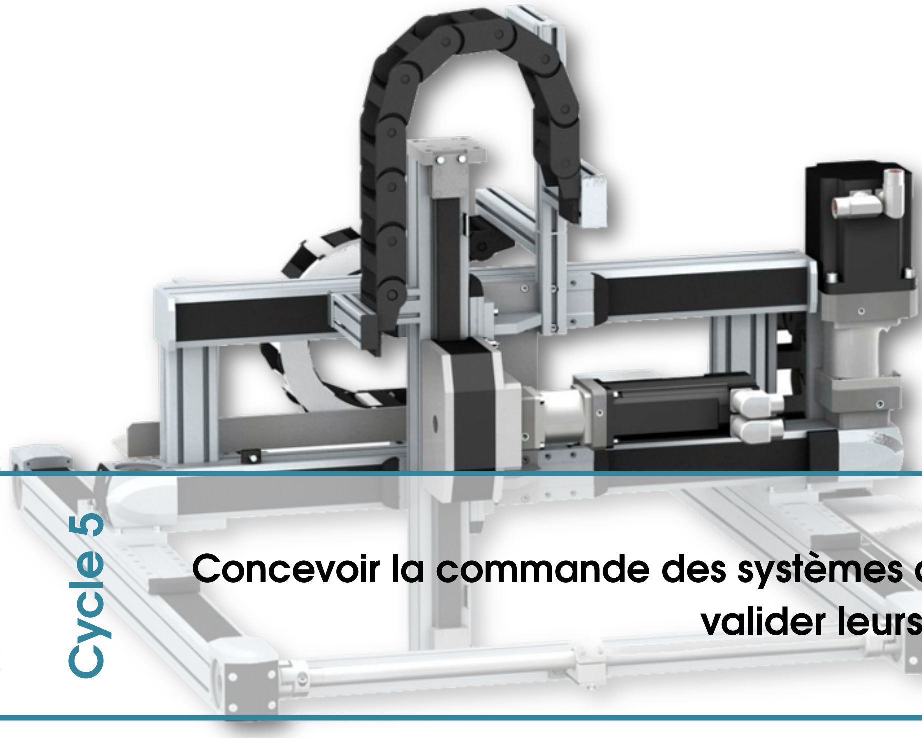
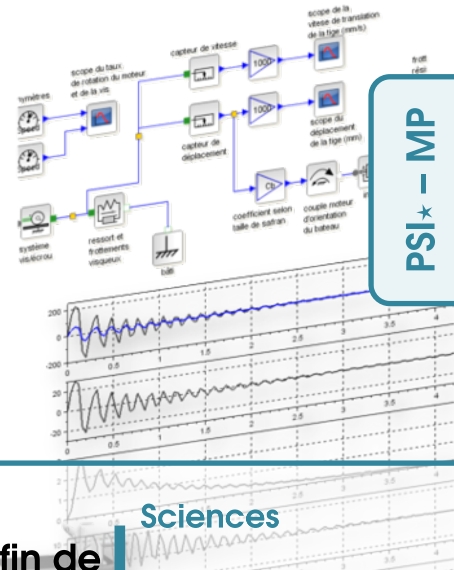




PSI⋆ – MP

Cycle 5

Concevoir la commande des systèmes asservis afin de valider leurs performances

Sciences Industrielles de l'Ingénieur

Cours

Chapitre 1

Introduction aux méthodes numériques

Savoirs et compétences :

- *B2-12 : proposer un modèle cinématique à partir d'un système réel ou d'une maquette numérique ;*
- *B2-15 : Simplifier un modèle de mécanisme.*

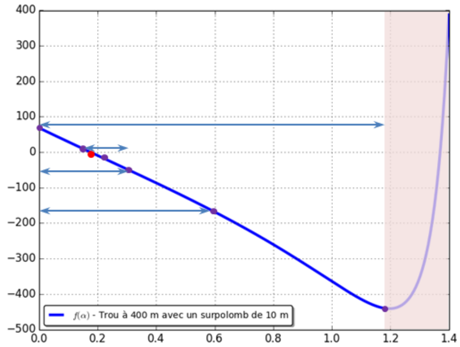
Xavier Pessoles
PSI⋆ – MP

1 Equations stationnaires – Résolution de $f(x)=0$

1.1 Principe de la méthode de dichotomie

Théorème **Théorème des valeurs intermédiaires**
Soit f une fonction définie et continue sur l'intervalle $[a,b]$ à valeur dans $\mathbb{R}$. Pour tout $u \in [f(a), f(b)]$, il existe au moins un réel $c \in [a,b]$ tel que $f(c)=u$.
En particulier (Théorème de Bolzano), si $f(a)$ et $f(b)$ sont de signes différents, il existe au moins un réel c tel que $f(c)=0$.

Ainsi, pour une fonction donnée définie sur un intervalle donné, le but de l'algorithme de dichotomie va être de découper en 2 l'intervalle [a,b] en deux, afin d'y trouver la solution. Par divisions successives de l'intervalle, on convergera vers la solution.



R **Tester le signe de $f(a)$ et $f(b)$.**
Il existe plusieurs méthodes pour tester si $f(a)$ et $f(b)$ sont de signes différents. Si on ne se préoccupe pas de savoir la relation d'ordre entre $f(a)$ et $f(b)$, un test efficace consiste en un test du signe de $f(a)\cdot f(b)$.

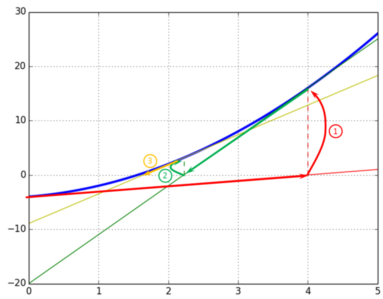
1.2 Principe de la méthode de Newton

Théorème **Développement de Taylor à l'ordre 1**
Soit f une fonction C^1 sur un intervalle I et $a \in I$. Le développement de Taylor à l'ordre 1 de f est donné par

$$f(x)=f(a)+f'(a)\cdot(x-a)+o(x-a)$$

Géométriquement, lorsqu'on néglige le reste, le développement de Taylor donne l'équation de la tangente en a. Notons $\Delta(x)$ cette équation.
L'abscisse c de l'intersection de la tangente avec l'axe des abscisses est donnée par la résolution de

$$\Delta(c)=0 \Longleftrightarrow f(a)+f'(a)\cdot(c-a)=0 \Longleftrightarrow c=a-\frac{f(a)}{f'(a)}$$



Évaluation de la dérivée numérique

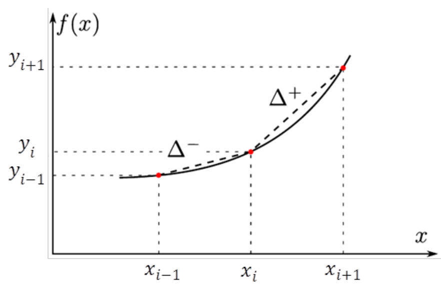
Résultat En première approximation, il est possible d'approximer la dérivée en approximant la tangente à la courbe par une droite passant par deux points successifs. Dans ces conditions, pour une valeur de h suffisamment faible, on a :

$$f'(x_0)\simeq \frac{f(x_0+h)-f(x_0)}{h}.$$

1.2.1 Méthodes à un pas

Résultat **Différence avant – Schéma d'Euler explicite**
Dans ce cas, l'estimation de la dérivée au point P_i s'appuie sur le point P_{i+1} :

$$f'(x_i)\simeq \frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i}$$



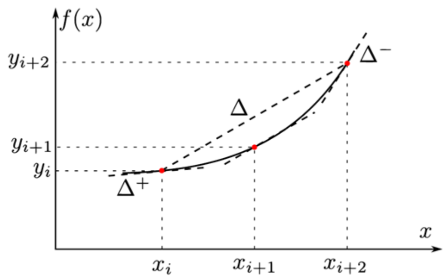
Résultat **Différence arrière – Schéma d'Euler implicite**
Dans ce cas, l'estimation de la dérivée au point P_i s'appuie sur le point P_{i-1} :

$$f'(x_i)\simeq \frac{f(x_i)-f(x_{i-1})}{x_i-x_{i-1}}$$

1.2.2 Méthode à deux pas

Résultat On peut aussi utiliser les points P_{i-1} et P_{i+1} pour estimer la dérivée en P_i :

$$f'(x_i) \simeq \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}}$$



R

- Lorsqu'il s'agit de dériver une fonction temporelle « en temps réel », le point suivant n'est pas encore connu donc seule la différence arrière peut être calculée.
- Le calcul de la dérivée conduit à un tableau de valeurs de dimension $n-1$.

1.3 Bibliothèque Python

Il est possible de résoudre l'équation $f(x) = 0$ en utilisant les modules de la bibliothèque `scipy` :

Python

Résolution de $\sin(x) = 0$ avec 0,5 comme valeur d'initialisation.

```
def f(x):
    return sin(x)

sol = newton(f,0.5)
print(sol)
print(f(sol))
```

Résolution du système :

$$\begin{cases} x + 10y - 3z - 5 = 0 \\ 2x - y + 2z - 2 = 0 \\ -x + y + z + 3 = 0 \end{cases}$$

```
from scipy.optimize import fsolve
# définition du système
def syst(var):
    # définition des variables
    x, y, z = var[0], var[1], var[2]
    eq1 = x +10*y-3*z-5
    eq2 = 2*x-y+2*z-2
    eq3 = -x+y+z+3
    res = [eq1, eq2, eq3]
    return res
    # Initialisation de la recherche
    # des solutions numériques
x0, y0, z0 = 0, 0, 0
sol_ini = [x0, y0, z0]
sol = fsolve(syst, sol_ini)
sol = newton(f,0.5)
print(sol)
```

2 Intégration numérique

Hypothèse(s) $f : [a, b] \to \mathbb{R}$ est une fonction continue sur $[a, b]$. On note $I = \int_a^b f(x)\mathrm{d}x$.

2.1 Principe des méthodes des rectangles

Définition Méthode des rectangles Dans cette méthode, la fonction à intégrer est interpolée par un polynôme de degré 0, à savoir une fonction constante. Géométriquement, l'aire sous la courbe est alors approximée par un rectangle. Plusieurs choix sont possibles.

Rectangles à gauche :

$$I = \int_a^b f(x)\mathrm{d}x \simeq (b-a) f(a)$$

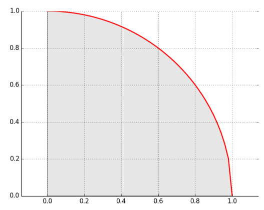
Point milieu :

$$I = \int_a^b f(x)\mathrm{d}x \simeq (b-a) f\left(\frac{a+b}{2}\right)$$

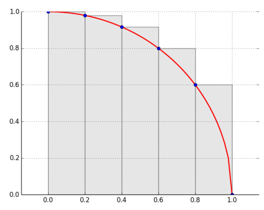
Rectangles à droite :

$$I = \int_a^b f(x)\mathrm{d}x \simeq (b-a) f(b)$$

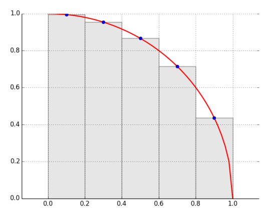
2.2 Interprétation graphique



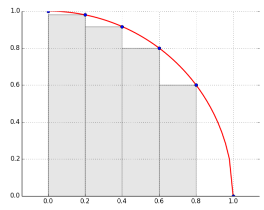
Calcul intégral



Rectangles à gauche



Point milieu

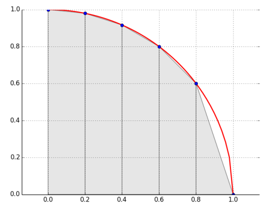


Rectangles à droite

2.3 Principe des méthodes des trapèzes

Définition Méthode des trapèzes Dans cette méthode, la fonction à intégrer est interpolée par un polynôme de degré 1, à savoir une fonction affine. Géométriquement, l'aire sous la courbe est alors approximée par un trapèze :

$$I = \int_a^b f(x)\mathrm{d}x \simeq (b-a)\frac{f(a)+f(b)}{2}$$



Notion d'erreur d'intégration

Résultat Dans chaque cas, on intègre f sur n subdivisions régulières de I.

Erreur sur la méthode des rectangles à gauche et à droite

Soit f fonction dérivable sur $I=[a,b]$ et dont f' est continue sur I. Soit M_1 un majorant de f' sur I. L'erreur ε commise lors de l'intégration par la méthode des rectangles à droite ou à gauche est telle que $\varepsilon \leq \dfrac{M_1}{2n}$.

Erreur sur la méthode des rectangles – point milieu

Si de plus f est deux fois dérivables sur $I=[a,b]$ et f'' est continue sur I, on note M_2 un majorant de f'' sur I.L'erreur ε commise lors de l'intégration par la méthode des rectangles – point milieu est telle que $\varepsilon \leq \dfrac{M_2}{12n^2}$.

Erreur sur la méthode des trapèzes

L'erreur commise ε est telle qu'il existe un entier M tel que $\varepsilon \leq \dfrac{M}{12n^2}$.

Bibliothèque Python

Il est possible d'intégrer une fonction en utilisant les modules de la bibliothèque `scipy` :

```python
from scipy.integrate import quad
from math import sin
# Définition des bornes de gauche et de droite
g,d = -1,1
def f(x):
    return sin(x)

I,erreur = quad(f,g,d)
print(I,erreur)
```

3 Résolution d'équations différentielles

3.1 Problème de Cauchy

Rappel Problème de Cauchy

Le problème consiste à trouver les fonctions y de $[0,T]\rightarrow \mathbb{R}^n$ telles que

$$\left\{\begin{array}{l} \dfrac{\mathrm{d}y(t)}{\mathrm{d}t} = f(t,y(t)) \\ y(t_0)=y_0 \quad \text{avec } t_0\in[0,T] \text{ et } y_0\in\mathbb{R}^n \text{ donnés} \end{array}\right. \qquad \text{avec } t_0\in[0,T] \text{ et } y_0\in\mathbb{R}^n \text{ donnés.}$$

L'existence et l'unicité de la solution peut se démontrer en utilisant le théorème de Cauchy-Lipschitz.

Définition Fonction lipschitzienne

f est lipschitzienne en y s'il existe un réel $k>0$ tel que $\forall y(t)\in\mathbb{R}^n$, $\forall z(t)\in\mathbb{R}^n$, $\forall t\in[0,T]$, alors

$$||f(t,y(t))-f(t,z(t))||\leq k||y(t)-z(t)||$$

Théorème — Théorème de Cauchy – Lipschitz
Soit f une fonction de $[0,T]\times\mathbb{R}^n\to\mathbb{R}^n$ continue et lipschitzienne en y.
Alors, $\forall t_0\in[0,T]$ et $\forall y_0\in\mathbb{R}^n$, le problème de Cauchy admet une unique solution définie sur $[0,T]$.

3.2 Méthode d'Euler

Pour un temps de simulation compris entre t_0 et t_1, si on choisi un nombre n d'échantillons, alors le pas d'intégration est défini par $h=\dfrac{t_1-t_0}{n}$. (On a donc $t_i=t_0+h\cdot i$ avec $i\in[0,n]$.)

Résultat En intégrant l'équation du problème de Cauchy sur un intervalle $[t_i,t_{i+1}]$, on a :

$$\int_{y_i}^{y_{i+1}}\mathrm{d}y=\int_{t_i}^{t_{i+1}}f(t,y(t))\mathrm{d}t\Longleftrightarrow y_{i+1}-y_i=\int_{t_i}^{t_{i+1}}f(t,y(t))\mathrm{d}t\text{ et donc : }y_{i+1}=y_i+\int_{t_i}^{t_{i+1}}f(t,y(t))\mathrm{d}t$$

R En utilisant la méthode des rectangles à gauche, $\int_{t_i}^{t_{i+1}}f(t,y(t))\mathrm{d}t\simeq h\cdot f(t_i,y(t_i))$.

3.3 Méthode d'Euler explicite

À l'instant i, $\dfrac{\mathrm{d}y(t)}{\mathrm{d}t}$ peut être approximé par $\dfrac{\mathrm{d}y(t_i)}{\mathrm{d}t}\simeq\dfrac{y_{i+1}-y_i}{h}$. Ainsi, $y_{i+1}=y_i+hf(t_i,y_i)$.

Méthode d'Euler implicite

À l'instant i, $\dfrac{\mathrm{d}y(t)}{\mathrm{d}t}$ peut être approximé par $\dfrac{\mathrm{d}y(t_i)}{\mathrm{d}t}\simeq\dfrac{y_i-y_{i-1}}{h}$. Ainsi, $y_i=y_{i-1}+hf(t_{i-1},y_i)$ ou encore $y_{i+1}=y_i+hf(t_i,y_{i+1})$.

Bibliothèque Python

Voir exemples : `http://python.physique.free.fr/outils_math.html`.

3.4 Exemples

Reformulation d'équations différentielles en vue de leur résolution numérique.

- Équation différentielle du premier ordre à coefficients constants : $\omega(t)+\tau\dfrac{\mathrm{d}\omega(t)}{\mathrm{d}t}=\omega_c$:
  - schéma d'Euler explicite : on a $\dfrac{\mathrm{d}\omega(t)}{\mathrm{d}t}\simeq\dfrac{\omega_{k+1}-\omega_k}{h}\Rightarrow\omega_k+\tau\dfrac{\omega_{k+1}-\omega_k}{h}=\omega_c\Leftrightarrow\omega_{k+1}=\dfrac{h}{\tau}(\omega_c-\omega_k)+\omega_k$;
  - schéma d'Euler implicite : on a $\dfrac{\mathrm{d}\omega(t)}{\mathrm{d}t}\simeq\dfrac{\omega_k-\omega_{k-1}}{h}\Rightarrow\omega_k+\tau\dfrac{\omega_k-\omega_{k-1}}{h}=\omega_c\Leftrightarrow\omega_k=\dfrac{h\omega_c+\tau\omega_{k-1}}{h+\tau}$.
- Équation différentielle du premier ordre à coefficients non constants : $\omega(t)f(t)+\tau\dfrac{\mathrm{d}\omega(t)}{\mathrm{d}t}g(t)=\omega_c h(t)$:
  - Euler explicite : $\dfrac{\mathrm{d}\omega(t)}{\mathrm{d}t}\simeq\dfrac{\omega_{k+1}-\omega_k}{h}\Rightarrow\omega_k f_k+\tau\dfrac{\omega_{k+1}-\omega_k}{h}g_k=\omega_c h_k\Leftrightarrow\omega_{k+1}=\dfrac{h}{\tau g_k}(\omega_c h_k-\omega_k f_k)+\omega_k$;
  - Euler implicite : $\dfrac{\mathrm{d}\omega(t)}{\mathrm{d}t}\simeq\dfrac{\omega_k-\omega_{k-1}}{h}\Rightarrow\omega_k f_k+\tau\dfrac{\omega_k-\omega_{k-1}}{h}g_k=\omega_c h_k\Leftrightarrow\omega_k=\dfrac{h\omega_c h_k+\tau g_k\omega_{k-1}}{f_k h+\tau g_k}$.
- Équation différentielle du premier ordre $\sin(\omega(t))+\dfrac{\mathrm{d}\omega(t)}{\mathrm{d}t}=K$:
  - Euler explicite : $\dfrac{\mathrm{d}\omega(t)}{\mathrm{d}t}\simeq\dfrac{\omega_{k+1}-\omega_k}{h}\Rightarrow\sin\omega_k+\dfrac{\omega_{k+1}-\omega_k}{h}=K\Leftrightarrow\omega_{k+1}=h(K-\sin\omega_k)+\omega_k$;
  - Euler implicite : $\dfrac{\mathrm{d}\omega(t)}{\mathrm{d}t}\simeq\dfrac{\omega_k-\omega_{k-1}}{h}\Rightarrow\sin\omega_k+\dfrac{\omega_k-\omega_{k-1}}{h}=K\Leftrightarrow h\sin\omega_k+\omega_k-\omega_{k-1}=hK$. Dans ce cas, il faut utiliser la méthode de Newton ou de dichotomie pour déterminer ω_k.
- Équation différentielle du second ordre : $\ddot{s}(t)+\dfrac{2\xi}{\omega_0}\dot{s}(t)+\omega_0^2 s(t)=Ke(t)$. On pose :

$$\left\{\begin{array}{l}y_1(t)=s(t)\\ y_2(t)=\dot{s}(t)\end{array}\right.\Rightarrow\left\{\begin{array}{l}\dot{y}_1(t)=\dot{s}(t)=y_2(t)\\ \dot{y}_2(t)=\ddot{s}(t)\end{array}\right.\Rightarrow\left\{\begin{array}{l}\dot{y}_1(t)=y_2(t)\\ \dot{y}_2(t)+\dfrac{2\xi}{\omega_0}y_2(t)+\omega_0^2y_1(t)=Ke(t)\end{array}\right.$$

Schéma d'Euler explicite : $\dfrac{\mathrm{d}y(t)}{\mathrm{d}t} \simeq \dfrac{y_{k+1}-y_k}{h}$. On a donc :

$$\left\{\begin{array}{l}\dfrac{y_{1,k+1}-y_{1,k}}{h} = y_{2,k}\\ \dfrac{y_{2,k+1}-y_{2,k}}{h} + \dfrac{2\xi}{\omega_0} y_{2,k} + \omega_0^2 y_{1,k} = K e_k\end{array}\right.$$

$$\Leftrightarrow \left\{\begin{array}{l} y_{1,k+1} = h\, y_{2,k} + y_{1,k}\\ y_{2,k+1} = h K e_k - \dfrac{2\xi h}{\omega_0} y_{2,k} - h\omega_0^2 y_{1,k} + y_{2,k}\end{array}\right.$$

Schéma d'Euler implicite : $\dfrac{\mathrm{d}y(t)}{\mathrm{d}t} \simeq \dfrac{y_k-y_{k-1}}{h}$. On a donc :

$$\left\{\begin{array}{l}\dfrac{y_{1,k}-y_{1,k-1}}{h} = y_{2,k}\\ \dfrac{y_{2,k}-y_{2,k-1}}{h} + \dfrac{2\xi}{\omega_0} y_{2,k} + \omega_0^2 y_{1,k} = K e_k\end{array}\right.$$

$$\Leftrightarrow \left\{\begin{array}{l} y_{1,k} = h\, y_{2,k} + y_{1,k-1}\\ y_{2,k} = \dfrac{h K e_k + y_{2,k-1} - h\omega_0^2 y_{1,k-1}}{1 + h\dfrac{2\xi}{\omega_0} + \omega_0^2 h^2}\end{array}\right.$$

❒ Équation différentielle du second ordre : $\ddot{\theta}(t) + k\sin\theta(t) = 0$. On pose :

$$\left\{\begin{array}{l} y_0(t) = \theta(t)\\ y_1(t) = \dot{\theta}(t)\end{array}\right. \Rightarrow \left\{\begin{array}{l} \dot{y}_0(t) = \dot{\theta}(t) = y_1(t)\\ \dot{y}_1(t) = \ddot{\theta}(t)\end{array}\right. \Rightarrow \left\{\begin{array}{l} \dot{y}_0(t) = y_1(t)\\ \dot{y}_1(t) + k\sin y_0(t) = 0\end{array}\right.$$

Schéma d'Euler explicite :

$$\left\{\begin{array}{l}\dfrac{y_{0,k+1}-y_{0,k}}{h} = y_{1,k}\\ \dfrac{y_{1,k+1}-y_{1,k}}{h} + k\sin y_{0,k} = 0\end{array}\right. \Leftrightarrow \left\{\begin{array}{l} y_{0,k+1} = h\, y_{1,k} + y_{0,k}\\ y_{1,k+1} = y_{1,k} - k h \sin y_{0,k}\end{array}\right.$$

Schéma d'Euler implicite :

$$\left\{\begin{array}{l}\dfrac{y_{0,k}-y_{0,k-1}}{h} = y_{1,k}\\ \dfrac{y_{1,k}-y_{1,k-1}}{h} + k\sin y_{0,k} = 0\end{array}\right. \Leftrightarrow \left\{\begin{array}{l} y_{0,k} = h\, y_{1,k} + y_{0,k-1}\\ y_{1,k} = y_{1,k-1} - k h \sin y_{0,k}\end{array}\right.$$

❒ Équation différentielle du second ordre : $(k_1 + k_2 \sin^2\theta(t))\ddot{\theta}(t) + (k_3\dot{\theta}(t)^2 + k_4)\sin 2\theta(t) + C(t) = 0$.

3.5 Bibliothèque Python

4 Résolution de systèmes linéaires