import numpy as np

import pandas as pd

epsilon = 0.001

lbd1, lbd2 = -1001, -1

y10, y20 = 1, 0

t0, t1, T = 0.0, 5 / max(abs(lbd1), abs(lbd2)), 1.0

n = 25

h\_min = t1 / n

r = t1 / 3

h\_max = 3\* h\_min

y1=y10

y2=y20

t\_values = [t0]

x\_values = [y1]

y\_values = [y2]

# Тестова задача.

def f(y1\_f, y2\_f, t\_f):

    dx = -501 \* (y1\_f + t\_f) + 500 \* (y2\_f + t\_f)

    dy = 500 \* (y1\_f + t\_f) - 501 \* (y2\_f + t\_f)

    return dx, dy

# Точний розв'язок.

def y1\_y2\_correct(t\_values):

    x,y=[],[]

    for t in t\_values:

        x.append(0.5 \* (y10 - y20) \* np.exp(-1001 \* t) + 0.5 \* (y10 + y20) \* np.exp(t))

        y.append(-0.5 \* (y10 - y20) \* np.exp(-1001 \* t) + 0.5 \* (y10 + y20) \* np.exp(t))

    return (x,y)

t=t0+h\_min

while t<t1:

    t\_values.append(t)

    t+=h\_min

t+=h\_max

while t<=T:

    t\_values.append(t)

    t+=h\_max

def rk3\_x(t, x, y, n,h):

    def k1(n):

        return f(x[n], y[n], t[n])[0]

    def k2(n):

        return f(x[n], y[n] + (h / 3) \* k1(n), t[n] + h / 3)[0]

    def k3(n):

        return f(x[n], y[n] + (h \* 2 / 3) \* k2(n), t[n] + h \* 2 / 3)[0]

    return x[n - 1] + (h / 4) \* (k1(n - 1) + 3 \* k3(n - 1))

def rk3\_y(t, x, y, n,h):

    def k1(n):

        return f(x[n], y[n], t[n])[1]

    def k2(n):

        return f(x[n] + (h / 3) \* k1(n), y[n], t[n] + h / 3)[1]

    def k3(n):

        return f(x[n] + (h \* 2 / 3) \* k2(n), y[n], t[n] + h \* 2 / 3)[1]

    return y[n - 1] + (h / 4) \* (k1(n - 1) + 3 \* k3(n - 1))

x\_values.append(rk3\_x(t\_values, x\_values, y\_values, 1,h\_min))

y\_values.append(rk3\_y(t\_values, x\_values, y\_values, 1,h\_min))

def na3\_x(t, x, y, n,h):

    x\_n0 = x[n - 1] + f(x[n - 1], y[n - 1], t[n - 1])[0]

    y\_n0 = y[n - 1] + f(x[n - 1], y[n - 1], t[n - 1])[1]

    while True:

        x\_n1 = x[n - 1] + (h / 12) \* (

                5 \* f(x\_n0, y\_n0, t[n])[0] + 8 \* f(x[n - 1], y[n - 1], t[n - 1])[0] - f(x[n - 2], y[n - 2], t[n - 2])[0])

        y\_n1 = y[n - 1] + (h / 12) \* (

                5 \* f(x\_n0, y\_n0, t[n])[1] + 8 \* f(x[n - 1], y[n - 1], t[n - 1])[1] - f(x[n - 2], y[n - 2], t[n - 2])[1])

        if (abs(x\_n1 - x\_n0) + abs(y\_n1 - y\_n0)) < epsilon:

            return [x\_n1, y\_n1]

        else:

            x\_n0 = x\_n1

            y\_n0 = y\_n1

i=2

while t\_values[i]<=t1:

    xy\_i = na3\_x(t\_values, x\_values, y\_values, i,h\_min)

    x\_values.append(xy\_i[0])

    y\_values.append(xy\_i[1])

    i+=1

while i<len(t\_values):

    xy\_i = na3\_x(t\_values, x\_values, y\_values, i,h\_max)

    x\_values.append(xy\_i[0])

    y\_values.append(xy\_i[1])

    i+=1

xt, yt = y1\_y2\_correct(t\_values)

delta\_x=[]

delta\_y=[]

i=0

while i<len(t\_values):

    delta\_x.append(abs(xt[i] - x\_values[i]))

    delta\_y.append(abs(yt[i] - y\_values[i]))

    i+=1

table = pd.DataFrame({'T': t\_values, 'X': x\_values, 'X\_tochne':xt, 'delta\_X':delta\_x, 'Y': y\_values, 'Y\_tochne':yt, 'delta\_Y':delta\_y})

table.to\_csv('output.csv', index=False)

print(t1)

