Numerical Methods
Lecture 2: Dynamic Programming

Zachary R. Stangebye

University of Notre Dame

Fall 2017

Deterministic Systems

- Multi-period models are workhorse of macroeconomics
  - Agents make decisions each period, rationally forecasting behavior in future periods
- Example: Multi-Period Consumption-Saving Model

$$\max_{\{c_t\}_{t=0}^{T} \geq 0, \{b_{t+1}\}_{t=0}^{T-1}} \sum_{t=0}^{T} \beta^t u(c_t)$$

$$s.t. \quad c_t + b_{t+1} = y_t + (1+r)b_t \quad \text{for } t < T$$

$$c_T = y_T + (1+r)b_T$$

- Note b_0 and $\{y_t\}_{t=0}^{T}$ are given
- Cannot save or borrow in last period

Solution Method: Backward Induction

1. Find optimal behavior in last period
   - Note this will be a *function* of b_T
2. Find behavior in penultimate period, *using* optimal behavior in next period
   - Again, function of b_{T-1}
3. Find behavior in period before that ...
4. Continue until period 0

Value Functions

- Helpful tool is to define **Value Functions** along the way
- For any $t < T$, we can define it as follows:

$$V_t(b_t) = \max_{\{c_s\}_{s=t}^{T}, \{b_{s+1}\}_{s=t}^{T-1}} \sum_{s=t}^{T} \beta^{s-t} u(c_s)$$

$$s.t. \;\; c_s + b_{s+1} = y_s + (1+r)b_s \;\; \text{for } s < T$$

$$c_T = y_T + (1+r)b_T$$

- Note that $V_0(b_0)$ delivers utility from our original problem
- If $\{c_s^\star(b_t)\}_{s=t}^{T}, \{b_{s+1}^\star(b_t)\}_{s=t}^{T-1}$ is solution, then

$$V_t(b_t) = \sum_{s=t}^{T} \beta^{s-t} u(c_s^\star(b_t))$$

Recursive Representation

- With value functions, we can conveniently write the problem in each period as follows:

$$V_t(b_t) = \max_{b_{t+1}} u\left(y_t + (1+r)b_t - b_{t+1}\right) + \beta V_{t+1}(b_{t+1})$$

- Once we know $V_{T-1}(b_{T-1})$, it's easy to find $V_{T-2}(b_{T-2})$ and so forth
- At each step, we attain $b^\star_{t+1}(b_t)$. Can find consumption with

$$c^\star_t(b_t) = y_t + (1+r)b_t - b^\star_{t+1}(b_t)$$

- Solution can be found from $V_0(b_0)$: Can compute policies along entire trajectory following this
- This general approach to solving dynamic problems is called **Dynamic Programming**

Solution Method 1: Grids

- Simplest approach

1. Make a grid over b_t for each t: $\mathcal{B} = \{b^0, b^1, \ldots, b^N\}$
2. For each b_t on the grid, search over the grid to find the b_{t+1} that maximizes the recursive value function i.e. $\forall b_t \in \mathcal{B}$

$$V_t(b_t) = \max_{b_{t+1} \in \mathcal{B}} u\left(y_t + (1+r)b_t - b_{t+1}\right) + \beta V_{t+1}(b_{t+1})$$

3. Yields solution vector $\mathbf{b}_t^\star \in \mathcal{B}^N$ i.e. easy to store
4. Continue until $\mathbf{b}_0^\star$.

- Accuracy (and solution time) increase with N

Example 1: Three Periods

$$\max_{c_0,c_1,c_2\geq 0,b_1,b_2} \; u(c_0)+\beta u(c_1)+\beta^2 u(c_2)$$

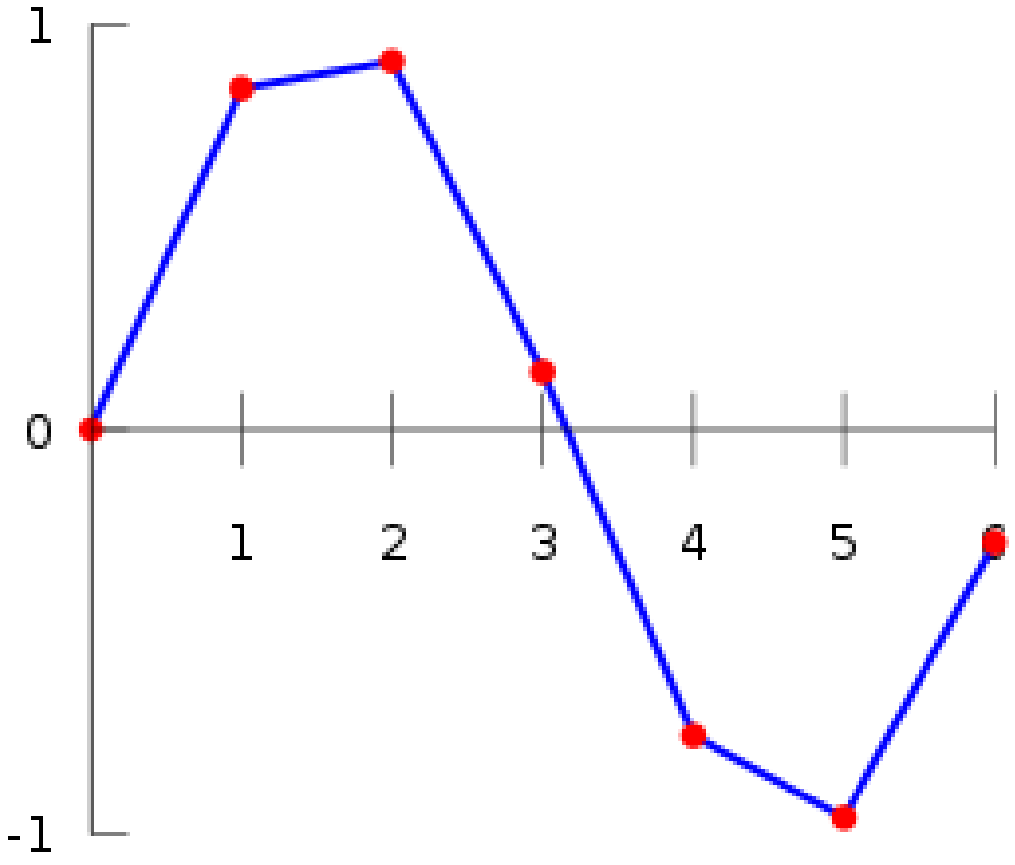
- Subject to same constraints as before
- Try two methods
  1. Simultaneous solution
  2. Backward induction/dynamic programming
- Notice when $\beta(1+r)=1$, it should be that $c_0=c_1=c_2$
- Backward induction scales up more easily than simultaneous solution as T grows

Solution Method 2: Interpolation

- Grid can sometimes be restrictive, lead to inaccuracies
- Want to be able to pick points *off* grid
- Linear Interpolation Method
  1. Solve period $T-1$ *on a grid of* b_{T-1}, but choosing *any* b_T
  2. For b_{T-1} off grid, define the value function as straight line connecting nearest grid points
  3. Solve period $T-2$ *on a grid of* b_{T-2}, but choosing *any* b_{T-1}
  4. ...
- Formula: If $b_{t+1} \in (b^n, b^{n+1})$

$$V_{t+1}(b_{t+1}) = V_{t+1}(b^n) + \left[\frac{V_{t+1}(b^{n+1}) - V_{t+1}(b^n)}{b^{n+1} - b^n}\right] \times (b_{t+1} - b^n)$$

Linear Interpolation



Extrapolation

- How to deal with points above/below whole grid?
- Two approaches
  1. Restrict them in the problem
  2. Continue first/last interior interpolation lines
- (1) is nice when possible!
- Often (2) is necessary; example later

Other Interpolations

- More sophisticated technique is a *spline* interpolation
- Splines are piecewise polynomials of order 2 or higher
  - Idea: Keep function smooth/differentiable at kinks
- Cubic spline most popular: If we have a set $(x_i, y_i)_{i=0}^n$, approximate each $[x_i, x_{i+1}]$ interval with a cubic

$$y = a_i + b_i x + c_i x^2 + d_i x^3$$

  - $4n$ coefficients: Point conditions and 1st/2nd derivative conditions $\implies$ Linear system

1. $y_i = a_i + b_i x_i + c_i x_i^2 + d_i x_i^3 \ (i = 1, \ldots, n)$
2. $y_i = a_{i+1} + b_{i+1} x_i + c_{i+1} x_i^2 + d_{i+1} x_i^3 \ (i = 0, \ldots, n-1)$
3. $b_i + 2c_i x_i + 3d_i x_i^2 = b_{i+1} + 2c_{i+1} x_i + 3d_{i+1} x_i^2 \ (i = 1, \ldots, n-1)$
4. $2c_i + 6d_i x_i = 2c_{i+1} + 6d_{i+1} x_i \ (1, \ldots, n-1)$

- Two conditions free: Often chosen setting $s'(x_0) = s'(x_n) = 0$

Life Lessons

- In solving models, *always* look for model-specific shortcuts
- Often more useful in speeding up computation than any supercomputer
- In our case, exploit the Euler Equation when $\beta(1+r)=1$:

$$u'(c_t) = \beta(1+r)u'(c_{t+1}) \implies c_t = \bar{c} \quad \forall t$$

- Put this together with the lifetime-budget constraint to get the solution:

$$\bar{c} = \frac{1}{1-(1+r)^{-T-1}}\left(\frac{r}{1+r}\right) \times \left[-(1+r)b_0 + \sum_{t=0}^{T}\left(\frac{1}{1+r}\right)^t y_t\right]$$

- Faster and better objective than any other method so far!

Extending the Horizon

- Take limit as $T \to \infty$...

$$\max_{\{c_t\}_{t=0}^{\infty} \geq 0, \{b_{t+1}\}_{t=0}^{T-1}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

$$s.t. \;\; c_t + b_{t+1} = y_t + (1+r)b_t$$

- What happens to last period condition? Need something to tie it down
  - Need to prevent *Ponzi schemes*: Rolling over borrowed money ad infinitum
  - Can achieve infinite utility with Ponzi schemes if not ruled out...

Ruling Out Ponzi Schemes

- Look at Lifetime BC of finite-horizon case:

$$NPV(\{c_t\}_{t=0}^{T}) = (1+r)b_0 + NPV(\{y_t\}_{t=0}^{T}) + \left(\frac{1}{1+r}\right)^{T} B_{T+1}$$

- To get a 'sensible' Lifetime BC, need only that

$$\lim_{T\to\infty} \left(\frac{1}{1+r}\right)^{T} B_{T+1}$$

  i.e. B_{T+1} cannot grow exponentially at a rate faster than r
- Call this the **No-Ponzi Condition**
  - Can be satisfied even if $\lim_{T\to\infty} B_{T+1} > 0$
- Past problem plus No-Ponzi Condition is well-defined

Solving

- How do we solve the infinite-horizon problem?
  - Backward induction directly no good...no terminal period
  - Not entirely true...more on this later!
- Last finite-horizon method will still work for *some* utilities e.g. log-utility

$$EE: \quad c_{t+1} = \beta(1+r)c_t$$

$$\implies NPV(\{c_t\}_{t=0}^{\infty}) = c_0 \sum_{t=0}^{\infty} \beta^t = \frac{c_0}{1-\beta}$$

$$\implies c_0 = (1-\beta) \times [(1+r)b_0 + NPV(\{y_t\}_{t=0}^{\infty})]$$

- Does not require that $\beta(1+r) = 1$

Solving: Shooting Algorithm

- If it doesn't happen that we can solve as above, we need another strategy
- Try to *solve it forward*: Bisection over c_0
  1. Conjecture a solution for the optimal c_0
  2. Iterate forward with the budget constraint/Euler Equation
  3. Check it at a large T: See if No-Ponzi Condition holds (approximately)
     - If it holds, we're done
     - If b_T too large, increase c_0 in next guess
     - If b_T too large (negatively), reduce c_0 in next guess
- Starting from b_t, conjecture a c_t and iterate forward with

$$b_{t+1} = y_t + (1+r)b_t - c_t$$

$$c_{t+1} = u'^{-1}\left(\frac{u'(c_t)}{\beta(1+r)}\right)$$

Equation (2) may need to be done numerically (not always)

Alternate Model: NCG Model

- Endogenize y_t process; save in capital instead of b_t
- Solve social planner's problem (economy efficient)
- $y_t = f(k_t)$
  1. $f(0) = 0$
  2. $f'(k_t) > 0$
  3. $f''(k_t) < 0$
- Capital depreciates at rate δ. k_0 given

$$\max_{\{c_t\}_{t=0}^{\infty}, \{k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t)$$

$$\text{s.t.} \quad c_t + k_{t+1} = f(k_t) + (1-\delta)k_t$$

NCG Model: Solution Characterization

1. Euler Equation

$$u'(c_t) = \beta[1 - \delta + f'(k_{t+1})]u'(c_{t+1})$$

2. Resource Constraint

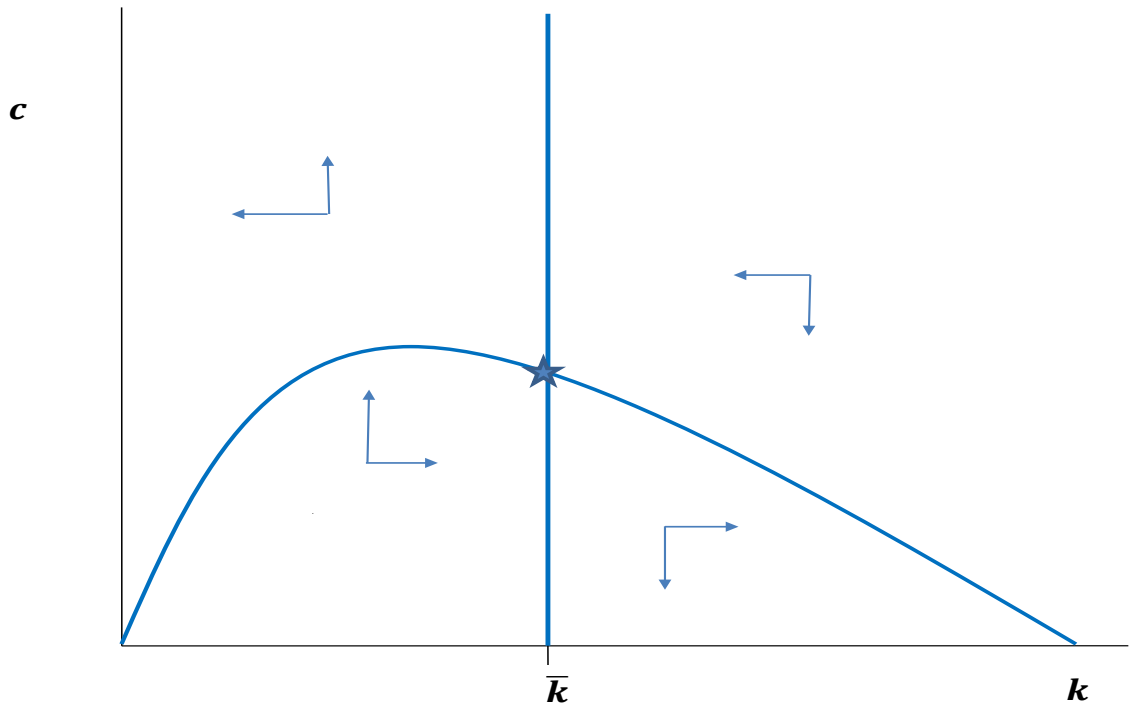
$$c_t + k_{t+1} = f(k_t) + (1 - \delta)k_t$$

- Can iterate like before on (c_0, k_0)
- Terminal condition? Transversality condition

$$\lim_{t \to \infty} k_{t+1}\beta^t u'(c_t) = 0$$

  - Like a FOC 'at $t = \infty$'
  - Prevents sub-optimal 'hoarding' of capital

NCG Model: SS Lines and Trajectories



- Steady state lines: Set $EE = 0$ and $BC = 0$
- Trajectories easy to derive on either side of SS line

NCG Model: Shooting Algorithm

- Only SS solution will satisfy TVC in long-run
- Search for $c_0(k_0)$ that sends system ratcheting to steady state
- This trajectory is unique (saddle-path stable)
- This problem is recursive! Delivers time-invariant solution along an endogenous grid:

$$c^\star(k_t) = c_t(k_t) \quad \forall k_t$$

- Could interpolate in between endogenous grid points for approximation to full-solution

The NCG Model Revisited

- Backward induction? How do we do it without a terminal period?
- Recall finite-horizon approach. For any $t < T$, equivalent problem:

$$V_t(k_t) = \max_{k_{t+1} \geq 0} u(f(k_t) + (1-\delta)k_t - k_{t+1}) + \beta V_{t+1}(k_{t+1})$$

- In infinite-horizon model, t should be irrelevant
  - Should be the case that

$$V(k) = \max_{k' \geq 0} u(f(k) + (1-\delta)k - k') + \beta V(k')$$

  - Solution $k'(k)$ should exactly be $k_1(k_0)$ in infinite-horizon

Thinking Recursively

- How to solve this model? Try something crazy...
  1. Guess any value function, $V^i(k)$
  2. Derive $V^{i+1}(k)$ by solving (for every k)

$$V^{i+1}(k) = \max_{k' \geq 0} u(f(k) + (1-\delta)k - k') + \beta V^i(k')$$

  3. Continue until $\sup_k ||V^{i+1}(k) - V^i(k)|| < \epsilon$ for a small $\epsilon > 0$
- For this model, it will work every time! Regardless of V^0
- This is called *Value Function Iteration*
- If it converges in I periods, you'll have approximations of
  1. The value function $V(k) = V^I(k)$
  2. The policy function $k'(k) = k^I(k)$

Analysis

- Provable results: $k'(k)$
  1. $k'(0) > 0$
  2. $\lim_{k \to \infty} k'(k) < k$
  3. $\frac{\partial k'(k)}{\partial k} < 1$
- Implies the model converges to a steady state (from either side)

Why?

- Why would we expect VFI to work?
- Answer lies in **Functional Analysis**
- A *Functional Space*, J, is a set of functions on a given domain, $\Omega \subset \mathcal{R}^N$ i.e. if $f \in J$, then

$$f : \Omega \to \mathcal{R}$$

- A *Functional Equation* maps functions into functions i.e. $\mathcal{H} : J^1 \to J^2$
- We can write most recursive economic problems as finding a function, $d \in J^1$ such that

$$\mathcal{H}(d) = \mathbf{0}$$

where $\mathbf{0}$ is the zero function, not the number zero

Example

- In NCG example, J^1 is all functions mapping $\mathcal{R}^+$ into $\mathcal{R}$.
  1. $d = V$
  2. $H(d) = 0 \iff$ for all $k \geq 0$,

$$V(k) - \left[\max_{k' \geq 0} u(f(k) + (1-\delta)k - k') + \beta V(k')\right] = 0$$

- Can write it equivalently by using Euler Equation
  1. $d = k'$
  2. $H(d) = 0 \iff$ for all $k \geq 0$,

$$u'\Big(f(k) + (1-\delta)k - k'(k)\Big) -$$

$$\beta \times [1 - \delta + f'(k'(k))] \times u'\Big(f(k'(k)) + (1-\delta)k'(k) - k'(k'(k))\Big) = 0$$

Fixed Points

- Equivalency to fixed points. If $T : J \to J$, an eq'm can often be described as a $d \in J$ such that

$$Td = d$$

  Notation: Often functional equations are just written as 'Td' instead of $T(d)$
- Completely equivalent to saying $Hd = Td - d$ and saying $Hd = \mathbf{0}$
- Fixed point approach lends itself to iterative approaches

Metric Spaces

- To characterize things cleanly, zoom out a little bit
- Reference: Stokey, Lucas, and Prescott (1989) Chapter 3

Definition

A **Metric Space** is a set S, together with a metric (distance function) ρ: $S \times S \to \mathcal{R}$ such that for all $x, y, z \in S$:

1. $\rho(x, y) \geq 0$, with equality iff $x = y$
2. $\rho(x, y) = \rho(y, x)$
3. $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$

- S could be a functional set e.g. $f, g \in J$
- Common metric is the sup-norm:
  $\rho(f, g) = \sup_{x \in \Omega} |f(x) - g(x)|$

Completeness

- A few more definitions

Definition

A sequence $\{x_n\}_{n=0}^{\infty}$ in S is a **Cauchy sequence** if for every $\epsilon > 0$, $\exists N_\epsilon$ such that

$$\rho(x_n, x_m) < \epsilon \quad \forall n, m \geq N_\epsilon$$

Definition

A metric space (S, ρ) is **complete** if every Cauchy sequence in S converges to an element in S.

- Complete metric space $\approx$ A closed interval on the real line (as opposed to an open one)

The Contraction Mapping Theorem

Definition

Let (S, ρ) be a metric space and $T : S \to S$. T is a **contraction mapping** with **modulus** β if for some $\beta \in (0, 1)$, $\rho(Tx, Ty) \leq \beta\rho(x, y)$ for all $x, y \in S$.

- If I apply my mapping to any two items in the set, those two elements always get closer together

Theorem (Contraction Mapping Theorem)

If (S, ρ) is a complete metric space and $T : S \to S$ is a contraction mapping with modulus β, then

1. *T has exactly one fixed point, $\nu \in S$*
2. *For any $\nu_0 \in S$, $\rho(T^n\nu_0, \nu) \leq \beta^n\rho(\nu_0, \nu)$ for $n = 0, 1, 2, \ldots$*

Application

- If we have a contraction mapping...great!
- We know that $\lim_{n\to\infty} \beta^n \rho(\nu_0, \nu) = 0$ since $\beta < 1$
  - Implies $\lim_{n\to\infty} T^n \nu_0 = \nu$
  - i.e. repeated iteration will eventually get us to the fixed point/equilibrium
- How do we know if it's a contraction?...

Application

Theorem (Blackwell's Sufficiency Theorem)

Let $X \subset \mathcal{R}^l$ and let $B(X)$ be a space of bounded functions $f : X \to \mathcal{R}$ with the sup-norm. Let $T : B(X) \to B(X)$ be an operator satisfying

1. *(Monotonicity) $f, g \in B(X)$ and $f(x) \leq g(x)$ for all $x \in X$ implies $(Tf)(x) \leq (Tg)(x)$ for all $x \in X$*
2. *(Discounting) There exists some $\beta \in (0, 1)$ such that*

$$[T(f + a)](x) \leq (Tf)(x) + \beta a, \quad \forall f \in B(X), \ a \geq 0, \ x \in X$$

Then T is a contraction with modulus β

Back the NCG Model

- Define TV as follows: For any $k \in \mathcal{R}^+$

$$(TV)(k) = \max_{k' \geq 0} u(f(k) + (1-\delta)k - k') + \beta V(k')$$

- Check Blackwell's conditions
  1. Let $V_L(k) \leq V_H(k)$ for all k. Easy to see that $TV_L(k) \leq TV_H(k)$ for all k
  2. $[T(V+a)](k) \leq (TV)(k) + \beta a$ for any positive a
- Both conditions satisfied! Repeatedly applying T i.e. VFI is guaranteed to converge to unique solution

Example 2: Adding Shocks

- **Real Business Cycle Model**
  - NCG model with new production function

$$y_t = A_t f(k_t)$$

  - A_t is total factor productivity: $A_t = \exp(z_t)$ and

$$z_t = \rho z_{t-1} + \epsilon_t$$

  - Recursive representation:

$$V(A,k) = \max_{k' \geq 0} \; u(Af(k) + (1-\delta)k - k') + \beta E_{\tilde{A}|A}[V(\tilde{A}, k')]$$

- Two states now: Capital stock and productivity
- Still satisfies Blackwell's sufficiency conditions (verify)

RBC Model: Solution Approach 1

1. Tauchenize z_t shock: $z \in \{z_1, z_2, \ldots, z_N\}$
2. Cubic spline interpolation across k for each z_i

- Delivers a collection of continuous, differentiable functions, $\{V_i(k)\}_{i=1}^{N}$
  - $V_i(k) \approx V(\exp(z_i), k)$ for $i = 1, 2, \ldots, N$

RBC Model: Solution Approach 2

- 2-Dimensional Cubic Spline over A and k
  - Don't worry about theory behind 2D splines; a bit technical, but it works
  - Higher than 2D: Splining not so clear
  - Alternative, more efficient continuous methods (more later!)
- Delivers 1 function in two dimensions $V(A, k)$

Speeding Things Up 1

- VFI *linearly* at a rate β
- A little slow for large scale problems
- Speeding up: Many tricks devised over the years
- Approach 1 (Judd [1998]): **Policy Function Iteration Hybrid**
- Start with guess $V^i(A, k)$
  1. Find corresponding policy, $k^i(A, k)$ via maximization
  2. Fix policy function: Apply Bellman (without maximization) M times
     - Delivers new $V^{i+1}(A, k)$
- Takes more iterations than VFI, BUT...
  - Tends to converge faster than linearly in β...helpful!
  - Can often speed things up by 10x or more
  - Still a contraction (convergence guaranteed)

Speeding Things Up 2

- **Endogenous Grid Method** (Barillas and Fernandez-Villaverde [2007])
- Observe FOC:

$$u'(c^\star(A,k)) = \beta E_{\tilde{A}|A}\left[V_k(\tilde{A}, k^\star(A,k))\right]$$

- Notice that if $k^\star$ was fixed,

$$c^\star(k^\star) = u'^{-1}\left(\beta E_{\tilde{A}|A}\left[V_k(\tilde{A}, k^\star)\right]\right)$$

  No maximization/root-finding required!
- Exploit this to speed things up (a lot)
  - Fix grid over optimal capital choice
  - Derive grid over current capital endogenously

Endogenous Grid Method: Requirements

- Store two different value functions:
  1. $V^i(A_t, k_{t+1}) = \beta E_{\tilde{A}_{t+1}|A_t}[V^i_{standard}(\tilde{A}_{t+1}, k_{t+1})]$
     - Expected discounted value of having k_{t+1} tomorrow if shock is z_t today
  2. $V^i(A_t, Y_t) =$
     $\max_{k_{t+1}} u(Y_t - k_{t+1}) + \beta E_{\tilde{A}_{t+1}|A_t}[V^i_{standard}(\tilde{A}_{t+1}, k_{t+1})]$
     - Value function as a function of *available market resources* $Y_t = f(k_t) + (1 - \delta)k_t$
- Store three different grids
  1. Tauchenized grid over z_t, G_z
  2. Capital grid over investment decisions, G_k
  3. Market resources grid, G_y: $Y_t \in G_y$ if $\exists (k, z) \in G_k \times G_z$ such that $Y_t = e^{z_t} f(k_t) + (1 - \delta)k_t$

Endogenous Grid Method: Procedure

- Begin with a guess $V^i(A_t, k_{t+1})$

1. Compute approximate derivative $V^i_k(A_t, k_{t+1})$ by averageing slopes of linear interpolation
2. Compute $c^\star(A_t, k_{t+1}) = u'^{-1}(V^i_k(A_t, k_{t+1}))$
3. Compute **necessary market resources** $Y(A_t, k_{t+1}) = c^\star(A_t, k_{t+1}) + k_{t+1}$
4. Compute $V^i(A_t, Y(A_t, k_{t+1})) = u(c^\star(A_t, k_{t+1})) + V^i(A_t, k_{t+1})$
5. Compute $V^{i+1}(A_t, Y_t)$ by interpolating $V^i(A_t, Y(A_t, k_{t+1}))$ on G_Y
6. Compute $V^{i+1}(A_t, k_{t+1}) = \beta E_{\tilde{A}_{t+1}|A_t}[V^{i+1}(A_t, Y_t)]$
7. Stop if $\sup_{i,j} |V^{i+1}(A_i, k_j) - V^i(A_i, k_j)| < \epsilon$

Endogenous Grid Method

- Once done, use root-solver to find endogenous capital grid i.e. $k \in G_{k,state}$ if $Y(A_t, k_{t+1}) = A_t f(k) + (1-\delta)k$ for some A_t and k_{t+1}
  - Value function is $V_{standard}(A_t, k(A_t, Y_t)) = V(A_t, Y_t)$ for all Y_t such that $Y_t = Y(A_t, k_{t+1})$
  - Policy function is the original grid! (with domain being endogenous grid)

Non-contractions

- Without the CMT...
- Cannot always guarantee a process will converge
  - Sometimes it does anyway! (count your blessings)
  - Other times, exploit other features
- Example: Monotonicity
  - Instead of working with *distances* in a metric space...
  - Work with a notion of *inequality* in an ordered set
  - Different fixed point theorem, but still works!

Partially Ordered Sets

Definition

A **Partially Ordered Set**, $(X, \leq)$, is a set taken together with a partial order i.e.

1. For any $a \in X$, $a \leq a$ (Reflexivity)
2. For any $a, b \in X$, $a \leq b$ and $b \leq a$ implies $a = b$ (Antisymmetry)
3. For any $a, b, c \in X$, $a \leq b$ and $b \leq c$ implies $a \leq c$ (Transitivity)

Complete Lattices

Definition

A partially ordered set, $(L, \leq)$, is called a **Complete Lattice** if every subset has a least upper bound and a greatest lower bound in L i.e. for any $M \subset L$,

1. $\sup M \in L$
2. $\inf M \in L$

- Akin to the notion of closedness/boundedness, but there is no distance metric

Tarski's Fixed Point Theorem

Definition

Let $(L, \leq)$ be a complete lattice, and suppose $T : L \to L$ is a monotone function i.e. for any $x, y \in L$, the following holds

$$x \leq y \implies Tx \leq Ty$$

Then the set of all fixed points in L for the function T is also a complete lattice.

- Cool theorem! Some interesting implications

1. There exists a greatest ($\bar{u}$) and a least ($\underline{u}$) fixed point (possibly the same; fixed point set non-empty)
2. If $x \leq Tx$, then $x \leq \underline{u}$
3. If $x \geq Tx$, then $x \geq \bar{u}$

Applying Tarski's: The Eaton-Gersovitz/Arellano Model

- Based on NCG/RBC model, but a few simple differences
  1. Gov't controls all consumption decisions
  2. No investment (endowment economy)
  3. Foreigners buy debt
  4. Gov't monopolist in debt market/foreign lenders competitive
     - Internalizes price changes from debt issuance
  5. Gov't cannot commit to repay debt
     - Will default if ex-post optimal
     - If default, excluded from credit markets forever i.e. $c_t = y_t$ and pay default cost

The Arellano Model: Sovereign

- Sovereign wants to solve similar Bellman equation

$$V(y,b) = \max_{b'} \; u(y - b + q(y,b')b') + \beta E[\hat{V}(\tilde{y},b')]$$

where

$$\hat{V}(y,b) = \max\{V(y,b), X(y)\}$$

and $X(y)$ is the utility value of defaulting in state y

$$X(y) = u(y \times [1 - \phi(y)]) + \beta E[X(\tilde{y})]$$

The Arellano Model: Lenders

- Foreign lenders are risk-neutral, deep-pocketed
- Competitively price debt $\Longrightarrow$
  - Can get a risk-free return, r
  - Can invest in risky sovereign debt (may get defaulted on)

$$q(y, b') = \frac{E_{\tilde{y}|y}\left[\mathbf{1}\{V(\tilde{y}, b') \geq X(\tilde{y})\right]}{1 + r}$$

Solving the Arellano Model

- That's it! Pretty simple, BUT...
  - VFI is no longer a contraction
  - Discounting in Blackwell sufficiency no longer holds, since V enters into q
- Alternative approach via Tarski: Iterate on q^i
  1. Define $\mathcal{Q}$ to be a complete lattice of decreasing functions i.e.

$$q \in \mathcal{Q} \text{ then } q : \mathcal{Y} \times \mathcal{B} \to \left[0, \frac{1}{1+r}\right]$$

  2. Take our order to be the absolute order i.e. if $q_1, q_2 \in \mathcal{Q}$

$$q_1 \leq q_2 \iff q_1(y,b) \leq q_2(y,b) \quad \forall (y,b) \in \mathcal{Y} \times \mathcal{B}$$

Solving the Arellano Model

- Iterative operator T defined as follows
  - Fixing q, the sovereign's Bellman is a contraction: $q^i \implies V^i$
  - Update step

$$(Tq^i)(y,b) = \frac{E_{\tilde{y}|y}\left[\mathbf{1}\{V^i(\tilde{y},b') \geq X(\tilde{y})\right]}{1+r}$$

Proposition

The operator T is a monotone self-map on $\mathcal{Q}$

- Easy to show Tq must be decreasing, self-map
- Monotonicity follows from fact that if $q_1 \leq q_2$, then $V_1 \leq V_2$ since prices are always higher in world 2

Applying Tarski's

- T is a monotone operator on a complete lattice, so Tarski's theorem applies
  - Get that an equilibrium exists (may be many)
- To find it is easy: Repeatedly apply T from either top or bottom
  - Set $q^0 = 0$; know $q^0 \leq q_{eq}$, which implies $Tq^0 \leq Tq_{eq} = q_{eq}$

$$\lim_{n \to \infty} T^n q^0 \leq T^n q_{eq} = q_{eq}$$

  - Gets bigger with each iteration
  - Must converge to something: Will converge to lowest fixed point (worst eq'm)
  - Same logic applies starting from best i.e. $q^0 = \frac{1}{1+r}$
    - Will converge to best