Computational methods

Quantitative Macroeconomic Methods I

Kieran Larkin
kieran.larkin@iies.su.se

Institute for International Economic Studies

September 2023

Outline

1. Solving the Huggett model
   - Setting out the model
   - Solving the model on the computer
2. Transition paths
   - Shocks to the stationary equilibrium
   - Calculating welfare
3. Endogenous grid method
4. Tools for computational models

Solving the Huggett model

Why are we interested in heterogeneous agent problems?

In the data we see large variation in income and wealth

- We want to be able to study the source of this variation
- Understand how it matters for welfare and aggregate outcomes

Key mechanism that the model introduces is *precautionary savings*

- Households want to insure against idiosyncratic risk
- Increases the demand for assets which lowers equilibrium interest rate
- Introduces curvature into the consumption function
- Changes/increases the Marginal Propensity to Consume

Solution to the Huggett model

How to solve the Huggett model on the computer

- Today we are going to go through how to solve the canonical heterogeneous agent model on the computer
- Model is the core of modern macro analysis $\rightarrow$ important skill set
- Going to look at code in Matlab
  - Many other options available: Python, Julia, C++, Fortran...
  - Due to its speed Fortran still a good option if you plan to work in this area
- Learning to solve the model also very useful for thinking about the properties of the model itself

Recap of the environment

Classic consumption-savings problem

Asset market

- Households can borrow and save in a one period bond a
- Interest rate on borrowing and saving is r
- Households can borrow up to the exogenous borrowing constraint

$$a' \geq \underline{a}$$

- Borrowing constraint is tighter than the natural borrowing constraint
- Bonds provided in zero net supply $\overline{A} = 0$
- *General equilibrium:* the interest rate determined by asset market clearing

Recap of the environment

Classic consumption-savings problem

Income process

- Each period the household received an endowment $y(s)$ where s is the income state
- The state s follows an S-state Markov process.
- The transition matrix for the Markov process is given by $P(s, s') = Prob[s_{t+1} = s' | fs_t = s]$

State space

- the state space is defined over assets and income: (a, s)

Preferences

- Household preferences are given by:

$$\mathbf{E_0} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

- Where $\beta \in (0, 1)$ is the discount factor and c_t is the consumption of non-durables
- the felicity utility function is CRRA:

$$u(c_t) = \frac{c_t^{1-\gamma}}{(1-\gamma)}$$

- Where $\gamma > 1$ is the inverse of the *Intertemporal Elasticity of Substitution*
- The budget constraint is:

$$c_t + a_{t+1} = a_t(1 + r_t) + y_t$$

Household maximization: Bellman equation

Household maximization problem can be written in the recursive form:

$$
\begin{aligned}
V(a, s) &= \max_{c, a'} \; u(c) + \beta \sum_{s'=1}^{S} P(s, s') V(a', s') \\
c + a' &= a(1 + r) + y(s) \\
a' &\geq \underline{a}
\end{aligned}
$$

- The solution to the household problem is the household value function $V(a, s)$ and policy functions for:
  - consumption $c(a, s)$ and
  - assets $a_+(a, s)$

Law of motion and stationary distribution

- Define the unconditional distribution of (a_t, s_t) pairs: $\lambda_t(a, s) = Prob[a_t = a \cap s_t = s]$
- Together the Markov chain for the income process P and optimal policy a_+ induce a law of motion for the distribution:

$$\lambda_{t+1}(a', s') = \sum_s \sum_a \lambda_t(a, s) P(s, s') \cdot \mathbf{1}[a' = a_+(a, s)] \tag{1}$$

- The time-invariant distribution λ that solves equation (1) is called the **stationary distribution**

Stationary distribution interpretation

- The combination of P and $a_+(a,s)$ on the household state vector induces a Markov chain on the households state vector:

$$\mathcal{P}(a,s,a',s') = P(s,s') \cdot \mathbf{1}[a' = a_+(a,s)] \tag{2}$$

- Suppose the Markov chain $\mathcal{P}$ is asymptotically stationary and has a unique invariant distribution
- Typically all states will be recurrent and visited by the household occasionally
- So the distribution tells us the fraction of time that a household spends in each state
- *Alternatively* think of (a,s) as the state of a particular household at a particular time
- Then we can think of $\lambda(a,s)$ as the distribution over all agents over the state variables even as the individual household moves across states over time

Equilibrium definition

A **stationary equilibrium** is an interest rate r, a policy function $a_+(a, s)$, and a stationary distribution $\lambda(a, s)$ such that:

- The policy function $a_+(a, s)$ solves the household's optimum problem
- The stationary distribution $\lambda(a, s)$ is induced by P and $a_+(a, s)$
- The bond market clears $\sum_{a,s} \lambda(a, s)a_+(a, s) = \overline{A}$

Road map for solving model

1. Guess an interest rate r
2. Solve **optimal policies** from the household maximization problem
3. Solve for the ergodic **distribution** over the state space
4. Check market clearing
5. **Update** guess for interest rate
6. Repeat until market clearing satisfied

Solving the HH problem by Value Function Iteration

How do we find $V(a, s)$?

- We are going to take advantage of the *Contraction Mapping Theorem*
- If our problem satisfies the *Blackwell sufficiency conditions* we can show there is a unique solution V to household problem
  - See Stokey, Lucas and Prescott (1989) for details
- From any initial guess $V^0(a, s)$ if we iterate on the value function we will eventually converge to the true solution

Solving the HH problem by Value Function Iteration

Pre-loop

- Start by creating grids over assets $\mathcal{A} = [\underline{a}, a_2, ..., a_N]$ and income $S = [s_1, ..., s_S]$
- At each point on the state space (a_i, s_j) for a given choice of assets (a_k) calculate consumption:

$$c(a_i, s_j, a_k) = a_i(1 + r) + y(s_j) - a_k \tag{3}$$

- If $c(a_i, s_j, a_k) > 0$ find utility $u(c(a_i, s_j, a_k))$ else assign a large negative value

Solving the HH problem by Value Function Iteration

Within-loop

- From the initial guess $V^0(a,s)$ construct the expected value tomorrow:

$$\hat{V}(a,s) = \mathbf{E}V^0(a,s) = \sum_{s'} P(s,s')V^0(a,s')$$

- The value today is given as the solution to maximizing over a_l:

$$V(a_i,s_j) = \max_{a_k \in \mathcal{A}} \left\{ u(c(a_i,s_j,a_k)) + \beta \hat{V}(a_k,s_j) \right\}$$

- Then we check for convergence: $||V(a,s) - V^0(a,s)||_\infty < \epsilon$
- If the criteria is satisfied we have found the solution.
- Otherwise we replace the initial guess $V^0(a,s) = V(a,s)$ and repeat

Solving the stationary distribution

- Having solved the household problem we have a solution to $a_+(a, s)$ we now want to find $\lambda(a, s)$
- There are two options for this:

1. Iterate until convergence
2. Find the eigenvector associated with the unit eigenvalue

Solving the stationary distribution: by iteration

- Start with an initial guess $\lambda_0(a, s)$
  - Note: this is a probability measure so $\sum_{a,s} \lambda_0(a, s) = 1$
- Update the distribution:

$$\lambda(a_k, s_l) = \sum_{s_j} \sum_{a_i} \lambda_0(a_i, s_j) P(s_j, s_l) \cdot \mathbf{1}[a_k = a_+(a_i, s_j)]$$

- Then we check for convergence: $||\lambda(a, s) - \lambda_0(a, s)||_\infty < \epsilon$
- If the criteria is satisfied we have found the solution.
- Otherwise we replace the initial guess $\lambda_0(a, s) = \lambda(a, s)$ and repeat

Solving the stationary distribution: find the eigenvector

- Remember the Markov chain $\mathcal{P}(a, s, a', s')$ is just a transition matrix of dimensions $(NxS)x(NxS)$
- So if we construct $\mathcal{P}$ we can just apply the matrix operation:

$$\lambda^{\mathbf{T}} = \lambda_{\mathbf{0}}{}^{\mathbf{T}}\mathcal{P}$$

- This also suggests an alternate solution method. Rearranging and imposing a stationary solution:

$$(I - \mathcal{P}^{\mathbf{T}})\lambda = 0$$

- Where I is the identity matrix.
- This might remind you of solving eigenvalues & eigenvectors
- We can also find the stationary distribution as the *eigenvector* (appropriately normalized) associated with the unit eigenvalue
- The properties of $\mathcal{P}$ ($\mathcal{P}_{i,j} > 0$ and $\sum_j \mathcal{P}_{i,j} = 1$) mean at least one eigenvector exists and is unique if some regularity conditions are satisfied

Updating the guess for the interest rate

- We now check market clearing. If this this is satisfied we have found solution

$$A_{demand} = \sum_{a,s} a_{+}(a,s)\lambda(a,s) = \overline{A}$$

- If $A_{demand} > \overline{A}$ the interest rate is too high
  - Update with a lower guess for r
- If $A_{demand} < \overline{A}$ the interest rate is too low.
  - Update with a higher guess for r

Update interest rate by bi-section

- We can find the interest rate more efficiently by using a bi-section method
- The idea is to find an interval where the sign of a function changes sign
- Within this interval there must be a root of a continuous function $f(x)$
- Method:
  1. First check bounds $[r_{min}, r_{max}]$ contain a root
  2. Next try $\hat{r} = \frac{r_{min}+r_{max}}{2}$
     - If there is excess demand $A_{demand} > \overline{A}$ market clearing r must be less than $\hat{r}$. Set $r_{max} = \hat{r}$
     - Else set $r_{min} = \hat{r}$
  3. Set new guess $\hat{r} = \frac{r_{min}+r_{max}}{2}$
  4. Repeat until market clearing satisfied

Calibration

- Following the calibration of Huggett (93):
- Model period is two months
- Two state process for income with $y(s_1) = 0.1$ and $y(s_2) = 1$
- Persistence of income is: $Prob(s_t = s_2, s_{t+1} = s_2) = 0.925$ and $Prob(s_t = s_1, s_{t+1} = s_2) = 0.5$
  - Loosely matches some features of the data. Can think about $y(s_1)$ being unemployment
- Set $\beta = 0.99$. Implies annual $\beta^{ann} = 0.96$
- Coefficient of Relative Risk Aversion $\gamma = 1.5$
- And set borrowing constraint $\underline{a} = -4$
- Choose 1,000 grid points for asset dimension. with $\overline{a} = 10$

Policy functions: asset choice

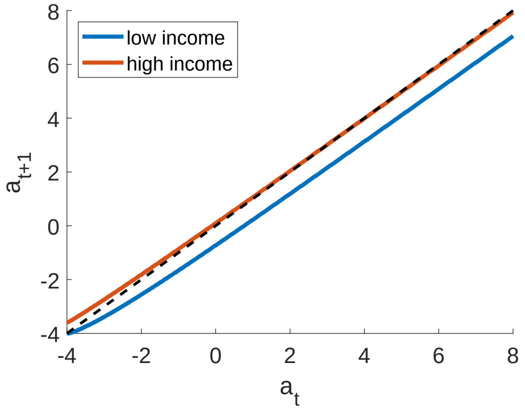


Figure: asset choice

Policy functions: consumption

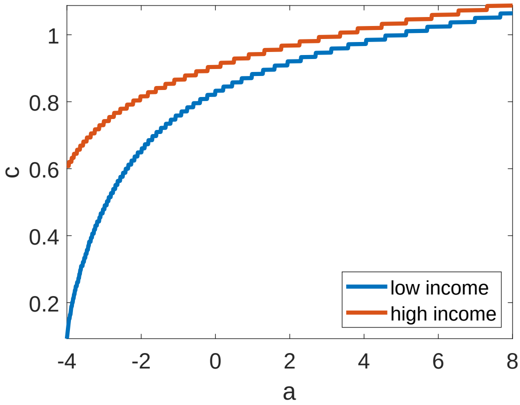


Figure: consumption function

Value function

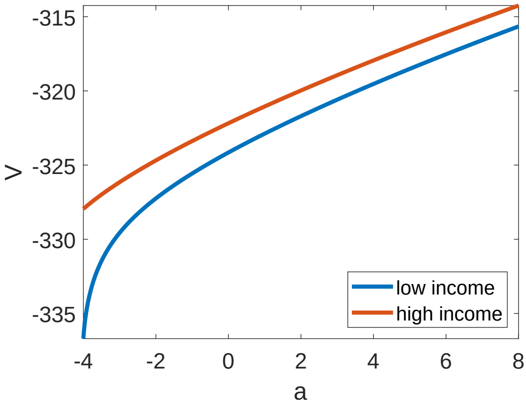


Figure: value function

Distribution

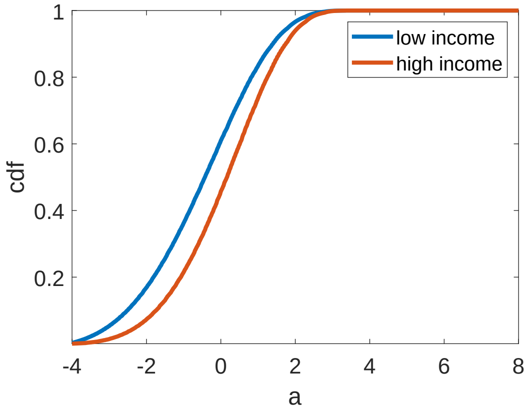


Figure: distribution over assets

Transition paths

Transitions

- Up to now we have focused on stationary equilibria in incomplete markets models
- However, often we are interested in dynamics *(more next course)* or the effect of policies
- Consider a new policy. What happens to aggregate variables and welfare?
- We can look at these issues by comparing different stationary equilibrium - why might this be wrong?
- If prices adjust (e.g. the interest rate) the existing holder of assets may gain or lose out from a policy. This is not reflected in the new long run allocation
- So we should take into account the transition

Transition in the Huggett economy

- Consider a relaxation in financial regulations that allows the borrowing constraint to relax from $\underline{a}_{pre} = -4$ to $\underline{a}_{post} = -6$ over 25 periods
- We think about this a change a surprise to households in period 1. But once the policy change is realized prices are deterministic
- How would the economy adjust?
- How would welfare be affected?
- A nice example of the transition we look at in a more realistic setting is Mendoza, Quadrini & Rios-Rull (09)
  - Role of financial development in global financial imbalances (multiple countries, capital)
  - Attribute the net negative financial account position of US to financial integration. Countries with looser borrowing constraints borrow from abroad

Equilibrium definition

Note the objects are now defined over time t

- Given the initial wealth distribution $\lambda(a,s)$ an equilibrium is given by *sequences* of
  - policy functions $\{a_{+,t}(a,s), c_t(a,s)\}_{t=0}^{\infty}$
  - prices $\{r_t\}_{t=0}^{\infty}$
  - distributions $\{\lambda_t(a,s)\}_{t=0}^{\infty}$
- such that:
  - the policy rules solve the households optimization problem
  - the asset market clears in all periods

$$\sum_{a,s} a_{+,t}(a,s)\lambda_t(a,s) = \overline{A}$$

  - the distributions are consistent with the initial distribution and prices

Transition equilibrium: algorithm

- Solve for the steady state under both policies: $\{\underline{a}_{pre}, \underline{a}_{post}\}$
- Choose a large number of transition periods T
- Guess a sequence of interest rates $\{r_t\}_{t=1}^T$ where $r_T = r_{post}$
- Given these prices solve *back* the policy rules for $t = T-1, ..., 1$
- Starting at the initial distribution $\lambda_0(a,s)$ and using the policy rules $\{a_{+,t}(a,s), c_t(a,s)\}_{t=1}^T$ solve for the distributions $\{\lambda_t(a,s)\}_{t=1}^T$
- Compute excess demand for assets in each period
- Update sequence of interest rates $\{r_t\}_{t=1}^T$
- Repeat until convergence

Transition in the Huggett economy

Relaxation in the borrowing constraint from $\overline{a} = -4$ to $\overline{a} = -6$

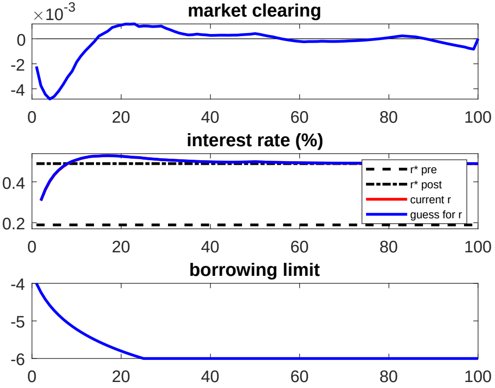


Figure: transition path

Calculating welfare

- A common question we are interested in is: "How does a policy/environment change affect the welfare of households?"
- To do this we compare the value functions and distributions of two economies
- For example, the average welfare in economy 1 vs economy 2:

$$\sum_{a,s} \lambda^{pre}(a,s) V^{pre}(a,s) \text{ vs. } \sum_{a,s} \lambda^{post}(a,s) V^{post}(a,s)$$

- However, that misses the effect of the reallocation of assets
- As we have computed the transition we can now compare

$$\sum_{a,s} \lambda^{pre}(a,s) V^{pre}(a,s) \text{ vs. } \sum_{a,s} \lambda^{pre}(a,s) V^{1}(a,s)$$

- Where $V_1(a,s)$ is the value in the first period of the transition

Calculating welfare: in consumption units

- Comparing values is a bit difficult to interpret
- It is often useful to express welfare in consumption units gained
- A widely used measure is the % increase in per-period consumption that would make a household indifferent
  - First used by Lucas (1987) studying cost of business cycles

$$\mathbf{E_0} \sum_{t=0}^{\infty} \beta^t u(c_t^1) = \mathbf{E_0} \sum_{t=0}^{\infty} \beta^t u((1+\alpha)c_t^2)$$

- With homothetic utility, α can often be factored out and written in terms of $V(a,s)$ and $\lambda(a,s)$

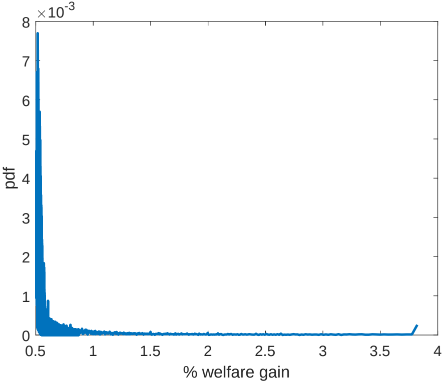
$$\alpha = \left(\frac{\sum_{a,s} \lambda^{pre}(a,s) V^1(a,s)}{\sum_{a,s} \lambda^{pre}(a,s) V^{pre}(a,s)} \right)^{1/(1-\gamma)} - 1$$

- Might be interested in the distribution of consumption gains $\alpha(a,s) = \left(V^1(a,s)/V^{pre}(a,s)\right)^{1/(1-\gamma)} - 1$
- Average consumption gain: $\sum_{a,s} \lambda^{pre}(a,s)\alpha(a,s)$

Calculating welfare: in Huggett economy

Gains are often quite small. Notice extra gain of interest rate increase to asset holders

	welfare gain (% consumption)
Stationary	0.50
Transition	0.54



Endogenous grid method

Endogenous grid method

- Often we need to solve the household problem many times
- This can end up taking a lot of compute time. It's important we try to make each step as efficient as possible
- In particular solving the household problem can be very computationally costly.
- One improvement that can save a lot of this is the *endogenous grid method*
- The basic idea is to use the household's first order condition to find the policy functions
- The method was first proposed by Carroll (06)

Endogenous grid method

Using the first order condition

- The key insight is that we can use the first order condition to find the solution to the household optimum problem
- Previous methods have also made of this type of approach (e.g. time iteration)
- However, in general solving for $a_+(a, s)$ is a costly non-linear problem:

$$(a(1+r) + y(s) - a_+(a, s))^{-\gamma} =$$

$$\beta(1+r)\sum_{s'=1}^{S} P(s, s')\Big(a_+(a, s)(1+r) + y(s') - a_+(a', s'))^{-\gamma}\Big)$$

Endogenous grid method

Fixing a' grid instead of a grid

- If we instead take a' as given (rather than a) and write the first order condition in terms of the derivative of the value function V_a:

$$c(a'|s) = \left(\beta(1+r)\sum_{s=1}^{S} P(s,s')V_a(a',s')\right)^{-1/\gamma}$$

- We get consumption immediately
- From the budget constraint we can recover assets or "cash on hand" $x = a(1+r) + y(s)$ from:

$$x = c(a'|s) + a'$$

- From this relationship we get the policy functions $a_+(a,s)$ and $c(a,s)$ without maximizing or solving a non-linear equation

Endogenous grid method algorithm (Carroll (06))

1. Define grid $\mathcal{A} = [\underline{a}, a_2, ..., a_N]$ over a'
   - It's useful to create a grid for cash on hand over tomorrow's asset choice and income state $\mathcal{X} = [x_1(\underline{a}, s_1), ..., x_M(a_N, s_S)]$
2. For the initial guess $V(a, s)$ compute the derivative $V_a(a', s)$ at each point of the grid (note need $V_a(a', s) > 0$)
3. Find the expectation of next period derivative: $\hat{V}_a(a', s) = \sum_{s=1}^{S} P(s, s')V_a(a', s')$
4. Using the f.o.c. find policy for $c(a'|s)$:

$$c(a'|s) = \left(\beta \hat{V}_a(a', s)\right)^{-1/\gamma}$$

5. For each choice a' compute endogenous cash on hand $x = c(a'|s) + a'$. We can redefine our policy function in terms of today's states: $c(x, s)$
   - **borrowing constraint:** the solution tells us *exactly* the level of cash on hand x where the borrowing constraint starts to bind $a' = \underline{a}$.
   - For all values of assets below this set $c(x, s) = x - \underline{a}$
   - It can be useful to add the point $c(\underline{a}, s) = 0$ to the policy to implement this

Endogenous grid method algorithm

5. Given the optimal policy, update our guess for the value function for iteration j:
$$V^j(x,s) = \frac{c(x,s)^{1-\gamma}}{(1-\gamma)} + \beta \sum_{s'=1}^{S} P(s,s')V^{j-1}(a',s')$$
6. Interpolate the consumption policy $c(x,s)$, and value function $V^j(x,s)$ onto the cash on hand grid $\mathcal{X}$
   - This also defines the asset policy $a_+(\mathcal{X},s) = \mathcal{X} - c(\mathcal{X},s)$
   - Notice as $\mathcal{X}(a,s)$ with $a \in \mathcal{A}$ we have $c(a,s)$ and $V^j(a,s)$ with $a \in \mathcal{A}$
7. Use the envelope condition to find an updated guess of $V_a^j(a',s) = (1+r)c(a',s)^{-\gamma}$
8. Check for convergence $||V^j(a',s) - V^{j-1}(a',s)||_\infty$ if reached exit
9. Otherwise repeat from **step 3**
10. After convergence you might want to use the final policies on the endogenous grid

Endogenous grid method algorithm: life cycle

The method is very similar but we don't need to check for convergence

- j is now age with T the terminal period
  - not iterations. *Count down*: $j = T, T-1, ..., 2, 1$
- Note: store policy $c_j(x_j, s_j)$ and endogenous grid X_j at every age
- Define an exogenous grid $\mathcal{A} = [\underline{a}, a_2, ..., a_N]$ over asset choice

1. Start with a set of final cash on hand points X_T and set consumption equal to cash on hand $c_T(x_T, s_T) = x_T$
   - We are assuming no bequests. If bequest motive, adjust final period to be bequest function
   - If pension income, no uncertainty/s-state during retirement
2. For the points on $\mathcal{A}$ construct a *new* set of cash on hand points $\mathcal{X}_{j+1} := x_{j+1} = a_{j+1}(1+r) + y_{j+1}(s_{j+1})$
3. Interpolate the function $c_{j+1}(x_{j+1}, s_{j+1})$ defined on X_j to $\mathcal{X}_{j+1}$
   - Now we have $c_{j+1}(a_{j+1}, s_{j+1})$ where a_{j+1} on $\mathcal{A}$

Endogenous grid method algorithm: life cycle

5. Compute $V_a^{j+1}(a_{j+1}, s_{j+1}) = (1+r)c_{j+1}(a_{j+1}, s_{j+1})^{-\gamma}$
6. Find the expectation of next period derivative: $\hat{V}_a^{j+1}(a_{j+1}, s_j) = \sum_{s_{j+1}=1}^{S} P(s_j, s_{j+1}) V_a^{j+1}(a_{j+1}, s_{j+1})$
7. Using the f.o.c. find policy for $c_j(a_{j+1}|s_j)$:

$$c_j(a_{j+1}|s_j) = \left(\beta \hat{V}_a^{j+1}(a_{j+1}, s_j)\right)^{-1/\gamma}$$

8. For each choice a_{j+1} compute endogenous cash on hand $x_j = c_j(a_{j+1}|s_j) + a_{j+1}$. We can redefine our policy function in term's of today's states: $c_j(x_j, s_j)$.
9. Store policy $c_j(x_j, s_j)$ and endogenous grid X_j
   - It can be useful to add the point $c_j(\underline{a}, s_j) = 0$ to the policy for constrained choices (see above)
10. Given the optimal policy construct value function for age j: $V^j(a_j, s_j)$, but this is not needed for solution
11. Move back one age period $j-1$ and return to **step 2.**. Repeat until $j=1$

Solving the distribution when policy function is off grid

- When we solved with value function iteration both the state a and choice a' were on the same grid
- This is now no longer the case
  - This isn't just restricted to EGM it can also be true when we solve with VFI off grid
- We can still compute the distribution $\lambda(a, s)$ but when computing the transition matrix we need to interpolate households choices

Solving the distribution when policy function is off grid

- If $\hat{a} = a_+(a,s)$ such that $\{a_k < \hat{a} < a_{k+1}\} \subset \mathcal{A}$. Now:

$$\lambda_{t+1}(a_k, s') = \sum_{a,s} \lambda_t(a,s)P(s,s') \cdot \alpha \mathbf{1}[\hat{a} = a_+(a,s)]$$

$$\lambda_{t+1}(a_{k+1}, s') = \sum_{a,s} \lambda_t(a,s)P(s,s') \cdot (1-\alpha) \mathbf{1}[\hat{a} = a_+(a,s)]$$

- where α is given by:

$$\alpha = \frac{a_{k+1} - \hat{a}}{a_{k+1} - a_k}$$

- Alternatively, we can run a Monte Carlo simulation of a panel of households and allow their asset choice to be continuous
  - Need to ensure there are enough households to ensure stochastic equicontinuity
  - Practical point: important to fix shocks/reset random number generator seed if calibrating the model or solving for a price

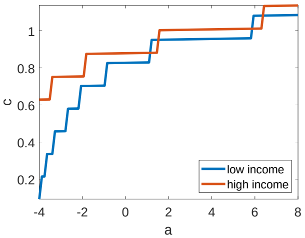
Tools for computational models

Checking accuracy of the solution: size of the grid

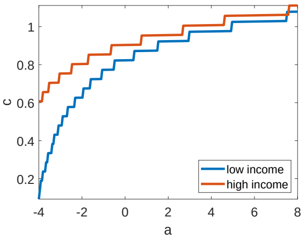
- Its important to make sure our solution is sufficiently accurate
- We need to check the largest asset grid point does not restrict household choice
  - Check the maximum grid point exceeds where a household would like to increase assets i.e. $a_+(a, s) > a$
  - Run a simulation to make sure no households exceed the maximum grid point (or sufficiently small proportion)

Checking the number of grid points

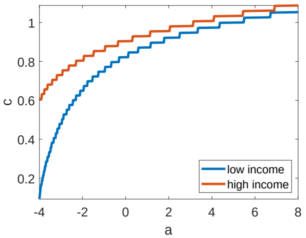
- We also need to check whether we are using enough grid points



(a) $N = 100$



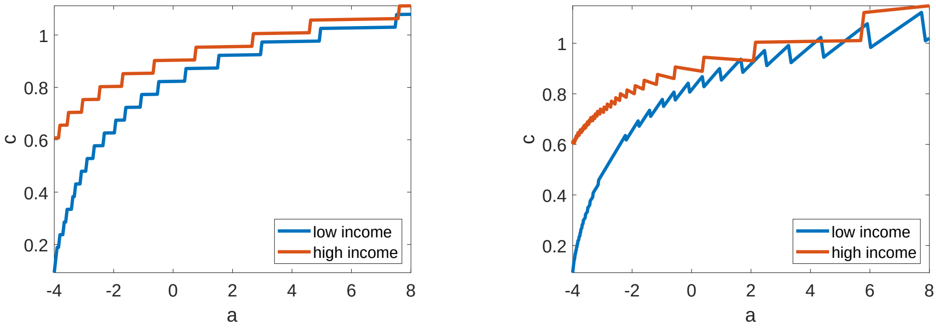
(b) $N = 250$



(c) $N = 500$

Non-uniform distribution of grid points

- One solution is to concentrate grid points in region of curvature
- Let $a_1 = 0$ and $a_n = \log(\overline{a} - \underline{a} + 1)$
- Distributed grid points uniformly between (a_1, a_n)
- Transform grid $\mathcal{A} = \exp(\{a_1, ..., a_n\}) - 1 + \underline{a}$



(a) linear (b) log

Checking accuracy of the solution: Euler errors

- One test for accuracy is to check the Euler errors
- Away from the borrowing constraint the EE should hold exactly

1. Create fine asset grid $a^f \in \mathcal{A}^f$ (e.g. including "off grid" points)
2. Evaluate the Euler equation error as:

$$\epsilon^{EE} = c(a^f, s)^{-\gamma} - \beta \sum_{s'} P(s, s')(1 + r)c(a_+(a^f, s), s)^{-\gamma}$$

3. Compute an accuracy criterion such as *Root Mean Squared Error* or maximum error

- It can be useful to write the errors in units of consumption:

$$\epsilon^{EE_2} = 1 - c(a^f, s)/\left(\beta \sum_{s'} P(s, s')(1 + r)c(a_+(a^f, s), s)^{-\gamma}\right)^{-1/\gamma}$$

Approximating income processes

We often want more realistic income processes than ($S = 2$) Huggett approximation

- The AR(1) process for $Y_t = \exp(y_t)$ is a popular choice in numerical applications:

$$y_t = \rho y_{t-1} + \epsilon_t$$

- Where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Note $\sigma_y^2 = \sigma_\epsilon^2 / (1 - \rho^2)$
- This is a continuous process (difficult for the computer)
- For numerical solutions we need to discretize it as a finite state Markov Chain
- This requires selecting a set of nodes and finding the transition matrix

Approximating income processes

Tauchen (86)

- Tauchen (86) provides a commonly used (and simple) method of discretization
- Create a set of equispaced nodes $\mathcal{Y} = \{y_1, ..., y_S\}$ with the max and (min) given by:

$$y_S = m\left(\frac{\sigma_\epsilon}{1-\rho^2}\right)^{1/2}$$

- Requires choosing m e.g. 3
- Let $d = y_{k+1} - y_k$. For $1 < k < N$ set transition probabilities:

$$\pi_{jk} = F\left(\frac{y_k + d/2 - \rho y_j}{\sqrt{\sigma_e}}\right) - F\left(\frac{y_k - d/2 - \rho y_j}{\sqrt{\sigma_e}}\right)$$

- And for the end points:

$$\pi_{j1} = F\left(\frac{y_1 + d/2 - \rho y_j}{\sqrt{\sigma_e}}\right)$$

- As $n \to \infty$ approximation gets better

Gauss-Hermite approximation of income (Tauchen-Hussey, 91)

- Alternative method of discretization based on *numerical integration* techniques
  - Do better by more efficient placement of points in $\mathcal{Y}$
  - Precise polynomial approximation possible with small number of nodes
- Create a set of nodes $\mathcal{Y} = \{y_1, ..., y_S\}$ determined by $y_i = \sqrt{2}\sigma x_i$
  - x_i are Gauss-Hermite nodes. ϕ_j are GH weights $\rightarrow$ same for every problem
  - TH normalization so grid in y_i same for every y_j
- The elements of transition matrix P are:

$$p_{i,j} = \frac{f(y_j|y_i)}{f(y_j|0)} \frac{w_j}{s_i}$$

- With $w_j = \phi_j/\sqrt{\pi}$, $f(\cdot|y_i)$ the density function for $\mathcal{N}(\rho, \sigma_\epsilon^2)$ and

$$s_i = \sum_{s=1}^{S} \frac{f(y_s|y_i)}{f(y_j|0)} w_j$$

- Still need to choose domain. In Tauchen-Hussey they advocate setting $\sigma = \sigma_\epsilon$
  - Floden (08) instead suggests $\sigma = \omega\sigma_\epsilon + (1-\omega)\sigma_y$ with $\omega = 0.5 + 0.25\rho$
  - Puts more weight on the unconditional variance as the persistence increases

Recursive approximation of income process

Rouwenhorst (95)

- Provides a method over a symmetric evenly spaced grid and *recursively* defined transition matrix
- Let $\mathcal{Y} = \{\mu_y - \nu, ..., \mu_y + \nu\}$
- For p and q with can define the n state grid recursively:

$$P_n = p\begin{bmatrix} P_{n-1} & \mathbf{0} \\ \mathbf{0}' & 0 \end{bmatrix} + (1-p)\begin{bmatrix} \mathbf{0} & P_{n-1} \\ 0 & \mathbf{0}' \end{bmatrix} + (1-q)\begin{bmatrix} \mathbf{0}' & 0 \\ P_{n-1} & \mathbf{0} \end{bmatrix} + q\begin{bmatrix} 0 & \mathbf{0}' \\ \mathbf{0} & P_{n-1} \end{bmatrix}$$

- For Markov chain:
  - First order serial correlation of process= $p + q - 1$
  - Variance=$\nu^2/(n-1)$
  - Choose to match properties of continuous process
- Turns out the Rouwenhorst method is the <u>most accurate</u> and robust for highly persistent processes e.g. $\rho \to 1$
  - See Kopecky & Suen (10)
- **Programs can be downloaded for all methods in Matlab**

Accelerator method

- Value function iteration is still sometimes the best/easiest solution method but it is slow
- One simple trick is the *accelerator* method
- The computationally costly part of VFI is finding the maxim but between iterations this doesn't change significantly
- We don't need to update the policies every period

1. Every 10 periods solve households problem as in standard VFI
2. In between update the value using the last policy solved for:

$$V(a,s) = u(c^*(a,s)) + \beta \sum_{s'} P(s,s') V^{j-1}(a_+^*(a,s), s')$$

3. Proceed until convergence of the value function. Update the policy in the last iteration

Policy Iteration

Also known as Howard's Improvement algorithm

- For infinite horizon problems we can speed up the solution by making greater use of the time invariance of the policy function
  1. Make an initial guess for the policy function
     - ★ Or make an initial guess for the value function and solve for the first iteration of policies $c_j(a, s)$ and $a_{+,j}(a, s)$
  2. Construct Markov matrix over probabilities and policies $\mathcal{P}^j$ and vector of utilities over states $\mathbf{U}^j$
  3. Update value function (matrix notation):

$$\mathbf{V}^{j+1} = (I - \beta \mathcal{P}^j)^{-1} \mathbf{U}^j$$

  4. Continue until convergence $||\mathbf{V}^{j+1} - \mathbf{V}^j||_\infty$
- Quicker because the algorithm incorporates that the new policy is used forever not just in one period
  - Notice we are solving: $\mathbf{V} = \mathbf{U} + \beta \mathcal{P} \mathbf{V}$
- Can run into difficulties inverting matrix as state space gets large

Off grid value function iteration

- So far when solving by VFI we've focused on making choices on the grid $\mathcal{A}$, but this is not required
- We can always solve:

$$V(a_i, s_j) = \max_{\hat{a}} \left\{ u(a_i(1+r) + y(s_j) - \hat{a}) + \beta \sum_{s'} P(s_j, s') V(\hat{a}, s') \right\}$$

- To do this we need a minimization routine
  - For example: Nelder-Mead (simplex search). `fminsearch` in Matlab. If we can compute $V_a(\hat{a}, s')$ efficiently can use Newton-method
- And to interpolate over future values of $V(a', s)$
- Such methods are still usually slower than EGM

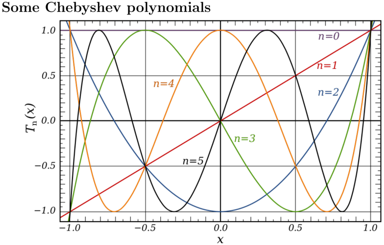
Polynomial approximation of value function

- We can reduce the number of grid points to evaluate by using a polynomial $C(V, \theta)$ to approximate the value function V
  - θ is a set of parameter coefficients to be solved
- The polynomial evaluated at $n + 1$ nodes can require far fewer grid points in the asset dimension
  - We can choose these nodes efficiently to provide a good approximation e.g. like numerical integration
- We then evaluate choices off the nodes by evaluating the polynomial
- The polynomial may also provide us with derivatives of the value function $C_a(V, \theta)$
- Need to be a bit careful that approximation doesn't doesn't behave badly
  - e.g. not all methods preserve concavity

Chebychev polynomials

Popular choice for approximation

- Consider approximation of function $f(x)$: $\hat{f}(x) = \sum_{p=0}^{N} \kappa_p T_p(x)$
- $T_p(x)$ is the *basis functions* of polynomial order p and κ_p are weights
- Choose **Chebychev** polynomials with $T_p(x) = \cos(p \arccos(x))$



- Have good *orthogonality* property $\rightarrow$ approximate arbitrarily well continuous function
- And can be recursively defined: $T_{i+1} = 2xT(x) - T_{i-1}(x), \; x \in [-1, 1]$

Chebychev polynomials

Popular choice for approximation

- Defining coefficients κ_p:

$$\kappa_p = \frac{\sum_{k=1}^{M} f(x_k) T_p(x_k)}{\sum_{k=1}^{M} T_p(x_k)^2}$$

- Can show this approximation exact for x_k equal to every 0 of $T_N(x)$
- And also this is OLS estimator!

$$\min_{\kappa_p} \left\{ \sum_{k=1}^{M} \left[f(x_k) - \sum_{p=0}^{N} \kappa_p T_p(x_k) \right]^2 \right\}$$

- As $N \to \infty$ get closer to true function
- But for small N good approximation for continuous functions
- *Smooth spreading* out of error important property of optimal approximation

Chebychev polynomials

Brief outline of method

1. Compute Chebyshev nodes: $x_k = -\cos\left(\frac{2k-1}{2M}\pi\right),\ k = 1, ..., M$
2. These are zeros of a polynomial order M
3. Mapping to asset space: $a_k = -a_{min} + (x_k + 1)\left(\frac{a_{max}+a_{min}}{2}\right)$
4. Guess initial function $C(V, \kappa)$
5. Solve Bellman equation using off grid polynomial approximation
   - or compute policy function non-linear solution to f.o.c.
6. Find new Chebyshev coefficients κ_p for updated function of order $N < M$
7. Check convergence of Chebyshev coefficients: $\min_p |\kappa_p^n - \kappa_p^{n-1}| < \epsilon$

For lots more detail on numerical approximation methods see Judd (98)

Golden section search algorithm

Method for single variable maximization/minimization

- Method for finding an extremum on a bounded interval. Will find a **local** minimum
  - Regardless of points evaluated so far a minimum always lies between the *two* smallest values found so far
  - Nice property that the interval widths follow golden ratio $\varphi = \frac{1+\sqrt{5}}{2}$ so maximally efficient
  - Interval shrinks by a constant proportion each period

1. Pick bounds $[x_1, x_2]$ evaluate function at bounds: $f(x_1)$ and $f(x_2)$
2. Evaluate two new points $x_3 = x_2 - (x_2 - x_1)/\varphi$ and $x_4 = x_1 + (x_2 - x_1)/\varphi$. Evaluate function at both points
   - If $f(x_3) < f(x_4)$. Update upper bound $x_2 = x_4$
   - Else update lower bound $x_1 = x_3$
3. Repeat until converge
   - But notice we've *already evaluated* one of the internal points for the next iteration

Golden section search

Graphical explanation

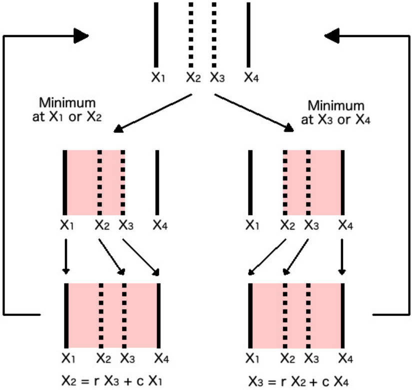


Figure: Golden section search

- See more details at Wikipedia

Curse of dimensionality

- Many problems that we wish to study are limited by the "*curse of dimensionality*"
- As we expand the state-space we exponentially increase the number of individual states we need to evaluate
- This is even worse when we add new states that are also choice variable e.g. durables, housing or illiquid assets
- Some solutions are:
  - Try to reduce the dimension of a problem (see permanent income trick next week)
  - Use efficient solution methods like EGM
  - Many papers use a small number of grid points in certain dimension e.g. two state income process
  - Use well chosen grid points and polynomial approximations
  - Parallelize the solution method

Reading list

- Ljungvist, L. & Sargent T. (2004) *Recursive Macroeconomic Theory*
- Huggett, M (1993) "The Risk Free Rate in Heterogenous-Agent, Incomplete-Insurance Economies," *Journal of Economic Dynamics and Control*
- Aiyagari, R. (1994) "Uninsured Idiosyncratic Risk and Aggregate Saving," *Quarterly Journal of Economics*
- Guvenen, F. (2011) "Macroeconomics with Heterogeneity: A practical guide," *NBER Working Paper*
- Carroll, C. (2006) "The method of endogenous gridpoints for solving dynamic stochastic optimization problems," *economic letters*
- Judd, K. (1998) *Numerical Methods in Economics*