



Lycée voie(s)	Générale	Technologique	Professionnelle	Enseignement
				Commun
				De spécialité
				Optionnel
	CAP	Seconde	Première	Terminale
Physique-chimie				

À la découverte de la radioactivité

Modélisation d'un phénomène aléatoire par une décroissance exponentielle

Cette séquence est une proposition d'introduction à la notion sur **l'évolution temporelle d'une population de noyaux radioactifs**. Les élèves expérimentent à la maison un tirage aléatoire à l'aide de pièces de monnaie et remplissent un tableur collaboratif. Un programme Python leur est ensuite proposé, permettant de modifier la probabilité du processus, de faire varier le nombre de tirages et d'étudier un échantillon arbitrairement grand. En effet, l'utilisation du programme Python permet d'étendre le modèle utilisant la fonction exponentielle à d'autres types de phénomènes aléatoires tels que la décroissance radioactive.

Enfin, facultativement, une expérience quantitative et très simple à réaliser à la maison est proposée pour étudier un autre processus physique modélisé par une fonction exponentielle.

Prérequis

Les élèves ont déjà abordé :

- l'évolution temporelle d'un système chimique ;
- dynamique d'un système électrique ;
- la loi phénoménologique de Newton en thermodynamique.

Références au programme

Notions et contenus

- Évolution temporelle d'une population de noyaux radioactifs ; constante radioactive ; loi de décroissance radioactive ; temps de demi-vie ; activité.

Capacités exigibles

- Établir l'expression de l'évolution temporelle de la population de noyaux radioactifs.
- Exploiter la loi et une courbe de décroissance radioactive.

Capacité mathématique : Résoudre une équation différentielle linéaire du premier ordre à coefficients constants.

Compétences travaillées dans le cadre de la démarche scientifique

Analyser Raisonner - Capacités associées

- Choisir un modèle ou des lois pertinentes.
- Faire des prévisions à l'aide d'un modèle.
- Procéder à des analogies.
- Comprendre le fonctionnement d'un programme Python fourni.

Réaliser - Capacités associées

- Effectuer des procédures courantes : calculs, représentations, collectes de données.
- Utiliser l'outil numérique (tableur-grapheur, et simulation par programme Python).

Valider - Capacités associées

- Confronter un modèle à des résultats expérimentaux.
- Proposer d'éventuelles améliorations de la démarche ou du modèle.
- Faire preuve d'esprit critique.

Description de la séquence sur quatre séances sous forme d'enseignement hybride

Séance 1



Lancer de pièces de monnaie, et mise en commun des résultats dans un tableur collaboratif.

Séance 2



Modélisation des résultats obtenus lors du tirage des pièces de monnaie:

Séance 3



Exploitation d'un programme Python fourni, permettant d'étudier un échantillon de taille arbitrairement grande, et de modifier la probabilité du processus, afin d'améliorer la confiance dans la modélisation théorique.

Séance 4



Retour sur le programme Python : mise en commun des observations.
Institutionnalisation des connaissances sur les transformations nucléaires.

Séance 5 (optionnelle)



Réalisation et exploitation d'une expérience quantitative très simple modélisable par le même type de loi temporelle du 1^{er} ordre en $N_0 e^{-\lambda t}$.



Séance 1 - Lancer de pièces et remplissage du tableur collaboratif

1^{re} étape du travail à la maison

On demande à chaque élève de prendre un grand nombre $N(0) = N_0$ de pièces de monnaies (une vingtaine au minimum). L'élève les lance toutes simultanément sur un support, les étale, et élimine définitivement les pièces étant tombées sur FACE. Il compte le nombre de pièces restantes (sur PILE) : $N(1)$. L'élève relance alors ces pièces, élimine les FACE, et compte les PILE restantes : $N(2)$, etc.

Le professeur a préalablement préparé un tableau à remplir sur un tableur collaboratif en ligne (Google Sheets¹, Framacalc², etc.) où chaque élève doit remplir dans sa colonne le nombre de pièces restantes $N(i)$ à chaque lancer. Le total de tous les élèves à chaque lancer se calcule automatiquement dans une colonne dédiée (colonne TOTAL).

	TOTAL	Elève 1	Elève 2	Elève 3	Elève 4	Elève 5	Elève 6	Elève 7	Elève 8	Elève 9	Elève 10	Elève 11	Elève 12	Elève 13	Elève 14	Elève 15	Elève
Nb total de pièces	315	20	20	35	20	20	20	20	20	20	20	20	20	30	30		
1e lancer	141	9	9	15	12	9	8	6	8	7	12	9	11	15	11		
2e lancer	64	3	4	5	8	5	6	1	3	4	6	5	5	5	4		
3e lancer	37	2	4	2	6	3	2	1	2	1	5	1	2	3	3		
4e lancer	21	1	3	1	4	2	2	1	0	1	3	0	0	1	2		
5e lancer	9	1	1	0	1	1	0	0	0	1	3	0	0	0	1		
6e lancer	5	1	0	0	1	0	0	0	0	0	3	0	0	0	0		
7e lancer	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0		
8e lancer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
9e lancer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
10e lancer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Figure 1 – Exemple de tableur collaboratif (déjà rempli par des élèves).

Remarque

Il peut être judicieux pédagogiquement de scinder la classe en deux et préparer deux tableaux, un pour chaque moitié de classe. Ainsi, la modélisation de l'évolution en $e^{-\lambda t}$ sur les deux sous-groupes sera probablement jugée plus convaincante par les élèves, car reproductible.

2^e étape du travail à la maison

Une fois que tous les élèves ont rempli le tableau, on leur demande retourner sur le tableur pour récupérer la colonne **TOTAL**, et de tracer graphiquement le nombre de pièces $N(i)$ en fonction du numéro i de lancer avec le logiciel tableur-grapheur de leur choix.

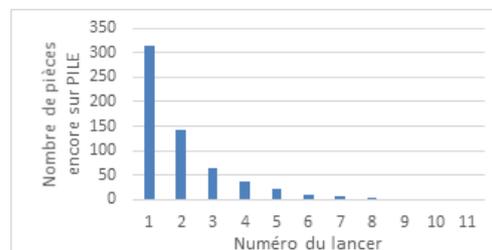


Figure 2 – Graphe obtenu à partir du résultat des quatorze élèves saisis dans le tableau précédent.



Séance 2 - Validation du modèle exponentiel

Il s'agit lors de cette séance d'amener les élèves à se questionner quant à la modélisation des résultats du lancer de pièces par une fonction théorique, pour aboutir à la modélisation suivante :

$$N(i) = N_0 e^{-\lambda i}$$

On pourra questionner les élèves : « A-t-on déjà rencontré cette année des situations où une grandeur physique présentait une évolution semblable au cours du temps, avec une décroissance asymptotique vers 0 ? » Les réponses attendues sont la concentration d'un réactif lors d'une réaction chimique dont la loi de vitesse était d'ordre 1, et la décharge d'un condensateur dans un circuit RC.

¹ <http://sheets.google.com>

² <http://www.framacalc.org>

Suivant les propositions des élèves, le professeur peut alors effectuer la modélisation des résultats obtenus à l'aide de l'outil informatique de son choix, devant la classe (utilisation du vidéoprojecteur...) :

- soit en traçant directement $N(i)$ en fonction de i et en modélisant par une loi exponentielle ;
- soit après avoir linéarisé, c'est-à-dire en traçant $\ln\left(\frac{N(i)}{N(0)}\right)$ en fonction de i et en effectuant une régression linéaire $\ln\left(\frac{N(i)}{N(0)}\right) = -\lambda \cdot i$ (si le nombre de pièces restantes a eu le temps d'atteindre 0, il faut éliminer ces valeurs avant de passer au logarithme)

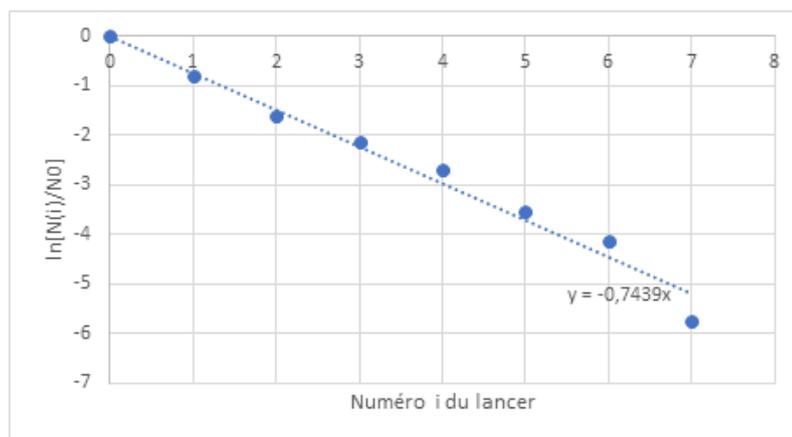


Figure 3 – Exemple de régression linéaire, peu convaincante, obtenue à partir des résultats de la figure 2.

Il est important de remarquer que le seul paramètre d'ajustement est λ , en effet il est légitime de fixer N_0 à la valeur du nombre initiale de pièces. Or, à ce stade de la séquence pédagogique proposée, il n'est pas possible pour les élèves de valider ou d'invalider le modèle autrement qu'en se contentant de vérifier qualitativement que les points expérimentaux semblent positionnés aléatoirement autour de la droite... En effet, il est impossible de comparer la pente obtenue à une valeur attendue car les élèves ne connaissent pas cette valeur à ce stade. Pour information, la valeur théorique attendue de la pente est : $-\lambda = -\ln(p)$ avec p la probabilité de tomber sur PILE, soit $-\lambda = -\ln(p) = -\ln(0,5) \approx 0,69$ (expliqué plus loin dans ce document [séance 4]).

Les élèves peuvent être sceptiques face à ce choix de modélisation (avec raison). C'est l'occasion de poursuivre le questionnement : *Comment pourrait-on améliorer notre expérience pour rendre notre choix de modélisation plus robuste ?*

Les réponses attendues sont :

- utiliser un nombre de pièces beaucoup plus grand ;
- répéter l'expérience un très grand nombre de fois.

Les élèves sont ainsi amenés à l'idée de l'utilisation d'une simulation numérique de ce processus aléatoire, permettant d'explorer des échantillons arbitrairement grands et de répéter l'expérience plusieurs fois extrêmement rapidement.

Le professeur fournit alors le programme Python (fourni en [annexe 1](#)), et demande aux élèves comme travail à la maison pour la séance prochaine, d'utiliser ce programme pour répondre au questionnaire (fourni en [annexe 2](#)).

Remarque pratique

Si les élèves n'ont pas la possibilité d'exécuter un programme Python sur leur machine personnelle, le professeur peut déposer le programme sur <https://trinket.io> (attention : choisir « Python 3 », et non « Python », pour que ce programme fonctionne: <https://trinket.io/features/python3>). Il suffit alors de fournir l'URL du programme aux élèves, qui peuvent alors l'exécuter et modifier le code à leur guise depuis un simple navigateur Internet.

```

1 import random
2 from matplotlib import pyplot
3 import numpy as np
4 import matplotlib.path_effects as path_effects
5
6 nbPieces = int(input("Combien de pièces ?\n"))
7 nbTirages = int(input("Combien de tirages ?\n"))
8 proba = float(input("Quelle probabilité de tomber sur PILE (entre 0.0 et 1
9 affichage = input("Voulez-vous afficher l'état de chacune des " + str(nbPi
10 tracerExp = input("Voulez-vous ajouter sur le graphe la courbe exponentiel
11 tableau = input("Voulez-vous afficher le tableau de valeurs (p.ex. pour un
12
13 listePieces = ["P" for i in range(0,nbPieces)]
14
15 if affichage=="0" or affichage=="o":
16     print(listePieces)
17
18 nbDePiles = [""] for i in range(0,nbTirages)]
19 nbDePiles[0] = listePieces.count("P") # compte le nombre de PILES au tirage
20
21 for tirage in range(1,nbTirages):
22     for numeroPiece in range(0,nbPieces): # pour chaque pièce
23         if listePieces[numeroPiece]=="P": # si cette pièce est sur pile (:
24             aleatoire = random.random() # tire un nombre au hasard
25             if aleatoire >= proba: # si ce nombre est > à la p
26                 listePieces[numeroPiece]="F" # la pièce devient FACE
27
28             if affichage=="0" or affichage=="o":
29                 print(listePieces)
30
31             nbDePiles[tirage] = listePieces.count("P") # compte le nombre de PILES
32
33 # Définition de la fonction exponentielle théorique
34 if tracerExp=="1" or tracerExp=="o":
35     def expo(x):
36         return nbDePiles[0]*proba**x
37
38

```

grand...
n
Voulez-vous ajouter sur le graphe la courbe exponentielle théorique ? (O/N)
o
Voulez-vous afficher le tableau de valeurs (p.ex. pour un copier/coller vers un tableur) ? (O/N)
n

10 tirages de 50 pièces avec une proba. de tomber sur PILE de 50.0%

Nombre de pièces encore sur PILE

Numéro du tirage

trinket_plot.png

Figure 4 – Copie d'écran du programme tournant sur le site Internet Trinket.io.



Séance 3 - Utilisation du programme Python

Les élèves s'approprient le programme Python ([annexe 1](#)) en le « manipulant » librement. Ils répondent aux questions du [questionnaire](#) en [annexe 2](#). Les élèves transfèrent sur l'ENT leurs réponses en amont afin d'anticiper la séance suivante au lycée. Ce questionnaire permet de réaliser une évaluation diagnostique du travail pour le professeur afin d'adapter le temps d'institutionnalisation des connaissances.

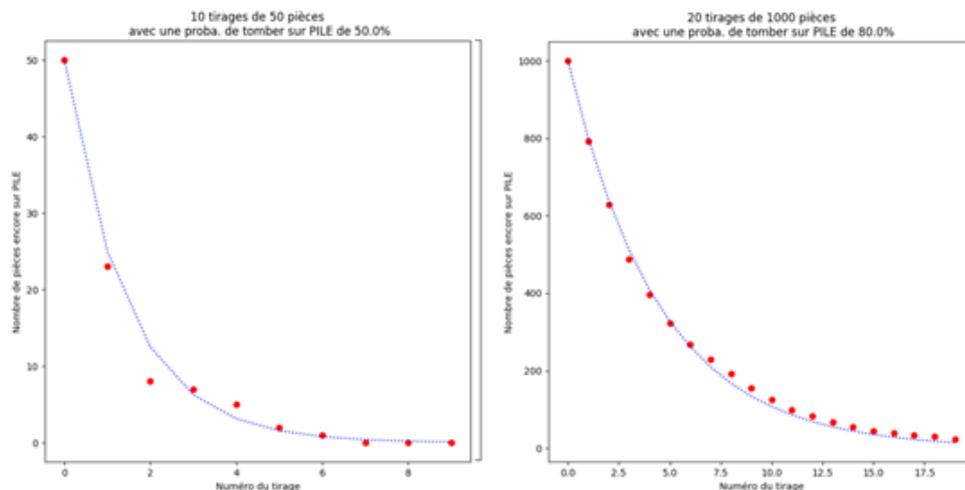


Figure 5 – Exemples sur le site Internet Trinket.io.



Séance 4 - Institutionnalisation des connaissances sur les transformations nucléaires

En classe, le professeur revient sur les résultats que les élèves ont obtenus avec le programme Python, leurs réponses au questionnaire, ce qui permet de valider définitivement ce modèle d'évolution par une fonction exponentielle.

Le professeur indique aux élèves que **la désintégration radioactive est également un phénomène aléatoire**, au même titre que le tirage des pièces de monnaie, et qu'en conséquence elle suit également la même loi d'évolution temporelle. Il pourra également préciser que le caractère aléatoire du processus de désintégration radioactive, à l'échelle microscopique, se retrouve également dans l'étude cinétique de la réaction chimique, avec la notion de chocs efficaces, ce qui explique que macroscopiquement on observe une concentration en réactif évoluant de la même manière, pour les réactions d'ordre 1³.

Le professeur amène les élèves à dresser les correspondances entre grandeurs :

- $N(i)$ le nombre de pièces restantes sur PILE avec le nombre de noyaux instables qui ne se sont pas encore désintégrés,
- le numéro i du lancer (ou le nombre de lancers effectués) et le nombre d'unités de temps t .

Comme les élèves ont pu le constater avec le programme Python, si la probabilité de tomber sur PILE (ou, pour un noyau radioactif, de se désintégrer) augmente, la diminution de la population se fait plus lentement. Le professeur indique qu'alors le paramètre λ dans la modélisation diminue.

Complément possible

Si le niveau de la classe le permet, le professeur peut montrer explicitement le lien entre ce paramètre λ et la probabilité p de désintégration par unité de temps (ou de tomber sur PILE par lancer, pour une pièce de monnaie), et ainsi mettre en évidence l'obtention de la loi exponentielle dans la limite des grands nombres :

$$N(0) = N_0$$

$$N(1) = N(0) \times p = N_0 \times p \quad (p = 1/2 \text{ par exemple pour une pièce non truquée})$$

$$N(2) = N(1) \times p = N_0 \times p^2$$

$$N(3) = N(2) \times p = N_0 \times p^3$$

...

$$N(i) = N(i-1) \times p = N_0 \times p^i$$

$$N(i) = N_0 \times e^{\ln(p) \times i} = N_0 \times e^{-\lambda \times i}$$

Avec $-\lambda = -\ln(p)$ (on a $\lambda > 0$ car la probabilité $p < 1$).

Pour les lancers de pièces (processus discret), l'argument de l'exponentielle est $-\lambda i$ avec i le numéro de lancer, ainsi $-\lambda = -\ln(p)$ et est sans dimension.

Pour les désintégrations radioactives (modélisation continue du processus), l'argument de l'exponentielle est $-\lambda t$, ainsi λ est associé à la probabilité de désintégration *par seconde* (ou fréquence de désintégration), d'où sa dimension T^{-1} .

³ Le mécanisme réactionnel de ces réactions chimiques ne comportant qu'un acte élémentaire.

En exportant les résultats obtenus avec le programme Python vers un logiciel de traitement de données (Regressi, tableur...), il est possible à nouveau de modéliser par une droite après linéarisation, comme cela avait été fait en séance 2, mais cette fois-ci avec un nombre de pièces beaucoup plus grand.

On tracera donc $Y = \ln\left(\frac{N(i)}{N(0)}\right) = -\lambda \cdot i$ (avec $N(0)$ à remplacer par la valeur choisie) en ordonnées et $X = i$ le numéro du lancer en abscisses, et on modélisera par une droite passant par l'origine $Y = aX$.

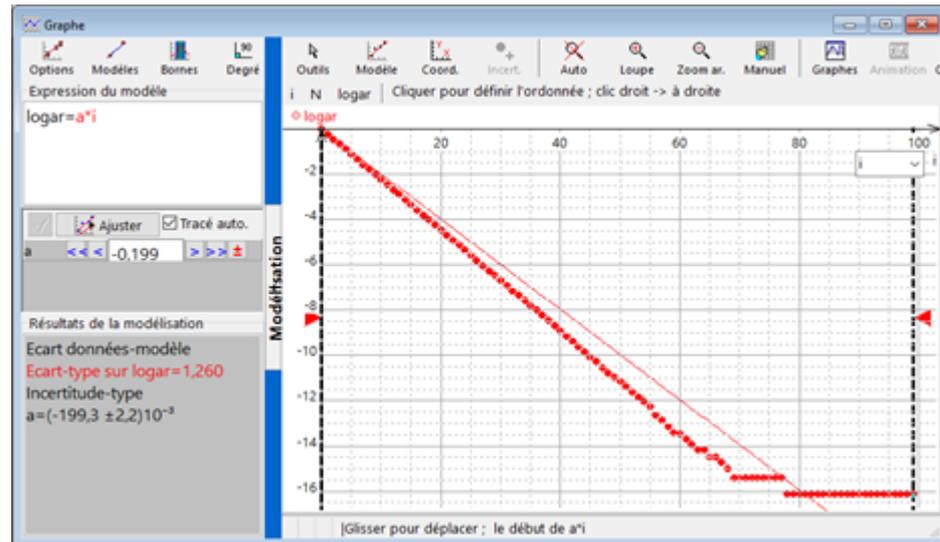


Figure 6 - Exemple de modélisation obtenue sur Regressi avec la simulation sur 100 lancers de 10^7 pièces, avec une probabilité de tomber sur PILE de $p = 0,8$.

Remarques

Il est possible d'importer très rapidement dans Regressi les valeurs numériques i ; N retournées par le programme Python en les copiant dans le presse-papiers (Ctrl + C) et en faisant Fichier > Nouveau > Presse-papiers dans Regressi.

Comme lors de la séance 2, si le nombre de pièces restantes a eu le temps d'atteindre la valeur 0, il faut éliminer ces dernières valeurs avant de passer au logarithme.

Si le professeur a montré (ou fait admettre) aux élèves que la pente attendue théoriquement est de $-\lambda = \ln(p)$ avec p la probabilité de tomber sur PILE (voir encadré « Complément possible » plus haut), il est donc désormais possible de pousser plus loin la validation de la modélisation en comparant la pente de la droite de régression avec cette valeur attendue.

L'effectif étant désormais beaucoup plus grand que lors de l'expérience avec les lancers réels, l'adéquation des résultats avec la droite de régression est beaucoup plus convaincante. Toutefois, on voit très clairement que les points s'écartent notablement de la modélisation lorsque le nombre de pièces restantes (ou de noyaux radioactifs restants) devient trop petit, ce qui est compréhensible puisque nous ne sommes plus dans la limite des grands nombres. On fera remarquer aux élèves qu'en radioactivité, le nombre de noyaux dans un échantillon radioactif n'atteint en pratique jamais des valeurs aussi faibles, mais reste de l'ordre du nombre d'Avogadro. Ainsi, il sera judicieux d'exclure les derniers points de la modélisation.

Points d'attention

N_0 doit être suffisamment grand, mais l'ordinateur ne peut évidemment pas gérer un échantillon de l'ordre de 10^{23} noyaux, sans risque de blocage. En effet, le programme stocke des listes de $N(0)$ de valeurs 'P' ou 'F' (pile ou face) pour chaque tirage (voir figure 6), et fait une boucle sur chacune de ces valeurs⁴, et ceci pour chaque tirage. Nous suggérons vivement au professeur de tester préalablement quelle valeur du nombre total de pièces/noyaux il peut choisir sans bloquer l'ordinateur, selon la puissance de ce dernier⁵.

Le cours sur les transformations nucléaires peut alors se faire à ce stade de la séquence. Le professeur est libre de réaliser cette institutionnalisation des savoirs sur la radioactivité comme il le souhaite.

S'en suit une évaluation formative qui peut avoir lieu en autonomie, en distanciel, sous forme de quizz par exemple. Outre l'évaluation des connaissances acquises sur les phénomènes radioactifs, cette évaluation peut **mettre en évidence les liens avec les autres domaines du programme où cette modélisation par une fonction exponentielle est utilisée** : dynamique du circuit électrique « RC », cinétique chimique d'ordre 1, et loi de Newton en thermodynamique (par exemple : lien entre la constante de temps τ et temps de demi-réaction ou de demi-vie $t_{1/2}$, etc.).

⁴ Cela se fait en particulier très simplement et visuellement avec le logiciel Regressi, en déplaçant à la souris le curseur vertical définissant la borne de droite de l'intervalle à modéliser.

⁵ À titre d'exemple, nous avons choisi un nombre de pièces de 10 millions, ce qui a nécessité plus d'une minute à l'ordinateur pour exécuter le programme. Il semble raisonnable de ne pas aller beaucoup plus loin.



Séance 5 (optionnelle) - Expérimentation pratique hors la classe d'une évolution modélisable par $e^{-\lambda t}$.

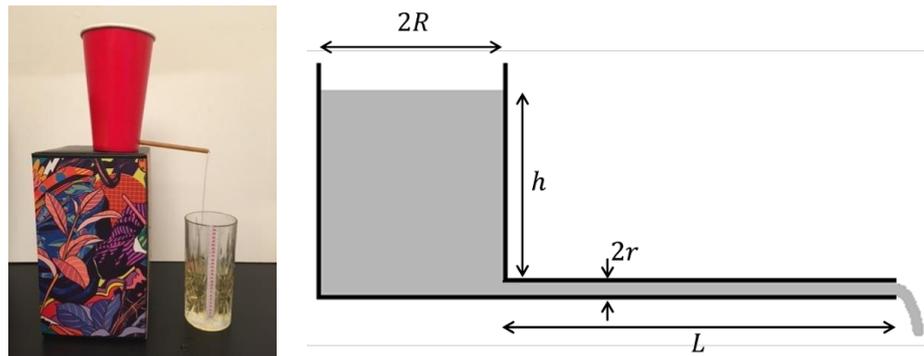
L'inconvénient pédagogique des transformations nucléaires est qu'elles ne se prêtent à aucune expérimentation en classe ou à la maison. Les autres processus physiques rencontrés au cours de l'année et modélisables une évolution en $e^{-\lambda t}$ peuvent être étudiés en travaux pratiques, mais ne peuvent être expérimentés à la maison (réaction chimique d'ordre 1, circuits électriques du 1^{er} ordre et transferts thermiques).

Nous proposons ici une expérience dans un tout autre domaine de la physique, qui présente le double avantage :

- d'être réalisable très simplement à la maison,
- d'être exploitable quantitativement.

Dispositif et principe de l'étude

Un grand gobelet, percé d'un trou au bout duquel est collée une paille, contient de l'huile alimentaire (masse volumique ρ , viscosité η) :



Figures 7 et 8 - Photo de l'expérience élève et Schéma de l'expérience.

La hauteur $h(t)$ d'huile dans le gobelet (ou le volume $V(t) = \pi R^2 h(t)$) varie au cours du temps selon la loi :

$$h(t) = h_0 e^{-t/\tau}$$

avec une constante de temps τ proportionnelle à la longueur L de la paille. Le fait que la section du gobelet ne soit pas tout-à-fait uniforme n'influe pas grandement sur les résultats.

Matériel nécessaire

- De l'huile de cuisine ;
- une paille ;
- un grand gobelet* qu'il est possible de percer (en carton, ou en plastique) ;
- de la colle forte ;
- un second récipient*, si possible de section intérieure à peu près uniforme.

*L'un des deux récipients (celui que l'on perce, ou l'autre) doit être transparent.

Protocole expérimental

Préparation du dispositif

Percer le gobelet d'un trou sur le côté, proche du fond, et y coller la paille avec une colle liquide forte.

Il faut que l'un des deux récipients soit transparent, afin de pouvoir repérer le niveau d'huile au cours du temps.



Figure 7 - Photo de l'expérience élève.

Si l'élève dispose chez lui d'un verre doseur, il peut l'utiliser comme récipient de récupération : celui-ci sera donc déjà gradué.

Dans le cas contraire, il faut graduer manuellement le récipient transparent (soit le gobelet-réservoir, soit le récipient de récupération).

Si la section du récipient est à peu près uniforme, il suffit donc de placer des graduations équidistantes.

Si ce n'est pas le cas, on peut verser à plusieurs reprises un petit volume d'eau identique et tracer une graduation sur le niveau d'eau à chaque remplissage.

Si le fond du récipient est « arrondi » sur les bords, on peut le remplir préalablement d'huile sur une très faible hauteur, et placer la graduation « 0 » à ce niveau de départ.

Prise de mesures

On place le gobelet en hauteur, on le remplit d'huile en bouchant l'extrémité de la paille avec un doigt. On retire le doigt, et on suit l'évolution de la hauteur d'huile dans le récipient transparent gradué (soit le gobelet-réservoir, soit le récipient de récupération), en fonction du temps.

Pour plus de commodité, il est astucieux de filmer le niveau d'huile avec un téléphone. La visualisation de l'enregistrement avec l'affichage du temps vidéo permet de repérer facilement à quels instants le niveau passe sur une graduation.

Exploitation des résultats

On trace le volume V restant dans le réservoir en fonction du temps t , en unités arbitraires correspondant aux graduations :

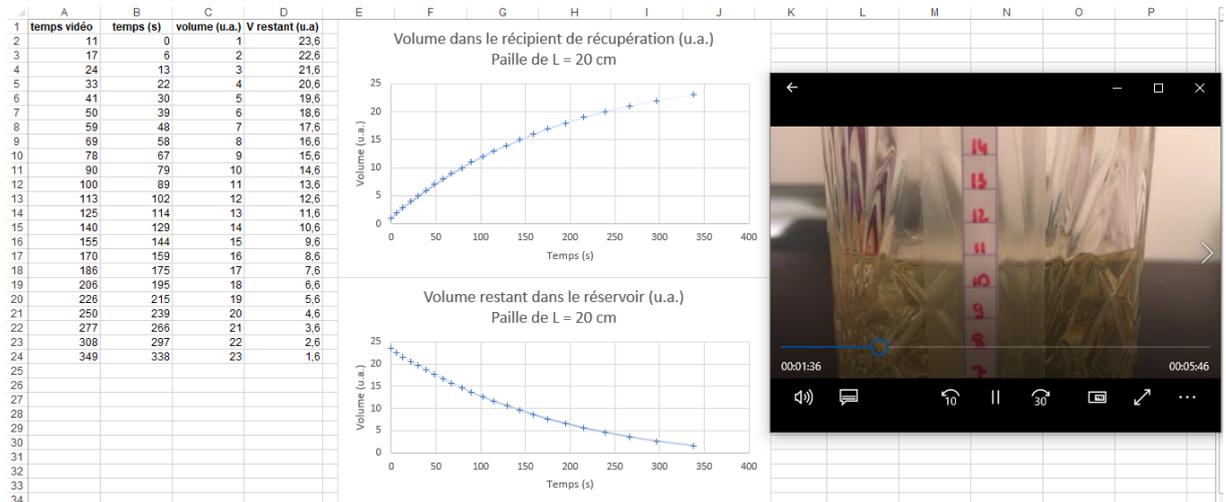


Figure 9 – Capture d'écran avec l'outil de modélisation et la vidéo d'expérience.

À l'aide d'un outil de modélisation au choix, on modélise la courbe de h en fonction de t par :

$$h(t) = h_0 e^{-\lambda t}$$

Il faut prendre soin d'affecter l'instant $t = 0$ au passage par la graduation 0.

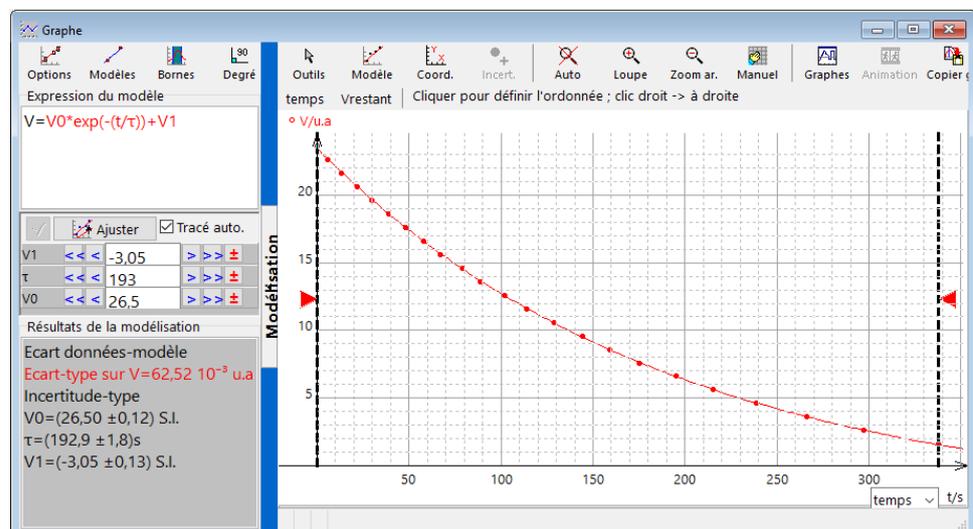


Figure 10 – Exemple de modélisation effectuée avec le logiciel Regressi.

On constate la bonne adéquation des mesures avec une loi d'évolution exponentielle.

Pour aller plus loin, les élèves peuvent recommencer l'expérience avec une paille de longueur différente, et constater que le paramètre λ croît avec la longueur de la paille.

Annexes

Annexe 1 - Programme python

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Jan 02 11:42:40 2021
@author: thomasroy
"""
import random
from matplotlib import pyplot
import numpy as np
import matplotlib.path_effects as path_effects

nbPieces = int(input("Combien de pièces ?\n"))
nbTirages = int(input("Combien de tirages ?\n"))
proba = float(input("Quelle probabilité de tomber sur PILE (entre 0.0 et 1.0) ?\n"))
affichage = input("Voulez-vous afficher l'état de chacune des " + str(nbPieces) + " pièces à chaque tirage ? (O/N) \n (Déconseillé si le nombre de pièces est grand...)\n")
tracerExp = input("Voulez ajouter sur le graphe la courbe exponentielle théorique ? (O/N)\n")

listePieces = ["P" for i in range(0,nbPieces)]

if affichage=="O":
    print(listePieces)

nbDePiles = [" " for i in range(0,nbTirages)]
nbDePiles[0] = listePieces.count("P") # compte le nombre de PILES au tirage n° 0

for tirage in range(1,nbTirages):

    for numeroPiece in range(0,nbPieces): # pour chaque pièce

        if listePieces[numeroPiece]=="P": # si cette pièce est sur pile (si elle est sur face, on ne la relance plus)

            aleatoire = random.random() # tire un nombre au hasard entre 0.0 et 1.0
            if aleatoire >= proba: # si ce nombre est > à la proba
                listePieces[numeroPiece]="F" # la pièce devient FACE

            if affichage=="O":
                print(listePieces)

            nbDePiles[tirage] = listePieces.count("P") # compte le nombre de PILES au tirage n° "tirage"

# Définition de la fonction exponentielle théorique
if tracerExp=="O":
    def expTheorique(x):
        return (nbPieces*np.exp(np.log(proba)*x))
    listeYtheorique = [expTheorique(j) for j in range(nbTirages)]

listeX = [j for j in range(nbTirages)]

figure = pyplot.figure(figsize=(8, 8))
axes = figure.add_subplot(111)

axes.plot(listeX,nbDePiles,color="red", linestyle='None', marker='o')

if tracerExp=="O":
    axes.plot(listeX,listeYtheorique,color="blue", linestyle=':')

titre = str(nbTirages) + " tirages de " + str(nbPieces) + " pièces\n avec une proba. de tomber sur PILE de " + str(proba*100) + "%
axes.set_title(titre)
axes.set_xlabel("Numéro du tirage")
axes.set_ylabel("Nombre de pièces encore sur PILE")

if tracerExp=="O":
    axes2 = figure.add_axes([0.1, -0.05, 0.8, .1])
    residus = [nbDePiles_elt - listeYtheorique_elt for nbDePiles_elt, listeYtheorique_elt in zip(nbDePiles, listeYtheorique)]
    axes2.bar(listeX,residus,color="green")
    axes2.set_frame_on(False)
    axes2.xaxis.set_visible(False)
    texte = axes2.text(0.5,0.7,"Résidus (écarts à l'exponentielle)",horizontalalignment='center', verticalalignment='center', transform = axes2.transAxes)
    texte.set_path_effects([path_effects.Stroke(linewidth=3, foreground='white'), path_effects.Normal()])
    pyplot.show()
    pyplot.close()
```

Annexe 2 - Questionnaire lié au programme Python

Remarque

Le professeur a toute liberté pour adapter ce questionnaire au niveau de ses élèves, en ôtant certaines questions jugées trop ardues, ou en ajoutant d'autres questions qui lui semblent pertinentes.

1. Identifier le nom de la variable contenant le nombre de pièces total initial, et indiquez la ligne du programme où elle est définie.
2. Identifier le nom de la variable contenant le nombre de lancers effectués, et indiquez la ligne du programme où elle est définie.
3. Dans le cas d'un tirage à pile ou face avec une pièce non truquée, que vaut la probabilité p de tomber sur PILE ? Identifier le nom de la variable contenant cette valeur, et indiquer la ligne du programme où elle est définie.
4. Identifier la ligne du programme où il est demandé à l'ordinateur de tirer un nombre au hasard. Comment se nomme la fonction effectuant cette opération ?
5. Exécuter le programme plusieurs fois avec un nombre de pièces et de tirages de plus en plus grands. Noter les observations. La modélisation des résultats par une loi $N(i) = N_0 e^{-\lambda i}$ vous semble-t-elle pertinente ?
6. On souhaite désormais simuler un lancer de dés, en éliminant à chaque lancer les dés tombés sur 6, et en ne gardant que les dés tombés sur 1, 2, 3, 4 ou 5. Quelle variable doit-on modifier dans le programme, et par quelle valeur la remplacer ?
7. Exécuter le programme simulant cette nouvelle expérience, et noter vos observations : comment est modifiée l'évolution du nombre d'objets non éliminés, par rapport à la simulation avec le tirage à PILE ou FACE ? La modélisation par une loi $N(i) = N_0 e^{-\lambda i}$ est-elle toujours pertinente ?