</b> .....	2
<b>Povzetek</b> .....	5
<b>Abstract</b> .....	5
<b>1 UVOD</b> .....	6
<b>1.1 Cilj raziskave</b> .....	7
<b>1.2 Raziskovalno vprašanje</b> .....	7
<b>1.3 Hipoteze in napovedi</b> .....	7
<b>1.4 Spremenljivke</b> .....	7
<b>2 RAZISKOVALNI DEL</b> .....	8
<b>2.1 Teoretično ozadje in značilnosti modela</b> .....	8
<b>2.2 Izpeljava enačb gibanja z uporabo Pythona</b> .....	11
<b>2.3 Simulacija poti izstrelka v Pythonu</b> .....	14
<b>2.4 Izdelava modela trebušaja</b> .....	17
<b>2.5 Zbiranje neobdelanih podatkov</b> .....	18
<b>2.6 Analiza podatkov</b> .....	21
<b>2.7 Vrednotenje analiziranih podatkov in razprava</b> .....	25
2.7.1 Izmerjene vrednosti.....	25
2.7.2 Viri napak .....	26
2.7.3 Izboljšave .....	26
<b>3 ZAKLJUČEK</b> .....	26
<b>5 PRILOGE</b> .....	29
<b>5.1 Tabele neobdelanih podatkov</b> .....	29
<b>5.2 Koda, uporabljena za izpeljavo enačb</b> .....	34
<b>5.3 Koda, ki se uporablja za integracijo enačb</b> .....	35

## KAZALO SLIK

Slika 1: Skica trebušeja z enojno ročico .....	6
Slika 2: Podrobna skica trebušeja z opombami .....	8
Slika 3: Model trebušeja.....	17
Slika 4: Primer interferenčnega vzorca .....	18
Slika 5: Analiza podatkov v LoggerPro .....	19

## KAZALO TABEL

Tabela 1: Teoretična napoved hitrosti pri vseh razmerjih vzvodov .....	17
Tabela 2: dolžine ročic pri različnih razmerjih ročic.....	18
Tabela 3: Položaj izstrelka in hitrosti izstrelka in ročice pri razmerju ročice 1:2 .....	20
Tabela 4: Točke in pripadajoče hitrosti v tabeli.....	22
Tabela 5: Vrednosti, dobljene z interpolacijo .....	23
Tabela 6: Položaji ob izstrelitvi in hitrosti izstrelka ter ročice pri različnih razmerjih ročic .....	24
Tabela 7: Učinkovitost pri različnih razmerjih ročic .....	25
Tabela 8: Položaji in hitrosti izstrelkov in ročice pri razmerju ročic 1:3.....	29
Tabela 9: Položaji in hitrosti izstrelkov in ročice pri razmerju ročic 1:4.....	30
Tabela 10: Položaji in hitrosti izstrelkov in ročice pri razmerju ročic 1:5.....	31
Tabela 11: Položaji in hitrosti izstrelkov in ročice pri razmerju ročic 1:7 .....	32
Tabela 12: Položaji in hitrosti izstrelkov in ročice pri razmerju ročic 1:9.....	33

## KAZALO GRAFOV

Graf 1: Koti po času, teoretična napoved pri razmerju ročic 1:2.....	15
Graf 2: Položaj izstrelka, teoretična napoved pri razmerju ročic 1:2 .....	15
Graf 3: Graf točk, kjer je naklon poti izstrelka 45 stopinj glede na vodoravnico, teoretična napoved pri razmerju ročice 1: 2.....	16
Graf 4: Graf pozicije v daljšem časovnem obdobju, teoretična napoved .....	16
Graf 5: Pot izstrelka pri razmerju vzvoda 1:2 .....	21

## **Povzetek**

S to nalogo želim pokazati, kako sem ugotovil, da je najboljše razmerje za trebuše z enojno ročico 1:7. Najprej sem s pomočjo Python-a (*Van Rossum & Drake, 1995*) izpeljal enačbe, ki opisujejo gibanje trebušaja skozi Lagrangeevo mehaniko. Nato sem s Python-om dobljene enačbe tudi rešil, kar je potrebovalo tudi nekaj inovativnosti, ker Python ne omogoča reševanja diferencialnih enačb drugega ali višjega reda. Da pa ta naloga ne bi bila zgolj teoretična, sem izdelal tudi pomanjšan model trebušaja, katerega sem tudi testiral, in ugotovil, da je glede na vse relevantne energije, prisotne v sistemu, najbolj učinkovit prav trebuše, katerega razmerje ročic je 1:7. Trebuše sem testiral tako, da sem gibanje posnel s fotoaparatom, nato pa sem hitrost gibanja in položaj izračunal s pomočjo programa LoggerPro, ter nato še dalje v Excelu. Po izračunih je učinkovitost trebušaja 0,33, kar je precej dobro za mehanski sistem te vrste.

## **Abstract**

In this research article, I shall show how I obtained that the best mechanical advantage of a single arm trebuchet is 1:7. Firstly, I have derived the equations of motion of the trebuchet using Lagrangian mechanics. Then, I used Python to solve the derived equations, which required some innovativity, as Python does not allow for solving second or higher order differential equations. In order for this research article not to be purely theoretical, I made a downscaled model of a trebuchet. I have found out, that, taking all the relevant energies inside the system into account, the most effective setup of the arm is 1:7. I have tested the trebuchet by filming the motion with a camera, and then calculating the position and velocity of the projectile in LoggerPro, and then further analysing the data in Excel. The calculations provide that the efficiency of the single arm trebuchet at 1:7 lever ratio is 0,33, which is very well for a mechanical system of this sort.

## **Ključne besede**

Trebuše, Lagrangeeva mehanika, programersko modeliranje, kaotični sistemi

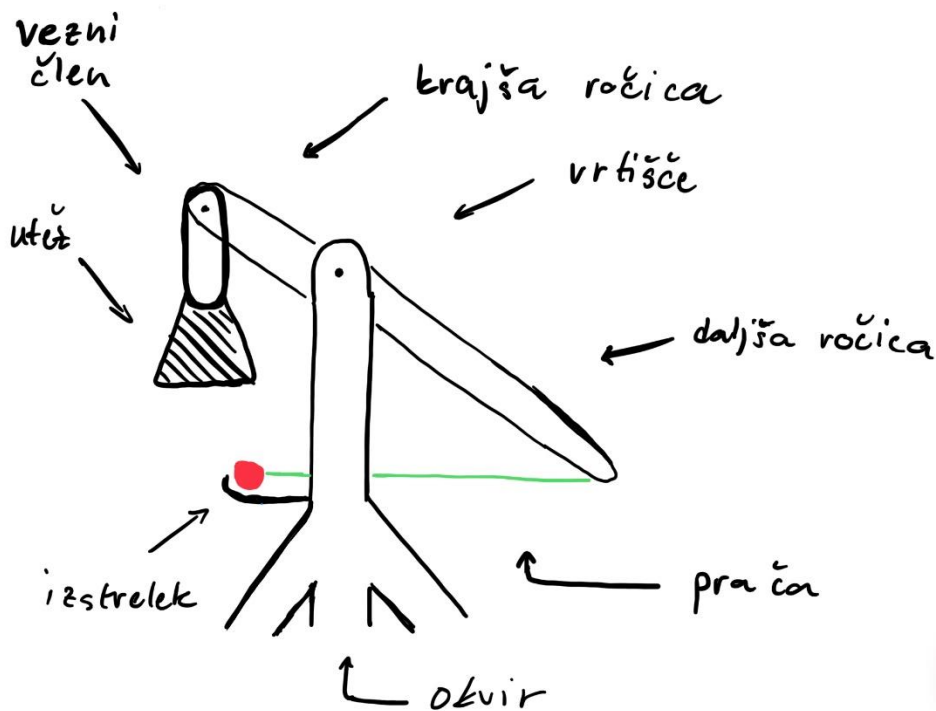
## **Keywords**

Trebuchet, Lagrangian mechanics, computer modelling, chaotic systems

# 1 UVOD

Trebuše je starodavno oblegovalno orožje, natančneje katapult, ki je bil namenjen metanju težkih kamnov ali ognjenih krogel proti sovražniku. Z izumom smodnika je uporaba trebušejev strmo upadla. Zanimanje je naraslo pred nekaj desetletji, ko so se pojavila prva tekmovanja v metanju buč. Razvoj katapultov na splošno, zlasti trebušejev, se je začel z znanstvenim pristopom, vse z vprašanjem "Kako izgubiti čim manj energije?" v mislih, tako pa so nastali trebušaji s prosto ročico (floating arm), trebušaji s sestavljeno ročico (whipper trebuchet) ali celo kombinacije obeh. Za fizikalno analizo sta oba trebušaja takšne vrste izjemno zahtevna, zato se bom raje lotil klasičnega primera, trebušaja z enojno ročico.

Slika 1: Skica trebušaja z enojno ročico



S fizikalnega vidika je zgradba zelo preprosta. Konstrukcija temelji na dvojnem nihalu, ki ga poganja protiutež. Vendar, če bi bila protiutež fiksno pritrjena na ročico, bi se nihanje nadaljevalo, zato uporabimo vezni člen, ki se lahko prosto vrtil. To pomeni, da imamo še eno dvojno nihalo, ki zaradi velikega vztrajnostnega momenta ustavi nihanje na najnižji točki. Vendar nas za namene te raziskovalne naloge ta stran ne zanima toliko kakor daljša stran ročice in izstrelek.

## 1.1 Cilj raziskave

Cilj te raziskave je ugotoviti, pri kakšnem razmerju ročic je izhodna hitrost izstrelka najvišja, torej, ima izstrelak najvišjo kinetično energijo, ko je naklon poti izstrelka 45 stopinj na vodoravnico. Cilj je tudi teoretično napovedati izid s simulacijo trebušaja pri različnih razmerjih ročic. To bom storil s programskim jezikom Python, nato pa praktično z izdelavo trebušaja in nato opazovanjem gibanja daljše ročice in izstrelka.

## 1.2 Raziskovalno vprašanje

Kot je pojasnjeno v cilju raziskave, je raziskovalno vprašanje naslednje:

Kakšno je najboljšo razmerje ročic pri fiksni dolžini celotne ročice, da dobi izstrelak največjo kinetično energijo?

## 1.3 Hipoteze in napovedi

V idealnem primeru bi si želeli, da bi imel trebuše čim bolj kratko krajšo ročico in čim bolj dolgo daljšo ročico, vendar smo omejeni s težo protiuteži. Ker je trebuše omejen tudi s strukturno trdnostjo materialov, mora obstajati optimum, kjer je pri fiksni masi protiuteži izkoristek največji. Viri iz interneta in iz mojih lastnih izkušenj kažejo, da je najboljšo razmerje 1:4 ali manj. Članek na blogu (*Kalif, 2017*) pravi, da je 1:3,75 najboljšo razmerje za ročico trebušaja. Zato predvidevam, da je to res, vendar bi teoretična napoved v Pythonu in testiranje morala pokazati, ali so ti podatki pravilni ali ne.

## 1.4 Spremenljivke

**Neodvisna spremenljivka** v tem poskusu je razmerje ročic .

**Odvisna spremenljivka** je kinetična translacijska energija izstrelka.

**Nadzorovane spremenljivke** so masa protiuteži, masa ročice, skupna dolžina ročice in začetni položaj ročice pri 45 stopinjah glede na vodoravnico.





kot protiutež, toliko bolj, če dodamo še eno nihalo. Toda težava se pojavi, ko poskušamo opisati gibanje trebušaja znotraj newtonske mehanike. Zato bom uporabil Lagrangeevo mehaniko, približek mehanske teorije, ki se uporablja za opis poti najmanjšega upora, da bi poskusil rešiti to težavo.

V nadaljevanju lahko definiramo element, imenovan Lagrangian, kot razliko med vsoto vseh kinetičnih in vseh potencialnih energij (*Lagrangian mechanics*, 2024).

$$L = T - V \quad (1)$$

Kjer je L Lagrangian, T je vsota kinetičnih energij in V vsota potencialnih energij znotraj sistema. Kinetične in potencialne energije so izračunane na enak način, kot bi bile v newtonskem sistemu. Vse spremenljivke ali količine, uporabljene pri izračunih, so prikazane na sliki 2.

Položaj protiuteži lahko definiramo kot:

$$x_{cw} = d_{sa} \cdot \sin \theta \quad (2)$$

$$y_{cw} = (h - d_{cw}) + d_{sa} \cdot \cos \theta \quad (3)$$

Kjer je  $d_{sa}$  dolžina krajše ročice,  $d_{cw}$  veznega člena in  $\theta$  kot med ročico in normalo. V tem primeru je  $\theta$  neznana funkcija časa. Spremenljivka, označena kot h,  $h = d_{la} \cdot \cos\left(\frac{\pi}{4}\right)$ , je torej za vsako razmerje ročic teoretično različna.

Položaj projektila lahko definiram kot:

$$x_p = d_s \cdot \sin(\theta + \phi) - d_{la} \cdot \sin \theta \quad (4)$$

$$y_p = h + d_s \cdot \cos(\theta + \phi) - d_{la} \cdot \cos \theta \quad (5)$$

Y koordinato težišča ročice lahko definiram kot:

$$y_a = h + (0,5 \text{ m [meter]} - d_{sa}) \cdot \cos \theta \quad (6)$$

Kjer je  $d_s$  dolžina prače,  $d_{la}$  pa dolžina daljše ročice.  $\phi$  je enako kot  $\theta$  neznana funkcija časa.

Tako lahko V v Lagrangianu zapišem kot:

$$V = g \cdot (m_p \cdot y_p + m_{cw} \cdot y_{cw} + m_a \cdot y_a) \quad (7)$$

Različni viri z interneta, ki so že izvajali simulacije trebušaja prek Lagrangeeve mehanike, so privzeli, da je ročica brez mase (*de Jong, 2020*). To pomeni, da sem v tej raziskovalni nalogi dodal malo razširitve k temu, kar je de Jong zelo dobro pokazal.

Toda T je bolj zapleten, zato ga bom razdelil na štiri enačbe:  $T_1$  in  $T_2$  sta kinetični translacijski energiji, medtem ko sta  $T_3$  in  $T_4$  rotacijski kinetični energiji. Hitrosti, ki so označene z  $v$ , so dejansko prvi odvodi translacijskih in kotnih hitrosti, označeni kasneje s prvim odvodom  $\theta$  ali  $\phi$  po času.

$$T = T_1 + T_2 + T_3 + T_4 \quad (8)$$

$$T_1 = \frac{1}{2} m_p \cdot (v_{px}^2 + v_{py}^2) \quad (9)$$

$$T_2 = \frac{1}{2} m_{cw} \cdot (v_{cwx}^2 + v_{cwy}^2) \quad (10)$$

$$T_3 = \frac{1}{2} (I_p \cdot \omega_p^2) \quad (11)$$

$$T_4 = \frac{1}{2} \cdot (I_a \cdot \omega_a^2) \quad (12)$$

Tu sem implementiral dve novi spremenljivki oziroma količini,  $I_p$  in  $I_a$ , ki sta vztrajnostna momenta ročice in izstrelka. Izpeljal sem ju iz Steinerjevega pravila.

$$I_p = m_p \cdot d_s^2 \quad (13)$$

In:

$$I_a = I_{arm} + m_a \cdot e^2$$

Kjer je  $I_{arm} = \frac{m_a \cdot (d_s + d_{la})^2}{12}$  in  $e = \frac{d_{la} + d_{sa}}{2} - d_{sa}$  ter  $d_{la} + d_{sa} = 1 \text{ m [meter]}$ , tako iz tega sledi:

$$I_a = \frac{m_a}{12} + (0,5 - d_{sa})^2 \cdot m_a \quad (14)$$

Lagrangian se nato reši s posebno metodo, ki se imenuje Euler-Lagrangeev odvod (*Lagrangian mechanics, 2024*).

$$\frac{\partial L}{\partial \theta} - \frac{d}{dt} \left( \frac{\partial L}{\partial \theta'} \right) = 0 \quad (15)$$

$$\frac{\partial L}{\partial \phi} - \frac{d}{dt} \left( \frac{\partial L}{\partial \phi'} \right) = 0 \quad (16)$$

Iz tega bom nadalje pokazal, kako izpeljati enačbe Lagrangeevega gibanja z uporabo jupyter-ja (*Perez & Granger, 2007*), ki lahko prevaja programski jezik Python v besedilno enačbo.

## 2.2 Izpeljava enačb gibanja z uporabo Pythona

Ker sem že določil enačbe položaja vseh členov, moram izpeljati še kotne in translacijske hitrosti vsakega člena, vključenega v sistem. Najprej bi rad izpeljal odvode položaja, ki predstavljajo translacijske hitrosti in pospeške v smeri x in y ter odvode kotov  $\theta$  in  $\phi$ , ki predstavljajo kotne hitrosti in pospeške. Za to bom uporabil knjižnico za Python, imenovano SymPy (*SymPy, 2017*). Celotna koda je v poglavju 5.2 v Prilogi. SymPy omogoča uvedbo simbolov ali spremenljivk in nato uporabo za simbolno računanje, zaradi česar je postopek izpeljave parcialnih odvodov ali odvodov po neznani spremenljivki veliko enostavnejši. Enačbe položaja lahko definiram kot (glej `#defining the equations of position`):

$$\begin{aligned}x_{cw} &= d_{sa} \sin(\theta(t)) \\ y_{cw} &= -d_{cw} + d_{sa} \cos(\theta(t)) + h\end{aligned}$$

In:

$$\begin{aligned}x_p &= -d_{la} \sin(\theta(t)) + d_s \sin(\phi(t) + \theta(t)) \\ y_p &= -d_{la} \cos(\theta(t)) + d_s \cos(\phi(t) + \theta(t)) + h \\ y_a &= h + (0,5 - d_{sa}) \cdot \cos(\theta(t))\end{aligned}$$

Kar je enako, kot je bilo napisano pod prejšnjo točko.

Odvodi  $\theta$  in  $\phi$  po času so opredeljeni kot (glej `#defining first and second derivatives of theta and phi`):

$$\frac{d}{dt} \theta(t), \frac{d^2}{dt^2} \theta(t), \frac{d}{dt} \phi(t), \frac{d^2}{dt^2} \phi(t)$$

Odvodi pozicije skozi čas, ki so hitrosti (glej `#defining derivatives (velocities) of counterweight and projectile position`):

$$\begin{aligned}\frac{dx_{cw}}{dt} &= d_{sa} \cos(\theta(t)) \frac{d}{dt} \theta(t) \\ \frac{dy_{cw}}{dt} &= -d_{sa} \sin(\theta(t)) \frac{d}{dt} \theta(t) \\ \frac{dx_p}{dt} &= -d_{la} \cos(\theta(t)) \frac{d}{dt} \theta(t) + d_s \left( \frac{d}{dt} \phi(t) + \frac{d}{dt} \theta(t) \right) \cos(\phi(t) + \theta(t)) \\ \frac{dy_p}{dt} &= d_{la} \sin(\theta(t)) \frac{d}{dt} \theta(t) - d_s \left( \frac{d}{dt} \phi(t) + \frac{d}{dt} \theta(t) \right) \sin(\phi(t) + \theta(t))\end{aligned}$$

Členi Lagrangiana, kot je razvidno iz (7) do (12) (za kodo glej #defining potential (V) and kinetic energies (T))

$$T_1 = \frac{1}{2} m_{cw} \left( d_{sa}^2 \sin^2(\theta(t)) \left( \frac{d}{dt} \theta(t) \right)^2 + d_{sa}^2 \cos^2(\theta(t)) \left( \frac{d}{dt} \theta(t) \right)^2 \right)$$

$$T_2 = \frac{1}{2} m_p \left( \left( d_{la} \sin(\theta(t)) \frac{d}{dt} \theta(t) - d_s \left( \frac{d}{dt} \phi(t) + \frac{d}{dt} \theta(t) \right) \sin(\phi(t) + \theta(t)) \right)^2 + \left( -d_{la} \cos(\theta(t)) \frac{d}{dt} \theta(t) + d_s \left( \frac{d}{dt} \phi(t) + \frac{d}{dt} \theta(t) \right) \cos(\phi(t) + \theta(t)) \right)^2 \right)$$

$$T_3 = \frac{1}{2} I_p \left( \frac{d}{dt} \phi(t) \right)^2$$

$$T_4 = \frac{1}{2} I_a \left( \frac{d}{dt} \theta(t) \right)^2$$

$$V = g \left( m_a (h + (0.5 - d_{sa}) \cos(\theta(t))) + m_{cw} (-d_{cw} + d_{sa} \cos(\theta(t)) + h) + m_p (-d_{la} \cos(\theta(t)) + d_s \cos(\phi(t) + \theta(t)) + h) \right)$$

Da izračunamo Lagrangian, seštejemo vse kinetične energije in odštejemo potencialne (glej #defining lagrangian):

$$L = T_1 + T_2 + T_3 + T_4 - V$$

Kar izračuna:

$$\begin{aligned}
L = & \frac{1}{2}I_a \left( \frac{d}{dt} \theta(t) \right)^2 + \frac{1}{2}I_p \left( \frac{d}{dt} \phi(t) \right)^2 \\
& - g \left( m_a (h + (0.5 - d_{sa}) \cos(\theta(t))) + m_{cw} (-d_{cw} + d_{sa} \cos(\theta(t)) + h) \right. \\
& \left. + m_p (-d_{la} \cos(\theta(t)) + d_s \cos(\phi(t) + \theta(t)) + h) \right) \\
& + \frac{1}{2}m_{cw} \left( d_{sa}^2 \sin^2(\theta(t)) \left( \frac{d}{dt} \theta(t) \right)^2 + d_{sa}^2 \cos^2(\theta(t)) \left( \frac{d}{dt} \theta(t) \right)^2 \right) \\
& + \frac{1}{2}m_p \left( \left( d_{la} \sin(\theta(t)) \frac{d}{dt} \theta(t) - d_s \left( \frac{d}{dt} \phi(t) + \frac{d}{dt} \theta(t) \right) \sin(\phi(t) + \theta(t)) \right)^2 \right. \\
& \left. + \left( -d_{la} \cos(\theta(t)) \frac{d}{dt} \theta(t) + d_s \left( \frac{d}{dt} \phi(t) + \frac{d}{dt} \theta(t) \right) \cos(\phi(t) + \theta(t)) \right)^2 \right)
\end{aligned}$$

In kot je razvidno iz (15) in (16), moram izračunati in Euler-Lagrangeev odvod, ki ga bom rešil kasneje, v naslednjem poglavju (glej #lagrange euler derivatives for theta and phi).

$$\begin{aligned}
0 = & I_a \frac{d^2}{dt^2} \theta(t) + d_{sa}^2 m_{cw} \frac{d^2}{dt^2} \theta(t) + g \left( -d_{sa} m_{cw} \sin(\theta(t)) + m_a (d_{sa} - 0.5) \sin(\theta(t)) + \right. \\
& m_p (d_{la} \sin(\theta(t)) - d_s \sin(\phi(t) + \theta(t))) \left. \right) + m_p \left( d_{la}^2 \frac{d^2}{dt^2} \theta(t) + d_{la} d_s \sin(\phi(t)) \left( \frac{d}{dt} \phi(t) \right)^2 + \right. \\
& 2d_{la} d_s \sin(\phi(t)) \frac{d}{dt} \phi(t) \frac{d}{dt} \theta(t) - d_{la} d_s \cos(\phi(t)) \frac{d^2}{dt^2} \phi(t) - 2d_{la} d_s \cos(\phi(t)) \frac{d^2}{dt^2} \theta(t) + \\
& \left. d_s^2 \frac{d^2}{dt^2} \phi(t) + d_s^2 \frac{d^2}{dt^2} \theta(t) \right) \tag{17}
\end{aligned}$$

In:

$$\begin{aligned}
0 = & I_p \frac{d^2}{dt^2} \phi(t) - d_{la} d_s m_p \sin(\phi(t)) \left( \frac{d}{dt} \theta(t) \right)^2 - d_{la} d_s m_p \cos(\phi(t)) \frac{d^2}{dt^2} \theta(t) + d_s^2 m_p \frac{d^2}{dt^2} \phi(t) + \\
& d_s^2 m_p \frac{d^2}{dt^2} \theta(t) - d_s g m_p \sin(\phi(t) + \theta(t)) \tag{18}
\end{aligned}$$

### 2.3 Simulacija poti izstrelka v Pythonu

Potem, ko sem izpeljal Lagrangeeve enačbe gibanja, je postopek precej preprost. Edina težava je, da Python ne more neposredno rešiti diferencialnih enačb drugega ali višjega reda, ampak le prvega reda. Omogoča pa vsaj reševanje sklopljenih diferencialnih enačb prvega reda, zato sem to izkoristil. Obe Lagrangeevi enačbi sem pretvoril v štiri enačbe prvega reda, ki jih je mogoče rešiti s funkcijo `scipy.integrate.odeint` (Virtanen et al., 2020). Težava pa je v tem, da definicija drugega odvoda  $\phi$  vključuje tudi drugi odvod  $\theta$ , in obratno, kot je razvidno iz končnih enačb (Euler-Lagrangeevi odvodi, kot so opredeljeni v (17) in (18)), zato potrebujemo približek, da bi to lahko rešili. Približek sestoji iz zakona o navoru, ki sem ga povezal s kotnim pospeškom. Celotno kodo lahko vidite v prilogi.

Drži:

$$\Gamma = I \cdot \alpha \text{ kjer je } \alpha = \frac{d^2\theta}{dt^2}$$

In:

$$\Gamma = (\Gamma_1 + \Gamma_2 - \Gamma_3) = g \cdot \sin\theta \cdot (m_{cw} \cdot d_{sa} + \frac{1}{2} \cdot d_{sa}^2 \cdot m_a - \frac{1}{2} \cdot d_{la}^2 \cdot m_a)$$

Ki je opredeljen iz vseh sil, ki delujejo na ročico, in masnega razmerja ročice na vsaki strani vrtilišča, pri čemer se upošteva, da je ročica homogena in da:  $d_{la} + d_{sa} = 1 \text{ m [meter]}$

$$\Gamma_1 = m_{cw} \cdot g \cdot d_{sa} \cdot \sin\theta$$

$$\Gamma_2 = g \cdot \frac{d_{sa}}{d_{sa} + d_{la}} \cdot \frac{d_{sa}}{2} \cdot m_a \cdot \sin\theta = g \cdot \frac{d_{sa}^2}{2} \cdot m_a \cdot \sin\theta$$

$$\Gamma_3 = g \cdot \frac{d_{la}}{d_{sa} + d_{la}} \cdot \frac{d_{la}}{2} \cdot m_a \cdot \sin\theta = g \cdot \frac{d_{la}^2}{2} \cdot m_a \cdot \sin\theta$$

Iz tega sledi:

$$\frac{d^2\theta}{dt^2} = \frac{m_{cw} \cdot g \cdot d_{sa} \cdot \sin\theta + \frac{1}{2} \cdot d_{sa}^2 \cdot m_a \cdot g \cdot \sin\theta - \frac{1}{2} \cdot d_{la}^2 \cdot m_a \cdot g \cdot \sin\theta}{\frac{m_a}{12} + (0,5 - d_{sa})^2 \cdot m_a}$$

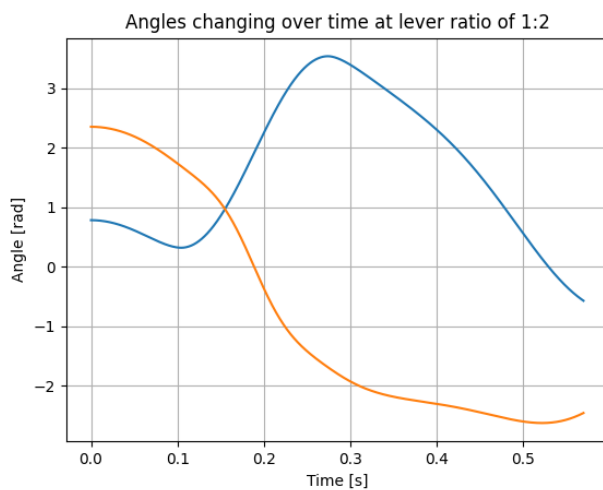
Približek tukaj temelji na dejstvu, da je masa izstrelka veliko manjša od mase protiuteži ali ročice, in privzemam, da je navor izstrelka na ročico ničeln. To seveda prinaša manjšo napako v procesu integracije, vendar je najpreprostejša rešitev za nezmožnost integracije sklopljenih diferencialnih enačb drugega reda. Del pod ulomkovo črto je vztrajnostni moment ročice, ki sem ga izpeljal iz Steinerjevega pravila. Privzel sem tudi, da je protiutež tukaj zunaj sistema, čeprav je povezana, vendar, če bi gledal kako deluje, če je del sistema, se potem vztrajnostni moment spreminja po

času, kar je težavno, medtem ko lahko nanj gledam le kot na neko umetno silo, ki deluje na ročico, kar je veliko preprostejše.

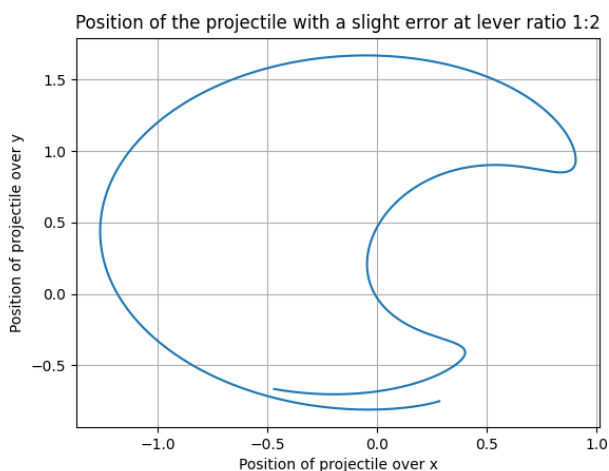
Rešitve, ki jih dobim v matriki sol, so vrednosti in odvodi obeh kotov. Nato moram ustvariti matriko vrednosti X in Y za risanje grafa, kar je precej preprosto, saj samo vstavim prvi in drugi stolpec matrike v enačbo za položaje X in Y izstrelka. Ker imam tudi enačbe X in Y komponent hitrosti, lahko izračunam tudi te in ugotovim, na katerem indeksu so absolutne vrednosti hitrosti enake in tako lahko izračunam celotne hitrosti.

Program izriše tri grafe, kote po času (graf 1), pot izstrelka z rahlo napako (zamaknjeni del) , (graf 2) in položaj točk, kjer so absolutne hitrosti približno enake (graf 3).

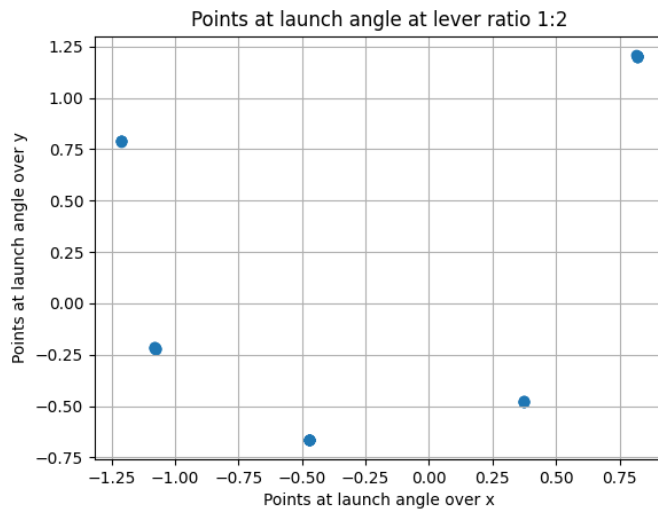
*Graf 1: Koti po času, teoretična napoved pri razmerju ročic 1:2,  $\theta$  je modra,  $\phi$  oranžen.*



*Graf 2: Položaj izstrelka, teoretična napoved pri razmerju ročic 1:2*

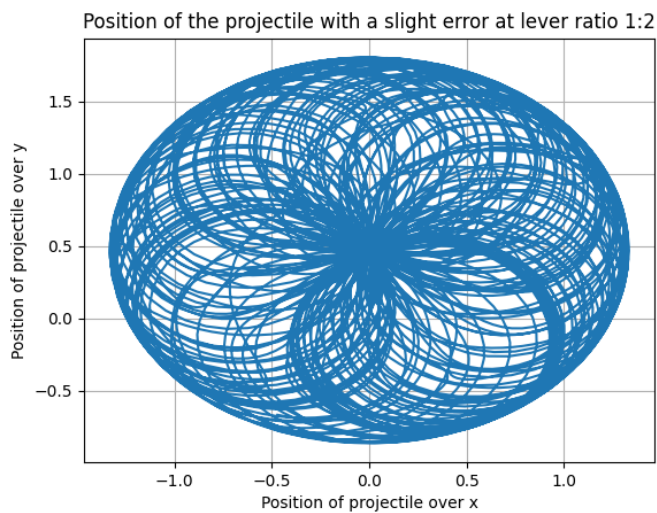


Graf 3: Graf točk, kjer je naklon poti izstrelka 45 stopinj glede na vodoravnico, teoretična napoved pri razmerju ročice 1:2



Vidimo lahko tudi, da se v daljšem časovnem obdobju napaka skoraj popolnoma zmanjša, saj so vse vrednosti znotraj elipsoida (graf 4). To je verjetno posledica prej opisanih približkov in Lagrangeeve mehanike, ki je že sama po sebi približek, ki se uporablja za opis kaotičnih sistemov.

Graf 4: Graf pozicije v daljšem časovnem obdobju, teoretična napoved



Graf 4 je relativno natančen, saj je sredina elipsoida približno v (0.0, 0.5), moral pa bi biti v (0.0, 0.47).

Podatke, ki jih izračuna program sem prikazal v tabeli, na naslednji strani.



Tabela 1: Teoretična napoved hitrosti pri vseh razmerjih vzvodov

Razmerje ročic	Predvidena hitrost [ $\text{m s}^{-1}$ ]
1:2	3,44
1:3	4,17
1:4	6,64
1:5	11,0
1:7	19,7
1:9	14,4

Iz tega sledi, da naj bi bilo najboljši razmerje ročic 1:7.

## 2.4 Izdelava modela trebušaja

Slika 3: Model trebušaja



Konstrukcija mojega modela trebušaja je bila precej preprosta. Konstrukcija je sestavljena iz okvirja iz masivnega lesa, ročice, protiuteži, ki je le poln kanister vode in prače, na katero je pritrjen projektil. Roka tega modela je  $(d_{La} + d_{Sa}) = 1.000 \text{ m} \pm 0,005 \text{ m}$  in teže  $m_a = 1.265 \pm 0,005 \text{ kg}$ . Vezni člen v tem primeru je  $d_{Cw} = 0,200 \pm 0,005 \text{ m}$ . Teža izstrelka je  $m_p = 0,050 \pm 0,005 \text{ kg}$ . Protiutež tehta  $m_{Cw} = 5.650 \pm 0,005 \text{ kg}$ , dolžina prače pa je bila pri vsakem poskusu spremenjena zaradi spreminjanja razmerja ročice. Dolžina niza je bila enaka dolžini daljše ročice ( $d_s = d_{la}$ ), saj je videti, da to

omogoča najboljše rezultate raziskovalcem na spletni strani SiegeEngine.com, ki so dosledno testirali različne prače, tako dolžine kakor same zasnove prače. (*SiegeEngine.com: How to Make Slings and Pouches*, n.d.).

Preizkušene dolžine ročic in njihova razmerja so bila naslednja (tabela na naslednji strani):

Tabela 2: dolžine ročic pri različnih razmerjih ročic

Razmerje	Dolžina daljše ročice - $d_{la}$ [m, $\pm 0,005$ m]	Dolžina krajše ročice - $d_{sa}$ [m, $\pm 0,005$ m]
1:2	0,666	0,333
1:3	0,750	0,250
1:4	0,800	0,200
1:5	0,833	0,166
1:7	0,875	0,125
1:9	0,900	0,100

Na trebuše je pritrjen tudi poseben mehanizem, ki omogoča sprostitve roke v stalnem začetnem položaju, 45 stopinj glede na vodoravnico. Projektil je trdno pritrjen na ročico s pomočjo prače in ne odleti. To je zato, ker me zanima samo hitrost, ki jo ima projektil, medtem ko je naklon poti 45 stopinj glede na vodoravnico. Trebuše je bil sprožen s palico, ki sem jo uporabil za razklop omenjenega mehanizma. Vse je bilo posneto s hitrostjo 30 sličic na sekundo z digitalnim fotoaparatom, nato pa izvoženo v LoggerPro, kjer sem uporabil movie tool, in LoggerPro je izračunal hitrosti ročice in projektila.

## 2.5 Zbiranje neobdelanih podatkov

Kot smo že povedali, so bili podatki zbrani s kamero Sony a7iii pri goriščni razdalji 28 mm, ki ustvari zelo ravno sliko z minimalnim popačenjem, da bi ustvarili natančne posnetke. Ker video ni bil hiter

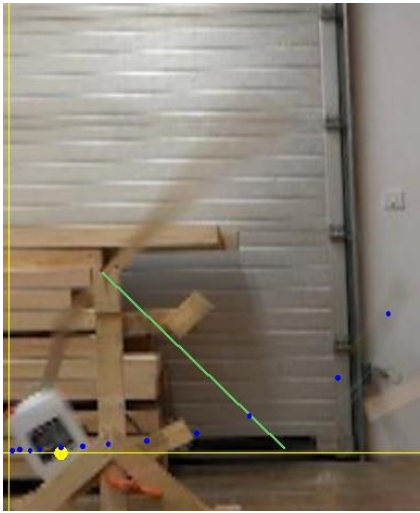
Slika 4: Primer interferenčnega vzorca



posnetek (high-speed), so zaradi hitrosti izstrelka nastali interferenčni vzorci (slika 4), a je bil srednji vrh še vedno dovolj dobro viden, da se je dalo najti lokacijo izstrelka. Položaju izstrelka sem dodelil  $\pm 0,05$  m napake zaradi moje napake, ki je nastala zaradi netočnega klikanja položaja izstrelka in zaradi interferenčnih vzorcev. Nato so bili videoposnetki uvoženi v LoggerPro, ki je verjetno najbolj dostopna programska oprema za fizikalno video analizo, in si zagotovil surove podatke za nadaljnjo

analizo. Celoten postopek v LoggerPro je zelo enostaven, najprej se video uvozi, nato se določi

Slika 5: Analiza podatkov v LoggerPro



ničla koordinatnega sistema, doda se merilo dolžine in nato izberemo ustrezne točke, torej položaj izstrelka, ki je bil obarvan svetlo zeleno (slika 3), da se zagotovi dobra vidljivost. Programska oprema nato samodejno izvleče položaje, hitrosti pa določi kot hitrost spremembe položaja na časovno enoto, ki je 0,033 s. Čas ima zanemarljivo napako, saj je bil določen po snemanju videoposnetka, zato ni reakcijskega časa ali karkoli drugega, kar bi bilo treba vključiti kot napako za čas. To pomeni, da ima hitrost enako relativno napako kot položaj. Na sliki 5 lahko vidimo primer videoposnetka, ki ga analiziramo v programu LoggerPro.

Rumene črte predstavljajo koordinatni sistem, ki ima ničlo v prvem položaju izstrelka, zelena črta je dolžinska mera, ki predstavlja razdaljo med koncem ročice in vrtiliščem, v tem primeru  $0.800 \pm 0.005$  m. Modre pike predstavljajo položaj izstrelka v vsaki sličici (Frame-u). Čeprav je slika morda videti zamegljena ali nizke ločljivosti, so glavne značilnosti zelo dobro vidne ob uporabi velikega monitorja, zato kakovost posnetkov ne vpliva na zbiranje podatkov. Pomembno je tudi omeniti, da se ničla koordinatnega sistema premakne iz ničle, ki je pod vrtiliščem, na ničlo, ki je začetni položaj izstrelka. To ne vpliva na veljavnost izračunanih podatkov. Zdaj bom pridobljene podatke predstavil v tabeli, za vsako časovno točko, x in y položaj izstrelka in daljše ročice ter njune x in y komponente hitrosti. Preostalih pet tabel je v Prilogi.

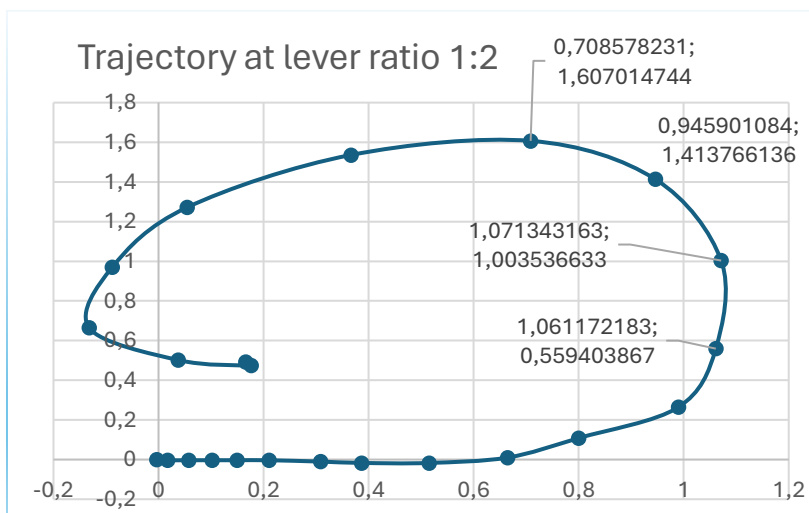
Tabela 3: Položaj izstrelka in hitrosti izstrelka in ročice pri razmerju ročice 1:2

Projektil					Ročica			
t [s]	X [m, ± 0,05 m]	Y [m, ± 0,05 m]	v <sub>x</sub> [m <sup>s-1</sup> , ± 0,05 m s <sup>-1</sup> ]	v <sub>y</sub> [m <sup>s-1</sup> , ± 0,05 m s <sup>-1</sup> ]	x [m, ± 0,05 m]	y [m, ± 0,05 m]	V <sub>x</sub> [m <sup>s-1</sup> , ± 0,05 m s <sup>-1</sup> ]	V <sub>y</sub> [M <sup>s-1</sup> , ± 0,05 m s <sup>-1</sup> ]
0,000	-0,003	0,000	0,787	-0,074	0,499	-0,442	0,662	0,646
0,033	0,017	-0,003	0,997	-0,041	0,522	-0,419	0,665	0,678
0,067	0,058	-0,003	1,228	-0,012	0,539	-0,409	0,770	0,949
0,100	0,102	-0,003	1,434	-0,008	0,565	-0,366	1,068	1,499
0,133	0,149	-0,003	1,745	-0,028	0,615	-0,303	1,188	1,926
0,167	0,210	-0,003	2,279	-0,096	0,655	-0,239	1,010	2,417
0,200	0,309	-0,010	2,695	-0,124	0,688	-0,146	0,621	3,019
0,234	0,386	-0,017	3,222	0,076	0,695	-0,027	0,182	3,364
0,267	0,515	-0,017	3,979	0,724	0,708	0,080	-0,538	3,639
0,300	0,665	0,010	4,265	2,116	0,675	0,209	-1,746	3,935
0,334	0,800	0,108	4,368	4,192	0,595	0,362	-3,060	3,673
0,367	0,990	0,264	3,349	7,079	0,469	0,462	-4,265	3,131
0,400	1,061	0,559	1,009	10,111	0,286	0,575	-4,622	2,4071
0,434	1,071	1,004	-1,922	10,905	0,140	0,628	-4,443	1,565
0,467	0,946	1,414	-5,216	7,772	0,000	0,652	-4,320	1,049
0,501	0,709	1,607	-7,829	1,922				
0,534	0,366	1,536	-8,532	-3,930				
0,567	0,054	1,271	-6,255	-7,347				
0,601	-0,088	0,970	-2,659	-8,064				
0,634	-0,132	0,665	1,087	-6,397				
0,667	0,037	0,502	3,051	-3,393				
0,701	0,176	0,475	2,275	-1,251				
0,734	0,166	0,492	0,984	-0,082				

## 2.6 Analiza podatkov

Podatke, ki sem jih pridobil/, sem izvozil v Excelovo preglednico, kjer sem lahko interpoliral podatke za znane manjše presledke med izmerjenimi točkami. Uporabljal sem programsko opremo SRS1 splines (*SRS1 Software, LLC Software in Digital Imaging Products and Services, n.d.*). programska oprema mi omogoča uporabo procesa interpolacije, imenovanega Bessel spline, ki ustvarja zelo dober nabor podatkovnih točk, ki jih je lahko uporabim kot nadaljnje izhodišče. Prva stvar, ki jo je treba najti, je čas, ko je odvod poti -1 zaradi specifik trebušaja. Če pogledamo graf poti za lažje razumevanje; lahko vidimo, da so točke na intervalih  $x \in [0.8, 1.2] \wedge y \in [0.8, 1.6]$ , torej zgornji desni del krivulje. Lepa značilnost LoggerPro je ravno sposobnost ugotavljanje hitrosti, ki so v bistvu le odvodi položaja. Čeprav hitrosti LoggerPro izračuna samo v smeri x in y, hitro ugotovimo, da ko je absolutna vrednost x komponente hitrosti približno enaka komponenti y, ima skupni odvod gradient -1 oziroma je vektor hitrosti 45 stopinj glede na vodoravnico, tako da v tem časovnem okviru dobimo precej natančen čas, ko naj bi izstrelka izstrelilo iz prače, in izračunamo njegovo skupno hitrost in kinetično energijo. Ker poznamo kot, s katerim se je gibanje začelo, torej 45 stopinj glede na vodoravnico, lahko izračunamo tudi skupno energijo, shranjeno v sistemu, ki je gravitacijska potencialna energija protiuteži. Da bi ugotovili učinkovitost trebušaja, bomo kinetično translacijsko energijo izstrelka razdelili s potencialno energijo protiuteži v začetnem položaju. Pokazal bom primer za razmerje vzvoda 1:2.

Graf 5: Pot izstrelka pri razmerju vzvoda 1:2



Iz grafa lahko zdaj vidimo 4 točke, ki nas zanimajo. Nekje med njimi mora biti naklon poti izstrelka 45 stopinj. Da pa ugotovim, kje je to, bom ustvaril tabelo s časi, ki bodo ločeni 0.002 sekunde in glede na te umetno ustvarjene vrednosti interpoliral hitrosti, in tako naj bi našli tudi čas, ko sta hitrosti x in y približno enaki.

Tabela 4: Točke in pripadajoče hitrosti v tabeli

t [s]	X [m]	Y [m]	V <sub>x</sub> [m s <sup>-1</sup> ]	V <sub>y</sub> [m s <sup>-1</sup> ]
0,400	1.061 ± 0,05	0,559 ± 0,05	1.009 ± 0,05	10.111 ± 0,90
0,434	1.071 ± 0,05	1.004 ± 0,05	-1.922 ± 0,09	10.905 ± 0,54
0,467	0,946 ± 0,05	1.414 ± 0,05	-5.216 ± 0,28	7.772 ± 0,27
0,501	0,709 ± 0,05	1.607 ± 0,05	-7.829 ± 0,55	1.922 ± 0,06

Na naslednji strani sem tako ustvaril tabelo časovnih vrednosti in interpoliranih vrednosti položaja ter hitrosti. Čas, ko so hitrosti približno enake, je obarvan rdeče.

Tabela 5: Vrednosti, dobljene z interpolacijo

t [s]	X [m, ± 0,05 m]	Y [m, ± 0,05 m]	v <sub>x</sub> [m s <sup>-1</sup> , varying error]	v <sub>y</sub> [m s <sup>-1</sup> , varying error]	t [s]	X [m, ± 0,05 m]	Y [m, ± 0,05 m]	v <sub>x</sub> [m s <sup>-1</sup> varying error]	v <sub>y</sub> [m s <sup>-1</sup> varying error]
0,400	1,061	0,559	1,009	10,111	0,452	1,017	1,243	-3,744	9,582
0,402	1,063	0,586	0,850	10,206	0,454	1,009	1,269	-3,951	9,368
0,404	1,064	0,613	0,689	10,302	0,456	1,000	1,294	-4,157	9,144
0,406	1,066	0,639	0,526	10,399	0,458	0,991	1,318	-4,359	8,911
0,408	1,068	0,666	0,362	10,494	0,460	0,982	1,342	-4,558	8,669
0,410	1,070	0,692	0,196	10,586	0,462	0,972	1,364	-4,753	8,420
0,412	1,072	0,719	0,028	10,673	0,464	0,962	1,385	-4,943	8,165
0,414	1,074	0,745	-0,141	10,755	0,466	0,951	1,404	-5,126	7,904
0,416	1,076	0,771	-0,312	10,829	0,468	0,940	1,423	-5,304	7,637
0,418	1,077	0,797	-0,484	10,894	0,470	0,929	1,440	-5,478	7,362
0,420	1,078	0,823	-0,658	10,948	0,472	0,918	1,457	-5,650	7,078
0,422	1,079	0,849	-0,834	10,990	0,474	0,906	1,473	-5,819	6,785
0,424	1,079	0,875	-1,011	11,019	0,476	0,894	1,488	-5,985	6,482
0,426	1,079	0,901	-1,190	11,033	0,478	0,881	1,502	-6,148	6,170
0,428	1,078	0,927	-1,370	11,030	0,480	0,868	1,515	-6,308	5,849
0,430	1,077	0,952	-1,553	11,008	0,482	0,855	1,528	-6,466	5,519
0,432	1,074	0,978	-1,736	10,967	0,484	0,841	1,539	-6,621	5,179
0,434	1,071	1,004	-1,922	10,905	0,486	0,827	1,550	-6,774	4,831
0,436	1,067	1,029	-2,111	10,822	0,488	0,812	1,561	-6,923	4,473
0,438	1,063	1,056	-2,305	10,721	0,490	0,797	1,570	-7,070	4,106
0,440	1,058	1,082	-2,504	10,603	0,492	0,782	1,579	-7,214	3,729
0,442	1,053	1,109	-2,706	10,469	0,494	0,766	1,586	-7,355	3,344
0,444	1,046	1,136	-2,911	10,319	0,496	0,750	1,593	-7,494	2,949
0,446	1,040	1,163	-3,119	10,154	0,498	0,734	1,599	-7,630	2,545
0,448	1,033	1,190	-3,327	9,976	0,500	0,717	1,605	-7,763	2,132
0,450	1,025	1,217	-3,536	9,785	0,502	0,700	1,609	-7,894	1,711

V tabeli 6 bom predstavil izračunane vrednosti za vsa različna razmerja ročic.

Tabela 6: Položaji ob izstrelitvi in hitrosti izstrelka ter ročice pri različnih razmerjih ročic

Razmerje vzvoda	t [s]	X [m, ± 0,05 m]	Y [m, ± 0,05 m]	Vx [ms <sup>-1</sup> ]	ΔVx [m s <sup>-1</sup> ]	Vy [m s <sup>-1</sup> ]	ΔVy [m s <sup>-1</sup> ]	Va [m s <sup>-1</sup> ]	ΔVa [m s <sup>-1</sup> ]
1:2	0,478	0,881	1,528	-6,188	0,35	6,083	0,20	4,39	0,67
1:3	0,480	1,299	1,917	-5,458	0,21	5,421	0,14	2,35	0,33
1:4	0,482	1,011	1,636	-7,391	0,37	7,203	0,22	2,65	0,36
1:5	0,438	1,173	1,854	-7,075	0,30	7,254	0,20	3,16	0,40
1:7	0,437	1,169	1,972	-8,986	0,38	8,800	0,22	4,33	0,60
1:9	0,612	1,349	1,992	-7,159	0,27	7,305	0,18	1,88	0,23

Učinkovitost trebušaja je v tem primeru izračunana zelo preprosto. Gre za to, koliko razpoložljive energije se je preneslo v kinetično translacijsko energijo izstrelka. To lahko izračunamo kot:

$$\eta = \frac{E_{kp}}{E_{pcw}}$$

Glede na podatke, ki sem jih pridobil, lahko izračunam vse te vrednosti. Predstavljene vrednosti so za razmerje vzvodov 1:2.

$$E_{kp} = \frac{1}{2} \cdot m_p \cdot v_p^2 = \frac{1}{2} \cdot 0,05 \cdot (v_x^2 + v_y^2) = \frac{1}{2} \cdot 0,05 \cdot (-6,188^2 + 6,083^2) = 1,88 \pm 0,20 J$$

Napaka je bila določena kot:

$$\Delta E_{kp} = \left( \frac{\Delta m_p}{m_p} + \frac{\Delta v_x}{v_x} + \frac{\Delta v_y}{v_y} \right) \cdot E_{kp} = \left( \frac{0,005 kg}{0,05 kg} + \frac{0,35 m s^{-1}}{6,188 m s^{-1}} + \frac{0,20 m s^{-1}}{6,083 m s^{-1}} \right) \cdot 1,88 J = 0,20 J$$

Nato se potencialna energija protiuteži izračuna kot:

$$E_{pcw} = m_{cw} \cdot g \cdot (d_{sa} \cdot \cos \theta_{initial} - d_{sa} \cdot \cos \theta_{final}), \text{ where } \theta_{initial} = \frac{\pi}{4} \text{ and } \theta_{final} = \pi$$

$$E_{pcw} = m_{cw} \cdot g \cdot d_{sa} \cdot \left( 1 + \frac{\sqrt{2}}{2} \right) = 5,65 kg \cdot 9,81 m s^{-2} \cdot 0,666 m \cdot \left( 1 + \frac{\sqrt{2}}{2} \right) = 31,2 \pm 0,5 J$$

Nato se napaka izračuna kot:

$$\Delta E_{pcw} = \left( \frac{\Delta d_{sa}}{d_{sa}} + \frac{\Delta m_{cw}}{m_{cw}} \right) \cdot E_{pcw} = \left( \frac{0,005 m}{0,666 m} + \frac{0,005 kg}{5,650 kg} \right) \cdot 31,2 J = 0,5 J$$

In učinkovitost je:



$$\eta = \frac{E_{kp}}{E_{pcw}} = \frac{1,88}{31,2} = 0,06 \pm 0,01$$

Napaka se določi kot:

$$\Delta\eta = \left( \frac{\Delta E_{kp}}{E_{kp}} + \frac{\Delta E_{pcw}}{E_{pcw}} \right) \cdot \eta = \left( \frac{0,20J}{1,88J} + \frac{0,5J}{31,2J} \right) \cdot 0,06 = 0,01$$

Tukaj sem predstavil vse različne  $\eta$  vrednosti v tabeli.

Tabela 7: Učinkovitost pri različnih razmerjih ročic

	01:02	01:03	01:04	01:05	01:07	01:09
$\eta$ []	0,06	0,06	0,14	0,16	0,33	0,28
$\Delta\eta$ []	0,01	0,01	0,03	0,03	0,07	0,06

Tukaj lahko vidimo, da je najbolj optimalno razmerje ročic za trebuše z enojno ročico 1:7, kot sem bil napovedal s programom v Python-u.

## 2.7 Vrednotenje analiziranih podatkov in razprava

Mislím, da sem z vsemi poglavji do zdaj uspešno dokazal, da je najboljše razmerje ročic za trebuše 1:7. Zdaj bi rad komentiral še izmerjene vrednosti, vire napak in možne izboljšave.

### 2.7.1 Izmerjene vrednosti

Opazno je, da je bilo v tej raziskavi veliko izmerjenih podatkov, vendar ni ponovitev poskusov. Čeprav sem opravil tri preliminarne teste za razmerje vzvodov 1:2, sem se odločil, da tega ne bom vključil v raziskovalno nalogo, ker so bile vrednosti praktično enake, vse znotraj  $\pm 0,05$  m napake. Tudi zato sem se odločil, da ne bom opravljal ponovitev pri manjših razmerjih ročic, saj je preliminarni test pokazal, da je poskus ponovljiv. To tudi pomeni, da so vrednosti  $x$  in  $y$ , ki sem jih dobil, relevantne. Zdaj, če primerjam podatke, kar sem posnel in kar sem predvidel s Python-om, opažam, da se hitrosti zelo razlikujejo (glej tabeli 1 in 7). Vir tega je lahko približek z navorom in kotnim pospeškom, s čimer sem zaobšel pomanjkanje definicije drugega odvoda v Euler-Lagrangeevem odvodu (glej poglavje 2.2). Predvideni podatki za razmerje ročic 1:2 so približno 2,5-krat premajhni, približno dvakrat za 1:3, podobno za 1:4, napoved za 1:5 je presenetljivo točna, medtem ko je za 1:7 in 1:9 napoved 1,5-2 krat prevelika. Čeprav za to morda obstaja možna razlaga, je sam ne najdem. Kljub temu je programska napoved pravilno napovedala vrh pri razmerju ročic 1: 7. Zaključek tu je, da če bo na neki točki prišlo do maksimuma, ker je to navsezadnje le vrsta ekstremalnega problema, ga lahko programska oprema reši, vendar bo koristna le do te mere, da pokaže, kje je ta maksimum in ne tudi kakšna je njegova dejanska vrednost.

### 2.7.2 Viri napak

Ugotavljam, da obstajata dva glavna vira napak v mojih eksperimentalno pridobljenih podatkih. Ena je kamera, druga pa interpolacijski algoritem, za katerega sem privzel, da reproducira podatke z isto napako, kot so bili izmerjeni, vendar, če sem morda malo zamočil z označevanjem točke, lahko Besselov interpolacijski algoritem, ki je eden boljših, povzroči nekaj napake, vendar praktičnega načina za oceno te napake ni, zato sem ga zanemaril. Drugi vir napak je uporaba kamere, ki je zaradi slabih svetlobnih pogojev lahko snemala le pri 30 sličicah na sekundo. Z več podatkovnimi točkami, ki bi jih lahko dobili s kamero, ki podpira več sličic na sekundo pri relativno nizkih osvetlitvah, ali pa s specializirano hitro kamero interpolacijo izvedli veliko bolje, oziroma pri nekih 500-1000 sličicah na sekundo interpolacije niti nebi več rabili.

### 2.7.3 Izboljšave

Kot sem že omenil, bi bila ena glavnih izboljšav uporaba hitre kamere. Če bi uporabil hitro kamero, bi tako ne rabil interpolacije in odstranil enega od možnih virov napak. Tudi s fotoaparatom bi lahko dobil do 4x več sličic, vendar ne v svetlobnih pogojih v garaži, kjer sem snemal videoposnetke. Morda z zelo močnim sistemom osvetlitve, do katerega nisem imel dostopa. Pomembno je tudi, da se uporabi ozadje, ki je v barvnem kontrastu z izstrelkom, da je izstrellek dobro viden.

## 3 ZAKLJUČEK

V zaključku sem večinoma odgovoril na vsa vprašanja, ki jih sem jih postavil v to raziskovalno nalogo. Želel sem pokazati, kako lahko podrobno znanje fizike in objektivno usmerjeno programiranje uporabimo za napovedovanje izida zapletenega problema v resničnem svetu. S Python-om sem najprej izpeljal enačbe gibanja trebušaja, nato pa sem z uporabo Lagrangeeve mehanike in Euler-Lagrangeevega odvoda uspešno narisal graf poti izstrelka, ki temelji na dvojnem nihalu. Iz tega sem izračunal tudi hitrosti izstrelka v teoretičnem okolju. Prav tako sem moral najti rešitev, (približek z navori), sicer problem ne bi bil rešljiv. Nato sem zgradil tudi majhen trebuše, ga preizkusil in posnel s kamero. Dobljene rezultate sem nato analiziral in uporabil za nadaljnje izračune. Rezultati izračunov podpirajo mojo tezo. Praktično pridobljeni podatki prav tako kažejo, da so hitrosti, ki jih je ustvarila simulacija v Python-u, za približno faktor dva, nekoliko manj prevelike ali premajhne od realnih vrednosti. Kot je bilo že navedeno, v 2.7.1 ne morem podati verodostojne razlage, zakaj bi to lahko bilo tako. Ena hipotetična razlaga je, da je model v Python-u precej ekstremističen v smislu, da reproducira podatke, ki so premajhni ali preveliki zaradi pomnoženih napak v procesu integracije. Za raziskovanje čemu bi bilo tako nisem posvečal

prevelike pozornosti, ker se v nalogi ne sprašujem, kakšna je izhodna hitrost izstrelka, pač pa kje je najboljše razmerje ročic.

## 4 VIRI IN LITERATURA

de Jong, R. (2020, January 31). *Modelling the movements of a counterweight trebuchet while firing*. TU Delft.

Kalif, W. (2017, June). *The string and sling*. Projects.raspberrypi.org.  
<https://projects.raspberrypi.org/en/projects/trebuchet/7>

*Lagrangian mechanics*. (2024, February 22). Wikipedia.  
[https://en.wikipedia.org/wiki/Lagrangian\\_mechanics](https://en.wikipedia.org/wiki/Lagrangian_mechanics)

*Lever*. (2024, February 8). Wikipedia. <https://en.wikipedia.org/wiki/Lever>

Meurer, A., Smith, C., Paprocki, M., Čertík, O., Kirpichev, S., Rocklin, M., Kumar, A., Ivanov, S., Moore, J., Singh, S., Rathnayake, T., Vig, S., Granger, B., Muller, R., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M., Terrel, A., Roučka, Š., Saboo, A., Fernando, I., Kulal, S., Cimrman, R., & Scopatz, A. (2017). SymPy: symbolic computing in Python. *PeerJ Computer Science*, 3, e103.

Perez, F., & Granger, B. (2007). IPython: a System for Interactive Scientific Computing. *Computing in Science and Engineering*, 9(3), 21–29.

*Siege Engine.com: How to Make Slings and Pouches*. (n.d.). <http://www.siege-engine.com/SlingsAndPouches.shtml>

*SRS1 Software, LLC software and digital imaging products and services*. (n.d.).  
<https://www.srs1software.com/>

Van Rossum, G., & Drake Jr, F. (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., . . . Vázquez-Baeza, Y. (2020, February 3). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272. <https://doi.org/10.1038/s41592-019-0686-2>

*Whipcracking*. (2023, September 30). Wikipedia. <https://en.wikipedia.org/wiki/Whipcracking>

## 5 PRILOGE

### 5.1 Tabele neobdelanih podatkov

Tabela za razmerje vzvodov 1:2 je že v raziskovalni nalogi (glej Tabela 3).

Tabela 8: Položaji in hitrosti izstrelkov in ročice pri razmerju ročic 1:3

Projektil					Ročica			
t [s]	X [m, ± 0,05 m]	Y [m, ± 0,05 m]	v <sub>x</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	v <sub>y</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	x [m, ± 0,05 m]	y [m, ± 0,05 m]	v <sub>x</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	v <sub>y</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]
0,000	0,004	-0,004	1,650	0,037	0,557	-0,498	1,099	0,988
0,033	0,056	-0,004	1,796	0,102	0,588	-0,474	1,242	1,366
0,067	0,112	0,000	2,133	0,206	0,641	-0,425	1,301	1,935
0,100	0,197	0,007	2,531	0,361	0,682	-0,345	1,156	2,527
0,133	0,279	0,026	2,965	0,464	0,717	-0,251	0,926	3,013
0,167	0,391	0,045	3,566	0,458	0,756	-0,143	0,349	3,487
0,200	0,518	0,048	4,208	0,742	0,749	-0,035	-0,483	4,339
0,234	0,663	0,075	5,100	1,699	0,721	0,164	-1,340	4,717
0,267	0,868	0,149	5,667	3,223	0,665	0,310	-2,360	4,270
0,300	1,057	0,264	5,531	5,471	0,571	0,446	-3,428	3,667
0,334	1,266	0,506	4,472	7,863	0,411	0,557	-3,770	2,922
0,367	1,370	0,819	2,855	9,396	0,292	0,651	-3,332	1,936
0,400	1,460	1,173	0,916	9,424	0,198	0,686	-3,007	0,987
0,434	1,448	1,463	-1,576	8,670	0,094	0,700	-2,719	0,487
0,467	1,366	1,787	-4,366	6,667	0,021	0,703	-2,444	0,226
0,501	1,154	1,962	-7,027	3,133				
0,534	0,853	1,985	-8,354	-0,216				
0,567	0,577	1,929	-8,656	-2,976				
0,601	0,242	1,769	-7,789	-5,016				
0,634	0,026	1,564	-5,847	-6,075				
0,667	-0,160	1,348	-3,208	-6,427				
0,701	-0,212	1,124	0,071	-6,325				
0,734	-0,153	0,920	3,022	-6,011				
0,767	0,048	0,704	4,377	-5,070				
0,801	0,194	0,599	4,534	-4,001				

Tabela 9: Položaji in hitrosti izstrelkov in ročice pri razmerju ročic 1:4

Projektil					Ročica			
t [s]	X [m, ± 0,05 m]	Y [m, ± 0,05 m]	$v_x$ [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	$v_y$ [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	x [m, ± 0,05 m]	y [m, ± 0,05 m]	$v_x$ [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	$v_y$ [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]
0	0,009	0,006	0,741	0,053	0,579	-0,549	0,568	0,661
0,033	0,032	0,009	0,858	0,022	0,599	-0,532	0,590	0,860
0,067	0,063	0,006	1,022	0,036	0,609	-0,496	0,815	1,120
0,1	0,092	0,009	1,398	0,111	0,649	-0,462	1,126	1,498
0,133	0,155	0,016	1,837	0,139	0,685	-0,409	1,360	2,258
0,167	0,218	0,019	2,239	0,176	0,755	-0,299	1,159	2,785
0,2	0,294	0,025	2,935	0,302	0,768	-0,233	0,731	3,575
0,234	0,408	0,035	3,879	0,629	0,798	-0,070	0,337	4,547
0,267	0,557	0,057	4,750	1,367	0,808	0,096	-0,602	4,873
0,3	0,712	0,108	5,703	2,790	0,772	0,263	-1,970	4,804
0,334	0,975	0,222	5,482	4,937	0,682	0,412	-3,503	4,662
0,367	1,124	0,412	3,815	7,804	0,532	0,599	-4,761	3,804
0,4	1,244	0,766	1,284	9,752	0,336	0,692	-4,962	2,171
0,434	1,235	1,105	-2,094	10,163	0,160	0,725	-4,015	1,080
0,467	1,130	1,494	-6,021	8,731	0,067	0,742	-2,962	0,547
0,501	0,795	1,782	-8,756	4,479	0,000	0,748	-2,327	0,307
0,534	0,465	1,805	-9,060	-0,377				
0,567	0,165	1,722	-7,929	-4,383				
0,601	-0,098	1,510	-5,479	-7,860				
0,634	-0,244	1,152	-1,729	-9,600				
0,667	-0,215	0,766	1,990	-7,923				
0,701	-0,076	0,576	4,427	-4,847				
0,734	0,174	0,469	4,409	-2,202				
0,767	0,231	0,507	3,158	-0,209				

Tabela 10: Položaji in hitrosti izstrelkov in ročice pri razmerju ročic 1:5

Projektil					Ročica			
t [s]	X [m, ± 0,05 m]	Y [m, ± 0,05 m]	v <sub>x</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	v <sub>y</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	x [m, ± 0,05 m]	y [m, ± 0,05 m]	v <sub>x</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	v <sub>y</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]
0,000	0,000	0,000	2,106	0,147	0,624	-0,567	1,014	1,254
0,033	0,074	0,007	2,067	0,099	0,654	-0,533	1,153	1,562
0,067	0,123	0,004	2,315	0,133	0,694	-0,469	1,350	1,915
0,100	0,228	0,014	2,690	0,203	0,748	-0,409	1,430	2,396
0,133	0,312	0,021	2,887	0,207	0,794	-0,318	1,293	3,121
0,167	0,382	0,028	4,030	0,253	0,845	-0,198	0,798	3,797
0,200	0,586	0,021	5,141	0,891	0,855	-0,064	0,039	4,448
0,234	0,768	0,077	5,254	2,062	0,851	0,104	-0,948	4,938
0,267	0,898	0,119	6,112	4,448	0,801	0,275	-2,250	5,103
0,300	1,196	0,351	6,000	7,463	0,714	0,456	-3,789	4,710
0,334	1,381	0,649	3,517	9,661	0,530	0,613	-4,720	3,609
0,367	1,452	1,006	0,165	11,227	0,365	0,704	-4,630	2,286
0,400	1,396	1,441	-3,482	10,846	0,201	0,751	-3,834	1,379
0,434	1,199	1,823	-6,687	7,727	0,114	0,781	-3,100	0,894
0,467	0,957	1,992	-9,910	3,317	0,023	0,798	-2,825	0,644
0,501	0,550	2,020	-12,895	-0,782				
0,534	-0,028	1,967	-12,120	-5,470				
0,567	-0,382	1,658	-7,238	-9,636				
0,601	-0,508	1,217	-1,961	-10,512				
0,634	-0,491	0,856	2,715	-8,190				
0,667	-0,295	0,649	5,908	-4,875				
0,701	-0,028	0,561	7,217	-2,246				
0,734	0,228	0,579	7,675	-0,545				
0,767	0,231	0,507	3,158	-0,209				

Tabela 11: Položaji in hitrosti izstrelkov in ročice pri razmerju ročic 1:7

Projektil					Ročica			
t [s]	X [m, ± 0,05 m]	Y [m, ± 0,05 m]	v <sub>x</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	v <sub>y</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	x [m, ± 0,05 m]	y [m, ± 0,05 m]	v <sub>x</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	v <sub>y</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]
0,000	0,000	-0,004	1,855	-0,026	0,630	-0,607	0,633	0,858
0,033	0,061	-0,008	1,862	0,068	0,647	-0,580	0,796	0,983
0,067	0,126	0,000	1,901	0,133	0,681	-0,550	0,969	1,257
0,100	0,180	0,004	2,262	0,144	0,707	-0,506	1,211	1,797
0,133	0,256	0,008	3,250	0,242	0,761	-0,429	1,415	2,327
0,167	0,375	0,015	4,763	0,498	0,821	-0,345	1,102	2,763
0,200	0,597	0,031	5,754	1,168	0,831	-0,248	0,818	3,303
0,234	0,803	0,088	5,600	2,201	0,872	-0,134	0,540	4,066
0,267	0,933	0,142	6,157	4,363	0,882	0,034	-0,172	4,579
0,300	1,228	0,356	5,963	7,228	0,878	0,184	-1,411	4,605
0,334	1,404	0,635	3,639	9,867	0,788	0,339	-2,696	4,579
0,367	1,495	1,013	0,250	12,230	0,684	0,503	-3,569	4,114
0,400	1,457	1,503	-4,352	12,346	0,543	0,630	-4,060	3,074
0,434	1,197	1,943	-8,667	9,199	0,396	0,714	-3,975	1,881
0,467	0,838	2,188	-11,627	3,726	0,265	0,738	-3,494	1,177
0,501	0,375	2,188	-12,798	-1,923	0,158	0,784	-2,898	0,791
0,534	-0,103	2,038	-11,087	-6,834	0,080	0,795	-2,531	0,367
0,567	-0,436	1,694	-6,738	-10,073	0,003	0,795	-2,383	0,146
0,601	-0,574	1,277	-1,598	-10,418				
0,634	-0,524	0,945	3,000	-8,724				
0,667	-0,333	0,654	5,995	-5,589				
0,701	-0,065	0,562	7,345	-2,211				
0,734	0,199	0,616	7,848	0,073				



Tabela 12: Položaji in hitrosti izstrelkov in ročice pri razmerju ročic 1:9

Projektil					Ročica			
t [s]	X [m, ± 0,05 m]	Y [m, ± 0,05 m]	v <sub>x</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	v <sub>y</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	x [m, ± 0,05 m]	y [m, ± 0,05 m]	v <sub>x</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]	v <sub>y</sub> [m s <sup>-1</sup> , ± 0,05 m s <sup>-1</sup> ]
0,000	-0,007	0,000	1,008	-0,058	0,659	-0,616	0,630	0,702
0,033	0,024	-0,003	1,078	-0,011	0,679	-0,593	0,682	0,742
0,067	0,065	0,000	1,152	-0,007	0,702	-0,570	0,788	0,867
0,100	0,100	-0,003	1,270	-0,011	0,731	-0,541	0,899	1,154
0,133	0,138	0,000	1,719	-0,031	0,757	-0,498	1,103	1,583
0,167	0,217	-0,007	2,134	-0,020	0,810	-0,436	1,200	2,051
0,200	0,283	-0,003	2,503	0,054	0,839	-0,357	1,224	2,451
0,234	0,386	0,000	2,833	0,080	0,885	-0,272	1,316	2,886
0,267	0,476	-0,003	3,131	0,299	0,947	-0,174	0,824	3,585
0,300	0,582	0,010	3,765	0,858	0,957	-0,036	-0,182	4,271
0,334	0,713	0,045	4,783	1,787	0,924	0,121	-0,982	4,630
0,367	0,903	0,114	5,725	3,146	0,888	0,292	-1,702	4,428
0,400	1,117	0,252	5,869	4,591	0,816	0,429	-2,491	3,742
0,434	1,327	0,451	5,013	5,386	0,718	0,534	-3,156	3,210
0,467	1,458	0,576	3,781	7,059	0,603	0,636	-3,675	2,862
0,501	1,596	0,903	1,974	9,177	0,459	0,738	-3,738	2,215
0,534	1,606	1,251	-0,406	9,672	0,328	0,793	-3,078	1,334
0,567	1,572	1,558	-3,032	9,370	0,256	0,820	-2,357	0,622
0,601	1,423	1,896	-6,165	8,123	0,184	0,823	-1,975	0,276
0,634	1,161	2,164	-9,120	5,060	0,125	0,833	-1,664	0,140
0,667	0,762	2,254	-10,204	1,307	0,072	0,833	-1,366	0,038
0,701	0,410	2,202	-9,270	-1,122	0,033	0,833	-1,065	0,012
0,734	0,145	2,140	-8,072	-2,417	0,010	0,833	-0,849	0,000
0,767	-0,100	2,051	-7,249	-3,768				
0,801	-0,383	1,909	-5,374	-5,594				
0,834	-0,472	1,672	-3,118	-7,247				
0,868	-0,569	1,392	-1,300	-7,918				
0,901	-0,576	1,117	0,922	-7,653				

0,934	-0,510	0,862	3,119	-6,404
0,968	-0,352	0,662	4,822	-4,312
1,001	-0,169	0,565	5,829	-2,089
1,034	0,062	0,548	6,013	-0,483
1,068	0,245	0,582	5,738	0,460

## 5.2 Koda, uporabljena za izpeljavo enačb

```
# importing SymPy library
import sympy as sym

#defining variables used in computation.
t = sym.symbols('t')
m_a, m_p, m_cw, g, d_la, d_sa, d_s, h, d_cw, I_p, I_a = sym.symbols('m_a m_p
m_cw g d_la d_sa d_s h d_cw I_p I_a', real=True, positive=True)

theta, phi = sym.symbols(r'\theta \phi', cls=sym.Function)

theta = theta(t)
phi = phi(t)

#defining the equations of position
x_cw = d_sa*sym.sin(theta)
y_cw = (h-d_cw) + d_sa*sym.cos(theta)

x_p = d_s*sym.sin(theta+phi)-d_la*sym.sin(theta)
y_p = h + d_s*sym.cos(theta+phi) - d_la*sym.cos(theta)

x_a = (0.5-d_sa)*sym.sin(theta)
y_a = h + (0.5-d_sa)*sym.cos(theta)

#defining first and second derivatives of theta and phi
dtheta = sym.diff(theta, t)
ddtheta = sym.diff(dtheta, t)

dphi = sym.diff(phi, t)
ddphi = sym.diff(dphi, t)

#defining derivatives (velocities) of counterweight and projectile position
dx_cw = sym.diff(x_cw, t)
dy_cw = sym.diff(y_cw, t)

dx_p = sym.diff(x_p, t)
dy_p = sym.diff(y_p, t)
```

```

#defining potential (V) and kinetic energies (T)

#linear kinetic
T1 = 0.5*(m_cw*((dx_cw**2)+(dy_cw**2))) #counterweight
T2 = 0.5*m_p*((dx_p**2)+(dy_p**2))     #projectile

#rotational kinetic
T3 = 0.5*I_p*dphi**2                    #projectile
T4 = 0.5*I_a*dtheta**2                  #counterweight

#potential energy of both counterweight and projectile
V = g*((m_cw*y_cw)+(m_p*y_p)+(m_a*y_a))

#defining the lagrangian
L = T1 + T2 + T3 + T4 - V

#lagrange euler derivatives for theta and phi
LE1 = sym.diff(sym.diff(L, dtheta), t) - sym.diff(L, theta)
LE2 = sym.diff(sym.diff(L, dphi), t) - sym.diff(L, phi)

#simplifying the equation, removing zero components, etc
LE3 = LE1.simplify()
LE4 = LE2.simplify()

```

### 5.3 Koda, ki se uporablja za integracijo enačb

```

from scipy.integrate import odeint
import matplotlib.pyplot as plt
import numpy as np

ma = 1.265
mp = 0.05
mcw = 5.65
g = 9.81
dla = 0.666
dsa = 0.333

ds = dla
h = np.cos(np.pi/4)*dla
dcw = 0.2
Ia = (ma / 12) + ((0.5-dsa)**2)*ma
Ip = ds*mp**2

def dSdt(t, S):
    theta, fi, wt, wf = S
    return [wt,
            wf,
            (-g*(dsa*mcw*np.sin(theta)-mp*(dla*np.sin(theta)-
ds*np.sin(theta+fi)))+mp*(dla*ds*np.sin(theta)*wt**2)+wt*wf**2*dla*ds*np.sin(f

```

```

i)-dla*np.cos(fi)*(dla*ds*mp*np.sin(fi)*wt**2-
dla*ds*mp*np.cos(fi)*((mcw*g*dsa*np.sin(theta)+dsa*ma*(dsa/2)*g*np.sin(theta)-
dla*(dla/2)*np.sin(theta)*ma*g)/Ia)+ds**2*mp*((mcw*g*dsa*np.sin(theta)+dsa*ma*
(dsa/2)*g*np.sin(theta)-dla*(dla/2)*np.sin(theta)*ma*g)/Ia)-
ds*g*mp*np.sin(theta+fi))/(-Ip-ds**2*mp))/(-Ia-dsa**2*mcw-mp*dla**2-
2*dla*ds*np.cos(fi)+ds),
        (dla*ds*mp*np.sin(fi)*wt**2-
dla*ds*mp*np.cos(fi)*((mcw*g*dsa*np.sin(theta)+dsa*ma*(dsa/2)*g*np.sin(theta)-
dla*(dla/2)*np.sin(theta)*ma*g)/Ia)+ds**2*mp*((mcw*g*dsa*np.sin(theta)+dsa*ma*
(dsa/2)*g*np.sin(theta)-dla*(dla/2)*np.sin(theta)*ma*g)/Ia)-
ds*g*mp*np.sin(theta+fi))/(-Ip-ds**2*mp)]

theta_0 = (np.pi/4)
fi_0 = 3*(np.pi/4)
wt_0 = 0
wf_0 = 0

S_0 =(theta_0, fi_0, wt_0, wf_0)

t_vals = np.linspace(0, 10, 10000)

sol = odeint(dSdt, y0=S_0, t=t_vals, tfirst=True)

plt.plot(t_vals, sol.T[0], label='Theta')
plt.plot(t_vals, sol.T[1], label='Phi')
plt.xlabel('Time [s]')
plt.ylabel('Angle [rad]')
plt.title('Angles changing over time at lever ratio of 1:2')
plt.grid(True)
plt.show()

xp_vals = ds*np.sin(sol.T[0]+sol.T[1])-dla*np.sin(sol.T[0])
yp_vals = h+ds*np.cos(sol.T[0]+sol.T[1])-dla*np.cos(sol.T[0])

plt.plot(xp_vals, yp_vals)
plt.xlabel('Position of projectile over x')
plt.ylabel('Position of projectile over y')
plt.title('Position of the projectile with a slight error at lever ratio 1:2')
plt.grid(True)
plt.show()

dxdpt = (ds*np.cos(sol.T[0]+sol.T[1])-dla*np.cos(sol.T[0]))*sol.T [2] +
ds*np.cos(sol.T[0]+sol.T[1])*sol.T[3]
dypdt = (-ds*np.cos(sol.T[0]+sol.T[1])+dla*np.sin(sol.T[0]))*sol.T[2] -
ds*np.sin(sol.T[0]+sol.T[1])*sol.T[3]

```

```

close_vals = np.isclose(np.abs(dxpdt), np.abs(dypdt), atol=0.1)

cv1 = np.array(close_vals)

cv2 = np.nonzero(cv1)[0]

vx = dxpdt[cv2]
vy = dypdt[cv2]

velocities = np.sqrt(vx**2+vy**2)

xp = xp_vals[cv2]
yp = yp_vals[cv2]

plt.scatter(xp, yp)
plt.xlabel('Points at launch angle over x')
plt.ylabel('Points at launch angle over y')
plt.title('Points at launch angle at lever ratio 1:2')
plt.grid(True)
plt.show()

print('velocities: ', velocities)
print('x position: ', xp)

```