Topic 1
Solving Representative Agent Partial Equilibrium Models

Adam Hal Spencer

The University of Nottingham

Applied Computational Economics

Roadmap

Outline

- Three lectures that focus on dynamic model solving.
- The schedule is as follows:
  - **(1)** Theory of dynamic programming and how to implement it on a computer. Application to solving partial equilibrium models with representative agents.
  - **(2)** Solving representative agent models in general equilibrium,
  - **(3)** Solving heterogeneous agent models with idiosyncratic uncertainty.

Outline

- Today we'll look at the theory of dynamic programming.
- Then move on to how to implement it on a computer.
- All talk about lots of numerical recipes you can use to this end.
- I'm containing all of this to one lecture so we can move right on to more interesting stuff in the next class.

Markets v.s. Social Planners

- The bulk of this course will focus on solving models of market economies, (i.e. decentralised economies).
- As opposed to solving social planner's problems, (centralised economies).
- Market economies have the more interesting stuff: we can think about policy changes and the like.

Roadmap

Sequence Problems

- Consider a consumption-savings problem for a household who owns a capital stock

$$\max_{\{c_t,k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \frac{c_t^{1-\sigma}}{1-\sigma}$$

subject to their budget constraints and law of motion for capital

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

$$k_{t+1} \geqslant 0 \ \forall t \quad (2)$$

$$k_0 \text{ given}$$

where r is the return to saving exogenous to the household.

Sequence Problems

- Assume r is a constant for today.
- Partial equilibrium — we won't determine r in equilibrium — see the next topic.

Solving Sequence Problems

- We can solve the problem using a Lagrangian

$$\mathscr{L} = \sum_{t=0}^{\infty} \beta^t \frac{c_t^{1-\sigma}}{1-\sigma} + \sum_{t=0}^{\infty} \lambda_t [rk_t - c_t - k_{t+1} + (1-\delta)k_t]$$

- First order conditions (with respect to the controls)

$$\frac{\partial \mathscr{L}}{\partial c_t} = 0 \Rightarrow \beta^t c_t^{-\sigma} - \lambda_t = 0$$

$$\frac{\partial \mathscr{L}}{\partial k_{t+1}} = 0 \Rightarrow -\lambda_t + [\lambda_{t+1}\{r + (1-\delta)\}] = 0$$

Solving Sequence Problems

- Combine the two FOCs to get the inter-temporal Euler equation

$$c_t^{-\sigma} = \beta \left[c_{t+1}^{-\sigma} \{ r + (1 - \delta) \} \right] \tag{3}$$

Solving Sequence Problems

- The solution to the sequence problem is an infinite sequence $\{c_t^*, k_{t+1}^*\}_{t=0}^{\infty}$ such that

  (i) k_0 and r are given exogenously,

  (ii) The resource constraint (1) is satisfied $\forall t$,

  (iii) The inter-temporal Euler equation (3) is satisfied $\forall t$,

  (iv) The transversality condition is satisfied.

- Condition (iii) is a necessary condition for the solution.

- Conditions (i) and (iv) are boundary conditions for the sequence problem.

  $\Rightarrow$ They pin-down the right solution.

Solving Sequence Problems

- What's the issue here?
- We have an infinite sequence to compute!
- No matter how sophisticated it may be, a computer can't solve an infinite dimensional problem.
- Is there any hope...?

Roadmap

Recursive Formulation

- An alternative approach to using a Lagrangian is to use a recursive formulation in conjunction with the Envelope theorem.
- All about state variables.
- A state variable totally describes the state of a dynamic system at a given time period.

Value Function

- The value function gives us the value of the objective at the optimal solution to the problem, (for the given state).

- For our consumption-savings problem, with initial state (k_0), the value function $V(k_0)$ is

$$V(k_0) = \sum_{t=0}^{\infty} \beta^t \frac{(c_t^*)^{1-\sigma}}{1-\sigma}$$

  where $\{c_t^*, k_{t+1}^*\}_{t=0}^{\infty}$ solves the sequence problem.

- It's just our objective with the optimal solution plugged-in.

Recursive Formulation

- Heuristically, see that

$$
\begin{aligned}
V(k_0) &= \max_{\{c_t,k_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t \frac{c_t^{1-\sigma}}{1-\sigma} \\
&= \max_{\{c_0,k_1\}} \frac{c_0^{1-\sigma}}{1-\sigma} + \max_{\{c_t,k_{t+1}\}_{t=1}^{\infty}} \sum_{t=1}^{\infty} \beta^t \frac{c_t^{1-\sigma}}{1-\sigma} \\
&= \max_{\{c_0,k_1\}} \frac{c_0^{1-\sigma}}{1-\sigma} + \beta[V(k_1)]
\end{aligned}
$$

where the β comes out the front since the value function at $t = 1$ doesn't have the period utility discounted.

Recursive Formulation

- The recursive formulation [starting at time $t = 0$] for the social planner's problem above is given as

$$V(k_0) = \max_{\{c_0, k_1\}} \frac{c_0^{1-\sigma}}{1-\sigma} + \beta[V(k_1)]$$

subject to

$$c_0 + k_1 - (1-\delta)k_0 = rk_0$$

- This setup is referred to as a Bellman equation or a functional equation.

Recursive Formulation

- What does this problem say?
- If you tell me your initial state, k_0, this formulation tells you the value associated with all your future decisions.
- Notice that at period $t = 1$, we'll have a new state k_1.
- Then the Bellman equation tells me the value $V(k_1)$.
- The problem is the same every period for this infinite-horizon problem.
- The only thing that matters is the state k!

Recursive Formulation

- The problem is the same every period

$$V(k) = \max_{\{c,k'\}} \frac{c^{1-\sigma}}{1-\sigma} + \beta[V(k')]$$

subject to

$$c + k' - (1-\delta)k = rk$$

where variables with $'$ superscripts denote next period's variables.

Recursive Formulation

- The solution to this problem will be given by functions $V(k), k'(k)$ and $c(k)$.
- The latter two are known as policy functions.
- Notice again that they are time invariant.
- Tell me the current state and I'll tell you the optimal control variables.

Solution

- What can we do with this thing?
- One option: sub-in the constraint and take derivatives

$$\frac{\partial V(k)}{\partial k'} = 0 \Rightarrow (-1)(c)^{-\sigma} + \beta \left[\frac{\partial V(k')}{\partial k'} \right] = 0$$

- Issue: we don't know what $\frac{\partial V(k')}{\partial k'}$ is!
- Envelope theorem to the rescue.

Envelope Theorem

- The Envelope Theorem says that

$$
\begin{aligned}
\frac{\partial V(k)}{\partial k} &= \frac{\partial}{\partial k}\left\{\frac{c^{1-\sigma}}{1-\sigma} + \beta[V(k')]\right\} \\
&= \frac{\partial}{\partial k}\left\{\frac{[rk + (1-\delta)k - k']^{1-\sigma}}{1-\sigma} + \beta[V(k')]\right\} \\
&= c^{-\sigma}[r + (1-\delta)]
\end{aligned}
$$

i.e. just look for the places where k features and take the derivative: no need to worry about functions of k.

Envelope Theorem

- We can then iterate forwards by one period

$$\frac{\partial V(k')}{\partial k'} = (c')^{-\sigma}[r + (1-\delta)]$$

Euler Equation

- Combine the updated envelope condition with the FOC for capital to get

$$c^{-\sigma} = \beta \left\{ (c')^{-\sigma} [r + (1 - \delta)] \right\}$$

which is our standard Euler equation!

- But this isn't that useful!

- We're right back to where we were with the sequence problem.

More on the Value Function

- We're so used to taking derivatives in these problems.
- In deriving the Euler equation using the Envelope theorem, we haven't made much use of the value function itself.
- The value function turns-out to be a special object.
- Can we go further using this object $V(k)$?
- Bellman equations turn out to be contraction mappings.
- We can leverage this in taking these equations to a computer.
- Did you pay attention in real analysis class as an undergrad?

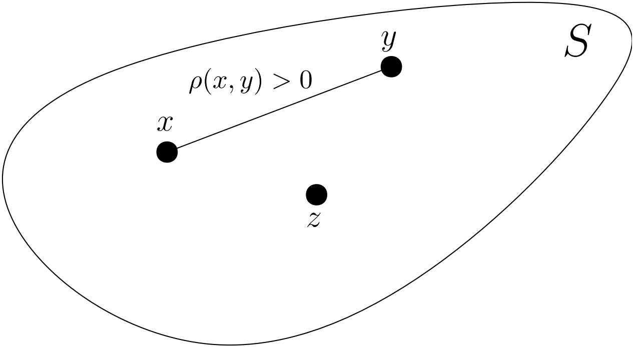
Metric Spaces and Sequences

- **Definition 1:** a metric space is a set S together with a metric $\rho : S \times S \to \mathbb{R}^+$ such that for all $x, y, z \in S$
  - $\rho(x, y) \geqslant 0$ with $\rho(x, y) = 0 \iff x = y$.
  - $\rho(x, y) = \rho(y, x)$,
  - $\rho(x, z) \leqslant \rho(x, y) + \rho(y, z)$

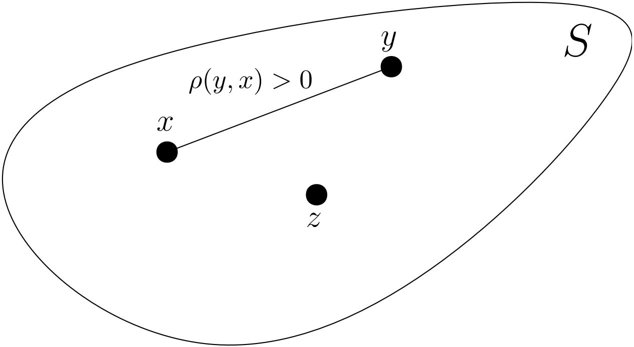
  which are often called the properties of positivity, symmetry and the triangle inequality.

- You can think of $\rho(x, y)$ as being like a distance measure between points in the set S.

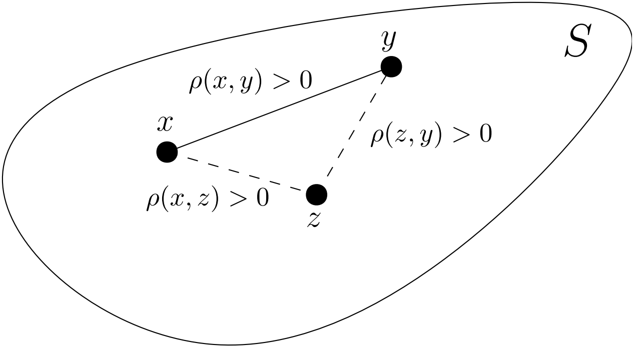
Metric Spaces and Sequences



Metric Spaces and Sequences: Symmetry



Metric Spaces and Sequences: Triangle Inequality



Metric Spaces and Sequences

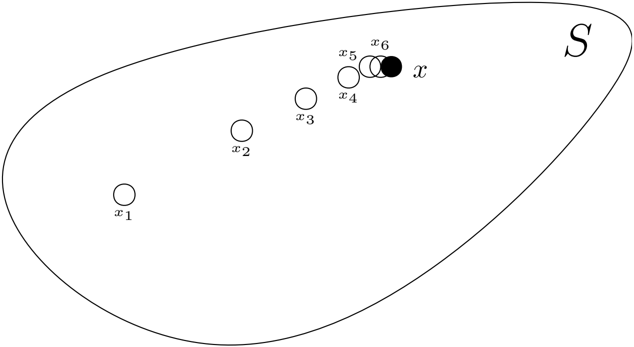
- **Definition 2:** a sequence $\{x_n\}_{n=0}^{\infty}$ in S converges to $x \in S$ if, for each $\epsilon > 0$, $\exists N_\epsilon \in \mathbb{N}$ such that

$$\rho(x_n, x) < \epsilon$$

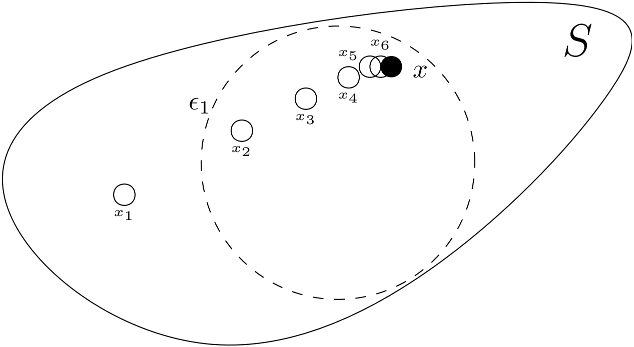
  for all $n \geqslant N_\epsilon$.

- After a certain point, we can trap the sequence inside an arbitrarily-small ball.

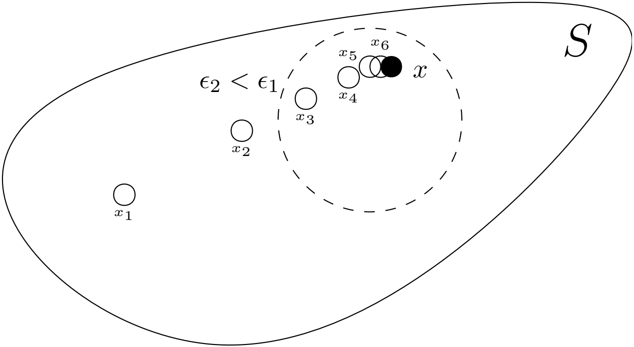
Metric Spaces and Sequences



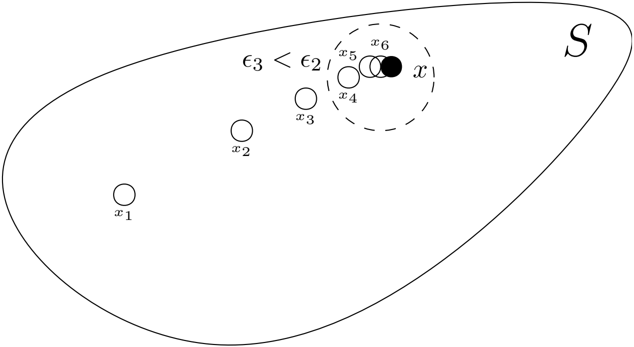
Metric Spaces and Sequences



Metric Spaces and Sequences



Metric Spaces and Sequences



Metric Spaces and Sequences

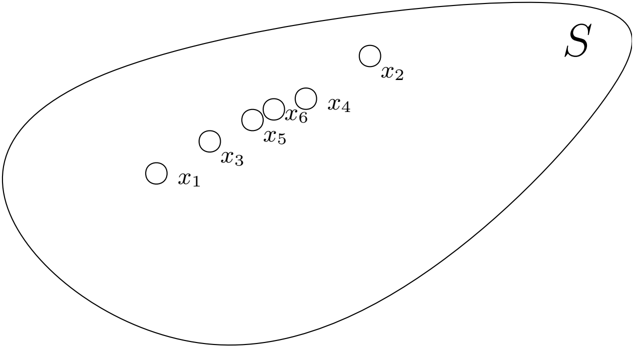
- **Definition 3:** a sequence $\{x_n\}_{n=0}^{\infty}$ in S is a Cauchy sequence if for each $\epsilon > 0$, $\exists N_\epsilon \in \mathbb{N}$ such that

$$\rho(x_n, x_m) < \epsilon$$

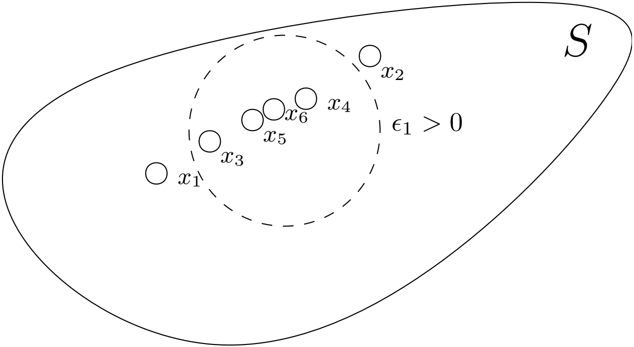
  for all $n, m \geqslant N_\epsilon$ with $n, m \in \mathbb{N}$.

- Points in the sequence are getting closer and closer.

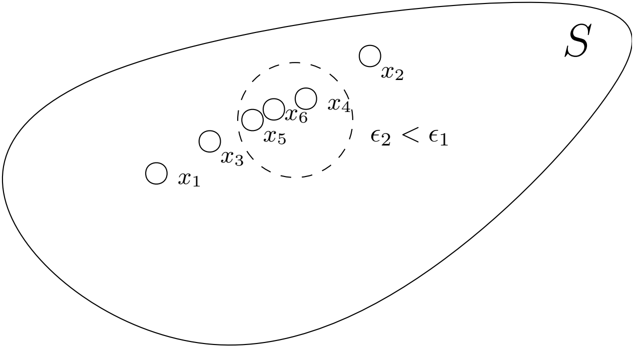
Metric Spaces and Sequences



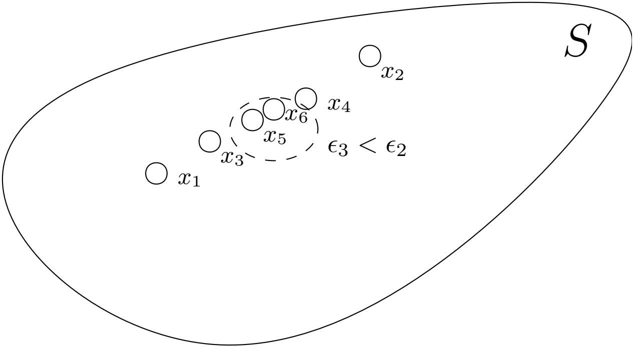
Metric Spaces and Sequences



Metric Spaces and Sequences



Metric Spaces and Sequences



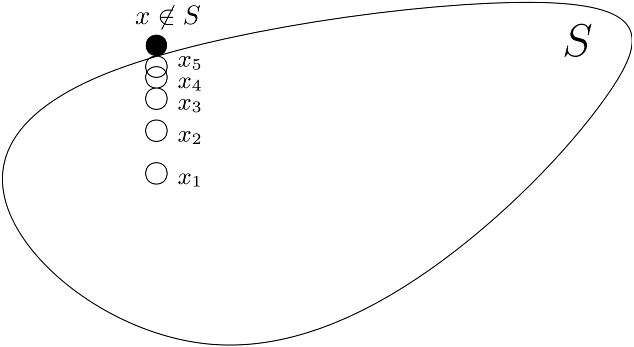
Metric Spaces and Sequences

- Sequence $\{x_n\}_{n=0}^{\infty} \in S$ convergent $\Rightarrow \{x_n\}_{n=0}^{\infty} \in S$ is Cauchy.

Metric Spaces and Sequences

- Sequence $\{x_n\}_{n=0}^{\infty} \in S$ is Cauchy $\not\Rightarrow \{x_n\}_{n=0}^{\infty} \in S$ convergent.
- E.g. $x_n = \frac{1}{n}$ for $n \in \mathbb{N}$ and $S = (0, 1]$ since $0 \notin S$.

Metric Spaces and Sequences



Metric Spaces and Sequences

- **Definition 4:** a metric space (S, ρ) is complete if every Cauchy sequence in S converges to a point in S.

Metric Spaces and Sequences

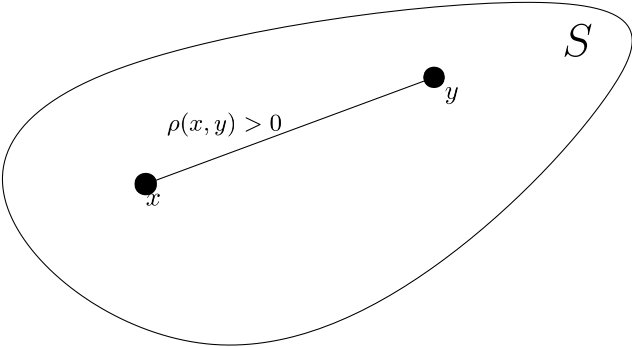
- **Definition 5:** let (S, ρ) be a metric space and $T : S \to S$ be a function mapping S into itself. T is a contraction mapping with modulus β if for $\beta \in (0, 1)$,

$$\rho(Tx, Ty) \leqslant \beta\rho(x, y)$$

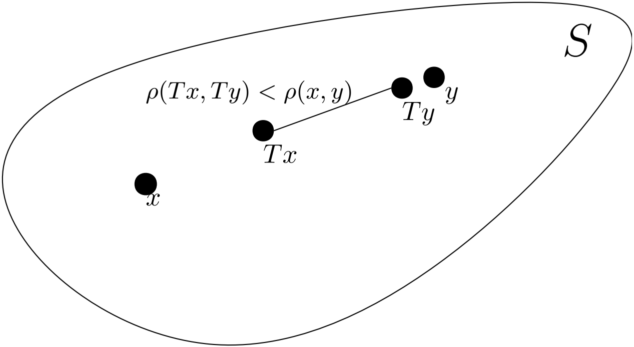
  for all $x, y \in S$.

- The function brings points closer and closer together.

Metric Spaces and Sequences



Metric Spaces and Sequences



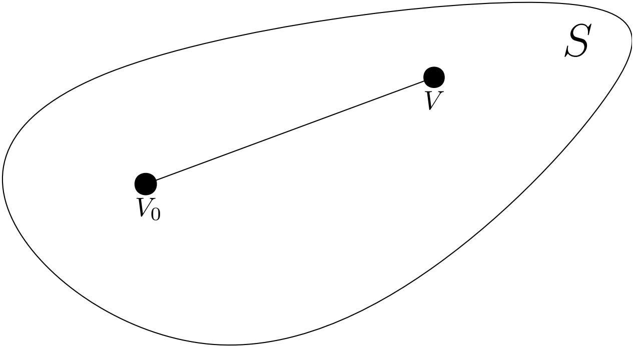
Contraction Mapping Theorem

- **Theorem 1 (Contraction Mapping Theorem)**: if (S, ρ) is a complete metric space and $T : S \rightarrow S$ is a contraction mapping with modulus $\beta \in (0, 1)$, then
  - T has exactly one fixed point $V \in S$ such that $V = TV$.
  - For any $V_0 \in S$, $\rho(T^n V_0, V) < \beta^n \rho(V_0, V)$ with $n = 0, 1, 2, \ldots$

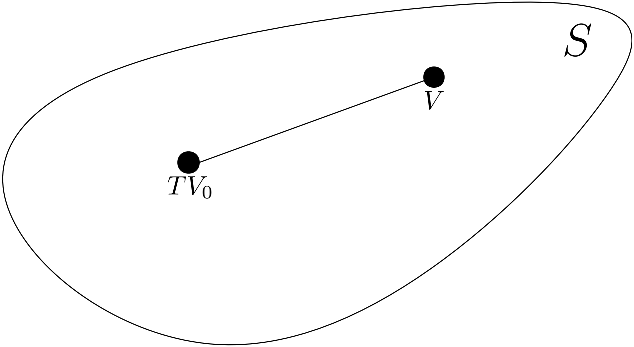
  Proof: ask Omar or Giammario in your theory classes!

- A sequence of successive applications of the function to a point brings us closer and closer to the unique fixed point.

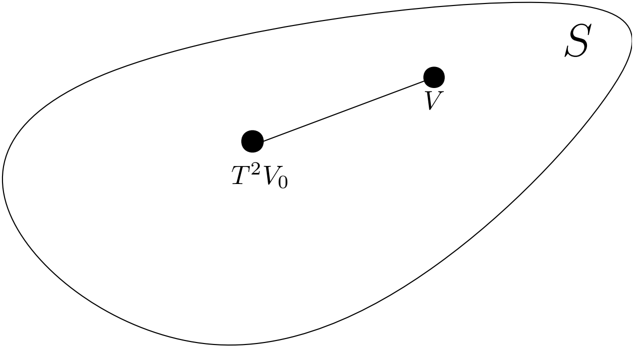
Metric Spaces and Sequences



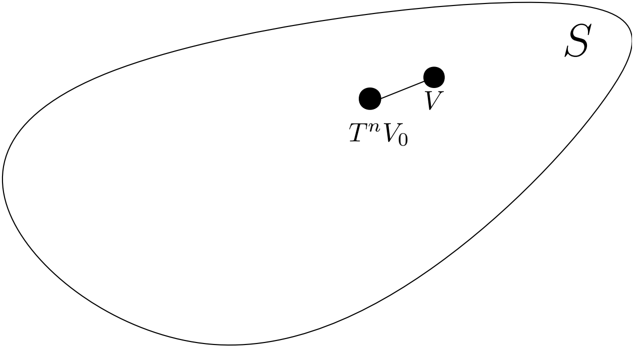
Metric Spaces and Sequences



Metric Spaces and Sequences



Metric Spaces and Sequences



Metric Spaces and Sequences

- So far we've looked at points in a set.
- Let's generalise now to talk about functions.

Contraction Mapping Theorem

- **Theorem 2 (Blackwell's Sufficient Conditions)**: let $X \subset \mathbb{R}^l$ and $B(X)$ be the space of bounded functions $V : X \to \mathbb{R}$ with the sup norm. Let $T : B(X) \to B(X)$ be an operator satisfying
  - (Monotonicity): let $V, W \in B(X)$, if $V(x) \leqslant W(x)$ for all $x \in X$ then $TV(x) \leqslant TW(x)$,
  - (Discounting): there exists some constant $\beta \in (0, 1)$ such that for all $V \in B(X)$ and $a \geqslant 0$, we have

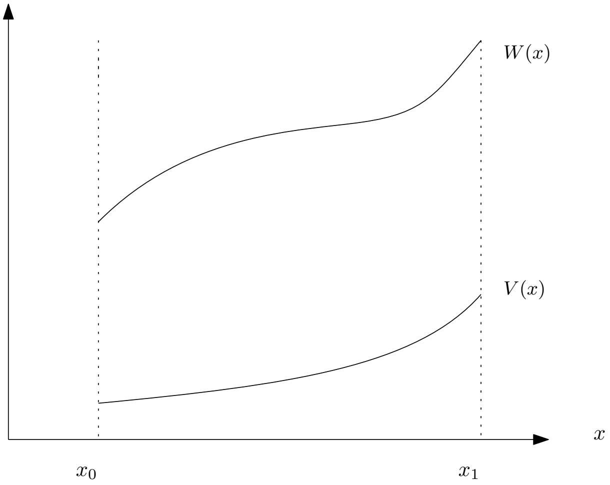
$$T(V + a) \leqslant TV + \beta a$$

  then T is a contraction with modulus β.

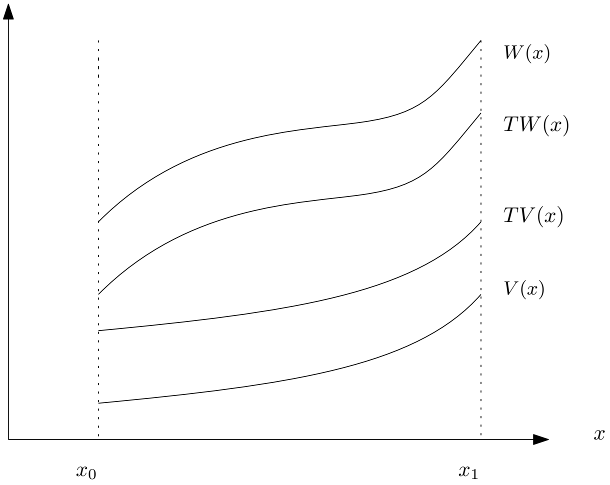
- Where note that the sup norm is defined as

$$||f||_\infty = \sup\{|f(x)| : x \in X\}$$

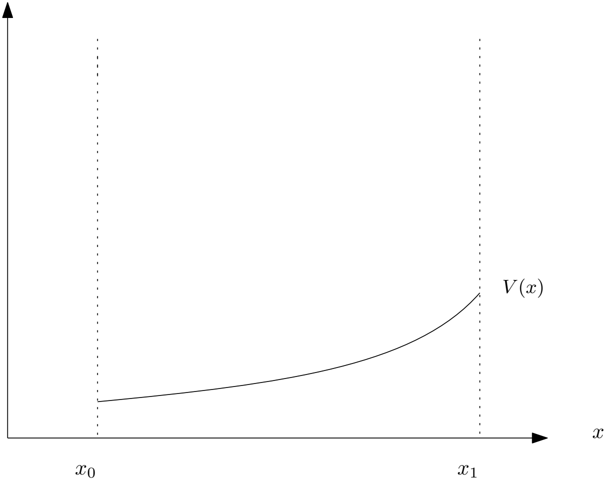
Metric Spaces and Sequences: Monotonicity



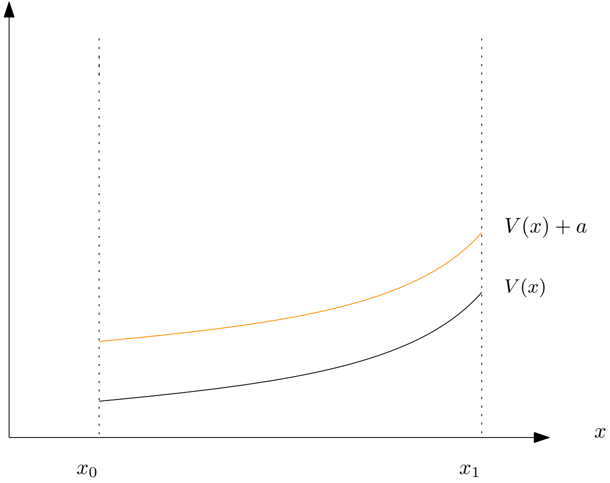
Metric Spaces and Sequences: Monotonicity



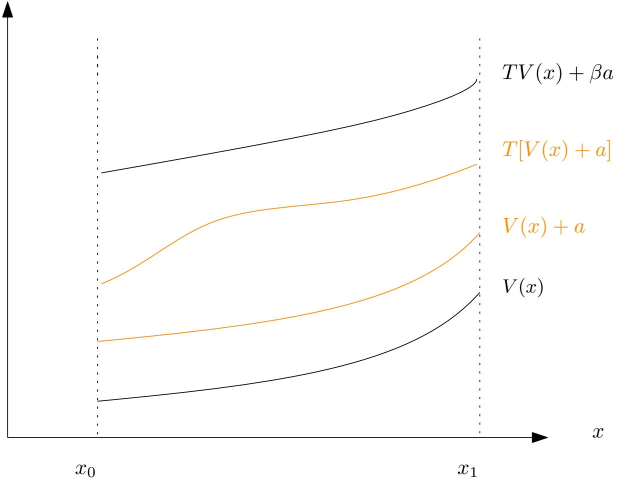
Metric Spaces and Sequences: Discounting



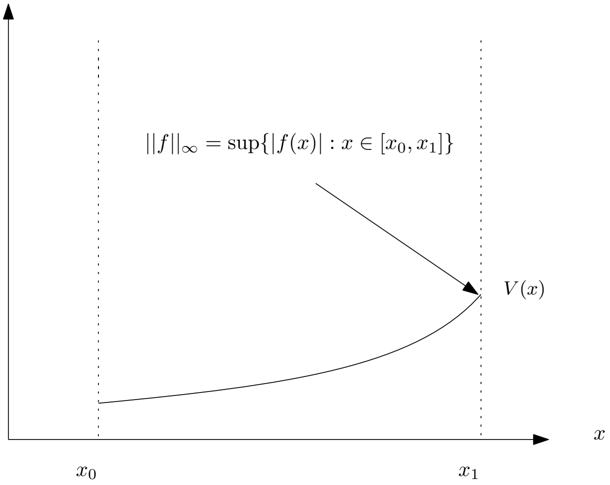
Metric Spaces and Sequences: Discounting



Metric Spaces and Sequences: Discounting



Metric Spaces and Sequences: Sup Norm



Contraction Mapping Theorem

- How does this help us?
- Our beloved Bellman equation turns-out to be a contraction mapping.

Contraction Mapping Theorem

- Recall our value function looked like

$$V(k) = \max_{k'} \frac{c^{1-\sigma}}{1-\sigma} + \beta[V(k')]$$

where $c = rk - k' + (1-\delta)k$.

- Let's define the operator T as

$$(TV)(k) = \max_{k'} \frac{[rk - k' + (1-\delta)k]^{1-\sigma}}{1-\sigma} + \beta[V(k')]$$

- Want to know if T is a contraction and does there exist a V unique such that $V(k) = (TV)(k)$.

Contraction Mapping Theorem

- Monotonicity: consider V, W such that $V(k) \leqslant W(k)$ for all k.
- Want to show that $(TV)(k) \leqslant (TW)(k)$.
- Denote $\tilde{k}$ the optimal investment (k') for the V functional.
- Follows then that

$$\begin{aligned}
(TV)(k) &= \frac{[rk - \tilde{k} + (1-\delta)k]^{1-\sigma}}{1-\sigma} + \beta[V(\tilde{k})] \\
&\leqslant \frac{[rk - \tilde{k} + (1-\delta)k]^{1-\sigma}}{1-\sigma} + \beta[W(\tilde{k})] \\
&\leqslant \max_{k'} \frac{[rk - k' + (1-\delta)k]^{1-\sigma}}{1-\sigma} + \beta[W(k')] \\
&= (TW)(k)
\end{aligned}$$

meaning that the Bellman equation is monotonic.

Contraction Mapping Theorem

- Discounting: consider a functional V and a positive constant a.
- See that

$$
\begin{aligned}
(T(V+a))(k) &= \max_{k'} \frac{[rk - k' + (1-\delta)k]^{1-\sigma}}{1-\sigma} + \beta[V(k') + a] \\
&= \max_{k'} \frac{[rk - k' + (1-\delta)k]^{1-\sigma}}{1-\sigma} + \beta[V(k')] + \beta a \\
&= (TV)(k) + \beta a
\end{aligned}
$$

meaning that the discounting property is satisfied.

Contraction Mapping Theorem

- FYI: the space of bounded functions with the sup norm is complete.
- Since the Bellman equation is a contraction, its fixed point is unique.
- We still have no analytical solution for $V(k)$.
- We can leverage the fact that the Bellman equation is a contraction to solve for $V(k)$ numerically.

Roadmap

Value Function Iteration

- Recall the second point from the contraction mapping theorem.
- If we start with some point in the metric space and keep applying the contraction, the sequence of iterates will eventually converge to the fixed point.
- Our primary object of interest in the consumption-savings model is the set of policy functions — $k'(k), c(k)$ — they tell us how to best allocate our resources.
- If we first solve for the value function, we can find these policy functions from the Bellman equation.

Value Function Iteration

- The general procedure is:
  1. Start with a guess for your value function, $V_0(k)$.
  2. Update your guess in the Bellman equation

$$V_1(k) = \max_{c,k'} \frac{c^{1-\sigma}}{1-\sigma} + \beta V_0(k')$$

where $c = rk - k' + (1-\delta)k$. That is — take $V_0(k')$ as the true value function for next period and then optimise over k', c. This gives you a new value function $V_1(k)$.

  3. Keep doing this

$$V_{n+1}(k) = \max_{c,k'} \frac{c^{1-\sigma}}{1-\sigma} + \beta V_n(k') \tag{4}$$

where $c = rk - k' + (1-\delta)k$ until convergence.

Convergence

- Recall that we we're dealing with a metric space here.
- Where a metric "measures the distance" between objects in the space.
- We can utilise the metric to see how close two points in the sequence of iterates, $V_{n+1}(k)$ and $V_n(k)$ are!
- When this distance is sufficiently small, we've approximately achieved convergence.

Convergence

- Utilise the sup norm

$$||V_{n+1} - V_n||_\infty = \sup\{|V_{n+1}(k) - V_n(k)| : k \in \mathbb{R}\}$$

- This norm, (a special type of metric), finds the biggest discrepancy in values between successive iterates.
- Keep on iterating until the "biggest difference" gets sufficiently small.

Convergence

- If I've done my job correctly, you'll see the following (rather than faces of loved ones), on your deathbed...

Convergence

```
Difference   3.5705566E-03
Difference   2.9792786E-03
Difference   2.4871826E-03
Difference   2.0761490E-03
Difference   1.7337799E-03
Difference   1.4495850E-03
Difference   1.2102127E-03
Difference   1.0108948E-03
Difference   8.4590912E-04
Difference   7.0858002E-04
Difference   5.9413910E-04
Difference   4.9781799E-04
Difference   4.1770935E-04
Difference   3.4999847E-04
Difference   2.9373169E-04
Difference   2.4652481E-04
Difference   2.0623207E-04
Difference   1.7333031E-04
Difference   1.4495850E-04
Difference   1.2159348E-04
Difference   1.0228157E-04
Difference   8.5830688E-05
```

Roadmap

Discretisation

- Everything I say probably makes intuitive sense.
- How do you actually do it when you're sitting-down at your computer screen?
- The starting point is called grid search.

Discretisation

- Recall that our state and control variables were over the space $\mathbb{R}$.
- We want to iterate several times on the Bellman equation.
- Notice that doing this as per equation (4) actually requires optimising at each iteration.
- That is: to find $V_{n+1}(k)$, we need to optimise over k' using $V_n(k)$ on the right-hand side.
- How do we do this? $\mathbb{R}$ is a large set to be optimising over...

Discretisation

- Postulate an upper-bound for capital, denoted $\bar{k}$.
- "Chop-up" the interval $[0, \bar{\mathscr{k}}]$ into M discrete increments.
- This will leave you with a set $\mathscr{k} \equiv \{0, \mathscr{k}_1, \mathscr{k}_2, \mathscr{k}_3, ..., \bar{\mathscr{k}}\}$.
- Just search over that set!
- E.g. if I come into the world with state $\mathscr{k}_5$, what choice from set $\mathscr{k}$ will maximise my value?

Discretisation

- What should your guess for $\bar{k}$ be?
- This is all partial equilibrium today: $\bar{k}$ will just be arbitrary in the problem set.
- There are tricks you can use to guess a good upper-bound for k when r is endogenous: we'll discuss next topic.

Roadmap

Stochastic Problems

- Everything we've considered so far has been deterministic.
- How do we implement solutions to problems with stochastic variables?
- Our partial equilibrium model: assume r_t is exogenous and time-varying.

Stochastic Problems

- Consider the social planner's problem from the stochastic consumption-savings problem.

$$\max_{\{c_t, k_{t+1}\}_{t=0}^{\infty}} \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \frac{c_t^{1-\sigma}}{1-\sigma}$$

subject to their resource constraints and law of motion for capital

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

$$\log(r_t) = \rho_r \log(r_{t-1}) + \epsilon_{r,t}, \quad \epsilon_{r,t} \sim N(0,1)$$

$$k_{t+1} \geqslant 0 \ \forall t$$

$$k_0, r_0 \text{ given}$$

Stochastic Problems

- Quantitative macro is obsessed with this AR(1) process.
- Consider a general stochastic process

$$y_t = \mu(1-\rho) + \rho y_{t-1} + \epsilon_t, \;\; \epsilon_t \sim N(0, \sigma^2) \tag{5}$$

- This is a process for a continuous variable.
- Again, we can discretise this process, just like we did with the state space for capital.

Stochastic Problems

- Take the continuous stochastic process and convert it into a discrete Markov process.
- How many gridpoints do we want to approximate (5) with?
- E.g. say we approximate with two gridpoints — high or low (denote them by $y_t \in \{y^L, y^H\}$).

Stochastic Problems

- A Markov process in this case would be a transition matrix of the form

$$Q = \begin{bmatrix} q_{LL} & q_{LH} \\ q_{HL} & q_{HH} \end{bmatrix} \tag{6}$$

where

$$q_{LL} + q_{LH} = 1$$
$$q_{HL} + q_{HH} = 1.$$

- The rows correspond to the period t state and columns are for $t+1$ state.
- Probability of staying in current state plus probability of moving to the other sums to unity.

Stochastic Problems

- How do we discretise (i.e. move from equation (5) to (6))?
- Two predominant approaches: Tauchen (1986) and Adda & Cooper (2003).
- The former chops the distribution for y_t up into equal interval lengths, while the latter instead looks at areas.

Adda & Cooper (2003) AR(1) Approximation

- We'll follows the Adda & Cooper (2003) approach.
- The procedure is:
  - **(1)** Discretise process into $N \in \mathbb{N}$ intervals,
  - **(2)** Get the conditional mean of each interval (discretised y_t values),
  - **(3)** Find the conditional transition probability of moving from one interval to the next, (transition matrix).
- See the recipe appendix slides for the procedure.

Adda & Cooper (2003) AR(1) Approximation

- The end result is a vector $\vec{y}$ (size $N \times 1$) of discretised y_t values and a transition matrix Q (size $N \times N$).
- How can we use this now?

Stochastic Model

- The recursive formulation of the stochastic consumption-savings model is given by

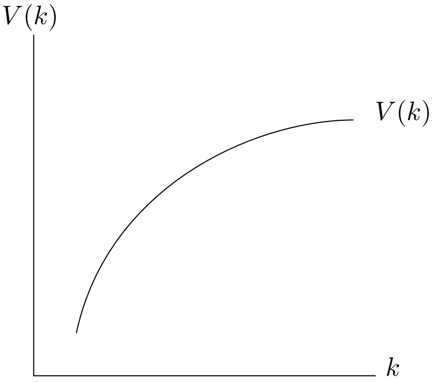
$$V(k, r) = \max_{\{c, k'\}} \frac{c^{1-\sigma}}{1-\sigma} + \beta \mathbb{E}_r[V(k', r')]$$

subject to

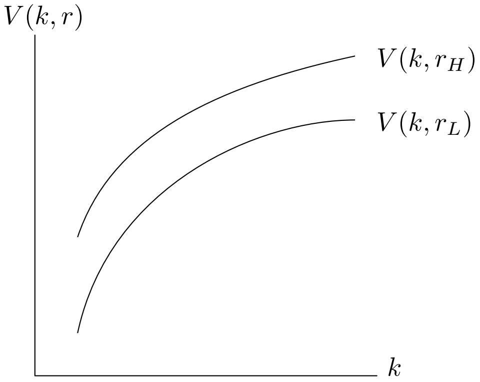
$$c + k' - (1-\delta)k = rk$$
$$\log(r) = \rho_r \log(r_{-1}) + \epsilon_r, \ \ \epsilon_r \sim N(0,1)$$

- The interest rate variable is a new state now, (r_{-} denotes last period's rate).
- The process for r is now summarised by our discretised vector and transition matrix.
- The expectation is over r' conditional on stochastic state r.

Deterministic Value Function



Stochastic Value Function



Stochastic Model

- How does the stochastic problem differ from the deterministic problem computationally?
- We need to account for the additional state, (an extra loop in the code).
- Our AR(1) discretisation process gives a vector of interest rate values $\vec{r}$ and transition matrix Q.
- We also need to crunch a sum in the Bellman equation for the expectation.

Stochastic Model

- The definition of the expectation for the discretised r variable

$$\mathbb{E}_r[V(k', r')] = \sum_{i=1}^{N} q(r, r' = r_i) V(k', r' = r_i)$$

where the stochastic state is discretised to $N \times 1$ vector $\vec{r}$ and $q(r, r' = r_i)$ is the transition probability from current state r to r_i next period from the $N \times N$ matrix Q.

Stochastic Value Function Iteration

- The general procedure is:
  1. Start with a guess for your value function, $V_0(k, r)$.
  2. Update your guess in the Bellman equation

$$V_1(k, r) = \max_{c,k'} \frac{c^{1-\sigma}}{1-\sigma} + \beta \mathbb{E}_r[V_0(k', r')]$$

     where $c = rk - k' + (1-\delta)k$. See that

$$\mathbb{E}_r[V_0(k', r')] = \sum_{i=1}^{N} q(r, r' = r_i) V_0(k', r' = r_i)$$

     where the current state is r. That is: compute the expectation assuming that the initial guess is the true value function.
  3. Keep doing this

$$V_{n+1}(k, r) = \max_{c,k'} \frac{c^{1-\sigma}}{1-\sigma} + \beta \mathbb{E}_r[V_n(k', r')]$$

     where $c = rk - k' + (1-\delta)k$ until convergence.

Roadmap

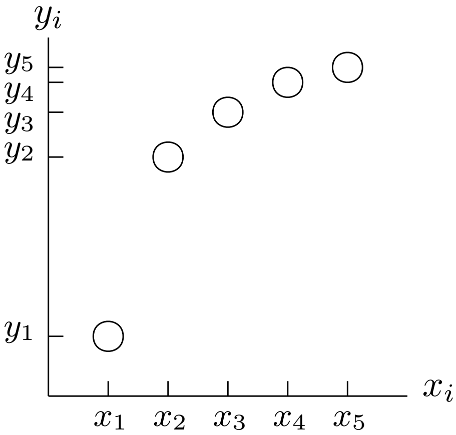
Functional Approximations

- So far we've been discretising everything.
- Means that we'll know the values of some function at a bunch of discrete points along an interval.
- What do we do if we need to know the value of the function at an arbitrary point outside of this grid?
- E.g. between two of our gridpoints.

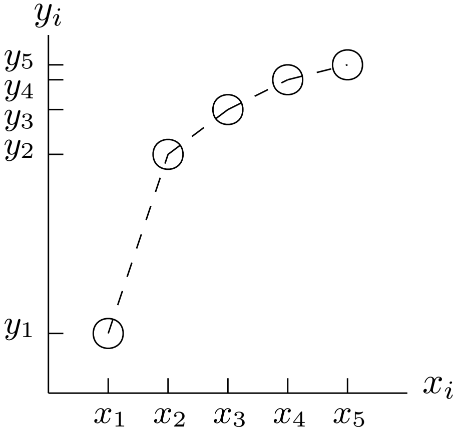
Piecewise Linear Interpolation

- Most basic application of this idea is to approximate a function using lines.
- Say we know $\{y_i = f(x_i)\}_{i=1}^{N}$ at some discrete set of points $\{x_i\}_{i=1}^{N}$.
- We can then construct an approximation that equals each of these evaluated points at the cut-offs, but assumes a linear form in all the intervals in between.

Piecewise Linear Interpolation



Piecewise Linear Interpolation



Piecewise Linear Interpolation

- Construct a function

$$I(x)_{[x_i,x_{i+1}]}(x) = A_i(x)y_i + (1 - A_i(x))y_{i+1}$$

where

$$A_i(x) = \frac{x_{i+1} - x}{x_{i+1} - x_i}.$$

- $A_i(x)$ measures how far along the interval $[x_i, x_{i+1}]$ the point x is.

Piecewise Linear Interpolation

- See that

$$\begin{aligned} I(x)_{[x_i,x_{i+1}]}(x_i) &= A_i(x_i)y_i + (1 - A_i(x_i))y_{i+1} \\ &= y_i \end{aligned}$$

  and

$$\begin{aligned} I(x)_{[x_i,x_{i+1}]}(x_{i+1}) &= A_i(x_i)y_i + (1 - A_i(x_i))y_{i+1} \\ &= y_{i+1}. \end{aligned}$$

- I.e. it hits the cut-offs exactly and is a linear combination for all the points in between.

Piecewise Linear Interpolation

- What does this process give us?
- A continuous approximation to the value function.
- Grid search only gives us the value function at the discrete points for the state space.
- We have an approximation for the value function for state values between each of the discretised state values.
- Simple approximation: just lines.
- See recipe appendix for other approximation methods.

Roadmap

Takeaways

- Dynamic programming.
- All the tools for the next lab.