Лабораторная работа №1

In [22]:
```
# подключить модуль
import math as m
import cmath as cmath
import matplotlib.pyplot as plt
from sympy import *
from numpy import *
from pylab import *
```

Решение задач на Python

Математический анализ, Комплексные числа

Пример 1: сложение, вычитание, умножение, деление, производится с помощью стандартных операций «+», «-», «*», «/»

Пусть:

$x = 1 + 3i$

$y = 2 - i$

$g = 1 - 2i$

$t = 10$

Найти:

$z = x \cdot y$

$h = \frac{t}{g}$

$n = p^2 = p \cdot p$

$C = z + h + n$

In [17]:
```
x=complex(1,3)
y=complex(2,-1)
z=x*y
print('z = ', z)

g=complex(1,-2)
print('g =', g)

t=complex(10,0)
print('t =', t)

h=t/g
print('h =', h)

p=complex(-1,-1)
n=p*p
print('n =', n)
```

```
C=z+h+n
print('C =', C)
```

```
z =  (5+5j)
g = (1-2j)
t = (10+0j)
h = (2+4j)
n = 2j
C = (7+11j)
```

Пример 2: Возведение в степень: pow (число, показатель степени в которую мы возводим число)

Степень i^2.

$x = i$,

$y = x^2$,

$y = -1$

In [20]:
```
x=complex(0,1)
y=pow(x,2) # Степень
print(y)
```

```
(-1+0j)
```

Пример 3: Найти значение функции

$$f(x) = x^4 + \frac{2+i}{x} - (-3+2i), \quad \text{при } x = 1-2i$$

In [27]:
```
x=complex(1,-2)
i=complex(0,1)

f=x**4+(2+i)/x-(-3+2*i)
print(f)
```

```
(-4+23j)
```

Пример 4: Выполнить указанные действия

$$\frac{(1+i)^8}{(1-i)^6}$$

In [31]:
```
(1 + i) ** 8 / (1 + i) ** 6
```

Out[31]:
```
(-0+2j)
```

Пример 5: Решить систему уравнений

$$\begin{cases} (2+i)\,x + (2-i)\,y = 6, \\ (3+2i)\,x + (3-2i)\,y = 8, \end{cases}$$

In [35]:
```
from sympy import Symbol, nsolve
import sympy
import mpmath
```

```
mpmath.mp.dps = 3

x = Symbol('x')
y = Symbol('y')

i=complex(0,1)

fl = (2+i)*x+y*(2-i)-6
f2 = (2-i)*x+(3-2*i)*y-8

print(nsolve( (fl, f2), (x, y), (-1, 1) ))
```

```
Matrix([[-0.0588 - 0.765*I], [1.82 + 1.71*I]])
```

Пример 6: Итак, мы можем с легкостью производить любые действия с комплексными числами в среде Python.

Вычислить

$$(1+3i)\cdot(2-i)+\frac{10}{(1+2i)}+(-1-i)^2=7+11j$$

In [21]:
```
x=complex(1,3)
y=complex(2,-1)

z=x*y
print(z)

g=complex(1,-2)
print(g)

t=complex(10,0)
print(t)

h=t/g
print(h)

p=complex(-1,-1)
n=p*p
print(n)

C=z+h+n
print(C)
```

```
(5+5j)
(1-2j)
(10+0j)
(2+4j)
2j
(7+11j)
```

Пример 7: С понятием комплексного числа связано решение квадратных уравнений, дискриминант которых меньше нуля.

Решить уравнение $x^2 — 2x + 5 = 0$.

Решение.
Чтобы решить уравнение $f(x) = 0$ используем функцию $solve(f(x))$.

In [24]:
```
import math
from sympy import*

x = Symbol("x")
print(solve(x**2-2*x+5))
```

```
[1 - 2*I, 1 + 2*I]
```

Пример 8: Вычислить

$$-(3+5i)^{10} - 25 \cdot \frac{3i-9}{2+8i}$$

In [3]:
```
i = complex(0,1)

-(3 + 5 * i)**10 - 25 * ( 3 * i - 9) / 2 + 8 * i
```

Out[3]:
```
(28984688.5+34989570.5j)
```

Пример 9: Найти модуль и аргумент (фазу) комплексного числа

$$z = 2 + 2 \cdot \sqrt{3} \cdot i$$

In [60]:
```
abs(z)
```

Out[60]:
```
3.9999999999999996
```

In [59]:
```
import cmath
z=complex(2,2*sqrt(3))

cmath.phase(z)
round(math.degrees(cmath.phase(z)))
```

Out[59]:
```
60
```

Примеры решения задач

Пример 1: Пусть Вычислите

$z_1 = -4 - 9i,$
$z2 = 1 - 8i$

Вычислите:

$$\frac{z_1 - \overline{z_2}}{\overline{z_1} + z_2}$$

In [3]:
```
z1 = -4 - 9 * 1j
z2 = 1 - 8 * 1j

print((z1-z2.conjugate())/(z1.conjugate()+z2))
```

```
(-0.19999999999999982+5.6000000000000005j)
```

Пример 2: Приведите число $z = 2 + 2\sqrt{3}i$ к тригонометрическому виду.

In [6]:
```
import math
import cmath

z=2+2*math.sqrt(3)*1j

fi=round(math.degrees(cmath.phase(z)))
print(fi)

r=abs(z)
print(r)
```

```
60
3.9999999999999996
```

Пример 7: Вычислите значение выражения $\frac{3+7i}{4i-5}$ и представьте результат в виде $a + bi$

In [7]:
```
print((3+7j)/(4j-5))
```

```
(0.31707317073170743-1.1463414634146343j)
```

Пример 9:Вычислите значение многочлена

$P(z) = (-4+4i)z^2 + (-1+3i)z + (-2-3i)$ в точке $z = 1 - 3i$

In [9]:
```
z=1+3j
p=(-4+4j)*(z*z)+(-1+3j)*z+(-2-3j)
print(p)
```

```
(-4-59j)
```

Пример 13: Вычислите модуль и аргумент числа

$z = -8 - 8i$

In [12]:
```
import math
import cmath

z=complex(-8,-8)
round(math.degrees(cmath.phase(z))), abs(z)
```

Out[12]: (-135, 11.313708498984761)

Пример 15: Найдите комплексные корни уравнения

$$x^2 + 8x + 20 = 0$$

In [13]:
```
import math
from sympy import *

x = Symbol("x")
print(solve(x**2+8*x+20))
```

```
[-4 - 2*I, -4 + 2*I]
```

Пример 18: Вычислите модуль и аргумент числа

$$z = -6$$

In [14]:
```
import math
import cmath

z=complex(-6,0)
round(math.degrees(cmath.phase(z))), abs(z)
```

Out[14]: (180, 6.0)

Пример 21: Приведите число $z = 6 - 6i$ к тригонометрическому виду.

In [16]:
```
import math
import cmath

z=complex(6,6)
print(round(math.degrees(cmath.phase(z))))

r=abs(z)
print(r)

c=r*(math.cos(-45)+1j*math.sin(-45))
print(c)
```

```
45
8.48528137423857
(4.4575048871930445-7.220155828003307j)
```

Пример 26: Вычислите модуль и аргумент числа

$$z = 3\sqrt{3} - 3i$$

In [17]:
```
import math
import cmath

z=complex(3*sqrt(3),-3)
round(math.degrees(cmath.phase(z))), abs(z)
```

Out[17]: (-30, 6.0)

Пример 32: Вычислите значение выражения $\frac{(6+5i)(-4+2i)}{2+5i}$ и представьте результат в виде $a+bi$

In [18]:
```
((6+5j)*(-4+2j))/(2+5j)
```

Out[18]: (-3.724137931034483+5.310344827586207j)

Пример 35: Найдите комплексные корни уравнения

$$x^2+10x+26=0$$

In [19]:
```
import math
from sympy import *

x=Symbol("x")

print(solve(x**2+10*x+26))
```

```
[-5 - I, -5 + I]
```

Пример 50: Вычислите значение выражения $\frac{(2-4i)(3-4i)}{2+5i}$ и представьте результат в виде $a+bi$

In [20]:
```
((2-4j)*(3-4j))/(2+5j)
```

Out[20]: (-4.137931034482759+0.3448275862068966j)

Задачи для самостоятельного решения

Вычислить модуль и аргумент числа $z=1+\sqrt{3}i$

Решение:

In [15]:
```
i = complex(0, 1)
z = 1 + sqrt(3) * i

print('z = ', abs(z))
print('$arg(z)$ =', round(m.degrees(cm.phase(z))))
```

z = 2.00000000000000
$arg(z)$ = 60

Ответ:

$|z|=2$,
$arg(z)=\frac{\pi}{3}$

Индивидуальное задание

Напряженность электрического поля и эквипотенциали

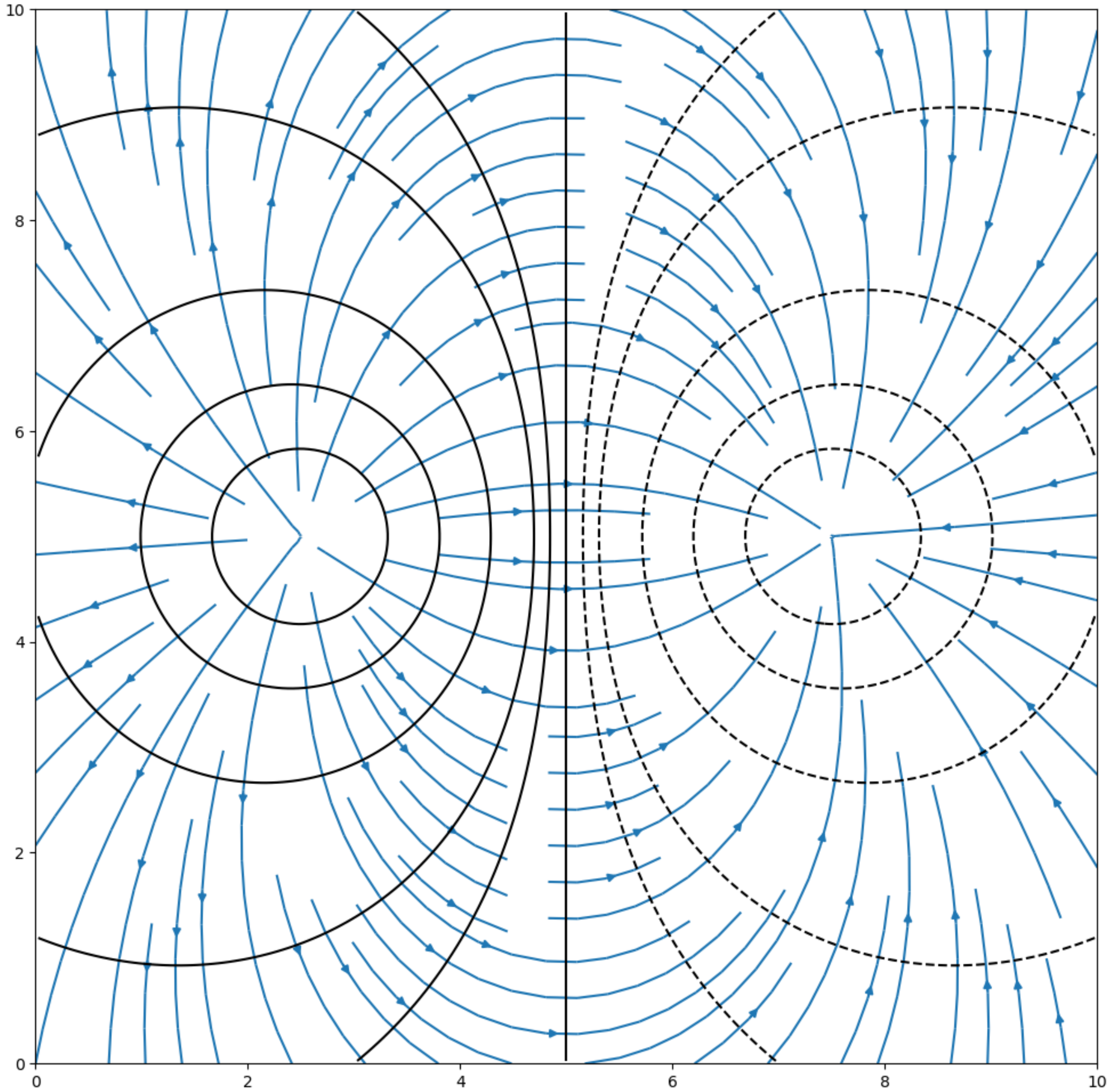
In [42]:
```
# Electric potential visualization
from numpy import linspace, array, sqrt, zeros, logspace, append, meshgrid, empt
from pylab import show, contour, contourf, savefig, quiver, figure, streamplot

Nx = 200
Ny = 200
xmin = 0.
xmax = 10.
ymin = 0.
ymax = 10.
x = linspace(xmin, xmax, Nx)
y = linspace(ymin, ymax, Ny)
X, Y = meshgrid(x,y)
phi = zeros((Ny, Nx), float)
dist = zeros((Ny, Nx), float)
x1 = xmin + (xmax-xmin)*0.25
y1 = ymin + (ymax-ymin)*0.5
x2 = xmin + (xmax-xmin)*0.75
y2=y1

charges = [(x1,y1,1.), (x2,y2,-1.)]
for q in charges:
    dist = sqrt((X-q[0])**2 + (Y-q[1])**2)
    phi += q[2]/dist

dx = (xmax-xmin)/(Nx-1)
dy = (ymax-ymin)/(Ny-1)
Ex = zeros((Ny, Nx), float)
Ey = zeros((Ny, Nx), float)
Ey, Ex = gradient(-phi, dx, dy)

plt.figure(figsize = (12,12))
#n = 24
#quiver(X[::n, ::n], Y[::n,::n], Ex[::n,::n], Ey[::n,::n], scale = 0.1, units="x
plt.streamplot(X, Y, Ex, Ey)
contourLevels = [-1., -0.5, -0.25, -0.1, -0.05, 0., 0.05, 0.1, 0.25, 0.5, 1.]
plt.contour(phi, levels = contourLevels,
        colors='k',
        origin = "lower",
        extent=(xmin,xmax,ymin,ymax)#,
        #linestyle = 'dotted'
        )
plt.show()
#savefig("GetPotential.png")
```



<Figure size 640x480 with 0 Axes>

Создание фрактала Мандельброта

In [23]:
```
from numpy import newaxis, zeros, empty

def compute_mandelbrot(N_max, some_threshold, nx, ny):
# Сетка, значение c
    x = np.linspace(-2, 1, nx)
    y = np.linspace(-1.5, 1.5, ny)
    c = x[:,newaxis] + 1j*y[newaxis,:]
    isMandelbrot = zeros((nx,ny), bool)
    mandelbrotValues = zeros((nx,ny), int)
    isMandelbrot[:,:] = True

# Итерация Мандельброта
    z = c
    for t in range(N_max):
        z = z**2 + c
        for i in range(nx):
            for j in range(ny):
                if abs(z[i,j]) > some_threshold and  isMandelbrot[i,j]:
                    mandelbrotValues[i,j] = t
                    isMandelbrot[i,j] = False
    for i in range(nx):
```

```
        for j in range(ny):
            if isMandelbrot[i,j]:
                mandelbrotValues[i,j] = N_max

    return mandelbrotValues

mandelbrot_set = compute_mandelbrot(20, 200., 601, 401)

plt.imshow(mandelbrot_set.T, extent=[-2, 1, -1.5, 1.5])
plt.gray()
plt.show()
```

```
/tmp/ipykernel_7683/936654850.py:15: RuntimeWarning: overflow encountered in squa
re
  z = z**2 + c
/tmp/ipykernel_7683/936654850.py:15: RuntimeWarning: invalid value encountered in
square
  z = z**2 + c
```