РГР №3

Былькова Кристина Алексеевна

Группа М8О-308Б-22

Задание:

Вычислите интеграл Лебега–Стилтьеса $\int\limits_{[a,b]} f(x)\,dF(x)$.

Вариант 2:

2) $[a,b] = [-k, 5l]$; $f(x) = 2\cos kx + 2\chi\left(3x - \frac{l}{5}\right) - x^2$, $F(x) = e^x + 2\chi(x+1) + 2\chi(4x-k) + x^3$;

Решение:

Итак, при $k = 8,\ l = 15$ дано:

$$[a; b] = [-8; 75]$$

$$f(x) = 2\cos(8x) + 2\chi(3\mathrm{x} - 3) - \mathrm{x}^2$$

$$F(x) = e^x + 2\chi(\mathrm{x} + 1) + 2\chi(4\mathrm{x} - 8) + \mathrm{x}^3$$

Для решения данной задачи необходимо разложить $F(x)$ на непрерывную и разрывную (скачковую) части. Затем найти производную гладкой части $F'(x)$. Потом находим точки скачков и их величины:

$$\Delta F(-1) = 2,\ \ \Delta F(2) = 2$$

Считаем вклад скачков:

$$\sum f(x_k) * \Delta F(x_k) = f(-1) * 2 + f(2) * 2$$

Далее вычисляем интеграл:

$$\int\limits_a^b f(x)F'(x)\,dx$$

В итоге складываем:

$$\int_a^b f(x)dF(x) = \int_a^b f(x)F'(x)dx + \sum f(x_k) * \Delta F(x_k)$$

Код:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
from math import exp, cos, pi

def chi(x):
    return 1 if x >= 0 else 0

def f(x):
    return 2 * cos(8 * x) + 2 * chi(3 * x - 3) - x**2

def dF(x):
    return exp(x) + 3 * x**2

def main():
    a, b = -8, 75
    integral_continuous, _ = quad(lambda x: f(x) * dF(x), a, b)

    jumps = []
    for xk in [-1, 2]:
        if a < xk < b:
            delta_F = 0
            delta_F += 2 * (chi(xk + 1) - chi(xk + 1 - 1e-9))  # для 2χ(x+1)
            delta_F += 2 * (chi(4 * xk - 8) - chi(4 * xk - 8 - 1e-9))  # для
2χ(4x-8)
            jumps.append(f(xk) * delta_F)

    integral_jumps = sum(jumps)
    total_integral = integral_continuous + integral_jumps

    print(f"Интеграл Лебега-Стильтьеса: {total_integral:.6f}")

# Векторизованные функции
def chi_vec(x):
    return np.where(x >= 0, 1, 0)

def F_vec(x):
        return np.exp(x) + x**3 + 2 * chi_vec(x + 1) + 2 * chi_vec(4 * x - 8)

def f_vec(x):
    return 2 * np.cos(8 * x) + 2 * chi_vec(3 * x - 3) - x**2


def plot_chi_functions():
    x = np.linspace(-6, 6, 1000)
```

```python
    y_chi1 = 2 * chi_vec(x + 1)
    y_chi2 = 2 * chi_vec(4 * x - 8)

    _, axs = plt.subplots(1, 2, figsize=(12, 5), sharey=True)

    for ax in axs:
        ax.spines['left'].set_position('zero')
        ax.spines['bottom'].set_position('zero')
        ax.spines['right'].set_color('none')
        ax.spines['top'].set_color('none')
        ax.yaxis.set_ticks_position('left')
        ax.xaxis.set_ticks_position('bottom')
        ax.set_xticks(np.arange(-6, 6, 1))

    # Левый график — 2χ(x+1)
    axs[0].plot(x, y_chi1, label='2χ(x+1)', color='pink')
    axs[0].axvline(x=-1, color='black', linestyle='--', linewidth=1.5)
    axs[0].set_title("2χ(x+1)")
    axs[0].set_xlabel("x")
    axs[0].grid(True)
    axs[0].legend()

    # Правый график — 2χ(4x-8)
    axs[1].plot(x, y_chi2, label='2χ(4x-8)', color='orange')
    axs[1].axvline(x=2, color='black', linestyle='--', linewidth=1.5)
    axs[1].set_title("2χ(4x-8)")
    axs[1].set_xlabel("x")
    axs[1].grid(True)
    axs[1].legend()

    plt.suptitle("Графики функций 2χ(x+1) и 2χ(4x-8)")
    plt.tight_layout(rect=[0, 0, 1, 1])
    plt.show()

def plot_f_and_F_functions():
    x_vals = np.linspace(-8, 75, 10000)
    y_f = f_vec(x_vals)
    y_F = F_vec(x_vals)

    _, axs = plt.subplots(1, 2, figsize=(12, 5))

    for ax in axs:
        ax.spines['left'].set_position('zero')
        ax.spines['bottom'].set_position('zero')
        ax.spines['right'].set_color('none')
        ax.spines['top'].set_color('none')
        ax.yaxis.set_ticks_position('left')
```

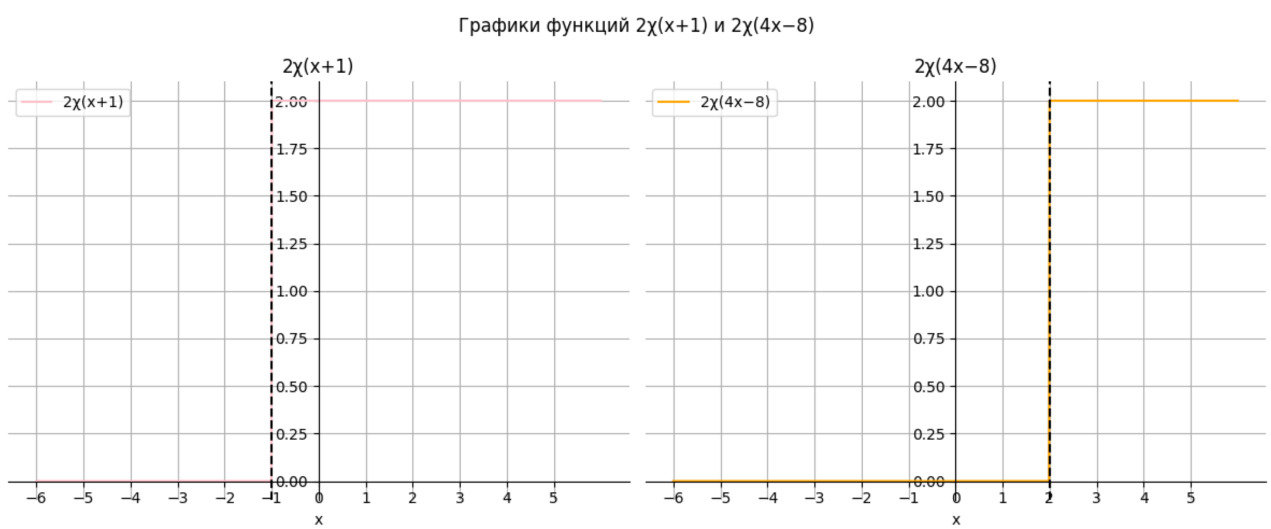
```
        ax.xaxis.set_ticks_position('bottom')
        ax.grid(True)

    # Левый график — f(x)
    axs[0].plot(x_vals, y_f, label=r'$f(x) = 2\cos(8x) + 2\chi(3x-3) - x^2$',
color='blue')
    axs[0].axvline(x=1, color='red', linestyle='--', linewidth=1,
label=r'2$\chi(3x-3)$')
    axs[0].set_title(r'$f(x)$')
    axs[0].set_xlabel('x')
    axs[0].legend(fontsize=9)

    # Правый график — F(x)
    axs[1].plot(x_vals, y_F, label=r'$F(x) = e^x + 2\chi(x+1) + 2\chi(4x-8) +
x^3$', color='green')
    axs[1].axvline(x=-1, color='purple', linestyle='--', linewidth=1,
label=r'скачок 2$\chi(x+1)$')
    axs[1].axvline(x=2, color='orange', linestyle='--', linewidth=1, label=r'скачок
2$\chi(4x-8)$')
    axs[1].set_title(r'$F(x)$')
    axs[1].set_xlabel('x')
    axs[1].legend(fontsize=9)

    plt.suptitle("Графики функций $f(x)$ и $F(x)$", fontsize=14)
    plt.tight_layout(rect=[0, 0, 1, 1])
    plt.show()

def plot_F_log():
    x_vals = np.linspace(-8, 75, 10000)
    y_vals = F_vec(x_vals)

    _, ax = plt.subplots(figsize=(10, 6))

    ax.plot(x_vals, y_vals, label=r'$F(x) = e^x + x^3 + 2\chi(x+1) + 2\chi(4x-8)$',
color='green')

    ax.spines['left'].set_position('zero')
    ax.spines['bottom'].set_position('zero')
    ax.spines['right'].set_color('none')
    ax.spines['top'].set_color('none')
    ax.yaxis.set_ticks_position('left')
    ax.xaxis.set_ticks_position('bottom')

    # Логарифмический масштаб (лучше видно скачки)
    ax.set_yscale('log')

    ax.grid(True)
```

```
    ax.set_title(r'График функции $F(x)$ (логарифмический масштаб)', fontsize=14)
    ax.set_xlabel('x')
    ax.set_ylabel(r'$F(x)$', fontsize=12)
    ax.legend()

    ax.axvline(x=-1, color='purple', linestyle='--', linewidth=1, label=r'скачок
$\chi(x+1)$')
    ax.axvline(x=2, color='orange', linestyle='--', linewidth=1, label=r'скачок
$\chi(4x-8)$')

    handles, labels = ax.get_legend_handles_labels()
    by_label = dict(zip(labels, handles))
    ax.legend(by_label.values(), by_label.keys())

    plt.tight_layout()
    plt.show()

if __name__ == "__main__":
    main()
    plot_chi_functions()
    plot_f_and_F_functions()
    plot_F_log()
```

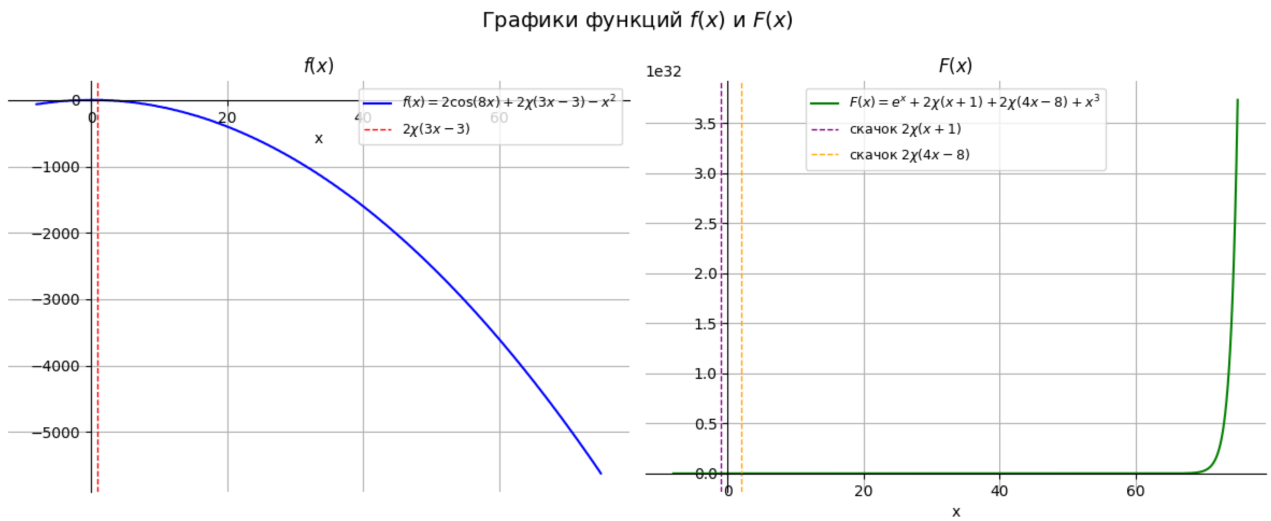
Вывод:

$ python main.py

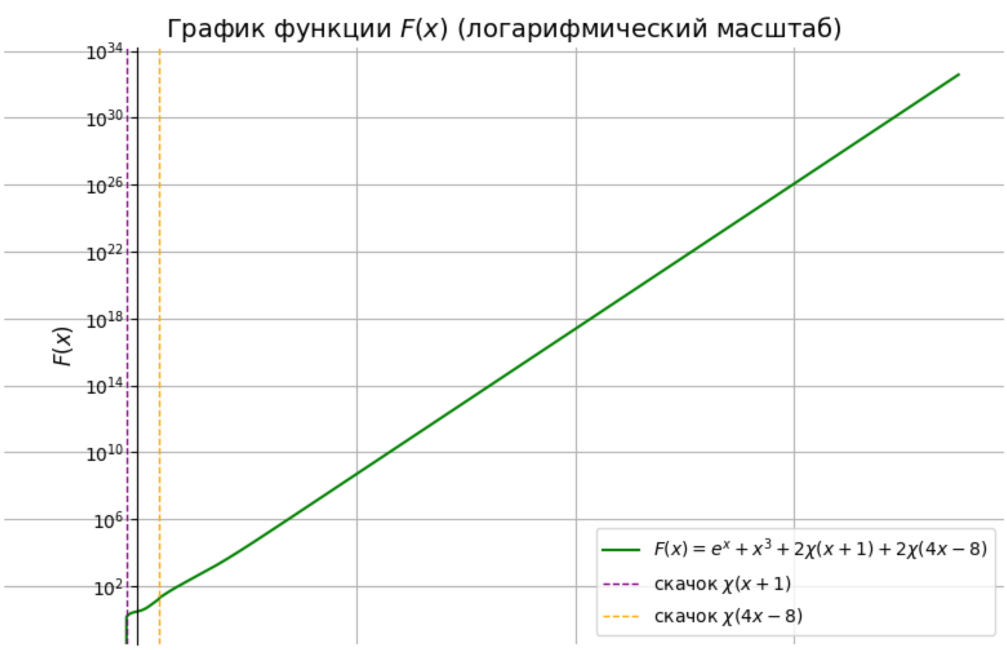
Интеграл Лебега-Стильтьеса: -2043957408774194279254538299421753344.000000



Изображение выше показывает графики функций $2\chi(x+1)$ и $2\chi(4x-8)$, в которых происходят скачки: в точках $x=-1$ и $x=2$ соответственно.



А здесь представлены графики функций $f(x)$ — левый подграфик и $F(x)$ — правый подграфик.



Также я построила график $F(x)$ в логарифмическом масштабе: в таком виде нагляднее виден участок между скачками.

Дело в том, что matplotlib по умолчанию устанавливает экспоненциальную нотацию, если числа слишком большие или маленькие (помогает избежать переполнение подписей на осях), но визуально данный вариант не всегда выглядит хорошо.