Travaux pratiques d'introduction 1

Affectations, variables

Informatique tronc commun MPSI et PCSI

I Présentation

I.1 Le shell (ligne de commande interactive)

Python étant un langage interprété, il est possible de faire exécuter les lignes de code au fur et à mesure qu'elles sont saisies. Une des applications de ce mode de fonctionnement est d'utiliser Python comme une calculatrice, mais avec le confort d'un ordinateur.
La figure 1 montre l'interface de l'IDE Thonny (des différences mineures peuvent exister, selon les versions et la configuration choisie).

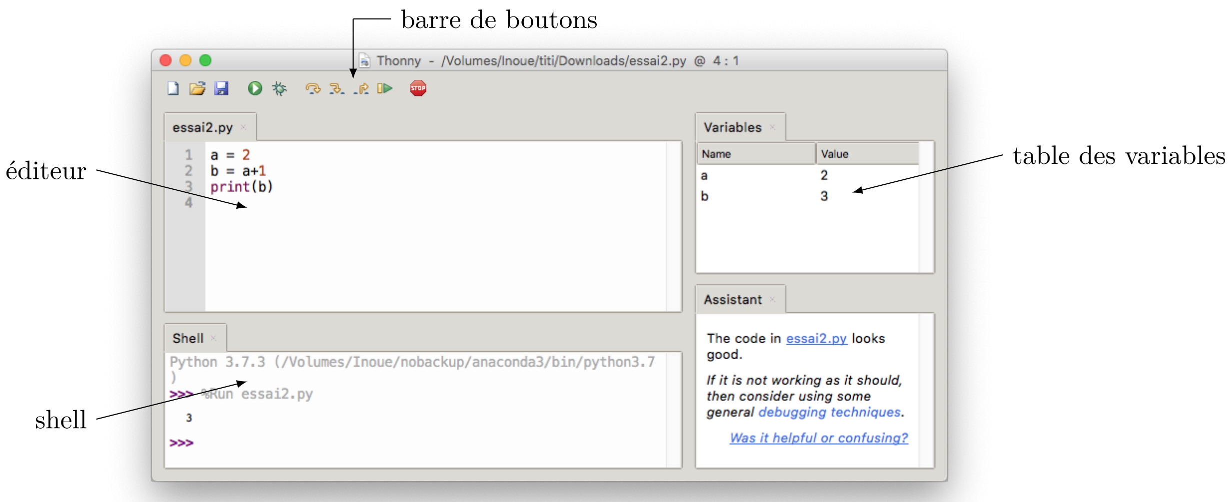

FIG. 1 : Interface l'IDE appelé Thonny.

Dans le TP d'aujourd'hui, l'éditeur ne sera pas utilisé et vous taperez votre code uniquement dans la *ligne de commande interactive*, ou *shell*. Contrairement à l'éditeur, le shell exécute immédiatement ce que vous venez de taper quand vous appuyez sur la touche Entrée.
Le shell permet toujours de distinguer visuellement les commandes que vous tapez et le résultat renvoyé. Deux interfaces équivalentes que l'on rencontre souvent sont :

- Figure 2a : les lignes que vous tapez sont précédées de chevrons `>>>`, tandis que les lignes renvoyées non.
- Figure 2b : les lignes que vous tapez sont précédées de `In[]` et les lignes renvoyées de `Out[]`. Un numéro incrémenté automatiquement est placé entre crochets pour repérer quel `Out` va avec quel `In`.

(a) (b)

Fig. 2 : Deux interfaces classiques pour le shell.

À retenir

Quand vous changez d'exercice, il peut être préférable de fermer le shell et d'en ouvrir un nouveau afin de purger la mémoire de toutes les variables utilisées précédemment. Dans Thonny, cette opération se fait via le bouton ressemblant à un panneau Stop dans la barre de boutons (voir figure 1).

I.2 Affecter une valeur à une variable

Affecter une valeur à une variable se fait selon la syntaxe :

```
nom = expression
```

L'expression est d'abord évaluée, puis sa valeur affectée à la variable dont le nom est à gauche. *Si besoin, la variable est automatiquement créée.* Pas de déclaration préalable en Python. De même, le type de la variable est déduit automatiquement de la valeur de l'expression.

À retenir

Un nom de variable ne doit contenir que des lettres de l'alphabet latin (majuscules ou minuscules), des chiffres ou le caractère underscore _ sachant que le premier caractère doit être une lettre.

Efforcez-vous d'utiliser des noms de variables explicites (`quotient` au lieu de `q` par exemple), de toute façon vous n'aurez pas besoin de taper les noms en entier : la touche Tabulation du clavier permettra d'autocompléter les noms des variables quand vous en aurez saisi les premiers caractères.

I.3 Obtenir la valeur d'une variable

Dans le shell, taper le nom d'une variable puis valider avec Entrée provoque l'affichage de sa valeur. Mais il peut être plus utile à long terme de prendre l'habitude de surveiller les valeurs des variables via la *table des variables*.
Celle-ci est affichée automatiquement dans l'IDE Thonny (voir figure 1), et il existe toujours un moyen de l'afficher dans les autres logiciels.

I.4 Les bibliothèques `math` et `cmath`

Afin de limiter l'encombrement mémoire, Python ne propose pas la totalité des commandes existantes à tout moment. La majorité des commandes sont regroupées dans des bibliothèques qu'il faut charger manuellement pour pouvoir les utiliser.
Dans le TP d'aujourd'hui, vous pourrez avoir besoin de la bibliothèque `math`, qui fournit un certain nombre de fonctions mathématiques usuelles (fonctions trigonométriques, exponentielle...) et quelques constantes (π et e).
Pour la charger, tapez et exécutez la ligne suivante :

```
import math as m
```

Cela va créer une variable `m` qui servira d'alias (nom abrégé) à la bibliothèque. Cet alias peut être choisi librement, pourvu qu'il respecte les règles ci-dessus pour le nommage. Si vous fermez le shell, cette variable est détruite comme les autres et il faudra réimporter `math`.
Les fonctions et variables mises à disposition par la bibliothèque doivent alors être préfixées par cet alias. Par exemple :

```
m.pi
m.cos(m.pi/4)
```

Vous aurez également besoin de la bibliothèque `cmath`, qui propose essentiellement les mêmes fonctionnalités mais avec la capacité à mener des calculs avec les nombres complexes.

À retenir

La nécessité de préfixer les noms des fonctions par l'alias de la bibliothèque permet justement d'éviter les ambiguïtés quand deux bibliothèques fournissent des fonctions portant le même nom, par exemple `math.cos` et `cmath.cos`.

Il vaut mieux ne charger `cmath` que quand on en a besoin, car elle est moins performante que `math` pour les calculs sur les nombres réels.

II Opérations simples

Exercice 1 - Deviner le résultat final
Prévoyez la valeur de `s` à la fin de ce code, puis vérifiez votre prévision en tapant ces commandes dans le shell.

```
x = 10
y = 15
z = x+y
x = y
y = z
s = x+y+z
```

En cas d'erreur, regardez dans la table des variables les valeurs finales de `x`, `y` et `z` pour localiser votre erreur.

Exercice 2 - Cherchez l'erreur
Que voulait-on obtenir avec le code suivant ? Modifiez-le pour obtenir le résultat voulu.

```
x = 10
y = 15
y = x
x = y
```

Là aussi, gardez un œil sur la table des variables à la fin de l'exécution pour vérifier que vous avez correctement anticipé ce qui se passe.

III Applications numériques physiques

La connaissance des lois physico-chimiques mentionnées dans ces exercices n'est pas nécessaire. Vous devez seulement mener les applications numériques demandées avec les contraintes suivantes :

- utiliser le shell Python,
- créer une ou plusieurs variables afin de faciliter la lecture et la détection des erreurs.

Exercice 3 - Deuxième équivalence
On dose l'acide oxalique (diacide de concentration C et de volume initial $V_0 = 20 \times 10^{-3}\,\mathrm{L}$) par de la soude de concentration $C_b = 5 \times 10^{-2}\,\mathrm{mol \cdot L^{-1}}$. La première équivalence est peu visible, donc on ne peut exploiter que la seconde, pour laquelle :

$$2CV_0 = C_b V_{\text{éq2}} \quad \text{avec} \quad V_{\text{éq}} = 12 \times 10^{-3}\,\mathrm{L}$$

Calculez numériquement C.

sachant que :

- L'ordinateur ne gère pas les unités, donc rentrez les valeurs numériques directement dans les bonnes unités (ici celles fournies conviennent déjà).
- La multiplication implicite (par exemple ab pour dire $a \times b$) n'est pas comprise par Python, donc utilisez forcément le symbole `*` pour les multiplications.
- Python permet une notation abrégée pour les nombres en notation scientifique. Par exemple, $4{,}21 \times 10^{-11}$ peut être saisi `4.21e-11`. L'exposant doit être entier, et il ne doit y avoir que des valeurs numériques explicites dans l'expression.

Exercice 4 - Pulsation plasma
À quelques dizaines de kilomètres d'altitude se trouve une couche d'atmosphère appelée *ionosphère*. Celle-ci est dans un état de la matière appelée *plasma* et ne laisse passer les ondes télécom que si leur fréquence est assez élevée. On donne :

$$\begin{cases} \nu = \dfrac{\omega}{2\pi} & \text{fréquence} \\ k^2 = \dfrac{\omega^2 - \omega_p^2}{c^2} & \text{nombre d'onde} \\ \omega_p = \sqrt{\dfrac{ne^2}{\varepsilon_0 m_e}} & \text{pulsation plasma} \end{cases} \quad \text{avec} \quad \begin{cases} e = 1{,}60 \times 10^{-19}\,\mathrm{C} \\ \varepsilon_0 = 8{,}85 \times 10^{-12}\,\mathrm{F \cdot m^{-1}} \\ m_e = 9{,}11 \times 10^{-31}\,\mathrm{kg} \\ n = 1{,}00 \times 10^{11}\,\mathrm{m^{-3}} \\ c = 3{,}00 \times 10^{8}\,\mathrm{m \cdot s^{-1}} \end{cases}$$

Calculez la fréquence minimale avec laquelle communiquer depuis la surface avec un satellite. Pour cela, on signale que le nombre d'onde doit être à valeur réelle, et donc de carré positif.

sachant que :

- x (réel) à la puissance n (réel) s'écrit `x**n`,
- donc (cas particulier) la racine carrée de `x` s'écrit `x**0.5`.
- Importez la bibliothèque `math` (voir paragraphe I.4) pour disposer de la valeur de π. Ne tapez pas une valeur (trop) approximative à la main !

Exercice 5 - Méthode de Bessel
Dans un home cinema, on veut projeter l'image d'un objet (le film) sur un écran (au mur) par une lentille. La distance objet-écran est imposée par la disposition de la salle et vaut $D = 5\,\mathrm{m}$. La lentille de projection a pour distance focale $f' = 0{,}4\,\mathrm{m}$.
Il existe deux positions de la lentille entre l'objet et l'écran telles que l'image est nette. Le grandissement (rapport entre la taille de l'objet et celle de l'image) vaut :

$$\gamma_1 = \frac{D + \sqrt{D^2 - 4Df'}}{-D + \sqrt{D^2 - 4Df'}} \quad \text{et} \quad \gamma_2 = \frac{D - \sqrt{D^2 - 4Df'}}{-D - \sqrt{D^2 - 4Df'}} \tag{1}$$

Calculez les deux grandissements. Vérification : leur produit doit valoir 1.

Exercice 6 - Résonance en transmission
Un électron d'énergie E arrive sur une zone de largeur l où son énergie potentielle vaut $V_0 < E$. D'après les lois de la physique quantique, la probabilité qu'il reparte en arrière (réflexion) est :

$$R = \frac{(k_2^2 - k_1^2)^2 \sin(k_2 l)^2}{4k_1^2 k_2^2 + (k_1^2 - k_2^2)^2 \sin(k_2 l)^2}$$

$$\text{avec} \quad \begin{cases} k_1 = \sqrt{\dfrac{2m_e E}{\hbar^2}} \\ k_2 = \sqrt{\dfrac{2m_e(E - V_0)}{\hbar^2}} \end{cases} \quad \text{avec} \quad \begin{cases} m_e = 9{,}11 \times 10^{-31}\,\mathrm{kg} \\ E = 5{,}00\,\mathrm{eV} \\ V_0 = 3{,}00\,\mathrm{eV} \\ l = 5{,}00 \times 10^{-9}\,\mathrm{m} \\ \hbar = 1{,}05 \times 10^{-34}\,\mathrm{J \cdot s} \end{cases}$$

Calculez cette probabilité.

sachant que :

- Dans k_1 et k_2, il faudra exprimer les énergies E et V_0 en joules. On donne le facteur de conversion des électron-volt aux joules : $1\,\mathrm{eV} = 1{,}60 \times 10^{-19}\,\mathrm{J}$.
- La fonction sinus est fournie par la bibliothèque `math` et s'appelle `sin`.

Exercice 7 - Angle de perte d'une bobine
Une bobine est un composant électronique caractérisé par deux coefficients réels positifs, son inductance L et sa résistance R. On peut alors définir son impédance, paramètre complexe défini par $Z = R + iL\omega$, avec ω la pulsation du signal électrique dans le montage.
On définit alors son *angle de perte* δ par :

$$\tan(\delta) = \frac{\mathrm{Re}(Z)}{\mathrm{Im}(Z)} \quad \text{avec} \quad \begin{cases} L = 0{,}1\,\mathrm{H} \\ R = 10\,\Omega \\ \nu = 50\,\mathrm{Hz} \\ \omega = 2\pi\nu \\ \delta \in \left[0, \dfrac{\pi}{2}\right[\end{cases}$$

Calculez l'angle de pertes en degrés. Vérifiez que c'est le complémentaire de $\arg(Z)$.

sachant que :

- Les fonctions tangente et arc tangente sont fournies par la bibliothèque `math`, sous les noms `tan` et `atan` respectivement.

- Il faut charger la bibliothèque `cmath` pour pouvoir gérer les nombres complexes. Ajoutez `import cmath as cm` au début de votre fichier. Vous disposez ensuite des commandes suivantes :
  - Le nombre complexe i s'écrit `1j` et peut ensuite être utilisé avec les opérations usuelles
  - $\mathrm{Re}(Z)$ s'écrit `Z.real` (c'est bien le nom de la variable à gauche du point)
  - $\mathrm{Im}(Z)$ s'écrit `Z.imag` (idem)
  - $\arg(Z)$ s'écrit `cm.phase(Z)` (résultat en radians)

Exercice 8 - Conversion de durée
Que vaut une durée de 325 415 s convertie en jours-heures-minutes-secondes ?

sachant que Python permet de mener la division euclidienne d'un entier `a` par un entier `b` :

- `a//b` renvoie le quotient (valeur entière),
- `a%b` renvoie le reste (valeur entière).

IV Calculs sur des polynômes

À retenir

Ne manipulez jamais les valeurs des coefficients directement, stockez-les dans des variables pour pouvoir relire plus facilement votre code.
Vous devriez d'ailleurs faire de même avec vos calculatrices !

Exercice 9 - Racines d'un trinôme
Calculez les racines du polynôme $aX^2 + bX + c$ avec $a = 1$, $b = -4$ et $c = 3$.

Les exercices suivants portent sur la recherche des racines d'un polynôme de degré 3 de la forme $P(X) = aX^3 + bX^2 + cX + d$ avec $a \neq 0$. Même dans le cas où tous les coefficients sont réels et où les racines sont réelles, certains calculs intermédiaires sont inévitablement intervenir des nombres complexes. La méthode est la suivante.
On associé à P deux autres polynômes :

- Le premier est un autre polynôme de degré 3 où le terme de degré 2 a été éliminé par changement de variable :

  $$Q(Y) = Y^3 + pY + q \quad \text{avec} \quad \begin{cases} Y = X + \dfrac{b}{2a} \\ p = -\dfrac{b^2}{3a^2} + \dfrac{c}{a} \\ q = \dfrac{b}{27a}\left(\dfrac{2b^2}{a^2} - \dfrac{9c}{a}\right) \end{cases}$$

  Les racines de Q et celles de P se déduisent les unes des autres par une simple translation.

  Remarque : l'expression de Q n'est donnée qu'à titre informatif, elle ne sera pas utile dans ce TP.

- Le second est un trinôme :

  $$R(Z) = Z^2 + qZ - \frac{p^3}{27}$$

  Si y est une racine de Q, on la décompose en $y = u + v$ tel que :

– u^3 et v^3 sont les deux racines de R.

– Le produit uv vaut $-p/3$.

L'exemple choisi est $P(X) = 6X^3 - 6X^2 + 12X + 7$. Vous admettrez que ce polynôme a une racine réelle et deux racines complexes conjuguées.

Exercice 10 - Détermination de u^3
Déterminez u^3. Pourquoi ne chercher qu'une des deux racines de R ?

Exercice 11 - Recherche de la racine réelle de P
À partir du résultat précédent, déterminez la valeur réelle de u, notée u_1, la valeur de v associée, notée v_1, puis la racine réelle de P, notée x_1.

Exercice 12 - Recherche des deux racines complexes de P
Chargez la bibliothèque `cmath` pour pouvoir calculer les deux valeurs complexes possibles de u, notées u_2 et u_3. Vérifiez qu'elles sont conjuguées, et déduisez-en les deux racines complexes de P (qui doivent être conjuguées aussi), notées x_2 et x_3.

sachant que :

- On remarque que les n nombres complexes de la forme $e^{i2\pi\frac{k}{n}}$, avec k et n entiers naturels tels que $k \in [\![0, n-1]\!]$, prennent la valeur 1 quand on les élève à la puissance n.
- Pour générer une valeur complexe de module r et d'argument φ donnés, la bibliothèque `cmath` fournit la fonction `rect` : `cm.rect(r, phi)`.

V Solutions

Solution de l'exercice 1 - Deviner le résultat final
`s` vaut 65, car `x`, `y` et `z` valent respectivement 15, 25 et 25 à la fin.

Solution de l'exercice 2 - Cherchez l'erreur
On voulait permuter les valeurs de `x` et `y`, mais la 3$^{\text{e}}$ ligne détruit la valeur de `y` avant qu'elle ait été enregistrée dans `x`.
Une manière « générique » de procéder est de transiter par une variable tampon :

```
tampon = y
y = x
x = tampon
```

Une manière plus spécifique à Python peut se faire en une ligne :

```
x, y = y, x
```

Ce code fonctionne car le membre de droite est entièrement évalué (`x` et `y` remplacées par leurs valeurs) avant l'affectation.

Solution de l'exercice 3 - Deuxième équivalence

```
V0 = 20e-3
Veq2 = 12e-3
Cb = 5e-2
C = Cb*Veq2/(2*V0)
```

ce qui donne $C = 1{,}5 \times 10^{-2}\,\mathrm{mol \cdot L^{-1}}$.

Solution de l'exercice 4 - Pulsation plasma
On veut $k^2 \geqslant 0$ et donc $\omega_{\min} = \omega_p$. D'où :

```
import math as m
e = 1.6e-19
eps0 = 8.85e-12
me = 9.11e-31
n = 1e11
c = 3e8
wp2 = n*e**2/(eps0*me)
fmin = wp2**0.5/(2*m.pi)
```

ce qui donne $\nu = 2{,}83 \times 10^6\,\mathrm{Hz}$.

Solution de l'exercice 5 - Méthode de Bessel
Le but est de minimiser les répétitions du même calcul (la racine du discriminant) et de prendre l'initiative de créer des variables intermédiaires autres que celle du problème physique initial.

```
D = 5
fp = 0.4
sq_delta = (D**2-4*D*fp)**0.5
gamma1 = (D+sq_delta)/(-D+sq_delta)
gamma2 = (D-sq_delta)/(-D-sq_delta)
produit = gamma1*gamma2
```

On trouve $\gamma_1 = -10{,}4$ et $\gamma_2 = -0{,}09$.

Solution de l'exercice 6 - Résonance en transmission
Pour certains élèves, il y aura peut-être aussi des soucis liés aux règles de priorité des opérations !

```
import math as m
me = 9.11e-31
hbar = 1.05e-34
l = 5e-9
conv_eV = 1.6e-19
E = 5*conv_eV
V0 = 3*conv_eV
k12 = 2*me*E/hbar**2
k22 = 2*me*(E-V0)/hbar**2
a = (k22-k12)**2*m.sin(k22**0.5*l)**2
R = a/(4*k12*k22+a)
```

On trouve $R = 17{,}6\,\%$.

Solution de l'exercice 7 - Angle de perte d'une bobine

```
import math as m
import cmath as cm
L = 0.1
R = 10
nu = 50
omega = 2*m.pi*nu
Z = R+1j*L*omega
tand = Z.real/Z.imag
delta_rad = m.atan(tand)
delta_deg = delta_rad*180/m.pi
```

On trouve $\delta = 17{,}7°$, qui est bien le complémentaire de $\arg(Z)$.

Solution de l'exercice 8 - Conversion de durée
On peut s'économiser l'utilisation de quelques variables en recyclant celle stockant la durée initiale.

```
t = 325415
jour_secondes = 3600*24
jours = t//jour_secondes
t = t%jour_secondes
heures = t//3600
t = t%3600
minutes = t//60
secondes = t%60
```

On trouve finalement 3 j 18 h 23 min 35 s.

Solution de l'exercice 9 - Racines d'un trinôme

```
a = 1
b = -4
c = 3
delta = b**2-4*a*c
r1 = (-b-delta**0.5)/(2*a)
r2 = (-b+delta**0.5)/(2*a)
```

On trouve $r1 = 3{,}0$ et $r2 = 1{,}0$. C'est l'occasion d'attirer l'attention sur la conversion inévitable en flottants, et les éventuelles erreurs numériques qui peuvent en résulter.

Solution de l'exercice 10 - Détermination de u^3

```
a = 6
b = -6
c = 12
d = 7
p = -b**2/(3*a**2)+c/a
q = b*(2*b**2/a**2-9*c/a)/(27*a)+d/a
delta = q**2-4*(-p**3/27)
ucube = (-q+np.sqrt(delta))/2
```

On trouve $u^3 = 0{,}092\,59$. Par la suite on pourra en déduire u, puis v puisque le produit uv est connu, ce qui est plus efficace que de calculer v^3 directement.

Solution de l'exercice 11 - Recherche de la racine réelle de P

```
u1 = ucube**(1/3)
v1 = -p/(3*u1)
y1 = u1+v1
x1 = y1-b/(3*a)
```

On trouve $x_1 = -0{,}442\,27$.

Solution de l'exercice 12 - Recherche des deux racines complexes de P
D'après l'énoncé, $u_2 = e^{\frac{2i\pi}{3}} u_1$ et $u_3 = e^{\frac{2i\pi}{3}} u_2$.

```
w = cm.rect(1, 2*np.pi/3)
u2 = u1*w
v2 = -p/(3*u2)
y2 = u2+v2
x2 = y2-b/(3*a)

u3 = u2*w
v3 = -p/(3*u3)
y3 = u3+v3
x3 = y3-b/(3*a)
```

On trouve $x_2 = 0,72113 - 1,45528i$ et $x_3 = 0,72113 + 1,45528i$.