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 2](#_Toc186486019)

[Решение 4](#_Toc186486020)

[Задание 1 4](#_Toc186486021)

[Задание 2 4](#_Toc186486022)

[Задание 3 5](#_Toc186486023)

[Задание 4 6](#_Toc186486024)

[Исходный код программы 6](#_Toc186486025)

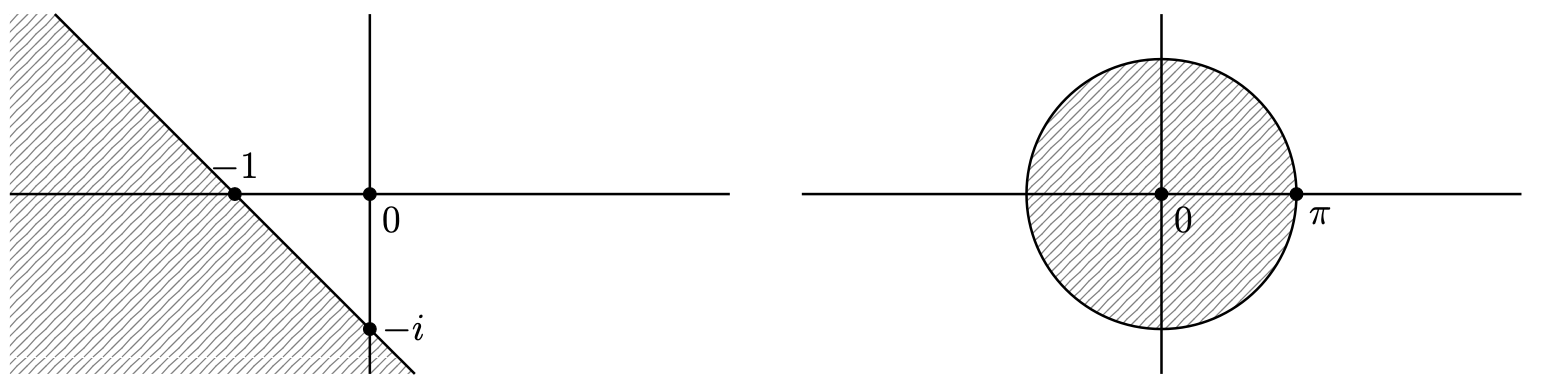
[Полученные визуализации 8](#_Toc186486026)

# Задание

1. Аналитически опишите заданные множества.
2. Воспользовавшись композицией̆ 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, передающий̆ 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 .

**Второе множество:**  
Круг радиуса с центром в точке 0.

## Задание 2

1. **Сдвиг области .**  
   Чтобы сделать прямую проходящей через начало координат, сдвинем область с помощью преобразования:

Теперь область примет вид:

1. **Поворот области.**

Чтобы сделать границу области вертикальной (вдоль оси , повернем плоскость на против часовой стрелки:

После этого область ​ преобразуется в:

1. **Преобразование в единичный круг**.  
   Для преобразования полуплоскости в единичный круг используется отображение:

Теперь область ​ переходит в:

1. **Растяжение круга**.  
   Чтобы масштабировать единичный круг до радиуса , используем преобразование:

**Итоговое конформное отображение:**

Композиция всех преобразований:

1. ,
2. ,
3. ,
4. .

Итоговая формула:

## Задание 3

1. **Обратное растяжение круга.**
2. **Обратное отображение единичного круга в полуплоскость.**
3. **Обратное вращение плоскости.**
4. **Обратный сдвиг.**

**Итоговое обратное отображение:**

Композиция всех обратных преобразований:

1. ​​,
2. ​,
3. ,
4. .

Итоговая формула:

## 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: Re(z) + Im(z) < -1", axs[0])plot\_area(x\_shifted, y\_shifted, "Сдвиг: z -> z + 1", axs[1])plot\_area(x\_rotated, y\_rotated, "Вращение: z -> 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 π", axs[4])  
  
 plt.tight\_layout()  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

### Полученные визуализации

