PS 4

Weimin Zhou

Due: 17, Oct, 2018

Change from Python to Matlab. In Python, we have many sources to do directly the VFI with deterministic and stochastic models such as Quantecon.org:
https://lectures.quantecon.org/py/optgrowth.html

Q1. Value Function Iteration

1. Without Labor Choice part.
[PS4Q1_1.m], and a short version to test: [SampleCode_for_PS4Q1_1.m]
This part trains us to have a deep understanding of how to do VFI and speed up our iteration by imposing known conditions.

Discussion: Below results are basically the same for each value function plotting, but just consume less times of iterations when imposing conditions. Different conditions of improvements exhibit no big differences but reduce a significant time if compared with Brute force iteration. In the later questions, I use monotonicity and Howard's Improvements for saving times.

. (a) brute force iterations

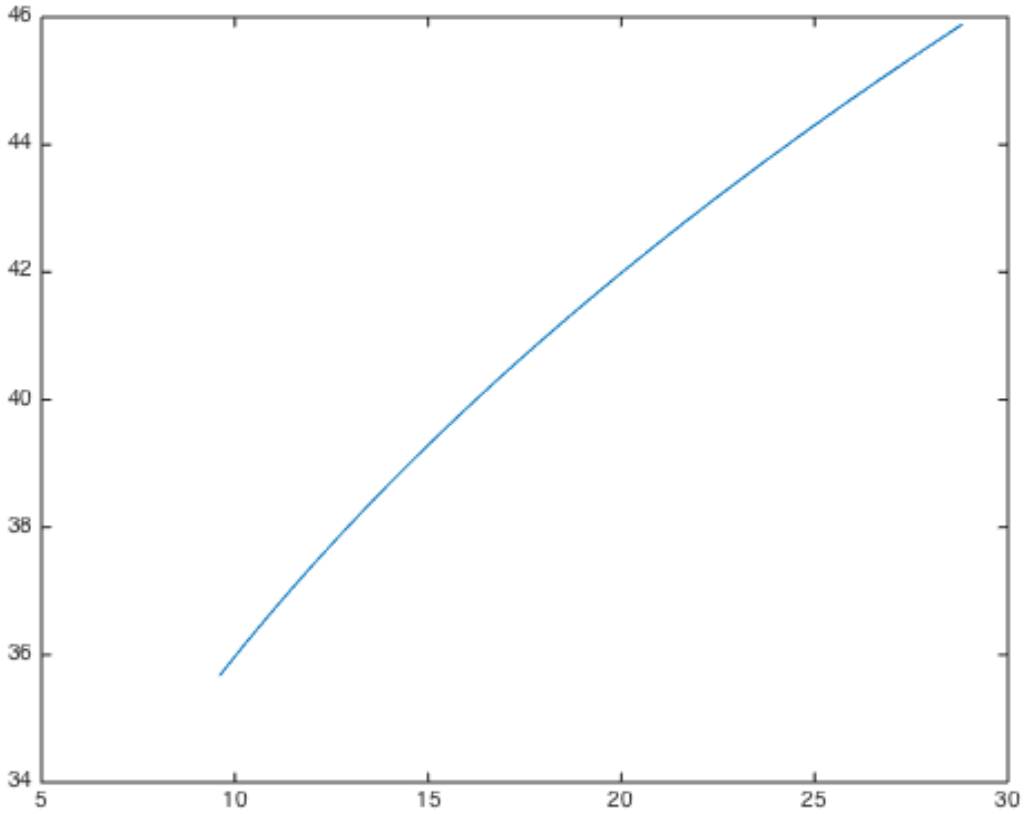
Number of iterations:
100
Number of Grid points:
2000
PARAMETER VALUES

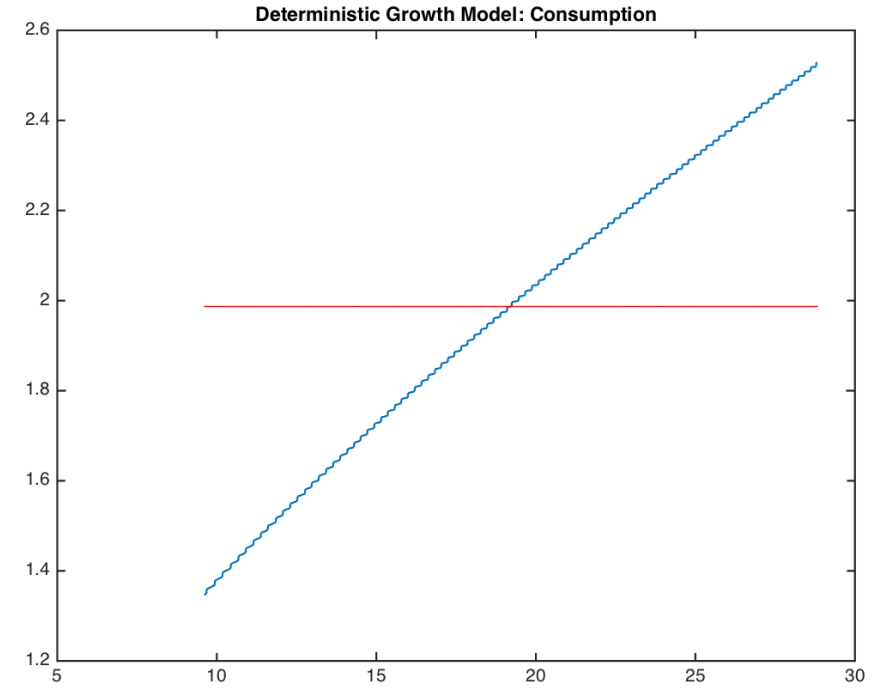
alfa	beta	delta
0.3210	0.9880	0.0310

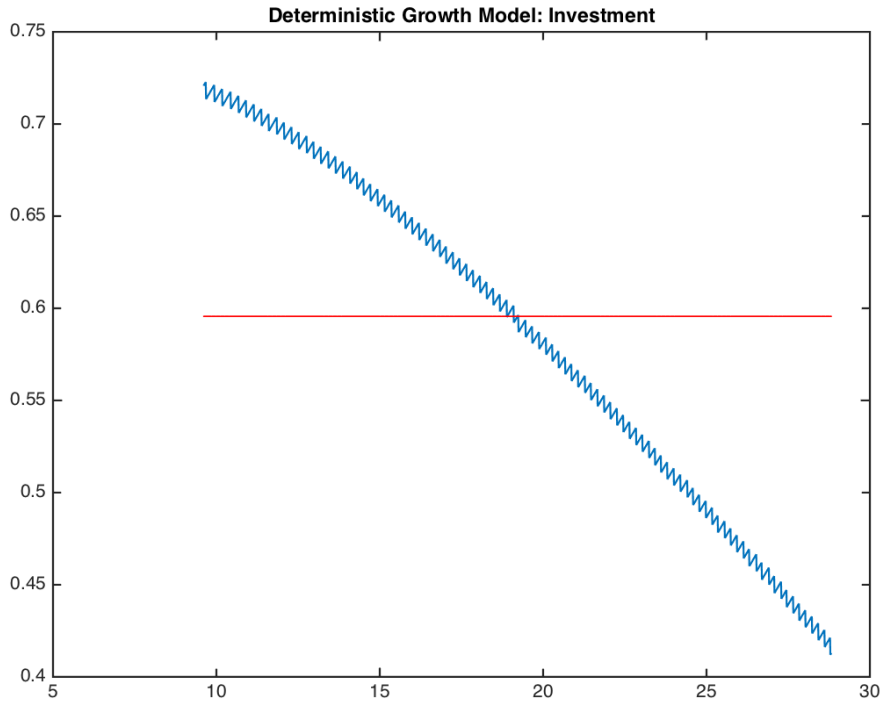
Value of Productivity

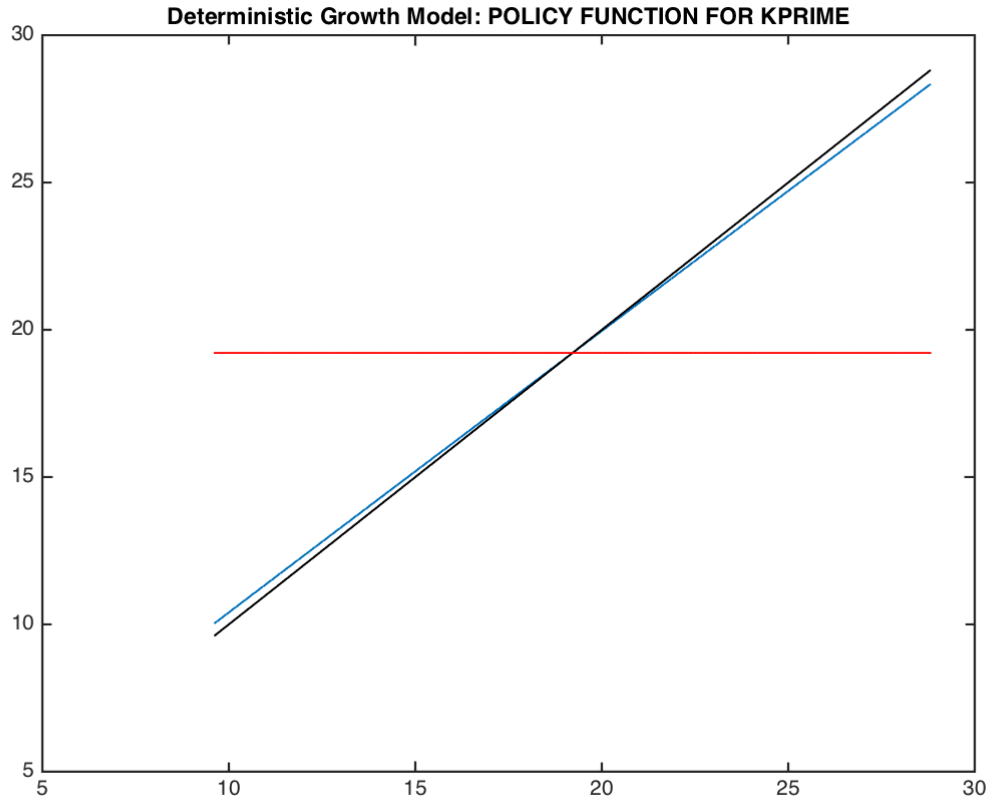
1

Elapsed time is 12.433784 seconds.




Deterministic Growth Model: Consumption
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
5
10
15
20
25
30


Deterministic Growth Model: Investment
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
5
10
15
20
25
30



the benchmark RBC model has a unique equilibrium, there's no sunspot, no bubble equilibrium.

. (b) Iterations of the value function taking into account monotonicity of the optimal decision rule.

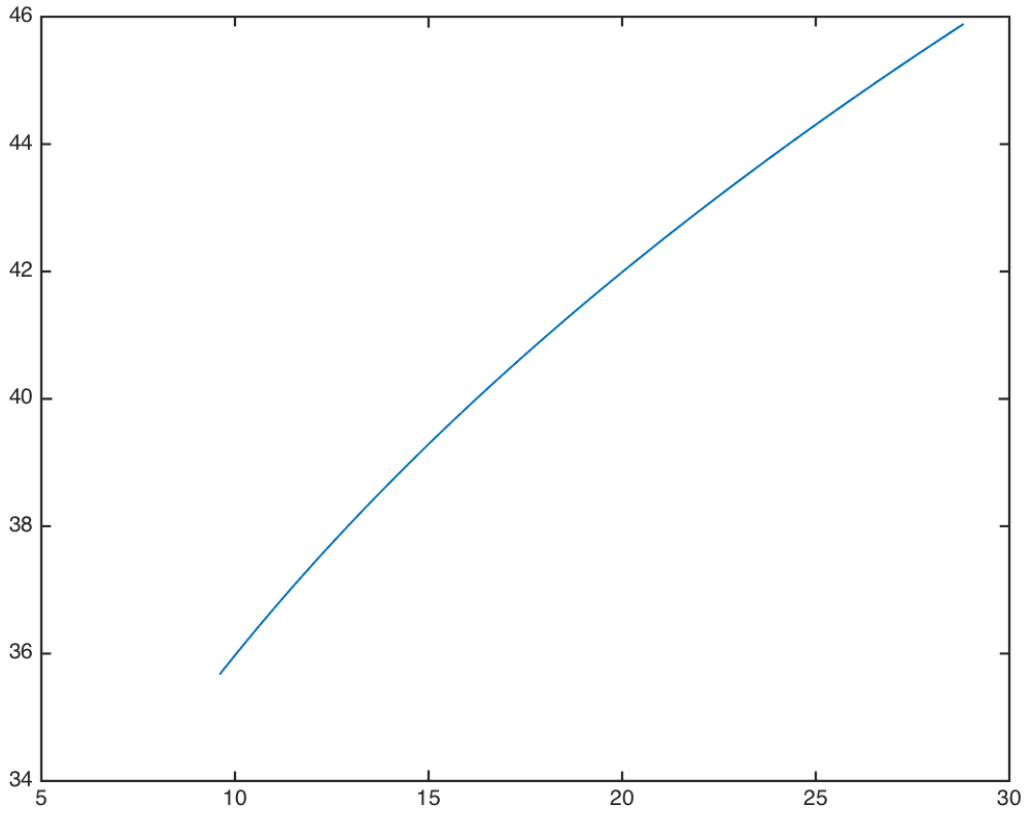
Value Function Iterations for the deterministic growth model are completed.

Elapsed time is 0.612944 seconds.

.

Number of iterations:

100


46
44
42
40
38
36
34
5
10
15
20
25
30

. (c) Iterations of the value function taking into account concavity of the value function.

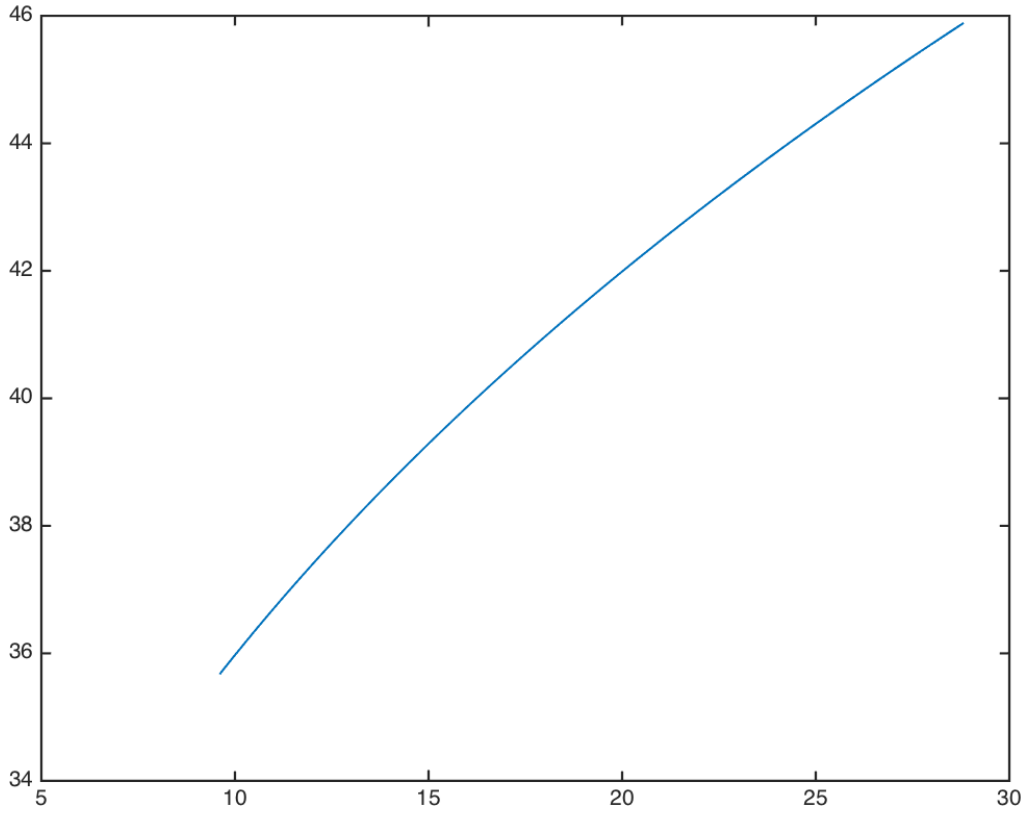
Value Function Iterations for the deterministic growth model are completed.

Elapsed time is 0.615340 seconds.

.

Number of iterations:

100



. (d) Iterations of the value function taking into account local search on the decision rule.

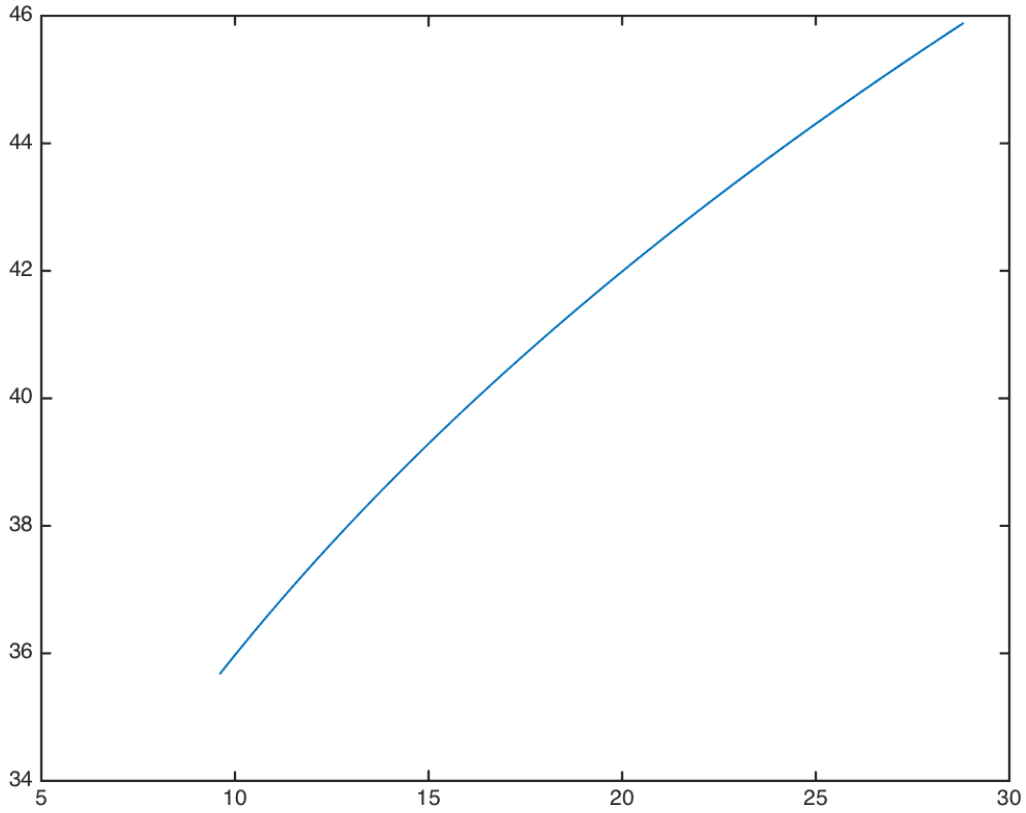
Value Function Iterations for the deterministic growth model are completed.

Elapsed time is 0.612639 seconds.

.

Number of iterations:

100



. (e) Iterations of the value function taking into account both concavity of the value function and monotonicity of the decision rule

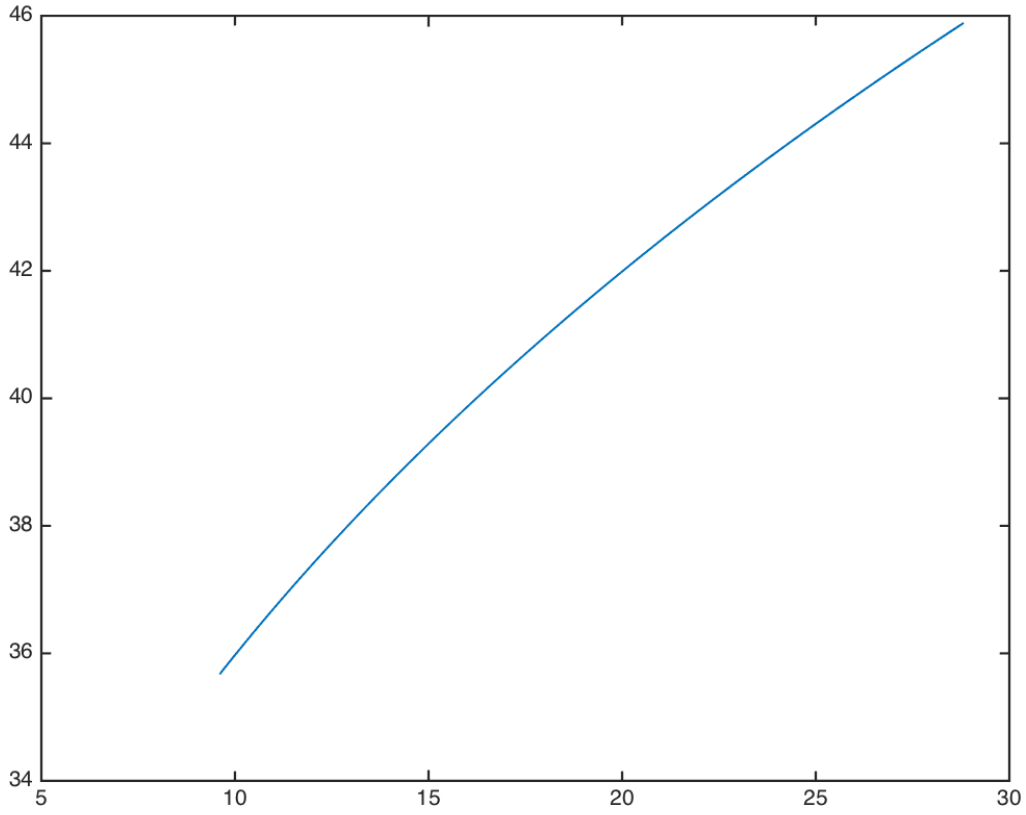
Value Function Iterations for the deterministic growth model are completed.

Elapsed time is 0.609081 seconds.

.

Number of iterations:

100



. (f) Use Howard's policy iterations waiting until converged to solve the problem.

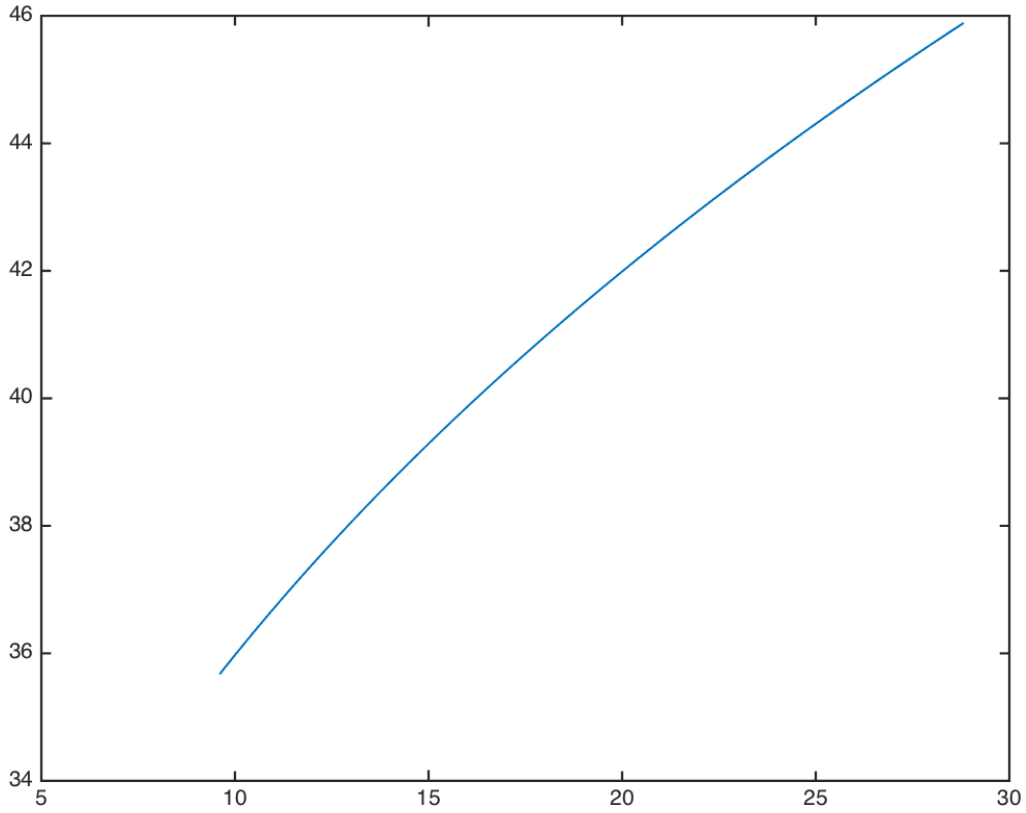
Value Function Iterations for the deterministic growth model are completed.

Elapsed time is 2.326120 seconds.

.

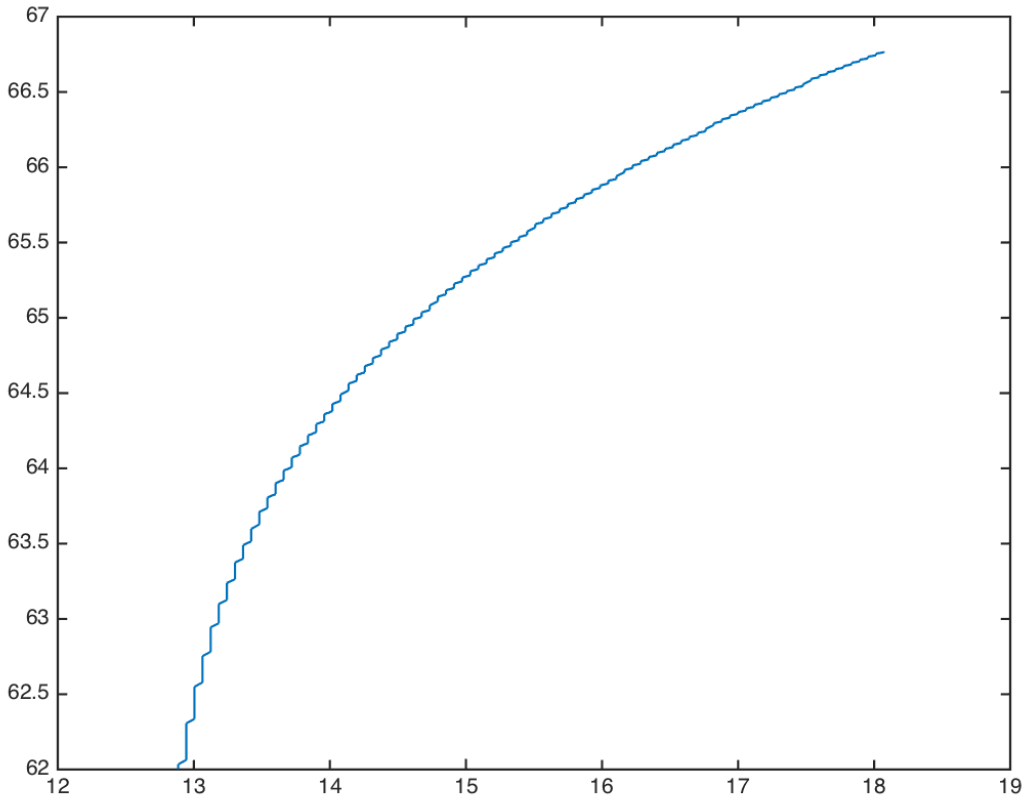
Number of iterations:

100



. (g) Use policy iterations with 5, 10, 20 and 50 steps in between policy reassessments.

For simplicity, I only attached the value function plotting with 5 steps policy reassessments.



2. Redo item 1 adding a labor choice that is continuous.

When labor choice is added, we can consider to use parametric system to solve, which I failed, because one part took over 2 hours, hence, I use the Algorithm which constructs two state variables in Matlab, the labor choice and the capital tomorrow.

In the folder Q1_2:

[PS4Q1_2.m]: the main part of codes;

[focss,m]: needed function for computing steady states;

[focdynamic1.m; focdynamic.m]: failed parametric system functions, which took many times.

I use Tricks for obtaining steady states: set initial guess by a reasonable conjecture: capital around 5; labor choice around .3; and consumption less than 1.

Steady States of the economic system

Kstar = 12.2977

Hstar = 0.2890

Cstar = 0.8035

Istar = 0.3812

% ====================================

% Elapsed time is 216.576990 seconds.

% Number of iterations: 769

% Number of Grid points: 500

% PARAMETER VALUES

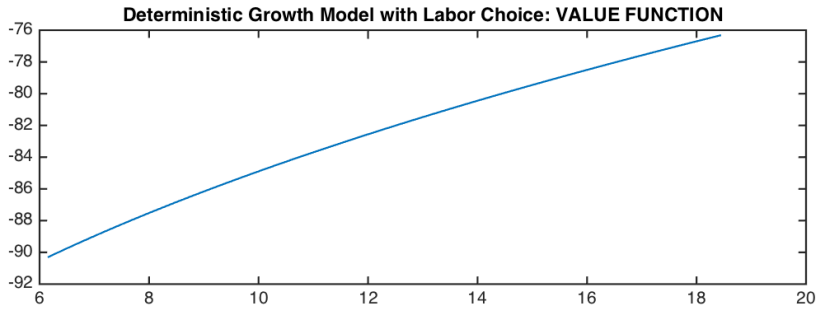
%	alfa	beta	delta	kappa	nu
%	0.3210	0.9880	0.0310	5.2400	2.0000

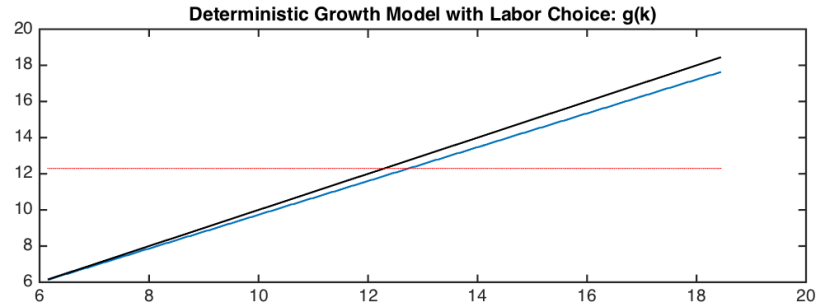
% ====================================

% Above I take into account monotonicity of the optimal decision

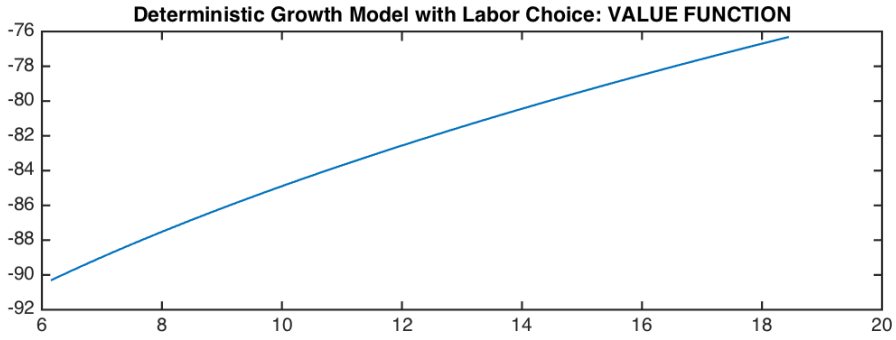
% notice that, it's two state variables now in computing

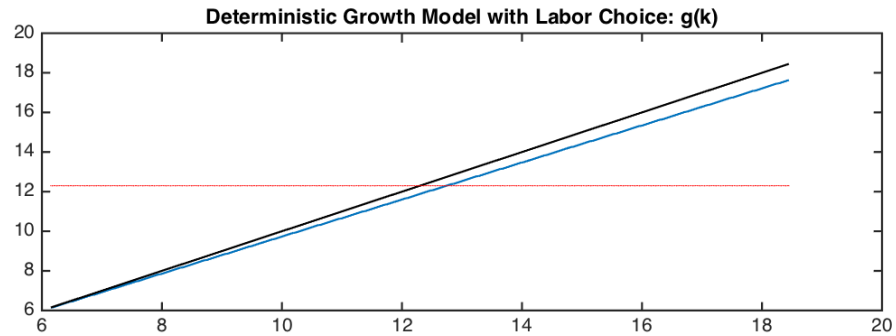
% hence, we need two variables to take into account our restrictions.





After adding labor choice, capital level is slower, and the Value function is slower, since labor is elastic supplying and agents could gain utility by leisure time (not working). For simplicity, I impose monotonicity, and concavity of the value function, it took 200 seconds for computing. Below, I add Howard Improvements, and it's only 15 seconds.



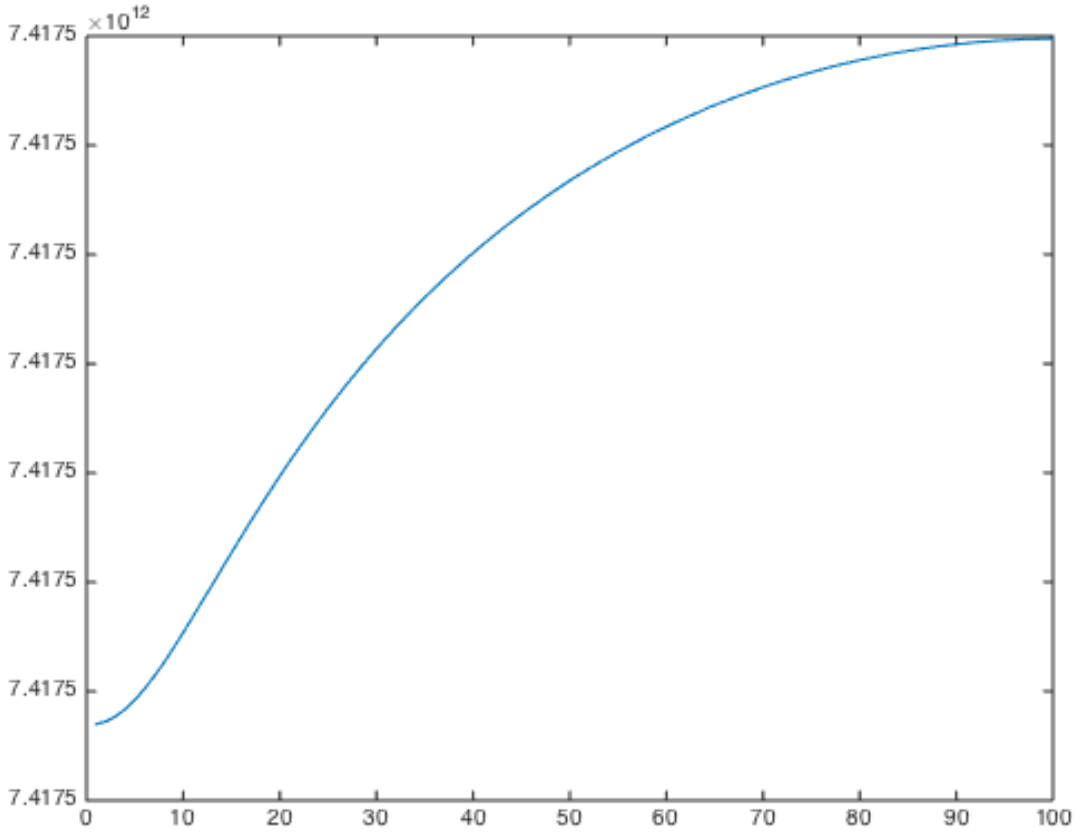


. 3. Redo item 1 using a Chebyshev regression algorithm to approximate the value function. Compare your results.

[Chebyshev.m]: needed function for Chebyshev fitting;
[value.m]: approximating value function by Chebyshev function;
[tv.m]: compute overall utility and recursive Value function
[PS4Q1_3.m]: main part of codes.

Again, for simplicity, I only report a few variants of results since each is consuming many times for iterations. I use fminunc for every capital in loop to gain a next-period capital, and then approximate again using Chebyshev approximation in tv.m; the results is not pretty, since Chebyshev polynomials do not put any assumption on the shape of the value function (i.e. concave and strictly increasing).

Value Function Iterations for the deterministic growth model are completed.
Elapsed time is 140.491061 seconds.


×10^12
7.4175
7.4175
7.4175
7.4175
7.4175
7.4175
7.4175
0
10
20
30
40
50
60
70
80
90
100