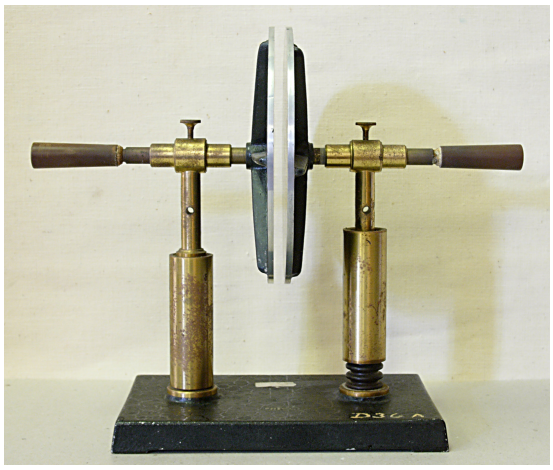
TP n°17 : Condensateur et villes de France



1 Charge et décharge d'un condensateur

L'objectif de cette première partie du TP est de se familiariser avec la manipulation de fichiers expérimentaux, constitués de mesures effectuées en séance de travaux pratiques. Cela permettra de réutiliser nos acquis sur les tableaux NUMPY et sur les tracés de graphes.

Les fichiers de mesures réalisés par votre professeur correspondent à la charge et à la décharge d'un condensateur. Dans ces fichiers, vous trouverez une colonne contenant les différents temps, une colonne contenant l'intensité du courant dans le circuit au cours du temps, une colonne contenant la tension délivrée par le générateur et une colonne contenant la tension aux bornes du condensateur au cours du temps.

Malheureusement, ce fichier enregistré au format texte (`.txt`) provient d'un tableur et présente quelques pièges :

- Les décimales sont situées après une virgule et non un point
- La première ligne correspond à l'en-tête des variables et non aux mesures
- Sur chaque ligne, les valeurs des différents attributs sont séparées par des ; (point-virgules)

Les mesures ont été faites avec $R = 10\ k\Omega$, $C \approx 100\ nF$ *et* $E = 7.44\ V$.

1. Écrire une fonction `lire(nom_fichier)` permettant de récupérer les mesures présentes dans le fichier sous forme d'une liste de quatre listes de flottants de mêmes tailles contenant les valeurs des différentes variables. *Conseil : allez-y progressivement, il y a beaucoup de choses à faire dans cette simple question. Utilisez les méthodes* `.strip()`, `.split()` *et* `.join()` *sur les chaînes de caractères.*

La commande suivante permet alors de récupérer les informations des mesures effectuée lors de la charge du condensateur :

PYTHON
```
temps, intensite, tension_gene, tension_cond = lire('charge.txt')
```

2. Représenter l'intensité en fonction du temps. On attend un graphe complet avec notamment un titre, une légende et des axes nommés.
3. Écrire une fonction `calcul_tau(intensite, temps)` qui renvoie le temps caractéristique τ du circuit *RC*. *On rappelle que le régime transitoire est terminé à 95% au bout de* 3τ.
4. Superposer sur un même graphe le tracé des mesures expérimentales et l'expression théorique de l'intensité en fonction du temps. Conclure. *On rappelle que l'intensité lors d'une charge ou d'une décharge d'un circuit RC série se met sous la forme* $i(t) = i(0)\exp\left(-\frac{t}{\tau}\right)$.
5. Représenter, à l'aide de la fonction `subplot`, les graphes de l'évolution de l'intensité, de la tension aux bornes du générateur, de la tension aux bornes du condensateur au cours du temps ainsi que le portrait de phase du système. On sauvegardera la figure dans son dossier personnel. *On rappelle que le portrait de phase d'une grandeur X correspond au tracé de* $\dot{X}$ *en fonction de X et que pour un condensateur idéal on a* $C\dot{u} = i$.
6. Reprendre la question précédente avec la décharge du condensateur.

https://imgs.xkcd.com/comics/urgent_mission.png

2 Villes de France

Dans cette deuxième partie on montre comment il est possible d'interagir avec une base de données et d'utiliser le résultat des requêtes pour en effectuer un traitement avec PYTHON.

Nous utiliserons la bibliothèque `sqlite3` que vous pouvez déjà importer :

PYTHON
```
import sqlite3 as sq
```

Voici comment manipuler les bases de données :

PYTHON

```python
# Ouverture d'une base de données
with sq.connect("villes.db") as bd_villes:

    # Création d'un "curseur" permettant d'interroger la base
    cursor = bd_villes.cursor()

    # Execution d'une requête
    cursor.execute("Votre requête")

    # Avec récupération des résultats
    resultat = cursor.execute("Votre requête")

for ligne in resultat:
    ...
```

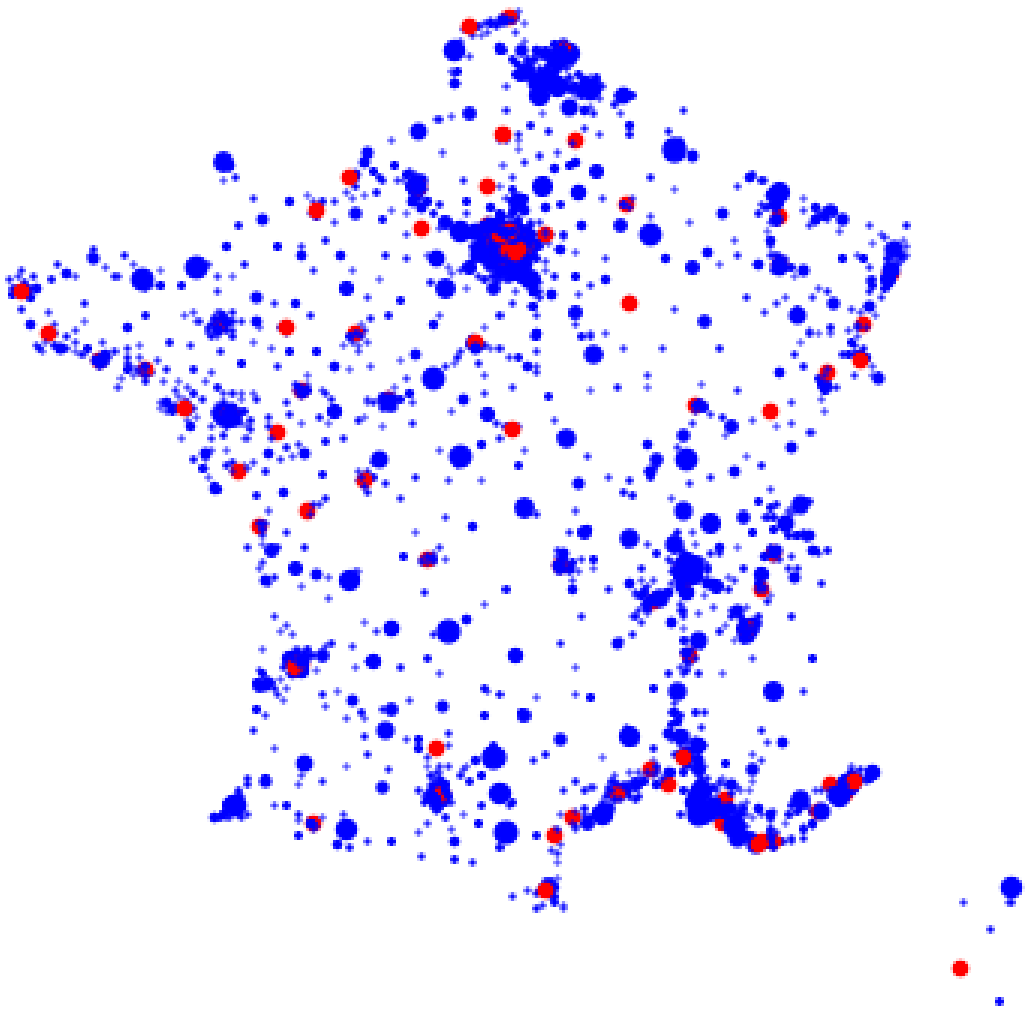


FIGURE 1 – Villes de plus de 2000 habitants.

On fournit un fichier `villes.db` contenant des données des villes françaises suivant le schéma :

SQL

```sql
CREATE TABLE villes (
  id INTEGER,
  departement TEXT,
  nom TEXT,
  code_postal TEXT,
  population INTEGER,
  longitude REAL,
  latitude REAL,
  zmin INTEGER,
  zmax INTEGER,
  PRIMARY KEY (id)
);
```

7. Tracer[1] la carte de France des villes de plus de 2000 habitants. Les coordonnées des villes sont $(\alpha \times \text{longitude}, \text{latitude})$ où $\alpha = \cos(47^o)$.
8. Tracer la carte de France des villes dont l'altitude minimale z_{min} est 0. Que remarquez-vous ?
9. Tracer la carte des villes de Côte-d'Or dont l'altitude minimale est inférieure au minimum des altitudes maximales des villes de Côte-d'Or. Cela permet de rafraîchir un peu sa mémoire SQL !
10. D'autres idées ? Amusez-vous ! Vous pouvez varier les plaisirs en imposant des conditions sur les villes à afficher, la taille des points, le choix du département, etc. N'hésitez pas à partager vos plus belles trouvailles !

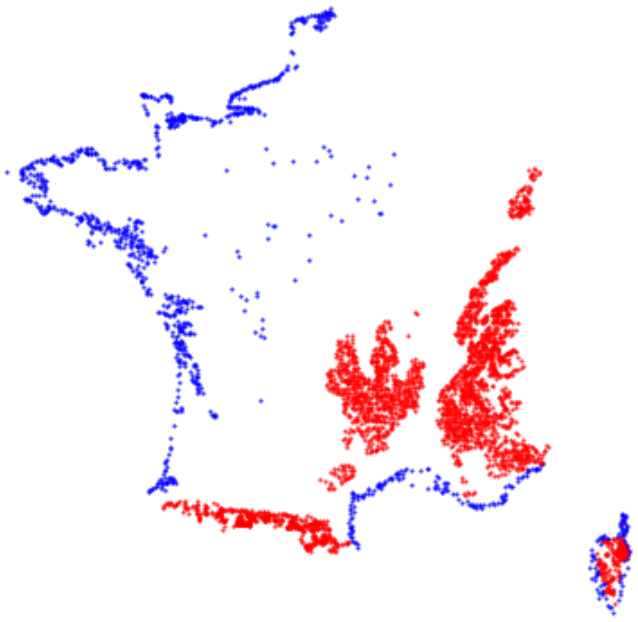


FIGURE 2 – Villes au niveau de la mer (en bleu) ou d'altitude supérieure à 1000 m (en rouge).

1. C'est normal si cela prend un peu de temps.