Méthodes numériques pour les EDP
Projet 3 : parallélisation de la résolution d'un problème de contrôle.

On souhaite paralléliser le problème de contrôle :

$$\min_{c} J(c) = ||y(T) - y_{cible}||^2 + \alpha \int c^2$$

où l'équation qui relie y et c est de type Schrödinger :

$i\hat{y}(t) = (A + c(t)B)y(t)$

avec un état initial y_o fixé.

1 Étude théorique

On commence par décomposer l'intervalle [0,T] en N sous-intervalles de même taille, les $[T_{i-1},T_i]$, avec $T_i=iT/N$. On examine alors une nouvelle fonctionnelle :

$$J_{||}(c,\Lambda) = \sum N x ||y_l(T_{l+1}) - \lambda_{l+1}||^2 + \alpha \int c^2$$

où $\Lambda = (\lambda_l)$ est une suite de N+1 états intermédiaires donnée, avec $\lambda_o = y_o$ et $\lambda_N = y_{cible}$, et les y_l sont définis sur $[T_l,T_{l+1}]$ par :
y_l vérifie la même équation que y sur $[T_l,T_{l+1}]$, et $y_l(T_l)=\lambda_l$. (on les étend à [0,T] naturellement)

1) J(c) minimise $J_{||}(c,\Lambda)$:

Soit Λ une suite d'états intermédiaires.
Par l'inégalité de Cauchy-Schwarz,

$$\sum N x ||y_l(T_{l+1}) - \lambda_{l+1}||^2 = \sum N x ||y_l(T_{l+1}) - \lambda_{l+1}||^2 \text{ x } \sum(1/N)$$

$$\geq \left(\sum ||y_l(T_{l+1}) - \lambda_{l+1}||\right)^2$$

On admet l'existence de $P_{j,k}$ de norme 1 tel que :
Pour tout j,k,l dans {0,...,N}, $P_{j,k}\, y_l(Tj) = y_l(Tk)$.

Lorsque l'on travaille avec l'algorithme permettant de calculer des valeurs approchées des y_l, l'existence de cette quantité ainsi que la formule permettant de la calculer sont données par la formule :

$y_l(n+1)=\exp(-IxAxdt)x\exp(-IxBxc(n)xdt)xy_l(n)$
$y_l(0)= \lambda_l$

Ainsi $\left(\prod \exp(-IxAxdt)x\exp(-IxBxc(n)xdt)\right) \text{ x } \lambda_l = y_l(M)$

$y_l(0)$ correspond à $y_l(T_l)$, $y_l(M)$ correspond à $y_l(T_{l+1})$, et le coefficient pour passer de l'un à l'autre est bien de norme 1. On peut itérer cette multiplication pour obtenir n'importe quel $y_l(Tk)$.

On peut maintenant écrire, pour l dans $\{0,...,N-2\}$:

$$\|y_l(T_{l+1}) - \lambda_{l+1}\| = \|P_{l,l+1}\, y_l(T_l) - \lambda_{l+1}\|$$
$$= \|P_{l+1,N}\left(P_{l,l+1}\, y_l(T_l) - y_{l+1}(T_{l+1})\right)\|$$
$$= \|P_{l,N}\, y_l(T_l) - P_{l+1,N}\, y_{l+1}(T_{l+1})\|$$

Pour $l = 0$, $P_{l,N}\, y_l(T_l) = y_0(T_N)$

Or $y_0 = y$ puisque ces deux fonctions vérifient la même équation et la même condition de départ.
Donc $P_{l,N}\, y_l(T_l) = y(T)$.

D'autre part, pour $l = N-1$, $\lambda_{l+1} = y_{cible}$ (c'est une condition nécessaire à la construction des états intermédiaires).

Au final, on obtient :

$$\sum N x\|y_l(T_{l+1}) - \lambda_{l+1}\|^2 \geq \left(\sum\|y_l(T_{l+1}) - \lambda_{l+1}\|\right)^2$$

et $\sum\|y_l(T_{l+1}) - \lambda_{l+1}\| \geq \| y(T) - y_{cible}\|$

Donc $J(c) \leq J_{||}(c,\Lambda)$, et ce pour tout c et pour tout Λ.

2) un Λ bien choisi donne $J(c) = J_{||}(c,\Lambda)$

On pose : pour l dans $\{0,..,N-1\}$, $\lambda_l = (1- l/N)y(T_l) + l/N\, p(T_l)$ où p est l'adjoint de y.

Alors la fonction $(1- l/N)y + l/N\, p$ vérifie l'équation différentielle, donc les conditions
y_l vérifie l'équation différentielle et $y_l(T_l)=\lambda_l$
donnent que $y_l = (1- l/N)y + l/N\, p$ sur tout $[0,T]$.

On calcule alors :

$$y_l(T_{l+1}) - \lambda_{l+1} = (1- l/N)y(T_{l+1}) + l/N\, p(T_{l+1}) - \left((1- (l+1)/N)y(T_{l+1}) + (l+1)/N\, p(T_{l+1})\right)$$
$$= 1/N\, y(T_{l+1}) - 1/N\, p(T_{l+1})$$

Donc $J_{||}(c,\Lambda) = \sum(1/N)x\|y(T_{l+1}) - p(T_{l+1})\|^2 + \alpha \int c^2$

On admet que $\|y - p\|$ est constant sur $[0,T]$.

On obtient alors : $J_{||}(c,\Lambda) = \sum(1/N)x\|y(T) - p(T)\|^2 + \alpha \int c^2$

Donc $J_{||}(c,\Lambda) = J(c)$

Un algorithme de parallélisation va consister en le choix d'un c, puis le calcul du Λ optimal pour ce c, puis le calcul du c optimal pour ce Λ, et ainsi de suite.
On pourra donc le voir comme un algorithme de « directions alternées ».

2 Étude pratique

Dans l'algorithme proposé en pièce jointe, la méthode avec parallélisation a besoin de commencer par le calcul des états intermédiaires, ce qui ne peut pas être parallélisé et qui empèche l'obtention d'une « full efficiency ».