Méthode d'Euler vectorielle

Informatique pour tous

Résumé de la méthode d'Euler

Si f est de classe C^1, l'équation différentielle d'ordre 1 suivante a une unique solution y avec une valeur initiale fixée $y(t_0)$:

$$y'(t) = f(t, y(t))$$

La méthode d'Euler, de pas h, consiste à approximer y par une suite récurrente $(y_k)_{0 \leq k < n}$ telle que:

1. $y_0 = y(t_0)$
2. $y_{k+1} = y_k + h \times f(t_k, y_k)$

On espère alors que $y_k \approx y(t_k)$.

Système d'équations différentielles

On veut maintenant résoudre un système d'équations différentielles sur $[a, b]$ (avec plusieurs fonctions inconnues y, z):

$$\begin{cases} y'(t) = 4y(t) - 2z(t) \\ z'(t) = y(t) + 3z(t) \end{cases}$$

Comment approximer $y(t)$ et $z(t)$?

Système d'équations différentielles

On veut maintenant résoudre un système d'équations différentielles sur $[a, b]$ (avec plusieurs fonctions inconnues y, z):

$$\left\{ \begin{array}{l} y'(t) = 4y(t) - 2z(t) \\ z'(t) = y(t) + 3z(t) \end{array} \right.$$

Comment approximer $y(t)$ et $z(t)$?

On peut appliquer la méthode d'Euler sur chaque équation, en calculant des approximations y_k et z_k ($y_k \approx y(t_k)$ et $z_k \approx z(t_k)$).

Vectorialisation

$$\begin{cases} y'(t) = 4y(t) - 2z(t) \\ z'(t) = y(t) + 3z(t) \end{cases}$$

On peut aussi **vectorialiser** en posant $Y(t) = \begin{pmatrix} y(t) \\ z(t) \end{pmatrix}$.

Vectorialisation

$$\begin{cases} y'(t) = 4y(t) - 2z(t) \\ z'(t) = y(t) + 3z(t) \end{cases}$$

On peut aussi **vectorialiser** en posant $Y(t) = \begin{pmatrix} y(t) \\ z(t) \end{pmatrix}$.

$$Y'(t) = \begin{pmatrix} y'(t) \\ z'(t) \end{pmatrix} = \begin{pmatrix} 4 & -2 \\ 1 & 3 \end{pmatrix} Y(t)$$

On est revenu à une ED du 1er ordre donc on peut appliquer la méthode d'Euler!

Méthode d'Euler **vectorielle**

Soit $F : U \subseteq \mathbb{R} \longrightarrow \mathbb{R}^n$ C^1, on considère le problème suivant, où $Y(t) \in \mathbb{R}^n$, $\forall t$:

$$\begin{cases} Y'(t) = F(t, Y(t)) \\ Y(t_0) = Y_0 \end{cases}$$

La méthode d'Euler vectorielle, de pas h, consiste à approximer la solution Y par une suite $(Y_k)_{0 \leq k < n}$ de **vecteurs** de $\mathbb{R}^n$ telle que:

1. $Y_0 = Y(t_0)$
2. $Y_{k+1} = Y_k + h \times F(t_k, Y_k)$

Vectorialisation

$$Y'(t) = \begin{pmatrix} y'(t) \\ z'(t) \end{pmatrix} = \begin{pmatrix} 4 & -2 \\ 1 & 3 \end{pmatrix} Y(t)$$

Équation de récurrence de la méthode d'Euler vectorielle, avec $Y_k = \begin{pmatrix} y_k \\ z_k \end{pmatrix}$:

$$Y_{k+1} = Y_k + h \times \begin{pmatrix} 4 & -2 \\ 1 & 3 \end{pmatrix} Y_k$$

Vectorialisation

Écrit différemment, avec $Y_k = \begin{pmatrix} y_k \\ z_k \end{pmatrix}$:

$$\begin{cases} y_{k+1} = y_k + h \times (4y_k - 2z_k) \\ z_{k+1} = z_k + h \times (y_k + 3z_k) \end{cases}$$

(ce qui revient à appliquer la méthode d'Euler sur les deux équations différentielles)

Question

En déduire un algorithme en Python pour approximer les solutions sur $[0, 1]$ (avec un pas de 0.1) de:

$$\begin{cases} y'(t) = 4y(t) - 2z(t) \\ z'(t) = y(t) + 3z(t) \\ y(0) = 3 \\ z(0) = 7 \end{cases}$$

Vectorialisation

$$\begin{cases} y_{k+1} = y_k + h \times (4y_k - 2z_k) \\ z_{k+1} = z_k + h \times (y_k + 3z_k) \end{cases}$$

Les approximations de la méthode d'Euler peuvent donc se calculer en Python (ici par exemple sur l'intervalle $[0, 1]$):

```
n = 100
h = 0.01

y = [0]
z = [2]
for k in range(n - 1):
    y.append(y[k] + h * (4*y[k] - 2*z[k]))
    z.append(z[k] + h * (y[k] + 3*z[k]))
```

Vectorialisation

$$\begin{cases} y_{k+1} = y_k + h \times (4y_k - 2z_k) \\ z_{k+1} = z_k + h \times (y_k + 3z_k) \end{cases}$$

Les approximations de la méthode d'Euler peuvent donc se calculer en Python (ici par exemple sur l'intervalle $[0, 1]$):

```
n = 100
h = 0.01

y = [0]
z = [2]
for k in range(n - 1):
    y.append(y[k] + h * (4*y[k] - 2*z[k]))
    z.append(z[k] + h * (y[k] + 3*z[k]))
```

Remarque: comme le temps n'intervient pas dans l'équation différentielle, il n'est pas nécessaire de calculer les t_k (sauf si on souhaite afficher graphiquement les solutions).

Cinétique chimique

Soient deux réactions chimiques d'ordre 1:

$$A \xrightarrow{\alpha} B$$

$$B \xrightarrow{\beta} C$$

On veut connaître les concentrations au cours du temps.

Cinétique chimique

Soient deux réactions chimiques d'ordre 1:

$$A \xrightarrow{\alpha} B$$

$$B \xrightarrow{\beta} C$$

On veut connaître les concentrations au cours du temps.

$$\left\{ \begin{array}{l} \frac{d[A]}{dt} = -\alpha[A] \\ \frac{d[B]}{dt} = \alpha[A] - \beta[B] \\ \frac{d[C]}{dt} = \beta[B] \end{array} \right.$$

Cinétique chimique

$$\begin{cases} \dfrac{d[A]}{dt} = -\alpha[A] \\ \dfrac{d[B]}{dt} = \alpha[A] - \beta[B] \\ \dfrac{d[C]}{dt} = \beta[B] \end{cases}$$

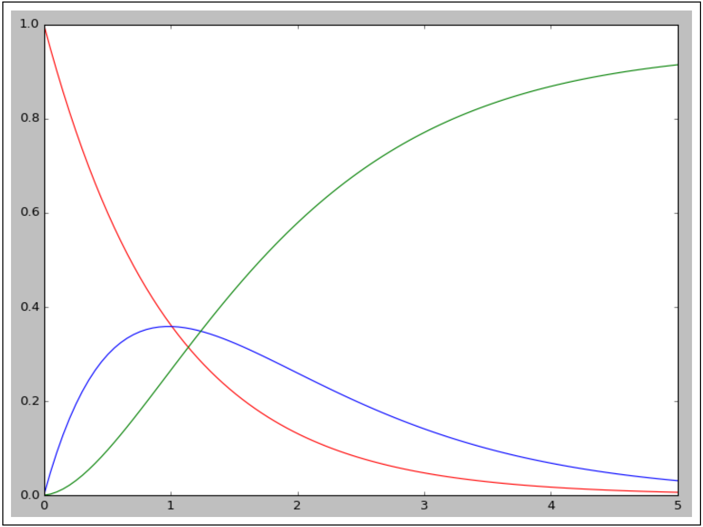
Écrire en Python la méthode d'Euler pour $t \in [0, 5]$, avec $[A]_0 = 1$ et $[B]_0 = [C]_0 = 0$.

Cinétique chimique

$$\begin{cases} \dfrac{d[A]}{dt} = -\alpha[A] \\ \dfrac{d[B]}{dt} = \alpha[A] - \beta[B] \\ \dfrac{d[C]}{dt} = \beta[B] \end{cases}$$

Écrire en Python la méthode d'Euler pour $t \in [0, 5]$, avec $[A]_0 = 1$ et $[B]_0 = [C]_0 = 0$.
Vérifier avec le fichier Python sur hugoprépa.

Cinétique chimique



Equation différentielle d'ordre 2

Comment appliquer la méthode d'Euler sur une équation différentielle d'ordre ≥ 2?
Par exemple (équation du pendule linéarisé):

$$\theta''(t) = -\theta(t)$$

Equation différentielle d'ordre 2

Comment appliquer la méthode d'Euler sur une équation différentielle d'ordre ≥ 2?
Par exemple (équation du pendule linéarisé):

$$\theta''(t) = -\theta(t)$$

On pose $z(t) = \theta'(t)$.

Equation différentielle d'ordre 2

Comment appliquer la méthode d'Euler sur une équation différentielle d'ordre ≥ 2?
Par exemple (équation du pendule linéarisé):

$$\theta''(t) = -\theta(t)$$

On pose $z(t) = \theta'(t)$.
On a alors $z'(t) = \theta''(t) = -\theta(t)$

Equation différentielle d'ordre 2

Comment appliquer la méthode d'Euler sur une équation différentielle d'ordre ≥ 2?
Par exemple (équation du pendule linéarisé):

$$\theta''(t) = -\theta(t)$$

On pose $z(t) = \theta'(t)$.
On a alors $z'(t) = \theta''(t) = -\theta(t)$
On obtient donc un système d'équa. diff. linéaires d'ordre 1:

$$\left\{ \begin{array}{l} \theta'(t) = z(t) \\ z'(t) = -\theta(t) \end{array} \right.$$

Equation différentielle d'ordre 2

Comment appliquer la méthode d'Euler sur une équation différentielle d'ordre ≥ 2?
Par exemple (équation du pendule linéarisé):

$$\theta''(t) = -\theta(t)$$

On pose $z(t) = \theta'(t)$.
On a alors $z'(t) = \theta''(t) = -\theta(t)$
On obtient donc un système d'équa. diff. linéaires d'ordre 1:

$$\begin{cases} \theta'(t) = z(t) \\ z'(t) = -\theta(t) \end{cases}$$

On peut alors appliquer la méthode d'Euler sur chaque équation pour obtenir des approximations $\theta_k \approx \theta(t_k)$ et $z_k \approx \theta'(t_k)$.

Équation différentielle d'ordre 2

$$\theta''(t) = -\theta(t)$$

On peut aussi vectorialiser en posant $\Theta(t) = \begin{pmatrix} \theta(t) \\ \theta'(t) \end{pmatrix}$.

Équation différentielle d'ordre 2

$$\theta''(t) = -\theta(t)$$

On peut aussi vectorialiser en posant $\Theta(t) = \begin{pmatrix} \theta(t) \\ \theta'(t) \end{pmatrix}$.

Alors $\Theta'(t) = \begin{pmatrix} \theta'(t) \\ \theta''(t) \end{pmatrix} = \begin{pmatrix} \theta'(t) \\ -\theta(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \theta(t) \\ \theta'(t) \end{pmatrix}$.

On s'est ramené à une équa diff du 1er ordre: $\Theta'(t) = A\ \Theta(t)$.

Equation différentielle d'ordre 2

$$\Theta'(t) = \begin{pmatrix} \theta'(t) \\ \theta''(t) \end{pmatrix} = \begin{pmatrix} \theta'(t) \\ -\theta(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} \theta(t) \\ \theta'(t) \end{pmatrix}.$$

On s'est ramené à une ED $\Theta'(t) = A\,\Theta(t)$.

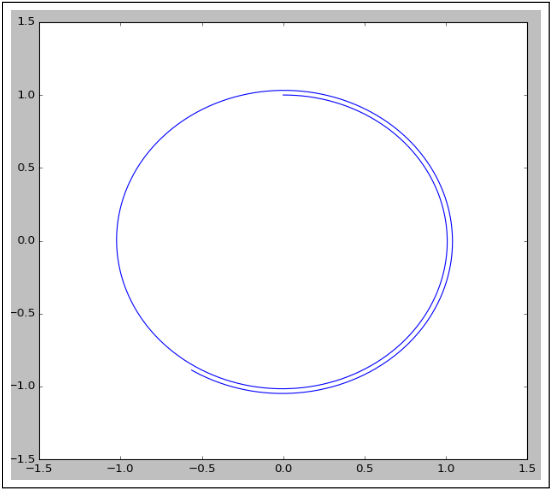
Les approximations de la méthode d'Euler sont donc des vecteurs Θ_k vérifiant:

$$\Theta_{k+1} = \Theta_k + h \times \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \Theta_k$$

Pendule linéarisé

```
Theta = [0]
Theta_p = [1]
for k in range(999):
    Theta.append(Theta[k] + 0.01*Theta_p[k])
    Theta_p.append(Theta_p[k] - 0.01*Theta[k])
plt.plot(Theta, Theta_p)
plt.show()
```

Portrait de phase du pendule linéarisé



Pendule non linéarisé

Équation du pendule non linéarisé:

$$\theta''(t) = -\sin(\theta(t))$$

Pendule non linéarisé

Équation du pendule non linéarisé:

$$\theta''(t) = -\sin(\theta(t))$$

On pose $\Theta(t) = \begin{pmatrix} \theta(t) \\ \theta'(t) \end{pmatrix}$.
Alors $\Theta'(t) = \begin{pmatrix} \theta'(t) \\ \theta''(t) \end{pmatrix} = \begin{pmatrix} \theta'(t) \\ -\sin(\theta(t)) \end{pmatrix}$.

Pendule non linéarisé

```
Theta = [0]
Theta_p = [2]
for k in range(9999):
    Theta.append(Theta[k] + 0.01*Theta_p[k])
    Theta_p.append(Theta_p[k] - 0.01*np.sin(Theta[k]))
plt.plot(Theta, Theta_p)
plt.show()
```

Pendule non linéarisé