Fatih Guvenen
University of Minnesota

# Homework #3

Due back: Beginning of class, Tuesday, November 24, 2009.

Questions indicated by a "star" are required for everybody who attends the class. You can use either MatLab or Fortran to do the homework.

**For each question, please discuss your answer.** (Please do not merely provide some numbers and a code).

1. This question will teach you about a different parameterization of CRRA utility functions that yields an approximately linear value function in a wide-range of economic problems. This approximate linearity will save you a substantial amount of time when you solve DP problems and need to interpolate the value function on the RHS. First, some preliminaries. This part asks you to do some algebra to get acquainted with Epstein-Zin preferences.

   (a) Consider a basic portfolio choice problem where preferences are given by the Epstein-Zin specification (also called "recursive preferences"). For the specifics of this problem, see Epstein (1988, JME). Derive the relative risk aversion coefficient with respect to wealth gambles, that is $-W((J''(W))/(J'(W)))$ (where $W$ is financial wealth and $J(.)$ is the value function) and show that it is equal to $(1/(1-\alpha))$ where $\alpha$ is the exponent of the future utility.

   (b) Now, we will modify the problem a bit. Suppose that there is a single risk-free asset for investment subject to an ad hoc borrowing constraint (in the language of Aiyagari (1994)). Furthermore, suppose that the individual receives a stochastic labor income that follows a first order Markov process with a bounded support (again as in Aiyagari). Solve the dynamic programming problem of the individual numerically using value function iteration. Repeat the experiment for risk aversion coefficient of 2, and 10, and EIS values of 2, and 0.1. [Discussion: When you do value function iteration you will need to interpolate the value function next period (ie, on the right hand side of the Bellman equation) at the off-grid points. You are free to use any kind of interpolation technique as you wish as long as you minimize the computational time necessary. If you feel like it, experiment with different choices (linear, spline, and Chebyshev) and report which one works best (in terms of delivering an accurate solution in as little time as possible). You should iterate until the discrepancy in the decision rule between successive iterations (under sup-norm) is less than $10^{-8}$. Finally, you should use a maximization routine that makes use of derivative information (such as Dbrent.f90 or some Newton-based algorithm for efficiency. In both cases, make sure to supply the correct *analytical* derivative of the right hand side of the Bellman equation, which is a bit more involved than with CRRA specification.

(c) Plot the decision rules and value function as a function of individual's financial wealth keeping his income shock constant at its average value.

(d) Now set the parameters of the Epstein-Zin specification such that it reduces to standard CRRA utility. Solve the problem in part (b) again taking a risk aversion value of 2. Repeat with RRA= 10.

(e) Now solve the portfolio choice problem in (d) again, BUT using the standard CRRA specification: $U(c) = \frac{c^{1-\alpha}}{1-\alpha}$ You should get the exact same decision rule as in part (d), but the value function should have much more curvature. Plot the decision rules and value function to verify this claim. Compare the time necessary to solve each version of the problem (you can use the built-in function "cpu_time" in Fortran 90 and the tic/toc commands in MatLab).

## Econ 8312. Computational Methods
## Homework #3
## Iskander Karibzhanov

## (a) Epstein-Zin preferences

Consider a basic portfolio choice problem where preferences are given by Epstein-Zin specification. Letting $w$ denote wealth at the beginning of period $t$, the budget constraint facing consumer in period $t$ is

$$w_{t+1} = R_{t+1}(w_t - c_t)$$

where interest rate $R_t$ is iid random variable. The value function solves the following functional equation

$$J(w_t) = \max_{0 \le c_t \le w_t} \left[ c_t^\rho + \beta \left( E_t J(w_{t+1}) \right)^{\frac{\rho}{\alpha}} \right]^{\frac{\alpha}{\rho}}, \quad t \ge 0$$

such that the above budget constraint holds. Conjecture that value function and policy function take the form

$$J(w) = (\psi w)^\alpha$$
$$c_t = \mu w_t$$

where $\psi$ and $\mu$ are yet undetermined constants. Then we can rewrite budget constraint as

$$w_{t+1} = (1-\mu) R_{t+1} w_t$$

Taking first order condition with respect to consumption c, we obtain

$$\rho c_t^{\rho-1} = \beta \frac{\rho}{\alpha} \left( E_t J(w_{t+1}) \right)^{\frac{\rho}{\alpha}-1} E_t \left[ J'(w_{t+1}) R_{t+1} \right]$$
$$\rho(\mu w_t)^{\rho-1} = \beta \frac{\rho}{\alpha} (E_t [(\psi w_{t+1})^\alpha])^{\frac{\rho}{\alpha}-1} E_t [\psi^\alpha \alpha w_{t+1}^{\alpha-1} R_{t+1}]$$
$$(\mu w_t)^{\rho-1} = \beta (E_t [(\psi(1-\mu) R_{t+1} w_t)^\alpha])^{\frac{\rho}{\alpha}-1} E_t \left[ \psi^\alpha \left( (1-\mu) R_{t+1} w_t \right)^{\alpha-1} R_{t+1} \right]$$
$$(\mu w_t)^{\rho-1} = \beta (E_t [R_{t+1}^\alpha])^{\frac{\rho}{\alpha}-1} E_t [R_{t+1}^\alpha] \, \psi^\rho ((1-\mu) w_t)^{\rho-1}$$
$$(1-\mu)\mu^{\rho-1} = \beta (E_t [R_{t+1}^\alpha])^{\frac{\rho}{\alpha}} \psi^\rho (1-\mu)^\rho$$

Substituting this into expression for value function we get

$$J(w_t)^{\frac{\rho}{\alpha}} = (\psi w_t)^\rho = (\mu w_t)^\rho + \beta (E_t (\psi w_{t+1})^\alpha)^{\frac{\rho}{\alpha}} = (\mu w_t)^\rho + \beta (E_t [(\psi(1-\mu) R_{t+1} w_t)^\alpha])^{\frac{\rho}{\alpha}}$$
$$= (\mu w_t)^\rho + \beta (E_t [R_{t+1}^\alpha])^{\frac{\rho}{\alpha}} (\psi(1-\mu) w_t)^\rho = (\mu w_t)^\rho + (1-\mu)\mu^{\rho-1} w_t^\rho = \mu^{\rho-1} w_t^\rho$$

Therefore, $\psi$ and $\mu$ are given by

$$\psi = \mu^{\frac{\rho-1}{\rho}}$$
$$\mu = 1 - \left[ \beta (E_t [R_{t+1}^\alpha])^{\frac{\rho}{\alpha}} \right]^{\frac{1}{1-\rho}}$$

Once we verified our guess, we can calculate the Arrow-Pratt measure of the curvature of the value function

$$-w \frac{J''(w)}{J'(w)} = -w \frac{\psi^\alpha \alpha (\alpha-1) w^{\alpha-2}}{\psi^\alpha \alpha w^{\alpha-1}} = 1 - \alpha$$

We thus have the following result: for Epstein-Zin preferences, the value function is a power function of wealth, and its curvature is governed by attitude toward risk $\alpha$.

## (b) Hugget model

The economy consists of many infinitely lived households differing only with regard to their income shock $y_t$ and their asset holdings $a_t$. Households maximize their intertemporal utility of the Epstein-Zin form. The value function is

$$V(a_t, y_t) = \max_{\bar{a} \le a_{t+1} \le Ra_t + y_t} \left[ (Ra_t + y_t - a_{t+1})^{\rho} + \beta (E_t\, V(a_{t+1}, y_{t+1})^{\alpha})^{\frac{\rho}{\alpha}} \right]^{\frac{1}{\rho}}$$

where $y_t$ is an income shock, logarithm of which I assume is a first-order Markov chain with only three states

$$\log y_t \in \{-0.1, \quad 0, \quad 0.1\}$$

and transition probability matrix $\Pi$ given by

$$\Pi = .75 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + .25 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

I assume that $\beta = 0.9$, that the gross interest rate is $R = 1.01$ and that the borrowing limit is equal to $\bar{a} = -2$.

To solve this dynamic programming problem of the individual, I used smart value function iteration method exploiting monotonicity of policy function and quasi-concavity of value function and cubic spline interpolation between grid-points. In implementing this numerical algorithm I closely followed the steps described in Chapter 4 of Maussner DGE book.

### *Initial guess of value function*

Assuming $\alpha = \rho$, I first solved for $V_0(\bar{a}, y)$ using the initial guess for the policy function $a_0'(a, y) = \bar{a}$

$$V_0(\bar{a}, y) = [(I - \beta\Pi)^{-1}(R\bar{a} + y - \bar{a})^{\rho}]^{1/\rho}$$

I used this $V_0(\bar{a}, y)$ as initial guess for the general case when $\alpha \neq \rho$. I iteratively solved the following nonlinear system of equations until convergence:

$$V_0^{t+1}(\bar{a}, y) = \left[ (R\bar{a} + y - \bar{a})^{\rho} + \beta [\Pi V_0^t(\bar{a}, y')^{\alpha}]^{\frac{\rho}{\alpha}} \right]^{1/\rho}$$

Using $V_0(\bar{a}, y)$, I solved for the initial guess of value function $V_0(a, y)$ at all other asset grid points using

$$V_0(a, y) = \left[ (Ra + y - \bar{a})^{\rho} + \beta [\Pi V_0(\bar{a}, y')^{\alpha}]^{\frac{\rho}{\alpha}} \right]^{1/\rho}$$

### *Binary search*

To exploit the monotonicity of policy function I used the previous policy index as a starting point to find the maximum on the discrete grid. To exploit quasi-concavity of value function, I used binary search algorithm which dramatically reduces the number of function evaluations. The idea uses the fact that a strictly quasi-concave function defined over a grid of N points either takes its maximum at one of the two boundary points or in the interior of the grid. In the first case the function is decreasing (increasing) over the whole grid. In the second case the function is first increasing and then decreasing. As a consequence, we can pick the mid-point of the grid and check if the maximum is to the le*ft of this midpoint (if the function is decreasing at this point) or to the right of this midpoint (if it is increasing). The binary search based on this principle needs at most $\log_2 N$ steps to reduce the grid to a set of three points that contains the maximum.

### *Cubic spline interpolation*

In order to interpolate the value function next period at the off-grid points, I used cubic spline interpolation by setting first derivative of value function at end points to their analytical values found from envelope condition:

$$V_1(a, y) = RV(a, y)^{1-\rho} \big( Ra + y - a'(a, y) \big)^{\rho - 1}$$

where $V(a, y)$ and $a'(a, y)$ were calculated in previous iteration.
The resulting tridiagonal system of linear equations was solved using LAPACK function `pttrs` which exploits the symmetric positive definite properties of the system.

## *Optimization solver (dbrent)*

I used binary search to constrain the maximum to lie between two adjacent points on the grid. Then I used cubic spline interpolation to approximate the r.h.s of the Bellman equation by a continuous function. What is left is to use an optimization method that is able to deal with both boundary and interior solutions of a one-dimensional optimization problem. The derivative based Brent algorithm described in Numerical Recipes is an efficient alternative to derivative free methods like golden search. I only need to provide the correct analytical derivate of the r.h.s. of the Bellman equation.

$$rhs(a') = (Ra + y - a')^{\rho} + \beta[\Pi V(a', y')^{\alpha}]^{\frac{\rho}{\alpha}}$$

$$drhs(a') = -\rho(Ra + y - a')^{\rho-1} + \beta\rho[\Pi V(a', y')^{\alpha}]^{\frac{\rho}{\alpha}-1}[\Pi \mathrm{V}(\mathrm{a}', \mathrm{y}')^{\alpha-1}\mathrm{V}_1(\mathrm{a}', \mathrm{y}')]$$

where $\mathrm{V}_1(\mathrm{a}', \mathrm{y}')$ was computed using analytical first derivative of the cubic spline that interpolates the value function. As a minor notice, as you can see I didn't include the power $1/\rho$ in the formula for r.h.s of Bellman equation because it was unnecessary computational burden (raising to real power is rather expensive). Instead I negated the rhs if $\rho < 0$.

## *Termination criteria*

The value function iteration method was stopped if the largest absolute value of the difference in the value function between successive iterations were increasing from previous iteration.

## *Mean Euler equation residual*

The Euler equation residuals were computed for 100,001 equally spaced points in the same interval $[-2, 10]$ using the following formula

$$\varepsilon(a, y) = \frac{\left[\beta R[\Pi V(a', y')^{\alpha}]^{\frac{\rho}{\alpha}-1}[\Pi \mathrm{V}(\mathrm{a}', \mathrm{y}')^{\alpha-\rho}(Ra' + y' - a'')^{\rho-1}]\right]^{\frac{1}{\rho-1}}}{Ra + y - a'} - 1$$

where $a', a''$ and $V(a', y')$ were computed using cubic interpolation of the optimal saving rule $a'(a, y)$ and value function $V(a, y)$ at the residual grid points between the original asset grid points.

The stationary distribution of Markov chain and uniform asset distribution was used as weighting matrix in averaging the Euler equation residuals.

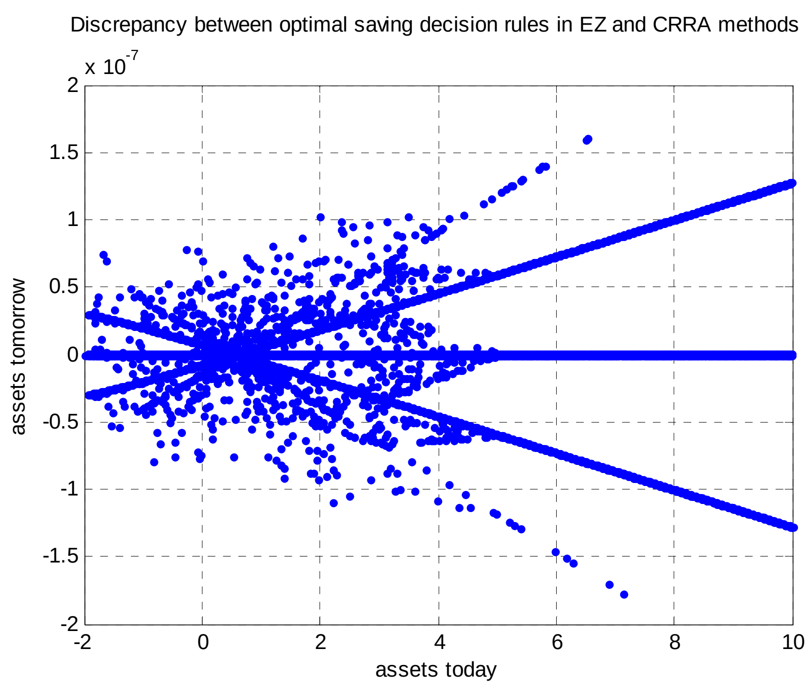
## *Results*

I compared the optimal saving decision rules from Epstein-Zin and CRRA specification with very precise numerical solution obtained using policy function iteration with endogenous grid method. For mean Euler equation residuals I used a fine grid of 100,001 points in the interval [-2,10].

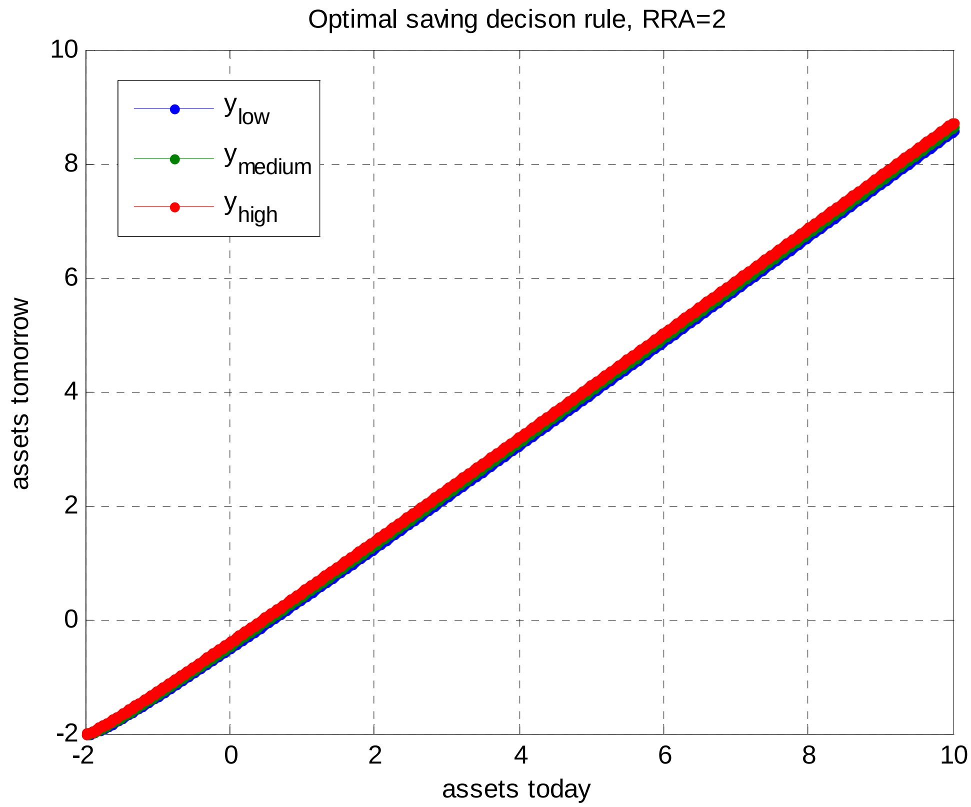
| | | | Epstein-Zin VFI with exogenous grid | | | | CRRA VFI with exogenous grid | | | | CRRA PFI with endogenous grid | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\alpha$ | $\rho$ | $N$-1 | #it | Mean EER | Time | Dendog | #it | Mean EER | Time | Dendog | #it | Mean EER | Time |
| -1 | -1 | 100000 | 26 | $8.64 \cdot 10^{-8}$ | 74.28 | $2.27 \cdot 10^{-8}$ | 27 | $8.61 \cdot 10^{-8}$ | 56.65 | $2.19 \cdot 10^{-8}$ | 25 | $1.54 \cdot 10^{-10}$ | 1.80 |
| -1 | -1 | 50000 | 26 | $3.90 \cdot 10^{-7}$ | 35.77 | $2.45 \cdot 10^{-8}$ | 27 | $3.90 \cdot 10^{-7}$ | 26.95 | $2.39 \cdot 10^{-8}$ | 25 | $4.39 \cdot 10^{-10}$ | 0.80 |
| -1 | -1 | 40000 | 26 | $7.05 \cdot 10^{-7}$ | 28.04 | $2.58 \cdot 10^{-8}$ | 27 | $7.05 \cdot 10^{-7}$ | 20.91 | $2.52 \cdot 10^{-8}$ | 25 | $1.58 \cdot 10^{-9}$ | 0.59 |
| -1 | -1 | 30000 | 26 | $1.25 \cdot 10^{-6}$ | 20.95 | $2.83 \cdot 10^{-8}$ | 27 | $1.25 \cdot 10^{-6}$ | 15.48 | $2.77 \cdot 10^{-8}$ | 25 | $2.40 \cdot 10^{-9}$ | 0.43 |
| -1 | -1 | 20000 | 26 | $2.63 \cdot 10^{-6}$ | 13.83 | $3.76 \cdot 10^{-8}$ | 27 | $2.63 \cdot 10^{-6}$ | 10.18 | $3.74 \cdot 10^{-8}$ | 25 | $8.65 \cdot 10^{-9}$ | 0.29 |
| -1 | -1 | 10000 | 26 | $1.03 \cdot 10^{-5}$ | 7.04 | $8.60 \cdot 10^{-8}$ | 27 | $1.03 \cdot 10^{-5}$ | 5.33 | $8.56 \cdot 10^{-8}$ | 25 | $3.09 \cdot 10^{-8}$ | 0.14 |
| -1 | -1 | 1000 | 30 | $4.73 \cdot 10^{-5}$ | 0.77 | $3.24 \cdot 10^{-6}$ | 28 | $4.73 \cdot 10^{-5}$ | 0.51 | $3.25 \cdot 10^{-6}$ | 26 | $3.94 \cdot 10^{-6}$ | 0.02 |
| -1 | -1 | 100 | 131 | $2.87 \cdot 10^{-3}$ | 0.32 | $7.59 \cdot 10^{-5}$ | 44 | $3.14 \cdot 10^{-4}$ | 0.08 | $7.91 \cdot 10^{-5}$ | 46 | $2.59 \cdot 10^{-4}$ | 0.00 |
| -1 | -1 | 20 | 234 | $1.32 \cdot 10^{-3}$ | 0.11 | $1.60 \cdot 10^{-3}$ | 698 | $1.49 \cdot 10^{-3}$ | 0.23 | $1.78 \cdot 10^{-3}$ | 79 | $1.59 \cdot 10^{-3}$ | 0.00 |
| -9 | -9 | 50000 | 261 | $1.58 \cdot 10^{-7}$ | 318.91 | $1.05 \cdot 10^{-8}$ | 279 | $1.56 \cdot 10^{-7}$ | 257.78 | $9.32 \cdot 10^{-9}$ | 173 | $5.66 \cdot 10^{-10}$ | 5.86 |
| -9 | -9 | 40000 | 290 | $1.37 \cdot 10^{-7}$ | 285.85 | $1.38 \cdot 10^{-8}$ | 291 | $1.35 \cdot 10^{-7}$ | 218.28 | $1.25 \cdot 10^{-8}$ | 167 | $9.85 \cdot 10^{-10}$ | 4.47 |
| -9 | -9 | 30000 | 257 | $4.40 \cdot 10^{-7}$ | 192.34 | $1.80 \cdot 10^{-8}$ | 279 | $4.38 \cdot 10^{-7}$ | 157.24 | $1.66 \cdot 10^{-8}$ | 197 | $1.55 \cdot 10^{-9}$ | 3.59 |
| -9 | -9 | 20000 | 295 | $1.11 \cdot 10^{-6}$ | 155.20 | $2.17 \cdot 10^{-8}$ | 278 | $1.10 \cdot 10^{-6}$ | 107.17 | $2.05 \cdot 10^{-8}$ | 195 | $2.72 \cdot 10^{-9}$ | 2.39 |
| -9 | -9 | 10000 | 293 | $3.08 \cdot 10^{-6}$ | 74.29 | $4.27 \cdot 10^{-8}$ | 284 | $3.08 \cdot 10^{-6}$ | 52.27 | $4.22 \cdot 10^{-8}$ | 182 | $8.58 \cdot 10^{-9}$ | 1.11 |
| -9 | -9 | 1000 | 286 | $1.77 \cdot 10^{-6}$ | 7.37 | $1.25 \cdot 10^{-6}$ | 282 | $1.76 \cdot 10^{-6}$ | 5.12 | $1.24 \cdot 10^{-6}$ | 185 | $1.85 \cdot 10^{-6}$ | 0.11 |
| -9 | -9 | 100 | 245 | $4.89 \cdot 10^{-5}$ | 0.53 | $1.42 \cdot 10^{-4}$ | 281 | $5.60 \cdot 10^{-5}$ | 0.50 | $1.65 \cdot 10^{-4}$ | 172 | $6.19 \cdot 10^{-5}$ | 0.01 |
| -9 | -9 | 20 | 164 | $3.21 \cdot 10^{-4}$ | 0.10 | $3.91 \cdot 10^{-3}$ | 299 | $3.85 \cdot 10^{-4}$ | 0.14 | $6.55 \cdot 10^{-3}$ | 201 | $4.07 \cdot 10^{-4}$ | 0.00 |

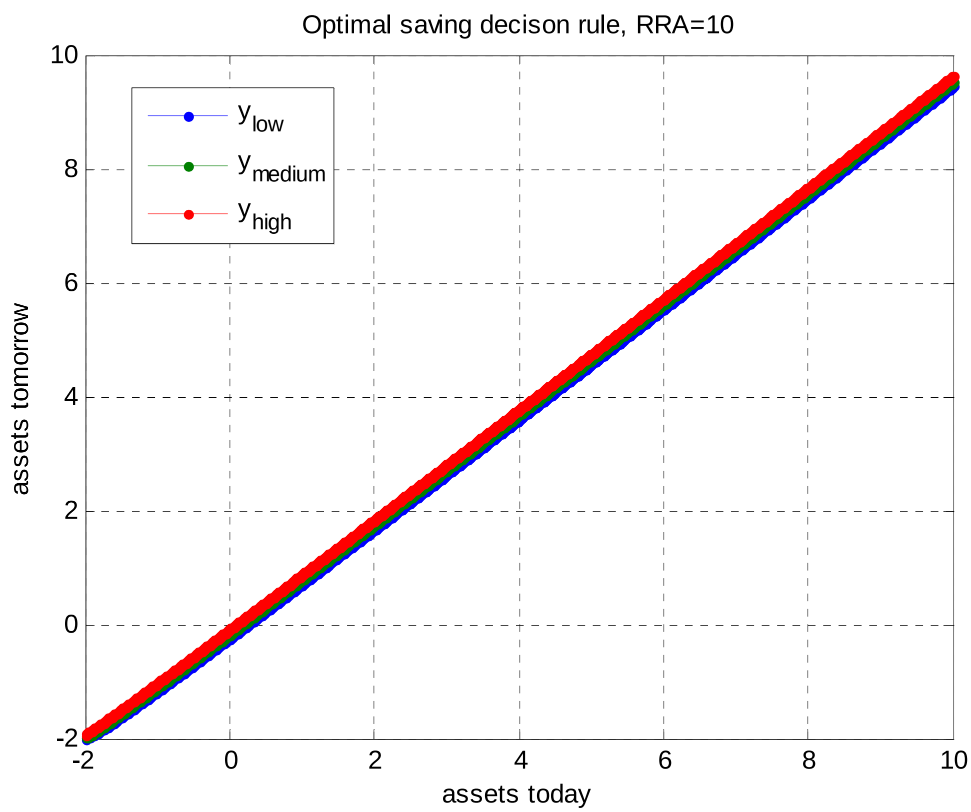
As we can see, the CRRA method with high risk aversion (RRA=10) requires 10 times longer run time to obtain the same accuracy as the same CRRA method with low RRA=2. The EZ method is more time consuming than CRRA method and also 10 times more consuming if we increase risk aversion from 2 to 10.

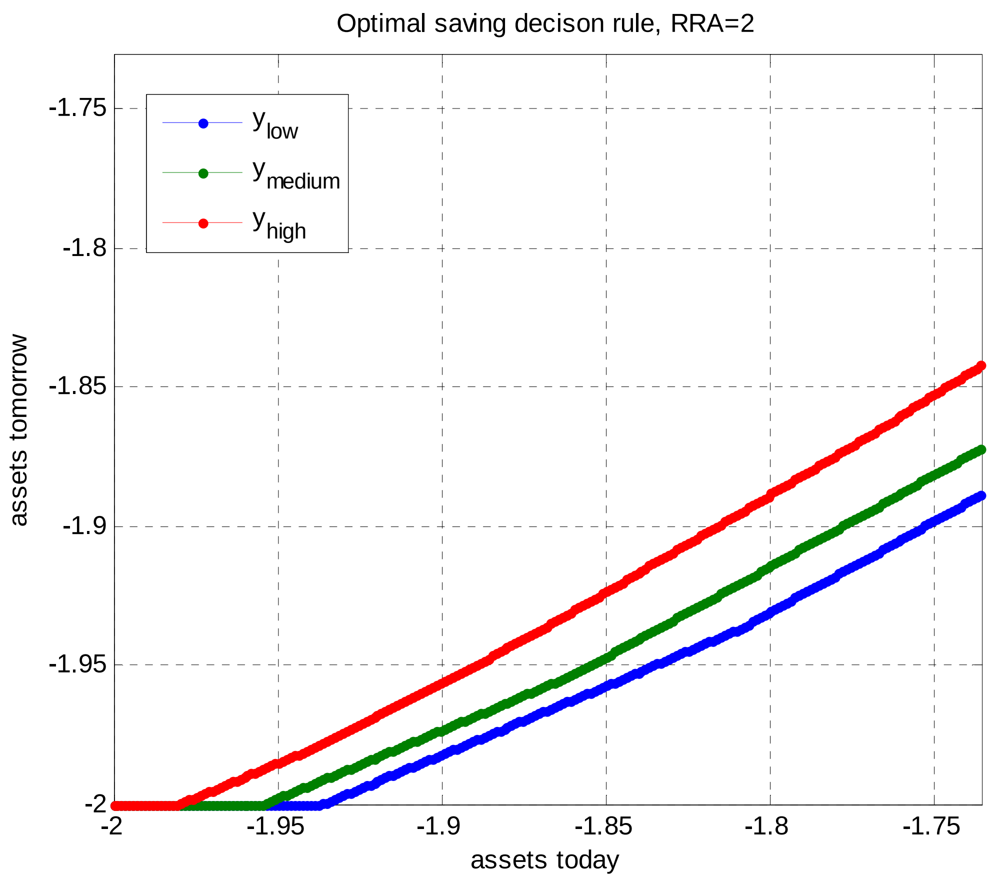
The optimal decision rules are almost exactly the same as can be seen from the following figure.

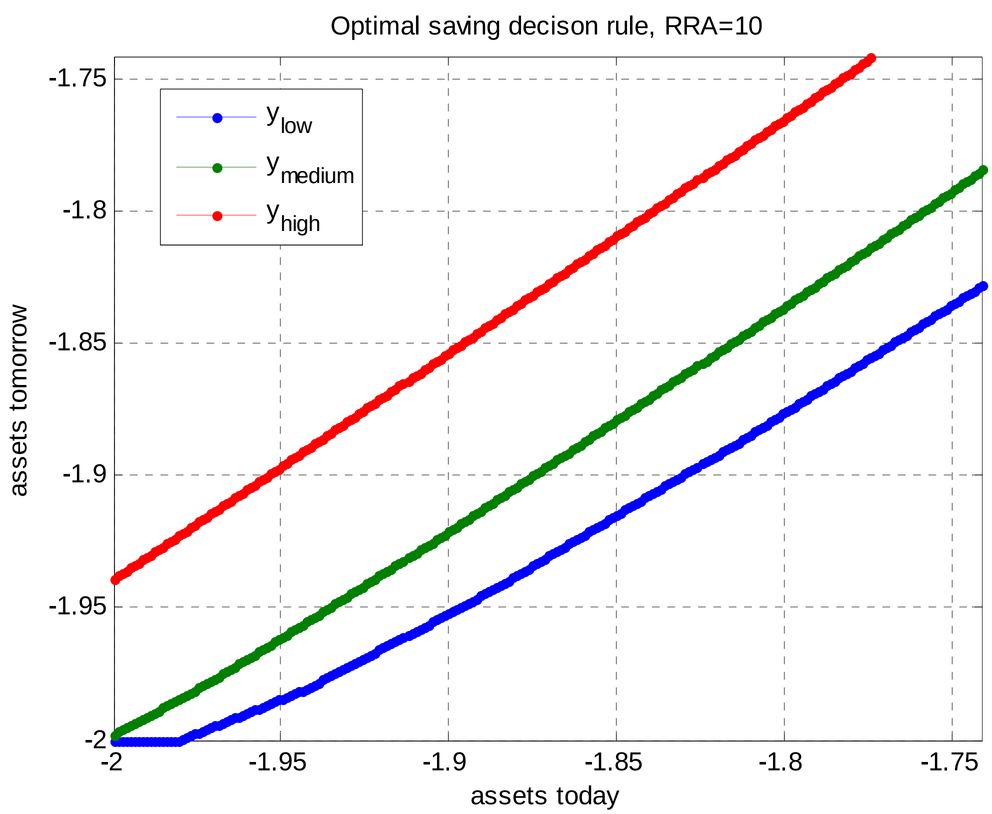


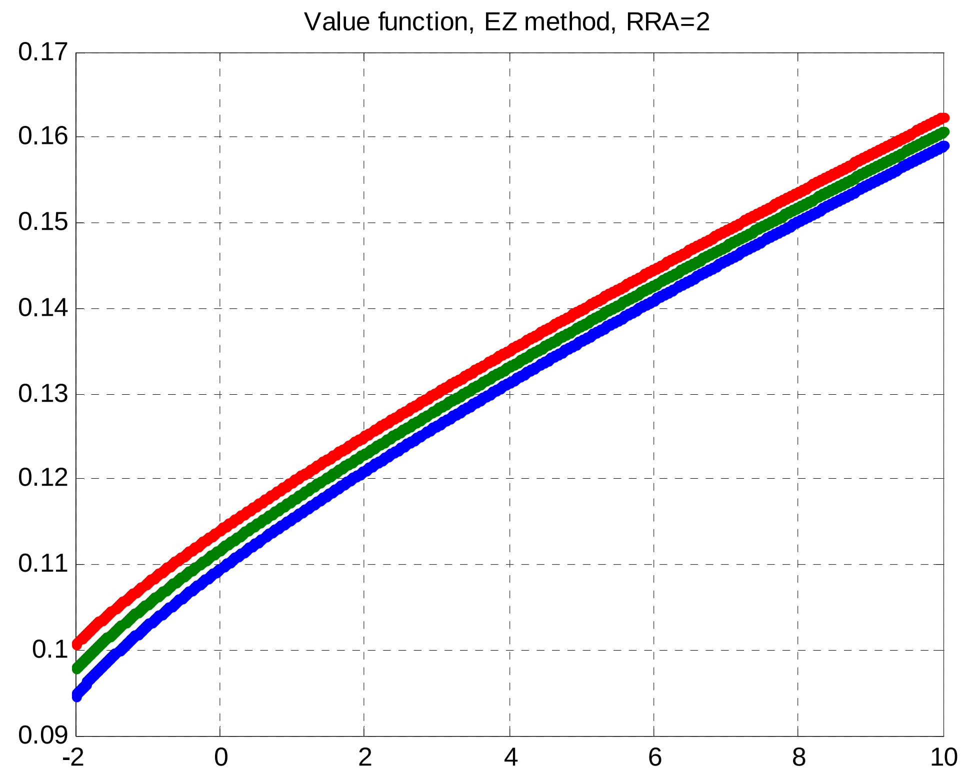

The following graphs illustrate the decision rules and value function as a function of individual's financial wealth.

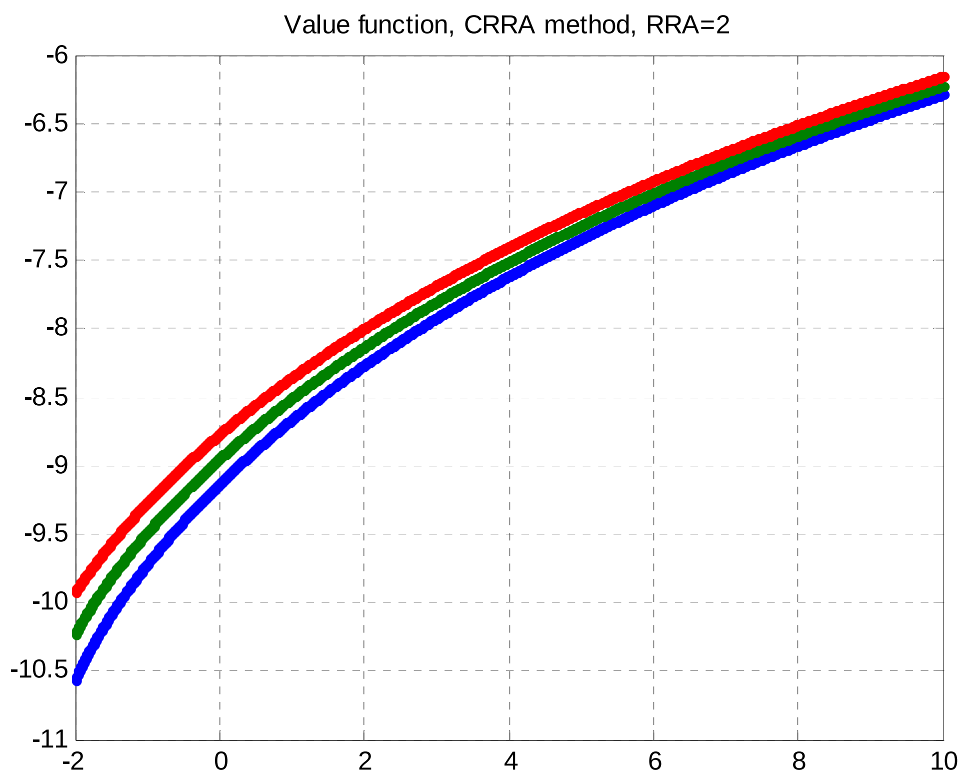

Optimal saving decison rule, RRA=2
$y_{low}$
$y_{medium}$
$y_{high}$
assets tomorrow
assets today
10
8
6
4
2
0
-2
-2
0
2
4
6
8
10

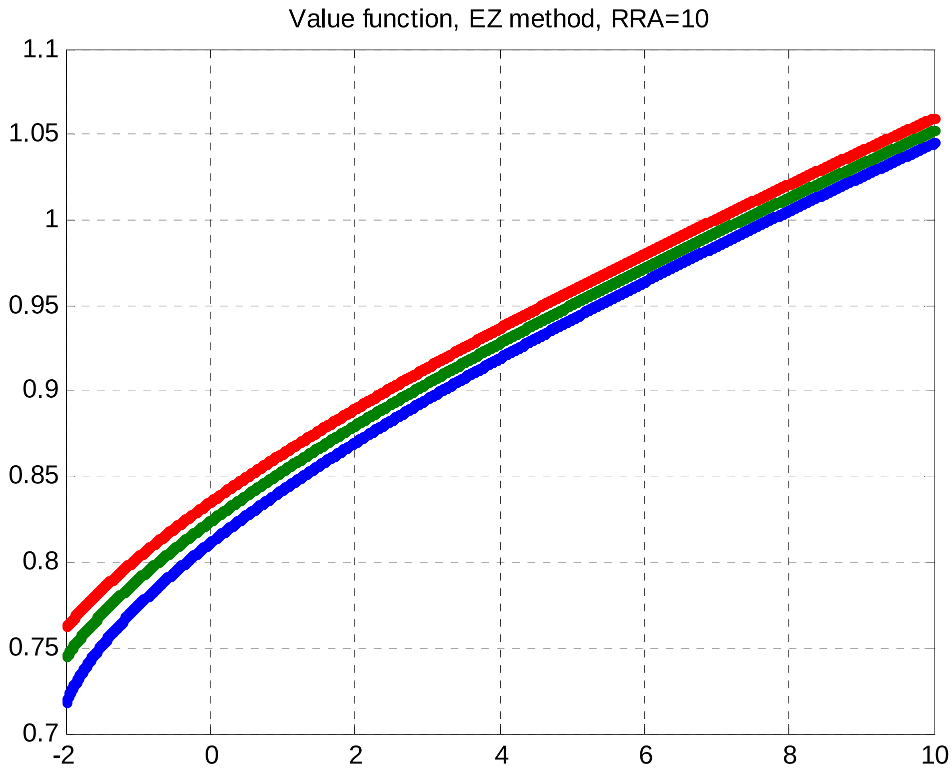

Optimal saving decison rule, RRA=10
$y_{low}$
$y_{medium}$
$y_{high}$
assets tomorrow
assets today
10
8
6
4
2
0
-2
-2
0
2
4
6
8
10

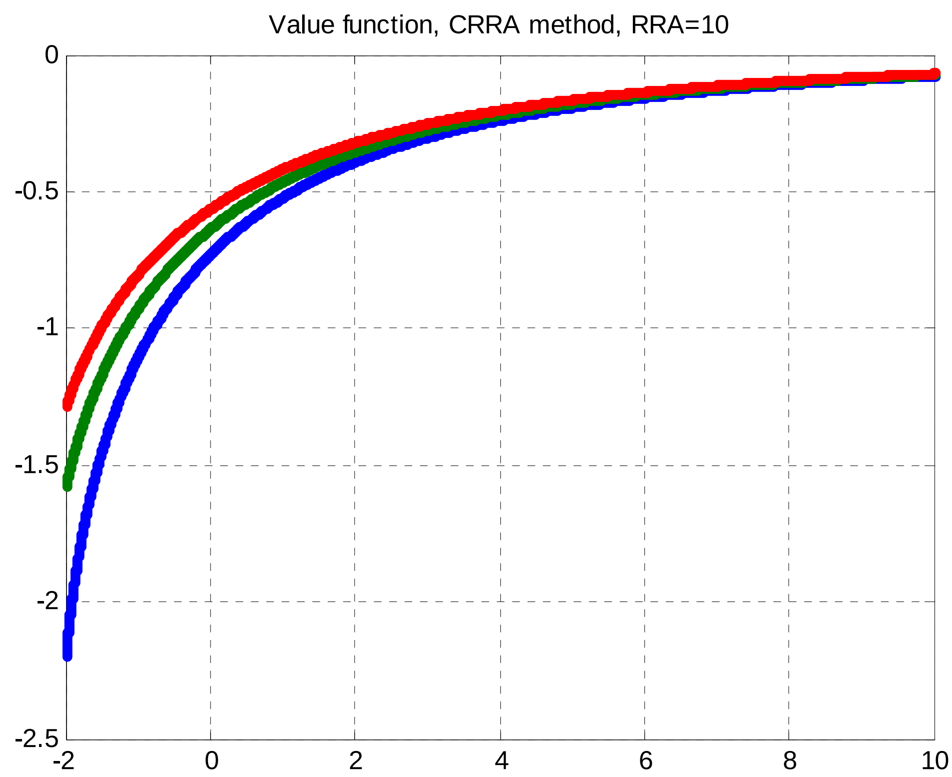

Optimal saving decison rule, RRA=2
assets tomorrow
assets today
-1.75
-1.8
-1.85
-1.9
-1.95
-2
-2
-1.95
-1.9
-1.85
-1.8
-1.75
$y_{low}$
$y_{medium}$
$y_{high}$


Optimal saving decison rule, RRA=10
assets tomorrow
assets today
-1.75
-1.8
-1.85
-1.9
-1.95
-2
-2
-1.95
-1.9
-1.85
-1.8
-1.75
$y_{low}$
$y_{medium}$
$y_{high}$


Value function, EZ method, RRA=2
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0.1
0.09
-2
0
2
4
6
8
10


Value function, CRRA method, RRA=2
-6
-6.5
-7
-7.5
-8
-8.5
-9
-9.5
-10
-10.5
-11
-2
0
2
4
6
8
10




As we expected, the value function in CRRA specification has more curvature than in EZ specification.