Plan

1 Complex numbers and roots of equations
2 Polar coordinates and de Moivre's formula

① Complex numbers and roots of equations

Ex:

$$x^2 - 4x + 5 = 0$$
$$x^2 - 4x = -5$$
$$x^2 - 4x + 4 = -5 + 4$$
$$(x-2)^2 = -1$$

$$x = \frac{4 \pm \sqrt{4^2 - 4 \cdot 5}}{2} = \frac{4 \pm \sqrt{-4}}{2} = \frac{4 \pm \sqrt{4} \cdot \sqrt{-1}}{2} = \underline{2 \pm i}$$

$i = \sqrt{-1}$:

$$x - 2 = \pm\sqrt{-1}$$
$$x - 2 = \pm i$$
$$x = 2 \pm i$$

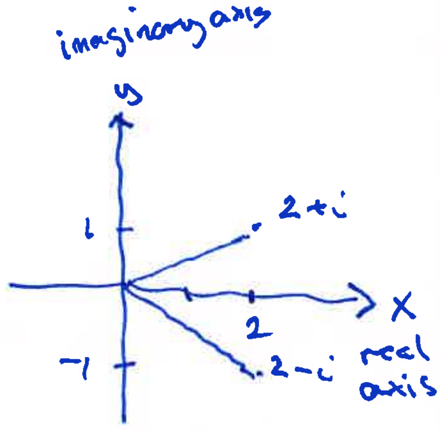
$$x_1 = \underline{2+i}, \quad x_2 = \underline{2-i}$$

Complex number: $z = \overset{x}{a} + \overset{y}{b} \cdot i$

where a, b are real numbers

$\updownarrow$

(a, b) in $\mathbb{R}^2$



complex plane

Ex: Computation:

$z_1 = 2 + 3i, \quad z_2 = 1 - i$

$z_1 + z_2 = 3 + 2i$

$z_1 - z_2 = 1 + 4i$

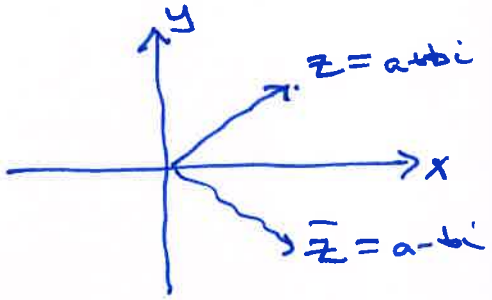
$$z_1 \cdot z_2 = (2+3i)\cdot(1-i) = 2 + 3i - 2i - 3i^2 = \underline{5 + i}$$

$i^2 = -1$

Ex: $(1+i)\cdot(1-i) = 1 + \cancel{i} - \cancel{i} - i^2 = \underline{2}$

Defn: $z = a+bi$

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



Formula:

$$\boxed{z\cdot\bar{z} = |z|^2}$$

$$\boxed{|z| = \sqrt{a^2+b^2}}$$ modulus of z

$$(a+bi)(a-bi)$$
$$= a^2 + \cancel{abi} - \cancel{abi} - b^2 i^2$$
$$= a^2 + b^2$$

Python:

```
z = 1+j
z = complex(1,1)
```

module: cmath

A : random matrix of real numbers

```
np.linalg.eig(A)
```

Division:

$$\frac{1+i}{1-i}\cdot\frac{1+i}{(1+i)} = \frac{1+i+i+i^2}{2} = \frac{\cancel{1}+2i-\cancel{1}}{2} = \underline{\underline{i}}$$

$$\frac{1+i}{2+3i} = \frac{(1+i)(2-3i)}{(2+3i)(2-3i)} = \frac{2+2i-3i+3}{4+9} = \frac{5-i}{13}$$

$$= \frac{5}{13} + i\cdot\left(\frac{-1}{13}\right)$$

Roots of equations:

Ex: $x^3 - 5x^2 + 17x - 13 = 0$ $\quad$ $\underline{x=1}$ is a solution

$(x-1)(x^2-4x+13) = 0$

$\underline{x=1}$ or $x^2-4x+13=0$

$x^2-4x+4 = -9$

$(x-2)^2 = -9$

$x-2 = \pm\sqrt{-9} = \pm 3i$

$x = \underline{2 \pm 3i}$

Roots: $\underline{x_1 = 1}$ $\quad$ $x_2 = \underline{2+3i}$, $\quad$ $x_3 = \underline{2-3i}$

Result: Any polynomial equation $a_n x^n + a_{n-1}x^{n-1} + \dots + a_1 x + a_0 = 0$ with real coeff $a_n, a_{n-1}, \dots, a_2, a_1, a_0$ in $\mathbb{R}$ and $a_n \neq 0$ has:

i) n complex solutions

ii) if z is a solution, then $\bar{z}$ is a solution

Ex: $x^3 = 1$ $\qquad$ $x^3 - 1 = 0$ $\qquad$ $\underline{x=1 \text{ is a solution}}$

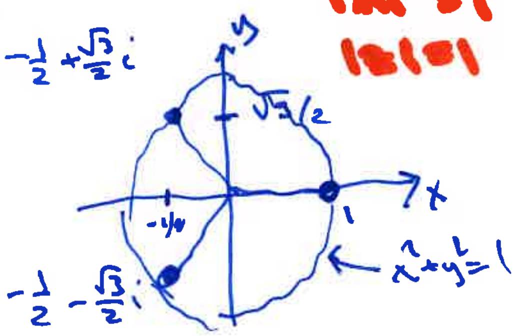
$|z|^3 = 1$

$|z| = 1$

$(x-1)\cdot(x^2+x+1) = 0$

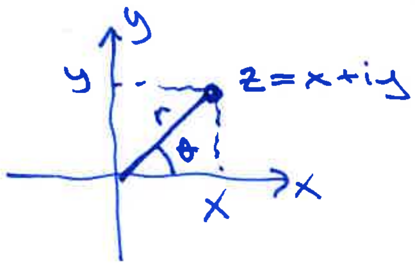
$x = 1$, $\quad x = \dfrac{-1 \pm \sqrt{1-4\cdot 1}}{2}$

$= \dfrac{-1 \pm \sqrt{-3}}{2} = \dfrac{-1 \pm \sqrt{3}\cdot i}{2}$



$\underline{x_1 = 1}$, $\quad x_2 = \underline{-\frac{1}{2} + \frac{\sqrt{3}}{2} i}$, $\quad x_3 = \underline{-\frac{1}{2} - \frac{\sqrt{3}}{2} i}$

2 Polar coordinates and de Moivre's formula



(x,y) : cartesian coordinates

(r,θ) : polar coordinates

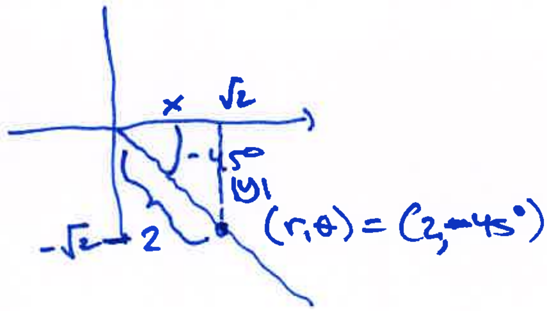
$r = |z| = \sqrt{x^2+y^2}$
distance to $(0,0)$

θ : angle with the positive x-axis (counterclockwise)

Ex:

$(r,\theta) = (2,-45^\circ)$

Note: θ is only defined up to integer multiples of 360°.

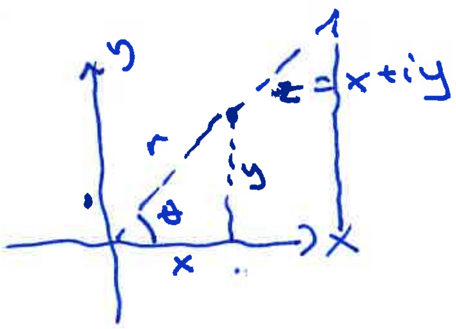


$x = |y| \Rightarrow x^2+y^2 = 2^2$
$2x^2 = 4$
$x^2 = 2$
$x = \sqrt{2}\,,\ y = -\sqrt{2}$

$(r,\theta) = (2,-45^\circ) = (2, 315^\circ)$

$\Updownarrow$

$(x,y) = (\sqrt{2}, -\sqrt{2})$



Defn:

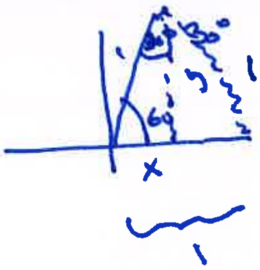
$$\boxed{\begin{aligned}\sin\theta &= y/r \\ \cos\theta &= x/r\end{aligned}}$$

$$\boxed{\begin{aligned}x &= r\cdot\cos\theta \\ y &= r\cdot\sin\theta\end{aligned}}$$

Ex:

$$\sin(45°) = \frac{\sqrt{2}/2}{1} = \frac{\sqrt{2}}{2} = \frac{1}{\sqrt{2}} \simeq \underline{0.707}$$

$$\cos(45°) = \frac{x}{r} = \frac{\sqrt{2}}{2} = \frac{1}{\sqrt{2}} \simeq \underline{0.707}$$



$x = \underline{1/2}$

$y^2 + x^2 = 1^2$

$y^2 + \frac{1}{4} = 1$

$y^2 = 3/4$

$y = \sqrt{3}/2$

$\underline{x = y}$: $2x^2 = 1$

$x^2 = 1/2$

$x = \sqrt{1/2} = \frac{1}{\sqrt{2}} = \frac{\sqrt{2}}{2} = y$

$\sin(60°) = \underline{\sqrt{3}/2}$

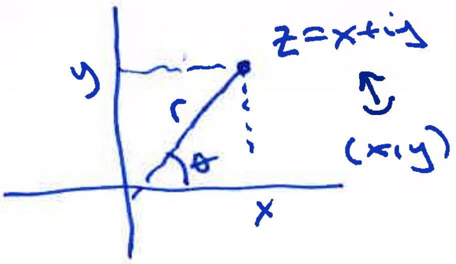
$\cos(60°) = \underline{1/2}$

$$\boxed{\begin{aligned} x &= r \cdot \cos\theta \\ y &= r \sin\theta \end{aligned}}$$

Complex numbers:

$z = x + iy = r \cdot \cos\theta + i \cdot r \sin\theta$

$$z = r \cdot (\cos\theta + i \sin\theta)$$

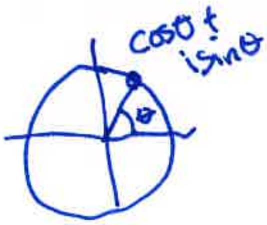


complex numbers using polar coordinates

$|z| = |x + iy|$
$= \sqrt{x^2 + y^2}$

Note:

i) $|z| = r$ since $|\cos\theta + i\sin\theta| = 1$



$$(\cos\theta)^2 + (\sin\theta)^2 = \cos^2\theta + \sin^2\theta = \left(\frac{x}{r}\right)^2 + \left(\frac{y}{r}\right)^2 = \frac{x^2 + y^2}{r^2} = 1$$

Could define this by

$e^x = 1 + x + x^2/2 + x^3/6 + \ldots$

ii) $\boxed{e^{i\theta} = \cos\theta + i\sin\theta}$

de Moivre's formula / Euler's formula:

$$z = r\cdot(\cos\theta + i\sin\theta) \Rightarrow z^n = r^n\left(\cos(n\theta) + i\sin(n\theta)\right)$$

$$z = r\cdot e^{i\theta} \Rightarrow z^n = r^n \cdot e^{i\cdot n\theta}$$

1.

$$z = r\cdot(\cos\theta + i\sin\theta)$$

Ex: $z^5 = 1$

$$r^5\cdot(\cos 5\theta + i\sin 5\theta) = 1$$

Modulus:

$r^5 = 1$

$r = 1$

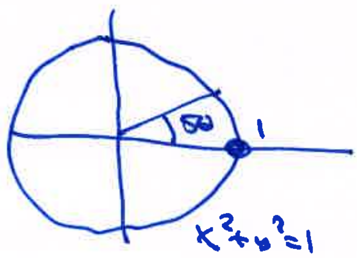
Angles:

$$\cos 5\theta + i\sin 5\theta = \cos 0^\circ + i\sin 0^\circ$$

$$\Rightarrow 5\theta = 0^\circ + k\cdot 360^\circ, \quad k \text{ int.}$$

$$\theta = \frac{0^\circ + k\cdot 360^\circ}{5}$$

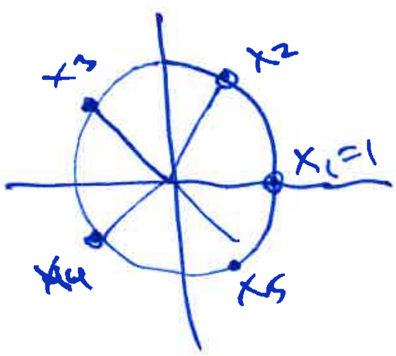
$$= k\cdot 72^\circ, \quad k = 0, 1, 2, 3, 4,$$



$$\theta = 0^\circ, 72^\circ, 144^\circ, 216^\circ, 288^\circ,$$

$$\Downarrow$$

$$z = 1\cdot(\cos\theta + i\sin\theta), \quad \theta = 0, 72^\circ, 144^\circ, 216^\circ, 288^\circ.$$



```python
import numpy as np
import math
import cmath

z = 3 - 2j
w = complex(3,1)

z*w

z/w

# The modulus of z
abs(z)

# The angle of z, in radians
cmath.phase(z)

# The angle of z, in degrees
math.degrees(cmath.phase(z))

# The polar coordinates of z, with angles in radians
cmath.polar(z)

# Examples of square (real) matrices
A = np.array([[0,-1],[1,0]])
B = np.random.randn(3,3)

print(A)
print(B)

# Eigenvalues of the matrices (in Lecture 4 next Friday)
np.linalg.eigvals(A)
np.linalg.eigvals(B)
```

Plan

① Introduction:

Lecture 1-12
Fri 08-10
2+1 maths

(with GRA6035)

+

Lecture A-C
Thu 13-15
2+1 maths + python

Small class

Exams:

Final exam
(end of Nov)
80%

Home exam
(midterm)
20%

② Python: NumPy and ndarrays

Book: Python for data analysis — for this lecture: Ch. 4

* NumPy: numerical Python package

```
import numpy as np
```

* ndarrays: n-dimensional arrays (data structure)

$$A = \begin{pmatrix} 1 & 1 & 1 & 3 \\ 1 & 2 & 4 & 7 \\ 1 & 3 & 9 & 13 \end{pmatrix}$$

```
A = np.array([[1,1,1,3],[1,2,4,7],
              [1,3,9,13]])

B = np.random.randn(3,5)
```

data types: A.dtype → np.int32 (integer)
B.dtype → np.float64 (floats)

rows, cols: A[1] → [1,2,4,7] (starts at 0)
A[1,1] → 2 — " —
A[:,1] → $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

pointers, not copies:

```
C = A
C[1,1] = 5
```

$A \to \begin{pmatrix} 1 & 1 & 1 & 3 \\ 1 & 5 & 4 & 7 \\ 1 & 3 & 9 & 13 \end{pmatrix}$

```
C = A.copy()
```

③ Project: Coding Gaussian process

Input: matrix (as an array)

Output: an echelon form

Functions:

```
def f(x):
    return(x+1)
```

$f(3) \to 4$

```
def fact(n):
    if n==1:
        return(1)
    return(n*fact(n-1))
```

$\text{fact}(3) \to 3 \cdot \text{fact}(2)$

$\to 3 \cdot \underline{2 \cdot \text{fact}(1)}$

$\to 3 \cdot 2 \cdot 1 = 6$

recursive fn.

Row operations:

```
def Rmult(matrix, i, c)
def Radd(matrix, i, j, c)
def Rswitch(matrix, i, j)
```

Note: i,j are actual row numbers / col. numbers

↓

i-1, j-1 in Python

```
In [12]: import numpy as np
```

```
In [13]: A=np.array([[1,1,1,3],[1,2,4,7],[1,3,9,13]])
```

```
In [14]: def Rmult(matrix,i,c):
             matrix[i-1]=matrix[i-1]*c
             return(matrix)
```

```
In [15]: def Radd(matrix,i,j,c):
             matrix[j-1]=matrix[j-1]+matrix[i-1]*c
             return(matrix)
```

```
In [16]: def Rswitch(matrix,i,j):
             r = matrix[i-1].copy()
             matrix[i-1]=matrix[j-1]
             matrix[j-1]=r
             return(matrix)
```

```
In [17]: A

Out[17]: array([[ 1,  1,  1,  3],
                [ 1,  2,  4,  7],
                [ 1,  3,  9, 13]])
```

```
In [18]: Radd(A,1,2,-1)

Out[18]: array([[ 1,  1,  1,  3],
                [ 0,  1,  3,  4],
                [ 1,  3,  9, 13]])
```

```
In [19]: Radd(A,1,3,-1)

Out[19]: array([[ 1,  1,  1,  3],
                [ 0,  1,  3,  4],
                [ 0,  2,  8, 10]])
```

```
In [20]: Radd(A,2,3,-2)

Out[20]: array([[1, 1, 1, 3],
                [0, 1, 3, 4],
                [0, 0, 2, 2]])
```

```
In [ ]: def Gauss(A):
            ...
```