

# Structural Estimation of Dynamic Stochastic Optimizing Models of Intertemporal Choice For Dummies!

Christopher Carroll<sup>1</sup>

<sup>1</sup>Johns Hopkins University and NBER  
ccarroll@jhu.edu

June 2012

<http://www.econ2.jhu.edu/people/ccarroll/SolvingMicroDSOPs-Slides.pdf>

- Efficient Solution Methods for Canonical  $C$  problem
  - CRRA utility
  - Plausible (microeconomically calibrated) uncertainty
  - Life cycle or infinite horizon
- How To Add a Second Choice Variable
- Method of Simulated Moments Estimation of Parameters

- Efficient Solution Methods for Canonical  $C$  problem
  - CRRA utility
    - Plausible (microeconomically calibrated) uncertainty
    - Life cycle or infinite horizon
- How To Add a Second Choice Variable
- Method of Simulated Moments Estimation of Parameters

- Efficient Solution Methods for Canonical  $C$  problem
  - CRRA utility
  - Plausible (microeconomically calibrated) uncertainty
    - Life cycle or infinite horizon
- How To Add a Second Choice Variable
- Method of Simulated Moments Estimation of Parameters

- Efficient Solution Methods for Canonical  $C$  problem
  - CRRA utility
  - Plausible (microeconomically calibrated) uncertainty
  - Life cycle or infinite horizon
- How To Add a Second Choice Variable
- Method of Simulated Moments Estimation of Parameters

- Efficient Solution Methods for Canonical  $C$  problem
  - CRRA utility
  - Plausible (microeconomically calibrated) uncertainty
  - Life cycle or infinite horizon
- How To Add a Second Choice Variable
- Method of Simulated Moments Estimation of Parameters

- Efficient Solution Methods for Canonical  $C$  problem
  - CRRA utility
  - Plausible (microeconomically calibrated) uncertainty
  - Life cycle or infinite horizon
- How To Add a Second Choice Variable
- Method of Simulated Moments Estimation of Parameters





## Bellman Equation

$$\mathbf{v}_t(m_t, \mathbf{p}_t) = \max_{c_t} u(c_t) + \mathbb{E}_t[\mathbf{v}_{t+1}(m_{t+1}, \mathbf{p}_{t+1})] \quad (4)$$

$m$  — 'market resources' (net worth plus current income)

$p$  — permanent labor income





- Non-CRRA utility
- Non-Friedman (transitory/permanent) income process
  - e.g., AR(1)
  - But micro evidence is consistent with Friedman

---

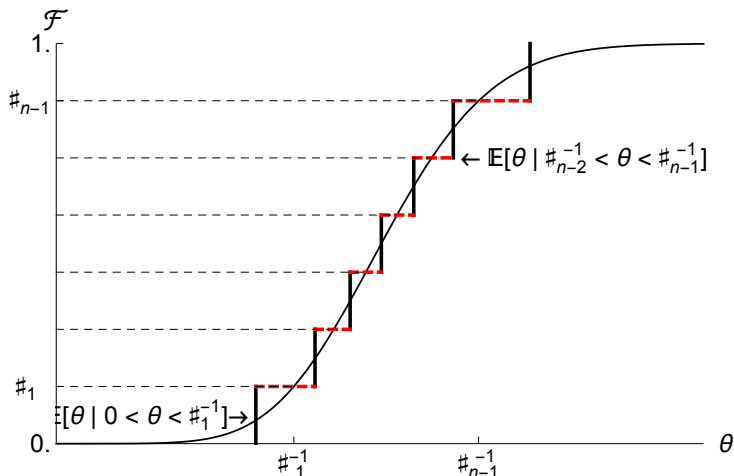
## When Doesn't Normalization Work?

- Non-CRRA utility
- Non-Friedman (transitory/permanent) income process
  - e.g., AR(1)
  - But micro evidence is consistent with Friedman

---

# Trick: Discretize the Risks

E.g. use an equiprobable 7-point distribution:





---



\_\_\_\_\_

# Trick: Interpolate a Consumption Rule

- 1 Define a grid of points  $\vec{m}$  (indexed  $m[i]$ )
- 2 Use numerical rootfinder to solve  $u'(c) = v'_t(m[i] - c)$ 
  - The  $c$  that solves this becomes  $c[i]$
- 3 Construct interpolating function  $\hat{c}$  by linear interpolation
  - 'Connect-the-dots'

# Trick: Interpolate a Consumption Rule

- ① Define a grid of points  $\vec{m}$  (indexed  $m[i]$ )
- ② Use numerical rootfinder to solve  $u'(c) = v'_t(m[i] - c)$ 
  - The  $c$  that solves this becomes  $c[i]$
- ③ Construct interpolating function  $\hat{c}$  by linear interpolation
  - 'Connect-the-dots'

# Trick: Interpolate a Consumption Rule

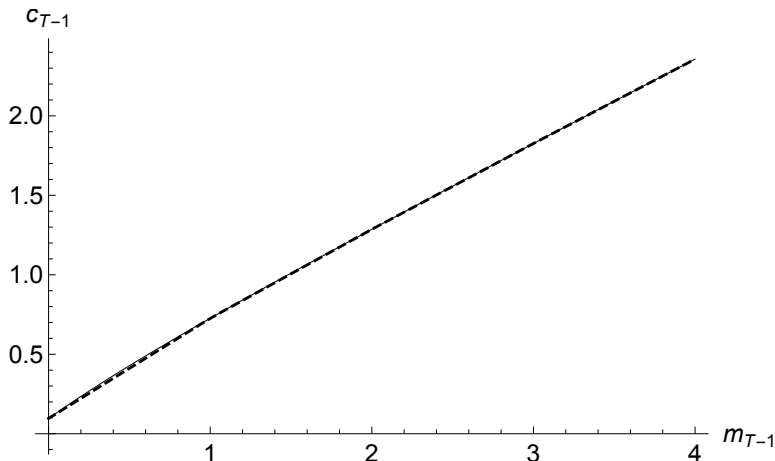
- ① Define a grid of points  $\vec{m}$  (indexed  $m[i]$ )
- ② Use numerical rootfinder to solve  $u'(c) = v'_t(m[i] - c)$ 
  - The  $c$  that solves this becomes  $c[i]$
- ③ Construct interpolating function  $\hat{c}$  by linear interpolation
  - 'Connect-the-dots'

# Trick: Interpolate a Consumption Rule

- ① Define a grid of points  $\vec{m}$  (indexed  $m[i]$ )
- ② Use numerical rootfinder to solve  $u'(c) = v'_t(m[i] - c)$ 
  - The  $c$  that solves this becomes  $c[i]$
- ③ Construct interpolating function  $\hat{c}$  by linear interpolation
  - 'Connect-the-dots'

# Trick: Interpolate a Consumption Rule

Example:  $\vec{m}_{T-1} = \{0., 1., 2., 3., 4.\}$  (solid is 'correct' soln)





# Problem: Numerical Rootfinding is *Slow*

Numerical search for values of  $c_{T-1}$  satisfying  $u'(c) = v'_t(m[i] - c)$  at, say, 6 gridpoints of  $\vec{m}_{T-1}$  may require hundreds or even thousands of evaluations of

$$v'_{T-1}(\overbrace{m_{T-1} - c_{T-1}}^{a_{T-1}}) = \beta_T \Gamma_T^{1-\rho} \left(\frac{1}{n}\right) \sum_{i=1}^n \frac{(\mathcal{R}_T a_{T-1} + \theta_i)^{1-\rho}}{1-\rho}$$

# Solution: The Method of Endogenous Gridpoints

- Define vector of *end-of-period* asset values  $\vec{a}$
- For each  $a[j]$  compute  $v'_t(a[j])$

Each of these  $v'_t[j]$  corresponds to a unique  $c[j]$  via FOC:

$$c[j]^{-\rho} = v'_t(a[j]) \quad (14)$$

$$c[j] = (v'_t(a[j]))^{-1/\rho} \quad (15)$$

But the DBC says

$$a_t = m_t - c_t \quad (16)$$

$$m[j] = a[j] + c[j] \quad (17)$$

So computing  $v'_t$  at a vector of  $\vec{a}$  values has produced for us the corresponding  $\vec{c}$  and  $\vec{m}$  values at virtually no cost!

From these we can interpolate as before to construct  $\hat{c}_t(m)$ .

# Solution: The Method of Endogenous Gridpoints

- Define vector of *end-of-period* asset values  $\vec{a}$ 
  - For each  $a[j]$  compute  $v'_t(a[j])$

Each of these  $v'_t[j]$  corresponds to a unique  $c[j]$  via FOC:

$$c[j]^{-\rho} = v'_t(a[j]) \quad (14)$$

$$c[j] = (v'_t(a[j]))^{-1/\rho} \quad (15)$$

But the DBC says

$$a_t = m_t - c_t \quad (16)$$

$$m[j] = a[j] + c[j] \quad (17)$$

So computing  $v'_t$  at a vector of  $\vec{a}$  values has produced for us the corresponding  $\vec{c}$  and  $\vec{m}$  values at virtually no cost!

From these we can interpolate as before to construct  $\hat{c}_t(m)$ .

# Solution: The Method of Endogenous Gridpoints

- Define vector of *end-of-period* asset values  $\vec{a}$
- For each  $a[j]$  compute  $v'_t(a[j])$

Each of these  $v'_t[j]$  corresponds to a unique  $c[j]$  via FOC:

$$c[j]^{-\rho} = v'_t(a[j]) \quad (14)$$

$$c[j] = (v'_t(a[j]))^{-1/\rho} \quad (15)$$

But the DBC says

$$a_t = m_t - c_t \quad (16)$$

$$m[j] = a[j] + c[j] \quad (17)$$

So computing  $v'_t$  at a vector of  $\vec{a}$  values has produced for us the corresponding  $\vec{c}$  and  $\vec{m}$  values at virtually no cost!

From these we can interpolate as before to construct  $\hat{c}_t(m)$ .

# Solution: The Method of Endogenous Gridpoints

- Define vector of *end-of-period* asset values  $\vec{a}$
- For each  $a[j]$  compute  $v'_t(a[j])$

Each of these  $v'_t[j]$  corresponds to a unique  $c[j]$  via FOC:

$$c[j]^{-\rho} = v'_t(a[j]) \quad (14)$$

$$c[j] = (v'_t(a[j]))^{-1/\rho} \quad (15)$$

But the DBC says

$$a_t = m_t - c_t \quad (16)$$

$$m[j] = a[j] + c[j] \quad (17)$$

So computing  $v'_t$  at a vector of  $\vec{a}$  values has produced for us the corresponding  $\vec{c}$  and  $\vec{m}$  values at virtually no cost!

From these we can interpolate as before to construct  $\hat{c}_t(m)$ .

# Solution: The Method of Endogenous Gridpoints

- Define vector of *end-of-period* asset values  $\vec{a}$
- For each  $a[j]$  compute  $v'_t(a[j])$

Each of these  $v'_t[j]$  corresponds to a unique  $c[j]$  via FOC:

$$c[j]^{-\rho} = v'_t(a[j]) \quad (14)$$

$$c[j] = (v'_t(a[j]))^{-1/\rho} \quad (15)$$

But the DBC says

$$a_t = m_t - c_t \quad (16)$$

$$m[j] = a[j] + c[j] \quad (17)$$

So computing  $v'_t$  at a vector of  $\vec{a}$  values has produced for us the corresponding  $\vec{c}$  and  $\vec{m}$  values at virtually no cost!

From these we can interpolate as before to construct  $\hat{c}_t(m)$ .

# Solution: The Method of Endogenous Gridpoints

- Define vector of *end-of-period* asset values  $\vec{a}$
- For each  $a[j]$  compute  $v'_t(a[j])$

Each of these  $v'_t[j]$  corresponds to a unique  $c[j]$  via FOC:

$$c[j]^{-\rho} = v'_t(a[j]) \quad (14)$$

$$c[j] = (v'_t(a[j]))^{-1/\rho} \quad (15)$$

But the DBC says

$$a_t = m_t - c_t \quad (16)$$

$$m[j] = a[j] + c[j] \quad (17)$$

So computing  $v'_t$  at a vector of  $\vec{a}$  values has produced for us the corresponding  $\vec{c}$  and  $\vec{m}$  values at virtually no cost!

From these we can interpolate as before to construct  $\hat{c}_t(m)$ .

# Solution: The Method of Endogenous Gridpoints

- Define vector of *end-of-period* asset values  $\vec{a}$
- For each  $a[j]$  compute  $v'_t(a[j])$

Each of these  $v'_t[j]$  corresponds to a unique  $c[j]$  via FOC:

$$c[j]^{-\rho} = v'_t(a[j]) \quad (14)$$

$$c[j] = (v'_t(a[j]))^{-1/\rho} \quad (15)$$

But the DBC says

$$a_t = m_t - c_t \quad (16)$$

$$m[j] = a[j] + c[j] \quad (17)$$

So computing  $v'_t$  at a vector of  $\vec{a}$  values has produced for us the corresponding  $\vec{c}$  and  $\vec{m}$  values at virtually no cost!

From these we can interpolate as before to construct  $\hat{c}_t(m)$ .



# Solution: The Method of Endogenous Gridpoints

- Define vector of *end-of-period* asset values  $\vec{a}$
- For each  $a[j]$  compute  $v'_t(a[j])$

Each of these  $v'_t[j]$  corresponds to a unique  $c[j]$  via FOC:

$$c[j]^{-\rho} = v'_t(a[j]) \quad (14)$$

$$c[j] = (v'_t(a[j]))^{-1/\rho} \quad (15)$$

But the DBC says

$$a_t = m_t - c_t \quad (16)$$

$$m[j] = a[j] + c[j] \quad (17)$$

So computing  $v'_t$  at a vector of  $\vec{a}$  values has produced for us the corresponding  $\vec{c}$  and  $\vec{m}$  values at virtually no cost!

From these we can interpolate as before to construct  $\hat{c}_t(m)$ .

# Why Directly Approximating $v_t$ is a Bad Idea

## Principles of Approximation

- Hard to approximate things that approach  $\infty$  for relevant  $m$ 
  - Not a prob for Rep Agent models: 'relevant'  $m$ 's are  $\approx$  SS
- Hard to approximate things that are highly nonlinear

# Why Directly Approximating $v_t$ is a Bad Idea

## Principles of Approximation

- Hard to approximate things that approach  $\infty$  for relevant  $m$ 
  - Not a prob for Rep Agent models: 'relevant'  $m$ 's are  $\approx$  SS
- Hard to approximate things that are highly nonlinear

# Why Directly Approximating $v_t$ is a Bad Idea

## Principles of Approximation

- Hard to approximate things that approach  $\infty$  for relevant  $m$ 
  - Not a prob for Rep Agent models: 'relevant'  $m$ 's are  $\approx$  SS
- Hard to approximate things that are highly nonlinear

# Approximate Something That Would Be Linear in PF Case

Perfect Foresight Theory:

$$c_t(m) = (m + h_t)\underline{\kappa}_t \quad (18)$$

for market resources  $m$  and end-of-period human wealth  $h$ .

This is why it's a good idea to approximate  $c_t$

Bonus: Easy to debug programs by setting  $\sigma^2 = 0$  and testing whether numerical solution matches analytical!

# Approximate Something That Would Be Linear in PF Case

Perfect Foresight Theory:

$$c_t(m) = (m + h_t)\underline{\kappa}_t \quad (18)$$

for market resources  $m$  and end-of-period human wealth  $h$ .

This is why it's a good idea to approximate  $c_t$

Bonus: Easy to debug programs by setting  $\sigma^2 = 0$  and testing whether numerical solution matches analytical!

# Approximate Something That Would Be Linear in PF Case

Perfect Foresight Theory:

$$c_t(m) = (m + h_t)\underline{\kappa}_t \quad (18)$$

for market resources  $m$  and end-of-period human wealth  $h$ .

This is why it's a good idea to approximate  $c_t$

Bonus: Easy to debug programs by setting  $\sigma^2 = 0$  and testing whether numerical solution matches analytical!

# But What if You *Need* the Value Function?

Perfect foresight value function:

$$\bar{v}_t(m_t) = u(\bar{c}_t)\mathbb{C}_t^T \quad (19)$$

$$= u(\bar{c}_t)\underline{\kappa}_t^{-1} \quad (20)$$

$$= u((\blacktriangle m_t + \blacktriangle h_t)\underline{\kappa}_t)\underline{\kappa}_t^{-1} \quad (21)$$

$$= u(\blacktriangle m_t + \blacktriangle h_t)\underline{\kappa}_t^{1-\rho}\underline{\kappa}_t^{-1} \quad (22)$$

$$= u(\blacktriangle m_t + \blacktriangle h_t)\underline{\kappa}_t^{-\rho} \quad (23)$$

where the second line uses the fact demonstrated in Carroll (2020) that  $\mathbb{C}_t = \kappa_t^{-1}$ .

This can be transformed as

$$\begin{aligned} \bar{\lambda}_t &\equiv ((1 - \rho)\bar{v}_t)^{1/(1-\rho)} \\ &= c_t(\mathbb{C}_t^T)^{1/(1-\rho)} \\ &= (\blacktriangle m_t + \blacktriangle h_t)\underline{\kappa}_t^{-\rho/(1-\rho)} \end{aligned}$$

which is linear.



# But What if You *Need* the Value Function?

Perfect foresight value function:

$$\bar{v}_t(m_t) = u(\bar{c}_t)\mathbb{C}_t^T \quad (19)$$

$$= u(\bar{c}_t)\underline{\kappa}_t^{-1} \quad (20)$$

$$= u((\blacktriangle m_t + \blacktriangle h_t)\underline{\kappa}_t)\underline{\kappa}_t^{-1} \quad (21)$$

$$= u(\blacktriangle m_t + \blacktriangle h_t)\underline{\kappa}_t^{1-\rho}\underline{\kappa}_t^{-1} \quad (22)$$

$$= u(\blacktriangle m_t + \blacktriangle h_t)\underline{\kappa}_t^{-\rho} \quad (23)$$

where the second line uses the fact demonstrated in Carroll (2020) that  $\mathbb{C}_t = \kappa_t^{-1}$ .

This can be transformed as

$$\begin{aligned} \bar{\lambda}_t &\equiv ((1 - \rho)\bar{v}_t)^{1/(1-\rho)} \\ &= c_t(\mathbb{C}_t^T)^{1/(1-\rho)} \\ &= (\blacktriangle m_t + \blacktriangle h_t)\underline{\kappa}_t^{-\rho/(1-\rho)} \end{aligned}$$

which is linear.

# Approximate Slope Too

Carroll (2020) shows that  $c_t^m$  exists everywhere.

Define *consumed* function and its derivative as

$$c_t(a) = (v'_t(a))^{-1/\rho} \quad (25)$$

$$c_t^a(a) = -(1/\rho) (v'_t(a))^{-1-1/\rho} v''_t(a) \quad (26)$$

and using chain rule it is easy to show that

$$c_t^m = c_t^a / (1 + c_t^a) \quad (27)$$

# Approximate Slope Too

Carroll (2020) shows that  $c_t^m$  exists everywhere.

Define *consumed* function and its derivative as

$$c_t(a) = (v'_t(a))^{-1/\rho} \quad (25)$$

$$c_t^a(a) = -(1/\rho) (v'_t(a))^{-1-1/\rho} v''_t(a) \quad (26)$$

and using chain rule it is easy to show that

$$c_t^m = c_t^a / (1 + c_t^a) \quad (27)$$

# Approximate Slope Too

Carroll (2020) shows that  $c_t^m$  exists everywhere.

Define *consumed* function and its derivative as

$$c_t(a) = (v'_t(a))^{-1/\rho} \quad (25)$$

$$c_t^a(a) = -(1/\rho) (v'_t(a))^{-1-1/\rho} v''_t(a) \quad (26)$$

and using chain rule it is easy to show that

$$c_t^m = c_t^a / (1 + c_t^a) \quad (27)$$

# To Implement: Modify Prior Procedures in Two Ways

- 1 Construct  $\vec{c}_t^m$  along with  $\vec{c}_t$  in EGM algorithm
- 2 Approximate  $c_t(m)$  using piecewise Hermite polynomial
  - Exact match to both level and derivative at set of points

# To Implement: Modify Prior Procedures in Two Ways

- 1 Construct  $\vec{c}_t^m$  along with  $\vec{c}_t$  in EGM algorithm
- 2 Approximate  $c_t(m)$  using piecewise Hermite polynomial
  - Exact match to both level and derivative at set of points

# To Implement: Modify Prior Procedures in Two Ways

- ① Construct  $\vec{c}_t^m$  along with  $\vec{c}_t$  in EGM algorithm
- ② Approximate  $c_t(m)$  using piecewise Hermite polynomial
  - Exact match to both level and derivative at set of points

# Problem: $\hat{c}$ Below Bottom $m$ Gridpoint and Extrapolation

Consider what happens as  $a_{T-1}$  approaches  $\underline{a}_{T-1} \equiv -\underline{\theta}\mathcal{R}_T^{-1}$ ,

$$\begin{aligned} \lim_{a \downarrow \underline{a}_{T-1}} v'_{T-1}(a) &= \lim_{a \downarrow \underline{a}_{T-1}} \beta R \Gamma_T^{-\rho} \left( \frac{1}{n} \right) \sum_{i=1}^n (a \mathcal{R}_T + \theta_i)^{-\rho} \\ &= \infty \end{aligned}$$

This means our lowest value in  $\vec{a}_{T-1}$  should be  $> \underline{a}_{T-1}$ .

Suppose we construct  $\hat{c}$  by linear interpolation:

$$\hat{c}_{T-1}(m) = \hat{c}_{T-1}(\vec{m}_{T-1}[1]) + \hat{c}'_{T-1}(\vec{m}_{T-1}[1])(m - \vec{m}_{T-1}[1])$$

True  $c$  is strictly concave  $\Rightarrow \exists m^- > \underline{m}_{T-1}$  for which

$$m^- - \hat{c}_{T-1}(m^-) < \underline{a}_{T-1}$$



# Solution: Hard-Code the Bottom Point

Theory says that

$$\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}(m) = 0 \quad (28)$$

$$\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}^m(m) = \bar{\kappa}_{T-1} \quad (29)$$

- ① Redefine  $\vec{a}$  *relative* to  $\underline{a}_{T-1}$
- ② Construct corresponding  $\vec{m}_{T-1}$  and  $\vec{c}_{T-1}$
- ③ Prepend  $\underline{m}_{T-1}$  to  $\vec{m}_{T-1}$
- ④ Prepend 0. to  $\vec{c}_{T-1}$
- ⑤ Prepend  $\bar{\kappa}_{T-1}$  to  $\vec{\kappa}_{T-1}$

then proceed as before.

# Solution: Hard-Code the Bottom Point

Theory says that

$$\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}(m) = 0 \quad (28)$$

$$\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}^m(m) = \bar{\kappa}_{T-1} \quad (29)$$

- ① Redefine  $\vec{a}$  *relative* to  $\underline{a}_{T-1}$
- ② Construct corresponding  $\vec{m}_{T-1}$  and  $\vec{c}_{T-1}$
- ③ Prepend  $\underline{m}_{T-1}$  to  $\vec{m}_{T-1}$
- ④ Prepend 0. to  $\vec{c}_{T-1}$
- ⑤ Prepend  $\bar{\kappa}_{T-1}$  to  $\vec{\kappa}_{T-1}$

then proceed as before.

# Solution: Hard-Code the Bottom Point

Theory says that

$$\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}(m) = 0 \quad (28)$$

$$\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}^m(m) = \bar{\kappa}_{T-1} \quad (29)$$

- 1 Redefine  $\vec{a}$  relative to  $\underline{a}_{T-1}$
- 2 Construct corresponding  $\vec{m}_{T-1}$  and  $\vec{c}_{T-1}$
- 3 Prepend  $\underline{m}_{T-1}$  to  $\vec{m}_{T-1}$
- 4 Prepend 0. to  $\vec{c}_{T-1}$
- 5 Prepend  $\bar{\kappa}_{T-1}$  to  $\vec{\kappa}_{T-1}$

then proceed as before.

# Solution: Hard-Code the Bottom Point

Theory says that

$$\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}(m) = 0 \quad (28)$$

$$\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}^m(m) = \bar{\kappa}_{T-1} \quad (29)$$

- ① Redefine  $\vec{a}$  *relative* to  $\underline{a}_{T-1}$
- ② Construct corresponding  $\vec{m}_{T-1}$  and  $\vec{c}_{T-1}$
- ③ Prepend  $\underline{m}_{T-1}$  to  $\vec{m}_{T-1}$
- ④ Prepend 0. to  $\vec{c}_{T-1}$
- ⑤ Prepend  $\bar{\kappa}_{T-1}$  to  $\vec{\kappa}_{T-1}$

then proceed as before.

# Solution: Hard-Code the Bottom Point

Theory says that

$$\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}(m) = 0 \quad (28)$$

$$\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}^m(m) = \bar{\kappa}_{T-1} \quad (29)$$

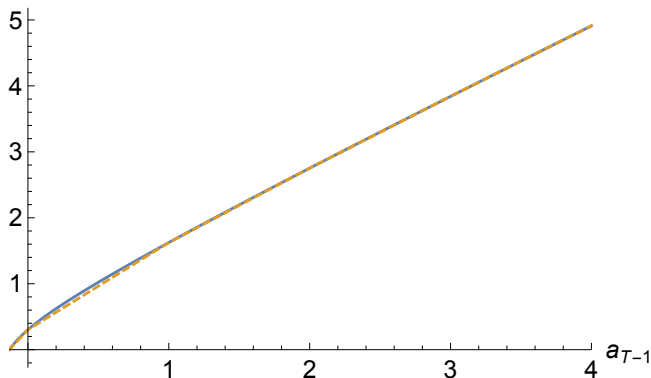
- ① Redefine  $\vec{a}$  *relative* to  $\underline{a}_{T-1}$
- ② Construct corresponding  $\vec{m}_{T-1}$  and  $\vec{c}_{T-1}$
- ③ Prepend  $\underline{m}_{T-1}$  to  $\vec{m}_{T-1}$
- ④ Prepend 0. to  $\vec{c}_{T-1}$
- ⑤ Prepend  $\bar{\kappa}_{T-1}$  to  $\vec{\kappa}_{T-1}$

then proceed as before.

# Trick: Improving the $a$ Grid

Grid Spacing: Uniform

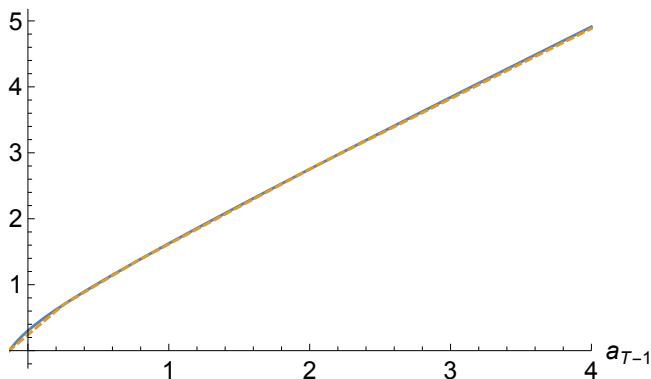
$$(u'_{T-1}(a_{T-1}))^{-1/\rho}, \dot{c}_{T-1}(a_{T-1})$$



# Trick: Improving the $a$ Grid

Grid Spacing: Same  $\{\underline{a}, \bar{a}\}$  But Triple Exponential  $e^{e^{\dots}}$  Growth

$$(u'_{T-1}(a_{T-1}))^{-1/\rho}, \dot{c}_{T-1}(a_{T-1})$$



# The Method of Moderation

- Further improves speed and accuracy of solution
- See my talk at the conference!



# The Method of Moderation

- Further improves speed and accuracy of solution
- See my talk at the conference!

# Imposing 'Artificial' Borrowing Constraints

$$\begin{aligned}
 v_{T-1}(m_{T-1}) &= \max_{c_{T-1}} u(c_{T-1}) + \mathbb{E}_{T-1}[\beta \Gamma_T^{1-\rho} v_T(m_T)] \\
 &\text{s.t.} \\
 a_{T-1} &= m_{T-1} - c_{T-1} \\
 m_T &= \mathcal{R}_T a_{T-1} + \theta_T \\
 a_{T-1} &\geq 0.
 \end{aligned}$$

Define  $\hat{c}_t^*$  as soln to unconstrained problem. Then

$$\hat{c}_{T-1}(m_{T-1}) = \min[m_{T-1}, \hat{c}_{T-1}^*(m_{T-1})]. \quad (30)$$

# Imposing 'Artificial' Borrowing Constraints

$$\begin{aligned}
 v_{T-1}(m_{T-1}) &= \max_{c_{T-1}} u(c_{T-1}) + \mathbb{E}_{T-1}[\beta \Gamma_T^{1-\rho} v_T(m_T)] \\
 &\text{s.t.} \\
 a_{T-1} &= m_{T-1} - c_{T-1} \\
 m_T &= \mathcal{R}_T a_{T-1} + \theta_T \\
 a_{T-1} &\geq 0.
 \end{aligned}$$

Define  $\hat{c}_t^*$  as soln to unconstrained problem. Then

$$\hat{c}_{T-1}(m_{T-1}) = \min[m_{T-1}, \hat{c}_{T-1}^*(m_{T-1})]. \quad (30)$$

# Imposing 'Artificial' Borrowing Constraints

Point where constraint makes transition from binding to not is

$$\begin{aligned} u'(m_{T-1}^{\#}) &= v'_{T-1}(0.) \\ m_{T-1}^{\#} &= (v'_{T-1}(0.))^{-1/\rho} \end{aligned}$$

Procedure is very easy:

- Add 0. as first point in  $\vec{a}$
- $\Rightarrow \vec{m}[1] = m_{T-1}^{\#}$
- Above  $m_{T-1}^{\#}$ ,  $\hat{c}_{T-1}(m)$  obtained as before
- Below  $m_{T-1}^{\#}$ ,  $\hat{c}_{T-1}(m) = m$

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ 🔍 ↺

---

$$m_{T-1}^\# = (\mathfrak{v}'_{T-1}(0.))^{-1/\rho}$$

- Add 0 as first point in  $\vec{p}$

# Imposing 'Artificial' Borrowing Constraints

Point where constraint makes transition from binding to not is

$$u'(m_{T-1}^{\#}) = v'_{T-1}(0.)$$

$$m_{T-1}^{\#} = (v'_{T-1}(0.))^{-1/\rho}$$

Procedure is very easy:

- Add 0. as first point in  $\vec{a}$
- $\Rightarrow \vec{m}[1] = m_{T-1}^{\#}$
- Above  $m_{T-1}^{\#}$ ,  $\hat{c}_{T-1}(m)$  obtained as before
- Below  $m_{T-1}^{\#}$ ,  $\hat{c}_{T-1}(m) = m$

# Imposing 'Artificial' Borrowing Constraints

Point where constraint makes transition from binding to not is

$$\begin{aligned} u'(m_{T-1}^{\#}) &= v'_{T-1}(0.) \\ m_{T-1}^{\#} &= (v'_{T-1}(0.))^{-1/\rho} \end{aligned}$$

Procedure is very easy:

- Add 0. as first point in  $\vec{a}$
- $\Rightarrow \vec{m}[1] = m_{T-1}^{\#}$
- Above  $m_{T-1}^{\#}$ ,  $\hat{c}_{T-1}(m)$  obtained as before
- Below  $m_{T-1}^{\#}$ ,  $\hat{c}_{T-1}(m) = m$



\_\_\_\_\_

1. *What is the purpose of this study?*

---

A set of small navigation icons typically found in Beamer presentations, including symbols for back, forward, search, and other slide controls.

---

## 1 Construct

$$\mathbf{c}_{t,i} = (\mathbf{v}'_t(a_{t,i}))^{-1/\rho}, \quad (31)$$

$$= \left( \beta \mathbb{E}_t \left[ \mathbf{R} \Gamma_{t+1}^{-\rho} (\dot{\mathbf{c}}_{t+1} (\mathcal{R}_{t+1} \mathbf{a}_{t,i} + \theta_{t+1}))^{-\rho} \right] \right)^{-1/\rho} (32)$$

- Call the result  $\vec{c}_t$  and generate the corresponding  $\vec{m}_t = \vec{c}_t + \vec{a}_t$
- Interpolate to create  $\hat{c}_t(m)$

# Recursion: Period $t$ Solution Given Period $t + 1$

## 1 Construct

$$\mathbf{c}_{t,i} = (\mathbf{v}'_t(\mathbf{a}_{t,i}))^{-1/\rho}, \quad (31)$$

$$= \left( \beta \mathbb{E}_t \left[ R \Gamma_{t+1}^{-\rho} (\hat{c}_{t+1}(\mathcal{R}_{t+1} \mathbf{a}_{t,i} + \theta_{t+1}))^{-\rho} \right] \right)^{-1/\rho} \quad (32)$$

- 2 Call the result  $\vec{c}_t$  and generate the corresponding  $\vec{m}_t = \vec{c}_t + \vec{a}_t$
- 3 Interpolate to create  $\hat{c}_t(m)$

# Consumption Rules $\hat{c}_{T-n}$ Converge

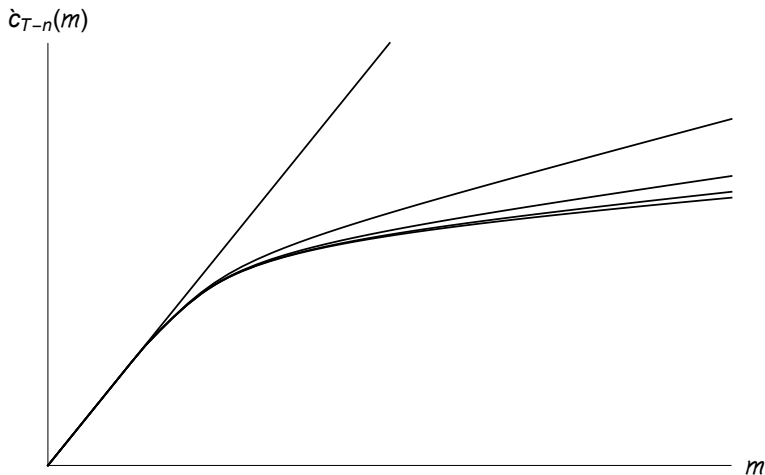


Figure: Converging  $\hat{c}_{T-n}(m)$  Functions for  $n = \{1, 5, 10, 15, 20\}$





\_\_\_\_\_

$$= \mathbf{R} + (\mathbf{R}_{t+1} - \mathbf{R})\varsigma_t \quad (34)$$

s.t.

$$0 \leq S_t \leq 1.$$



\_\_\_\_\_

$$= \mathbf{R} + (\mathbf{R}_{t+1} - \mathbf{R})\varsigma_t \quad (34)$$

s.t.

$$0 \leq \zeta_t \leq 1,$$

(continued)

# Portfolio Choice

The FOC with respect to  $c_t$  now yields an Euler equation

$$u'(c_t) = \mathbb{E}_t[\beta R_{t+1} u'(c_{t+1})]. \quad (35)$$

while the FOC with respect to the portfolio share yields

$$\begin{aligned} 0 &= \mathbb{E}_t[v'_{t+1}(m_{t+1})(\mathbf{R}_{t+1} - R)a_t] \\ &= a_t \mathbb{E}_t[u'(c_{t+1}(m_{t+1}))(\mathbf{R}_{t+1} - R)]. \end{aligned} \quad (36)$$

# Convergence

When the problem satisfies certain conditions (Carroll (2020)), it defines a ‘converged’ consumption rule with a ‘target’ ratio  $\check{m}$  that satisfies:

$$\mathbb{E}_t[m_{t+1}/m_t] = 1 \text{ if } m_t = \check{m} \quad (37)$$

Define the target  $m$  implied by the consumption rule  $c_t$  as  $\check{m}_t$ .

Then a plausible metric for convergence is to define some value  $\epsilon$  and to declare the solution to have converged when

$$|\check{m}_{t+1} - \check{m}_t| < \epsilon \quad (38)$$

---

# Convergence

When the problem satisfies certain conditions (Carroll (2020)), it defines a ‘converged’ consumption rule with a ‘target’ ratio  $\check{m}$  that satisfies:

$$\mathbb{E}_t[m_{t+1}/m_t] = 1 \text{ if } m_t = \check{m} \quad (37)$$

Define the target  $m$  implied by the consumption rule  $c_t$  as  $\check{m}_t$ .

Then a plausible metric for convergence is to define some value  $\epsilon$  and to declare the solution to have converged when

$$|\check{m}_{t+1} - \check{m}_t| < \epsilon \quad (38)$$

# Trick: Coarse then Fine $\theta$

- 1 Start with coarse grid for  $\theta$  (say, 3 points)
- 2 Solve to convergence; call period of convergence  $n$
- 3 Construct finer grid for  $\theta$  (say, 7 points)
- 4 Solve for period  $T - n - 1$  assuming  $\hat{c}_{T-n}$
- 5 Continue to convergence

# Trick: Coarse then Fine $\theta$

- 1 Start with coarse grid for  $\theta$  (say, 3 points)
- 2 Solve to convergence; call period of convergence  $n$
- 3 Construct finer grid for  $\theta$  (say, 7 points)
- 4 Solve for period  $T - n - 1$  assuming  $\hat{c}_{T-n}$
- 5 Continue to convergence



# Trick: Coarse then Fine $\theta$

- ① Start with coarse grid for  $\theta$  (say, 3 points)
- ② Solve to convergence; call period of convergence  $n$
- ③ Construct finer grid for  $\theta$  (say, 7 points)
- ④ Solve for period  $T - n - 1$  assuming  $\hat{c}_{T-n}$
- ⑤ Continue to convergence

# Trick: Coarse then Fine $\theta$

- ① Start with coarse grid for  $\theta$  (say, 3 points)
- ② Solve to convergence; call period of convergence  $n$
- ③ Construct finer grid for  $\theta$  (say, 7 points)
- ④ Solve for period  $T - n - 1$  assuming  $\hat{c}_{T-n}$
- ⑤ Continue to convergence

# Trick: Coarse then Fine $\theta$

- 1 Start with coarse grid for  $\theta$  (say, 3 points)
- 2 Solve to convergence; call period of convergence  $n$
- 3 Construct finer grid for  $\theta$  (say, 7 points)
- 4 Solve for period  $T - n - 1$  assuming  $\hat{c}_{T-n}$
- 5 Continue to convergence

Trick: Coarse then Fine  $\vec{a}_{T-1}$

- 1 Start with coarse grid for  $\vec{a}$  (say, 5 gridpoints)
- 2 Solve to convergence; call period of convergence  $n$
- 3 Construct finer grid for  $\vec{a}$  (say, 20 points)
- 4 Solve for period  $T - n - 1$  assuming  $\vec{c}_{T-n}$
- 5 Continue to convergence

# Trick: Coarse then Fine $\vec{a}_{T-1}$

- 1 Start with coarse grid for  $\vec{a}$  (say, 5 gridpoints)
- 2 Solve to convergence; call period of convergence  $n$
- 3 Construct finer grid for  $\vec{a}$  (say, 20 points)
- 4 Solve for period  $T - n - 1$  assuming  $\vec{c}_{T-n}$
- 5 Continue to convergence

# Trick: Coarse then Fine $\vec{a}_{T-1}$

- ① Start with coarse grid for  $\vec{a}$  (say, 5 gridpoints)
- ② Solve to convergence; call period of convergence  $n$
- ③ Construct finer grid for  $\vec{a}$  (say, 20 points)
- ④ Solve for period  $T - n - 1$  assuming  $\vec{c}_{T-n}$
- ⑤ Continue to convergence

# Trick: Coarse then Fine $\vec{a}_{T-1}$

- ① Start with coarse grid for  $\vec{a}$  (say, 5 gridpoints)
- ② Solve to convergence; call period of convergence  $n$
- ③ Construct finer grid for  $\vec{a}$  (say, 20 points)
- ④ Solve for period  $T - n - 1$  assuming  $\vec{c}_{T-n}$
- ⑤ Continue to convergence

# Trick: Coarse then Fine $\vec{a}_{T-1}$

- ① Start with coarse grid for  $\vec{a}$  (say, 5 gridpoints)
- ② Solve to convergence; call period of convergence  $n$
- ③ Construct finer grid for  $\vec{a}$  (say, 20 points)
- ④ Solve for period  $T - n - 1$  assuming  $\vec{c}_{T-n}$
- ⑤ Continue to convergence



---

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ 🔍 ↺

# Details follow Cagetti (2003)

- Parameterization of Uncertainty
- Probability of Death
- Demographic Adjustments to  $\beta$

# Details follow Cagetti (2003)

- Parameterization of Uncertainty
- Probability of Death
- Demographic Adjustments to  $\beta$

# Details follow Cagetti (2003)

- Parameterization of Uncertainty
- Probability of Death
- Demographic Adjustments to  $\beta$

# Empirical Wealth Profiles

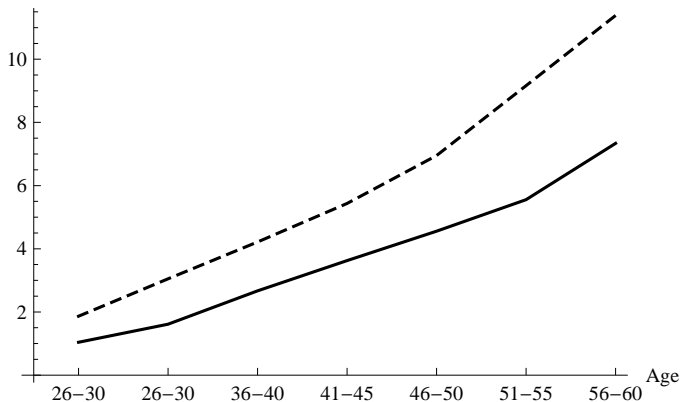


Figure:  $m$  from SCF (means (dashed) and medians (solid))

# Simulated Moments

Given a set of parameter values  $\{\rho, \Xi\}$ :

- Start at age 25 with empirical  $m$  data
- Draw shocks using calibrated  $\sigma_\psi^2, \sigma_\theta^2$
- Consume according to solved  $c_t$

$\Rightarrow m$  distribution by age

# Simulated Moments

Given a set of parameter values  $\{\rho, \Xi\}$ :

- Start at age 25 with empirical  $m$  data
- Draw shocks using calibrated  $\sigma_\psi^2, \sigma_\theta^2$
- Consume according to solved  $c_t$

$\Rightarrow m$  distribution by age





# Simulated Moments

Given a set of parameter values  $\{\rho, \Xi\}$ :

- Start at age 25 with empirical  $m$  data
- Draw shocks using calibrated  $\sigma_\psi^2, \sigma_\theta^2$
- Consume according to solved  $c_t$

⇒  $m$  distribution by age

# Simulated Moments

Given a set of parameter values  $\{\rho, \Xi\}$ :

- Start at age 25 with empirical  $m$  data
- Draw shocks using calibrated  $\sigma_\psi^2, \sigma_\theta^2$
- Consume according to solved  $c_t$

$\Rightarrow m$  distribution by age

```

GapEmpiricalSimulatedMedians[ $\rho, \triangleright$ ]:=
[  ConstructcFuncLife[ $\rho, \triangleright$ ];
    Simulate;
     $\sum_i^N \omega_i |\varsigma_i^\tau - \mathbf{s}^\tau(\xi)|$ 
];

```

---

$$\xi = \{\rho, \sqsupset\} \quad (39)$$

solve

$$\min_{\xi} \sum_i^N \omega_i |\varsigma_i^T - \mathbf{s}^T(\xi)| \quad (40)$$

## Bootstrap Standard Errors (Horowitz (2001))

## Yields estimates of

### Table: Estimation Results

| $\rho$ | $\beta$ |
|--------|---------|
| 4.68   | 1.00    |
| (0.13) | (0.00)  |

\_\_\_\_\_

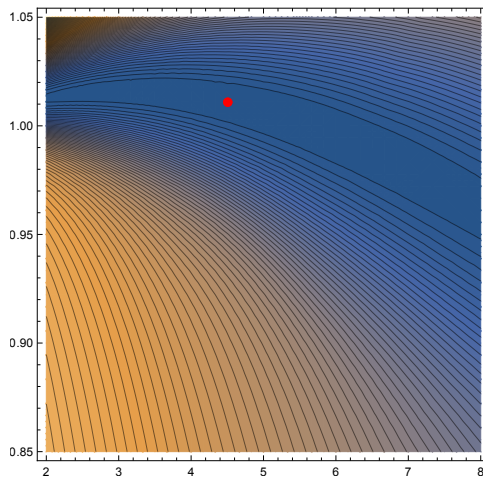


Figure: Point Estimate and Height of Minimized Function

# References I

CAGETTI, MARCO (2003): "Wealth Accumulation Over the Life Cycle and Precautionary Savings," Journal of Business and Economic Statistics, 21(3), 339–353.

CARROLL, CHRISTOPHER D. (2020): "Theoretical Foundations of Buffer Stock Saving," Forthcoming, Quantitative Economics, Available at <http://econ.jhu.edu/people/ccarroll/papers/BufferStockTheory>.

HOROWITZ, JOEL L. (2001): "The Bootstrap," in Handbook of Econometrics, ed. by James J. Heckman, and Edward Leamer, vol. 5. Elsevier/North Holland.