

a brief reminder on complex numbers.

complex numbers are central to Fourier analysis, and their understanding is needed to fully comprehend the beauty of Fourier analysis

Real numbers are *regular* numbers, going from $-\infty$ to $+\infty$ through -12.5 , 0 , 1 , π , and every number you might want to use. They are said to belong to \mathbb{R} , the set of all real numbers, \mathbb{R} can be seen as a line, going from $-\infty$ to $+\infty$.

If Reals are on a line, **Complex** numbers are on a plane. As any plane, the coordinates are defined on two axes, the horizontal axis is the \mathbb{R} line, the vertical one is the *Imaginary* axis, also holding real numbers, and labeled with i . This plane is called \mathbb{C} the complex plane.

A complex number z (a point in this plane) is thus described with two numbers, a and b :

$$z = a + ib$$

a is the real part, and b the imaginary part.

The position of the complex point z can also be described by its *module* (the distance to the center)

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

and the angle with the horizontal axis, called the *argument*, usually noted with a greek letter:

$$\arg(z) = \arctan\left(\frac{b}{a}\right) = \theta$$

It is defined only between $[0, 2\pi]$ as any other angle.

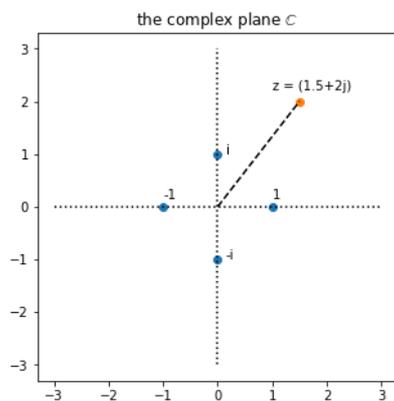
A complex number is fully determined by a and b , as well as by its modulus and argument. This is noted using the *Euler* notation:

$$z = a + ib$$
$$z = |z|e^{i \arg(z)} = |z|e^{i\theta}$$

see also: [Wikipedia:Complex plane \(https://en.wikipedia.org/wiki/Complex_plane\)](https://en.wikipedia.org/wiki/Complex_plane)

```
In [2]: 1 # let's draw this
2 %pylab inline
3 figure(figsize=(5,5))
4 plot([-3,3],[0,0],':k') # the real axis
5 plot([0,0],[-3,3],':k') # the imaginary axis
6 scatter([1,0,-1,0],[0,1,0,-1])
7 text(1,0.15,'1')
8 text(-1,0.15,'-1')
9 text(0.15,1,'i')
10 text(0.15,-1,'-i')
11 title('the complex plane $\mathbb{C}$')
12 z = 1.5 + 2j # i is noted j in python
13 scatter(z.real, z.imag)
14 plot([0,z.real],[0,z.imag], '--k')
15 text(1.2,2,'z = %s'%(str(z))):
```

Populating the interactive namespace from numpy and matplotlib



The supplement to a simple cartesian plane, is that you can do arithmetic with complex numbers. Complex numbers can be added and multiplied, they form an *algebra*.

The addition operation is simply the addition of both components. The multiplication of two complex numbers (noted here *) gives a new complex number,

$$z_1 * z_2 = a_1 a_2 - b_1 b_2 + i(a_1 b_2 + b_1 a_2)$$

If you work out the equations in sine and cosine, you will find that the modulus have been multiplied and the arguments added. *This explains the simplicity of the Euler notation:*

$$\begin{aligned} z_1 * z_2 &= |z_1| e^{i\theta_1} * |z_2| e^{i\theta_2} \\ z_1 * z_2 &= |z_1| |z_2| e^{i(\theta_1 + \theta_2)} \\ z_1 * z_2 &= |z_1 z_2| e^{i(\theta_1 + \theta_2)} \end{aligned}$$

Adding arguments (angles) is a rotation, so multiplication by a complex number is a dilatation (by the modulus - *just like multiplication by a real number*) and a rotation (by the argument).

Multiplication by a complex numbers of modulus 1.0 is thus just a rotation by θ . Those special numbers are located on the **unity circle** (of radius 1.0 and centered on (0,0)), and have a very special role.

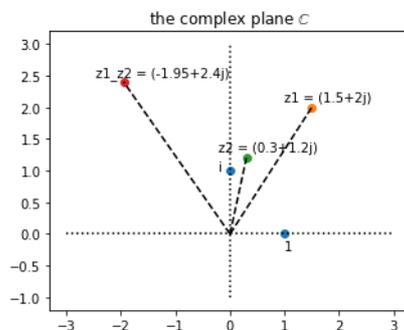
A very special number is i equal to $e^{i\pi/2}$, the basis for the imaginary numbers. It corresponds to a rotation by 90°. Of course multiplying i by itself, (rotating it by 90°) you end-up on the horizontal axis, backward. This is the well known definition:

$$i^2 = -1$$

or the mind boggling equation mixing $e \pi$ and i :

$$e^{i\pi} = -1$$

```
In [3]: 1 # let's show the effect of multiplication
2 figure(figsize=(5,4))
3 plot([-3,3],[0,0],':k') # the real axis
4 plot([0,0],[-1,3],':k') # the imaginary axis
5 scatter([1,0],[0,1])
6 text(1,-0.25,'1')
7 text(-0.2,1,'i')
8 title('the complex plane $\mathbb{C}$')
9 z1 = 1.5 + 2j
10 z2 = 0.3 + 1.2j
11 z1_z2 = z1*z2
12 for nz in ('z1','z2','z1_z2'):
13     z = eval(nz)
14     scatter(z.real,z.imag)
15     plot([0,z.real],[0,z.imag], '--k')
16     text(z.real-0.5,z.imag+0.1,'%s = %s'%(nz,str(z)))
```



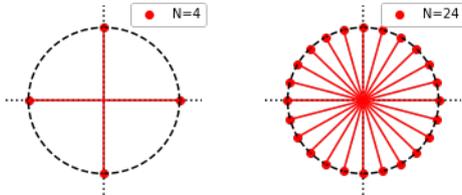
In particular, the points located at $e^{2i\pi \frac{k}{N}}$ are the N^{th} roots of 1, as any of these, raised to the power N is equal to 1.

```

In [4]: 1 #let's show the roots of 1.0 on the unity circle
2 f,(ax1,ax2) = subplots(ncols=2,figsize=(6.6,3))
3 t = linspace(0,2*pi,100)
4 for N,ax in ((4,ax1),(24,ax2)): # for each figure, with different N
5     ax.plot([-1.3,1.3],[0,0],':k') # the real axis
6     ax.plot([0,0],[-1.3,1.3],':k') # the imaginary axis
7     ax.plot(cos(t),sin(t),'--k') # the unity circle
8     k = arange(0,N) # compute the roots
9     z = exp(2j * pi * k/N) # e^(2 i pi k / N)
10    ax.scatter(z.real, z.imag, c='r',edgecolors='r',label='N=%d'%N) # draw roots
11    [ ax.plot([0,zk.real],[0,zk.imag], 'r') for zk in z]
12    ax.legend()
13    ax.set_axis_off()
14 f.text(0.25,1,'showing the $N^{\text{th}}$ roots of 1 on the unity circle');
15

```

showing the N^{th} roots of 1 on the unity circle



```

In [5]: 1 # let's make it interactive
2 def Nroot(k=1,N=8):
3     f,(ax) = subplots(figsize=(3,3))
4     t = linspace(0,2*pi,100)
5     ax.plot([-1.3,1.3],[0,0],':k') # the real axis
6     ax.plot([0,0],[-1.3,1.3],':k') # the imaginary axis
7     ax.plot(cos(t),sin(t),'--k') # the unity circle
8     z = exp(2j * pi / N) # e^(2 i pi / N)
9     zk = z**k
10    ax.scatter(zk.real, zk.imag, c='r',edgecolors='r') # draw roots
11    ax.plot([0,zk.real],[0,zk.imag], 'r')
12    ax.set_axis_off()
13    ax.set_title(r'showing $e^{2i\pi \text{ %d}/\text{ %d}}$ on the unity circle'% (k,N));
14 from ipywidgets import interactive
15 interactive(Nroot.k=(0.24).N=(2.48))

```

A Jupyter widget could not be displayed because the widget state could not be found. This could happen if the kernel storing the widget is no longer available, or if the widget state was not saved in the notebook. You may be able to create the widget by running the appropriate cells.

```

In [ ]: 1

```