Рычаг силы

Никита Б., Антон Б., Юрий Е., Егор С..

Аннотация

В работе проведено исследование изменений параметров системы рычага силы в результате приложения заданной силы. Приводятся результаты совершения работ рычага, в которых участвуют гипотетические и реально существующие объекты. Построена зависимость координаты от времени совершения работы на плечо с реально существующим объектом

Введение

Поднятие предметов является важным вопросом древней физики.

В рамках настоящей работы рассматривается результаты совершения работы рычагом для поднятия нашей планеты гипотетическим телом. Для этого используется среда для python 3.9. Таким образом, целью работы является написание кода для решения поставленной задачи.

Постановка задачи

Для описания этого события необходимо составить конечную формулу из следующих:



Исходя из модельных условий, изменение искомого параметра описываются уравнением

L1= (F2·L2)/F1

Начальные условия

Для решения поставленной задачи необходимо определить следующие начальные условия: F2, L2, F1. Рассмотрим такие значения параметров, при которых Земля становится в равновесии.

Результаты моделирования

В результате численного моделирования были получены следующие результаты:



Приведённые графики показывают, что плечи и приложенные силы образуют прямопропорциональную зависимость. Как видно из графика, решение приводит к равновесию, в то время как для других начальных условий, оно не выполняется.

Заключение

Проведённое исследование показало, что заданное условие равновесия достигается. В то же время, его можно нарушить подстановкой других начальных условий. Таким образом, решение задачи указывает на вариативность результатов. Дальнейшим развитием этой работы может стать созадние более качественной анимации и доработка программы исполнителя задачи.

Код

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import ArtistAnimation

import constmod as cm

lendth = 100

def arm(loadmass = 100, unit = [1, 10], high = 1):

"""

"""

if unit[0] == "mass":

F0 = loadmass \* cm.g

F = unit[1] \* cm.g

ratio = F0/F

S = high \* ratio

A = S\*F

t = ((2\*S/cm.g)\*\*0.5)

print("Отношение плеч =", round(ratio, 2), "Работа =", round(A, 2), "Путь =", round(S, 2), "Время =", round(t, 2))

return ratio, A, S, t

else:

F0 = loadmass \* cm.g

ratio = F0/unit[1]

S = high \* ratio

A = S\*unit[1]

print("Отношение плеч =", ratio, "Работа =", A, "Путь =", S)

return ratio, A, S

arm()

fig = plt.figure()

def point\_arm(first\_mass=100, second\_mass=10, lendth=100):

first\_lendth = round((second\_mass \* lendth) / (first\_mass + second\_mass), 2)

second\_lendth = round(lendth - first\_lendth, 2)

return [first\_lendth, second\_lendth]

def circle\_draw\_func(x\_centre=0, y\_centre=0, Radius=100, N=360):

"""

Пояснение будет писать егор...

"""

x = np.zeros(N)

y = np.zeros(N)

alpha = np.linspace(0, 2\*np.pi, N)

for i in range(0, N, 1):

x[i] = x\_centre + Radius \* np.cos(alpha[i])

y[i] = y\_centre + Radius \* np.sin(alpha[i])

return x, y

def triangle\_draw\_func(x=0, y=0):

x = [x, x-5, x+5, x]

y = [y, y-5, y-5, y]

return x, y

x\_move = np.linspace(point\_arm()[1], 0, 100)

anim\_list = []

point\_xn = point\_arm()[0]

point\_xk = point\_arm()[1]

plt.plot([-point\_xn, point\_xk], [0, 0], 'b')

plt.plot( circle\_draw\_func(-point\_xn, 10, 10)[0], circle\_draw\_func(-point\_xn, 10, 10)[1])

plt.plot( circle\_draw\_func(point\_xk, 1, 1)[0], circle\_draw\_func(point\_xk, 1, 1)[1] )

for i in range(0, 100, 1):

x, y = triangle\_draw\_func(x\_move[i], 0)

obj, = plt.plot(x, y, 'g')

anim\_list.append([obj])

plt.axis('equal')

ani = ArtistAnimation(fig, anim\_list, interval=50)

ani.save('ani.gif')