```
In [1]:
import sympy
from sympy import I
import numpy as np
import matplotlib.pyplot as plt
```

Занятие 10

Алгебра

Комплексные числа в sympy

Алгебраическая форма комплексного числа

$z = a + bi$, a и b вещественные числа, $i = \sqrt{-1}$ мнимая единица, z - комплексное число. В sympy мнимая единица записывается I, в python j.

Пример 1.

Даны комплексные числа

$$z_1 = 2 + 3i, \quad z_2 = -3 + i, \quad z_3 = 5i, \quad z_4 = -i, \quad z_5 = -5 - 10i,$$

вычислить

$$3z_1 - 5z_2, \quad z_2^2 - 3z_1^3, \quad \sqrt{z_3}, \quad z_4^{1/6}, \quad z_5^{-2/7}.$$

```
In [2]:
z1 = 2 + 3*I
z2 = -3 + I
z3 = 5*I
z4 = -I
z5 = -5 - 10*I
display(3*z1 - 5*z2, z2**2 - 3*z1**3, z3**sympy.S.Half, z4**(sympy.S(1)/6), z5**(-sympy.S(2)/7))
```

$21 + 4i$

$-3(2+3i)^3 + (-3+i)^2$

$\sqrt{5}\sqrt{i}$

$\sqrt[6]{-i}$

$\frac{1}{(-5-10i)^{\frac{2}{7}}}$

Упростим выражение $z_2^2 - 3z_1^3$

```
In [3]:
display(*((z2**2 - 3*z1**3).simplify(), (z5**(-sympy.S(2)/7)).simplify()))
```

$146 - 33i$

$\frac{5^{\frac{5}{7}}}{5(-1-2i)^{\frac{2}{7}}}$

Некоторые выражения получились в алгебраической форме, но не все. Приведем в подходящий вид их немного позже.

Пример 2.

Вычисления Примера 1 проделаем для мнимой единицы из python (numpy использует комплексные числа python, своих отдельных нет). Для получения комплексных чисел в python есть встроенная функция complex с аргументами - вещественной и мнимой частями комплексного числа.

```
In [4]:
z1 = complex(2, 3)
z2 = complex(-3, 1)
z3 = complex(0, 5)
z4 = complex(0, -1)
z5 = complex(-5, - 10)
display(3*z1 - 5*z2, z2**2 - 3*z1**3, z3**(1/2), z4**(1/6), z5**(-2/7))
```

```
(21+4j)

(146-33j)

(1.5811388300841898+1.5811388300841898j)

(0.9659258262890683-0.25881904510252074j)

(0.419301296507299+0.2754745230972611j)
```

Заметим, что у комплексного числа n различных корней степени n, а у нас получился только один.

Пример 3.

Для получения всех корней n-й степени комплексного числа в sympy есть функция root, ее аргументы - число (или выражение), степень корня и необязательный параметр - номер корня, по умолчанию 0, что соответствует главном значению корня. Используем ее для получения всех четвертой степени из $1 - i$:

```
In [5]:
n = 4
roots_list3 = [sympy.root(1 - I, n, k) for k in range(n)]
display(*roots_list3)
```

$\sqrt[4]{1-i}$

$i\sqrt[4]{1-i}$

$-\sqrt[4]{1-i}$

$-i\sqrt[4]{1-i}$

Для получения значения выражения, содержащего комплексные числа, в алгебраической форме можно использовать функцию expand_complex:

```
In [6]:
roots3 = [sympy.expand_complex(item) for item in roots_list3]
display(*roots3)
```

$\sqrt[8]{2}\cos\left(\frac{\pi}{16}\right) - \sqrt[8]{2}i\sin\left(\frac{\pi}{16}\right)$

$\sqrt[8]{2}\sin\left(\frac{\pi}{16}\right) + \sqrt[8]{2}i\cos\left(\frac{\pi}{16}\right)$

$-\sqrt[8]{2}\cos\left(\frac{\pi}{16}\right) + \sqrt[8]{2}i\sin\left(\frac{\pi}{16}\right)$

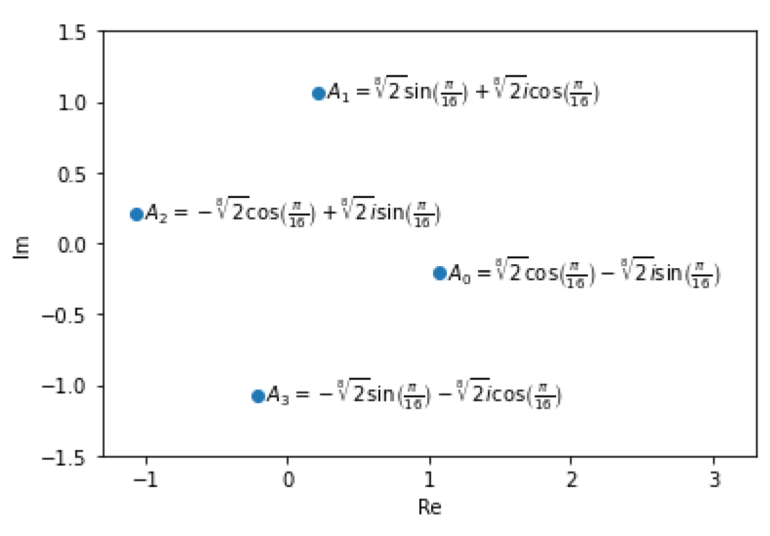
$-\sqrt[8]{2}\sin\left(\frac{\pi}{16}\right) - \sqrt[8]{2}i\cos\left(\frac{\pi}{16}\right)$

Пример 4.

Изобразим на координатной плоскости точки, соответствующие корням из Примера 3, подпишем точки номерами корней и их формулами.

```
In [7]:
roots_x = [sympy.re(item) for item in roots_list3]
roots_y = [sympy.im(item) for item in roots_list3]
plt.scatter(roots_x, roots_y)
for k, x_coord in enumerate(roots_x):
    plt.annotate('$A_' + str(k) + ' = ' + sympy.latex(roots3[k]) + '$',
            xy=(x_coord, roots_y[k]),
            xytext=(x_coord + 0.05, roots_y[k] - 0.05))
plt.axis('equal')
plt.xlabel("Re")
plt.ylabel("Im")
plt.xlim(-2.5, 4.5)
plt.ylim(-1.5, 1.5)
```

```
Out[7]: (-1.5, 1.5)
```



Пример 5.

Для получения всех корней многочлена с учетом кратности в sympy есть функция roots (в sympy.polys.polyroots). Используем ее для получения всех корней третьей степени из i:

```
In [8]:
z = sympy.Symbol('z')
display(*sympy.roots(z**3 - I))
```

$\frac{\sqrt{3}}{2} + \frac{i}{2}$

$-i$

$-\frac{\sqrt{3}}{2} + \frac{i}{2}$

Пример 6.

Для получения значения выражения, содержащего комплексные числа, в алгебраической форме можно использовать функцию expand_complex, а можно выделить вещественную и мнимую часть выражения и составить из них комплексное число:

```
In [9]:
res6 = (3 - 7*I)**(-sympy.S(2)/7)
sympy.re(res6) + sympy.im(res6)*I
```

Out[9]:

$$\frac{58^{\frac{6}{7}}\cos\left(\frac{2\operatorname{atan}\left(\frac{7}{3}\right)}{7}\right)}{58} + \frac{58^{\frac{6}{7}}i\sin\left(\frac{2\operatorname{atan}\left(\frac{7}{3}\right)}{7}\right)}{58}$$

Тригонометрическая форма комплексного числа

$$z = r(\cos(\varphi) + i\sin(\varphi)),$$

где r - модуль числа, φ - аргумент (угол)

Пример 7.

Запишем в тригонометрической форме комплексное число $\sqrt{3} - i$

```
In [10]:
z7 = 3**sympy.S.Half - 1*I
r0 = sympy.functions.Abs(z7)
phi0 = sympy.functions.arg(z7)
r, phi = sympy.symbols('r phi')
z7_trig = r*(sympy.cos(phi) + sympy.sin(phi)*I)
display(z7_trig)
print('r =', r0, ' phi =', phi0)
```

$r(i\sin(\phi) + \cos(\phi))$

```
r = 2  phi = -pi/6
```

Проверим подстановкой, что это то же самое число:

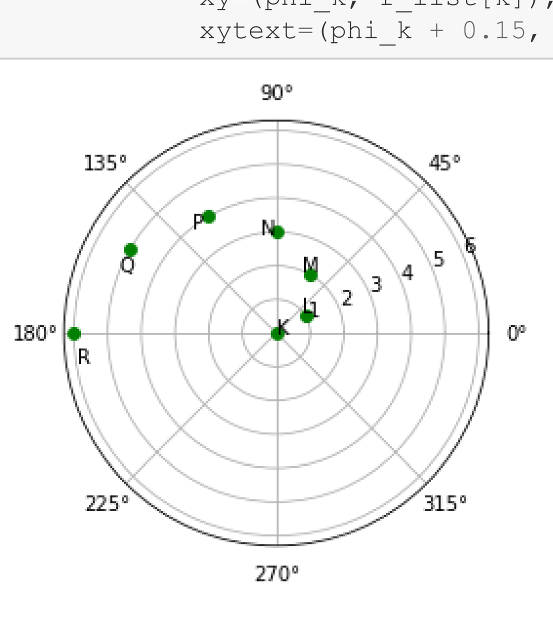
```
In [11]:
z7_trig.subs({r: r0, phi: phi0})
```

Out[11]: $\sqrt{3} - i$

Пример 8.

Изобразим комплексные числа $k(\sin(k\pi/6) + i\cos(k\pi/6))$, $k = 0, \ldots, 6$, заданные в тригонометрической форме, в полярной системе координат. Подписать точки K, L, M, N, P, Q, R.

```
In [12]:
n = 7
names = ('K', 'L', 'M', 'N', 'P', 'Q', 'R')
arg_list = [k*np.pi/6 for k in range(n)]
r_list = [k for k in range(n)]
plt.polar(arg_list, r_list, 'go')
for k, phi_k in enumerate(arg_list):
    plt.annotate(names[k],
            xy=(phi_k, r_list[k]),
            xytext=(phi_k + 0.15, r_list[k]))
```



Обратим внимание, что в полярной системе координат координаты - угол и радиус (расстояние от точки о начала координат), поэтому для координат подписей к точкам тоже нужно использовать такие полярные координаты, а не x и y.

Комплексно-сопряженное число

$$\bar{z} = a - bi$$

комплексно-сопряженное число для $z = a + bi$.

Пример 9.

Найдем комплексно-сопряденные числа для $-3 + 2i$, $(\sqrt{2} - i)^3$, $(1 + i)^{1/3}$

```
In [13]:
display(sympy.conjugate(-3 + 2*I),
        sympy.conjugate(sympy.expand_complex((2**(sympy.S.Half) - 1*I)**3)),
        sympy.conjugate((1 + 1*I)**(1/sympy.S(3))))
```

$-3 - 2i$

$-\sqrt{2} + 5i$

$\sqrt[6]{2}\left(\frac{\sqrt{2}}{4} + \frac{\sqrt{6}}{4}\right) - \sqrt[6]{2}i\left(-\frac{\sqrt{2}}{4} + \frac{\sqrt{6}}{4}\right)$

Произведение комплексного числа и его комплексно-сопряженного

равно квадрату модуля комплексного числа.

Пример 10.

Проверим $\bar{z} \cdot z = |z|^2$.

```
In [14]:
a, b = sympy.symbols('a b', real=True)
z8 = a + b*I
display(sympy.expand_complex(sympy.conjugate(z8)*z8), sympy.functions.Abs(z8)**2)
```

$a^2 + b^2$

$a^2 + b^2$