Pseudo-Code

Jenna Christensen and Kai Jensen

Describe the Returns and the Shock Process:

$R_{BC} = (R_{RF} + \mu)z$

$log(z') = \rho log(z) + \epsilon \ N(0, \sigma^2)$

$R_{BC}grid = (R_{RF} + \mu) * Zgrid$

Maximize the Following:

$$V(b, s) = max_{b',s'}\{\frac{c^{1-\gamma}}{1-\gamma} + E(\beta V(b', s'))\}$$

subject to

$$c + b' + s' = R_j b + R_s s$$
$$b' \geq b$$
$$s' > 0$$

Implement the Optimal Growth Model with Uncertainty:

1) Define parameters
2) Discretize the aggregate state/shock process into a finite grid (i.e. create Nz states)
3) Find the Markov transition probability using the Tauchen Algorithm
4) Create an iteration counter to make sure the solution is properly solved
5) Initiate a space for expected values of returns
6) Now, given current choices of Bitcoin, find which future Bitcoin maximizes the value function and store it
7) Choose the future Bitcoin that maximizes the lifetime value given each current Bitcoin and shock value
8) Update the initial guess to make sure the values of today and tomorrow are converging- use our current optimized value to replace our guess
9) Initiate spaces to store simulated results for realized values and suppose we start at a particular shock state
10) Randomly generate Nz numbers that follow the probability of the Markov Chain (have Matlab randomly return a single number that is uniformly distributed between zero and one)
11) Assume a starting amount of Bitcoin is held
12) Convert our simulated return values to our corresponding locations using the gridposition.m function file from Sakai