Marty Moesta

Computational Methods

10 May 2017

Computational Methods Final Project

Introduction

For my final project, I have performed a variant of the Timber Cutting Problem, originally made popular by Miranda & Fackler in an edition of their Applied Computational Finance and Economics textbook, with resources from Mason Gaffney. In the problem, an owner of a timber stand must decide at each time period t, usually considered a year, whether to cut down the stand and sell it for lumber, or allow the stand to develop for one more year. The stand biomass, *s*, grows at a deterministic rate where $s_{t+1} = g(s_t)$. The problem is a stochastic model, with a continuous state (time), and a discrete choice (whether or not to cut down the stand). The problem has an infinite horizon. The model has a binary choice in each state, where j = 1 means the stand is cut down, and j = 0 means that the stand continues to grow. The payouts are included below:

$$f(s,j) = \begin{cases} p\,s - K & j = 1 \\ 0 & j = 0. \end{cases}$$

K is equal to the cost of planting a new seedling and harvesting the biomass. The value of the stand, V(s), fits the conditions for Bellman's equation:

$$V(s) = \max\{\delta E_\epsilon[V(g(s))], p\,s - K + \delta E_\epsilon[V(0)]\}.$$

Methodology

In order to solve this problem, the collocation method is utilized. The method is flexible, accurate, and numerically efficient. The collocation method is a generalized form of interpolation, of which we studied linear and cubic spline in class. It uses a value approximant framework. To achieve this, a linear combination of *n* known basis functions whose *n* coefficients are fixed, by requiring that the function approximation satisfy the value function at *n* prescribed nodes, which are given to the model at the start of the simulation. Through this approach, the infinite-dimensional functional equation has become a finite-dimensional non-linear equation, which, in this case, will be solved using Newton's method[1]. The method then examines the residuals between the value function and the approximation at given points in order to give a reasonable idea of the accuracy of the solution. Since our initial formulation of the problem is fairly straightforward, we should expect the approximant value function to predict the actual value function quite well. The generalized form of collocation is given by:

$$V(s) \approx \sum_{j=1}^{n} c_j \phi_j(s).$$

With c_j representing each individual coefficient and Φ_j representing a given basis function for j = 1,n.

[1] Miranda and Fackler, "Computational Methods of Finance and Economics", Chapter 9, p 260-269.

Computational Methods

Actual computation in the model takes place in the MATLAB vector processing language. Being the first time using the MATLAB language, I spent considerable time understanding the framework and syntax of the language. I chose the MATLAB language because it is fast enough for the scope of my problem, and provides considerable documentation. In addition, Miranda and Fackler provide a CompEcon MATLAB library that helps the user solve problems using the collocation method. Integrating this library helped me better understand MATLAB syntax as well as solve the timber cutting problem.

Code Excerpts

In order to solve the problem in MATLAB I created an open-structured object *model* and then “packed” the model with the growth function, parameters and action variable (cut or no cut). The initial growth function is given by:

$$s_{t+1} = s_t + \chi * (s_{CC} - s_t)$$

Where s_{t+1} is the biomass in the next period, and s_{cc} is the carrying capacity of the biomass, or the largest the biomass can get. s_{cc} is a constant specified before the model is run. χ is the growth rate of the timber, given as constant to start. First, I enter model parameters:

```
% ENTER MODEL PARAMETERS
  price =  1;                                  % output price
  k     =  0.2;                                % replanting cost
  sbar  =  0.5;                                % carrying capacity
  gamma =  0.1;                                % growth rate
  delta =  0.95;                               % discount factor
```

In the model, ‘sbar’ = s_{CC}. Next, the action space and model structure is packed:

```
% CONSTRUCT ACTION SPACE
  x = [0;1];

% PACK MODEL STRUCTURE
  clear model
  model.func = 'timber';                       % reward/transition file
  model.discount = delta;                      % discount factor
  model.actions = x;                           % model actions
  model.params = {price k sbar gamma};         % other parameters
```

Where ‘timber’ is a file that contains the growth function. Finally, I need to set the amount of *n* nodes that help calculate the value function approximation, and provide initial guesses for those nodes (somewhere for the model to start its iterations):

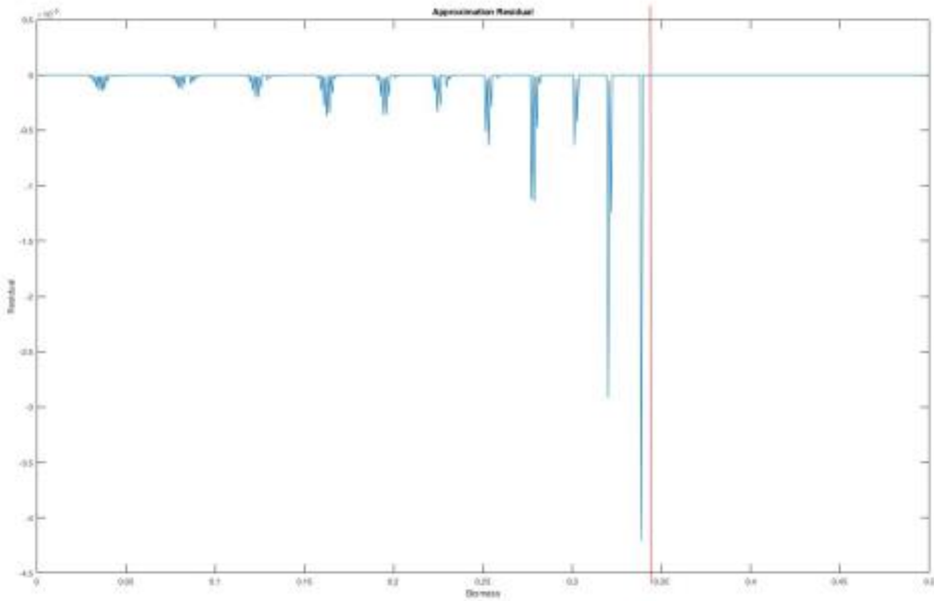
```
% DEFINE APPROXIMATION SPACE
  n      = 350;                                % degree of approximation
  fspace = fundefn('lin',n,0,sbar);            % function space
  snodes = funnode(fspace);                    % state collocaton nodes

% Provide inital guesses
  vinit = zeros(size(snodes));
```

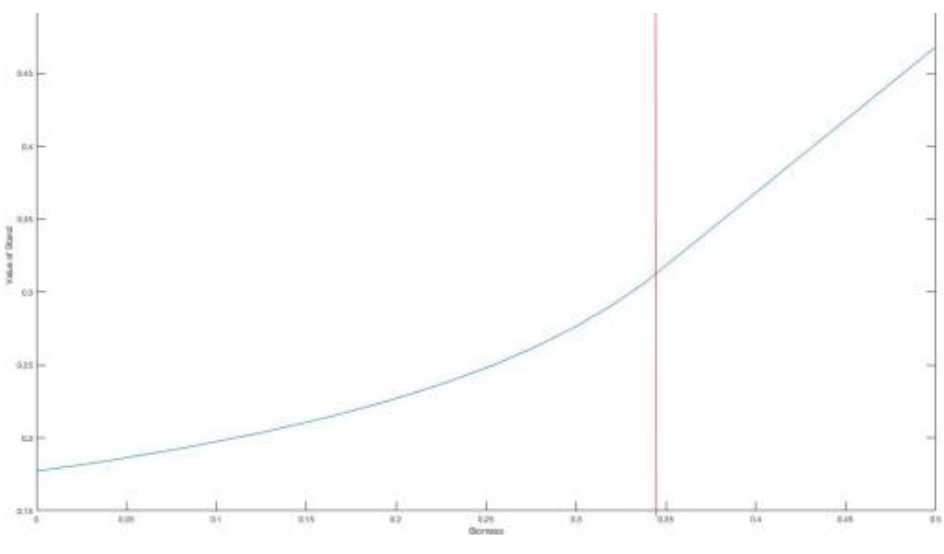
The ‘funnode’ function creates linearly-spaced nodes between 0 and s_{CC} for function approximation, while ‘fspace’ becomes the projection space structure.

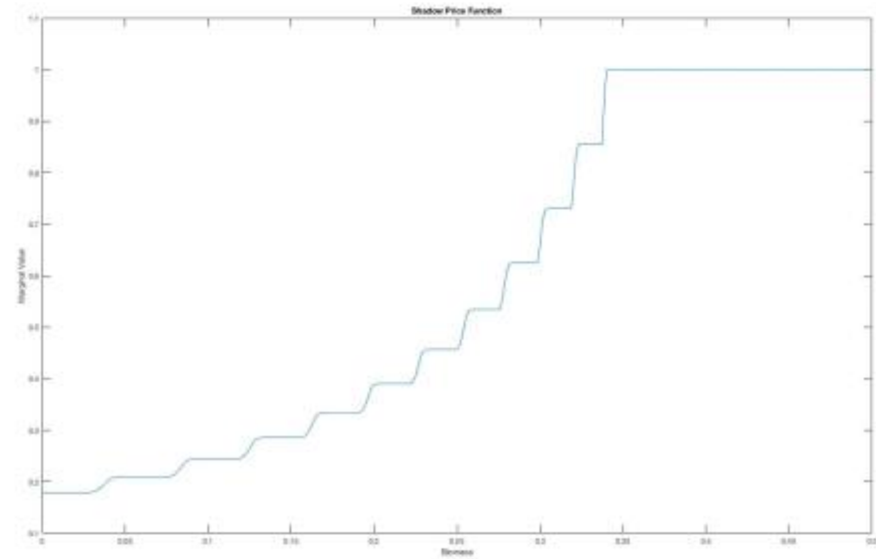
Results

With the initial specified parameters, the model returns the follow results:

Optimal cutting stock level: 0.34

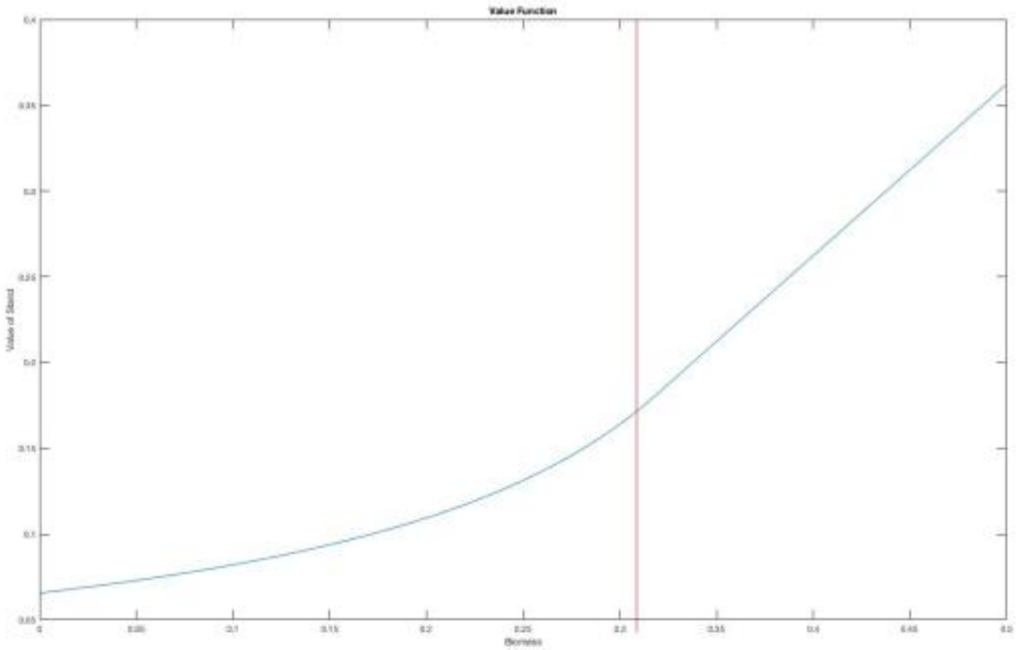




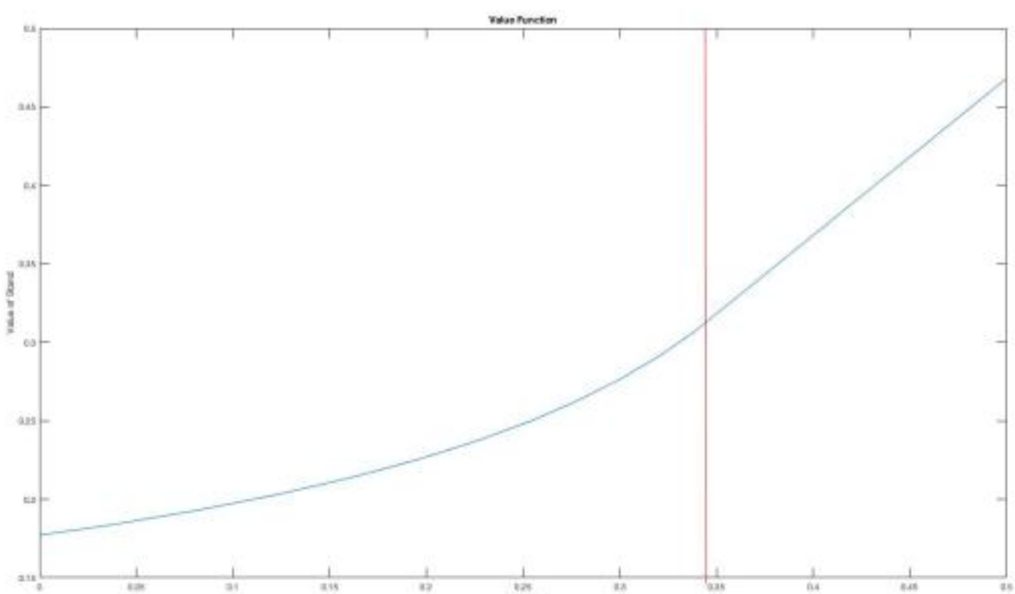


The graphs above show the approximation residual, the value function, and marginal value v. biomass, respectively. The residuals seem to minimize to zero at 0.34, and the marginal value of the stand biomass reaches a constant level, but what is happening with the value function? What is the intuition behind this curve? To investigate this further, we will examine the problem at different growth rates χ, and different β discount factors.

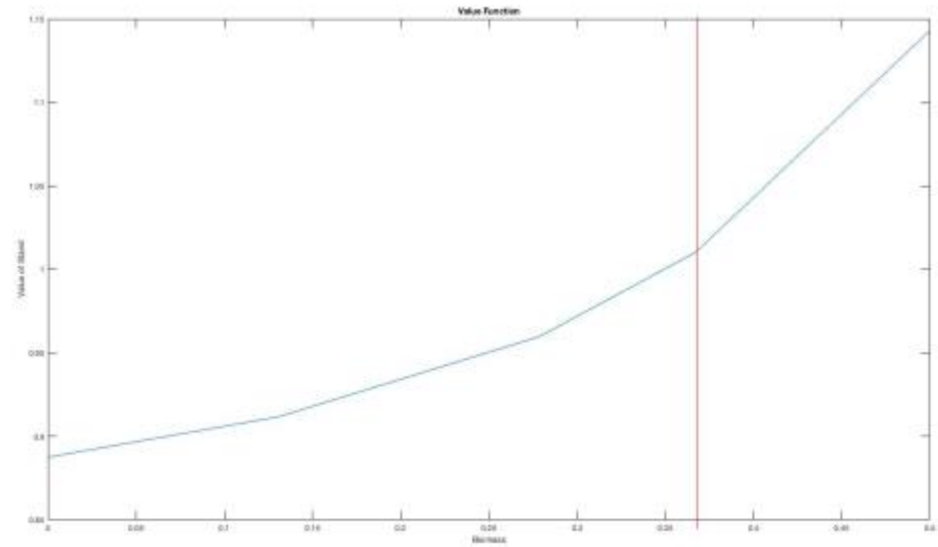
$\chi = 0.05$



$\chi = 0.1$

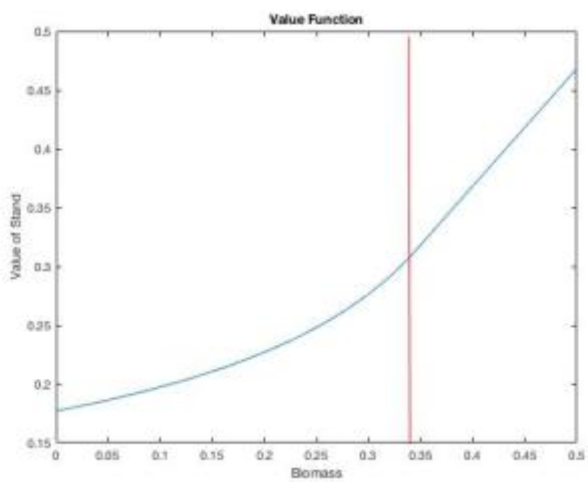


$\chi = 0.4$

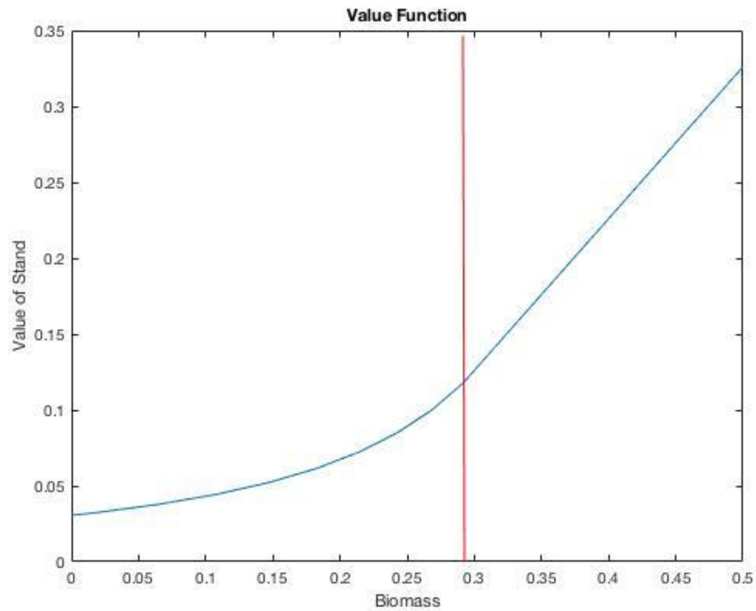


Examining different χ values shows us that as χ increases, the approximation of the value function loses its smoothness, becoming more linearly segmented. Also, the curvature of V is steeper when χ is lower, as well as high χ elicit monotonically greater value functions, as well as optimal cutting lengths that are closer and closer to s_{cc}, as indicated by the red line. Now holding χ constant, we will examine the approach at different discount factors.

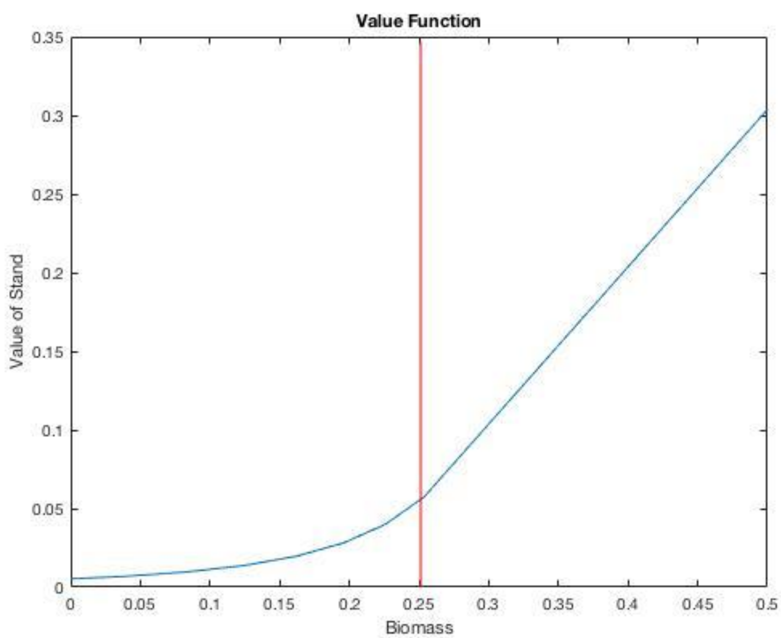
$\beta = 0.95$



β = 0.85

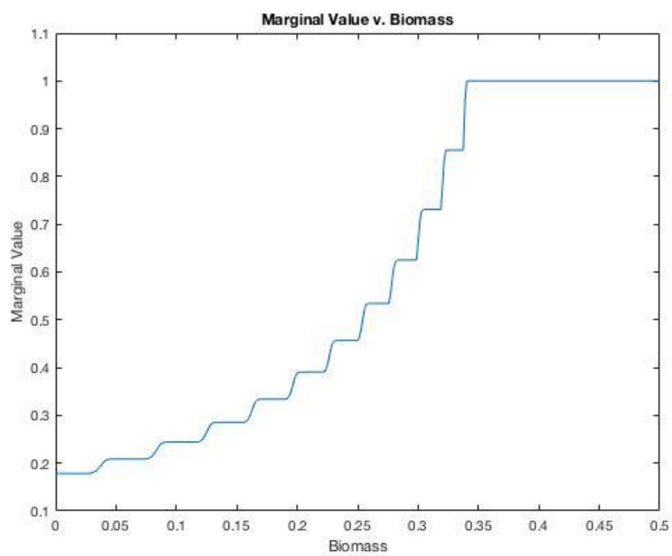


β = 0.7

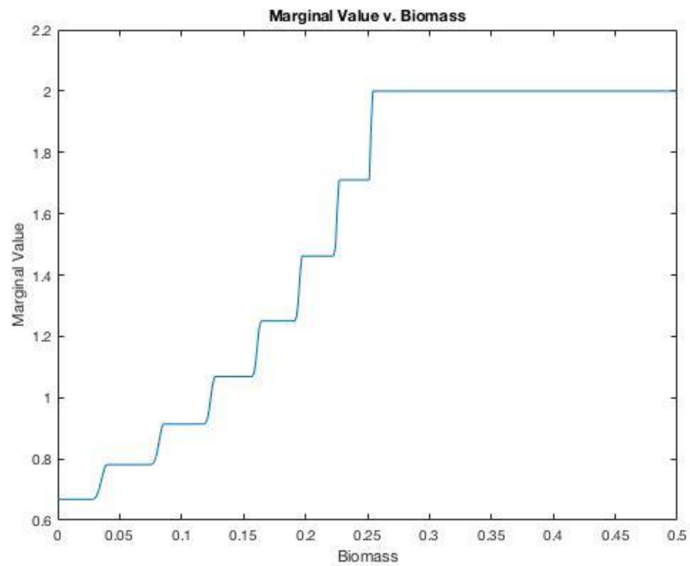


As the biomass grows, the value function increases at a rate that is proportional to β. At low β values, the value of the stand increases slowly until optimal cutting value, while higher β values elicit a curve that is that increases at a rate closer to the slope between $s_{optimal}$ and s_{cc}. This is intuitive, showing that the stand owner values the future of the stand less at lower β values, and once s = optimal cutting, the value function begins to grow at a continuous rate. Turning now towards *why* the optimal value is where it is, it becomes evident that marginal returns to value will reach a constant rate = price p, and once the biomass reaches a mass s such that the M.V.= p, the biomass owner should cut the stand.

P = 1



P = 2



Marginal value graphs for variations in χ and β yielded similar looking graphs, with some degree of a step function leading up to a plateau at price level *p*.

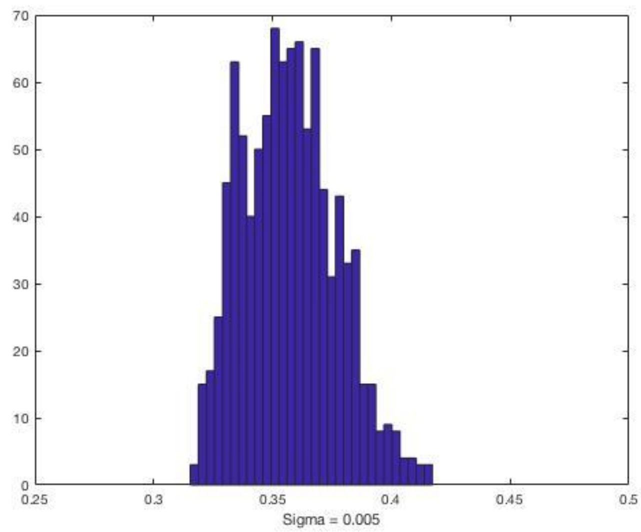
Adding Randomness to Growth Function

Now that we have some intuition for how each of the parameters is shaping the graph, as well as determining optimal cutting points, let's see how these optimal values change when a randomness is added into the growth function. In practice, this can be interpreted as fluctuations in climate due to sunlight, precipitation, etc. Our new growth function adds a simple error term $\varepsilon = N(0,\sigma)$:

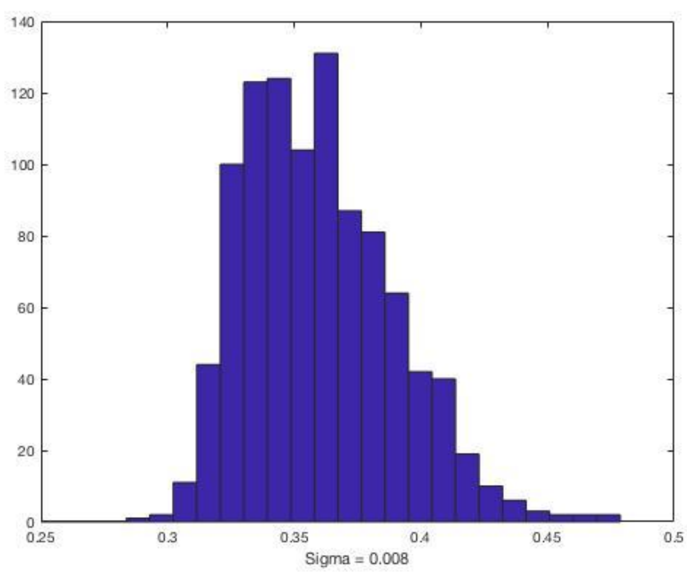
$$S_{t+1} = s_t + \chi^*(s_{cc} - s_t) + \varepsilon$$

With $\sigma = 0.005, 0.008, 0.01$. I iterated the collocation procedure 1000 times and recorded the optimal cutting level at each iteration for different σ values. Below are the histograms of optimal cutting:

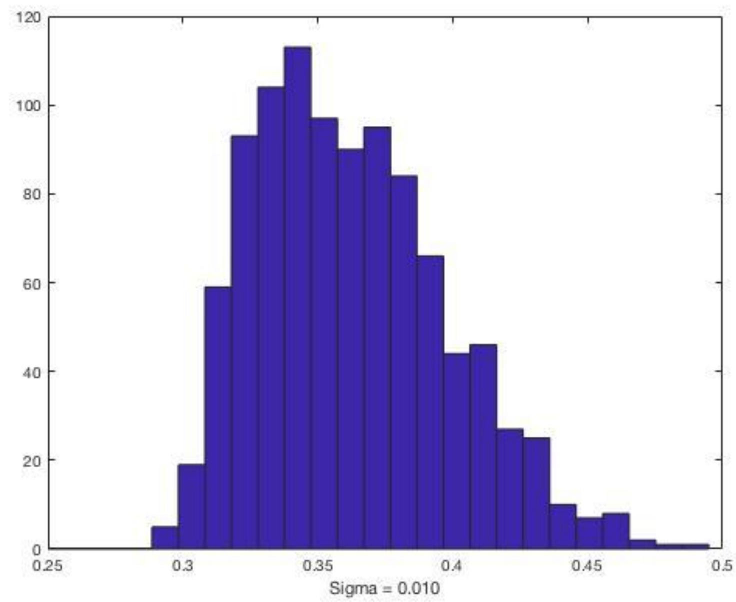
$\sigma = 0.005$



$\sigma = 0.008$



$\sigma = 0.01$

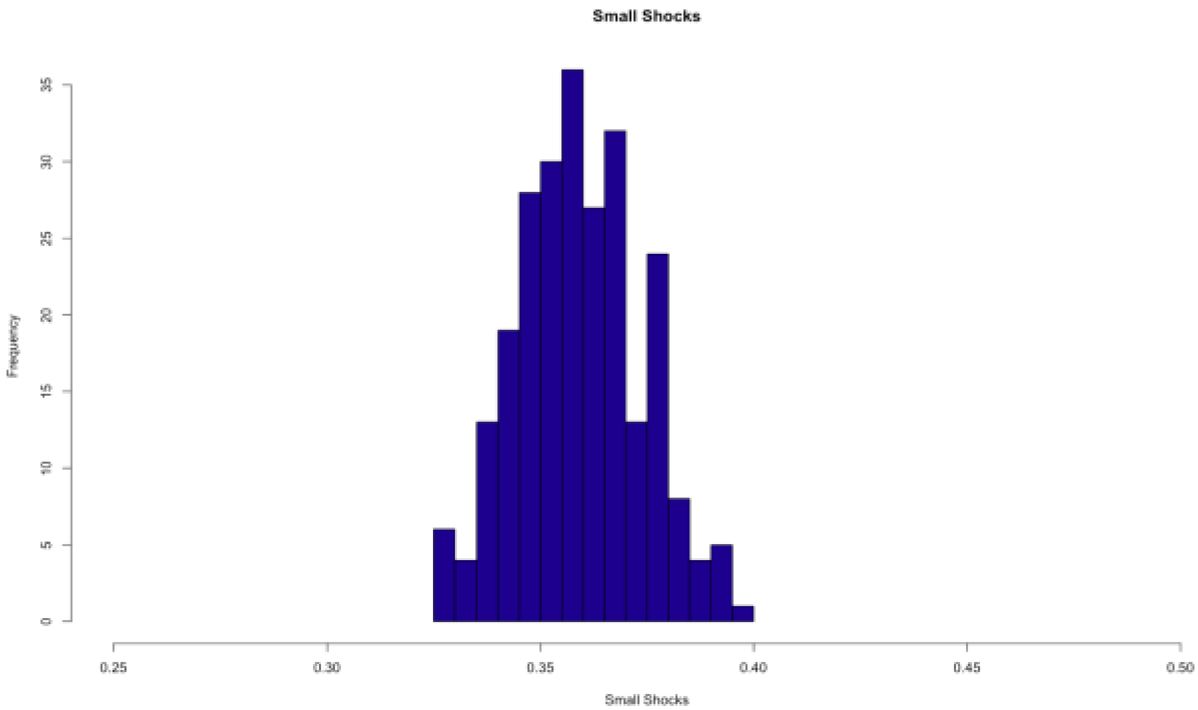


As expected, the variation in optimal cutting value increases proportionally with σ. The distribution also seems to get skewed right as σ increases.

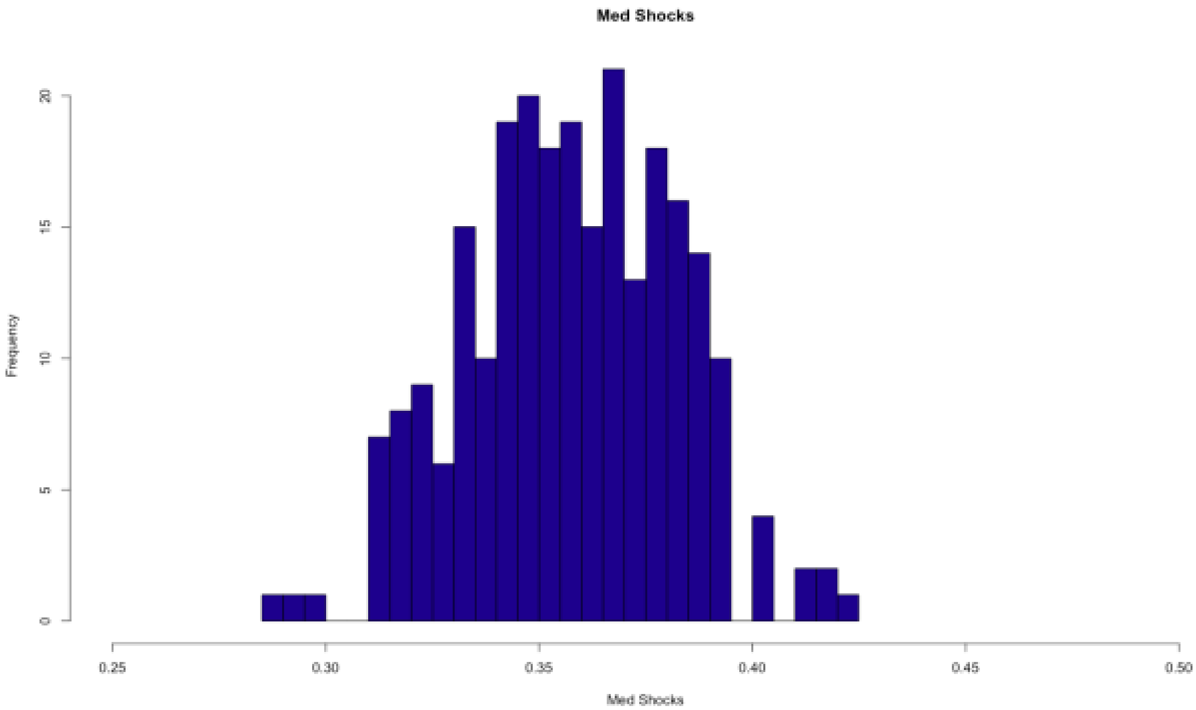
Adding Shocks

The final alteration to the model looks to add a series of discrete and normally distributed, random i.i.d. price shocks *z* and examine their respective effects on optimal cutting procedure. The iteration from the previous section will be replicated here, but instead of different σ values, different vectors of shocks will be inserted into the model to determine optimal cutting length subject to different variations in shock price z**p.*

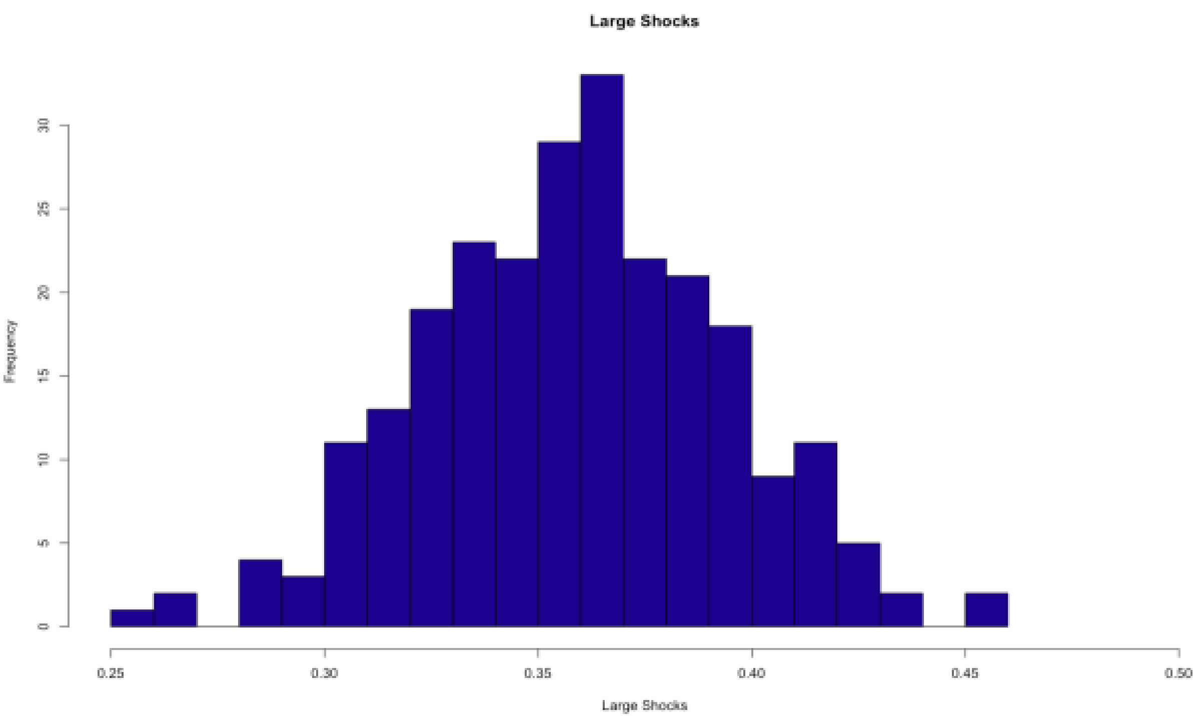
z = [0.9,0.95,1.0,1.05,1.1]



z = [0.86, 0.93,1.0.1.07,1.14]



z = [0.8.0.9,1.0,1.1,1.2]



Conclusions

This model was lightweight and flexible enough to perform numerous variants to the original problem. In addition, I was able to determine the pattern behind optimal cutting decisions based off of marginal value by looking at different beta, gamma, and price values. The CompEcon MATLAB library from Miranda and Fackler allowed me to integrate the collocation approach to my problem fairly easily. Using a combination of Newton and Collocation, I balanced implementing techniques learned in class with methods outside of course syllabus, all while operating with a somewhat foreign programming language. Thank you for everything Professor! I learned a lot and was challenged!