Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

УЧЕБНЫЙ ЦЕНТР ОБЩЕЙ ФИЗИКИ ФТФ

Группа М32091 **К работе допущен** ____

Студент Хапчаев Тимур Русланович **Работа выполнена** ____

Преподаватель Шоев В. И. **Отчет принят** ____

Численное моделирование по физике № 1

«Поведение фигур Лиссажу при небольшом несовпадении частот (визуализация)»

1. Цель работы.
Визуализация поведения фигур Лиссажу при небольшой расстройке частот

2. Задачи, **решаемые при выполнении работы**.
Визуализация траекторий, прочерчиваемой точкой, совершающей одновременно два гармонических колебания в двух взаимно перпендикулярных направлениях

3. Объект исследования.
Фигуры Лиссажу

4. Метод экспериментального исследования.
Теоретическое исследование

5. Рабочие формулы и исходные данные.

$$\begin{cases} x(t) = A\cos(\omega t + \varphi_0) \\ y(t) = B\cos(\omega t) \end{cases}$$

6. Выводы и анализ результатов работы:

В результате выполнения лабораторной работы удалось реализовать визуализацию поведения фигур Лиссажу при небольшом несовпадении частот – фигуры вращаются с течением времени.

$$\begin{cases} x(t) = A\cos(\omega t + \varphi_0) \\ y(t) = B\cos((\omega + d\omega)t) \end{cases}$$

Рассмотрим $d\omega \to 0$: при значении, близком или равном 0 и известном отношении частот, фигура Лиссажу получается статическая. Однако, при значениях, отличных от 0 с течением времени аргумент $(\omega + d\omega)t$ возрастает, и фигура Лиссажу начинает вращаться (фаза линейно растёт).

$$x = A\cos(\omega t)$$

$$y = B\cos\big((\omega + d\omega)t\big) = (*)$$

$$\frac{x}{A} = \cos\big((\omega + d\omega)t\big)$$

$$(*) = B\,(\cos(\omega t)\cos(d\omega t) - \sin(\omega t)\sin(\omega t)) = B(\cos(\omega t) - d\omega t\sin(\omega t)$$

$$y(t) = B(\cos(\omega t) - d\omega t\sin(\omega t) = B\left(\frac{x}{A} - d\omega t\sqrt{1 - \left(\frac{x}{A}\right)^2}\right) = \frac{x}{A} - d\omega t\sqrt{1 - \left(\frac{x}{A}\right)^2}$$

$$\left(\frac{y}{B} - \frac{x}{A}\right)^2 = \left(d\omega t\sqrt{1 - \left(\frac{x}{A}\right)^2}\right)^2$$

Тогда получаем:

$$\frac{y^2}{B^2} - \frac{2xy}{AB} + \frac{x^2}{A^2} = d\omega^2 t^2\left(1 - \left(\frac{x}{A}\right)^2\right).$$

Код:

```
import shutil
import matplotlib.pyplot as plt
import numpy as np
from numpy import sin, pi
from os import mkdir, path
from PIL import Image

first_source_frequency = int(input("First source frequency (integer): "))
second_source_frequency = int(input("Second source frequency (integer): "))
curve_colour = input("Colour of the Lissajous curve (r/g/b/c/m/y): ")
number_of_iterations = int(input("Number of iterations (frames in .gif) (integer): "))
phase_step = int(input("Phase step (integer): "))
result_directory_name = "Lissajous curves"


def create_result_directory():
    if path.exists(result_directory_name):
        shutil.rmtree(result_directory_name)
        mkdir(result_directory_name)
```

```python
    else:
        mkdir(result_directory_name)


def draw_lissajous_curves():

    create_result_directory()

    for iteration in range(number_of_iterations):
        phi = iteration / phase_step
        phase_shift = phi * pi

        x = [sin(first_source_frequency * i + phase_shift) for i in np.arange(0., 2 * pi, 0.0001)]
        y = [sin(second_source_frequency * j) for j in np.arange(0., 2 * pi, 0.0001)]

        plt.plot(y, x, curve_colour)
        plt.title(f"Phase shift: {phi}π.")

        plt.grid(True)

        plt.savefig(path.join(result_directory_name, f"Lissajous curve {iteration}.png"))
        plt.close()


def convert_frames_to_gif():
    frames = []

    for frame_number in range(number_of_iterations):
        frame = Image.open(path.join(result_directory_name, f"Lissajous curve {frame_number}.png"))
        frames.append(frame)

    frames[0].save(
        path.join(result_directory_name, "Lissajous curves.gif"),
        save_all=True,
        append_images=frames[1:],
        optimize=True,
        duration=100,
        loop=0
    )


draw_lissajous_curves()
convert_frames_to_gif()
```

Принцип выбранного численного алгоритма:

За построение фигур Лиссажу отвечает функция *draw_lissajous_curves().* По данным уравнениям с помощью утилит matplotlib и numpy строятся кривые с точностью 0.0001.

```python
x = [sin(first_source_frequency * i + phase_shift) for i in np.arange(0., 2 * pi, 0.0001)]
y = [sin(second_source_frequency * j) for j in np.arange(0., 2 * pi, 0.0001)]
```

Результат работы программы: