

# Value Function Iteration

July 1, 2018

```
In [1]: import sympy as sp
import numpy as np

In [2]: u=lambda x: sp.log(x) #Funcion de Utilidad

In [3]: f=lambda x: sp.sqrt(x) #Funcion de Produccion

In [4]: V0=np.zeros(200)      #Inicializa V0 para las iteraciones
V1=np.zeros(200)             #Inicializa V1 para las iteraciones
g=np.zeros(200)              #Inicializa g, funcion politica

In [5]: K=np.linspace(0.11,0.22,200) #Malla para K. Valores cerca de K*

In [6]: for i in range(0,200):
    V0[i]=u(f(K[i]))          #Primera iteracion de V

In [7]: for i in range(0,200):
    m=np.zeros(200)
    for j in range (0,200):
        m[j]=u(f(K[i])-K[j])+0.9*V0[j]
    M=max(m)
    V1[i]=M                    #Segunda Iteracion de V

In [8]: er=np.linalg.norm(V1-V0,np.inf) #Distancia entre las dos iteraciones V0 y V1

In [9]: er

Out[9]: 1.3962356730823471

In [10]: z=1 #Solo un contador para las iteraciones, optativo

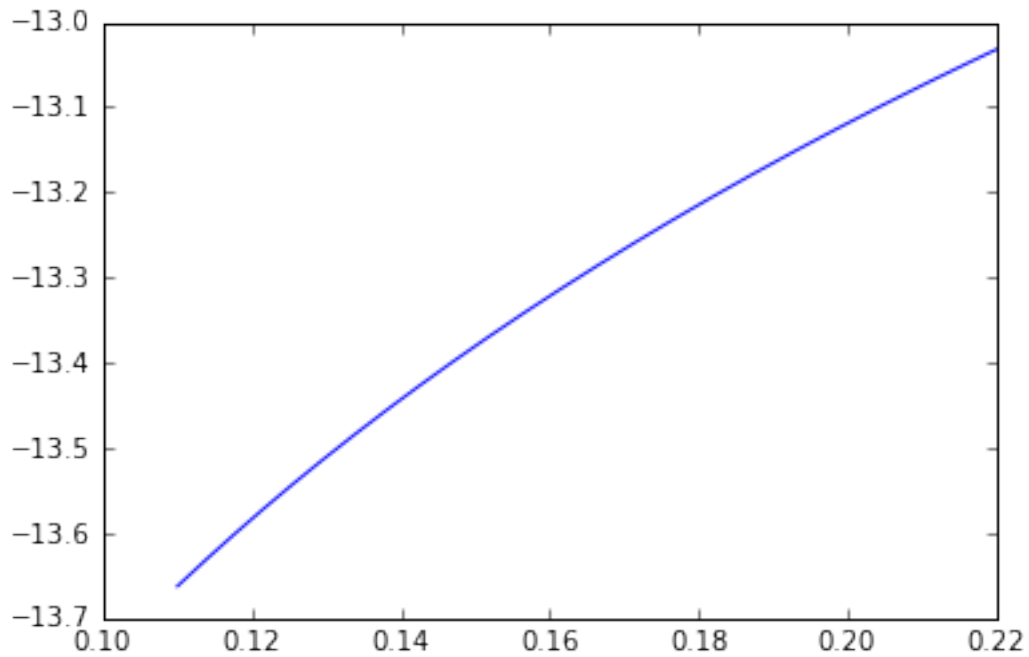
In [11]: while er>0.1:      #Iteracion completa
    z=z+1
    V0=V1                    #Cambiamos V0 por el siguiente
    V1=np.zeros(200)         #Borramos V1 para hacer el nuevo
    for i in range(0,200):   #Iteramos lo anterior
        m=np.zeros(200)
        for j in range (0,200):
            m[j]=u(f(K[i])-K[j])+0.9*V0[j]
        M=max(m)
        t=np.argmax(m)
        V1[i]=M
        g[i]=K[t]            #Almacenamos la funcion politica
    er=np.linalg.norm(V1-V0,np.inf) #calculamos la distancia de nuevo

In [12]: %matplotlib inline
```

```
In [13]: from matplotlib.pyplot import plot,show
```

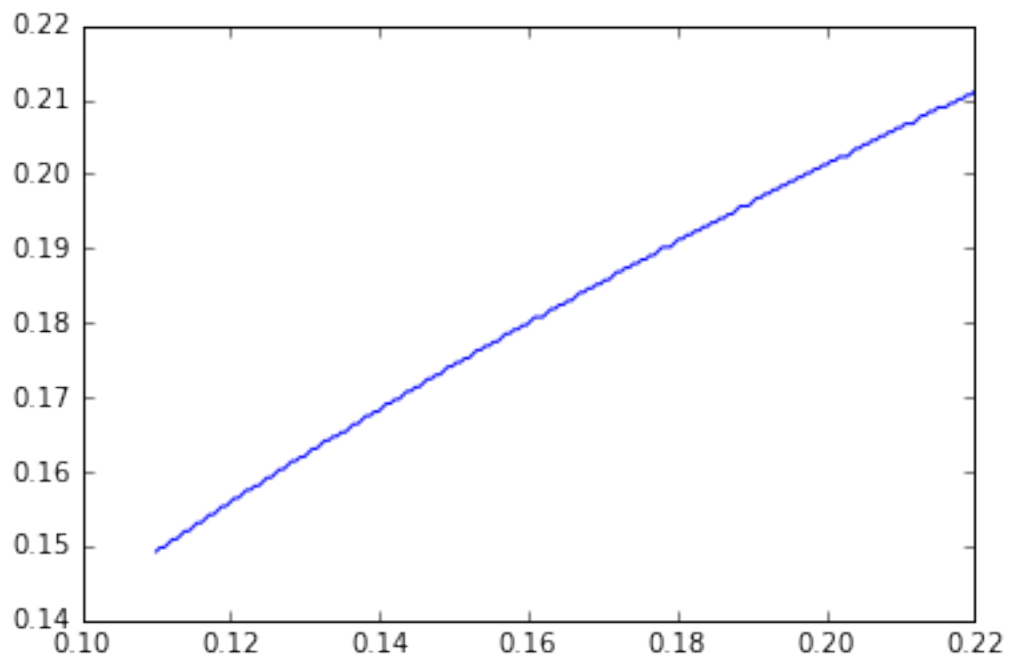
```
In [14]: plot(K,V1)
```

```
Out[14]: [<matplotlib.lines.Line2D at 0x7fc52f903390>]
```



```
In [15]: plot(K,g)
```

```
Out[15]: [<matplotlib.lines.Line2D at 0x7fc52f87c890>]
```



```

In [16]: import scipy.interpolate

In [17]: g_in=scipy.interpolate.interp1d(K,g)  #Armos la funcion politica interpolando

In [18]: g_in(0.1426)  #Probamos en un cierto K a ver si anda

Out[18]: array(0.16969849246231156)

In [19]: T=range(100)  #El T para la trayectoria

In [20]: tray=np.zeros(100)  #Inicializa vector tray

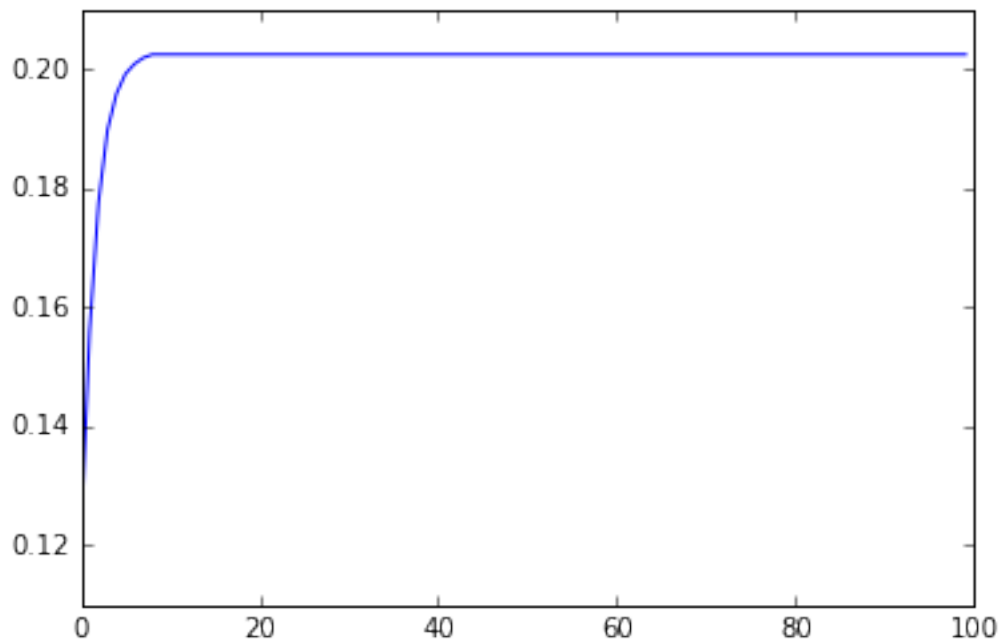
In [21]: tray[0]=0.12  #Elegimos un K0, e.g. K0=0.12

In [22]: for l in range(0,99):
            tray[l+1]=g_in(tray[l])  #La trayectoria es K_t+1=g(K_t)

In [23]: plot(T,tray)

Out[23]: [<matplotlib.lines.Line2D at 0x7fc526d5ee90>]

```



```

In [ ]:

```