Macroeconomics Sequence, Block I

Introduction to Numerical Methods

Nicola Pavoni

September 27, 2016

Introduction

- Recall the BE in the deterministic case:

$$V(x) = \max_{x' \in \Gamma(x)} F(x, x') + \beta V(x').$$

- Recall the value function iteration procedure:
  1. Initial guess V_0
  2. Apply the $\mathbf{T}$ operator to it and get a new function $V_1 = \mathbf{T} V_0$
  3. Apply again $\mathbf{T}$ to V_1 and get $V_2 = \mathbf{T} V_1$ and so on.

- The maximization in $\mathbf{T}$ must be using *numerical algorithms*.
- The operator $\mathbf{T}$ must work in finite spaces $\Rightarrow$ *(value) function approximations*.
- We can only perform finitely many iterations $\Rightarrow$ *tolerance.*

Discretization

1. Discretize X: $\hat{X} = \left[x^1,\ x^2,\ x^3, ..., x^N\right]$.
2. Pick V_0: $\hat{V}_0 = \left[v_0^1,\ v_0^2, ..., v_0^N\right]$.
3. For each $x^i \in \hat{X}$, look for the $x^j (= x') \in \hat{X}$ that solves

$$\max_{x^j \in \hat{X},\ x^j \in \hat{\Gamma}(x^i)} F\left(x^i, x^j\right) + \beta v_0^j$$

4. Denote v_1^i the value associated to $F\left(x^i, x^{j^*}\right) + \beta v_0^{j^*}$
   $\Rightarrow$ We get our new function $\hat{V}_1 = \left[v_1^1,\ v_1^2, ..., v_1^N\right]$.
5. Do so until below tolerance ε:

$$d\left(\hat{V}_n, \hat{V}_{n+1}\right) = \sum_i \varpi^i \left|v_n^i - v_{n+1}^i\right| < \varepsilon,$$

The Deterministic Growth Model

- Parametrize: Cobb-Douglas production function and log utility:

$$f(k) = k^{\alpha} + (1 - \delta)\, k \;\; \text{and} \;\; u(c) = \ln(c),$$

  where δ is the depreciation rate, α is the capital share.

- If $\delta = 1$:

$$V(k) = \max_{0 \leq k' \leq k^{\alpha}} \ln\left(k^{\alpha} - k'\right) + \beta V\left(k'\right),$$

- We know that $g(k) = \alpha \beta k^{\alpha}$.
- We now compute it using `Matlab`

The Computer Code I

- Initialize the problem

```
clear all;
close all;
```

- Define parameters

```
beta=.9; % β = .9
alpha=.35; % α = .35
NumPoints =100;
```

- Discretize the state space around the steady state capital stock

```
k_bar = (alpha*beta)^(1/(1-alpha)); % k* = (αβ)^(1/(1-α))
k_lo = k_bar*0.5;
k_hi = k_bar*2;
step = (k_hi-k_lo)/NumPoints;
K = k_lo:step:k_hi;
n=length(K);
```

The Computer Code II

- Build an $n \times n$ matrix whose columns are output at each value of k (in Matlab vectors are raw-vectors by default)
  ```
  Y= K.^alpha;
  YY = ones(n,1)*Y;
  ```
- Then another $n \times n$ matrix whose columns are capital
  ```
  KK = ones(n,1)*K;
  ```
- Consumption at each level of k' is then given by
  ```
  C = YY-KK';
  ```
- Calculate the utility arising from each level of consumption
  ```
  U=log(C);
  ```
- Take an initial guess at the value function
  `V = zeros(n,1);` (this is a column-vector)

The Computer Code III

- Apply the operator

$$W = U + \beta \begin{bmatrix} 0 \\ .. \\ 0 \end{bmatrix} [1 \; .....1]$$

```
VV=V*ones(1,n);
W=U+beta*VV;
```

- Given a k, we want to find the k' that solves

$$\begin{bmatrix} TV\left(k^1\right) \\ TV\left(k^2\right) \\ .. \\ TV\left(k^N\right) \end{bmatrix} = \max \; W$$

TV=max W;

The Computer Code III

- Main iteration loop for the value function.

```
flag=1;
while (flag > 10^(-2))
    VV=V*ones(1,n);
    W=U+beta*VV;
    V1=max(W)';
    flag = max(abs(V1-V));
    V=V1;
end
```

- When the value function has converged, find the policy function i.e. the k' that gives the maximum value of the operator for each k. To do this first find the vector of indices where W takes its maximum value

```
[val,ind]=max(W);
```

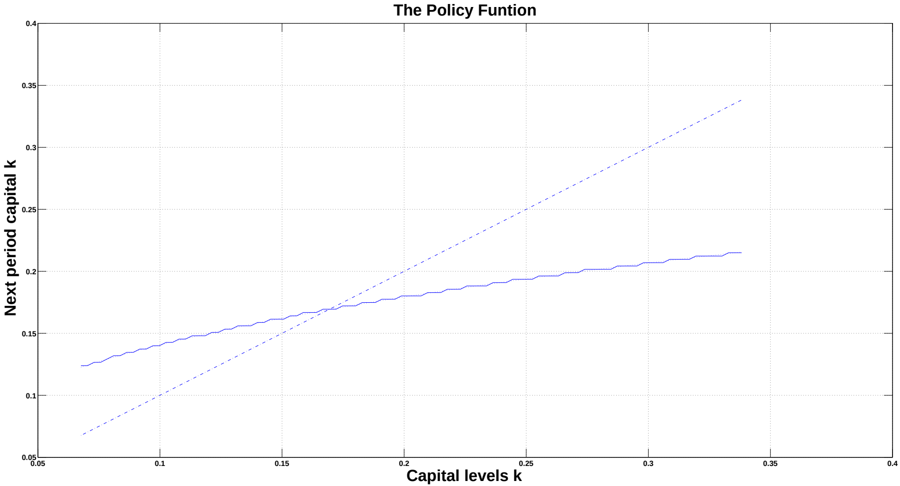
- Then use the indices to pick out the corresponding values of k.

```
k_star = K(ind);
```

Generating Figures

```
figure(1)
hold on
plot(K, k_star, 'b-')
```
This is the approximate policy
```
plot(K,g, 'r-')
```
This is the analytic policy
```
plot(K,K,'-.')
```
The 45 degree line
```
title('The Policy Funtion')
xlabel( ' Capital levels k ' )
ylabel( ' Next period capital k' )
hold off
```

Figure: Analytic and Approximate Policies



Have a look at the file `discretize.m`