ECN 7055 Macroéconomie B

Guillaume Sublet

Université de Montréal

Cours I.2: Méthodes numériques

Méthodes numériques

On a vu au cours I.1 : Résolution numérique du modèle néoclassique de croissance :

- Méthodes séquentielles
  - Approximation log-linéaire autour de l'état stationnaire
  - « Guess and verify » (conjecture et vérification)
- Méthodes récursives
  - Itération fonction valeur
  - Itération temps : Euler équation
  - Itération temps avec grille endogène
  - Algorithme d'amélioration d'Howard (itération de fonction de politique)
  - Lectures sur la théorie : LS chapitres 3 et 4
  - Mise en pratique : QuantEcon, Optimal Growth I, II, III, IV

Méthodes numériques

Méthodes séquentielles sont limitées

- Approximation log-linéaire est une méthode locale : c'est à dire, on se limite à étudier autour de l'état stationnaire et l'approximation de premier ordre nous limite dans la précision de l'analyse (e.g. bien-être)
- « Guess and verify » (conjecture et vérification) est une méthode globale et exacte, par contre, il nous faut pouvoir deviner la solution, ce qui est loin d'être facile

Méthodes récursives sont des méthodes globales. Dans ce cours, on va étudier des méthodes numériques pour la solution de formulation récursive des problèmes étudiés.

Rappel sur l'approche séquentielle

$$
\begin{aligned}
v(k_0, z_0) = \max_{(c_t, k_{t+1})_{t=0}^{\infty}} & \sum \beta^t E[u(c_t)] \\
& c_t + (k_{t+1} - (1-\delta)k_t) = z_t \; f(k_t) \qquad \text{pour}, t = 0, \ldots, \infty \\
& z_t \sim \phi \quad \text{pour}, t = 1, \ldots, \infty \\
& k_0, z_0 \text{ donné}
\end{aligned}
$$

L'approche séquentielle consiste à trouver une séquence infinie, un élément de ℓ^∞, qui résout le problème séquentiel.

Rappel sur l'approche récursive

$$
\begin{aligned}
v(k,z) = \max_{c} \left\{ u(c) + \beta E[v(k',z')] \right\} \\
c + (k' - (1-\delta)k) = z\ f(k) \\
z, z' \sim \phi
\end{aligned}
$$

L'approche récursive consiste à trouver une fonction valeur v, un élément de L^{∞}, qui résout le point fixe de l'équation de Bellman.

Méthodes numériques pour l'approche récursive

Exercice :

1. Établir formellement l'équivalence entre l'approche séquentielle et l'approche récursive. Nommer le principe qui établit cette équivalence.

2. L'approche séquentielle consiste à trouver une séquence infinie, an élément de ℓ^∞, qui résout le problème séquentiel.
   L'approche récursive consiste à trouver une fonction valeur (ou fonction de politique), un élément de L^∞, qui résout l'équation de Bellman. Les solutions des deux formulations sont des objets de dimension infinie. Pourquoi l'approche récursive est-elle si utile ?

Méthodes numériques

Exercice :

1. Coder par vous-même les cours Optimal Growth I, II, III, IV de QuantEcon puis comparer votre code avec celui du cours. Il est important de chercher à écrire votre propre code avant de regarder le code du cours sur QuantEcon.

2. Faire les exercices des cours Optimal Growth I, II, III, IV sur QuantEcon.

Approche récursive : importance de la variable d'état

$$v(k,z) = \max_{c} \left\{ u(c) + \beta E[v(k',z')] \right\}$$
$$c + (k' - (1-\delta)k) = z\ f(k)$$
$$z' = \rho z + \sigma\epsilon \qquad \text{où} \quad \epsilon \sim \phi$$

- La formulation récursive d'un problème dynamique requiert une *variable d'état* qui résume l'état du système. Par example, ici, cette variable est bi-dimensionelle (k, z)
- L'optimisation se fait en tenant compte de l'impact du choix présent c sur la variable d'état à la période suivante :

$$k' = z\ f(k) - c + (1-\delta)k$$
$$z' = \rho z + \sigma\epsilon$$

- en tenant en compte, grâce à la fonction valeur $\beta E[v(k',z')]$, que l'agent optimisera dans le futur.

Approche récursive : importance de la variable d'état

- Dans l'exemple ci-dessus, la variable d'état est bi-dimensionelle (k, z).
- On verra dans la section II.1 que l'état du système peut être un objet très difficile à résumer. Par example, lorsque les agents sont hétérogènes, la distribution des richesses fait partie de la description du système
- Or la distribution des richesses est un objet de dimension infinie et donc très difficile à résumer
- On verra une façon de réduire la complexité de la variable d'état (sans trop de perte) dans la section II.4 avec l'article de Krusell et Smith (1998) JPE

Approche récursive : importance de la variable d'état

Modélisation récursive de politique économique optimale

L'approche récursive est aussi utile pour la formulation de problème de politique optimale

- cependant, la politique gouvernementale optimale dépend du comportement des agents et le comportement des agents dépends des anticipations de politique gouvernementales futures ;
- cette simultanéité dans la détermination de la politique optimale et de la réponse des agents qui est tournée vers le future (anticipations rationnelles) semblait être une limitation de l'approche récursive dans les années 1970.
- Comme on le verra dans la section III.1 et III.3 du cours, il est possible d'ajouter une variable d'état tournée vers le future qui résume l'engagement du gouvernement en terme de politique future du gouvernement
- On en conclut qu'en choisissant bien la variable d'état, l'approche récursive est aussi utile pour la formulation de politique optimale

Approche récursive : importance de la variable d'état

L'approche récursive est aussi utile pour la formulation de problème d'incitations dynamiques

- les problèmes d'incitations dynamiques ont longtemps résisté à une formulation récursive car le contrat optimal dépend de l'historique
- une fois de plus, en choisissant bien la variable d'état, une formulation récursive est possible.
- la variable d'état judicieuse pour l'analyse de problème d'incitations dynamiques est l'utilité promise (« promised utility »)
- Nous n'aurons probablement pas le temps de couvrir ce sujet dans ce cours. Voir Chapitre 21 « Incentives and Insurance »de LS.
- On vient de voir que trouver la bonne variable d'état peut être un art. L'exercice suivant vous permet de mettre cet art en pratique.

Approche récursive : importance de la variable d'état

Exercice : Le problème étudié est le suivant :

$$v(k,z) = \max_{c} \left\{ u(c) + \beta E[v(k',z')] \right\}$$
$$c + (k' - (1-\delta)k) = z\ f(k)$$
$$z' = \rho z + \sigma\epsilon \qquad \text{où} \quad \epsilon \sim_{iid} \phi$$

1. Supposez que les chocs ne persistent pas dans le temps (c'est à dire $\rho = 0$) et la dépréciation est totale $\delta = 1$. Réécrivez le problème sous forme récursive avec une seule variable d'état.
2. Quel est l'avantage d'avoir une seule variable d'état au lieu de deux ?

Approche récursive

Considérons l'équation de Bellman suivante :

$$V(x) = \max_u \{r(x,u) + \beta E[V(g(x,u))]\}$$

Exercice en classe : Établir l'équivalence entre l'équation de Bellman ci-dessous et celle du modèle néoclassique stochastique suivante :

$$v(k,z) = \max_c \left\{u(c) + \beta E[v(k',z')]\right\}$$

$$c + (k' - (1-\delta)k) = z\ f(k)$$

$$z' = \rho z + \sigma\epsilon \qquad \text{où} \quad \epsilon \sim_{iid} \phi$$

Approche récursive

Considérons l'équation de Bellman suivante :

$$V(x) = \max_u \{r(x,u) + \beta E[V(g(x,u))]\}$$

Exercice en classe : Établir l'équivalence entre l'équation de Bellman ci-dessous et celle du modèle néoclassique stochastique suivante :

$$v(k,z) = \max_c \left\{u(c) + \beta E[v(k',z')]\right\}$$

$$c + (k' - (1-\delta)k) = z\ f(k)$$

$$z' = \rho z + \sigma\epsilon \qquad \text{où} \quad \epsilon \sim_{iid} \phi$$

Réponse : On identifie la variable d'état bi-dimensionnelle $x \equiv (k,z)$ et la variable de choix $u \equiv c$. On a donc

$$r(x,u) \equiv r((k,z),c) = u(c)$$

$$g(x,u) \equiv g((k,z),c) = \begin{cases} z\ f(k) + (1-\delta)k - c \\ \rho z + \sigma\epsilon \qquad \text{où} \quad \epsilon \sim_{iid} \phi \end{cases}$$

Approche récursive

Considérons l'équation de Bellman suivante :

$$V(x) = \max_u \{r(x,u) + \beta E[V(g(x,u))]\}$$

Exercice en classe : Établir l'équivalence entre l'équation de Bellman ci-dessous et celle du modèle néoclassique stochastique suivante :

$$v(k,z) = \max_c \left\{u(c) + \beta E[v(k',z')]\right\}$$

$$c + (k' - (1-\delta)k) = z\ f(k)$$

$$z' = \rho z + \sigma\epsilon \qquad \text{où} \quad \epsilon \sim_{iid} \phi$$

Réponse : On identifie la variable d'état bi-dimensionnelle $x \equiv (k,z)$ et la variable de choix $u \equiv c$. On a donc

$$r(x,u) \equiv r((k,z),c) = u(c)$$

$$g(x,u) \equiv g((k,z),c) = \begin{cases} z\ f(k) + (1-\delta)k - c \\ \rho z + \sigma\epsilon \qquad \text{où} \quad \epsilon \sim_{iid} \phi \end{cases}$$

Quelle serait la réponse si on avait choisit k' comme variable de choix au lieu de c ?

Condition de l'envelope (formule de Benveniste Scheinkman)

Considérons l'équation de Bellman suivante :

$$V(x) = \max_u \{r(x,u) + \beta E[V(g(x,u))]\}$$

La condition de l'enveloppe nous donne, pour $u = h(x)$ où h est la fonction de politique :

$$V'(x) = \frac{\partial r(x,u)}{\partial x} + \beta E\left[V'(g(x,u))\frac{\partial g(x,u)}{\partial x}\right]$$

Si $\frac{\partial g(x,u)}{\partial x} = 0$, on obtient la formule :

$$V'(x) = \frac{\partial r(x,h(x))}{\partial x} \qquad \text{(Benveniste Scheinkman)}$$

Condition de l'envelope (formule de Benveniste Scheinkman)

Considérons l'équation de Bellman suivante :

$$V(x) = \max_u \{r(x,u) + \beta E[V(g(x,u))]\}$$

La condition de l'enveloppe nous donne, pour $u = h(x)$ où h est la fonction de politique :

$$V'(x) = \frac{\partial r(x,u)}{\partial x} + \beta E\left[V'(g(x,u))\frac{\partial g(x,u)}{\partial x}\right]$$

Si $\frac{\partial g(x,u)}{\partial x} = 0$, on obtient la formule :

$$V'(x) = \frac{\partial r(x,h(x))}{\partial x} \qquad \text{(Benveniste Scheinkman)}$$

Exercice en classe : Est ce que la condition $\frac{\partial g(x,u)}{\partial x} = 0$ est vérifiée pour l'exemple de la diapositive 13 ? Oui si on choisit k' comme variable de choix ; non si on choisit c comme variable de choix.

Équation d'Euler

Considérons l'équation de Bellman suivante :

$$V(x) = \max_u \{r(x,u) + \beta E[V(g(x,u))]\}$$

L'équation d'Euler est :

$$\frac{\partial r(x,h(x))}{\partial u} + \beta E\left[V'(g(x,u))\frac{\partial g(x,u)}{\partial u}\right]$$

La formule de Benveniste Scheinkman (condition de l'enveloppe) nous donne l'équation d'Euler :

$$\frac{\partial r(x,h(x))}{\partial u} + \beta E\left[\frac{\partial r(x',h(x'))}{\partial x}\frac{\partial g(h(x))}{\partial u}\right] = 0$$

où $x' = g(u) = g(h(x))$.

Deux équations fonctionnelles

Pour résumer, on a le choix de résoudre une des deux équations fonctionnelles

1. **Équation de Bellman :**

$$V(x) = \max_u \{r(x,u) + \beta E[V(g(x,u))]\}$$

La solution de cette équation fonctionnelle est une fonction V. L'algorithme d'itération de fonction valeur a pour but de trouver la solution de cette équation.

Deux équations fonctionnelles

2. **Équation d'Euler :**

$$\frac{\partial r(x, h(x))}{\partial u} + \beta E\left[\frac{\partial r(x', h(x'))}{\partial x}\frac{\partial g(h(x))}{\partial u}\right] = 0$$

où $x' = g(u) = g(h(x))$ et donc

$$\frac{\partial r(x, h(x))}{\partial u} + \beta E\left[\frac{\partial r(g(h(x)), h(g(h(x))))}{\partial x}\frac{\partial g(h(x))}{\partial u}\right] = 0$$

La solution de cette équation fonctionnelle est une fonction h. L'algorithme d'itération de temps (équation d'Euler) a pour but de trouver la solution de cette équation.

Une fois la fonction valeur V trouvée, on peut facilement obtenir la fonction de politique :

$$\max_u \{r(x,u) + \beta E[V(g(x,u))]\}$$

Il suffit de résoudre, pour chaque x, le choix optimal de u ; ça nous donne la fonction de politique $h(x)$.

Une fois la fonction de politique $h(x)$ trouvée, on peut facilement calculer la fonction valeur en utilisant le principe d'optimalité :

$$V(x) = E\left[\sum_{t=0}^{\infty} \beta^t r(x_t, h(x_t))\right]$$

où $x_1 = g(x, h(x))$ et $x_{t+1} = g(x_t, h(x_t))$ pour $t \geq 1$

Trois méthodes numériques

1. Itération fonction valeur

2. Itération temps (équation d'Euler)

3. Itération temps (équation d'Euler) avec grille endogène

On verra aussi l'algorithme d'amélioration d'Howard (itération de fonction de politique)

Itération fonction valeur

Prenons comme exemple

$$
\begin{aligned}
v(y) &= \max_c \left\{ u(c) + \beta E[v(y')] \right\} \\
y' &= \xi \ f(y - c) \\
\xi &\sim \phi
\end{aligned}
$$

On définit l'opérateur de Bellman $T : v \mapsto L^\infty$ où $v \in L^\infty$. Pour chaque y, la fonction $T(v)$ évaluée à y est définie ainsi :

$$
\begin{aligned}
T(v)(y) &= \max_c \left\{ u(c) + \beta E[v(y')] \right\} \\
y' &= \xi \ f(y - c) \\
\xi &\sim \phi
\end{aligned}
$$

Itération fonction valeur

- La solution de l'équation de Bellman est un point fixe de l'opérateur de Bellman $v = T(v)$.

- L'algorithme d'itération fonction valeur consiste à itérer l'opérateur de Bellman sur une fonction initiale v_0 et on obtient une suite $(v_j)_{j=0}^{J}$ où J est le nombre d'itérations et $v_{j+1} = T(v_j)$.

- Si T est une contraction (conditions de Blackwell), on sait que $\lim_{j\to\infty} v_j = v$

Itération fonction valeur

Exercice en classe :

Que faire si les conditions de Blackwell ne sont pas satisfaites ?

Itération fonction valeur

Exercice en classe :
Que faire si les conditions de Blackwell ne sont pas satisfaites ?

On peut toujours essayer. On est juste pas sûr que l'itération va converger. Cependant, si l'itération de fonction valeur converge, on trouve un point fixe qui est la solution de l'équation de Bellman. Souvent les conditions de Blackwell ne sont pas satisfaites. Les fonctions d'utilité couramment utilisées ne sont pas bornées.

Exemple d'itération fonction valeur

Essayons avec le problème

$$v(y) = \max_c \left\{ u(c) + \beta E[v(y')] \right\}$$
$$y' = \xi \; f(y - c)$$
$$\xi \sim \phi$$

avec les paramétres suivants :

$$\beta = 0.96$$
$$u(c) = \ln(c)$$
$$f(k) = k^{0.4}$$
$$\phi(\xi) = \exp(\mu + s \; \zeta) \text{ où } \zeta \sim N(0, 1)$$
$$\mu = 0 \quad \text{et} \quad s = 0.1$$

Exemple d'itération fonction valeur

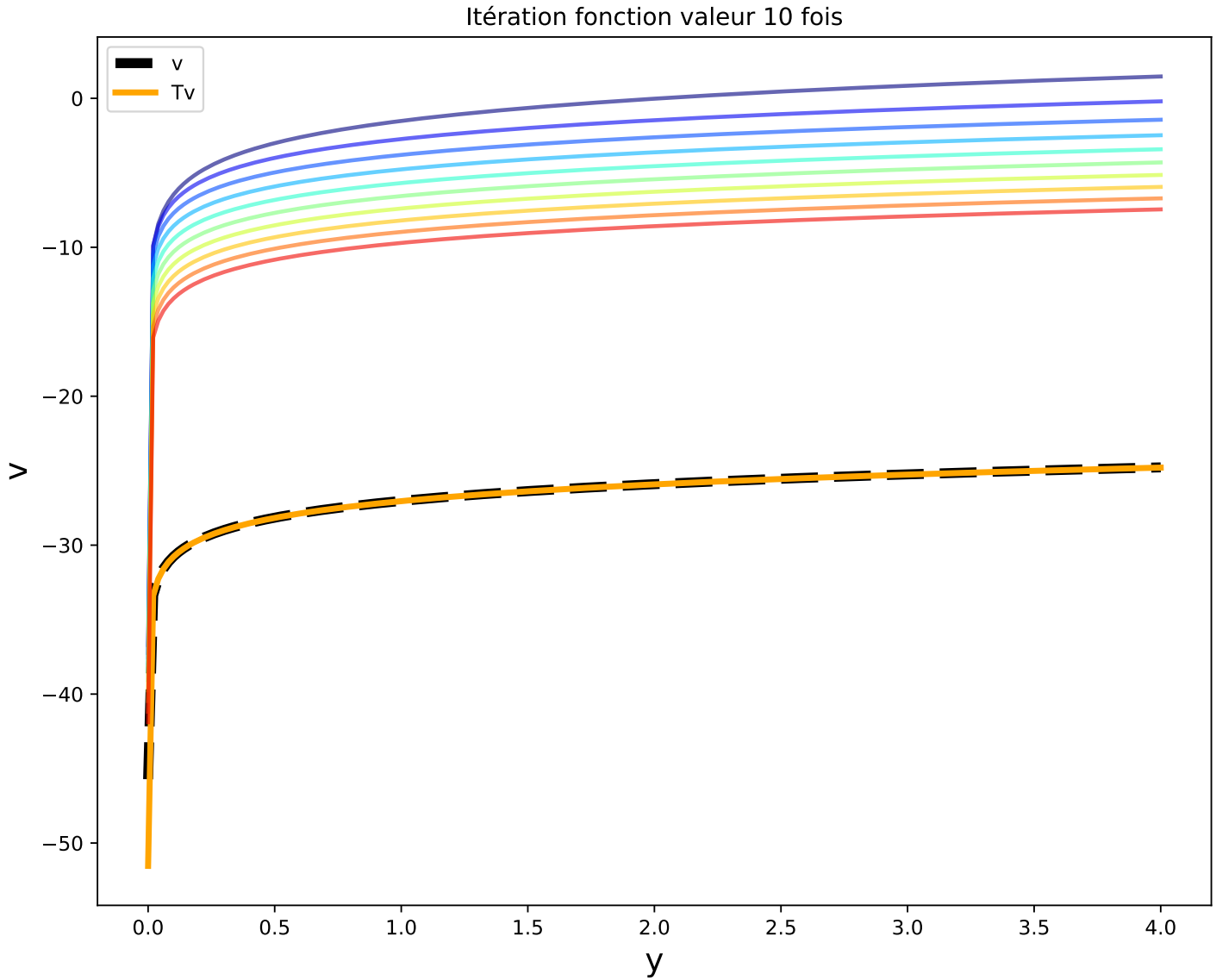
Avec utilité log et dépréciation totale, on a déjà trouver la solution par « guess and verify » qui est :

$$v(y) = \frac{\ln(1-\alpha\beta)}{1-\beta} + \frac{\mu + \alpha\ln(\alpha\beta)}{1-\alpha}\left[\frac{1}{1-\beta} - \frac{1}{1-\alpha\beta}\right] + \frac{1}{1-\alpha\beta}\ln(y)$$

$$h(y) = (1-\alpha\beta)y$$

On va se servir de la solution exacte pour s'assurer que les algorithmes numériques nous donnent quelque chose proche de la solution.

Exemple d'itération fonction valeur



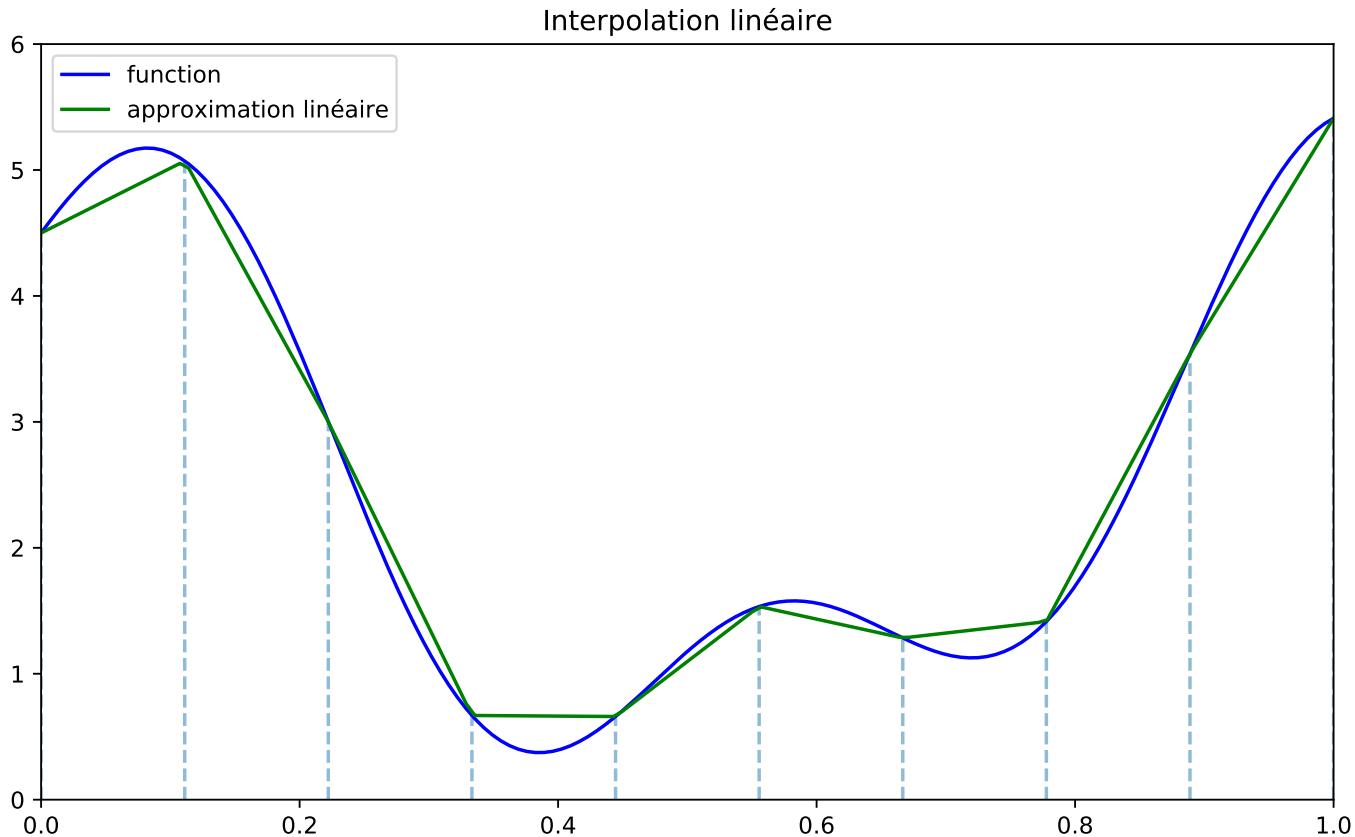
Exemple d'itération fonction valeur

- On voit que la solution v en pointillé noir est bien le point fixe de l'opérateur de Bellman avec $T(v)$ en orange
- La ligne bleue la plus haute est notre conjecture initiale v_0
- Les lignes entre la ligne bleue la plus haute et v sont les points de la séquence $(v_j)_{j=0}^{50}$
- Si on continue les itérations, la séquence converge vers $\lim_{j\to\infty} v_j = v$

Exemple d'itération fonction valeur

- Nous n'avons pas de forme analytique pour les fonctions v_t
- Nous allons donc stocker v_t sur une grille de points $(y_0, y_1, \ldots, y_n)$ où n est la taille de la grille
- Plus n est grand, plus les calculs sont longs car la fonction doit être calculée pour plus de points
- Ensuite on reconstruit v_t à partir de la grille $(y_0, y_1, \ldots, y_n)$ pour laquelle on a $(v(y_0), v(y_1), \ldots, v(y_n))$ en faisant une interpolation linéaire
- Plus n est grand, meilleure est l'approximation de v_t par interpolation linéaire de $(v(y_0), v(y_1), \ldots, v(y_n))$

Exemple d'approximation de fonction par interpolation linéaire



Algorithme d'itération fonction valeur

Algorithme :

1. Conjecture initiale de v_0 sur une grille $(y_1, \ldots, y_n)$ et choisir la tolérance (très petite). Aussi, $j = 0$.
2. Construire v_j par interpolation linéaire de $(v_j(y_1), \ldots, v_j(y_n))$ sur la grille $(y_1, \ldots, y_n)$
3. Calculer $v_{j+1}(y_i)$ pour chaque point de la grille y_i en appliquant l'opérateur de Bellman comme suit :

$$v_{j+1}(y_i) = \max_c \left\{ u(c) + \beta E[v_j(y')] \right\}$$
$$y' = \xi \; f(y_i - c)$$
$$\xi \sim \phi$$

4. Si $sup_{y \in \{y_1, \ldots, y_n\}} \mid (v_{j+1}(y) - v_j(y)) \mid <$ tolérance, alors on considère que l'algorithme a convergé et on rapporte v_{j+1} comme approximation du point fixe v.
   Si $sup_{y \in \{y_1, \ldots, y_n\}} \mid (v_{j+1}(y) - v_j(y)) \mid \geq$ tolérance on retourne au point 2 avec $j = j + 1$.

Algorithme d'itération fonction valeur

Voir QuantEcon « Optimal Growth I: The Stochastic Optimal Growth Model » pour un exemple de code qui utilise cet algorithme.

Une alternative est disponible sur ma page GitHub (lien web) sous le dépôt « repository : QuantEcon »et le code est « QuantEcon_Optimal_Growth_I_II_III.ipynb ». J'ai écrit ce code en cherchant à maximiser la clarté par rapport à ces notes de cours. Mon code est donc probablement moins efficace que le code disponible sur QuantEcon.

Itération temps

Rappel : Équation d'Euler

$$\frac{\partial r(x, h(x))}{\partial u} + \beta E\left[\frac{\partial r(g(h(x)), h(g(h(x))))}{\partial x}\frac{\partial g(h(x))}{\partial u}\right] = 0$$

La solution de cette équation fonctionnelle est une fonction h.
L'algorithme d'Iteration de temps (équation d'Euler) a pour but de trouver la solution de cette équation.

Itération temps

Dans le cas de notre exemple, léquation d'Euler est

$$u'(c) = \beta \int v'(f(y-c)z)\ f'(y-c)z\ \phi(dz)$$

et la condition de l'enveloppe est

$$v'(y) = \beta \int v'(f(y-c)z)\ f'(y-c)z\phi dz = u'(h(y)) = (u' \circ h)(y)$$

Ça nous donne l'équation d'Euler suivante :

$$u'(h(y)) = \beta \int (u' \circ h)(f(y-h(y))z)\ f'(y-h(y))z\phi dz$$

Il nous faut trouver $h(y)$, le point fixe de cette équation fonctionnelle.

Itération temps

On définit l'opérateur de Coleman-Reffett K comme suit :

Pour chaque y, $K(h_j(y))$ est la solution c de l'équation suivante :

$$u'(c) = \beta \int (u' \circ h_j)(f(y-c)z)\ f'(y-c)z\phi dz$$

Pour chaque y, il faut trouver la racine de l'équation d'Euler où la condition de l'enveloppe a permis de remplacé l'inconnue $v'(y)$ par $(u' \circ h_j)(y)$

Itération temps

- La fonction de politique est le point fixe de l'opérateur de Coleman-Reffett

$$h = K(h)$$

- L'algorithme d'itération temps consiste à itérer l'opérateur de Coleman-Reffett sur une fonction initiale h_0 et on obtient une suite $(h_j)_{j=0}^{J}$ où J est le nombre d'itérations et $h_{j+1} = K(h_j)$.

- Sous certaines condition $\lim_{j\to\infty} h_j = h$

- Voir QuantEcon Optimal Growth III: Time Iteration pour une comparaison des propriétés des deux algorithmes. Voir aussi W. J. Coleman (1990) "Solving the Stochastic Growth Model by Policy-Function Iteration" *JBES* et Reffett (1996) "Production-based asset pricing in monetary economies with transactions costs" *Economica*

Algorithme d'itération temps

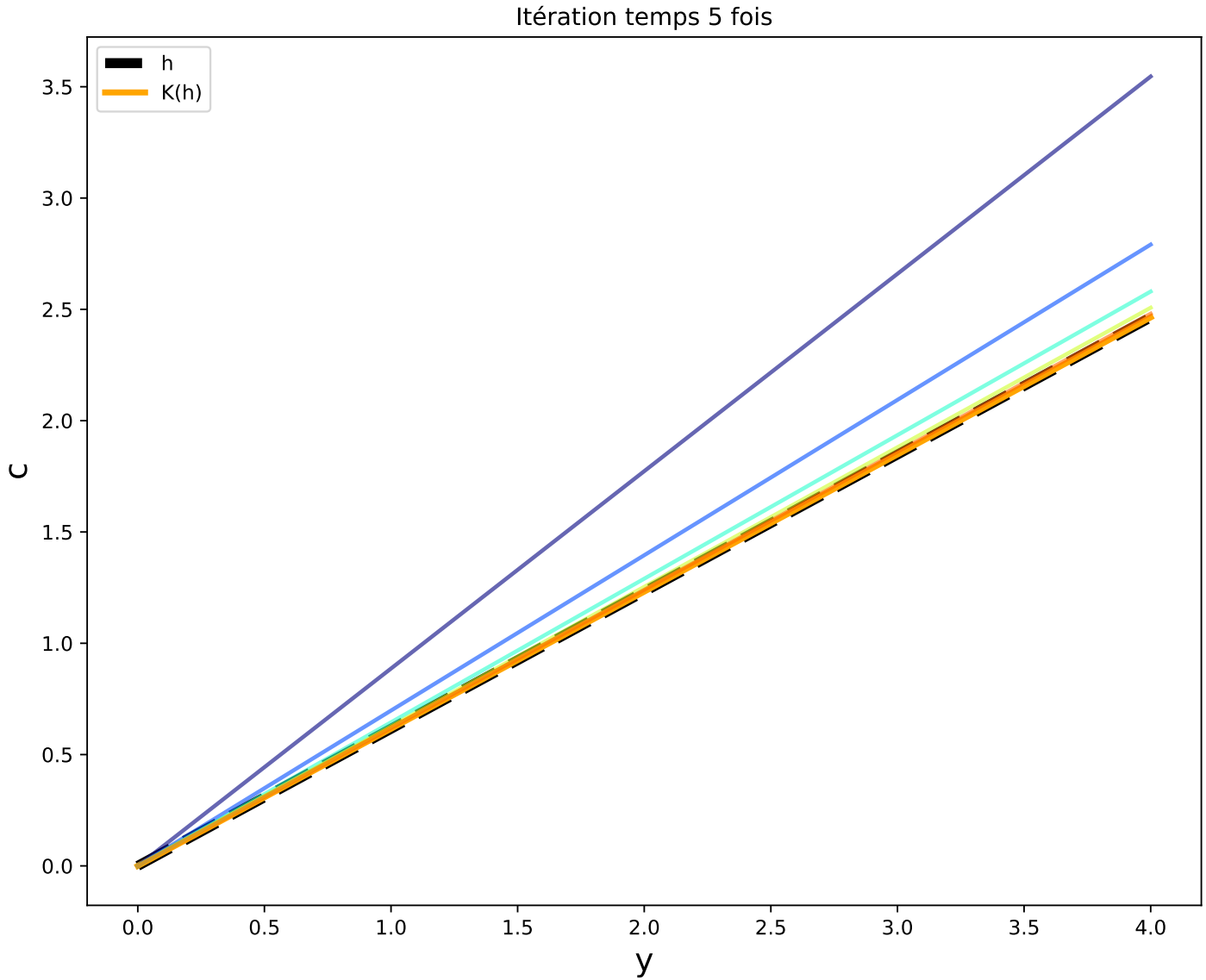
Algorithme :

1. Conjecture initiale de h_0 sur une grille $(y_1, \ldots, y_n)$ et choisir la tolérance (très petite). Aussi, $j = 0$.
2. Construire h_j par interpolation linéaire de $(h_j(y_1), \ldots, h_j(y_n))$ sur la grille $(y_1, \ldots, y_n)$
3. Calculer $h_{j+1}(y_i)$ pour chaque point de la grille y_i en appliquant l'opérateur de ColemanReffett à h_j comme suit : $h_{j+1}(y_i)$ est le c qui satisfait léquation d'Euler

$$u'(c) = \beta \int u'\Big(h_j\big(f(y-c)z\big)\Big)\, f'(y-c)z\, \phi dz$$

4. Si $sup_{y \in \{y_1, \ldots, y_n\}} \mid (h_{j+1}(y) - h_j(y)) \mid <$ tolérance, alors on considère que l'algorithme a convergé et on rapporte h_{j+1} comme approximation du point fixe h.
   Si $sup_{y \in \{y_1, \ldots, y_n\}} \mid (h_{j+1}(y) - h_j(y)) \mid \geq$ tolérance on retourne au point 2 avec $j = j + 1$.

Exemple d'itération temps



Exemple d'itération temps

- On voit que la solution h en pointillé noir est bien le point fixe de l'opérateur de Coleman-Reffett avec $K(h)$ en orange
- La ligne bleue la plus haute est notre conjecture initiale h_0
- Les lignes entre la ligne bleue la plus haute et h sont les points de la séquence $(h_j)_{j=0}^{5}$
- Si on continue les itérations, la séquence converge vers $\lim_{j\to\infty} h_j = h$

Algorithme d'itération temps

Voir QuantEcon « Optimal Growth III: Time Iteration » pour un exemple de code qui utilise cet algorithme.
Une alternative est disponible sur ma page GitHub (lien web) sous le dépôt « repository : QuantEcon »et le code est « QuantEcon_Optimal_Growth_I_II_III.ipynb ». J'ai écrit ce code en cherchant à maximiser la clarté par rapport à ces notes de cours. Mon code est donc probablement moins efficace que le code disponible sur QuantEcon.

Itération temps avec grille endogène

- Pour appliquer l'opérateur de Coleman-Reffett, on doit trouver la racine de l'équation d'Euler pour chaque point de la grille y_i
- Cette opération est numériquement coûteuse
- L'approche de la grille endogène nous permet d'éviter cette opération comme suit si la fonction u' a une inverse u'^{-1} :

$$c = u'^{-1}\left(\beta \int u'\Big(h_j\big(f(y-c)z\big)\Big)\ f'(y-c)z\ \phi dz\right)$$

au lieu de trouver la racine de

$$u'(c) = \beta \int u'\Big(h_j\big(f(y-c)z\big)\Big)\ f'(y-c)z\ \phi dz$$

Algorithme d'itération temps avec grille endogène

Algorithme :

1. Conjecture initiale de h_0 sur une grille initiale $(y_1^0, \ldots, y_n^0)$ et choisir la tolérance (très petite). Aussi, $j = 0$.
2. Construire h_j par interpolation linéaire de $(h_j(y_1^j), \ldots, h_j(y_n^j))$ sur la grille $(y_1^j, \ldots, y_n^j)$
3. Construire une grille pour k, étant donné la grille pour y actuelle, comme suit :

$$k^j = y^j - h_j(y^j)$$

Algorithme d'itération temps avec grille endogène

Algorithme (suite) :

4. Pour chaque y_i^j, calculer $h^{j+1}(y_i^j)$ ainsi :

$$h^{j+1}(y_i^j) \equiv (u')^{-1}\left(\beta \int (u' \circ h^j)(f(k^j(y_i^j))z)\ f'(k^j(y_i^j))z\phi dz\right)$$

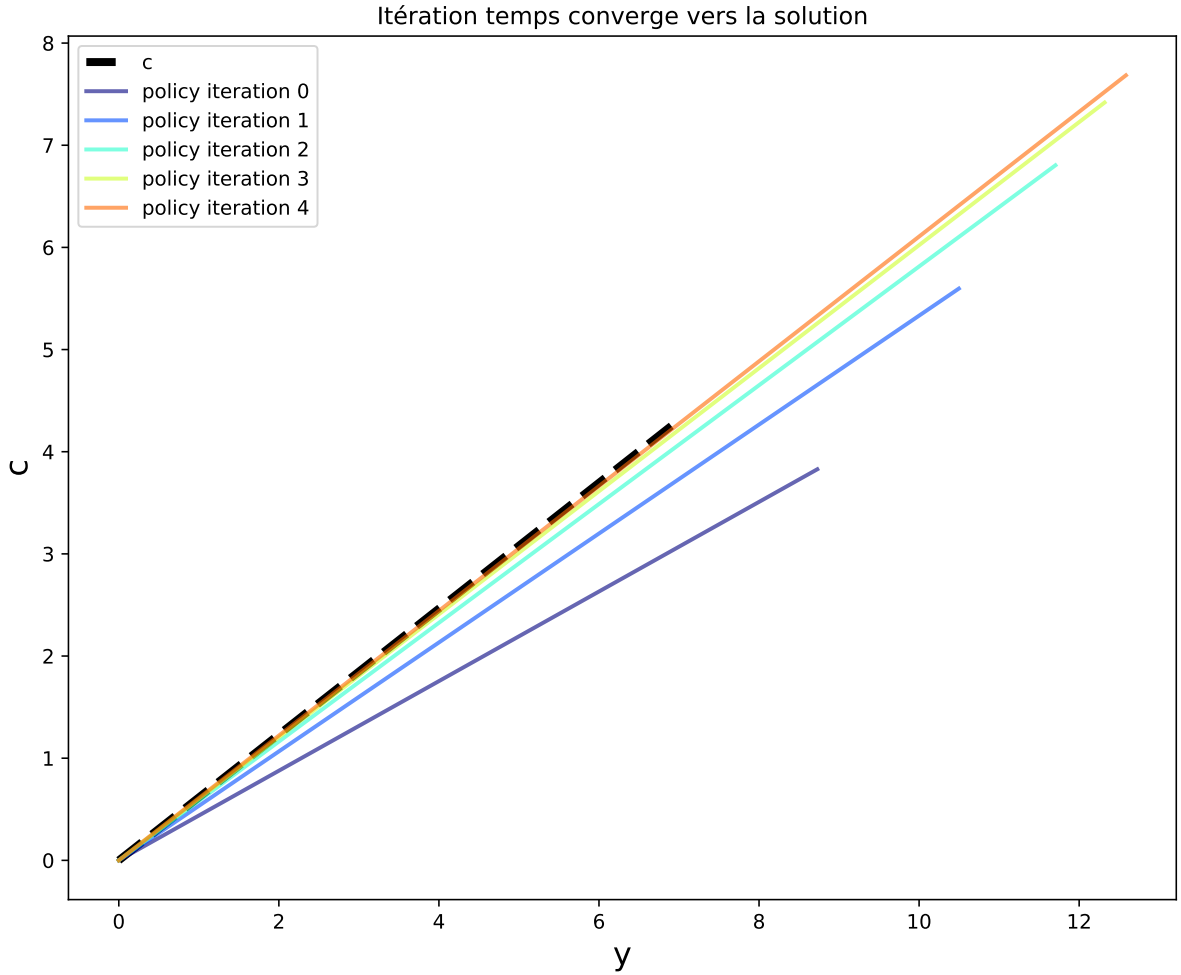
5. Construire une nouvelle grille (d'où l'endogénéité) pour y :

$$y^{j+1} = k^j(y^j) + h^{j+1}(y^j) = y^j - h_j(y^j) + h^{j+1}(y^j)$$

6. Si $sup_{y \in \{y_1^j, \ldots, y_n^j\}} \mid (h_{j+1}(y) - h_j(y)) \mid <$ tolérance, alors on considère que l'algorithme a convergé et on rapporte h_{j+1} sur la grille $(y_1^j, \ldots, y_n^j)$ comme approximation du point fixe h.

   Si $sup_{y \in \{y_1^j, \ldots, y_n^j\}} \mid (h_{j+1}(y) - h_j(y)) \mid \geq$ tolérance on retourne au point 2 avec $j = j + 1$.

Exemple d'itération temps avec grille endogène



On remarque que l'endogénéité de la grille se manifeste clairement au niveau du point maximum de la grille.

Algorithme d'itération temps avec grille endogène

Voir QuantEcon Optimal Growth IV: The Endogenous Grid Method pour un exemple de code qui utilise cet algorithme.

Une alternative est disponible sur ma page GitHub (lien web) sous le dépôt « repository : QuantEcon »et le code est « QuantEcon_Optimal_Growth_IV.ipynb ». J'ai écrit ce code en cherchant à maximiser la clarté par rapport à ces notes de cours. Mon code est donc probablement moins efficace que le code disponible sur QuantEcon.

Méthodes numériques

Exercice : Considérer le problème suivant :

$$
\begin{aligned}
v(y) &= \max_c \left\{ u(c) + \beta E[v(y')] \right\} \\
y' &= z' \; f(y - c) \\
c + (k' - (1 - \delta)k) &= z \; f(k) \\
z' &= \rho z + \sigma \epsilon \qquad \text{où} \quad \epsilon \sim_{iid} \phi \; .
\end{aligned}
$$

1. Décrire l'algorithme d'Itération de Fonction Valeur et le programmer.
2. Décrire l'algorithme d'Itération de temps (équation d'Euler) et le programmer.
3. Décrire l'algorithme d'Itération de temps (équation d'Euler) avec grille endogène et le programmer.

Exercice :

1. Décrire l'algorithme d'amélioration d'Howard (itération de fonction de politique).
2. Comparer l'algorithme d'amélioration d'Howard et l'algorithme d'itération fonction valeur.

Pour une description de l'algorithme d'amélioration d'Howard voir LS ou Section 3.2.2 « Policy Function Iterations »du livre « Dynamic Economics » d' Adda et Cooper.

Pour approfondir

Grand espace d'état : le « fléau de la dimensionalité »

Si l'espace d'état est grand, vous pourriez faire face au « fléau de la dimensionalité »(« curse of dimensionality »). Les méthodes de projections permettent de résoudre ces problèmes programmation dynamique avec un grand espace d'état. Voir Section 3.2.3 « Projection Methods »du livre « Dynamic Economics » d' Adda et Cooper.

L'idée est la suivante :

- Utiliser l'algorithme d'itération de temps (équation d'Euler)
- Approximer la fonction de politique à l'aide de n éléments d'une base pour l'espace des fonctions continues

...

Pour approfondir

Grand espace d'état : le « fléau de la dimensionalité »

...

- Approximer la fonction de politique à l'aide de n éléments d'une base $(p_i(x) = x^i)_{i=0}^{n-1}$ pour l'espace des fonctions continues :

$$h(x) = \sum_{i=1}^{n} \psi_i p_i(x)$$

  où le choix de la fonction h se résume au choix de n paramètres $(\psi_i)_{i=0}^{n-1}$
- Exemple de choix de base souvent utilisées :
  - Les n premiers polynômes $(p_i(x) = x^i)_{i=0}^{n-1}$
  - Les n premiers polynômes de Chebyshev pour avoir une base orthogonal
- Le choix de $(\psi_i)_{i=0}^{n-1}$ se fait à chaque itération par une projection, d'où le nom de l'approche.
- Plus n est grand, plus l'approximation est précise, mais plus les calculs sont coûteux.

Pour approfondir

Temps continu

Une autre approche qui permet parfois de résoudre un modèle plus rapidement est de le formuler en temps continu.

Méthodes de résolution de modèles formulées en temps continu :

- Phelan Eslami (2021) "Applications of Markov Chain Approximation Methods to Optimal Control Problems in Economics"
- Ben Moll a un site web dédié à ces méthodes (lien web)

Ces méthodes peuvent être beaucoup plus rapides que pour le résolution de modèles formulés en temps discret.

Pour approfondir

Méthodes d'interpolation

1. Linéaire
   (Méthode la plus simple)

2. Spline cubique
   (Performance : souvent très bonne)

3. Moindres carrés
   (Performance : souvent mauvaise car bonne en dans l'ensemble mais mauvaise en tous points)

Section 3.5.1 « Interpolation Methods » du livre « Dynamic Economics » d' Adda et Cooper.

Pour approfondir

Méthodes d'integration numériques

1. Méthode de quadrature

2. Approximation d'un processus AR(1) par une chaîne de Markov (on en rediscutera au cours I.5)

Section 3.5.2 « Numerical Integration » du livre « Dynamic Economics » d' Adda et Cooper.