

Info. TP8: Titrage et grille

- <https://learn.heeere.com/2018-infospichi-8a82>
- http://matplotlib.org/api/pyplot_api.html (doc pyplot)

Rendu : sous forme d'archive

À la fin, vos réponses seront rendues dans une archive zip : cette archive contiendra un fichier `compte-rendu.txt` (avec votre **nom**, **groupe**, et les réponses aux questions précédées d'une apostrophe) et vos **fichiers pythons**.

Pour faire une **archive zip** de votre TP pour le rendu, exécutez, dans le terminal :

```
cd ~/info-spichi/  
zip -r tp8.zip tp8/
```

Important : mise en place

- Suivre les consignes des TP précédents, et travailler dans un dossier `tp8`.

Exercice 1 – Titrage

Rappel : pour cet exercice, travaillez dans un dossier `ex1` lui même dans le dossier `tp8`. On veut écrire un programme permettant de calculer une concentration en acide ou base à partir d'une courbe de titrage (courbe pH en fonction du volume de solution titrante versée, réalisée lors d'un dosage)

Pour ne pas avoir à faire des dosages pour tester notre programme on utilisera des données simulées. Téléchargez le fichier `titrage.py`:

```
wget https://learn.heeere.com/2018-infospichi-8a82/raw/titrage.py
```

`titrage.py` contient la fonction `generer_titrage` qui retourne un tableau contenant le volume total dans le bécher à chaque ajout de solution titrante, et un tableau contenant les valeurs de pH correspondantes. On considère qu'on commence toujours avec 100mL de solution à titrer et on ajoute toujours 1mL de solution titrante.

Q1) Générez des valeurs correspondant à un titrage d'un acide de pH 2.3 par une base de pH 11.5 avec un volume d'équivalence de 50mL.

Q2) Écrivez une fonction `tracer_courbe` qui prend un tableau des volumes et un tableau de pH et trace la courbe de titrage grâce au module `pyplot`. Utilisez cette fonction pour affichez les données simulées précédemment.

Le but d'un titrage est de trouver le volume d'équivalence où le nombre de mol d'acide (ou base) dans la solution est égal au nombre de mol de base (ou acide) ajouté à la solution. Il existe 2 moyens de trouver ce volume: trouver le point dont la dérivée est maximale ou utiliser la méthode des tangentes.

Q3) Écrivez la fonction `derivee` qui prend en arguments les valeurs de pH et les volumes et retourne la dérivée

Aide: la dérivée d'une courbe peut être approximée par $f'(x) = \frac{f(x+h)-f(x)}{h}$ ou pour tout x_1 : $f'(x_1) = \frac{f(x_2)-f(x_1)}{x_2-x_1}$ avec $x_1 < x_2$

Aide: `np.diff(a)` renvoie un tableau contenant les différences des paires successives de valeurs dans `a`.

Exemple: `np.diff(np.array([28, 40, 36, 16, 23]))` renvoie `array([12, -4, -20, 7])`

Q4) Ajoutez un argument à `tracer_courbe` qui contiendra le volume d'équivalence. Par défaut la valeur sera -1, si on donne une valeur ≥ 0 alors on affichera le volume d'équivalence sur la courbe. N'oubliez pas que le volume d'équivalence est le volume de solution titrante ajouté.

Aide: Utilisez la fonction `plt.annotate`. pour ajouter une flèche vous pouvez utiliser l'argument `arrowprops` avec par exemple la valeur: `dict(facecolor='black', shrink=0.05, headwidth=8, width=2)`.

Q5) Grâce à la méthode de la dérivée, affichez le volume d'équivalence dans vos données simulées précédemment.

Rappel: On pourra utiliser la fonction `np.argmax`.

Q6) Écrivez la fonction `concentration_titre` qui prend le volume d'équivalence et la concentration de la solution titrante et retourne la concentration de la solution à titrer. Avec les données simulées, affichez la concentration d'acide sachant que la concentration de base est de 0.5mol/L.

Q7) Écrivez la fonction `tangente` qui à partir d'un volume V , d'un tableau de volume, et d'un tableau de pH, renvoie la pente et l'ordonnée à l'origine de la tangente à la courbe (volume, pH) au point de volume V .

Aide: Équation d'une tangente à une courbe: $y = a * x + b$, où a est la dérivée de la courbe au point de croisement avec la tangente.

Q8) Ajoutez 2 arguments à `tracer_courbe`, chacun attend un tuple contenant la pente et l'ordonnée à l'origine de chaque tangente. Par défaut ils contiendront None. Si un tuple est donné alors on tracera la tangente correspondante. Si chaque argument reçoit un tuple, alors, tracez en plus la droite passant entre les 2 tangentes.

Par exemple: `tracer_courbe(volumes, ph, 50, (1, -3))` affichera la courbe ph en fonction des volumes, le volume d'équivalence à $V=50\text{mL}$, et la tangente à la courbe ayant une pente de 1 et une ordonnée à l'origine de -3. Il n'affichera pas la 2ème tangente et donc pas la droite non plus.

Exercice 2 – Grille

Téléchargez le fichier grille.py.

```
wget https://learn.heeere.com/2018-infospichi-8a82/raw/grille.py
```

Q9) Lancez `grille.py`. Que se passerait-il si on appelait `recharger_grille_au_hasard(grille)` avant `grille = grille_au_hasard((H, W))` ?

Q10) Créez une figure de taille $(H \times S, W \times S)$, dans cette figure ajoutez un unique "subplot" à la figure, vous sauvez la valeur retournée par `add_subplot` dans la variable `ax_grille` (objet de type `pyplot.Axes`).

Q11) Créez la fonction `afficher_grille` où vous afficherez l'objet `grille` grâce à `ax_grille.imshow`. Ensuite, pour mettre à jour la figure, appelez `fig.canvas.draw()`.

```
def comportement_touche(event):  
    print("Touche", event.key)
```

```
fig.canvas.mpl_connect('key_press_event', comportement_touche)
```

Ce code python ajoute un “listener” à la figure, celui-ci permet d’afficher les touches du clavier qui sont appuyées.

Q12) Complétez `comportement_touche` de sorte que si la touche “a” est pressée alors on recharge une grille au hasard.

Q13) Complétez `comportement_touche` de sorte que si la touche “z” est pressée alors on met tous les éléments de la grille à 0.

Q14) Complétez `comportement_touche` de sorte que si la touche “e” est pressée alors on met tous les éléments dans la moitié basse de la grille à 12. Qu’est ce que vous observez ? Pourquoi ?

Q15) Complétez `comportement_touche` de sorte que si la touche “r” est pressée alors on change toutes les colonnes de la ligne centrale sauf la première et la dernière de sorte qu’elles contiennent des nombres également espacés entre 1 et 12 (inclus).

Q16) Complétez `comportement_touche` de sorte que si la touche “t” est pressée alors on inverse les valeurs des colonnes.

Q17) Complétez `comportement_touche` de sorte que si la touche “y” est pressée alors on inverse les valeurs des colonnes et des lignes.