

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

Christopher Carroll¹

¹Johns 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

[illegible]

Bellman Equation

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

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

p — permanent labor income

Trick: Normalize the Problem

$$\begin{aligned}
 v_t(m_t) &= \max_{c_t} u(c_t) + \beta \mathbb{E}_t[\mathcal{G}_{t+1}^{1-\rho} v_{t+1}(m_{t+1})] \\
 &\text{s.t.} \\
 a_t &= m_t - c_t \\
 m_{t+1} &= \underbrace{(R/\mathcal{G}_{t+1})}_{\equiv \mathcal{R}_{t+1}} a_t + \boldsymbol{\theta}_{t+1}.
 \end{aligned} \tag{5}$$

where nonbold variables are bold ones normalized by \boldsymbol{p} :

$$m_t = m_t / \boldsymbol{p}_t \tag{6}$$

Yields $c_t(m)$ from which we can obtain

$$c_t(m_t, \boldsymbol{p}_t) = c_t(m_t / \boldsymbol{p}_t) \boldsymbol{p}_t \tag{7}$$

- 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: View Everything from End of Period

Define

$$v_t(a_t) = \mathbb{E}_t[\beta \mathcal{G}_{t+1}^{1-\rho} v_{t+1}(\mathcal{R}_{t+1} a_t + \theta_{t+1})] \quad (8)$$

so

$$v_t(m_t) = \max_{c_t} u(c_t) + v_t(m_t - c_t) \quad (9)$$

with FOC

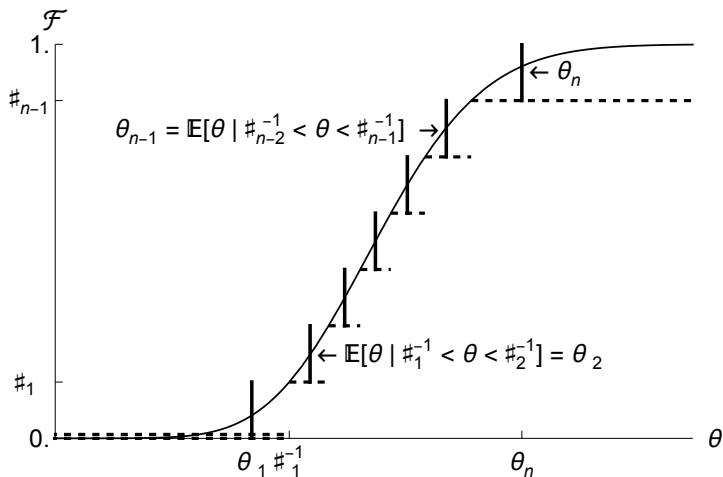
$$u'(c_t) = v'_t(m_t - c_t). \quad (10)$$

and Envelope relation

$$u'(c_t) = v'_t(m_t) \quad (11)$$

Trick: Discretize the Risks

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



$$\mathbf{v}'_t(a_t) = \beta \mathcal{R} \mathcal{G}_{t+1}^{-\rho} \left(\frac{1}{n} \right) \sum_{i=1}^n u'(\mathbf{c}_{t+1}(\mathcal{R}_{t+1} a_t + \boldsymbol{\theta}_i)) \quad (12)$$

100

$$\mathbf{v}'_t(a_t) = \beta \mathcal{R} \mathcal{G}_{t+1}^{-\rho} \left(\frac{1}{n} \right) \sum_{i=1}^n u'(\mathbf{c}_{t+1}(\mathcal{R}_{t+1} a_t + \boldsymbol{\theta}_i)) \quad (12)$$

So for any particular m_{T-1} the corresponding c_{T-1} can be found using the FOC:

$$u'(c_t) = v'_t(m_t - c_t). \quad (13)$$

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

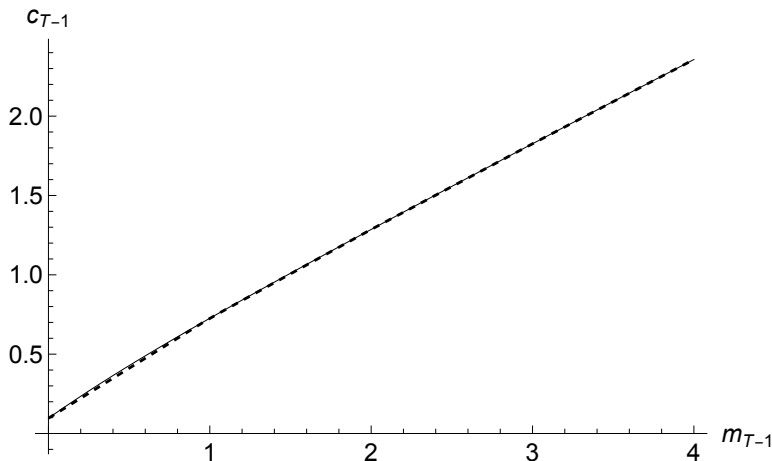
- ① Define a grid of points \vec{m} (indexed $m[i]$)
- ② Use numerical rootfinder to solve $u'(c) = v'_t(m[i] - c)$
 - The c that solves this becomes $c[i]$
- ③ Construct interpolating function \hat{c} by linear interpolation
 - 'Connect-the-dots'

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

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

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:

$$\begin{aligned} c[j]^{-\rho} &= v'_t(a[j]) \\ c[j] &= (v'_t(a[j]))^{-1/\rho} \end{aligned} \quad (14)$$

But the DBC says

$$\begin{aligned} a_t &= m_t - c_t \\ m[j] &= a[j] + c[j] \end{aligned} \quad (15)$$

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:

$$\begin{aligned} c[j]^{-\rho} &= v'_t(a[j]) \\ c[j] &= (v'_t(a[j]))^{-1/\rho} \end{aligned} \quad (14)$$

But the DBC says

$$\begin{aligned} a_t &= m_t - c_t \\ m[j] &= a[j] + c[j] \end{aligned} \quad (15)$$

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:

$$\begin{aligned} c[j]^{-\rho} &= v'_t(a[j]) \\ c[j] &= (v'_t(a[j]))^{-1/\rho} \end{aligned} \tag{14}$$

But the DBC says

$$\begin{aligned} a_t &= m_t - c_t \\ m[j] &= a[j] + c[j] \end{aligned} \tag{15}$$

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:

$$\begin{aligned} c[j]^{-\rho} &= v'_t(a[j]) \\ c[j] &= (v'_t(a[j]))^{-1/\rho} \end{aligned} \tag{14}$$

But the DBC says

$$\begin{aligned} a_t &= m_t - c_t \\ m[j] &= a[j] + c[j] \end{aligned} \tag{15}$$

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:

$$\begin{aligned} c[j]^{-\rho} &= v'_t(a[j]) \\ c[j] &= (v'_t(a[j]))^{-1/\rho} \end{aligned} \tag{14}$$

But the DBC says

$$\begin{aligned} a_t &= m_t - c_t \\ m[j] &= a[j] + c[j] \end{aligned} \tag{15}$$

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:

$$\begin{aligned} c[j]^{-\rho} &= v'_t(a[j]) \\ c[j] &= (v'_t(a[j]))^{-1/\rho} \end{aligned} \tag{14}$$

But the DBC says

$$\begin{aligned} a_t &= m_t - c_t \\ m[j] &= a[j] + c[j] \end{aligned} \tag{15}$$

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:

$$\begin{aligned} c[j]^{-\rho} &= v'_t(a[j]) \\ c[j] &= (v'_t(a[j]))^{-1/\rho} \end{aligned} \tag{14}$$

But the DBC says

$$\begin{aligned} a_t &= m_t - c_t \\ m[j] &= a[j] + c[j] \end{aligned} \tag{15}$$

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:

$$\begin{aligned} c[j]^{-\rho} &= v'_t(a[j]) \\ c[j] &= (v'_t(a[j]))^{-1/\rho} \end{aligned} \tag{14}$$

But the DBC says

$$\begin{aligned} a_t &= m_t - c_t \\ m[j] &= a[j] + c[j] \end{aligned} \tag{15}$$

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 ∞ 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 ∞ 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 ∞ 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:

$$c_t(m) = (m + h_t)\underline{\kappa}_t \quad (16)$$

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:

$$c_t(m) = (m + h_t)\underline{\kappa}_t \quad (16)$$

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:

$$c_t(m) = (m + h_t)\underline{\kappa}_t \quad (16)$$

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:

$$\begin{aligned}
 \bar{v}_t(m_t) &= u(\bar{c}_t)\mathbb{C}_t^T \\
 &= u(\bar{c}_t)\underline{\kappa}_t^{-1} \\
 &= u((\blacktriangle m_t + \blacktriangle h_t)\underline{\kappa}_t)\underline{\kappa}_t^{-1} \\
 &= u(\blacktriangle m_t + \blacktriangle h_t)\underline{\kappa}_t^{1-\rho}\underline{\kappa}_t^{-1} \\
 &= u(\blacktriangle m_t + \blacktriangle h_t)\underline{\kappa}_t^{-\rho}
 \end{aligned} \tag{17}$$

where the second line uses the fact demonstrated in Carroll (2023) 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:

$$\begin{aligned}
 \bar{v}_t(m_t) &= u(\bar{c}_t) \mathbb{C}_t^T \\
 &= u(\bar{c}_t) \underline{\kappa}_t^{-1} \\
 &= u((\blacktriangle m_t + \blacktriangle h_t) \underline{\kappa}_t) \underline{\kappa}_t^{-1} \\
 &= u(\blacktriangle m_t + \blacktriangle h_t) \underline{\kappa}_t^{1-\rho} \underline{\kappa}_t^{-1} \\
 &= u(\blacktriangle m_t + \blacktriangle h_t) \underline{\kappa}_t^{-\rho}
 \end{aligned} \tag{17}$$

where the second line uses the fact demonstrated in Carroll (2023) 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 (2023) shows that c_t^m exists everywhere.

Define *consumed* function and its derivative as

$$\begin{aligned} c_t(a) &= (v'_t(a))^{-1/\rho} \\ c_t^a(a) &= -(1/\rho) (v'_t(a))^{-1-1/\rho} v''_t(a) \end{aligned} \tag{19}$$

and using chain rule it is easy to show that

$$c_t^m = c_t^a / (1 + c_t^a) \tag{20}$$

Approximate Slope Too

Carroll (2023) shows that c_t^m exists everywhere.

Define *consumed* function and its derivative as

$$\begin{aligned} c_t(a) &= (v'_t(a))^{-1/\rho} \\ c_t^a(a) &= -(1/\rho) (v'_t(a))^{-1-1/\rho} v''_t(a) \end{aligned} \tag{19}$$

and using chain rule it is easy to show that

$$c_t^m = c_t^a / (1 + c_t^a) \tag{20}$$

Approximate Slope Too

Carroll (2023) shows that c_t^m exists everywhere.

Define *consumed* function and its derivative as

$$\begin{aligned} c_t(a) &= (v'_t(a))^{-1/\rho} \\ c_t^a(a) &= -(1/\rho) (v'_t(a))^{-1-1/\rho} v''_t(a) \end{aligned} \tag{19}$$

and using chain rule it is easy to show that

$$c_t^m = c_t^a / (1 + c_t^a) \tag{20}$$

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 $\underline{a}_{T-1} \equiv -\underline{\theta}\mathcal{R}_T^{-1}$,

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

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

$$\begin{aligned} \lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}(m) &= 0 \\ \lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}^m(m) &= \bar{\kappa}_{T-1} \end{aligned} \tag{21}$$

- 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

$$\begin{aligned} \lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}(m) &= 0 \\ \lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}^m(m) &= \bar{\kappa}_{T-1} \end{aligned} \tag{21}$$

- 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

$$\begin{aligned}\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}(m) &= 0 \\ \lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}^m(m) &= \bar{\kappa}_{T-1}\end{aligned}\tag{21}$$

- 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

$$\begin{aligned}\lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}(m) &= 0 \\ \lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}^m(m) &= \bar{\kappa}_{T-1}\end{aligned}\tag{21}$$

- 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

$$\begin{aligned} \lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}(m) &= 0 \\ \lim_{m \downarrow \underline{m}_{T-1}} c_{T-1}^m(m) &= \bar{\kappa}_{T-1} \end{aligned} \tag{21}$$

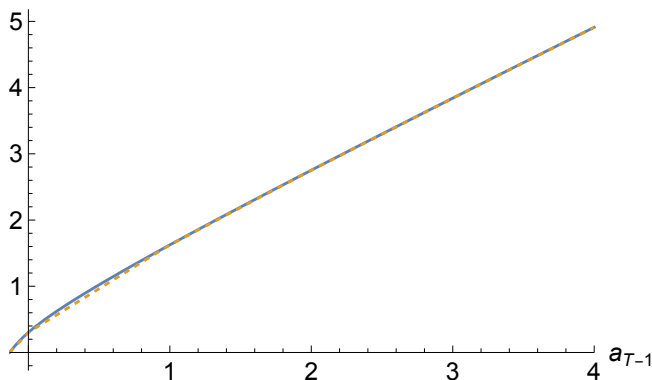
- ① Redefine \vec{a} *relative* to \underline{a}_{T-1}
- ② Construct corresponding \vec{m}_{T-1} and \vec{c}_{T-1}
- ③ Prepend \underline{m}_{T-1} to \vec{m}_{T-1}
- ④ Prepend 0. to \vec{c}_{T-1}
- ⑤ Prepend $\bar{\kappa}_{T-1}$ to $\vec{\kappa}_{T-1}$

then proceed as before.

Trick: Improving the a Grid

Grid Spacing: Uniform

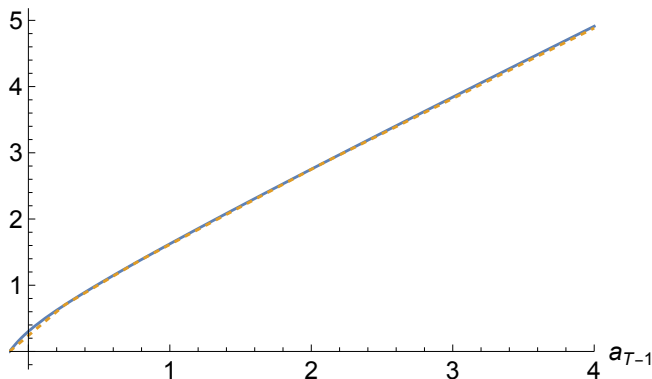
$$(u'_{T-1}(a_{T-1}))^{-1/\rho}, \dot{c}_{T-1}(a_{T-1})$$



Trick: Improving the a Grid

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

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



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

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

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

$$\hat{c}_{T-1}(m_{T-1}) = \min[m_{T-1}, \hat{c}_{T-1}^*(m_{T-1})]. \quad (22)$$

Imposing 'Artificial' Borrowing Constraints

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

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

$$\check{c}_{T-1}(m_{T-1}) = \min[m_{T-1}, \check{c}_{T-1}^*(m_{T-1})]. \quad (22)$$

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

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

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

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

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

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

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 $m_{T-1}^{\#}$, $\hat{c}_{T-1}(m) = m$

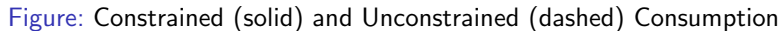
Imposing 'Artificial' Borrowing Constraints

Point where constraint makes transition from binding to not is

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

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 $m_{T-1}^{\#}$, $\hat{c}_{T-1}(m) = m$



③ Interpolate to create $\hat{c}_t(m)$

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

1 Construct

$$\begin{aligned} c_{\bar{t},i} &= (v'_{\bar{t}}(a_{t,i}))^{-1/\rho}, \\ &= \left(\beta \mathbb{E}_t \left[R \mathcal{G}_{t+1}^{-\rho} (\dot{c}_{t+1}(\mathcal{R}_{t+1} a_{t,i} + \theta_{t+1}))^{-\rho} \right] \right)^{-1/\rho}, \end{aligned} \quad (23)$$

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

Consumption Rules $\dot{c}_{T-n}(m)$ Converge

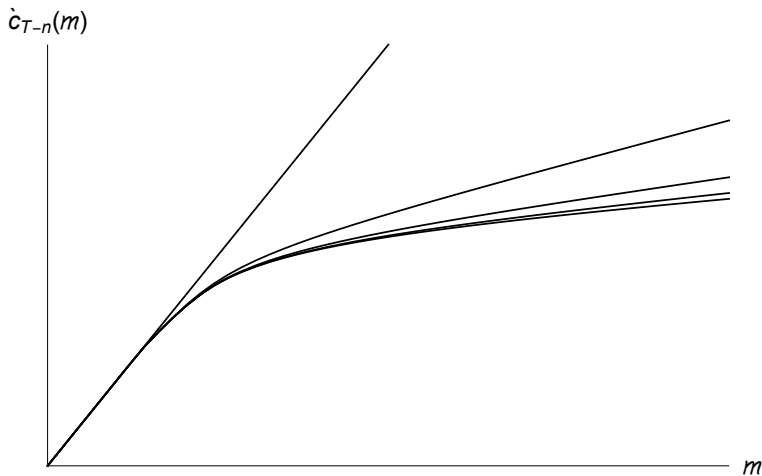


Figure: Converging $\dot{c}_{T-n}(m)$ Functions for $n = \{1, 5, 10, 15, 20\}$

Portfolio Choice

Now the consumer has a choice between a risky and a safe asset.
The portfolio return is

$$\begin{aligned}\mathbb{R}_{t+1} &= R(1 - \varsigma_t) + \mathbf{R}_{t+1}\varsigma_t \\ &= R + (\mathbf{R}_{t+1} - R)\varsigma_t\end{aligned}\quad (24)$$

s.t.

$$\mathbb{R}_{t+1} = R + (\mathbf{R}_{t+1} - R)\zeta_t$$

$$m_{t+1} = (m_t - c_t)\mathbb{R}_{t+1} + \theta_{t+1}$$

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

Portfolio Choice

Now the consumer has a choice between a risky and a safe asset.
The portfolio return is

$$\begin{aligned}\mathbb{R}_{t+1} &= R(1 - \varsigma_t) + \mathbf{R}_{t+1}\varsigma_t \\ &= R + (\mathbf{R}_{t+1} - R)\varsigma_t\end{aligned}\tag{24}$$

so (setting $\mathcal{G} = 1$) the maximization problem is

$$v_t(m_t) = \max_{\{c_t, \varsigma_t\}} u(c_t) + \beta \mathbb{E}_t[v_{t+1}(m_{t+1})]$$

s.t.

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

$$m_{t+1} = (m_t - c_t)\mathbb{R}_{t+1} + \theta_{t+1}$$

$$0 \leq \varsigma_t \leq 1,$$

$$u'(c_t) = \mathbb{E}_t[\beta R_{t+1} u'(c_{t+1})]. \quad (25)$$

while the FOC with respect to the portfolio share yields

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

Convergence

When the problem satisfies certain conditions (Carroll (2023)), it defines a ‘converged’ consumption rule with a ‘target’ ratio \check{m} that satisfies:

$$\mathbb{E}_t[m_{t+1}/m_t] = 1 \text{ if } m_t = \check{m} \quad (26)$$

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 ϵ and to declare the solution to have converged when

$$|\check{m}_{t+1} - \check{m}_t| < \epsilon \quad (27)$$

Convergence

When the problem satisfies certain conditions (Carroll (2023)), it defines a ‘converged’ consumption rule with a ‘target’ ratio \check{m} that satisfies:

$$\mathbb{E}_t[m_{t+1}/m_t] = 1 \text{ if } m_t = \check{m} \quad (26)$$

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 ϵ and to declare the solution to have converged when

$$|\check{m}_{t+1} - \check{m}_t| < \epsilon \quad (27)$$

Convergence

When the problem satisfies certain conditions (Carroll (2023)), it defines a ‘converged’ consumption rule with a ‘target’ ratio \check{m} that satisfies:

$$\mathbb{E}_t[m_{t+1}/m_t] = 1 \text{ if } m_t = \check{m} \quad (26)$$

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 ϵ and to declare the solution to have converged when

$$|\check{m}_{t+1} - \check{m}_t| < \epsilon \quad (27)$$

Trick: Coarse then Fine θ

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

Trick: Coarse then Fine θ

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

Trick: Coarse then Fine θ

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

Trick: Coarse then Fine θ

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

Trick: Coarse then Fine θ

- 1 Start with coarse grid for θ (say, 3 points)
- 2 Solve to convergence; call period of convergence n
- 3 Construct finer grid for θ (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 β

Details follow Cagetti (2003)

- Parameterization of Uncertainty
- Probability of Death
- Demographic Adjustments to β

Details follow Cagetti (2003)

- Parameterization of Uncertainty
- Probability of Death
- Demographic Adjustments to β

Empirical Wealth Profiles

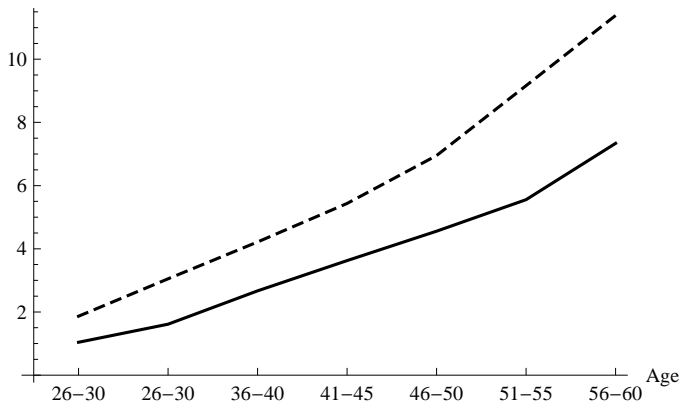


Figure: m from SCF (means (dashed) and medians (solid))

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

⇒ m distribution by age

```

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

];

```

$$\xi = \{\rho, \sqsupset\} \quad (28)$$

solve

$$\min_{\xi} \sum_i^N \omega_i |\varsigma_i^{\tau} - \mathbf{s}^{\tau}(\xi)| \quad (29)$$

Bootstrap Standard Errors (Horowitz (2001))

Yields estimates of

Table: Estimation Results

ρ	β
4.68	1.00
(0.13)	(0.00)

Contour Plot

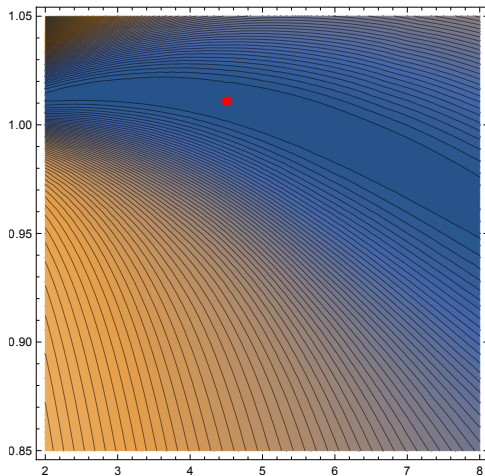


Figure: Point Estimate and Height of Minimized Function

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. (2023): "Theoretical Foundations of Buffer Stock Saving," Submitted.

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