

# Semaine 6 : du 6 au 10 octobre

---

*La colle commence par une application de cours extraite de la liste ci-dessous et se poursuit par un exercice.*

*Je rappelle que vous trouverez sur mon site la version complétée du poly de cours, ainsi que les corrigés des TD et des DM. N'hésitez surtout pas à me signaler tout lien manquant ou défectueux.*

---

## Au programme

---

### Chapitre E2 : Transitoires du premier ordre

Applications de cours et exercices.

- Le TP E2.1 est à revoir au même titre que le TD.

### Chapitre M1 : Lois de Newton

Applications de cours et exercices.

### Outils pour la physique : Incertitudes

- Estimation de type A (plusieurs répétitions d'une même mesure) d'une valeur expérimentale et de son incertitude-type (paragraphes II.A, V.A et V.B de la fiche outil);
- Composition des incertitudes dans le cas d'une somme (paragraphe III.A);
- Estimation de l'incertitude sur une grandeur calculée par simulation Monte-Carlo (paragraphe III.B);
- Comparaison à une valeur attendue ou entre deux valeurs par le z-score (paragraphe IV).

Aucun développement théorique, ni aucun exercice spécifique, en revanche une question d'incertitudes est la bienvenue au sein d'un exercice.

### Outils pour la physique : Utilisation de Python

Écrire un code Python en entier n'est pas exigible des étudiants cette semaine, en revanche compléter une partie de code et/ou savoir expliquer un code élémentaire l'est (méthode d'Euler, traitement de données expérimentales, tracé de courbes). On utilisera (exclusivement) les bibliothèques numpy et matplotlib.

---

## Applications de cours

---

*Ces applications de cours sont des « briques élémentaires » des raisonnements à mener dans les exercices : les maîtriser est incontournable. Elles sont toutes traitées de manière exhaustive dans le cours.*

*Le travail demandé consiste à se les approprier, afin d'être capable de les réinvestir dans un sujet d'écrit ou d'oral. Je n'attends pas des étudiants un apprentissage par cœur, mais j'attends qu'ils les aient travaillées suffisamment pour les mener à bien en autonomie, c'est-à-dire savoir refaire seul les raisonnements, sans aide de l'interrogateur.*

**E2.1** - En raisonnant par analyse dimensionnelle, établir l'expression du temps caractéristique  $\tau$  d'un circuit RC ou RL, au choix de l'interrogateur. On supposera que  $\tau$  peut a priori dépendre des composants et de l'amplitude  $E$  de l'échelon de tension imposé.

**E2.2** - Établir et résoudre l'équation différentielle vérifiée par l'intensité dans un circuit RC série soumis à un échelon de tension  $0 \rightarrow E$  à l'instant initial. Représenter l'allure de la courbe  $i(t)$ .

**E2.3** - Établir et résoudre l'équation différentielle vérifiée par l'intensité dans un circuit RL série alimenté par un générateur de fém constante  $E$  après fermeture de l'interrupteur à l'instant initial. Représenter l'allure de la courbe  $i(t)$ .

**E2.4** - La tension aux bornes du condensateur d'un circuit RC série soumis à un échelon de tension a pour expression

$$u_C(t) = E \left( 1 - e^{-t/\tau} \right).$$

Procéder au bilan énergétique : calculer de manière séparée le travail électrique total fourni par le générateur, l'énergie reçue par le condensateur et l'énergie dissipée dans la résistance. Calculer le rendement énergétique. Commenter.

**E2.5** - Écrire un code Python permettant de résoudre par la méthode d'Euler explicite l'équation différentielle

$$\frac{du}{dt} + \frac{1}{\tau}u = \frac{e(t)}{\tau}.$$

- Avant toute écriture de code, on commencera par établir la relation de récurrence pertinente.
- Les paramètres tau, dt (pas de temps) et N (nombre de pas de temps) seront considérés comme des variables globales **déjà définies**, et on supposera disposer d'une liste  $e$  **déjà remplie** de manière adéquate.
- L'étudiant devra construire correctement (initialisation comprise) les listes  $t$  et  $u$ .

**M1.1** - Établir l'équation du mouvement d'une chute libre, et la résoudre pour établir les lois horaires  $(x(t), z(t))$ . **Les intégrations seront menées par séparation de variables.**

**M1.2** - Les équations horaires d'un mouvement de chute libre sont données par

$$\begin{cases} x(t) = v_0 \cos \alpha t \\ z(t) = -\frac{1}{2}gt^2 + v_0 \sin \alpha t \end{cases}$$

En déduire l'équation de la trajectoire, la portée et la flèche du tir.

**M1.3** - Établir l'équation différentielle vérifiée par la vitesse  $v$  d'un parachutiste chutant à la verticale dans l'air, en considérant une force de frottement quadratique. Adimensionner l'équation différentielle pour identifier la vitesse limite  $v_\infty$  et le temps caractéristique  $\tau$  nécessaire pour atteindre cette vitesse.

**M1.4** - Écrire un code Python permettant de résoudre par la méthode d'Euler explicite l'équation différentielle adimensionnée

$$\frac{dv}{dt} + v^2 = 1 \quad \text{avec} \quad v(0) = 0.$$

- Avant toute écriture de code, on commencera par établir la relation de récurrence pertinente.
- L'étudiant devra définir les paramètres de la simulation (pas de temps et durée totale adimensionnés), et construire correctement les listes  $t$  et  $v$ .
- En revanche, je n'attends pas le code permettant le tracé de la solution.

## À quoi s'attendre pour les programmes suivants ?

- Chapitre O2 : Instruments d'optique ;
- Chapitre C1 : Cinétique chimique.