Nombres complexes I

Nous allons faire des calculs avec les nombres complexes. Ce sera facile car `Python` *sait les manipuler.*

Cours 1 (Nombres complexes).
Avec `Python`, tu manipules les nombres complexes comme les autres nombres. La notation pour le nombre complexe i (qui vérifie $i^2 = -1$) est le symbole `j` (plus exactement `1j`). Par exemple, le nombre complexe $4-3i$ se note `4-3j`. Ensuite les opérations classiques s'écrivent comme d'habitude : par exemple le calcul $(1+2i)(4-i)$ s'écrit `(1+2j)*(4-1j)` et `Python` renvoie `6+7j`.

- Addition $z_1 + z_2$: `z1 + z2`
- Multiplication $z_1 \cdot z_2$: `z1 * z2`
- Puissance z^n : `z1 ** n`
- Inverse $\frac{1}{z}$: `1/z`
- Partie réelle a de $z = a + ib$: `z.real` (sans parenthèses)
- Partie imaginaire b de $z = a + ib$: `z.imag` (sans parenthèses)
- Module $|z| = \sqrt{a^2 + b^2}$: `abs(z)`
- Conjugué $\bar{z} = a - ib$: `z.conjugate()`

Bien sûr `Python` ne fait pas toujours des calculs exacts (par exemple, lors d'une division ou pour le calcul d'un module), car les nombres sont des nombres complexes flottants.

Activité 1 (Manipuler les nombres complexes).

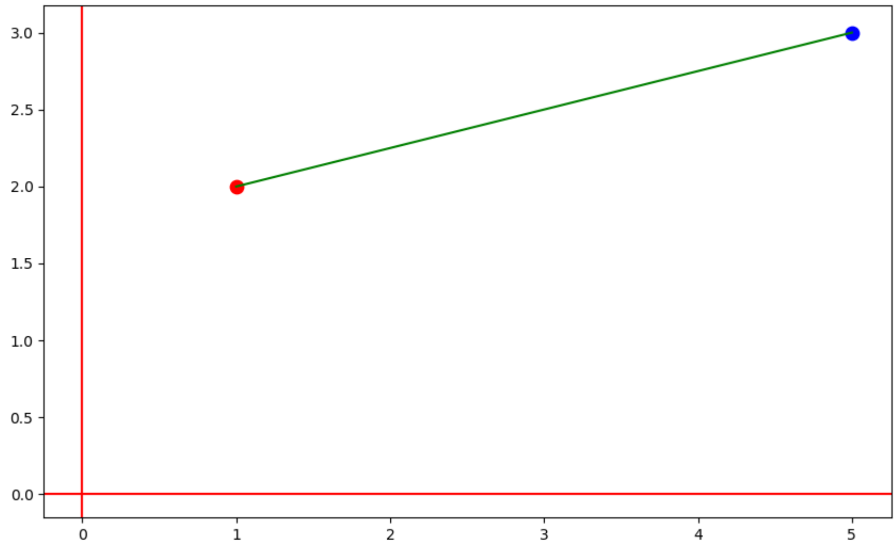
Objectifs : faire des calculs avec les nombres complexes.

1. Définis les nombres complexes $z_1 = 1 + 2i$ et $z_2 = 3 - i$. Demande à la machine de calculer :
$$z_1 + z_2 \qquad z_1 z_2 \qquad z_1^2 \qquad |z_1| \qquad \frac{1}{z_1}$$
2. Définis le nombre complexe $z = (3-4i)^2(2+i)$. Calcule à la machine la partie réelle de z, sa partie imaginaire et son conjugué.
3. Définis tes propres fonctions pour les opérations sur les nombres complexes. Représente le nombre complexe $z = a + ib$ par le couple de réels (a, b) et $z' = a' + ib'$ par le couple de réels (a', b') (tu n'as pas le droit d'utiliser les nombres complexes de `Python`).
   - Programme une fonction `addition(a,b,aa,bb)` qui renvoie le couple de réels correspondant au résultat de $(a + ib) + (a' + ib')$.
   - Programme une fonction `multiplication(a,b,aa,bb)` qui renvoie le couple de réels correspondant au résultat de $(a + ib) \times (a' + ib')$.

- Programme une fonction `conjugue(a,b)` qui renvoie le couple de réels correspondant au conjugué de $a+ib$.
- Programme une fonction `module(a,b)` qui renvoie le module de $a+ib$ (c'est un nombre réel).
- Programme une fonction `inverse(a,b)` qui pour $z=a+ib$ teste d'abord si z n'est pas nul et dans ce cas renvoie le couple de réels associé à l'inverse de z en utilisant une des formules :
$$\frac{1}{z}=\frac{\bar{z}}{|z|^2} \qquad \text{ou} \qquad \frac{1}{z}=\frac{a-ib}{a^2+b^2}.$$
- Programme une fonction `puissance(a,b,n)` qui pour $z=a+ib$ et $n\geqslant 0$, renvoie le couple de réels associé à z^n. (*Indications.* Par définition $z^0=1$. On pourra construire une boucle et utiliser une des fonctions précédentes.)

Cours 2 (Rappels Matplotlib).
Voici comment afficher des points de coordonnées (x,y) et un segment à l'aide du module `matplotlib`.

```
import matplotlib.pyplot as plt

plt.clf()  # Efface tout
plt.axhline(y=0, color='r', linestyle='-')  # Axe x
plt.axvline(x=0, color='r', linestyle='-')  # Axe y
plt.axes().set_aspect('equal')  # Repère orthonormé

x1 = 1
y1 = 2
plt.scatter(x1,y1,color='red',s=80)     # Un premier point

x2 = 5
y2 = 3
plt.scatter(x2,y2,color='blue',s=80)    # Un second point

# Un segment reliant les points
plt.plot([x1,x2],[y1,y2],color='green')

plt.show()  # Lancement de la fenêtre
```



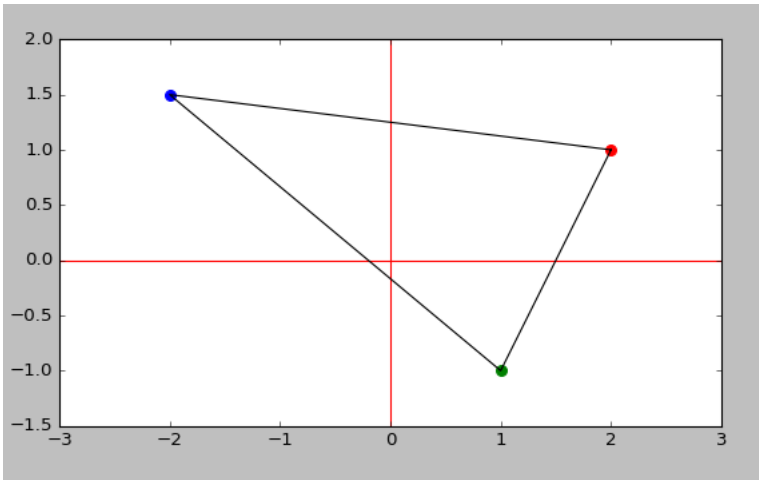
Activité 2 (Visualiser les nombres complexes)**.**

Objectifs : afficher un point connaissant son affixe z.

1. Dessine le point d'affixe 1 et celui d'affixe i. Pour un nombre complexe z, par exemple $z = -2+3i$, dessine le point d'affixe z.
2. Pour un nombre complexe z, par exemple $z = 3-2i$, dessine les points d'affixes :

$$z \qquad 2z \qquad iz \qquad \bar{z} \qquad \frac{z^2}{|z|} \qquad \frac{1}{z}$$

3.
   - Programme une fonction `affiche_triangle(z1,z2,z3)` qui trace le triangle dont les sommets ont pour affixes z_1, z_2, z_3.



   - On fixe $z \in \mathbb{C}$. Quelle semble être la nature du triangle déterminé par z, $2z$, $(1+2i)z$?
   - On pose $\omega = -\frac{1}{2} + \frac{\sqrt{3}}{2}i$. On fixe $z \in \mathbb{C}$. Quelle semble être la nature du triangle donné par z, ωz, $\omega^2 z$?

Activité 3 (Résolution d'une équation linéaire).

Objectifs : résoudre des équations linéaires en utilisant les nombres complexes. Ici nous allons « hacker » les nombres complexes. En informatique un « hack » est un détournement d'une fonctionnalité.

Résolution. Tu vas programmer une fonction `solution_equation_lineaire(equation)` qui calcule et renvoie la solution d'une équation linéaire. L'équation est donnée en paramètre sous la forme d'une chaîne de caractères. Par exemple avec `"7*x+3 = 0"`, la fonction renvoie `-0.42857...` comme valeur approchée de $-\frac{3}{7}$. Autre exemple avec : `"3*(x+1) + x = 2*x+1"`, la fonction renvoie `-1.0`. Attention : il faut explicitement écrire les multiplications avec le caractère « `*` ».

Astuce. L'idée est la suivante, une équation linéaire se ramène à la forme :

$$a + bx = 0$$

dont une solution est $-\frac{a}{b}$. L'astuce est de remplacer chaque « x » de l'équation par le nombre complexe i. On obtient ainsi un nombre complexe $a + ib$. En extrayant la partie réelle et la partie imaginaire, on renvoie la solution (réelle) $-a/b$.

Exemple simple. Partant de l'équation $7x+3=0$, on ne garde que la partie gauche de l'équation $7x+3$. On remplace la lettre x par le nombre complexe i, on obtient $7i+3$. On extrait sa partie réelle $a=3$ et sa partie imaginaire $b=7$. On renvoie la solution $-a/b=-\frac{3}{7}$.

Exemple compliqué. Soit l'équation $3(x+1)+x=2x+1$. Il faut d'abord tout basculer à gauche du signe égal afin de se ramener à l'équation $3(x+1)+x-\big(2x+1\big)=0$. On ne garde que l'expression à gauche du signe égal : $3(x+1)+x-\big(2x+1\big)$. On remplace ensuite la variable x par le nombre complexe i. L'expression devient un nombre complexe $3(i+1)+i-\big(2i+1\big)$. On note z ce nombre complexe, on calcule sa partie réelle $a=2$ et sa partie imaginaire $b=2$. On calcule $-a/b=-1$. La solution de notre équation est donc $x=-1$.

Algorithme.

- — Entrée : une chaîne de caractères représentant une équation linéaire en x.
  — Sortie : la valeur numérique de la solution x.
- Soient `G` et `D` les deux chaînes de part et d'autre du signe « = ». (Utilise `equation.split("=")`.)
- Former la chaîne correspondant à la partie gauche moins la partie droite : `G + "-(" + D + ")"`.
- Pour remplacer x par i, il faut remplacer le caractère `"x"` par la chaîne `"1j"`. (Utilise `chaine.replace(mot,nouv_mot)`.) On obtient ainsi une chaîne `z_str`.
- Transformer la chaîne en un nombre complexe par l'opération :
  `z = eval(z_str)`
- Calculer la partie réelle `a` et la partie imaginaire `b` de `z`.
- Renvoyer `-a/b`.

Cours 3 (Équation du second degré).
Soit $a,b,c\in\mathbb{R}$ avec $a\neq 0$. On considère l'équation

$$az^2+bz+c=0.$$

On note $\Delta=b^2-4ac$. Selon le signe de Δ les solutions sont les suivantes :

- $\Delta > 0$: deux solutions réelles $z_1 = \dfrac{-b+\sqrt{\Delta}}{2a}$ et $z_2 = \dfrac{-b-\sqrt{\Delta}}{2a}$.
- $\Delta = 0$: solution double $z_0 = \dfrac{-b}{2a}$.
- $\Delta < 0$: deux solutions complexes $z_1 = \dfrac{-b+i\sqrt{|\Delta|}}{2a}$ et $z_2 = \dfrac{-b-i\sqrt{|\Delta|}}{2a}$.

Activité 4 (Équation du second degré)**.**

Objectifs : résoudre les équations du second degré, y compris lorsque le discriminant est négatif.

On considère l'équation :

$$az^2 + bz + c = 0 \qquad \text{avec} \quad a, b, c \in \mathbb{R} \quad \text{et} \quad a \neq 0.$$

1. **Équation du second degré.** Programme une fonction `solution_trinome(a,b,c)` qui renvoie les solutions de l'équation sous la forme d'une liste de deux nombres `[z1,z2]` (si la solution est double, renvoie la liste `[z0,z0]`).

   Exemple. Calcule les solutions de :

   $$z^2 - 2z + 1 = 0 \qquad z^2 + z - 1 = 0 \qquad z^2 + z + 1 = 0$$

2. **Somme et produit.** Lorsque l'on connaît la somme S et le produit P de deux nombres z_1 et z_2, on peut retrouver z_1 et z_2. Comment faire ? Réponse : z_1 et z_2 sont les solutions de l'équation :

   $$z^2 - Sz + P = 0.$$

   Programme une fonction `solution_somme_produit(S,P)` qui renvoie la liste `[z1,z2]` des solutions.

   Exemple. Trouve deux nombres dont la somme est 10 et le produit est 20.

3. **Équation bicarrée.** Une équation bicarrée est de la forme :

   $$ax^4 + bx^2 + c = 0.$$

   Nous étudions seulement les cas où $\Delta = b^2 - 4ac \geqslant 0$, pour lesquels l'équation admet 4 solutions (réelles ou complexes).

   On commence par poser $X = x^2$ et résoudre l'équation du second degré :

   $$aX^2 + bX + c = 0.$$

   Cette dernière équation admet deux solutions réelles X_1 et X_2. Pour chacune de ces solutions X :
   - si $X \geqslant 0$, on obtient deux solutions, $+\sqrt{X}$ et $-\sqrt{X}$;
   - si $X < 0$, on obtient deux solutions, $+i\sqrt{-X}$ et $-i\sqrt{-X}$.

   On obtient ainsi 4 solutions (2 associées à X_1 et 2 associées à X_2).

   Programme une fonction `solution_bicarre(a,b,c)` qui renvoie les 4 solutions de l'équation $ax^4 + bx^2 + c = 0$ après avoir vérifié que $\Delta = b^2 - 4ac \geqslant 0$.

   Exemple. Trouve les 4 solutions de l'équation $x^4 - 2x^2 - 3 = 0$.

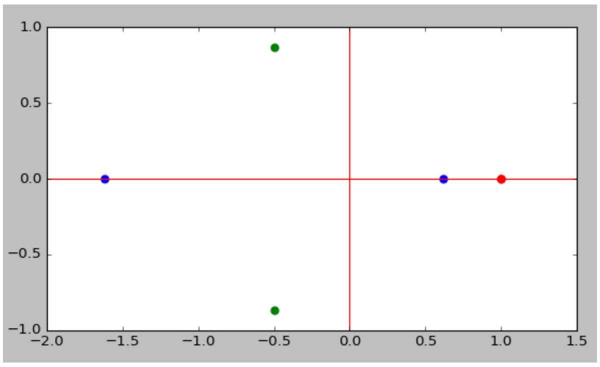
Activité 5 (Famille de racines)**.**

Objectifs : afficher les solutions d'une famille d'équations du second degré.

1. **Afficher les racines.** Programme une fonction `affiche_racines(a,b,c)` qui affiche les deux points correspondant aux deux solutions de l'équation $ax^2 + bx + c = 0$.

Amélioration. C'est mieux d'autoriser en argument optionnel le choix de la couleur du point par une entête `affiche_racines(a,b,c,couleur='red')`.

Retrouve sur la figure ci-dessous les racines des polynômes $x^2-2x+1=0$, $x^2+x-1=0$ et $x^2+x+1=0$.



2. **Famille de racines.** On considère deux polynômes

$$P_0(x) = x^2 + b_0x + c_0 \quad \text{avec} \quad \Delta_0 = b_0^2 - 4c_0 \leqslant 0,$$
$$P_1(x) = x^2 + b_1x + c_1 \quad \text{avec} \quad \Delta_1 = b_1^2 - 4c_1 \leqslant 0.$$

On définit pour $0 \leqslant t \leqslant 1$:

$$P_t(x) = (1-t)P_0(x) + tP_1(x) = x^2 + \big((1-t)b_0 + tb_1\big)x + (1-t)c_0 + tc_1.$$

Question. Quelle forme a l'ensemble des racines de la famille $\{P_t(x)\}_{0\leqslant t\leqslant 1}$?

Programme une fonction `affiche_famille(b0,c0,b1,c1)` :
- qui affiche les solutions de $P_0(x) = 0$ (en rouge par exemple),
- qui affiche les solutions de $P_1(x) = 0$ (en vert par exemple),
- qui affiche les solutions de $P_t(x) = 0$ pour n valeurs t, avec $0 \leqslant t \leqslant 1$ (en bleu par exemple).

Pour les valeurs de t, tu peux les choisir de la forme k/n avec $0 \leqslant k < n$.

Amélioration. C'est mieux d'autoriser que n soit un argument optionnel par une entête du type `affiche_famille(b0,c0,b1,c1,n=100)`.

Exemple ci-dessous $P_0(x) = x^2 - 2x + 2$ *et* $P_1(x) = x^2 + 3x + \frac{12}{5}$, *avec seulement* $n = 3$ *points bleus intermédiaires. Pour répondre à la question il faut afficher plus de points intermédiaires.*