

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерных технологий

## Теория функций комплексной переменной

### Лабораторная работа №2

Выполнила:

Павличенко Софья Алексеевна, Р3215

Преподаватель:

Попов Арсений Михайлович

Санкт-Петербург 2024г.

## Оглавление

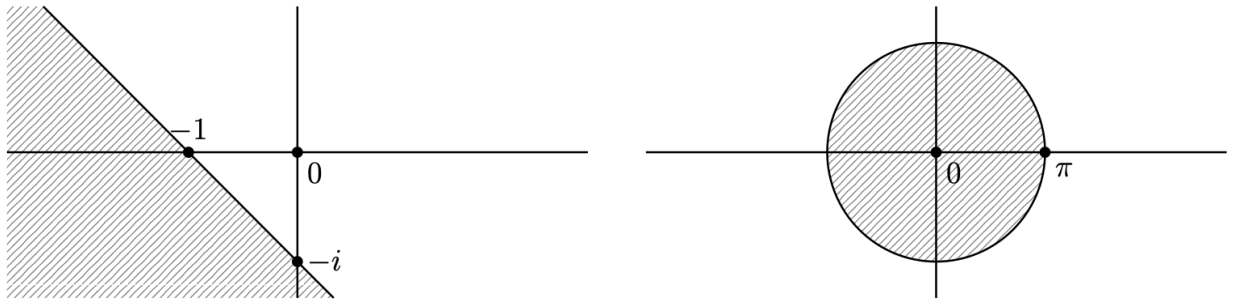
Задание.....	3
Решение .....	4
Задание 1 .....	4
Задание 2.....	4
Задание 3.....	5
Задание 4.....	6
Исходный код программы.....	6
Полученные визуализации .....	8

## Задание

1. Аналитически опишите заданные множества.
2. Воспользовавшись композицией классических преобразований, составьте конформное отображение, которое переводит первую область во вторую. Табличка с преобразованиями может быть найдена в конце данного документа.
3. Составьте обратное отображение, переводящее второе множество в первое.
4. На любом удобном вам языке программирования напишите программу, которая нарисует первое множество и все этапы его преобразования во второе. Достаточно наглядным будет взять набор точек множества, передающий его форму (учтите, что может понадобится сделать набор «более плотным» в какой-то части множества).

## Решение

### Задание 1



#### Первое множество:

Область на комплексной плоскости, ограниченная прямой  $Re(z) + Im(z) = -1$ .

$$M_1 = \{z \in \mathbb{C} : Re(z) + Im(z) < -1\}$$

#### Второе множество:

Круг радиуса  $\pi$  с центром в точке 0.

$$M_2 = \{z \in \mathbb{C} : |z| < \pi\}$$

### Задание 2

#### 1. Сдвиг области $M_1$ .

Чтобы сделать прямую  $Re(z) + Im(z) = -1$  проходящей через начало координат, сдвинем область с помощью преобразования:

$$w_1 = z + 1$$

Теперь область  $M_1$  примет вид:

$$M'_1 = \{w_1 \in \mathbb{C} : Re(w_1) + Im(w_1) < 0\}.$$

#### 2. Поворот области.

Чтобы сделать границу области вертикальной (вдоль оси  $Re(w_1)$ ), повернем плоскость на  $45^\circ$  против часовой стрелки:

$$w_2 = w_1 e^{-i\pi/4}$$

После этого область  $M'_1$  преобразуется в:

$$M_1'' = \{w_2 \in \mathbb{C} : \operatorname{Re}(w_2) < 0\}$$

### 3. Преобразование в единичный круг.

Для преобразования полуплоскости  $\operatorname{Re}(w_2) < 0$  в единичный круг используется отображение:

$$w_3 = \frac{1 + w_2}{1 - w_2}$$

Теперь область  $M_1''$  переходит в:

$$M_1''' = \{w_3 \in \mathbb{C} : |w_3| < 1\}$$

### 4. Растяжение круга.

Чтобы масштабировать единичный круг  $|w_3| \leq 1$  до радиуса  $\pi$ , используем преобразование:

$$w_4 = \pi w_3$$

## Итоговое конформное отображение:

Композиция всех преобразований:

1.  $w_1 = z + 1,$
2.  $w_2 = w_1 e^{-i\pi/4} = (z + 1)e^{-i\pi/4},$
3.  $w_3 = \frac{1+w_2}{1-w_2} = \frac{1+(z+1)e^{-i\pi/4}}{1-(z+1)e^{-i\pi/4}},$
4.  $w_4 = \pi w_3 = \pi \cdot \frac{1+(z+1)e^{-i\pi/4}}{1-(z+1)e^{-i\pi/4}}.$

Итоговая формула:

$$w = \pi \cdot \frac{1 + (z + 1)e^{-i\pi/4}}{1 - (z + 1)e^{-i\pi/4}}$$

## Задание 3

### 1. Обратное растяжение круга.

$$w_3 = \frac{w_4}{\pi}$$

### 2. Обратное отображение единичного круга в полуплоскость.

$$w_2 = \frac{w_3 - 1}{w_3 + 1}$$

### 3. Обратное вращение плоскости.

$$w_1 = w_2 e^{-i\pi/4}$$

### 4. Обратный сдвиг.

$$z = w_1 - 1$$

## Итоговое обратное отображение:

Композиция всех обратных преобразований:

1.  $w_3 = \frac{w_4}{\pi},$
2.  $w_2 = \frac{w_3 - 1}{w_3 + 1} = \frac{\frac{w_4}{\pi} - 1}{\frac{w_4}{\pi} + 1},$
3.  $w_1 = w_2 e^{-i\pi/4} = \frac{\frac{w_4}{\pi} - 1}{\frac{w_4}{\pi} + 1} e^{-i\pi/4},$
4.  $z = w_1 - 1 = \frac{\frac{w_4}{\pi} - 1}{\frac{w_4}{\pi} + 1} e^{-i\pi/4} - 1.$

Итоговая формула:

$$z = \left( \frac{\frac{w_4}{\pi} - 1}{\frac{w_4}{\pi} + 1} \right) e^{-i\pi/4} - 1.$$

## Задание 4

Исходный код программы

```
import numpy as np
import matplotlib.pyplot as plt

def generate_M1_points(density, extent):
    """Генерация точек для области M1: Re(z) + Im(z) < -1."""
    x = np.linspace(-extent, extent, density)
    y = np.linspace(-extent, extent, density)
    X, Y = np.meshgrid(x, y)
    mask = X + Y < -1
    return X[mask], Y[mask]

def shift_z(x, y):
    """Сдвиг z -> z + 1."""
    return x + 1, y
```

```

def rotate_z(x, y, angle=-np.pi / 4):
    """Вращение на заданный угол."""
    z = x + 1j * y
    rotated_z = z * np.exp(1j * angle)
    return rotated_z.real, rotated_z.imag

def map_to_unit_circle(x, y):
    """Преобразование в единичный круг."""
    z = x + 1j * y
    mapped_z = (1 + z) / (1 - z)
    return mapped_z.real, mapped_z.imag

def scale_circle(x, y, scale=np.pi):
    """Масштабирование до радиуса pi."""
    return x * scale, y * scale

def plot_area(x, y, title, ax):
    """Визуализация области."""
    ax.fill(x, y, color='skyblue', alpha=0.5)
    ax.axhline(0, color='black', linewidth=0.5)
    ax.axvline(0, color='black', linewidth=0.5)
    ax.set_aspect('equal', adjustable='box')
    ax.set_title(title)
    ax.grid(True)
    ax.set_xlim([-np.pi - 1, np.pi + 1])
    ax.set_ylim([-np.pi - 1, np.pi + 1])

def main():
    density = 2000 # Плотность точек
    extent = 10 # Границы области

    # Этап 1: Множество M1
    x, y = generate_M1_points(density, extent)

    # Этап 2: Сдвиг
    x_shifted, y_shifted = shift_z(x, y)

    # Этап 3: Вращение
    x_rotated, y_rotated = rotate_z(x_shifted, y_shifted)

    # Этап 4: Преобразование в единичный круг
    x_circle, y_circle = map_to_unit_circle(x_rotated, y_rotated)

    # Этап 5: Масштабирование до радиуса pi
    x_final, y_final = scale_circle(x_circle, y_circle)

    fig, axs = plt.subplots(1, 5, figsize=(20, 4))

    plot_area(x, y, "M1:  $\text{Re}(z) + \text{Im}(z) < -1$ ", axs[0])
    plot_area(x_shifted, y_shifted, "Сдвиг:  $z \rightarrow z + 1$ ", axs[1])
    plot_area(x_rotated, y_rotated, "Вращение:  $z \rightarrow z * e^{-i\pi/4}$ ", axs[2])
    plot_area(x_circle, y_circle, "Преобразование:  $w = (1+z)/(1-z)$ ", axs[3])
    plot_area(x_final, y_final, "Круг радиуса  $\pi$ ", axs[4])

    plt.tight_layout()
    plt.show()

```

```
if __name__ == "__main__":  
    main()
```

## Полученные визуализации

