4 Équ différentielles ordinaires (EDO, anglais : ODE)

I exemples :

$\ddot{x} = -g$ chute libre

$\ddot{x} = -\sin x$ pendule

$\ddot{\vec{r}}_i = \sum_j \frac{G m_j}{r_{ij}^3} (\vec{r}_j - \vec{r}_i)$ N corps en interaction gravitationnelle

$\dot{U} = -\frac{1}{RC} U$ décharge condensateur

$\left\{ \begin{array}{l} \dot{H} = rH - aHP \\ \dot{P} = bHP - mP \end{array} \right\}$ modèle proie(H) - prédateur(P) de Lotka - Volterra

on cherche $x(t), \vec{r}_i(t), U(t), H(t), P(t), \ldots$

bien posé ? non : manque condit° initiale ou aux bords !

II idée

résolut° numérique : idée $U(t) = U(0) + \dot{U}(0)\,dt + O(dt^2)$

[EDO $\dot{U} = f(U,t)$ nous donne $\dot{U}(0)$]

$\rightsquigarrow$ discrétisat° du temps $t_n = n \cdot h$, h : pas de temps, $n = 0, 1, 2, \ldots$

\+ ~~résolution~~ + intégrateur $U(t_{n+1}) = F(U(t_n), t_n)$

$\Rightarrow$ construct° d'une suite $U_n = U(t_n)$, $n = 0, 1, 2, \ldots$

par l'algorithme d'Euler :

initialisat° : $n = 0$, h, t_0, U_0

↓

calculer $U_{n+1} = F(U_n, t_n)$ $t_{n+1} = t_n + h$

↓

$n \leftarrow n+1$

La suite U_n, $n = 0, 1, 2, \ldots$ sera un approximat° de la fonction $U(t)$

remarques :

* algorithme fonctionne aussi pour une collection de fonctions $\vec{U} = \begin{pmatrix} U^{(1)} \\ U^{(2)} \\ \vdots \\ U^{(m)} \end{pmatrix}$, $\dot{\vec{U}} = f(\vec{U}, t)$

ex: $\ddot{\vec{r}}_i = \sum_j \frac{G m_j}{r_{ij}} (\vec{r}_j - \vec{r}_i)$ N corps, $\rightsquigarrow \vec{U} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ \vdots \end{pmatrix} \begin{array}{l} \}\, \vec{r}_1 \\ \\ \\ \}\, \vec{r}_2 \\ \\ \vdots \end{array}$

ou $\left. \begin{array}{l} \dot{H} = rH - aHP \\ \dot{P} = bHP - mP \end{array} \right\} \Rightarrow \vec{U} = \begin{pmatrix} H \\ P \end{pmatrix}$

$\Rightarrow$ pas de différence entre équ. simple et système d'équations !

* equ du second ordre → système d'équ. d'ordre 1

ex : $\ddot{x} = -\frac{k}{m}x$ (masse - ressort)

introduire une deuxième quantité à intégrer : la vitesse $v(t)$

⇒ équ d'évolution pour x et v : $\begin{aligned}\dot{x} &= v \\ \dot{v}(=\ddot{x}) &= -\frac{k}{m}x\end{aligned}$ EDO d'ordre 1

* généralement :

- ordre 2 : $\ddot{x} = f(x, \dot{x}, t)$ introduire $v = \dot{x}$

$$\Rightarrow \quad \begin{aligned}\dot{x} &= v \\ \dot{v} &= f(x, v, t)\end{aligned}$$

- ordre n : $x^{(n)} = f(x, x', x'', \ldots, x^{(n-1)}, t)$ introduire $\begin{aligned}u_1 &= x' \\ u_2 &= x'' \\ &\vdots \\ u_{n-1} &= x^{(n-1)}\end{aligned}$

$$\Rightarrow \quad \begin{aligned}x' &= u_1 \\ u_1' &= u_2 \\ u_2' &= u_3 \\ &\vdots \\ u_{n-1}' &= f(x, u_1, u_2, \ldots u_{n-1}, t)\end{aligned}$$

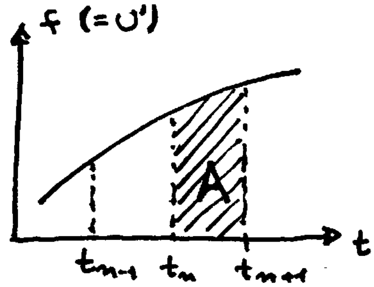
- rendre équ autonome : $\dot{\vec{x}} = \vec{f}(\vec{x}, t)$ introduire $\vec{U} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ t \end{pmatrix} = \begin{pmatrix} \vec{x} \\ t \end{pmatrix}$

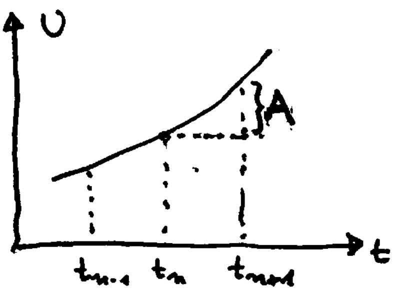
$$\leadsto \quad \frac{d}{dt}\vec{U} = \begin{pmatrix} \frac{d}{dt}\vec{x} \\ \frac{dt}{dt} \end{pmatrix} = \begin{pmatrix} \vec{f} \\ 1 \end{pmatrix} = \vec{g}(\vec{U})$$

↑ ne dépend plus de t explicitement : syst autonome

III intégrateurs explicites

solution exacte : $U_{n+1} = U_n + \int_{t_n}^{t_{n+1}} f(U(t), t)\, dt$





① Euler : $A \simeq h \cdot f(U_n, t_n)$ (dévelop^t linéaire ⇒ erreur $O(h^2)$)

② Euler modifié : $A \simeq h \cdot f(U(t_n + \frac{h}{2}), t_n + \frac{h}{2})$

utiliser ① pour calculer $U(t_n + \frac{h}{2}) \simeq U_n + \frac{h}{2} \cdot f(U_n, t_n)$

plus précis !

preuve :

$$U(t + h) = U(t + \tfrac{h}{2}) + \tfrac{h}{2} U'(t + \tfrac{h}{2}) + (\tfrac{h}{2})^2 \frac{U''(t+\frac{h}{2})}{2} + O(h^3)$$

$$U(t) = U(t + \tfrac{h}{2}) - \tfrac{h}{2} U'(t + \tfrac{h}{2}) + (\tfrac{h}{2})^2 \frac{U''(t+\frac{h}{2})}{2} + O(h^3)$$

$$\Rightarrow U(t+h) - U(t) = h \cdot U'(t + \tfrac{h}{2}) + O(h^3)$$

$$= h f(\underbrace{U(t+\tfrac{h}{2})}_{\simeq U(t) + \frac{h}{2} U'(t) + O(h^2)}, t + \tfrac{h}{2}) + O(h^3)$$

$$= h f(U(t) + \tfrac{h}{2} f(U(t), t) + O(h^2), t + \tfrac{h}{2}) + O(h^3)$$

$$\Rightarrow O(h^3)$$

$$= h f(U(t) + \tfrac{h}{2} f(U(t), t), t + \tfrac{h}{2}) + O(h^3)$$

③ ~~(beaucoup d'intégrateurs différents)~~

Runge - Kutta 4 :

$$k_1 = h f(U_n, t_n)$$

$$k_2 = h f(U_n + \tfrac{k_1}{2}, t_n + \tfrac{h}{2})$$

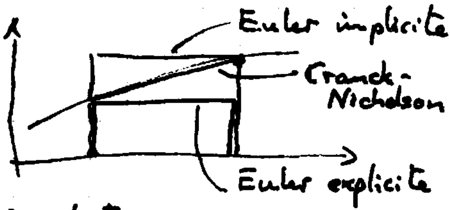
$$k_3 = h f(U_n + \tfrac{k_2}{2}, t_n + \tfrac{h}{2})$$

$$k_4 = h f(U_n + k_3, t_n + h)$$

$$U_{n+1} = \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

précision : $\simeq O(h^4)$!

④ .... (beaucoup d'intégrateurs différents)



IV intégrateurs implicites

① $A \simeq h\, f(U_{n+1}, t_{n+1})$ Euler implicite

② $A \simeq \frac{1}{2} h\, (f(U_n, t_n) + f(U_{n+1}, t_{n+1}))$ Cranck-Nicholson ③ ...

m̂ précision que Euler explicite, mais plus stable

inconvénient : $U_{n+1} = U_n + h f(U_{n+1}, t_{n+1})$ (beaucoup) plus difficile à résoudre

convient quand schémas explicites ont problème de stabilité

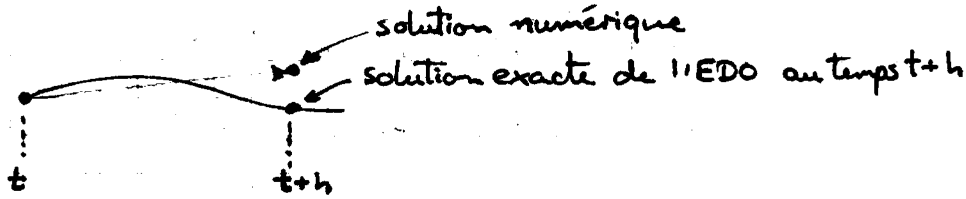
V stabilité

3 critères pour choix d'un intégrateur :

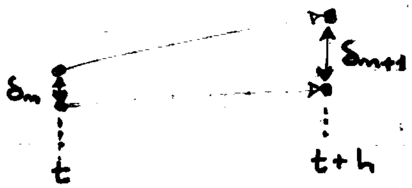
- précision (fonction du pas de temps) c.f. plus haut [Euler : Ordre 1, R-K4 : Ordre 4, ...]
- stabilité (dépend du schéma d'intégration et des équ. à intégrer)
- complexité de calcul [moins important]

ne pas confondre précision et stabilité :

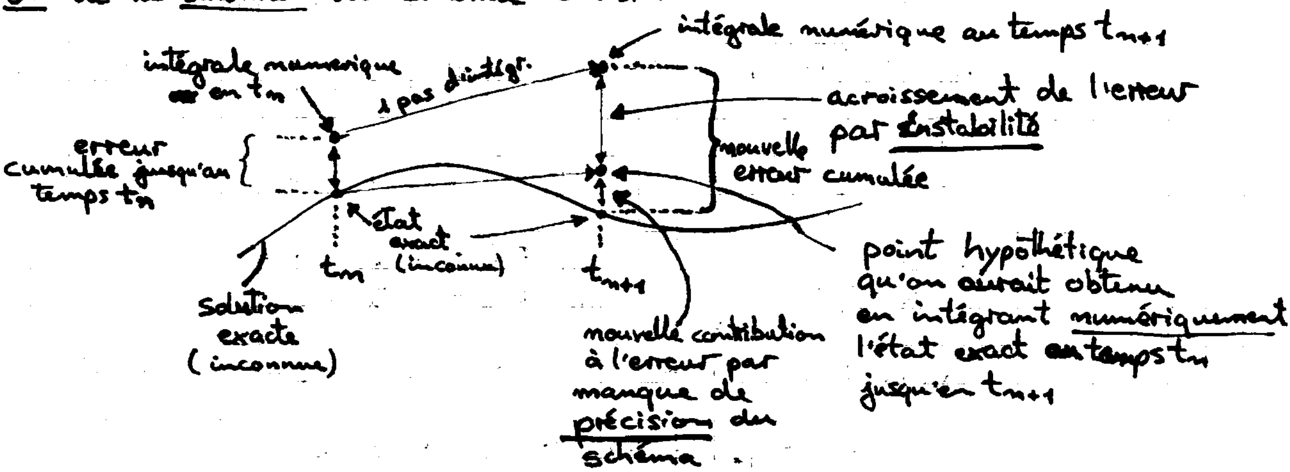
__précision__ : partant d'un état initial exact, de combien s'en éloigne-t-on en 1 pas de temps ?



__stabilité__ : ~~on~~ l'écart entre 2 états proches se creuse-t-il sous l'effet de l'intégration numérique ?



L'erreur faite lors d'une intégration numérique dépendra de la __précision__ __et__ de la __stabilité__ du schéma choisi :



Soit $F : U_n \longmapsto U_{n+1} = F(U_n, t_n)$ un schéma d'intégration et U_n et $\hat{U}_n$ deux états proches ($\delta_n = \hat{U}_n - U_n$ petit). Alors

$$\delta_{n+1} = \hat{U}_{n+1} - U_{n+1} = F(\hat{U}_{n+1}, t_n) - F(U_n, t_n)$$

$$= F(U_n + \delta_n, t_n) - F(U_n, t_n)$$

$$\simeq F(U_n, t_n) + \delta_n \left.\frac{\partial F}{\partial U}\right|_{U_n, t_n} - F(U_n, t_n) = \overbrace{\frac{\partial F(U_n, t_n)}{\partial U}}^{A} \delta_n$$

définition : si $|A|>1$ pour tout pas de temps h, alors le schéma d'intégration F
est dit <u>inconditionnellement instable</u>

si $|A|<1$ pour tout pas de temps h, alors le schéma F
est dit <u>inconditionnellement stable</u>

si $|A|<1$ pour $h<h^*$, alors F est dit <u>conditionnellement stable</u>

remarque : Dans le cas où on a plusieurs variables, $\vec{U}_{n+1}=\vec{F}(\vec{U}_n,t_n)$,
$A=\frac{\partial F}{\partial U}(\vec{U}_n,t_n)$ est une matrice (éléments $a_{ij}=\frac{\partial F_i}{\partial U_j}(\vec{U}_n,t_n)$) et
il faut considérer les valeurs propres pour connaître la stabilité.

exemples : ① système masse-ressort en Euler explicite

$$\begin{cases}\dot{x}=v\\ \dot{v}=-\frac{k}{m}x\end{cases} \xRightarrow[\text{Euler expl.}]{\text{discret.}} \begin{cases}x_{n+1}\simeq x_n+hv_n\\ v_{n+1}\simeq v_n+h(-\frac{k}{m}x_n)\end{cases} \Leftrightarrow \begin{pmatrix}x_{n+1}\\ v_{n+1}\end{pmatrix}=F\begin{pmatrix}x_n\\ v_n\end{pmatrix}$$

t_n n'intervient pas

$$\rightsquigarrow A=\begin{pmatrix}\frac{\partial F_x}{\partial x} & \frac{\partial F_x}{\partial v}\\ \frac{\partial F_v}{\partial x} & \frac{\partial F_v}{\partial v}\end{pmatrix}_{\substack{x=x_n\\ v=v_n}}=\begin{pmatrix}1 & h\\ -h\frac{k}{m} & 1\end{pmatrix}$$

Valeurs propres solution de $\overbrace{\lambda^2-\mathrm{Tr}A\cdot\lambda+\det A}^{\det(A-\lambda)=}=0$

$$\lambda^2-2\lambda+1+\frac{k}{m}h^2=0 \rightsquigarrow \lambda_{1/2}=1\pm i\sqrt{\frac{k}{m}}\,h$$

$$|\lambda_{1/2}|=\sqrt{1+\frac{k}{m}h^2}>1 \quad\Rightarrow\quad \underline{\text{inconditionnellement instable}}$$

② système masse-ressort, en Euler implicite

$$\begin{cases}\dot{x}=v\\ \dot{v}=-\frac{k}{m}x\end{cases} \xRightarrow[\text{Euler implicite}]{\text{discret.}} \begin{cases}x_{n+1}\simeq x_n+hv_{n+1}\\ v_{n+1}\simeq v_n+h(-\frac{k}{m}x_{n+1})\end{cases}$$

en substituant l'une dans l'autre ces équ., on obtient

$$x_{n+1}=\frac{1}{1+\frac{k}{m}h^2}(x_n+hv_n) =: F_x(x_n,v_n)$$

$$v_{n+1}=\frac{1}{1+\frac{k}{m}h^2}(v_n-h\tfrac{k}{m}x_n) =: F_v(x_n,v_n)$$

$$\rightsquigarrow A=\alpha\begin{pmatrix}1 & h\\ -h\frac{k}{m} & 1\end{pmatrix} \quad\text{avec } \alpha=\frac{1}{1+\frac{k}{m}h^2}$$

$\Rightarrow$ valeurs propres $\lambda_{1/2}=\alpha(1\pm i\sqrt{\frac{k}{m}}\,h)$, $|\lambda_{1/2}|=\frac{1}{\sqrt{1+\frac{k}{m}h^2}}<1$ pour tout h

$\Rightarrow$ schéma <u>inconditionnellement stable</u>

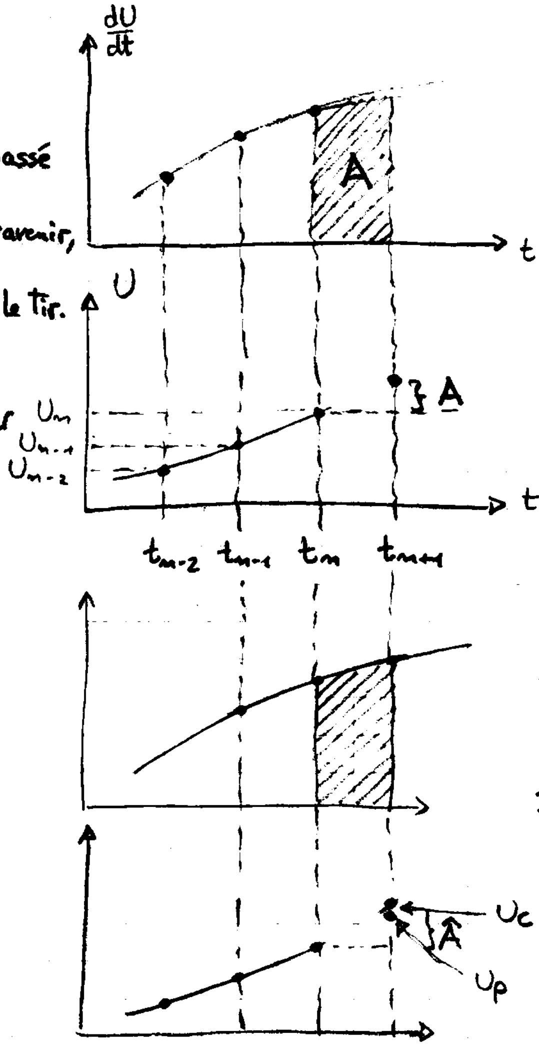
VI schémas d'intégration prédicteur - correcteurs

idée :

utiliser le passé pour prédire l'avenir, puis corriger le tir.

I - IV : prédicteur

V - VIII : correcteur



I ajuster un polynôme aux trois points les + récents de la dérivée (U'_{n-2}, U'_{n-1}, U'_n)

II l'extrapoler ~~passé et~~ jusqu'à t_{n+1} et intégrer sous la courbe pour obtenir A

III On obtient ~~le point~~ une estimation («prédiction») $U_p = U_n + A$ du nouveau point en t_{n+1}

IV calculer $U'_{n+1} = f(U_p, t_{n+1})$,

V Comme en I, ajuster polynôme aux trois points les + récents de la dérivée : U'_{n-1}, U'_n et U'_{n+1}

VI Intégrer sous la courbe entre t_n et t_{n+1} pour obtenir un $\hat{A}_c$

VII On obtient un nouveau point

$U_c = U_n + \hat{A}$ et $U'_{n+1,c} = f(U_c, t_{n+1})$

plus précis.

La correction apportée par ~~le~~ l'étape « correcteur » (V - VIII), $|U_c - U_p|$, ~~peut servir~~ donne une idée de la précision du calcul.

En pratique :

① prédiction $U_{n+1,p} = U_{n-3} + \frac{4h}{3}(2U'_{n-2} - U'_{n-1} + 2U'_n)$

(résultat des étapes I-III), en utilisant une parabole

② estimation de $U'_{n+1,p} = f(U_{n+1,p}, t_{n+1})$

③ correction $U_{n+1,c} = U_{n-1} + \frac{h}{3}(U'_{n-1} + 4U'_n + U'_{n+1})$

④ vérifier que $\frac{|U_{n+1,p} - U_{n+1,c}|}{20}$ n'est pas trop grand

⑤ (optionnel :) répéter ② ③ pour obtenir de nouvelles estimations

⑥ accepter $U_{n+1} := U_{n+1,c}$ et $U'_{n+1} := f(U_{n+1,c}, t_{n+1})$ comme nouveaux points et boucler sur ① (après $n \to n+1$)

2 remarques :

* si la correction $|U_{n+1,p} - U_{n+1,c}|$ est faible, on peut augmenter le pas de temps ⇝ schémas à pas adaptatif

* Si on n'a qu'un point au départ, il faut utiliser un autre schéma (ou développement Taylor) pour générer suffisamment de points pour amorcer cet algo.