



## Application Chapitre 5 (corrigé)

### Un problème de partage

Dans ce document, je propose une correction de l'application du Chapitre 5 de mon manuel de Macroéconomie quantitative (p.74). Les fonctions et les données utiles sont disponibles sur mon site <sup>a</sup>. Un ou plusieurs scripts Octave/Matlab accompagnent également ce document, dont une partie des éléments qui sont repris et commentés ici, apparaissent sous une police à chasse fixe.

a. [www.christophecahn.fr/macroquant](http://www.christophecahn.fr/macroquant)

### Problème d'allocation intertemporelle

On considère le problème d'allocation intertemporelle suivant :

$$\max_{\{c_t, w_{t+1}\}_{t=0}^{\infty}} E_0 \sum_{t=0}^{\infty} \beta^t u(c_t)$$

sous la contrainte  $w_{t+1} \leq w_t - c_t + y_t$  pour  $t = 0, 1, 2, \dots$  et  $w_0$  donné. La dotation courante  $y_t$  suit un processus i.i.d. qui peut prendre deux valeurs, basse ( $y_b$ ) et haute ( $y_h$ ). Soit  $\pi_i$  la probabilité d'une réalisation  $y_i$ ,  $i = \{b, h\}$ .

**1a** Écrire le problème de manière récursive.

Premièrement, on peut noter qu'un individu qui fait face à ce problème d'allocation base sa décision sur sa richesse initiale,  $w$ , et la réalisation de sa dotation  $y$ . En conséquence, une façon direct de décrire l'état du système sur lequel l'individu prend sa décision est le couple  $s = (w, y)$ . Puisque la dotation est i.i.d., nous pouvons restreindre la variable d'état à la taille perçue de la dotation,  $X = w + y$ , après la réalisation du choc (cf le manuel d'Adda et Cooper). Soit  $V(s)$  la somme actualisée des flux futurs espérés d'utilité d'un individu dans l'état  $s$ . Soit  $\Gamma(s)$ , la correspondance représentative des contraintes qui s'appliquent aux choix de l'individu  $\Gamma(s = (w, y)) = \{s' = (w', y') / 0 \leq w' \leq w + y, y' \in \{y_b, y_h\}\}$ . Si les conditions idoines sont vérifiées, la solution au problème séquentiel est équivalente à la solution de l'équation fonctionnelle suivante

$$V(s) = \max_{s' \in \Gamma(s)} \{r(s, s') + \beta E[V(s')|s]\}$$

avec  $r(s, s') = u(w + y - w')$  (cf Stokey, Lucas, Prescott, Recursive Models in Economic Dynamics). Ceci correspond à la formulation récursive du problème initial.

On suppose que  $u(c) = c^{1-\gamma}/(1-\gamma)$ . Soit  $\beta = 0.95$ ,  $\gamma = 2$ ,  $y_l = 0.8$ ,  $y_h = 1.2$ ,  $\pi_l = 0.4$ ,  $\pi_h = 0.6$ . Résoudre le problème par itération sur la fonction valeur. Tracer la fonction valeur comme une fonction de la taille de la richesse.

L'équation fonctionnelle précédente définit une contraction  $T$ . En conséquence, nous pouvons itérer sur l'équation  $V_n = T(V_{n-1})$  à partir d'une condition initiale  $V_0$ , et s'attendre à une convergence vers la fonction valeur.

Nous allons utiliser la méthode de l'itération sur la fonction valeur (VFI) sur une grille discrete de l'espace état. Pour une limite supérieure  $\bar{w}$ , la taille (arbitraire) maximale atteignable pour la

richesse, nous définissons la grille  $\{w\}_1^n = (w_1 = 0 < \dots < w_i < \dots < w_n = \bar{w})$ , et l'état  $s_{ij} = (w_i, y_j)$ . En conséquence, la VFI a pour but de calculer la matrice  $\mathbf{V} = [V(s_{ij})]$ .

On définit deux  $n \times n$  matrices  $\mathbf{U}_k$  dont l'élément  $(i, j)$  est  $U_k(i, j) = u(w_i + y_k - w_j)$ ,  $i = 1, \dots, n$ ,  $j = 1, \dots, n$ ,  $k \in \{b, l\}$ . On définit également un opérateur  $T(v_k)$  qui associe à un vecteur  $\mathbf{v}_k$  à un autre vecteur  $\mathbf{T}\mathbf{v}_k$ . Ensuite, on itère sur l'équation sous la forme compacte

$$\mathbf{T}\mathbf{v}_k = \max \left\{ \mathbf{U}_k + \beta [\mathbf{v}_b \ \mathbf{v}_l] \begin{bmatrix} \pi_b & \dots & \pi_b \\ \pi_l & \dots & \pi_l \end{bmatrix} \right\}$$

Lorsque la convergence est atteinte, on note  $\mathbf{V} = [\mathbf{v}_b \ \mathbf{v}_l]$ .

**1b** Procédez à l'affectation des paramètres à partir des valeurs de l'énoncé et définissez la grille de richesse  $w$  qui prendra  $nw=500$  valeurs sur l'intervalle  $[0,2]$ .

```
gam = 2;
bet = 0.95;
y = [1.2 0.8];
pii = [0.6 0.4; 0.6 0.4];
ny = length(y);
nw = 500;
w_m = 2;
w = linspace(0, w_m, nw)';
```

**1c** On note  $ny=2$  le nombre d'état possible de la dotation. Calculer sous forme matricielle la matrice des flux d'utilité  $U$  pour différentes tailles de richesse et de dotation.  $U(i, j, k)$  prend la valeur  $U(w_i + y_k - w_j)$ . On s'assurera que la consommation est toujours positive.

```
C = zeros(nw, nw, ny);
for kk = 1:ny
C(:, :, kk) = repmat(w', nw, 1) + y(kk)*ones(nw) - repmat(w, 1, nw);
end
C(C<0)=0;
U = C.^(1-gam)./(1-gam);
```

**1d** Le processus itératif consiste à calculer en premier la valeur de continuation, c.-à-d. l'espérance de la fonction valeur de la prochaine période. Calculez cette valeur, puis à retransformer la matrice obtenue dans un tableau  $n \times n \times 2$ . On initialisera la matrice  $\mathbf{V}$  et  $\mathbf{g}$  à zéro.

```
V = zeros(nw, ny) ;
g = zeros(nw, ny) ;

Vc = V*pii' ;
Vc = reshape(Vc, nw, 1, ny) ;
Vc = repmat(Vc, [1 nw 1]) ;
```

**1e** Appliquer la contraction et récupérez la fonction valeur et la règle de décision, c.-à-d. l'indice correspondant à la grille qui indique le niveau optimal de richesse en fin de période.

```
[TV, Tg] = max(U + bet*Vc);
TV = reshape(TV, nw, ny);
Tg = reshape(Tg, nw, ny);
```

**1f** Le critère d'arrêt est soit si le changement dans la fonction valeur entre deux itérations est plus petite qu'une valeur arbitrairement petite, ou si le nombre maximum d'itération est atteint, ou enfin si la règle de décision est invariante entre deux itérations. Calculez  $dV$ , la différence maximale en valeur absolue entre deux itérations sur la fonction valeur. Écrire également sous forme de valeur logique (booléen) les 3 conditions d'arrêt.

**% initialisation pour l'exemple**

```
ii = 1;

dV = max(max(abs(TV-V))) ;
eV = (dV <= 1.0e-06);
ei = (ii >= 100);
eg = isequal(g,Tg);
```

**1g** Mettre à jour la fonction valeur et la règle de décision.

```
V = TV ;
g = Tg ;
```

**1h** Répétez l'opération à l'aide d'une boucle `while`.

**% La boucle s'écrit alors**

```
V = zeros(nw,ny) ;
g = zeros(nw,ny) ;
eV = 0;
ei = 0;
eg = 0;
ii = 0;

while ~eV && ~ei && ~eg
% calcul de la valeur de continuation
Vc = V*pii' ;
Vc = reshape(Vc,nw,1,ny) ;
Vc = repmat(Vc,[1 nw 1]) ;

% contraction
[TV,Tg] = max(U + bet*Vc);
TV = reshape(TV,nw,ny);
Tg = reshape(Tg,nw,ny);

% calcul des critères d'arrêt
dV = max(max(abs(TV-V))) ;
eV = (dV <= 1.0e-06);
ei = (ii >= 100);
eg = isequal(g,Tg);

% mise à jour
V = TV ;
g = Tg ;
ii = ii + 1;
end
```

**1i** Écrire un ensemble de commande permettant de connaître les raisons de l'arrêt de la boucle.

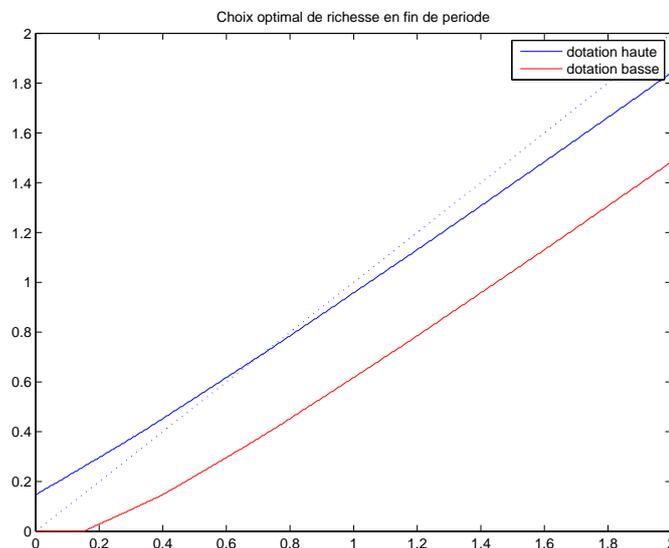
```
if eg
disp('Règle de décision invariante. ');
elseif ei
disp('Maximum d'itérations atteint. ');
else
disp('Convergence obtenue. ');
end
```

**1j** L'algorithme retourne deux matrices, celle des valeurs de la fonction valeur évaluées aux noeuds  $(w_i, y_j)$ , et les indices dans la grille  $\{w_k\}_{k=1}^n$  qui définit le choix optimal du prochain niveau de richesse. Calculez la règle de décision en valeur  $\bar{W}$ , ainsi que le choix de consommation optimal  $c$ .

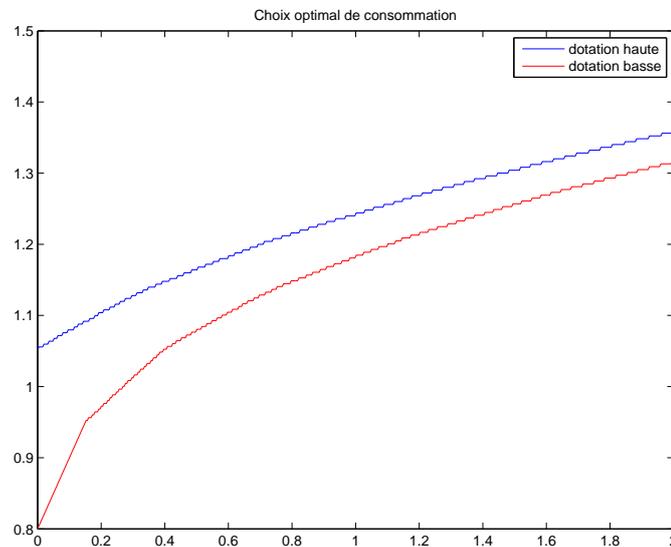
```
 $\bar{W} = w(g)$  ;
 $c = \text{repmat}(w,1,2) + \text{repmat}(y,nw,1) - \bar{W}$  ;
```

**1k** Tracer les règles de décision comme une fonction de la taille de la richesse.

```
figure;
plot(w,W(:,1),'b',w,W(:,2),'r',w,w,':');
legend('dotation haute','dotation basse');
title('Choix optimal de richesse en fin de période');
print('./pdf/App_5_fig_W.pdf','-dpdf')
```



```
figure;
plot(w,c(:,1),'b',w,c(:,2),'r');
legend('dotation haute','dotation basse');
title('Choix optimal de consommation');
print('./pdf/App_5_fig_c.pdf','-dpdf')
```



11 On pose  $\gamma = 0.5$ . Comparer le niveau et la pente de la règle de décision pour les deux valeurs de  $\gamma$ . Comment interpréter ces résultats ?

```
% On recommence
gam = .5 ;
for kk = 1:ny
C(:, :, kk) = repmat(w', nw, 1) + y(kk)*ones(nw) - repmat(w, 1, nw);
end
C(C<0)=0;
U = C.^(1-gam)./(1-gam);

V = zeros(nw,ny) ;
g = zeros(nw,ny) ;
eV = 0;
ei = 0;
eg = 0;
ii = 0;

while ~eV && ~ei && ~eg
% calcul de la valeur de continuation
Vc = V*pii' ;
Vc = reshape(Vc, nw, 1, ny) ;
Vc = repmat(Vc, [1 nw 1]) ;

% contraction
[TV, Tg] = max(U + bet*Vc);
TV = reshape(TV, nw, ny);
Tg = reshape(Tg, nw, ny);

% calcul des critères d'arrêt
dV = max(max(abs(TV-V))) ;
eV = (dV <= 1.0e-06);
ei = (ii >= 100);
```

```

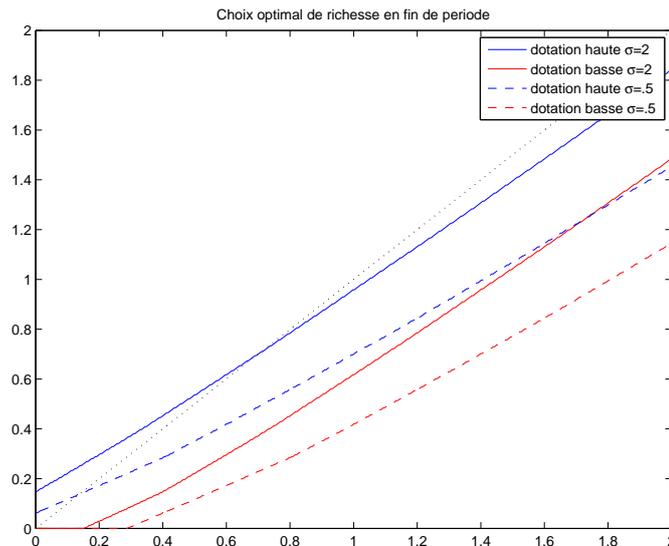
eg = isequal(g,Tg);

% mise à jour
V = TV ;
g = Tg ;
ii = ii + 1;
end

W2 = w(g) ;
c2 = repmat(w,1,2) + repmat(y,nw,1) - W2 ;

figure;
plot(w,W(:,1),'b',w,W(:,2),'r',w,W2(:,1),'--b',w,W2(:,2),'--r',w,w,':k');
legend('dotation haute \sigma=2','dotation basse \sigma=2',...
'dotation haute \sigma=.5','dotation basse \sigma=.5');
title('Choix optimal de richesse en fin de période');
print('./pdf/App_5_fig_W2.pdf','-dpdf')

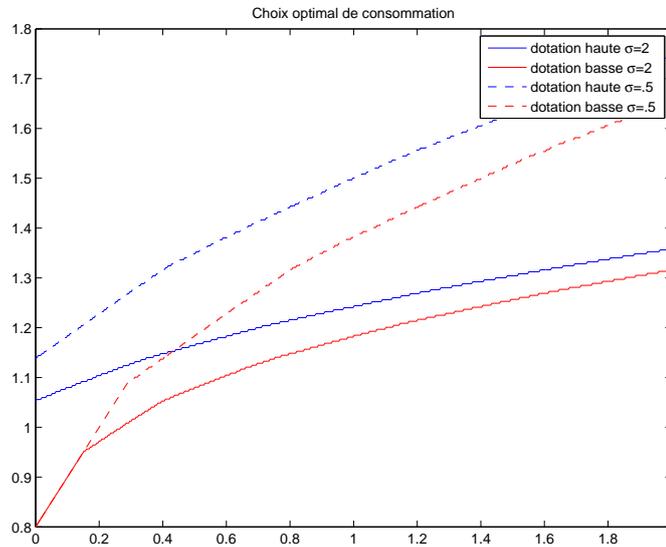
```



```

figure;
plot(w,c(:,1),'b',w,c(:,2),'r',w,c2(:,1),'--b',w,c2(:,2),'--r');
legend('dotation haute \sigma=2','dotation basse \sigma=2',...
'dotation haute \sigma=.5','dotation basse \sigma=.5');
title('Choix optimal de consommation');
print('./pdf/App_5_fig_c2.pdf','-dpdf')

```



### Le modèle de croissance néoclassique

Le but est d'écrire un programme qui permette de résoudre le modèle de croissance néoclassique en utilisant la méthode de l'itération sur la fonction valeur et une interpolation linéaire. Un planificateur social bienveillant a pour objectif

$$\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

sous la contrainte  $c_t + k_{t+1} = A(k_t)^\alpha + (1 - \delta)k_t$  et  $c_t \geq 0, k_{t+1} \geq 0, k_0 > 0$  donné. On pose  $u(c) = \log(c)$  et  $\beta = 0.95, \alpha = 0.36$  et  $A = 1$ . Obtenir des résultats numériques à la fois pour le cas de la dépréciation totale ( $\delta = 1$ ) et pour la cas d'une dépréciation moindre ( $\delta < 1$ , par exemple  $\delta = 0.1$ ). Pour  $\delta = 1$ , comparer les résultats numériques avec la forme analytique de la solution pour la fonction valeur et la règle de décision (Cela nécessite de résoudre le modèle en question au préalable).

#### Solution

On a besoin dans un premier temps de résoudre analytiquement le cas  $\delta = 1$ . La première étape consiste à donner une forme récursive au problème :

$$V(k) = \max_{0 \leq k' \leq Ak^\alpha} \{\log(Ak^\alpha - k') + \beta V(k')\}.$$

On suppose une solution de la forme  $V(k) = B + C \log(k)$ . Le maximum du membre de droite de l'équation de Bellman est atteint lorsque

$$-\frac{1}{Ak^\alpha - k'} + \frac{\beta C}{k'} = 0 \Rightarrow k' = \frac{\beta CAk^\alpha}{1 + \beta C}.$$

En substituant  $k'$  dans l'équation de Bellman et en rassemblant les termes, on a

$$(\alpha(1 + \beta C) - C) \log(k) + (1 + \beta C) \log\left(\frac{A}{1 + \beta C}\right) + \beta B + \beta C \log(\beta C) - B = 0, \forall k > 0.$$

En conséquence, pour  $k \rightarrow 0$  et  $k = 1$ , cette équation mène au système

$$\begin{aligned} \alpha(1 + \beta C) - C &= 0 \\ (1 + \beta C) \log\left(\frac{A}{1 + \beta C}\right) + \beta B + \beta C \log(\beta C) - B &= 0 \end{aligned}$$

ce qui donne  $C = \alpha/(1 - \alpha\beta)$  et  $B = (1 - \beta)^{-1}(1 - \alpha\beta)^{-1}[\log(A(1 - \alpha\beta)) + \alpha\beta \log(\alpha/(1 - \alpha\beta))]$ . Alors, la règle de décision pour le stock de capital de la période suivante est  $k'(k) = \alpha\beta Ak^\alpha$  et la consommation  $c(k) = (1 - \alpha\beta)Ak^\alpha$ . Notons qu'il existe un niveau de capital pour lequel  $k' = k = \bar{k} = (\alpha\beta A)^{1/(1-\alpha)}$ . Dans le cas  $\delta < 1$ , le niveau de l'état stationnaire du capital est donné par  $\bar{k} = (\alpha A / (1/\beta - 1 + \delta))^{1/(1-\alpha)}$ .

```
% Nettoyage
clear all; clc; close all;
```

On commence par définir les paramètres et la grille pour le stock de capital. Il peut être souhaitable d'inclure le niveau de capital de l'état stationnaire.

```
bet = 0.95;
alp = 0.36;
A = 1;
delt = 1;
kbar = 2*(alp*A/(1/bet-1+delt))^(1/(1-alp)) ;
N = 100;
kmin = 0.02;
K = linspace(kmin,kbar,N)';
u = @(c) log(c);
```

```
% initialisation
V = zeros(N,1);
TV = zeros(N,1);
g = zeros(N,1);
Tg = zeros(N,1);
n = 0;
eV = 0;
eg = 0;
en = 0;
```

Pour chaque élément  $k_i$  dans la grille, nous devrions être capables de calculer le maximum de la fonction  $f(k) = u(Ak_i^\alpha + (1 - \delta)k_i - k) + \beta \hat{v}_i(k)$ , où  $\hat{v}_i$  est une approximation linéaire de la fonction valeur basée sur la  $i$ ème itération.

```
while ~eV && ~en && ~eg
fV = @(k) interp1(K,V,k,'linear');
for ii = 1:N
TfV = @(k) u(max(A*K(ii).^alp+(1-delt)*K(ii)-k,1e-15)) + bet*fV(k);

[kmax,Vmax] = goldsekrule(TfV,K([1 end]));

Tg(ii) = kmax ;
TV(ii) = Vmax ;
end
```

On définit également les critères d'arrêt comme dans la partie précédente.

```
dV = max(max(abs(TV-V)))/(1+max(max(abs(TV)))) ;
eV = (dV <= 1.0e-09) ;
```

```

dg = max(max(abs(Tg-g)))/(1+max(max(abs(Tg)))) ;
eg = (dg <= 1.0e-09);
en = (n >= 100);

```

Finalement, on met à jour la fonction valeur et la règle de décision optimale

```

V = TV ;
g = Tg ;
n = n + 1 ;

```

```
end
```

On cherche les raisons de l'arrêt de l'algorithme.

```

if eg
disp('Invariance de la règle de décision.');
```

```
elseif en
disp('Nombre d'itération maximum atteint.');
```

```
else
disp('Convergence de la fonction valeur.');
```

```
end
```

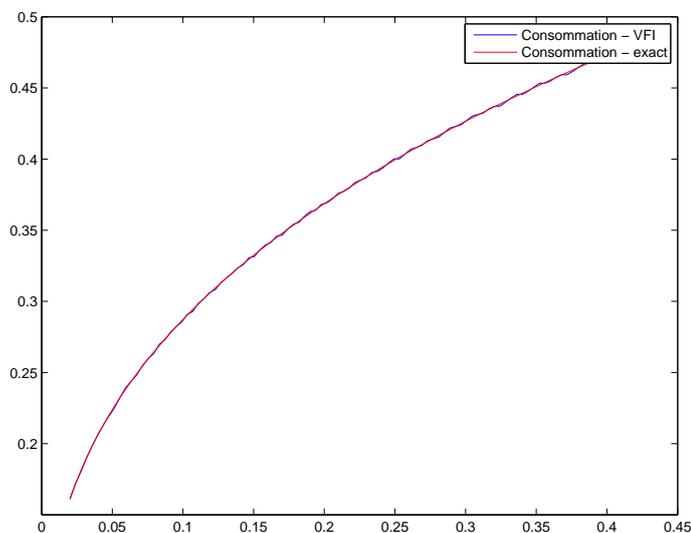
Et on peut tracer le graphe des règles de décision pour le capital de la période suivante et la consommation

```

C = A*K.^alp+(1-delt)*K-g;
C_exact = (1-alp*bet)*A*K.^alp;
g_exact = alp*bet*A*K.^alp;

figure;
plot(K,C, 'b',K,C_exact, 'r');
legend('Consommation - VFI', 'Consommation - exact')
print('./pdf/App_5_fig_cons.pdf', '-dpdf')

```



```
figure;  
plot(K,g,'b',K,g_exact,'r',K,K,':');  
legend('Capital - VFI','Capital - exact')  
print('./pdf/App_5_fig_cap.pdf','-dpdf')
```

