

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

Московский авиационный институт
(национальный исследовательский университет)
(МАИ)

Институт № 8 «Компьютерные науки и прикладная математика»

ЛАБОРАТОРНАЯ РАБОТА

ПО ОСНОВАМ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ

Выполнила: студентка группы М8О-307Б-23

Жданович Елизавета Таймуразовна

Проверил: _____

Подпись: _____

Москва 2024

Введение

В данной лабораторной работе мы исследуем движение точки по кривой, а именно по спирали Архимеда, и изучим векторы, связанные с этим движением: касательный, нормальный и бинормальный векторы. Спираль Архимеда — это кривая, описываемая точкой, движущейся с постоянной скоростью вдоль луча, который вращается с постоянной угловой скоростью вокруг своего начала. Эта кривая имеет широкое применение в различных областях науки и техники, включая механику, электронику и компьютерную графику.

Целью работы является разработка и исследование закона движения, построение траектории движения объекта и визуализация процесса с помощью анимации.

Закон движения

Спираль Архимеда - это кривая, описываемая точкой, движущейся с постоянной скоростью вдоль луча, который вращается с постоянной угловой скоростью вокруг своего начала.

Параметрическое уравнение спирали Архимеда:

$$\begin{cases} x(t) = a \cdot t \cdot \cos(t) \\ y(t) = a \cdot t \cdot \sin(t) \\ z(t) = b \cdot t \end{cases}$$

где:

- a - параметр, определяющий плотность витков спирали,
- b - параметр, определяющий шаг спирали вдоль оси z ,
- t - параметр, определяющий время или угол поворота.

Иллюстрация закона движения:



Анимация движения и векторов

Для анимации движения точки по спирали Архимеда и отображения векторов нормальной, касательной и бинормальной, можно использовать библиотеку `matplotlib` в Python.

Код:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.animation import FuncAnimation

# Параметры спирали Архимеда
a = 0.1
b = 0.1
t_max = 10 * np.pi
t_values = np.linspace(0, t_max, 1000)

# Вычисление координат точек спирали
x = a * t_values * np.cos(t_values)
```

```
y = a * t_values * np.sin(t_values)
```

```
z = b * t_values
```

```
# Функция для вычисления векторов
```

```
def compute_vectors(t):
```

```
    # Координаты точки
```

```
    x_t = a * t * np.cos(t)
```

```
    y_t = a * t * np.sin(t)
```

```
    z_t = b * t
```

```
# Производные для вычисления касательного вектора
```

```
dx_dt = a * (np.cos(t) - t * np.sin(t))
```

```
dy_dt = a * (np.sin(t) + t * np.cos(t))
```

```
dz_dt = b
```

```
# Касательный вектор
```

```
T = np.array([dx_dt, dy_dt, dz_dt])
```

```
T = T / np.linalg.norm(T)
```

```
# Вычисление нормального вектора
```

```
d2x_dt2 = a * (-2 * np.sin(t) - t * np.cos(t))
```

```
d2y_dt2 = a * (2 * np.cos(t) - t * np.sin(t))
```

```
d2z_dt2 = 0
```

```
N = np.array([d2x_dt2, d2y_dt2, d2z_dt2])
```

```
N = N / np.linalg.norm(N)
```

```
# Бинормальный вектор
```

```
B = np.cross(T, N)
```

```
return x_t, y_t, z_t, T, N, B
```

```
# Создание 3D графика
```

```

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Начальные значения для анимации
t_current = 0
x_t, y_t, z_t, T, N, B = compute_vectors(t_current)

# Отображение спирали
ax.plot(x, y, z, color='gray', alpha=0.5)

# Точка на спирали
point, = ax.plot([x_t], [y_t], [z_t], 'ro')

# Векторы
T_line, = ax.plot([x_t, x_t + T[0]], [y_t, y_t + T[1]], [z_t, z_t + T[2]], 'r-', label='Касательный')
N_line, = ax.plot([x_t, x_t + N[0]], [y_t, y_t + N[1]], [z_t, z_t + N[2]], 'g-', label='Нормальный')
B_line, = ax.plot([x_t, x_t + B[0]], [y_t, y_t + B[1]], [z_t, z_t + B[2]], 'b-', label='Бинормальный')

# Функция для обновления анимации
def update(frame):
    global t_current
    t_current += 0.1
    if t_current > t_max:
        t_current = 0

    x_t, y_t, z_t, T, N, B = compute_vectors(t_current)

    point.set_data([x_t], [y_t])
    point.set_3d_properties([z_t])

    T_line.set_data([x_t, x_t + T[0]], [y_t, y_t + T[1]])
    T_line.set_3d_properties([z_t, z_t + T[2]])

```

```

N_line.set_data([x_t, x_t + N[0]], [y_t, y_t + N[1]])
N_line.set_3d_properties([z_t, z_t + N[2]])

B_line.set_data([x_t, x_t + B[0]], [y_t, y_t + B[1]])
B_line.set_3d_properties([z_t, z_t + B[2]])

return point, T_line, N_line, B_line

# Анимация
ani = FuncAnimation(fig, update, frames=np.arange(0, 100), interval=50, blit=True)

# Отображение легенды
ax.legend()

# Отображение графика
plt.show()

```

Описание кода:

1. **Параметры спирали:** Задаются параметры a и b , определяющие форму спирали, а также максимальное значение параметра t .
2. **Координаты:** Используются параметрические уравнения для вычисления координат точек спирали.
3. **Векторы:** Функция `compute_vectors` вычисляет координаты точки и векторы касательной, нормальной и бинормальной в зависимости от текущего значения t .
4. **Создание графика:** Создается 3D график с использованием `matplotlib`.
5. **Анимация:** Используется `FuncAnimation` для обновления положения точки и векторов на каждом кадре анимации.

Результат:

При запуске кода будет отображена анимация движения точки по спирали Архимеда, а также векторы касательной, нормальной и бинормальной в каждый момент времени.

