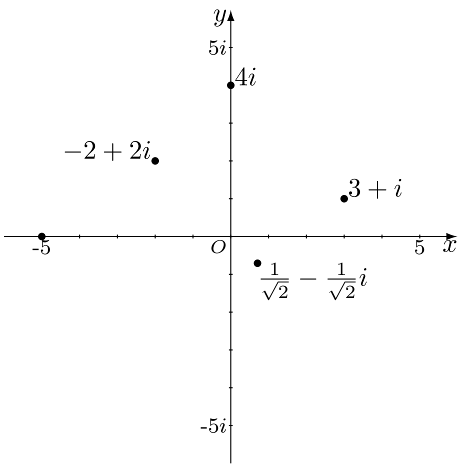
## 8.2 複素数

実数と虚数は数直線上に表せるが、複素数とは実際どんな数なのだろう。複素数は数直線ではなく、**複素平面**上に表す。




たとえば、この中のいくつかを使って **Python3** に計算してもらおう。

[py script]

```
>>> a = 3+1j
>>> b = -2+2j
>>> c = -1+2j
>>> a + b, a - b, a * b, a / b, a / c
((1+3j), (5-1j), (-8+4j), (-0.5-1j), (-0.2-1.4j))
```

**Python3** の演算子 "/" は整数どうしの計算結果が整数でも小数点をつけて表したが、複素数では整数には小数点はつかない。もちろん計算結果が小数ならそのまま表示する。

問題は、複素数はどんな規則で計算されるかということだ。規則といっても、一般的な複素数 $a+bi$ と $c+di$ で考えれば

$$
\begin{aligned}
(a+bi)+(c+di) &= (a+c)+(b+d)i \\
(a+bi)-(c+di) &= (a-c)+(b-d)i \\
(a+bi)(c+di) &= (ac-bd)+(ad+bc)i
\end{aligned}
$$

であることはすぐ分かるし、割り算の場合は、

$$
\begin{aligned}
(a+bi)/(c+di) &= \frac{(a+bi)(c-di)}{(c+di)(c-di)} \\
&= \frac{ac+bd}{c^2+d^2}+\frac{-ad+bc}{c^2+d^2}i
\end{aligned}
$$

のように、分子・分母に $c-di$ を掛ければ計算規則が確立するので、大した問題ではない。ちなみに、$c+di$ と $c-di$ の符号違いのペアを**共役複素数**といい、この先の道で大活躍するものである。

問題は数値がどうなるかではなく、数値が複素平面のどこに現れるか、なのである。たとえば $(3+i)(-2+2i)=-8+4i$ だ。どうして変な場所に現れてしまうのだろう。

結論を言おう。複素数は "点" ではなく "**ベクトル**" である。ベクトルは大きさと向きがある矢線なので、足したり引いたりすることは、矢線を継ぎ足すことである（引き算は逆向きにして継ぎ足す)。それは、実際に矢線を描いてみれば分かる。小学校で力の合成に矢線を使って作図したことを思い出すかもしれない。

ベクトルの大きさとは、単純に矢線の長さである。**三平方の定理**によって $a+bi$ の大きさは $\sqrt{a^2+b^2}$ で求められる。具体的に $3+i$ の大きさは $\sqrt{3^2+1^2}=\sqrt{10}$、$-2+2i$ の大きさは $\sqrt{(-2)^2+2^2}=2\sqrt{2}$ である。だから、$(3+i)(-2+2i)=-8+4i$ となるのは大きさの観点から見て正しい。なぜなら、$-8+4i$ の大きさは $\sqrt{(-8)^2+4^2}=4\sqrt{5}$ なので、たしかに $\sqrt{10}\times 2\sqrt{2}=4\sqrt{5}$ だからだ。割り算 $(3+i)/(-2+2i)=-0.5-i$ にしても、$-0.5-i$ の大きさは $\sqrt{\left(-\frac{1}{2}\right)^2+1^2}=\frac{\sqrt{5}}{2}$ なので、たしかに $\sqrt{10}/2\sqrt{2}=\frac{\sqrt{5}}{2}$ だ。

では、複素数が現れる "場所" はどのように決まるのだろう。これにはベクトルが持つ**偏角**が関係している。偏角とは、ベクトルが $x$ 軸の正の方向となす角をいう。これは少し分かりづらいし、正確に求めるには三角関数を用いて計算しなくてはならない。ここでは大まかに見ておこう。矢線を引けば大体の雰囲気がつかめると思うが、$3+i$ の偏角はおよそ $18^\circ$ で、$-2+2i$ の偏角はちょうど $135^\circ$ である。そこで、ふたつの複素数の積はそれぞれの偏角の和の位置に、商はそれぞれの偏角の差の位置に現れる。$3+i$ と $-2+2i$ の偏角の和はおよそ $153^\circ$、偏角の差はおよそ $-117^\circ$ である。これらは、積 $-8+4i$ の偏角と商 $-0.5-i$ の偏角に等しいのである。

話は変わるが、実数の単位は何であろうか。1 である。では、虚数の単位は？　もちろん $i$ である。ならば、複素数の単位は何であろう。$1+i$ では単位とは言えない。というのは、単位とは大きさが 1 のものを指すと思うからだ。ベクトル $1+i$ と同等で大きさが 1 となると、それは $\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i$ である。さて、再び強引な話になるけれど、単位 1 の符号は $+$、単位 $i$ の符号は $i$ であるとは前に言ったね。じゃあ、単位 $\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i$ の符号って何だろう？　実際は、符号という考えはおかしいのだが、$i$ の符号をそのまま $i$ と見たので、$\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i$ はそのまま符号扱いしておこう。すると符号の積が

$$\left(\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i\right)\times\left(\frac{1}{\sqrt{2}}+\frac{1}{\sqrt{2}}i\right)=i$$

となる。やった、2 乗して $i$ になる符号の登場だ。そして、符号扱いした複素数は 8 回の積を経て 1 へたどり着く。その様子は **Python3** で見よう。

[py script]

```
>>> import math
>>> sgn = 1 / math.sqrt(2) + 1j / math.sqrt(2)
>>> a = 1
>>> for i in range(8):
...   a = a * sgn
...   print(a)
...
(0.7071067811865475+0.7071067811865475j)
0.9999999999999998j
(-0.7071067811865474+0.7071067811865474j)
(-0.9999999999999997+0j)
(-0.7071067811865472-0.7071067811865472j)
-0.9999999999999996j
(0.7071067811865471-0.7071067811865471j)
(0.9999999999999993+0j)
```

多少の計算誤差があるようなのでぴったりではないが、結果の 2, 4, 6, 8 行目は `1j`, `(-1+0j)`, `-1j`, `(1+0j)` と表示してほしかったね[1]。

まとめるとこうだ。$(-)^2 = (+)$ とは計算上の約束であり、$(?)^2 = (-)$ となる符号はないものとしてきた。逆の見方をすれば、$(+)$ は $(-)$ どうしの積にできるが、$(-)$ はそれ以上分解できないものと考えてきたわけである。しかし定義を拡張することで、$(-) = (i)^2$ との約束を取り付けることができた。これは、$(-)$ は $(i)$ の積に分解できると見なしてよい。まるで、これ以上分解できないと思っていた原子が、陽子と中性子に分けられることが分かったように。

さらに、$(i)$ を符号と見ると、$\left(\dfrac{1}{\sqrt{2}} + \dfrac{1}{\sqrt{2}}i\right)^2 = (i)$ だから、$(i)$ でさえ別の符号どうしの積にできるともいえる。ただし、複素数で表される数を符号とみなすには無理があるけれど。このことは、原子の根源であると思った陽子でさえ、さらにいくつかのクォークに分けられることに似ている。あろうことか、複素数はさらにその上を行く。$\left(\dfrac{\sqrt{2+\sqrt{2}}}{2} + \dfrac{\sqrt{2-\sqrt{2}}}{2}i\right)^2 = \dfrac{1}{\sqrt{2}} + \dfrac{1}{\sqrt{2}}i$ になるからだからだ。そして、さらにさらに $\dfrac{\sqrt{2+\sqrt{2}}}{2} + \dfrac{\sqrt{2-\sqrt{2}}}{2}i$ でさえ... ということが際限なく続くのである。でも、この行き着く先は意外な場所である。それは $1 + 0i$ である。

$+1 = (-1)^2$、$-1 = i^2$、$i = \left(\dfrac{1}{\sqrt{2}} + \dfrac{1}{\sqrt{2}}i\right)^2$、$\dfrac{1}{\sqrt{2}} + \dfrac{1}{\sqrt{2}}i = (\cdots)^2$、... というように、深く深く突き詰めた先が、もとの $+1$ 戻ってしまうなんて。複素数には不思議な関連があるものだ。

[1] version 2.x ではそうなるようだ。