

```

1 clc
2
3 %%%%%%%%%%%
4 % Paramètres
5 %%%%%%%%%%%
6 beta = 0.96;
7 alpha = 0.33;
8 delta = 0.1;
9
10 %%%%%%%%%%%
11 % Grille des variables d'état (k) et de choix (k')
12 %%%%%%%%%%%
13 points = 500;      % nombre PAIR de points - en vue de créer la grille de capital.
14 ratio_inf = 0.8;   % position de la borne inférieure, relativement au capital stationnaire.
15 ratio_sup = 1.2;   % position de la borne supérieure, relativement au capital stationnaire.
16
17 kstat = ((1/beta - 1 + delta) / alpha) ^ (1 / (alpha-1)); % capital stationnaire.
18 borne_k_inf = ratio_inf * kstat; borne_k_sup = ratio_sup * kstat;
19
20 moitie_inf = linspace(borne_k_inf, kstat, points/2); moitie_sup = linspace(kstat, borne_k_sup, points/2);
21 k = union(moitie_inf, moitie_sup)'; % (points-1) x 1
22 capital = repmat(k', [points-1, 1]); % (points-1) x (points-1)
23
24 %%%%%%%%%%%
25 % Avant la boucle de convergence
26 %%%%%%%%%%%
27 consommation = capital.^alpha - capital' + (1-delta) * capital; % (points-1) x (points-1)
28 utilite = log(consommation); % (points-1) x (points-1)
29 consommation_negative = find(consommation <= 0);
30 utilite(consommation_negative) = -1e10;
31 ustat = log(kstat^alpha - delta*kstat); % utilite stationnaire.
32
33 %%%%%%%%%%%
34 % Boucle de convergence
35 %%%%%%%%%%%
36 tolerance = 1e-6;
37 comparaison = 1;
38 valeur = repmat(ustat/(1-beta), [points-1, 1]);
39
40 while max(abs(comparaison)) > tolerance

```

```
41     Bellman = utilite + beta * valeur;
42     Tvaleur = max(Bellman);
43     comparaison = Tvaleur - valeur;    comparaison = comparaison ./ valeur;
44     valeur = Tvaleur;
45 end;
46
47 [valeur, regle_index] = max(Bellman);
```