Exo (fin) : comme $\dot{p}(t) = p(t)$, on a $p(t) = e^t \cdot p(0)$

$$= e^t \cdot \frac{2}{e^2 - 1}$$

Finalement, puisque $u(t) = p(t)$, $t \in [0,1]$ (cf. ii) cond. maximisation), on en déduit le contrôle optimal (ainsi que la trajectoire et le coût associés par intégration).

4. Méthode de tir.

Sur l'exo précédent, on a vu qu'on se ramène à intégrer le système suivant en (x,p) (après élimination de u avec la condition ii)) :

$$\begin{cases} \dot{x}(t) = -x(t) + u(t) = -x(t) + p(t) \\ \dot{p}(t) = p(t) \end{cases}$$

$\longleftarrow$ cf. $u(x(t), p(t)) = p(t)$

$p(0)$ inconnu

avec les conditions aux deux bouts $x(0) = -1$ et $x(1) = 0$.

Ce problème est équivalent à déterminer la condition initiale manquante : $p(0)$.

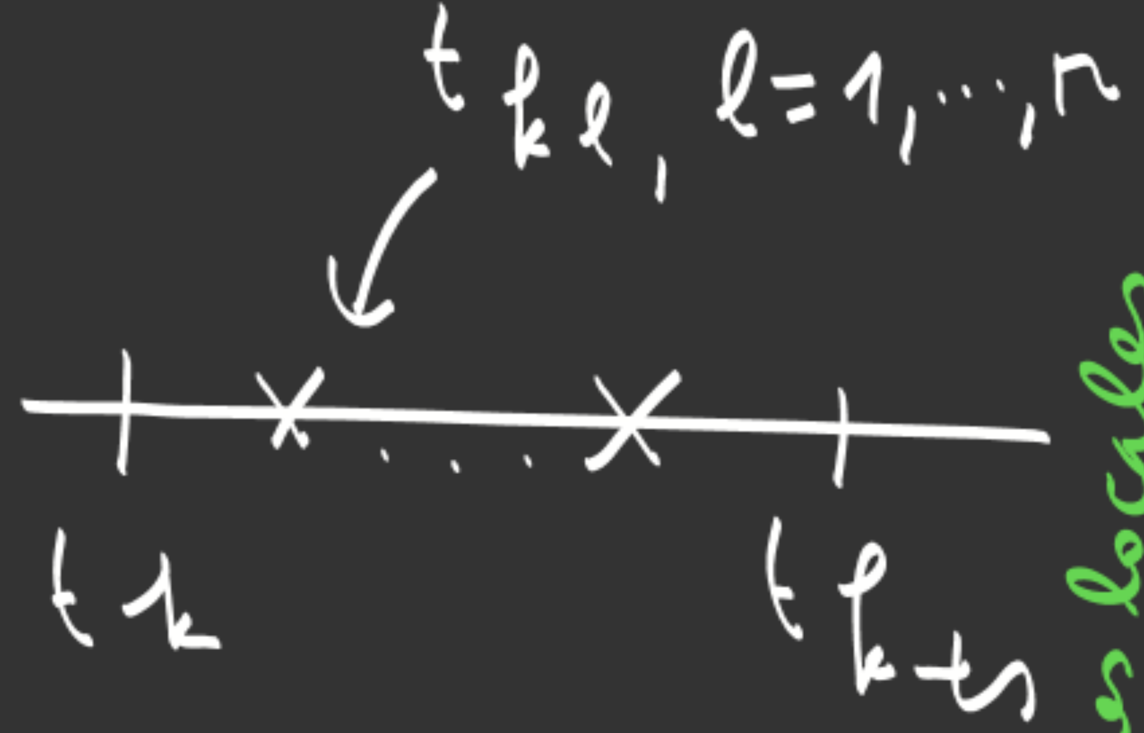
Rq. : de façon symétrique, on pourrait se ramener à chercher $p(1)$ (en intégrant de $1 \longrightarrow 0$, de manière rétrograde).

$\longrightarrow$ Équation de tir : trouver $p_0 \in \mathbb{R}$ $(n=1)$ tq

$$S(p_0) = 0 \quad \text{où} \quad S(p_0) = x(t=1, \underset{=-1}{x_0}, p_0) - \underset{=0}{x_f}$$

où $x(t_f, x_0, p_0)$ désigne la solution au temps t_f (on suppose qu'on a existence et unicité, et que la solution est bien définie jusqu'à t_f) de

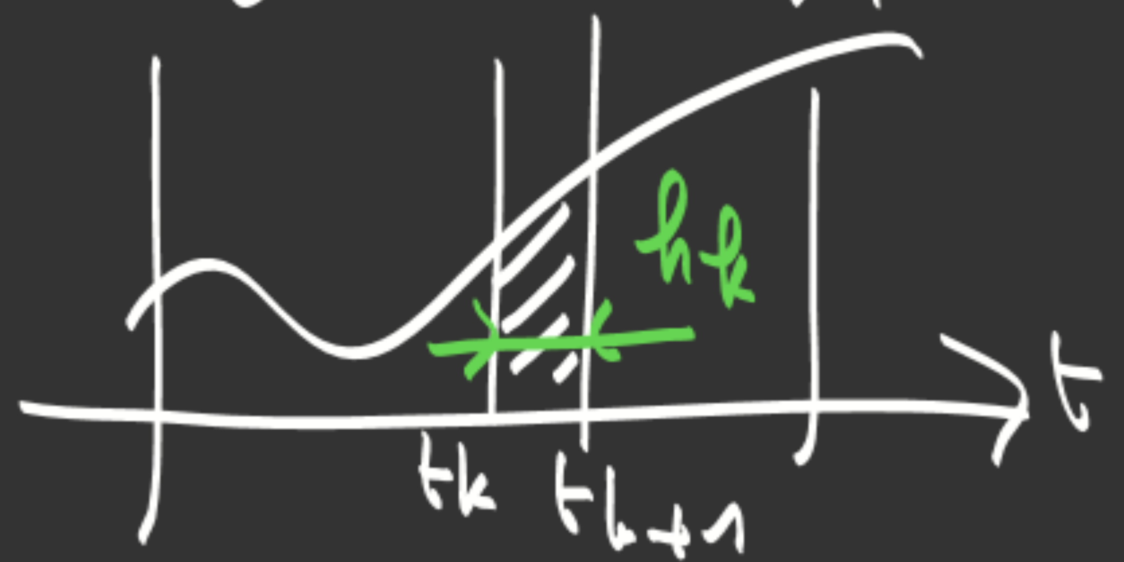
$$\begin{cases} \dot{x}(t) = -x(t) + p(t) \\ \dot{p}(t) = p(t) \end{cases} \quad \text{avec} \quad \begin{aligned} x(0) &= x_0 \\ p(0) &= p_0 \end{aligned}$$



En pratique, on combine deux solveurs :

- un solveur ODE (= Ordinary Differential Equation) : méthode à un pas (type RK = Runge-Kutta), multipas (Adams...)

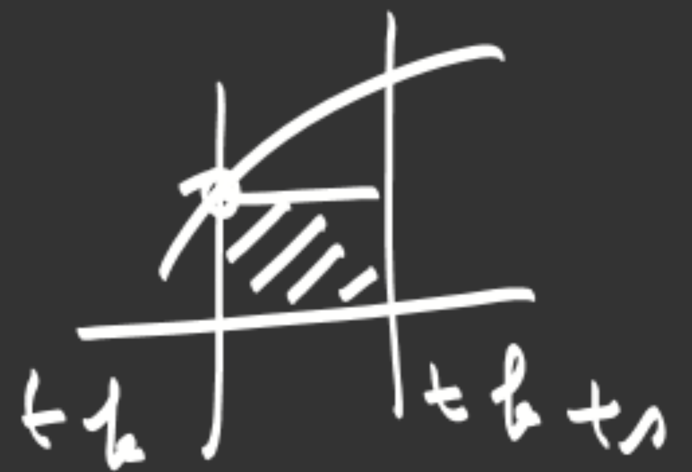
ex. : méthode à un pas : $\dot{y}(t) = g(t, y(t))$



$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} \underbrace{\dot{y}(t)}_{g(t,\, y(t))} \, dt \simeq y(t_k) + \sum_{\ell=1}^{n} \alpha_{k\ell} \, g(t_{k\ell}, y_{k\ell})$$

$y_{k\ell} \simeq y(t_{k\ell})$ → inconnues locales

Parmi les schémas de RK classiques, citons



. Euler explicite : $y_{k+1} = y_k + h_k \cdot g(t_k, y_k)$

. Euler implicite : $y_{k+1} = y_k + h_k \cdot g(t_{k+1}, y_{k+1})$

implicite : équ. de pt fixe (si g loc. lipschitz en y, on a une contraction si le pas h_k est assez petit)

$\longrightarrow$ quelques itérations de pt fixe (approximations successives) suffisent...

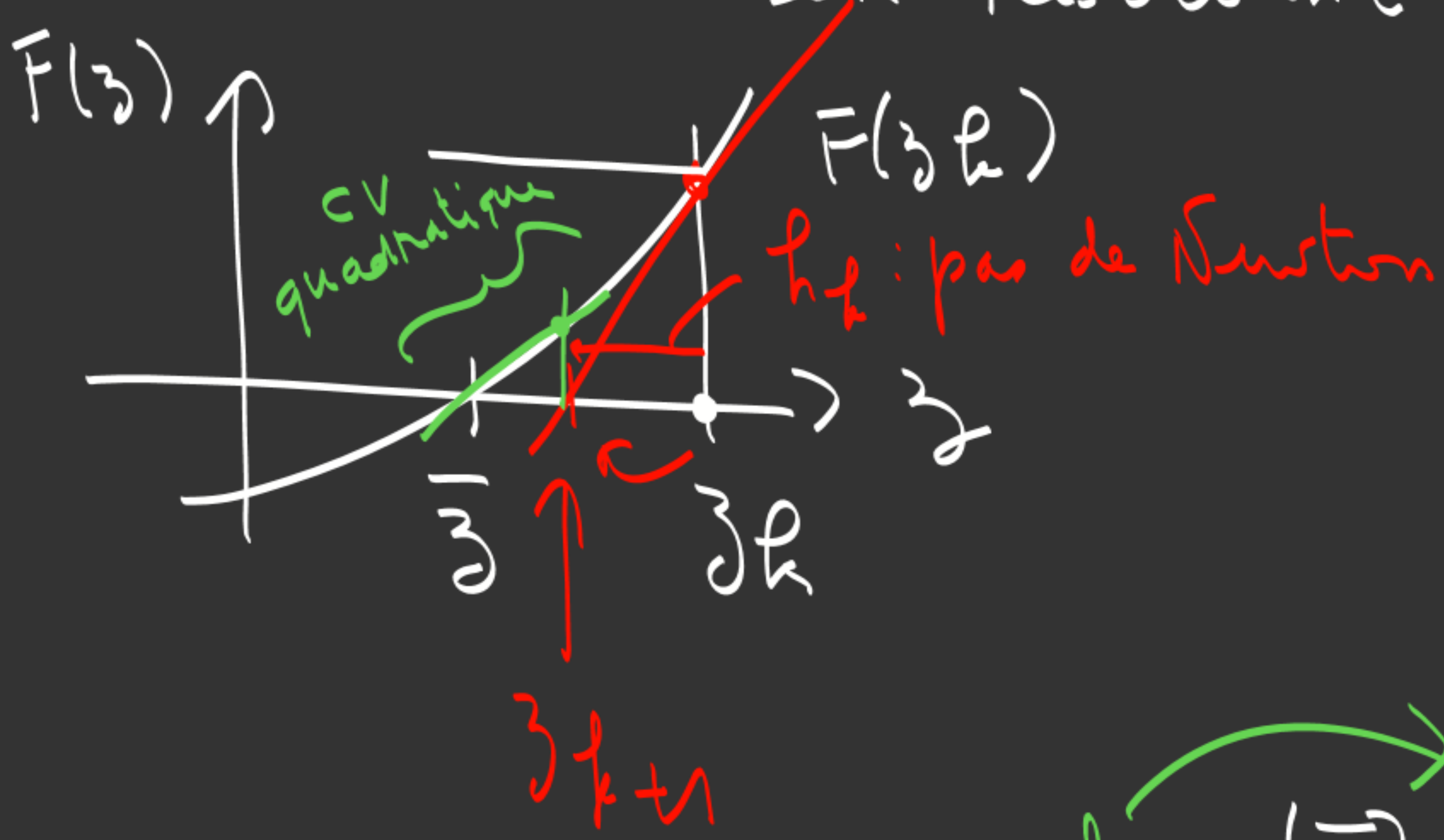
Le tout avec, quel que soit le schéma, pas variable (h_k n'est pas constant)

↑ contrôle erreur locale

. Trapèze / Crank-Nicolson : $y_{k+1} = y_k + \frac{h_k}{2}\left(g(t_k, y_k) + g(t_{k+1}, y_{k+1})\right)$

implicite

– un solveur NLE (Nonlinear Equation), par ex. de type Newton ($F(z)=0$, $F:\mathbb{R}^q \to \mathbb{R}^q$; aussi : moindres carrés nonlinéaires en résolvant $\frac{1}{2}\|F(z)\|^2 \longrightarrow \min$, $z \in \mathbb{R}^q$ ...)



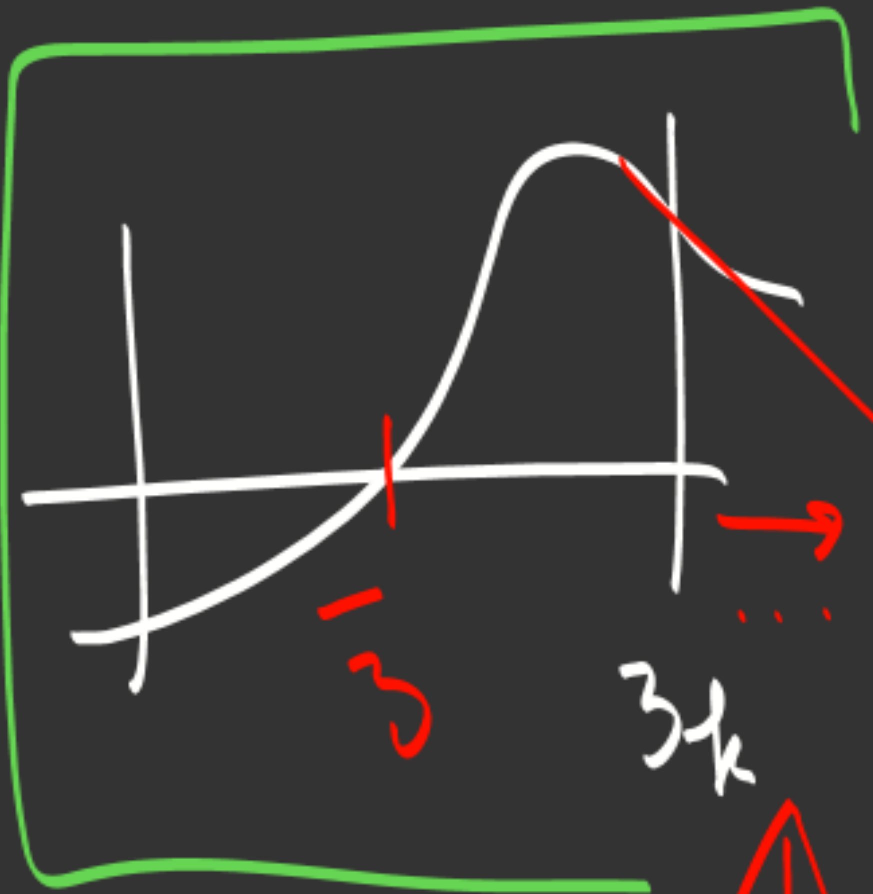
$h_k := z_{k+1} - z_k$ est sol. de "tangente en z_k" $= 0$:

$$F(z_k) + F'(z_k).\,h_k = 0$$

$$\Leftrightarrow \underbrace{F'(z_k)\,h_k = -F(z_k)}_{\text{1 système linéaire à FACTORISER (LU, QR...) à chaque itération}}$$: itération de Newton.

calcul de F'...?

Rq : i) CV quadratique, ie $\|\bar{z} - z_{k+1}\| \leq c \cdot \|\bar{z} - z_k\|^2 \ldots$



quand la CV a lieu : difficulté du choix de l'initial guess z_0 (Rappel : s'il existe $\bar{z} \in \mathbb{R}^q$ tq $F(\bar{z}) = 0$ et tq $F'(\bar{z}) \in GL(q, \mathbb{R})$, alors il existe $\varepsilon > 0$ tq, $\forall z_0 \in B_f(\bar{z}, \varepsilon)$, Newton initialisé par z_0 CV quadratiquement vers $\bar{z}$)

ii) $z_{k+1} = z_k + \alpha_k \cdot h_k$ ($\alpha_k = 1$: pas de Newton)

iii) en pratique, on réutilise la factorisation (LU, QR...) de $\underset{q \times q}{F'(z_k)}$ pour plusieurs itérations ($k+1, k+2 \ldots$)

À cette composition de solveur ODE (pour évaluer la fonction de tir) et de solveur NLE (pour en trouver un zéro), il convient d'ajouter un mécanisme de calcul de la dérivée $S'(p_0)$:

$$S'(p_0) = \underbrace{\left[\frac{\partial S}{\partial p_{0_1}}, \dots, \frac{\partial S}{\partial p_{0_n}}\right]}_{n\times n} \Big\updownarrow n$$

- DF (différences finies) : $e_j = (0,\dots,0,\overset{j}{1},0,\dots,0)$

$$\frac{\partial S}{\partial p_{0_j}}(p_0) \simeq \frac{S(p_0 + h.e_j) - S(p_0)}{h} \in \mathbb{R}^n, \; j = 1,\dots,n$$

$\rightsquigarrow n+1$ intégrations ODE

- AD (automatic differentiation / differentiable programming) : comme S est sol. d'une EDO, $S'(p_0)$ se calcule (cf. fonctions implicites) comme solution de l'EDO (linéaire matricielle) suivante :

$$\begin{cases} \dot{Y}(t) = \underbrace{g'(y(t,x_0,p_0))}_{2n\times 2n} \overbrace{Y(t)}^{2n\times 2n}, \quad t\in[0,1] \\ Y(0) = I_{2n} \end{cases}$$

AD

(connaissant $Y(1)$, on peut calculer $S'(p_0)$) où g est le second membre de l'équation en $y=(x,p)$:

$$\begin{cases} \dot{x}(t) = \dots \\ \dot{p}(t) = \dots \end{cases} \iff \dot{y}(t) = g(y(t)) \quad (\text{et } y(0) = (x_0,p_0))$$

AD : code $g : \mathbb{R}^{2n} \to \mathbb{R}^{2n} \xRightarrow{AD}$ code $g' : \mathbb{R}^{2n} \to \mathcal{L}(\mathbb{R}^{2n}, \mathbb{R}^{2n})$

Rq. : AD/calcul de dérivée : critique pour optimisation, ML ...

Sur notre exemple :

$u(t) = p(t) \dots$

$$\begin{cases} \dot{x}(t) = -x(t) + p(t) \\ \dot{p}(t) = p(t) \end{cases} \iff \begin{cases} \dot{x}(t) = \nabla_p H(x(t), p(t), u(t)) \\ \dot{p}(t) = -\nabla_x H(\text{---} \,\shortparallel\, \text{---}) \end{cases}$$

$$\iff \left.\begin{cases} \dot{x}(t) = \nabla_p h(x(t), p(t)) \\ \dot{p}(t) = -\nabla_x h(\text{---} \,\shortparallel\, \text{---}) \end{cases}\right\}$$

hamiltonien au sens strict (sans u !)

avec $h(x,p) := \max_{v \in U = \mathbb{R}} H(x,p,v)$

$= H(x, p, u = p)$

$= -\frac{1}{2} p^2 + p(-x + p)$

$= \frac{1}{2} p^2 - px$

$\leadsto \begin{pmatrix} \dot{x} \\ \dot{p} \end{pmatrix} = \vec{h}(x,p) = \begin{bmatrix} \nabla_p h \\ -\nabla_x h \end{bmatrix}(x,p)$

$\frac{1}{2} \int_0^1 u^2(t)\,dt$, $\dot{x} = -x + u$

$\Longrightarrow H = -\frac{1}{2} u^2 + p(-x + u)$

$p^0 = -1$