

## CN – TP 3

# Méthode d'Euler explicite

## Systeme d'équations différentielles

### Équation différentielle d'ordre 2

L'objectif de ce TP est d'étendre la méthode d'Euler explicite abordée précédemment afin de résoudre des équations différentielles qui ne se limitent pas à l'ordre 1. Plus précisément, nous généraliserons cette approche pour traiter n'importe quel système d'équations différentielles du premier ordre, ouvrant ainsi la voie à la résolution d'équations et de systèmes d'ordre 2<sup>1</sup>.

## I - Méthode d'Euler explicite : réponse d'un circuit RC à un échelon de tension

### I.1 - Rappel de la méthode d'Euler explicite

La méthode d'Euler consiste à chercher une solution approchée d'une équation différentielle d'ordre 1 en approximant la variation de la fonction entre deux instant  $t$  et  $t + \delta t$  à son accroissement selon sa dérivée :

$$y'(t) = \lim_{dt \rightarrow 0} \frac{y(t + dt) - y(t)}{dt} \approx \frac{y(t + \delta t) - y(t)}{\delta t}$$

En intégrant cette relation dans l'équation différentielle qu'on cherche à résoudre :  $y'(t) = F(y(t), t)$  on trouve une valeur approchée pour  $y(t + \delta t)$  :

$$y(t + \delta t) \approx y(t) + F(y(t), t) \times \delta t$$

En partant des conditions initiales ( $y_0 = y(t_0)$ ), on peut alors déterminer par récurrence des valeurs approchées de  $y$  aux différents instants  $t_{k+1} = t_k + \delta t$  par la relation :

$$y_{k+1} = y_k + F(y_k, t_k) \times \delta t$$

### I.2 - Exemple d'implémentation en langage Python

La fonction permettant de mettre en œuvre cet algorithme a été écrite dans le TP précédent :

```

1 def Euler(F, y0, t0, tf, dt):
2     """
3     Résout le problème de Cauchy  $y'(t)=F(y(t),t)$  avec  $y(0)=y0$  par la méthode
4     d'Euler explicite.
5     Arguments d'entrée :
6     - F : fonction donnant  $y'$  (fonction de 2 variables  $y$  et  $t$ ) ;
7     - y0 : condition initiale sur  $y$  (flottant) ;
8     - t0 et tf : bornes de l'intervalle de résolution (flottants) ;
9     - dt : pas de discrétisation utilisé pour la résolution (flottant).
10    Variables de sortie :
11    - t : vecteur contenant l'ensemble des instants  $t_k$  (array numpy) ;
12    - y : vecteur contenant l'ensemble des valeurs approchées  $y_k$  (array numpy).
13    """
14    # Initialisation des variables de sortie
15    t = np.arange(t0, tf+dt, dt) # la dernière valeur est dans [tf,tf+dt[
16    N = len(t)

```

1. On pourrait généraliser la méthode encore plus et alors résoudre une équation différentielle d'ordre quelconque, mais cela n'a pas d'application concrète, au moins en physique.

```

17  y = np.zeros(N)                # initialisation du array y
18  y[0] = y0                      # prise en compte de la CI
19
20  # Boucle permettant le calcul des yk par récurrence
21  for k in range(0,N-1):
22      y[k+1] = y[k] + F(y[k],t[k])*dt
23
24  return t, y # retourne un tableau contenant les temps tk
25              # et un tableau contenant les valeurs approchées yk
26

```

L'objectif est de généraliser cette méthode pour qu'elle s'applique non plus à une fonction scalaire  $y(t)$  mais à une fonction vectorielle  $\mathbf{r}(t)$  où  $\mathbf{r}$  est un vecteur de taille quelconque. On aura ainsi, pour un vecteur de taille 3 :

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \mathbf{F}(\mathbf{r}_k, t_k) \times \delta t \iff \begin{cases} r_{1,k+1} = r_{1,k} + F_1(\mathbf{r}_k, t_k) \times \delta t \\ r_{2,k+1} = r_{2,k} + F_2(\mathbf{r}_k, t_k) \times \delta t \\ r_{3,k+1} = r_{3,k} + F_3(\mathbf{r}_k, t_k) \times \delta t \end{cases} \quad (1.1)$$

### Manipulation 1

Reprendre la fonction `Euler` et la modifier pour qu'elle s'applique maintenant à la fonction vectorielle  $\mathbf{r}(t)$ . Les paramètres de la fonction seront inchangés, seuls le type de l'entrée  $\mathbf{r0}$  et la sortie  $\mathbf{r}$  seront modifiés :  $\mathbf{r0}$  sera une liste de taille quelconque et  $\mathbf{r}$  sera un `np.array` de dimension  $(N, R)$  où  $N$  est le nombre de points et  $R$  la taille de  $\mathbf{r0}$ .

### Manipulation 2

Reprendre le programme utilisé pour simuler la réponse d'un circuit RC à un échelon de tension par la méthode d'Euler et le modifier de façon à remplacer l'entrée scalaire  $u0$  par une entrée vectorielle  $\mathbf{r0}=[u0]$ . Le tableau de sortie sera alors un `np.array` de dimension  $(N, 1)$ . Tester le programme et vérifier qu'il donne les mêmes résultats qu'avant modification.

Pour cette partie, vous pouvez partir du script `CN-TP3-Euler-circuitRC-echelon` disponible sur le site de la PTSI et dans le répertoire `/Ressources/PTSI/PhysiqueChimie/TP3`.

La fonction `Euler` est maintenant applicable telle quelle à toute fonction vectorielle  $\mathbf{r}(t)$ , quelle que soit sa dimension. Elle est finalisée et ne nécessite plus aucune modification ; un simple copier-coller suffira pour l'intégrer dans les prochains programmes.

## II - Système d'équations différentielles d'ordre 1 – Attracteur de Lorenz

Nous allons maintenant appliquer cette méthode pour chercher une solution numérique approchée d'un « vrai » système d'équations différentielles : l'attracteur de Lorenz.

Décrit pour la première fois par Edward Lorenz en 1963, l'attracteur de Lorenz est une structure chaotique qui émerge de l'évolution d'un système dynamique tridimensionnel représentant des écoulements atmosphériques simplifiés. Il illustre un phénomène clé en théorie du chaos : la sensibilité aux conditions initiales, où de petites variations entraînent des divergences exponentielles des trajectoires. Ce comportement imprévisible, souvent comparé à l'effet papillon, a des implications profondes en météorologie, physique et ingénierie des systèmes complexes.

Le système de Lorenz est un ensemble de trois équations différentielles adimensionnées décrivant l'évolution d'un fluide soumis à des forces de convection dans un modèle simplifié de l'atmosphère. Il est défini par :

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(\rho - z) - y \\ \frac{dz}{dt} = xy - \beta z \end{cases} \quad (1.2)$$

où :

- $x$  représente l'intensité du mouvement de convection ;
- $y$ , la différence de température entre les courants ascendants ;

—  $z$  correspond à la distorsion du profil de température vertical.

Les paramètres  $\sigma$ ,  $\rho$  et  $\beta$  influencent la dynamique du système :

- $\sigma$  est le nombre de Prandtl,
- $\rho$  est lié à la force de flottabilité,
- $\beta$  reflète la proportion de dissipation thermique.

Comme on l'aura compris, il ne faut pas chercher dans cette équation différentielle une représentation fidèle du phénomène physique... Guidé par des considérations physiques, Lorenz est amené à choisir les valeurs numériques  $r = 28$ ;  $\sigma = 10$ ;  $\beta = 8/3$ ; c'était un bon choix et ces valeurs sont restées traditionnelles. Lorsque ces paramètres sont choisis de manière appropriée, le système présente une dynamique chaotique, caractérisée par l'attracteur de Lorenz, dont la forme évoque des ailes de papillon.

### Manipulation 3

Compléter le script CN-TP3-Euler-Lorenz-start en définissant les fonctions Euler (rappel, par simple copier-coller depuis le programme précédent) et Lorenz qui définit le système d'équations différentielles. Le vecteur en entrée de la fonction Euler et en sortie de la fonction Lorenz sera défini par  $\mathbf{r}=[x, y, z]$ . Vous partirez de la position initiale  $(x_0, y_0, z_0) = (1, 1, 1)$ .

Vous pourrez ensuite modifier l'angle de vue de la figure, les conditions initiales où la valeur des paramètres de l'équation pour observer les effets sur la figure tracée.

## III - Équation différentielle d'ordre 2 – Le pendule pesant

Le Botafumeiro est un encensoir en laiton argenté, suspendu à une corde sous le transept de la cathédrale de Saint-Jacques-de-Compostelle. À l'origine, cet encensoir servait à parfumer la cathédrale, il est maintenant utilisé lors de certaines cérémonies religieuses, lors desquelles il est animé par une excitation paramétrique humaine.

On peut le modéliser par une masse  $m = 54 \text{ kg}$  accrochée à l'extrémité d'un fil de longueur  $L = 21 \text{ m}$ .

### Manipulation 4

Établir l'équation du mouvement d'un pendule simple en coordonnées polaires dans le cas d'un frottement visqueux :

$$\ddot{\theta} + \frac{h}{m}\dot{\theta} + \frac{g}{L}\sin\theta = 0$$

où  $h$  est le coefficient de frottement linéaire par :

- a) la relation fondamentale de la dynamique;
- b) le théorème du moment cinétique;
- c) le théorème de la puissance cinétique.



Par Mireia Garcia Bermejo  
Travail personnel, CC BY-SA 3.0 es

<https://commons.wikimedia.org/w/index.php?curid=21026882>

Il est possible d'utiliser le schéma d'Euler pour résoudre numériquement une équation différentielle d'ordre 2 du type

$$\frac{d^2y}{dt^2} = F\left(\frac{dy}{dt}, y, t\right)$$

où l'on recherche la solution  $y(t)$  passant par les conditions initiales  $y(t_0) = y_0$  et  $\frac{dy}{dt}(t_0) = v_0$ .

Il faut pour cela définir un vecteur dont la première composante est la fonction inconnue et la deuxième sa dérivée :  $\mathbf{r} = (r_1, r_2)$  avec  $r_1 = y$  et  $r_2 = \frac{dy}{dt}$ . L'équation différentielle d'ordre deux peut alors se réécrire sous la

forme d'un système d'équations différentielles d'ordre 1 :

$$\begin{cases} \frac{dr_1}{dt} = \frac{dy}{dt} \\ \frac{dr_2}{dt} = \frac{d^2y}{dt^2} = F\left(\frac{dy}{dt}, y, t\right) \end{cases} \implies \begin{cases} \frac{dr_1}{dt} = r_2 \\ \frac{dr_2}{dt} = F(r_2, r_1, t) \end{cases} \quad (1.3)$$

Dans le cas du pendule, le vecteur  $\mathbf{r} = (r_1, r_2)$  est défini par  $r_1 = \theta$  et  $r_2 = \frac{d\theta}{dt}$  et le système d'équations différentielles est

$$\begin{cases} \frac{dr_1}{dt} = \dot{\theta} \\ \frac{dr_2}{dt} = \ddot{\theta} = -\frac{h}{m}\dot{\theta} - \frac{g}{L}\sin\theta \end{cases} \implies \boxed{\begin{cases} \frac{dr_1}{dt} = r_2 \\ \frac{dr_2}{dt} = -\frac{h}{m}r_2 - \frac{g}{L}\sin r_1 \end{cases}} \quad (1.4)$$

### Manipulation 5

Compléter le script `CN-TP3-Euler-Pendule-start` en définissant les fonctions `Euler` (rappel, par simple copier-coller depuis le programme précédent) et `Pendule` qui définit le système d'équations différentielles. Vous choisirez un pas de temps qui vous paraît raisonnable, par exemple en vous basant sur la période propre du pendule : une centaine de points par période paraît suffisant pour obtenir une approximation raisonnable du mouvement. Vous pourrez tester plusieurs conditions initiales permettant de mettre en évidence les différents mouvements possibles (mouvement pendulaire ou rotatif).

### Manipulation 6

Compléter le script précédent pour ouvrir une seconde fenêtre graphique et y tracer le portrait de phase du système. Pour rappel, le portrait de phase est le graphe paramétrique  $(\theta(t), \dot{\theta}(t))$ . En jouant sur les conditions initiales retrouver les cas du mouvement pendulaire ou rotatif.

## IV - Système d'équations différentielles d'ordre 2 – Chute libre avec frottement

Nous allons maintenant utiliser la méthode d'Euler pour déterminer le mouvement d'un objet de masse  $m$  en chute libre. Nous travaillerons dans le référentiel terrestre supposé galiléen, en choisissant un repère cartésien  $(Oxyz)$  où  $O$  sera la position initiale de l'objet, l'axe  $(Oy)$  sera vertical, dirigé vers le haut et  $(Ox)$  sera l'axe vertical contenant le vecteur vitesse initial  $\vec{v}_0$ . L'angle entre  $(Ox)$  et  $\vec{v}_0$  sera noté  $\alpha$ . L'objet est soumis à deux forces : son poids et une force de frottement quadratique de coefficient de frottement  $k$ .

### Manipulation 7

- Déterminer la dimension du coefficient  $k$ .
- Montrer que le mouvement est plan, contenu dans le plan  $(Oxy)$ .
- Démontrer que les équations du mouvements sont

$$\begin{cases} \frac{d^2x}{dt^2} = -\frac{k}{m}v \cdot v_x = -\frac{k}{m}v \cdot \frac{dx}{dt} \\ \frac{d^2y}{dt^2} = -\frac{k}{m}v \cdot v_y - g = -\frac{k}{m}v \cdot \frac{dy}{dt} - g \end{cases} \quad (1.5)$$

$$\text{avec } v = \sqrt{v_x^2 + v_y^2} = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}.$$

Pour pouvoir résoudre numériquement ce système d'équation grâce à la méthode d'Euler explicite, nous aurons besoin d'un vecteur de taille 4, contenant les coordonnées  $x$  et  $y$  ainsi que leurs dérivées par rapport au temps. Il faudra donc manipuler un vecteur  $\mathbf{r} = [x, y, v_x, v_y]$ .

### Manipulation 8

- Établir le système d'équations différentielles qui s'applique sur  $\mathbf{r}$ .
- Écrire le code python correspondant. Les paramètres du problème ( $m, g, h, v_0$  et  $\alpha$ ) seront définis au début du programme, de même que les paramètres de la résolution numérique (intervalle et pas de temps). Le programme devra alors résoudre numériquement l'équation du mouvement puis tracer la trajectoire sur un premier graphique et l'évolution temporelle de la vitesse sur un second.