High dimensional projection methods

Felix Kubler

\(^1\)DBF, University of Zurich and Swiss Finance Institute

October 11, 2017
Overview

1. Models with finitely many agents – projection methods
   - Sparse-grid collocation for high-dimensional problems
   - Adaptive sparse grids
Literature

- **Economics:**
Math:

The exchange economy

- $L$ perishable commodities, exogenous shocks follow Markov-process $(z_t)$ with support $\mathbf{Z}$. Histories are $\sigma = z^t = (z_0, \ldots, z_t)$
- Each period $H$ agents, $h = 1, \ldots, H$ are born, live for $A$ periods
- Individual endowments are $e^{z^t,h}(z^{t+a-1}) = e_{a,h}(z_{t+a-1})$, preferences are

$$U_{z^t,h}(x_{z^t,h}) = E_t \sum_{a=1}^{A} (\prod_{i=1}^{a} \delta_{a,h}) u_{a,h} \left(x_{z^t,h}(z^{t+a-1})\right),$$

where $\delta_{a,h} \in [\underline{\delta}, \overline{\delta}]$ and $u_{a,h}(.)$ satisfies all assumptions of smooth preferences
- Lucas tree in unit net supply, drops fruits $d(z_t) \in \mathbb{R}^L_+$. Aggregate endowment is $\bar{\omega}(z_t) = d(z_t) + \sum_{(a,h)} e_{a,h}(z_t)$
First suppose that the tree is the only asset available for trade, markets are incomplete. $L = 1$. Easy to write problem recursively...

We will later look at the case where there are also financial securities that complete the markets and possibly borrowing constraints.
An abstract environment, FREE

- State space $\mathbf{S} = \Theta \times \mathbf{Z}$,
- policy and pricing functions $F(\theta, z)$ satisfy
  \[
  g\left(\theta, z, F(\theta, z), E_z \left[ h\left( F(\theta, z), F\left( F(\theta, z), z' \right), z' \right) \right] \right) = 0
  \]
  for all $(\theta, z) \in \Theta \times \mathbf{Z}$.
- $g$ and $h$ are (known) smooth functions
- Want algorithms to approximate $F$
(Smooth) functions $\theta^{(a,h)}_z : \Theta \to \mathbb{R}$ as well as $q_z : \Theta \to \mathbb{R}_+$ are defined on $d = (A - 1)H - 1$ dimensional boxes $\Theta_z$, for all $z \in Z$, such that for all $\theta_- \in \Theta$, $z \in Z$,

$$u'_{a,h}(a_{a,h}(z) + \theta_-(q_z(\theta_-) + d(z))) - \theta_z(\theta_-) + \delta_{a,h}$$

$$\sum_{z'} \pi(z'|z)(q_{z'}(\theta_z(\theta_-) + d(z')))$$

$$u'_{a+1,h}(a_{a+1,h}(z) + \theta_z(\theta_-)(q_{z'}(\theta_z(\theta_-) + d(z'))) - \theta_{z'}(\theta_z(\theta_-)) = 0$$

$$\sum_{a,h} \theta_z(\theta_-) = 1$$
Why projection methods?

- Easy to formulate, independently of number of assets, utility and production functions (as long as first order conditions characterize optimality)
- Programming costs fairly high, but once toolbox is built up, very mechanical
- Comparative advantage compared to perturbations methods when solution is ‘very non-linear’
- HPC for time-iteration collocation
A solution method - time iteration collocation

0: Select a d-box $\Theta$, a family of functions $\hat{F}$ which can be parametrized by numbers

1: Select a finite grid $G \subset \hat{^}$ of collocation points and the parameters $\xi(0)$ for a starting $\hat{F}^0$

2: Given parameters $\xi(n)$ and thus the function $\hat{F}^n$, $\forall \theta \in G$, $\forall z \in Z$, solve system

$$g \left( \theta, z, x, E_z \left[ h \left( x, \hat{F}^n(x, z'), z' \right) \right] \right) = 0$$

for the unknown $x$

3: Compute the new coefficients $\xi(n + 1)$ by interpolation of the solutions in 2

4: Check some stopping criterion, if not satisfied, go to 2

5: Conduct error analysis
What is needed

- Approximation of policy functions
  - How to select $\Theta$: What variable to use? Domain?
  - How to select $G$
  - Interpolation, picking $\xi(0)$
- Solving nonlinear equations
- Integration
- Acceleration (e.g. Axelsson (1996))
To come

* High-dimensional interpolation
Approximating the policy functions

- We want to approximate $f : [0, 1]^d \rightarrow \mathbb{R}$, using sparse grids.
- Number of points in sparse grids grow slowly with dimension, approximation error of full grids is preserved up to logarithmic constant.
  1. Piecewise $d$-linear approximation on sparse grids.
  2. Smolyak’s method for polynomial interpolation.
  3. Adaptive sparse grids.
Piecewise linear interpolation

\[ y(x) = x^2 \sin(nx) \]
Piecewise linear interpolation

$y(x) = x^2 \sin(nx)$
Piecewise linear interpolation

\[ y(x) = x^2 \sin(nx) \]
Piecewise linear interpolation

\[ f(x) = x^2 \sin(\pi x) \]

\[ u(x) \]

\[ \alpha_{1,1} \]

\[ \alpha_{2,3} \]

\[ \alpha_{2,1} \]
Piecewise linear interpolation
Piecewise linear approximation

- In one dimension, we take as basic function on \([-1, 1]\)

\[
\phi(x) = \max(0, 1 - |x|)
\]

and twist them to generate a family of basis functions on \([0, 1]\)

\[
\phi_{l,i}(x) = \phi(2^l x - i), \ i = 1, \ldots, 2^l - 1, \ i \text{ odd}
\]

- Define

\[
l_l = \{i \in \mathbb{N} : 1 \leq i \leq 2^l - 1, \ i \text{ odd}\}
\]

and

\[
W_l = \text{span}\{\phi_{l,i}, i \in l_l\}
\]

The space of piecewise linear functions is then

\[
V_n = \bigoplus_{l \leq n} W_l
\]
Piecewise linear approximation - coefficients

\[ f(x) = x^2 \sin(\pi x) \]

\[ u(x) \]

The coefficients, \( \alpha_{k,i} \) are *hierarchical surpluses*. They correct the interplant of level \( l-1 \) at \( x_{l,i} \) to the actual value of \( f(x_{l,i}) \).

Become small as approximation becomes better.
Piecewise d-linear interpolation

Take as basic function on $[-1, 1]^d$ the tensor product of 1-dimensional basis functions:

$$\phi(x) = \prod_{j=1}^{d} \phi(x_j)$$

$$z = (1 - |y|)(1 - |x|)$$
Piecewise d-linear interpolation

- Take as basic function on $[-1, 1]^d$ the tensor product of 1-dimensional basis functions:

$$\phi(x) = \prod_{j=1}^{d} \phi(x_j)$$

- As before, we look at $[0, 1]^d$, define

$$l = (l_1, \ldots, l_d) \in \mathbb{N}^d$$

and

$$i = (i_1, \ldots, i_d) \in \mathbb{N}^d$$

- Define

$$W_l = \text{span}\{\phi_{l,i} : i \in l\}$$

with $l_l = \{i : 1 \leq i_j \leq 2^{l_j} - 1, i_j \text{ odd}, 1 \leq j \leq d\}$
Basis functions of $W_{2,1}$
Full grids

- Take

\[ V_n = \bigoplus_{|l|_\infty \leq n} W_i \]

with piecewise d-linear approximation

\[ f(x) \approx u(x) = \sum_{|l|_\infty \leq n, i \in I} \alpha_{l,i} \phi_{l,i}(x) \]

- Unfortunately, number of terms grows at an order $2^{nd}$, making this method infeasible for $d > 4$. 
Sparse vs full grids

- Assume bounded mixed second derivatives of $f$. Given a fixed number of points, which grid refinements do we use (i.e. which $W_l$) in order to minimize $L^2$ error.
- Turns out that this can be solved and one gets as space of functions

$$V_n^S = \bigoplus_{||l||_1 \leq n+d-1} W_l$$
Sparse vs full grids
Sparse vs full, Level 4
**Sparse vs full grids**

- Number of points grows much slower in $d$
- For example, for refinement level $n = 4$ we obtain

<table>
<thead>
<tr>
<th>$d$</th>
<th>points in full grid</th>
<th>points in sparse grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>225</td>
<td>49</td>
</tr>
<tr>
<td>3</td>
<td>3375</td>
<td>111</td>
</tr>
<tr>
<td>4</td>
<td>50625</td>
<td>209</td>
</tr>
<tr>
<td>5</td>
<td>759375</td>
<td>351</td>
</tr>
<tr>
<td>10</td>
<td>$5.77 \cdot 10^{11}$</td>
<td>2001</td>
</tr>
<tr>
<td>15</td>
<td>$4.37 \cdot 10^{17}$</td>
<td>5951</td>
</tr>
<tr>
<td>20</td>
<td>$3.33 \cdot 10^{23}$</td>
<td>13201</td>
</tr>
<tr>
<td>30</td>
<td>$1.92 \cdot 10^{35}$</td>
<td>41601</td>
</tr>
<tr>
<td>50</td>
<td>$6.38 \cdot 10^{58}$</td>
<td>182001</td>
</tr>
<tr>
<td>100</td>
<td>$&gt; 10^{100}$</td>
<td>1394001</td>
</tr>
</tbody>
</table>
Smolyak’s method

- Piecewise linear approximation is theoretically well understood and numerically stable.
- In economics, policy and pricing functions are sometimes ‘close’ to being relatively low degree polynomials.
- In these cases it might be advantageous to use polynomials for a global approximation of the unknown functions.
- Smolyak’s method is a simple way to use the sparse grid construction for polynomials.
Smolyak’s method

- Sequence of 1-D interpolation points $\chi^i \subset [-1, 1]$, $i=1,2,...$
- Define $m_1 = 1$ and $m_i = 2^{i-1} + 1$, $i > 1$ to be the total number of elements of set $\chi^i$
- Choose $\chi^1 = \{0\}$ and for $i > 1$, $\chi^i = \{x_1^i, ..., x_{m_i}^i\} \subset [-1, 1]$ as the set of the extrema of the Chebyshev polynomials

$$x_j^i = -\cos \frac{\pi(j - 1)}{m_i - 1} \quad j = 1, ..., m_i$$

- So $\chi^1 = \{0\}$, $\chi_2^{\Delta} = \{-1, 1\}$, $\chi_3^{\Delta} = \{\cos(\frac{3\pi}{4}), -\cos(\frac{3\pi}{4})\}$ and

$$\chi_4^{\Delta} = \{-\cos(\frac{\pi}{8}), -\cos(\frac{3\pi}{8}), \cos(\frac{\pi}{8}), \cos(\frac{3\pi}{8})\}$$
Smolyak’s method

- For a given level, \( l \) define a \( d \)-dimensional grid as
  \[
  \mathcal{H}(l, d) = \bigcup_{|\mathbf{i}| \leq d + l} (\chi^{i_1} \times \ldots \times \chi^{i_d}),
  \]

- The Smolyak construction of an interpolating polynomial is then
  \[
  \mathcal{A}(l, d) = \sum_{|\mathbf{i}| \leq d + l} (-1)^{d+l-|\mathbf{i}|} \binom{d-1}{d+l-|\mathbf{i}|} \alpha_{i_1,\ldots,i_d}(\mathcal{U}_1^{i_1} \otimes \ldots \otimes \mathcal{U}_d^{i_d})
  \]

\( \mathcal{U} \) are interpolating polynomials (e.g. Chebyshev or Lagrange)
Smolyak’s grid
Remarks

- \( A(2, d) \) reproduces the polynomials \( x_j^4, x_j^3, x_j^2, x_j, 1, x_j^2 x_k^2, x_j^2 x_k, x_j x_k \).
- \( A(k, d) \) is exact for polynomials up to degree \( k \).
- The number of points in \( \mathcal{H}(l, d) \) is given by
  - \( l = 1 : 1 + 2d \)
  - \( l = 2 : 1 + 4d + 4 \frac{d(d-1)}{2} \)
  - \( l = 3 : 1 + 8d + 16 \frac{d(d-1)}{2} + 8 \frac{d(d-1)(d-2)}{6} \)
- Number of points grows exponentially in \( l \).
- Method only really useful for \( l = 2 \) or \( l = 3 \).
- For large \( d \) (e.g. 50) need a lot of smoothness for a good approximation.
Maliar and Maliar have matlab code on their website: http://stanford.edu/maliars/files/Codes.html
Adaptive sparse grids

- Sparse grids with local approximations have the huge advantage that it is easy to construct *adaptive* grids.
- Simply add points where hierarchical surplus (abs. value of coefficients) is large.
- Formally, each grid point has $2d$ neighbors, add a grid point if and only if hierarchical surplus is greater than some $\varepsilon$
Hierarchical surplus

\[ x(x) = x^2 \sin(\pi x) \]
Hierarchical surplus

\[ x(x) = x^2 \sin(\pi x) \]
Hierarchical surplus

\[ x(x) = x^2 \sin(\pi x) \]

\[ u(x) \]

\[ \alpha_{1,1} \]

\[ \alpha_{2,3} \]

\[ \alpha_{2,1} \]

\[ x_{2,1} \]

\[ x_{1,1} \]

\[ x_{2,3} \]
Hierarchical surplus

\[ x(x) = x^2 \sin(\pi x) \]

\[ u(x) \]

\[ \alpha_{1,1}, \alpha_{2,3}, \alpha_{3,5}, \alpha_{3,7}, \alpha_{3,1}, \alpha_{2,1}, \alpha_{3,3} \]
Hierarchical surplus

\[ x(x) = x^2 \sin(\pi x) \]
Adaptive sparse grids

Consider test function

\[ f(x) = \frac{1}{|0.5 - x^4 - y^4| + 0.1} \]

How many grid points do we need to get an error of 0.001?
- Full grid: around $10^9$
- Sparse grid: 311297
- Adaptive sparse grid: 4411
Funny function
Adaptive sparse grids
Adaptive sparse grids
Adaptive sparse grids
Adaptive sparse grids

Refinement Level 4

0.0 0.2 0.4 0.6 0.8 1.0
0.0 0.2 0.4 0.6 0.8 1.0

Felix Kubler
Comp.Econ. Gerzensee, Ch6
October 11, 2017
Adaptive sparse grids

Refinement Level 5

0 0.2 0.4 0.6 0.8 1

0 0.2 0.4 0.6 0.8 1
Adaptive sparse grids
Adaptive sparse grids
Adaptive sparse grids
Adaptive sparse grids
Adaptive sparse grids
Adaptive sparse grids
Some cautionary remarks

- Need to get lucky with initial level
- If kink does not get detected at initial level it might never get detected
- But easy to use information about kink in setup of problem
Adaptive time iteration collocation

0: Select a d-box $\Theta$, a family of functions $\hat{F}$ which can be parametrized by numbers

1: Select a finite grid $G^0 \subset \hat{\mathcal{F}}$ of collocation points and the parameters $\xi(0)$ for a starting $\hat{F}^0$

2: Given parameters $\xi(n)$ and thus the function $\hat{F}^n, \forall \theta \in G^n, \forall z \in Z$, solve system

$$g \left( \theta, z, x, E_s \left[ h \left( x, \hat{F}^n(x, z'), z' \right) \right] \right) = 0$$

for the unknown $x$

3: Compute the new coefficients $\xi(n + 1)$ by interpolation of the solutions in 2

4: Refine Grid, new grid $G^n$

5: Check some stopping criterion, if not satisfied, go to 2
Sparse Grids – Summary

- To solve problems with several agents one needs to approximate high-dimensional functions
- Tensor product is not a good space for this
- Sparse grids, either using piecewise d-linear functions or polynomials (Smolyak) can handle around 50 dimensions when functions are smooth
- Adaptive sparse grids provide a way to handle up to 100 dimensions when functions are smooth, 30-40 when they have nondifferentiabilities
Sparse Grids – Software: Tasmanian

- Toolkit for Adaptive Stochastic Modeling and Non-Intrusive Approximation
  Available at: http://tasmanian.ornl.gov

- A suite of C++ routines that implements a number of different function basis. In particular, one can choose between global grids (Smolyak) and local polynomial grids (local d-linear approximation and more). Local grids support iterative refinement

- Three components
  1. library written in C++ that implements the \textit{TasmanianSparseGrid} class
  2. An executable that reads and writes data to text files
  3. A MATLAB interface to use the executable
Sparse Grids – Other Software

- Sparse Grid Interpolation Toolbox (Matlab)
  
  http://www.ians.uni-stuttgart.de/spinterp/index.html

  Is not maintained, last update 2008!!!
Simple OLG model
Physical economy

- Time is $t=0,1,...$
- Aggregate shocks ($z_t$) follow Markov chain
- Every period continuum of ex ante identical agents enters economy. Live for $A$ periods.
- Supply labor inelastically to firm, consume a single good and save in risky capital. Epstein-Zin utility function.
- Representative competitive firms has neoclassical production function and buys labor at wage $w$ and rents capital at price $r$. 
Aggregate shocks $z_t \in \mathbf{Z}$, follow Markov chain with transition $\pi(z'|z)$

Idiosyncratic shocks $y_a \in \mathbf{Y}$ follow a Markov chain with transition $\eta(y'|y)$

At $z^t$ an agent is identified by $y^a \in \mathbf{Y}^a$ and maximizes recursive utility

$$U_{y^a}(z^t) = \left\{ \left[ c_{y^a}(z^t) \right]^\rho + \beta_a \left[ \sum_{z_{t+1}} \pi(z_{t+1}|z_t) \sum_{y_{a+1}} \eta(y_{a+1}|y_a) \left( U_{y_{a+1}}(z^{t+1}) \right)^\sigma \right]^{\frac{1}{\rho}} \right\}^{\frac{1}{\rho}}$$

subject to: $c_{y^a}(z^t) + k_{y^a}(z^t) = (1 + r(z^t))k_{y^{a-1}}(z^{t-1}) + l_{y^a}(z^t)w(z^t)$

In this version: No risk-free bond!
Firm has production function $f(K, L; z_t) = \xi(z_t)K^\alpha L^{1-\alpha} + K(1 - \delta(z_t))$

$$L(z^t) = \sum_{a=1}^{A} \sum_{y^a \in Y^a} \nu(y^a)l_{y^a}(z^t)$$

Market clearing:

$$K(z^t) = \sum_{a=1}^{A} \sum_{y^a \in Y^a} \nu(y^a)k_{y^a}(z^{t-1})$$
Case 1: No idiosyncratic shocks

Take \( s(z^t) = \left( \sum_{a=1}^{A-1} k_a(z^{t-1}), (1 + r(z^t))k_1(z^{t-1}), \ldots, (1 + r(z^t))k_{A-2}(z^{t-1}) \right) \)

- We approximate the unknown equilibrium asset-demand and value functions \( k_i(z,\cdot), v_i(z,\cdot), i=1,\ldots,A-1 \), by piecewise multilinear functions that are uniquely defined by finitely many coefficients.

- In order to solve for the unknown coefficients, we require that the functional defining a recursive equilibrium holds exactly at \( M \) collocation points.

- We transform the infinite dimensional functional equation into the following finite dimensional (nonlinear) system of equations.
Case 1: Equations

\[
\hat{c}_a(z, \tilde{s} | \alpha)^{\rho^{-1}} + \beta_a \left[ \sum_{z'} \pi(z' | z) \left( \hat{v}_{a+1}(z', s' | \alpha) \right)^{\sigma \rho^{-1}} \right]^{\rho \sigma^{-1}} = 0, \quad a = 1, \ldots, A - 1
\]

\[
\hat{v}_a(z, \tilde{s} | \alpha) = \hat{c}_a(z, \tilde{s} | \alpha)^{\rho} + \beta_a \left[ \sum_{z'} \pi(z' | z) \left( \hat{v}_{a+1}(z', s' | \alpha) \right)^{\sigma \rho^{-1}} \right]^{\rho \sigma^{-1}}, \quad a = 1, \ldots, A - 1
\]

\[
s' = \left( s'_1, \hat{k}_1(z, \tilde{s})(1 + r(z', s'_1)), \ldots, \hat{k}_{(A-2)}(z, \tilde{s})(1 + r(z', s'_1)) \right) \in B
\]

\[
\hat{c}_1(z, s | \alpha) = l^1(z)w(z, s_1) - \hat{k}_1(z, s; \alpha)
\]

\[
\hat{c}_i(z, s; \alpha) = s_i + l^i(z)w(z, s_1) - \hat{k}_i(z, s; \alpha) \quad \text{for} \quad i = 2, \ldots, N - 1
\]

\[
\hat{c}_N(z, s; \alpha) = \left( s_1(1 + r(z, s_1)) - \sum_{i=2}^{A-1} s_i \right) + l^i(z)w(z, s_1)
\]
Case 2: Idiosyncratic shocks

- Need to approximate intragenerational wealth distribution
- As above, we keep track of aggregate capital and describe the wealth distribution across individuals by their financial wealth.
- We allow for $N$ different wealth levels for each generation $a=2,\ldots,A-1$.
- For each $a=2,\ldots,A-1$, $\omega_{ai}$ denotes the average capital holding of all agents of age $a$ that are richer than the $(i-1)/N$ percentile of the wealth distribution yet poorer than the $i/N$ percentile.
- This choice of representing the wealth distribution only works well in a model without borrowing constraints. Other methods should be employed if it can be expected that a large fraction of the population holds zero wealth.
Case 2: Equations

\[
\bar{c}(z, s, \theta)\rho^{-1} + \beta_a \left[ \sum_{z'} \pi(z' | z) \sum_{y'} \eta(y')(v_{a+1}(z', s', \theta'(y', z')))^{\sigma \rho^{-1}} \right]^\frac{\rho}{\sigma} - 1
\]

\[
\sum_{z'} \pi(z' | z) \sum_{y'} \eta(y')(1 + r(z', s'))v_{a+1}(z', s', \theta'(y', z')))^{\sigma \rho^{-1}} \bar{c}_{a+1}(z', s', \theta'(y', z')))^{\rho-1} = 0,
\]

\[
v_a(z, s, \theta) = \bar{c}_a(z, s, \theta)^\rho + \beta_a \left[ \sum_{z'} \pi(z' | z) \sum_{y'} \eta(y')(v_{a+1}(z', s', \theta'(y', z')))^{\sigma \rho} \right]^{\frac{\rho}{\sigma}},
\]

with \( \theta'(y', z') = k_a(z, s, \theta)(1 + r(z', s')) + l_{a+1}(y', z')w(z', s') \)

\[
s' = \left( \sum_{i=1}^{A-1} \sum_{n=1}^{N} \frac{1}{N} \sum_{y \in Y} \eta(y)k_i(z, s, \omega_n + w(z, s)l^i(y, z)), (\omega'_{an})_{a=2...A-1, n=1...N} \right) \in \bar{\mathcal{B}},
\]
Solving the collocation equations

- In both cases, we approximate the unknown functions using adaptive sparse grids
- We solve for the unknown coefficients using a time-iteration scheme:
  - Given a grid, G, and a policy for next period, p', solve equilibrium equations at all points in the grid and for all shocks
  - Brumm and Scheidegger (2015): Construct grid within the time iteration step...
1. Start with a coarse grid $G^z_{\text{old}} \subset S$ (a “classical” sparse grid of a “low” level $L_0$), and generate $G^z$ by adding for each $x \in G^z_{\text{old}}$ all $2d$ neighboring points. Choose a refinement threshold $\epsilon$ and a maximal level $L_{\text{max}} > L_0$ and set $l = 1$.

2. For each each grid point

$$s \in \begin{cases} G^z \text{ if } l = 1 \\ G^z \setminus G^z_{\text{old}} \text{ if } l > 1 \end{cases}$$

solve for the optimal policies

$$p(z, s) = (k_1(z, s), \ldots, k_{A-1}(z, s), v_1(z, s), \ldots, v_{A-1}(z, s))$$

3. For each $z$, generate $G^z_{\text{new}}$ from $G^z$ by adding for each $x \in G^z \setminus G^z_{\text{old}}$ its $2d$ neighboring points if hierarchical surplus above $\epsilon$

4. Set $G^z_{\text{old}} = G^z$, $G^z = G^z_{\text{new}}$.

5. If $G^z = G^z_{\text{old}}$ or $L_0 + l = L_{\text{max}}$, then and go to 6., otherwise set $l = l + 1$ and go to 2..

6. Define the new policy function $p'(z, \cdot)$ as the (adaptive sparse grid) interpolation of $\{p(z, s)\}_{s \in G^z}$. 

### Numbers

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A )</td>
<td>60</td>
</tr>
<tr>
<td>( \beta )</td>
<td>0.96</td>
</tr>
<tr>
<td>( 1/(1-\rho) )</td>
<td>0.5</td>
</tr>
<tr>
<td>((1-\sigma))</td>
<td>3</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>0.3</td>
</tr>
<tr>
<td>( \delta )</td>
<td>(0.08, 0.08, 0.38, 0.38)</td>
</tr>
<tr>
<td>( \xi )</td>
<td>(1.03, 0.97, 1.03, 0.97)</td>
</tr>
</tbody>
</table>
## Probabilities

<table>
<thead>
<tr>
<th></th>
<th>z'=1</th>
<th>z'=2</th>
<th>z'=3</th>
<th>z'=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>z=1</td>
<td>0.9x0.98=0.882</td>
<td>0.1x0.98=0.098</td>
<td>0.9x0.02=0.018</td>
<td>0.1x0.02=0.002</td>
</tr>
<tr>
<td>z=2</td>
<td>0.1x0.98=0.098</td>
<td>0.9x0.98=0.882</td>
<td>0.1x0.02=0.002</td>
<td>0.9x0.002=0.018</td>
</tr>
<tr>
<td>z=3</td>
<td>0.49</td>
<td>0.49</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>z=4</td>
<td>0.49</td>
<td>0.49</td>
<td>0.01</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Labor profile

No idiosyncratic shocks
Euler errors and #points
Euler errors and node hours

We use 256 (out of 5272) nodes!
Disaster shocks and the wealth distribution

- Disaster shocks have large effect on aggregate consumption and prices.
- Since old hold a lot of capital one would also expect large effects on wealth distribution.
- Not as large as one would think…
Aggregate Consumption
Wealth distribution
Wealth distribution

![Graph showing the wealth distribution over age.](image)
Wealth distribution

![Graph showing wealth distribution with age on the x-axis and capital share on the y-axis. The graph has two lines: one for Timestep: 4; State: 2 and another for Timestep: 5; State: 2. The curve peaks around age 60.]
Wealth distribution
Wealth distribution