

# ECN 6338 Cours 12 et 13

## La Programmation Dynamique

William McCausland

2025-11-21

# Un problème d'optimisation dynamique déterministe à horizon fini

- ▶ Objectif : choisir  $(u_1, \dots, u_T)$  pour maximiser

$$\sum_{t=1}^T \pi(x_t, u_t, t) + W(x_{T+1}),$$

où

- ▶  $t$  est l'index du temps,
  - ▶  $x_t$  est l'état en période  $t$ ,  $t = 1, \dots, T + 1$ ,
  - ▶  $u_t$  est la décision (une variable de choix, control en anglais),
  - ▶  $\pi(\cdot)$  est le flux de valeur (souvent profit ou utilité),
  - ▶  $W(\cdot)$  est la valeur terminale.
- ▶ Contraintes :
    - ▶  $x_1$  donné,
    - ▶  $x_{t+1} = F(x_t, u_t, t)$ ,  $t = 1, \dots, T$ ,
    - ▶  $u_t \in D(x_t, t)$ ,  $t = 1, \dots, T$ .

# La version à horizon infini

- ▶ L'objectif devient

$$\sum_{t=0}^{\infty} \beta^t \pi(x_t, u_t).$$

- ▶ le temps commence à zéro,
  - ▶ il n'y a pas de valeur terminale,
  - ▶ cas très spécial de  $\pi(x_t, u_t, t)$  :  $\beta^t \pi(x_t, u_t)$ .
- ▶ Quant aux contraintes :
  - ▶  $F(x_t, u_t)$  au lieu de  $F(x_t, u_t, t)$  (pas vraiment une restriction),
  - ▶  $u_t \in D(x_t)$  au lieu de  $D(x_t, t)$ ,
  - ▶  $x_0$  donnée au lieu de  $x_1$ .

# Un exemple : accumulation de la richesse

- Le modèle :

$$\max_{c_t} \sum_{t=0}^{\infty} \beta^t u(c_t) \quad \text{s.c.} \quad k_{t+1} = F(k_t) - c_t, \quad k_0 \text{ donné},$$

- Notes :

- L'état est le stock du capital :  $x_t = k_t$
- La décision est la consommation :  $u_t = c_t$
- $F$  est une fonction de production d'un bien homogène (pas de distinction entre le capital et le seul bien de consommation).
- Cas spécial important :  $F(k) = (1 - \delta)k + f(k)$ , avec amortissement du capital.
- Une interprétation du problème : celui du planificateur central dans un modèle simple de croissance néoclassique : il y a un équilibre concurrentiel qui donne le même résultat.

## Autres exemples

- ▶ Problème dynamique d'un monopôle (Judd, page 404) :
  - ▶ la décision est la quantité de travail  $l$  et le dividende  $c$
  - ▶ l'état est la quantité de capital  $k$  dans la firme
  - ▶ l'état prochain est  $k$  plus le profit (comme fonction de  $l$  et  $k$ ) moins le dividende
  - ▶ le flux de valeur est le dividende versé durant une période
- ▶ Problème d'inventaire agricole (Judd, page 428)
  - ▶ la décision est la quantité d'un intrant de production agricole et la quantité du produit à consommer
  - ▶ l'état est l'inventaire agricole
  - ▶ l'état prochain est l'état actuel plus la production moins la consommation
  - ▶ le flux de valeur est l'utilité de la consommation d'une période
- ▶ Problème de gestion d'un forêt
  - ▶ la décision est la quantité récoltée de chaque type (âge) d'arbre,
  - ▶ l'état est la quantité de chaque type d'arbre,
  - ▶ l'état prochain est déterminé par la croissance des arbres et les quantités récoltées,
  - ▶ le flux de valeur est le profit apporté par la récolte d'une période

## La fonction de valeur

La fonction de valeur est comme une fonction d'utilité indirecte :

- ▶ Soit  $u(x, y)$  la fonction d'utilité pour deux biens, en quantités  $x$  et  $y$ .
- ▶ La contrainte budgétaire est  $p_x x + p_y y = m$ .
- ▶ La fonction d'utilité indirecte est

$$v(p_x, p_y, m) = \max_{x, y} u(x, y) \quad \text{s.c.} \quad p_x x + p_y y = m.$$

La fonction de valeur pour le problème à horizon fini est

$$V(x, t) = \sup_{u_t, \dots, u_T} \sum_{s=t}^T \pi(x_s, u_s, s) + W(x_{T+1}),$$

sous les contraintes

- ▶  $x_t = x$ ,
- ▶  $x_{s+1} = F(x_s, u_s, s)$ ,  $s = t, \dots, T$ ,
- ▶  $u_s \in D(x_s, s)$ ,  $s = t, \dots, T$ .

## Le principe d'optimalité de Bellman

*Une commande (décision) optimale  $u_t, \dots, u_T$  a la propriété que quelque soit l'état initial  $x_t$  et la décision initiale  $u_t$ , les décisions restantes  $(u_{t+1}, \dots, u_T)$  doivent être une commande optimale pour l'état  $x_{t+1} = F(x_t, u_t, t)$  résultant de la décision  $u_t$ .*

Notes :

- ▶ Une relation entre fonctions de valeur de différentes périodes :

$$V(x, t) = \sup_{u \in D(x, t)} \pi(x, u, t) + V(F(x, u, t), t + 1).$$

- ▶ Si  $u \in D(x, t)$  atteint le sup, la fonction de décision est

$$U(x, t) = \arg \max_{u \in D(x, t)} \pi(x, u, t) + V(F(x, u, t), t + 1).$$

- ▶  $U(x, t)$ ,  $V(x, t)$  sont des fonctions seulement de  $x$  et  $t$ .
- ▶ La valeur de  $x$  résume toute la mémoire du passé.

## Trouver la fonction de valeur $V(x, 1)$

- ▶ Une condition terminal :  $V(x, T + 1) = W(x)$ , une fonction connue.
- ▶ Les autres  $V(\cdot, t)$  par raisonnement à rebours : l'application de l'opérateur défini par

$$V(x, t) = \sup_{u \in D(x, t)} \pi(x, u, t) + V(F(x, u, t), t + 1).$$

- ▶ Attention : l'opérateur prend une fonction de  $x$ ,  $V(x, t + 1)$ , et donne une fonction de  $x$ ,  $V(x, t)$ . La valeur de  $t$  est fixe et les fonctions  $\pi(\cdot)$  et  $F(\cdot)$  sont primitives.



## Le problème déterministe à horizon infinie

- ▶ La fonction de valeur est maintenant définie de façon récursive :

$$V(x) = \sup_{u \in D(x)} \pi(x, u) + \beta V(F(x, u)),$$

ou  $V = TV$ ;  $V$  est une fonction de  $x \in X$ ,  $T$  est un opérateur.

- ▶ L'équation est une équation *fonctionnelle*, sa solution est  $V$ , la *fonction* de valeur.
- ▶ Une fonction de décision est maintenant une fonction  $U(x)$  qui vérifie

$$U(x) \in \arg \max_{u \in D(x)} \pi(x, u) + \beta V(F(x, u)).$$

# Le théorème des applications contractantes et l'itération de la fonction de valeur

- ▶ Le théorème des applications contractantes (contraction mapping theorem) : si  $0 < \beta < 1$ ,  $\pi(x, u)$  est borné et  $X$  est compacte, alors
  - ▶  $T$  est monotone ( $V_1 \geq V_2 \Rightarrow TV_1 \geq TV_2$ )
  - ▶  $T$  est une contraction de module  $\beta$   
( $\|TV_1 - TV_2\| \leq \beta\|V_1 - V_2\|$ )
  - ▶  $V = TV$  a une solution ( $T$  a un point fixe) unique.
- ▶ L'itération de la fonction de valeur donne une suite de fonctions de valeur  $V^l$  qui converge vers  $V$  :  $V^{l+1} = TV^l$ , ou

$$V^{l+1}(x) = \sup_{u \in D(x)} \pi(x, u) + \beta V^l(F(x, u)).$$

- ▶ Un sous-produit est une suite de fonctions de décision  $U^l$  correspondantes:

$$U^{l+1}(x) \in \arg \max_{u \in D(x)} \pi(x, u) + \beta V^l(F(x, u)).$$

## Comment appliquer l'opérateur en pratique

- ▶ Prenons l'exemple de l'accumulation de richesse.
- ▶ La fonction de valeur est un point fixe de l'opérateur  $T$ , où

$$(TV)(k) \equiv \max_{0 \leq c \leq F(k)} u(c) + \beta V(F(k) - c).$$

- ▶ Comment trouver une approximation de  $V(k)$ ?
- ▶ Une méthode : permettre seulement les valeurs de  $k$  sur une grille  $K = \{k^m, \dots, k^M\}$ .
- ▶ En pratique il faut prendre  $k^+ \equiv F(k) - c$ , le capital à la prochaine période, comme la variable de décision.
- ▶ L'équation Bellman avec le changement de variables :

$$V(k) = \max_{0 \leq k^+ \leq F(k)} u(F(k) - k^+) + \beta V(k^+).$$

- ▶ L'équation Bellman avec  $V: K \rightarrow \mathbb{R}$  au lieu de  $V: \mathbb{R}_+ \rightarrow \mathbb{R}$  :

$$V(k) = \max_{k^+ \in K, k^+ \leq F(k)} u(F(k) - k^+) + \beta V(k^+).$$

- ▶ Elle est un système d'équations non-linéaire ; la solution est un vecteur  $(V(k^m), \dots, V(k^M))$  qui vérifie les  $|K|$  équations.

## Exemple (page 425)

Prenons :

- ▶  $\beta = 0.96$
- ▶  $u(c) = c^{\gamma+1}/(\gamma+1)$ ,  $\gamma = -2$ .
- ▶  $F(k) = k + f(k)$ , où  $f(k) = \frac{1-\beta}{\alpha\beta} k^\alpha$ ,  $\alpha = 0.25$
- ▶  $k^m = 0.8$ ,  $k^M = 1.2$ ,  $\kappa = 0.001$ , alors

$$K = \{0.800, 0.801, 0.802, \dots, 1.200\}.$$

Notes :

- ▶  $k^{ss}$  est un état *stationnaire* si  $k^{ss} + f(k^{ss}) - U(k^{ss}) = k^{ss}$ , où  $U(\cdot)$  est la fonction de décision optimale.
- ▶  $\beta$  dans la fonction de production est étrange, le coefficient  $(1 - \beta)/(\alpha\beta)$  fait en sorte que  $k^{ss} = 1$ .

## Choisir une fonction de valeur initiale

- ▶ Distinguez  $U_c(k)$ , la fonction de décision qui donne la consommation actuelle, et  $U_k(k)$ , celle qui donne le capital prochain.
- ▶ Notez que  $U_c(k) = f(k)$  donne  $k^+ = k$  ou  $U_k(k) = k$ .
- ▶ Cette stratégie est faisable mais généralement sous-optimale lorsque  $k \neq k^{ss} = 1$ .
- ▶ Autour de  $k = k^{ss}$ ,  $U_c(k)$  et  $U_k(k)$  devraient être près des fonctions de décision optimales.
- ▶ La fonction de valeur (non-optimale)  $V^c(k)$  associée à  $U_c(k) = f(k)$  (non-optimale) vérifie

$$V^c(k) = u(f(k)) + \beta V^c(k), \text{ ou } V^c(k) = u(f(k))/(1 - \beta).$$

- ▶ Notez que  $V^c(k) \leq V(k)$ , où  $V(k)$  est la fonction de valeur associée à la fonction de décision optimale.
- ▶ Considérez aussi  $V_z(k) = 0$  comme fonction de valeur initiale.
- ▶ Dans le code suivant,  $U$  signifie  $U_k$ , le choix de capital comme fonction de capital.

## Code initial : préférences et technologies

```
# Valeurs des paramètres
gamma <- -2.0      # Paramètre de préférence
beta  <- 0.96      # Paramètre d'impatience
alpha <- 0.25      # Paramètre de production

# Fonction d'utilité
u <- function(c) {
  ifelse(c>0, c^(gamma + 1)/(gamma + 1), -Inf)
}

# Fonction de production
f <- function(k) {
  ((1-beta)/(alpha*beta)) * k^alpha
}
```

## Code initial : grille et précomputation

```
# Grille de capital
kappa <- 0.001
k <- seq(0.5, 1.5, by=kappa)
N <- length(k)

# Matrice de valeurs de  $u(k + f(k) - k+)$ 
u_k <- function(k, kplus) {
  u(k + f(k) - kplus)
}
u_tab <- outer(k, k, u_k)

# Fonction de valeur initiale, épargne zéro
V_c <- u(f(k))/(1-beta)

# Fonction de valeur initiale, zéro
V_z <- rep(0, N)
```

## Code, itération de la fonction de valeur

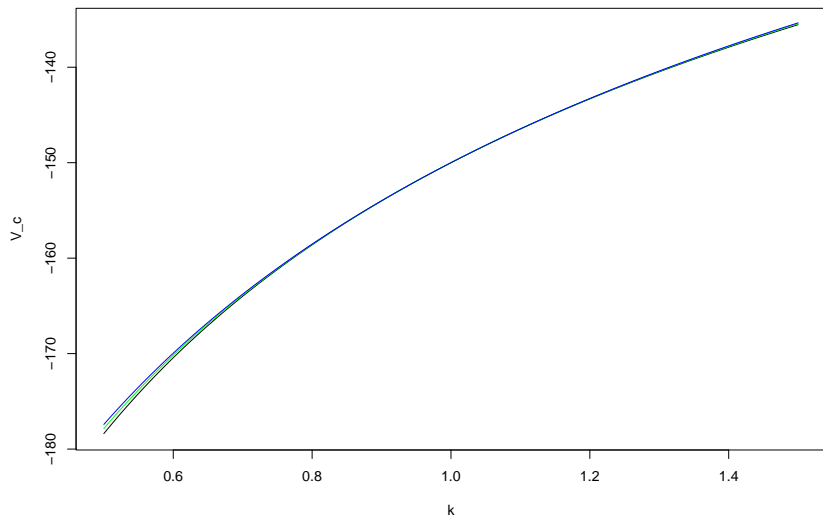
```
# Iteration de la fonction de valeur
VU_suiv <- function(V) {
  V_suiv <- rep(0, N);
  U_suiv <- rep(0, N); U_suiv_i <- vector('integer', N)
  for (i in 1:N) {
    # k[i_plus] est la décision optimale à l'état k[i]
    i_plus = which.max(u_tab[i,] + beta*V)
    U_suiv[i] = k[i_plus]; U_suiv_i[i] = i_plus
    # V_suiv[i] est la fonction V suivante à l'état k[i]
    V_suiv[i] = u_tab[i, i_plus] + beta*V[i_plus]
  }
  list(V=V_suiv, U=U_suiv, U_i=U_suiv_i)
}

# Deux itérations de T à partir de V_c
VUs_c <- VU_suiv(V_c); VUss_c <- VU_suiv(VUs_c$V)
```



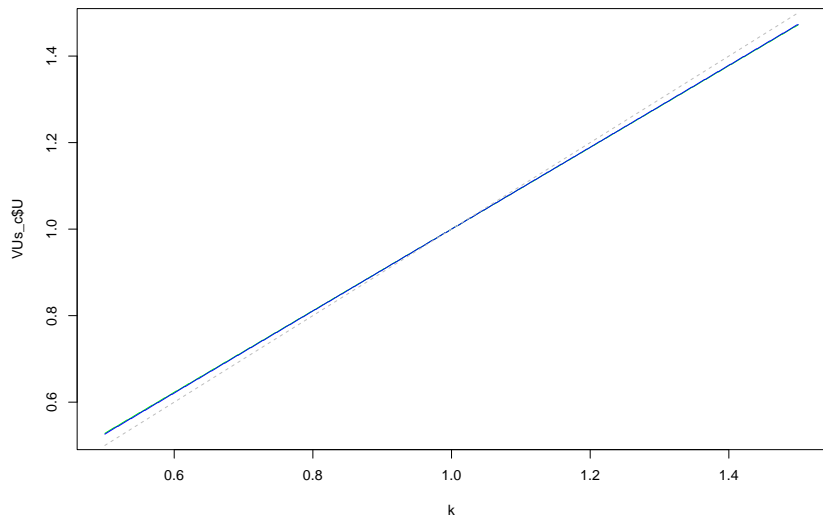
## La fonction de valeur $TV_c(k)$ (deux itérations)

```
plot(k, V_c, type='l')  
lines(k, VUs_c$V, col='green'); lines(k, VUss_c$V, col='blue')
```



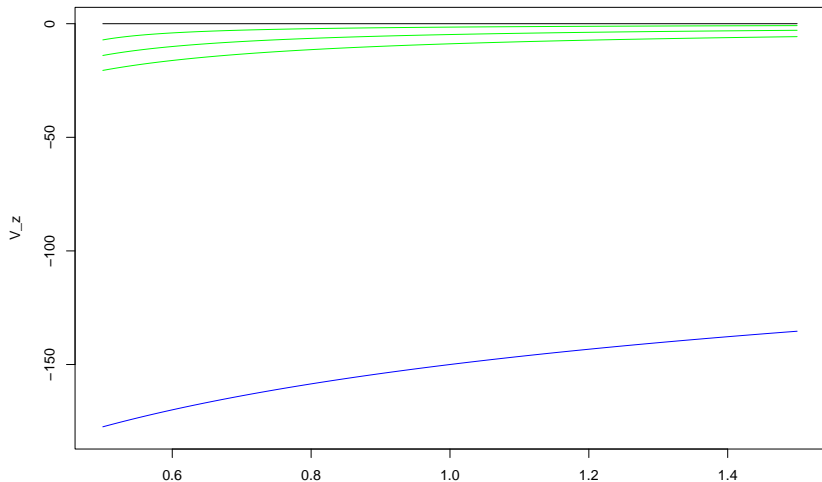
## La fonction de décision après deux itération de $V_c(k)$

```
plot(k, VUs_c$U, type='l', col='green'); lines(k, VUss_c$U,  
lines(k, k, lty='dashed', col='grey');
```



## Les fonctions de valeur $T^i V_z(k)$ , $i = 1, 2, 3$

```
plot(k, V_z, type='l', ylim = c(-180, 0)); V = V_z
for (i in 1:3)
  { V <- VU_suiv(V)$V; lines(k, V, col='green') }
lines(k, VUss_c$V, col='blue')
```



# Deux types d'itération

Les deux types d'itération :

1. Itération de la fonction de valeur : ce qu'on a déjà vu
2. Itération de la fonction de décision : la matière prochaine

Pour faire l'itération de la fonction de décision, il faut pouvoir calculer une approximation de la valeur de cette fonction de décision comme fonction de l'état : pour une fonction de décision

$U: K \rightarrow \mathbb{R}$  donnée, il faut approximer sa valeur  $V^U: K \rightarrow \mathbb{R}$ .

## Approximation de $V^U$ pour $U_k$ donnée

- Pour une fonction de décision  $k^+ = U_k(k)$ , pas forcément optimale, on peut calculer, pour chaque  $k_0 \in K$ , la suite d'état

$$k_0, U_k(k_0), U_k^{(2)}(k_0), U_k^{(3)}(k_0), \dots,$$

où  $U_k^{(0)}(k_0) = k_0$ ,  $U_k^{(i+1)}(k_0) = U_k(U_k^{(i)}(k_0))$ .

- On peut approximer la fonction de valeur correspondante, le vecteur  $V^U(k_0)$ ,  $k_0 \in K$  par  $\hat{V}^U(k_0)$  défini comme

$$\sum_{t=0}^T \beta^t u(U^{(t)}(k_0) + f(U^{(t)}(k_0)) - U^{(t+1)}(k_0)) + \beta^{T+1} \tilde{V}(U^{(T+1)}(k_0)),$$

où  $\tilde{V}$  est une approximation de la fonction de valeur optimale.

- Si on fait ce calcul direct pour chaque  $k_0$  il peut y avoir des calculs redondants; pire, le calcul direct ne fonctionne pas pour les problèmes analogues stochastiques.

## La version rétrograde de l'approximation de $V^U$

Cette version rétrograde évite des calculs redondants et se généralise aux problèmes stochastiques, équation (12.4.2) de Judd :

- ▶  $W^0 = V$  (un vecteur, une approximation initiale de  $V^U$ )
- ▶ Pour  $j = 0, \dots, T$  :
  - ▶ Pour  $k \in K$ ,

$$W^{j+1}(k) = u(k + f(k) - U(k)) + \beta W^j(U(k)).$$

- ▶  $V^U \approx \hat{V}^U \equiv W^{T+1}$  (un vecteur, l'approximation finale de  $V^U$ )

## Code, approximation de $V^U$ , version rétrograde

```
# Approximation de  $V^U$ , version rétrograde
VdeU <- function(VU, n_iter) {
  # VU$V est W0
  # VU$U est la fonction de décision qui donne  $k^+$ 
  # VU$Ui donne l'index de  $k^+$ 
  V <- VU$V; V_suiv = rep(0, N);
  for (i in 1:n_iter) {
    for (i in 1:N) {
      Ui = VU$U_i[i]
      V_suiv[i] = u_tab[i, Ui] + beta*V[Ui]
    }
    V <- V_suiv
  }
  V
}
```

# Itération de la fonction de décision

Itérez les étapes suivantes jusqu'à ce que  $\|V^{l+1} - V^l\| < \epsilon$  :

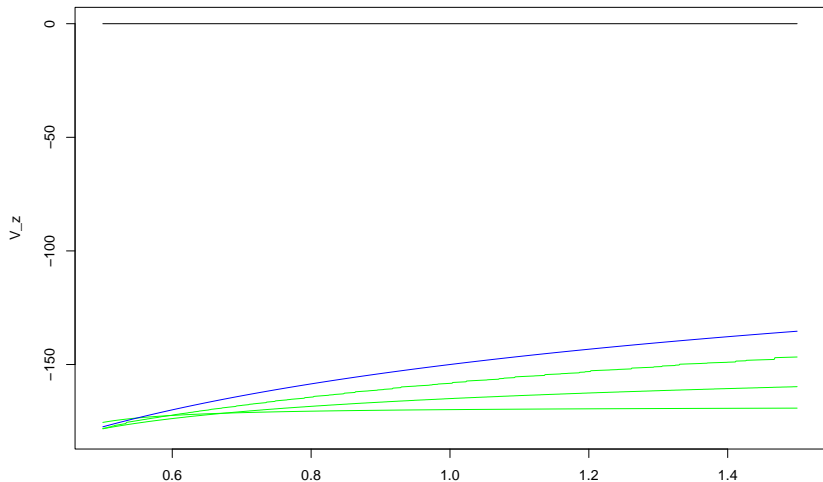
1. Utilisez  $V^l$  pour calculer  $U^{l+1}$ , la fonction de décision obtenue comme sous-produit de l'itération de la fonction de valeur.
2. Utilisez  $U^{l+1}$  et  $V^l$  (pour la valeur résiduelle à  $t = T + 1$ ) pour calculer  $V^{l+1}$  comme

$$V^{l+1} \equiv \hat{V}^{U^{l+1}}$$



## Iteration de la fonction de décision (graphique)

```
plot(k, V_z, type='l', ylim = c(-180, 0)); V <- V_z
for (i in 1:3)
  { VU <- VU_suiv(V); V <- VdeU(VU, 100); lines(k, V, col=
lines(k, VUss_c$V, col='blue')}
```



# Problèmes dynamiques avec incertitude

- ▶ Soit  $X$  l'ensemble d'états possibles.
- ▶ Rappel, évolution des états dans les modèles déterministes :

$$x_{t+1} = F(x_t, u_t, t) \quad \text{ou} \quad x_{t+1} = F(x_t, u_t).$$

- ▶ Dans les modèles stochastiques, l'évolution de l'état devient stochastique :

$$F(A, x_t, u_t, t) \equiv \Pr[x_{t+1} \in A | x_t, u_t, t], \quad A \subseteq X.$$

- ▶ Le problème général est la maximisation de

$$E \left[ \sum_{t=1}^T \pi(x_t, u_t, t) + W(x_{T+1}) \right] \quad \text{ou} \quad E \left[ \sum_{t=0}^{\infty} \beta^t \pi(x_t, u_t) \right].$$

sous les contraintes décrites en termes de  $F(\cdot)$ ,  $D(\cdot)$  et  $x_0$ .

- ▶ Notez la séparabilité et la linéarité :
  - ▶ temporelle dans les problèmes déterministes et stochastiques,
  - ▶ l'espérance dans les problèmes stochastiques.

# Structure intertemporelle avec incertitude

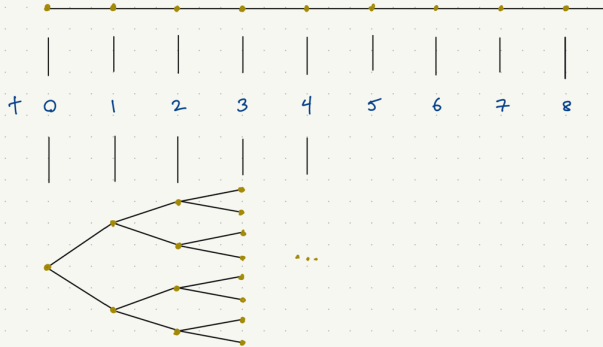


Figure 1: Deux structures avec autosimilarité

# Les équations de Bellman dans le cas stochastique

Horizon fini :

$$V(x, t) = \sup_{u \in D(x_t, t)} \pi(x, u, t) + E[V(x_{t+1}, t+1) | x_t = x, u_t = u],$$

$$V(x, T+1) = W(x).$$

Horizon infini :

$$V(x) = \sup_{u \in D(x)} \pi(x, u) + \beta E[V(x^+) | x, u].$$

## Exemple, modèle de recherche de McCall

- ▶ Une simplification du modèle de McCall (1970) dans “Quantitative Economics with Python” par Sargent et Stachurski, section [Job Search I: The McCall Search Model](#)
- ▶ L'état est la réalisation  $w_t$  d'une offre de salaire aléatoire.
- ▶ La densité  $q(w_t|w_{t-1}, u_{t-1}) = q(w_t)$  ne dépend pas de  $w_{t-1}$ ,  $u_{t-1}$ . Soit  $W$  l'ensemble de valeurs possibles des  $w_t$ .
- ▶ L'action  $u_t$  est d'accepter ( $u_t = 1$ ) l'offre  $w_t$  et gagner  $w_t$  chaque période à tout jamais ou de rejeter ( $u_t = 0$ ) l'offre, gagner  $c$  à  $t$  et attendre une autre offre.
- ▶ L'agent maximise

$$E \left[ \sum_{t=0}^{\infty} \beta^t y_t \right],$$

où  $y_t$  dépend du moment  $t^*$  où l'agent accepte l'offre :

$$y_t \equiv \begin{cases} c & t < t^* \\ w_{t^*} & t \geq t^* \end{cases}$$

# L'équation de Bellman

L'équation de Bellman est (pour  $W$  fini)

$$\begin{aligned} V(w) &= \max \left\{ \frac{w}{1-\beta}, c + \beta E[V(w')] \right\} \\ &= \max \left\{ \frac{w}{1-\beta}, c + \beta \sum_{v \in W} V(v) q(v) \right\}. \end{aligned}$$

Notes :

- ▶ Le terme  $c + \beta E[V(w')]$  ne dépend pas de  $w$ , ce qui simplifie énormément le problème.
- ▶ L'action optimale  $\sigma(w) \in \{0, 1\}$  doit être de la forme où l'agent accepte une offre plus grande ou égale à  $\bar{w}$  et rejette une offre moins grande que  $\bar{w}$ .
- ▶ Le salaire de réserve  $\bar{w}$  est  $(1 - \beta)(c + \beta E[V(w')])$ .
- ▶ Intuition :  $\bar{w}$  devrait être croissant en  $\beta$  et  $c$ .

# La spécification des détails du problème

La spécification :

- ▶ Valeurs des paramètres :  $\beta = 0.99$ ,  $c = 25$ .
- ▶ Loi de  $w$  :  $w - 10 \sim \text{Bb}(50, 200, 100)$  (beta-binomial)
- ▶  $k \equiv w - 10$  est un mélange :  $\pi \sim \text{Be}(200, 100)$ ,  
 $k|\pi \sim \text{Bi}(50, \pi)$ .

```
# Valeurs des paramètres
```

```
beta <- 0.99
```

```
c <- 25
```

```
# Loi de w : valeurs w, probabilités q
```

```
w <- 10:60
```

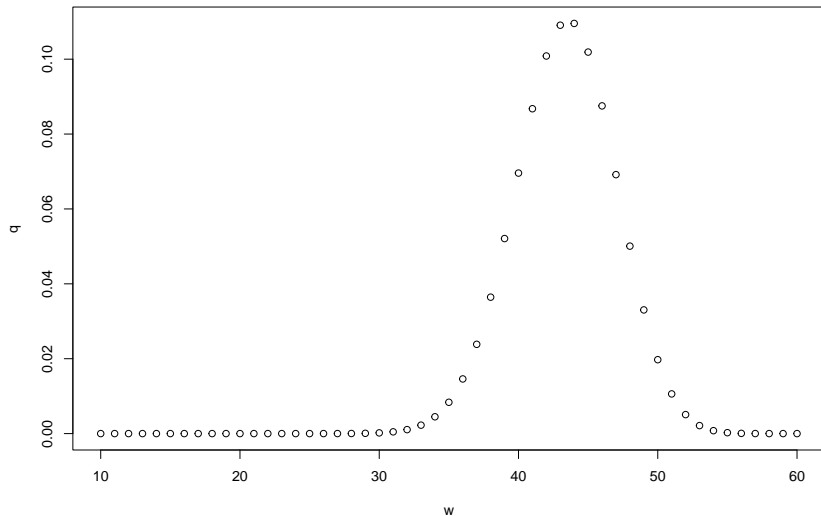
```
n = 50; a <- 200; b <- 100; k <- w-10
```

```
lnq <- lchoose(n, k) + lbeta(a+k, b+n-k) - lbeta(a, b)
```

```
q <- exp(lnq)
```

# La loi discrète des offres de salaire

```
plot(w, q, type='p')
```





# L'itération de la fonction de valeur

- ▶ Dans un premier temps, on ne profite pas de la structure du problème.
- ▶ On commence par la fonction de valeur qui correspond à la fonction de décision  $\sigma_0(w) = 1$ , selon laquelle on accepte chaque offre :

$$V^{\sigma_0}(w) = \frac{w}{1 - \beta}.$$

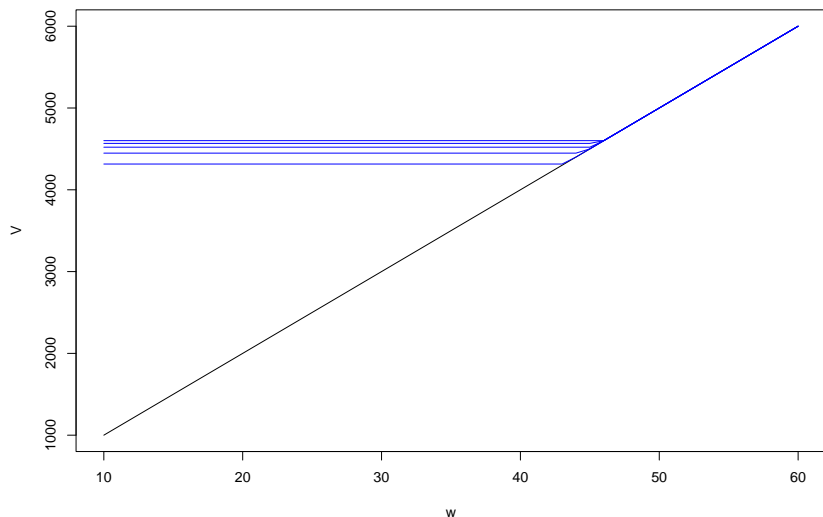
- ▶ On applique l'opérateur  $T$  de façon répétitive jusqu'à la convergence, où

$$(TV)(w) = \max \left\{ \frac{w}{1 - \beta}, c + \beta \sum_{v \in W} V(v)q(v) \right\}.$$

```
library(geometry)
V0 <- w/(1-beta)
T <- function(V) {pmax(w/(1-beta), c+beta*dot(V, q))}
```

## L'iteration de la fonction de valeur (graphique)

```
V = V0; plot(w, V, type='l')  
for (i in 1:5) {V = T(V); lines(w, V, col='blue')}
```



## L'itération du salaire de réserve

- ▶ On définit  $h = \bar{w}/(1 - \beta) = c + \beta E[V(w')]$ .
- ▶ En termes de  $h$ , l'équation de Bellman s'écrit

$$V(w) = \max \left\{ \frac{w}{1 - \beta}, h \right\}.$$

- ▶ Si on substitue  $V(w)$  dans la définition de  $h$ , on obtient

$$h = c + \beta E \left[ \max \left\{ \frac{w'}{1 - \beta}, h \right\} \right].$$

- ▶ On peut résoudre cette équation non-linéaire avec l'itération

$$h' = c + \beta \sum_{v \in W} \max \left\{ \frac{v}{1 - \beta}, h \right\} q(v).$$

- ▶ C'est une équation scalaire, donc plus facile à résoudre numériquement, bien qu'elle nécessite le calcul d'un produit intérieur par itération.

## L'itération du salaire de réserve

```
wbar <- function(beta_c, tol) {  
  beta <- beta_c[1]; c <- beta_c[2]  
  h <- c / (1-beta)  
  repeat {  
    h_plus <- c + beta * dot(pmax(w/(1-beta), h), q)  
    if (abs(h - h_plus) < tol) break  
    h <- h_plus  
  }  
  wbar <- (1-beta) * h  
}
```

Code pour une graphique de  $\bar{w}$  comme fonction de  $\beta, c$ .

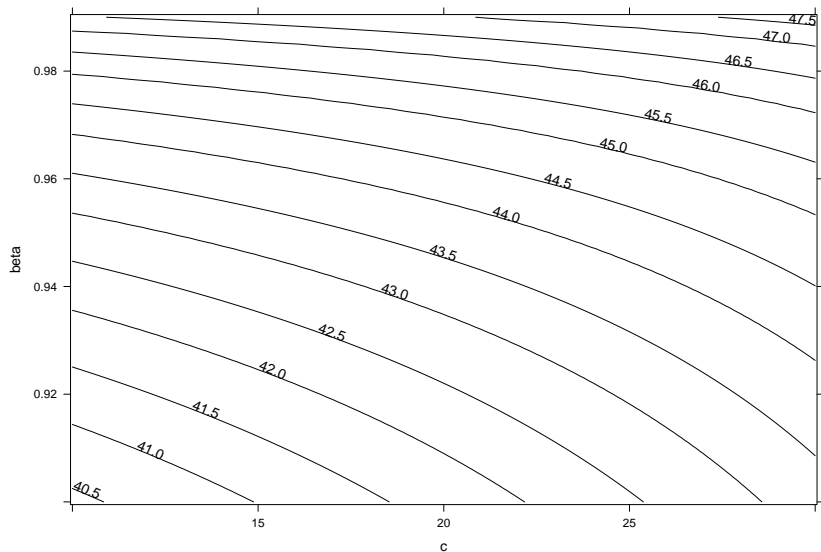
```
# Grilles de points pour beta et c
beta_gr <- seq(0.9, 0.99, by=0.001)
c_gr <- seq(10, 30, by=0.1)

# Tableau des pairs (beta, c)
bc <- as.matrix(expand.grid(beta_gr, c_gr))
colnames(bc) <- c('beta', 'c')

# Évaluer la fonction wbar à chaque pair (beta, c)
wbar_c <- apply(bc, 1, wbar, tol = 1e-6)
df <- data.frame(wbar_fn = wbar_c, bc)
```

## Salaire de réserve comme fonction de $\beta$ et $c$

```
library(lattice)
contourplot(wbar_fn ~ c*beta, data=df, cuts=12)
```



# Un modèle avec séparation

- ▶ Un autre modèle du type McCall au site QuantEcon
- ▶ [Job Search II: Search and Separation](#)
- ▶ L'agent entre dans la période  $t$ 
  - ▶ ou avec emploi à salaire  $w_e$ , auquel cas l'agent reçoit le flux d'utilité  $u(w_e)$  et ensuite est congédié avec probabilité  $\alpha$ .
  - ▶ ou sans emploi, auquel cas l'agent reçoit une offre  $w_t$  et choisit entre travailler au salaire  $w_t$  (jusqu'au congédiement) ou recevoir  $c$  et rester sans emploi au début de la prochaine période. Le flux d'utilité est  $u(w_t)$  ou  $u(c)$ , respectivement.
- ▶ La fonction d'utilité est

$$E \left[ \sum_{t=0}^{\infty} \beta^t u(y_t) \right].$$

- ▶ Soit  $v(w_e)$  l'utilité espérée d'un agent avec emploi au début de la période, au salaire  $w_e$ .
- ▶ Soit  $h(w)$  l'utilité espérée d'un agent sans emploi qui reçoit une offre  $w$  au début de la période.

## Une autre perspective

- ▶ L'état est  $(e, w)$ , où
  - ▶  $e \in \{0, 1\}$ , ( $e = 1$  veut dire avec emploi;  $e = 0$ , sans emploi)
  - ▶  $w$  est le salaire, interprété comme
    - ▶ le salaire d'emploi déjà accepté ( $w_e$ ) dans le passé si  $e = 1$ ,
    - ▶ l'offre de salaire actuel si  $e = 0$ .
- ▶ On peut définir une fonction  $V(e, w)$  comme

$$V(e, w) = \begin{cases} v(w) & e = 1 \\ h(w) & e = 0 \end{cases}$$

- ▶ Notez que l'agent a un choix (accepter ou rejeter l'offre) seulement quand  $e = 0$ .
- ▶ On n'a pas besoin d'une notation différente pour  $w_e$  (salaire actuel) et  $w$  (offre de salaire), l'interprétation est déterminée par  $e$ .



# Les équations de Bellman

Les deux équations tiennent de façon simultanée :

$$v(w_e) = u(w_e) + \beta \left[ (1 - \alpha)v(w_e) + \alpha \sum_{w' \in W} h(w')q(w') \right],$$

$$h(w) = \max \left\{ v(w), u(c) + \beta \sum_{w' \in W} h(w')q(w') \right\}.$$

Encore une fois, une reformulation simplifie l'analyse :

$$d \equiv E[h(w')] = \sum_{w' \in W} h(w')q(w').$$

On peut écrire maintenant (pas besoin de distinguer  $w$  et  $w_e$ )

$$v(w) = u(w) + \beta[(1 - \alpha)v(w) + \alpha d],$$

$$h(w) = \max\{v(w), u(c) + \beta d\},$$

$$d = \sum_{w' \in W} h(w')q(w') = \sum_{w' \in W} \max\{v(w'), u(c) + \beta d\}q(w').$$

# Notes sur le problème

- L'équation pour  $h(w)$  étant éliminée, on peut itérer les étapes suivantes jusqu'à ce que tienne une condition de tolérance :

$$d_{n+1} = \sum_{w' \in W} \max\{v_n(w'), u(c) + \beta d_n\} q(w'),$$

$$v_{n+1}(w) = u(w) + \beta[(1 - \alpha)v_n(w) + \alpha d_n].$$

- Le salaire de réserve  $\bar{w}$  est maintenant la solution de l'équation

$$v(\bar{w}) = u(c) + \beta d.$$

# La spécification du problème

- La fonction d'utilité :

$$u(c) = \frac{c^{1-\sigma} - 1}{1 - \sigma}$$

- Les valeurs des paramètres sont :  $\alpha = 0.2$ ,  $\beta = 0.98$ ,  $\sigma = 2$ ,  $c = 6$ .
- Loi de  $w$  :  $\frac{59}{10}(w - 10) \sim \text{Bb}(59, 600, 400)$  (beta-binomial)

```
# Valeurs des paramètres
```

```
alpha <- 0.2; beta <- 0.98; sigma <- 2; c <- 6
```

```
# Loi de w : valeurs w, probabilités q
```

```
w <- seq(10, 20, length=60)
```

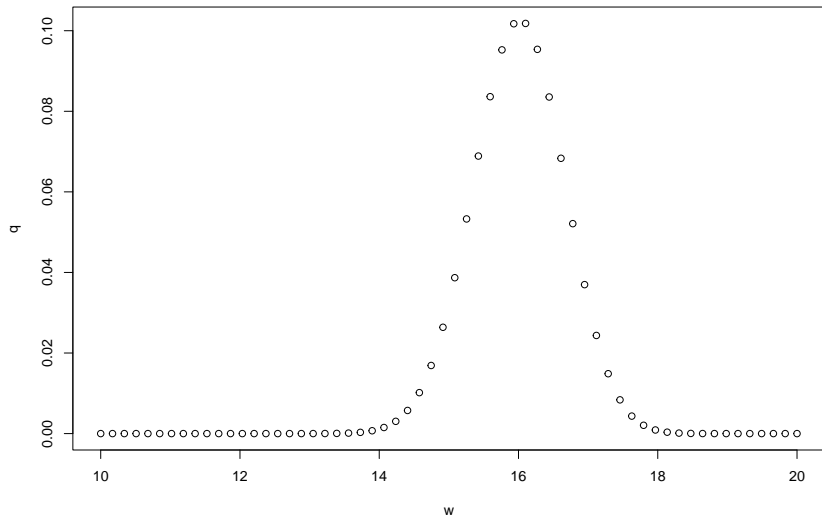
```
n = 59; a <- 600; b <- 400; k <- (59/10)*(w-10)
```

```
lnq <- lchoose(n, k) + lbeta(a+k, b+n-k) - lbeta(a, b)
```

```
q <- exp(lnq)
```

# La loi discrète des offres de salaire

```
plot(w, q, type='p')
```



# L'itération en code

```
# Fonction d'utilité et des évaluations préliminaires
u <- function(w, sigma) { (w^(1-sigma)-1) / (1-sigma) }
uw = u(w, sigma)
uc = u(c, sigma)

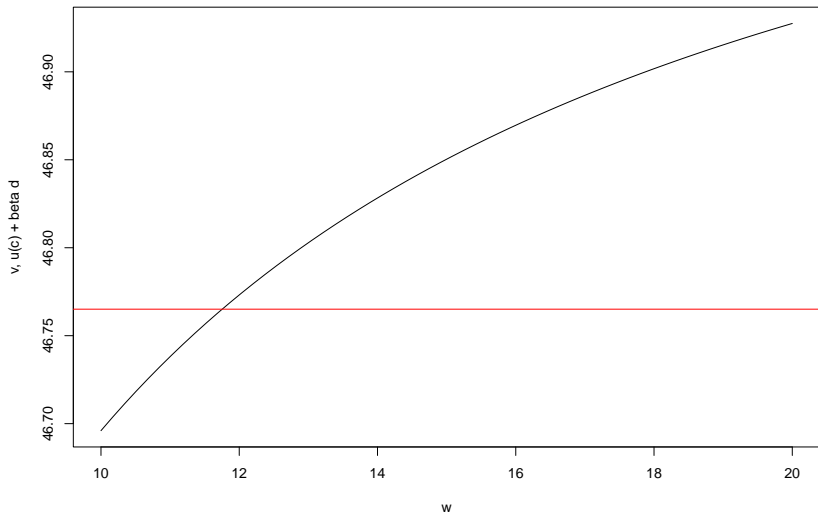
# Itération de Bellman
T <- function(vd) {
  v_plus <- uw + beta * ((1-alpha) * vd$v + alpha * vd$d)
  d_plus <- sum(pmax(vd$v, uc + beta * vd$d) * q)
  list(v = v_plus, d = d_plus)
}
```

Trouver la fonction de valeur  $v(w)$  et  $d \equiv E[h(w')]$

```
solve_model <- function(tol=1e-5, max_iter=2000) {  
  vd <- list(v = rep(1, length(w)), d=1)  
  for (i in 1:max_iter) {  
    Tvd <- T(vd)  
    err1 <- max(abs(vd$v - Tvd$v))  
    err2 <- abs(vd$d - Tvd$d)  
    if (max(err1, err2) < tol)  
      break  
    vd <- Tvd  
  }  
  Tvd  
}  
  
vd <- solve_model()
```

## Graphique de la fonction de valeur $v(w)$ et de $u(c) + \beta d$

```
plot(w, vd$v, type='l', xlab='w', ylab='v, u(c) + beta d')  
abline(h = uc + beta*vd$d, col='red') # h pour horizontal!
```



# Interprétation de la graphique

- ▶  $v(w)$ , la valeur d'avoir un emploi avec salaire  $w$ , est en noir.
- ▶  $u(c) + \beta d$ , la valeur de réservation, est en rouge.
- ▶ L'intersection  $(\bar{w}, u(c) + \beta d)$  donne le salaire de réservation  $\bar{w}$ .
- ▶  $h(w)$ , la valeur d'avoir une offre de  $w$  en état de chômage est le maximum des deux fonctions :

$$h(w) = \max(v(w), u(c) + \beta d).$$

- ▶ Notez que  $u(c) + \beta d$  est une constante et (dans l'équation) une fonction constante.
- ▶ Un agent rationnel qui a optimisé dans le passé ne devrait pas se trouver dans l'état  $(1, w)$ , si  $w < \bar{w}$ .
- ▶ Attention : l'argument `h` dans la commande `abline` est pour spécifier la position d'une droite horizontale.