

Compétences nécessaires associées :

- **utiliser les instructions itératives : boucle for ;**
- **définir une liste ;**
- **tracer un graphe** (on se limitera pour l'instant à un graphe de listes à 2 dimensions ...).

I. Compétences nécessaires associées

1. Liste

a) Définition

Une liste est un tableau à une ligne (ou une colonne). On peut donc aussi la voir comme un vecteur ...

Il existe deux manières pour définir une liste :

- soit on veut créer une liste dont les éléments sont régulièrement espacés, on fera alors appel à la commande `linspace` ;
- soit on veut créer une liste de plus quelconque : on la verra alors comme un vecteur dont on définira les composantes par itération dans une boucle.

b) Syntaxe de la commande `linspace`

Si on veut créer une liste appelée x dont les valeurs sont régulièrement espacées entre la valeur minimum a , la valeur maximum b et qui contient un nombre n de valeurs, on utilise la commande `linspace` :

`x=linspace(a,b,n);`

Exemple : `x=linspace(1,10,100);` (puis double-cliquer sur x dans le navigateur des variables pour regarder la liste créée).

Remarques :

- *Il existe une deuxième manière pour définir une liste x dont les valeurs sont régulièrement espacées entre a et b mais où on fixe le pas p (on prendra garde au fait que la valeur b n'est donc pas forcément incluse dans la liste ...). La commande est : `y=a:p:b`;
Exemple : `y=1:10:100`; (puis double-cliquer sur y dans le navigateur des variables pour regarder la liste créée).*
- *Dans un premier temps nous n'utiliserons pas ces commandes dans l'algorithme d'Euler pour limiter les difficultés, on définira donc une liste comme un vecteur.*

2. Tracer un graphe à 2D

Si on veut tracer une liste de points appelée y en fonction d'une autre liste de points appelée x , on fait appel à la commande `plot` :

```
plot(x,y) //tracer y(x)
xlabel("nom de x")
ylabel("nom de y")
title("y(x)")
```

Remarques :

- on définira les fonctions et on montrera comment tracer un graphe d'une fonction au TD2 ;
- on verra comment tracer des graphes à 3D plus tard.

II. Méthode d'Euler pour résolution approchée d'une équation différentielle

1. Introduction – Rappel – Définition (TP 1 de physique)

Nous avons déjà parlé de la méthode d'Euler afin d'approximer la vitesse $v = \frac{dx}{dt}$ d'un point matériel à l'aide d'un tableur où les listes t et x sont déjà définies. La programmation se faisait en créant la liste v sous la forme : $v_n = \frac{x_{n+1} - x_n}{t_{n+1} - t_n}$.

La méthode d'Euler consiste donc à approximer $\frac{dx}{dt}$ à $\frac{x_{n+1} - x_n}{t_{n+1} - t_n}$ lors d'un algorithme (quel qu'il soit) ; toute la subtilité de la méthode d'Euler tenant au fait que $t_{n+1} - t_n$ doit être suffisamment petit.

Lors de la résolution numérique des équations différentielles par la méthode d'Euler, nous utiliserons le même genre d'approximation mais sous la forme $x_{n+1} - x_n = v_n \times (t_{n+1} - t_n)$ soit $x_{n+1} = x_n + v_n \times dt$ avec $dt = t_{n+1} - t_n$ (la valeur de v_n sera définie par l'équation différentielle et on pourra donc déterminer x_{n+1} de manière approchée).

2. Principe sur l'exemple de la désintégration radioactive

Prenons l'exemple de la désintégration radioactive : on considère un échantillon initial de N_0 noyaux radioactifs à $t=0$. On admet que leur désintégration suit la loi classique : $\frac{dN}{dt} = -\lambda N(t)$ où λ est appelée constante de désintégration radioactive (s'exprime en s^{-1}).

Remarque : oui nous savons résoudre de manière exacte cette équation mais ce n'est pas le but ici. Nous voulons obtenir une résolution numérique approchée. Intérêt : cette méthode de résolution, bien qu'approximative, aura l'avantage de nous servir lors des équations différentielles non linéaires !

La méthode d'Euler consiste à écrire qu'à l'instant $t+dt$, $N(t+dt) = N(t) + dN$ où $dN \simeq -\lambda N(t)dt$ (elle est ici l'approximation d'Euler) en imposant dt suffisamment petit dans la boucle d'itération. Connaissant la condition initiale $N(t=0) = N_0$, on peut calculer N de manière approximative au cours du temps.

Remarque : mais très petit devant quoi ce dt ?! Pour une équation différentielle linéaire, la réponse est simple ... Il faut que dt soit très petit devant le temps caractéristique d'évolution du système, défini par les formes canoniques. Ici, si on met l'équation différentielle sous forme canonique, il vient $\tau = \frac{1}{\lambda}$ donc à vous d'imposer $dt \ll \tau$ lors de la programmation.

Et donc, que se passe-t-il pour des équations différentielles plus compliquées voire non linéaires ?! Ce sera le but du TD2 ...

3. Exemple d'algorithme pour la désintégration radioactive

```
clear; //effacer toutes les variables en mémoire

N(1) = 1000; //population initiale (début de liste N vue comme un vecteur)
lambda = 1; //constante de désintégration radioactive en s-1
t(1) = 0; //déclaration du temps et on commence l'étude de la désintégration à
           t=0 (début de liste t vue comme un vecteur)
dt = 1/100; //définition du pas d'intégration de la méthode d'Euler, il doit être
           très petit ... devant 1/lambda !

for i=1:1000 //nombre d'itérations de la méthode d'Euler : on va jusqu'à tfin =
           tini + imax*dt
    N(i+1)=N(i)-lambda*N(i)*dt; //évolution de N(t) par la méthode d'Euler
    t(i+1)=t(i)+dt; //incrémentatation du temps pour pouvoir tracer N(t)
                   ultérieurement
end

plot(t,N) //tracer N(t)
xlabel("temps(s) ")
ylabel("N")
title("N(t) ")
```

III.Exploitation (TD)

1. Chute d'une goutte d'eau (application directe)

On rappelle l'équation différentielle qui régit l'évolution temporelle de la norme de la vitesse de chute de la goutte : $\frac{dv}{dt} + \frac{\lambda}{m}v(t) = g$. Tracer $v(t)$ par une résolution numérique faisant appel à la méthode d'Euler. On prendra pour condition initiale $v(t=0)=0$.

Aide pour l'algorithme : pour pouvoir utiliser la méthode d'Euler, il faut voir l'équation différentielle précédente sous la forme $dv = \left(g - \frac{\lambda}{m}v(t)\right)dt$ puis on utilise $v(t+dt) = v(t) + dv$.

2. Oscillateur harmonique

Un point matériel M de masse m est accroché à l'extrémité d'un ressort de raideur k et de longueur à vide ℓ_0 dont l'autre extrémité est fixée au point O . Les forces de frottement fluide et solide sont négligées. On appelle (Ox) l'axe vertical. L'équation différentielle régissant l'évolution temporelle est : $\ddot{x} + \frac{k}{m}x(t) = \frac{k}{m}\ell_0$.

Tracer $x(t)$ par une résolution numérique faisant appel à la méthode d'Euler. On prendra pour condition initiale $x(t=0)=L$ et $\dot{x}(t=0)=0$.

Aide à l'algorithme :

- pour pouvoir utiliser la méthode d'Euler, il faut voir l'équation différentielle précédente sous la forme $dv_x = \frac{k}{m}(\ell_0 - x(t))dt$;
- il va donc falloir créer 3 listes ici : t , x et v_x ;
- $v_x(t+dt) = v_x(t) + dv_x$;
- $x(t+dt) = x(t) + dx$ avec $dx = v_x(t) \times dt$ (c'est encore Euler) donc on écrit $x(t+dt) = x(t) + v_x(t) \times dt$.

3. Lancer (pour aller plus loin ...)

Un point matériel M de masse m est lancé depuis un point O du sol avec une vitesse initiale $\vec{v}_0$ dans le plan (xOy) et faisant un angle α avec l'axe horizontal (Ox) . (Oy) est verticale ascendante. On suppose que les frottements sont nuls dans un premier temps.

On rappelle les équations (qui peuvent être vues comme des équations différentielles du second ordre ...) qui régissent l'évolution temporelles de $x(t)$ et $y(t)$:
$$\begin{cases} \ddot{x} = 0 \\ \ddot{y} = -g \end{cases}$$

1. Tracer l'équation de la trajectoire $y = f(x)$ en utilisant la méthode de l'Euler.
2. On rajoute une force de frottement fluide $\vec{F}_f = -\alpha \vec{v}$ donc les équations qui régissent l'évolution

temporelle deviennent :
$$\begin{cases} \ddot{x} = -\frac{\alpha}{m}\dot{x} \\ \ddot{y} = -g - \frac{\alpha}{m}\dot{y} \end{cases}$$

Tracer l'équation de la trajectoire $y = f(x)$.