Representing Dynamic Models in HARK

Matthew N. White, University of Delaware

September 28, 2016

Dynamic Models in HARK

- Make concepts from previous section more concrete
- What economic models are already in HARK?
- How do those models fit in the HARK framework?
- What models will be in HARK in the near future?

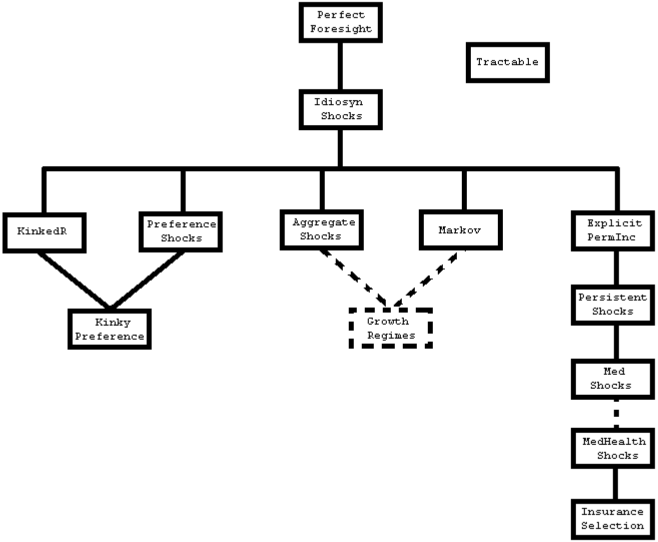
Object-Oriented Solution Methods

- Models in HARK build up from each other
- "Parent" models are special cases of "child" models

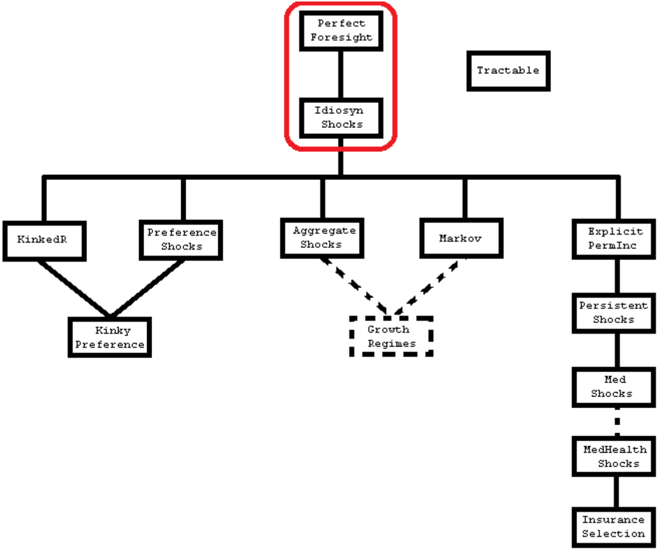
Object-Oriented Solvers

  - Solvers in HARK are objects that act (a lot) like functions
  - Each model specifies a new class for its solver
  - Inherit solution method from parent solver...
  - ...and add or change its methods / subroutines.

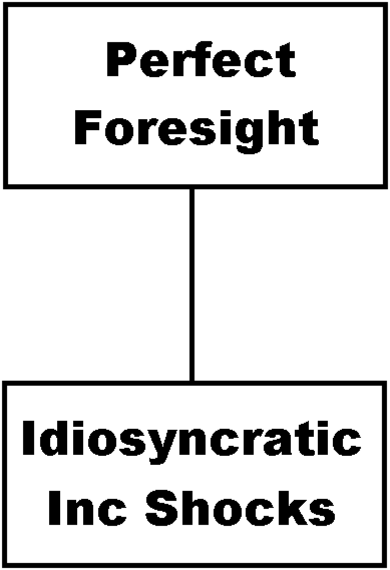
Consumption-Saving Model Tree



Consumption-Saving Model Tree



Consumption-Saving Model Tree



Perfect Foresight

Consider simplest possible consumption-saving model:

- CRRA utility
- Geometric discounting of future utility
- Exogenous interest rate
- Income growth and survival probability might vary by age
- No income risk

Perfect Foresight

$$
\begin{aligned}
V_t(M_t) &= \max_{C_t} u(C_t) + \beta \cancel{D}_{t+1} \mathbb{E}[V_{t+1}(M_{t+1})], \\
A_t &= M_t - C_t, \\
M_{t+1} &= RA_t + Y_{t+1}, \\
Y_{t+1} &= \Gamma_{t+1} Y_t, \\
u(C) &= \frac{C^{1-\rho}}{1-\rho}.
\end{aligned}
$$

Perfect Foresight, Normalized

$$
\begin{aligned}
v_t(m_t) &= \max_{c_t} u(c_t) + \beta \cancel{D}_{t+1} \mathbb{E}[v_{t+1}(m_{t+1})], \\
a_t &= m_t - c_t, \\
m_{t+1} &= (R/\Gamma_{t+1}) a_t + 1, \\
u(c) &= \frac{c^{1-\rho}}{1-\rho}.
\end{aligned}
$$

Perfect Foresight in HARK

Consumers are instances of `PerfForesightConsumerType`

- `time_inv = ['CRRA','DiscFac','Rfree']`
- `time_vary = ['PermGroFac','LivPrb']`

Solving one period makes an instance of `ConsPerfForesightSolver`

- `defUtilityFuncs`: Defines utility function, derivatives, inverses
- `makePFcFunc`: Linear perfect foresight consumption function
- `makevFuncs`: Value and marginal value functions

Solution is an instance of `ConsumerSolution`

Idiosyncratic Income Shocks

More interesting model with risk:

- Income subject to idiosyncratic risks
- Two shocks: fully transitory, fully permanent
- Maybe an exogenous borrowing constraint
- No closed form solution, use numeric methods

Idiosyncratic Income Shocks

$$
\begin{aligned}
v(m_t) &= \max_{c_t} u(c_t) + \beta \cancel{D}_{t+1} \mathbb{E}[v_{t+1}(m_{t+1})], \\
a_t &= m_t - c_t, \\
a_t &\geq \underline{a}, \\
m_{t+1} &= R/(\Gamma_{t+1}\psi_{t+1})a_t + \theta_{t+1}, \\
\theta_t \sim F_{\theta t}, & \qquad \psi_t \sim F_{\psi t}, \quad \mathbb{E}[F_{\psi t}] = 1, \\
u(c) &= \frac{c^{1-\rho}}{1-\rho}.
\end{aligned}
$$

Idiosyncratic Income Shocks

Consumers are instances of `IndShockConsumerType`

- `time_inv += ['BoroCnstArt','vFuncBool','CubicBool','aXtraGrid']`
- `time_vary += ['IncomeDstn']`

Distributions in HARK are discrete approximations

- `IncomeDstn` is a list of three `array`s
- First element is `array` of discrete probabilities
- Second element is `array` of permanent shock values
- Third element is `array` of transitory shock values

Constructed Solver Inputs

Constructing `aXtraGrid`:

- `aXtraMin = 0.0001,     aXtraMax = 80.0`
- `aXtraCount = 48,     aXtraNestFac = 3`

Constructing `IncomeDstn`:

- `PermShkStd = [0.10,0.13,0.15]`
- `TranShkStd = [0.10,0.09,0.08]`
- `PermShkCount = 9,     TranShkCount = 9`
- `UnempPrb = 0.05,     IncUnemp = 0.0`
- `T_retire = 0`

Solving Idiosyncratic Shocks

`ConsIndShockSolver` inherits from `ConsPefectForesightSolver`

- `setAndUpdateValues`: Calculate relevant constants from primitives: worst shocks, min and max MPC, human wealth, etc
- `defBoroCnst`: Find `cFunc` when borrowing constraint binds

Solving Idiosyncratic Shocks

`ConsIndShockSolver` inherits from `ConsPefectForesightSolver`

- `setAndUpdateValues`: Calculate relevant constants from primitives: worst shocks, min and max MPC, human wealth, etc
- `defBoroCnst`: Find `cFunc` when borrowing constraint binds
- `prepareToCalcEndOfPrdvP`: Generate array of m_{t+1} values, etc
- `calcEndOfPrdvP`: Calculate end-of-period marginal value on $\{a_t\}$

Solving Idiosyncratic Shocks

`ConsIndShockSolver` inherits from `ConsPefectForesightSolver`

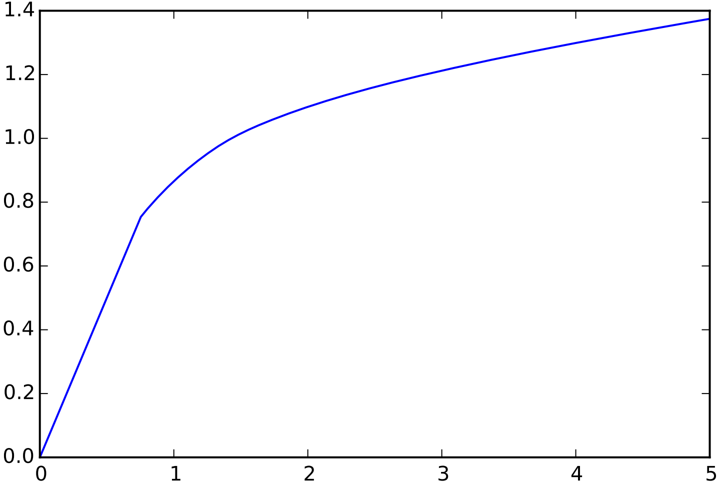
- `setAndUpdateValues`: Calculate relevant constants from primitives: worst shocks, min and max MPC, human wealth, etc
- `defBoroCnst`: Find `cFunc` when borrowing constraint binds
- `prepareToCalcEndOfPrdvP`: Generate array of m_{t+1} values, etc
- `calcEndOfPrdvP`: Calculate end-of-period marginal value on $\{a_t\}$
- `getPointsForInterpolation`: Calculate $\{m_t\}$ and $\{c_t\}$ points
- `usePointsForInterpolation`: Construct interpolated `cFunc`

Solving Idiosyncratic Shocks

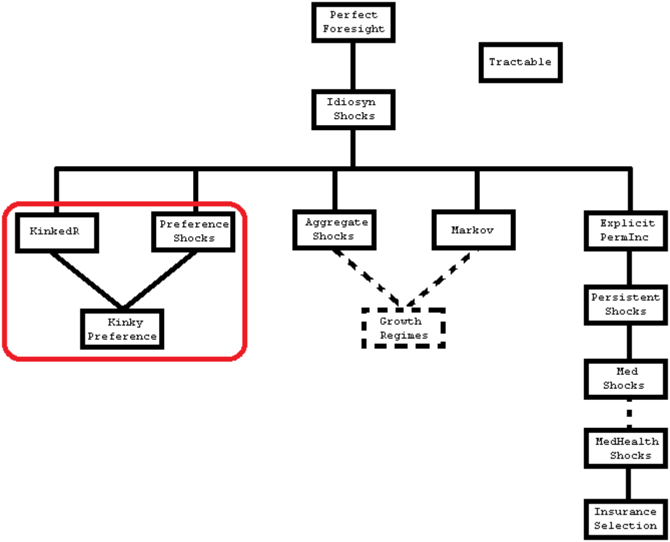
`ConsIndShockSolver` inherits from `ConsPefectForesightSolver`

- `setAndUpdateValues`: Calculate relevant constants from primitives: worst shocks, min and max MPC, human wealth, etc
- `defBoroCnst`: Find `cFunc` when borrowing constraint binds
- `prepareToCalcEndOfPrdvP`: Generate array of m_{t+1} values, etc
- `calcEndOfPrdvP`: Calculate end-of-period marginal value on $\{a_t\}$
- `getPointsForInterpolation`: Calculate $\{m_t\}$ and $\{c_t\}$ points
- `usePointsForInterpolation`: Construct interpolated `cFunc`
- `makevFunc`: Construct interpolated value function `vFunc`

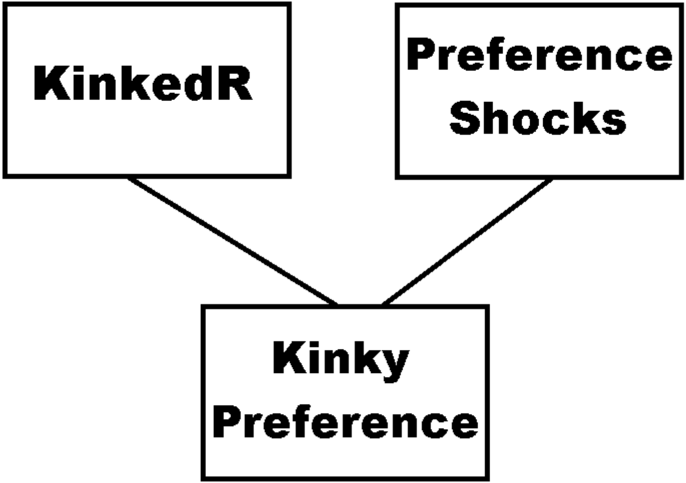
Idiosyncratic Income Shocks



Consumption-Saving Model Tree



Consumption-Saving Model Tree



Kinked R: Costly Borrowing

Make one small adjustment to idiosyncratic income shocks model:

- Interest rate on borrowing is higher than rate on saving

Kinked R: Costly Borrowing

$$
\begin{aligned}
v(m_t) &= \max_{c_t} u(c_t) + \beta \not{D}_{t+1} \mathbb{E}[v_{t+1}(m_{t+1})], \\
a_t &= m_t - c_t, \\
a_t &\geq \underline{a}, \\
m_{t+1} &= R/(\Gamma_{t+1}\psi_{t+1})a_t + \theta_{t+1}, \\
\theta_t \sim F_{\theta t}, & \quad\quad \psi_t \sim F_{\psi t}, \quad \mathbb{E}[F_{\psi t}] = 1, \\
u(c) &= \frac{c^{1-\rho}}{1-\rho}, \\
R &= \begin{cases} R_{boro} & \text{if } a_t < 0 \\ R_{save} & \text{if } a_t > 0 \end{cases}, \quad\quad R_{boro} \geq R_{save}.
\end{aligned}
$$

Kinked R: Costly Borrowing

`ConsKinkedRsolver` inherits from `ConsIndShockSolver`

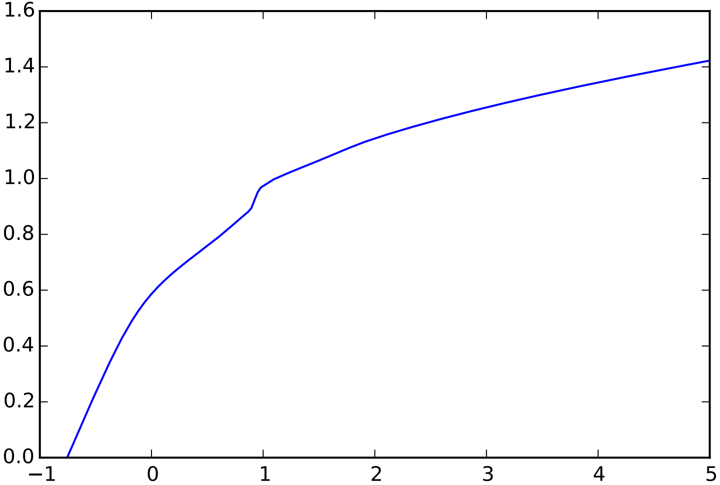
Additions to `__init__` method:

- Store new attributes `Rboro` and `Rsave`

Additions to `prepareToCalcEndOfPrdvP`:

- Four lines to use correct value of R for each value of a_t
- One line to apply that change to calculation of m_{t+1}
- Three lines to recalculate minimum MPC and human wealth

Kinked R: Costly Borrowing



Marginal Utility Shocks

Consider another small modification to `IndShockModel`:

- Multiplicative (idiosyncratic) shocks to utility each period.
- Consumption “more valuable” in some periods than others.

Marginal Utility Shocks

$$
\begin{aligned}
v(m_t) &= \max_{c_t} u(c_t) + \beta \cancel{D}_{t+1} \mathbb{E}[v_{t+1}(m_{t+1})], \\
a_t &= m_t - c_t, \\
a_t &\geq \underline{a}, \\
m_{t+1} &= R/(\Gamma_{t+1}\psi_{t+1}) a_t + \theta_{t+1}, \\
\theta_t \sim F_{\theta t}, & \qquad \psi_t \sim F_{\psi t}, \quad \mathbb{E}[F_{\psi t}] = 1, \\
u(c) &= \eta_t \frac{c^{1-\rho}}{1-\rho}, \qquad \eta_t \sim F_{\eta t}.
\end{aligned}
$$

Marginal Utility Shocks

New input `PrefShkDstn` is constructed:

- `PrefShkStd`: Standard deviation of (log) pref shocks
- `PrefShkCount`: Number of discrete shocks in “body”
- `PrefShkTailCount`: Discrete shocks in “augmented tail”

Marginal Utility Shocks

`ConsPrefShockSolver` inherits from `ConsIndShockSolver`

Additions to `__init__` method:

- 2 lines: Store preference shock distribution `PrefShkDstn`

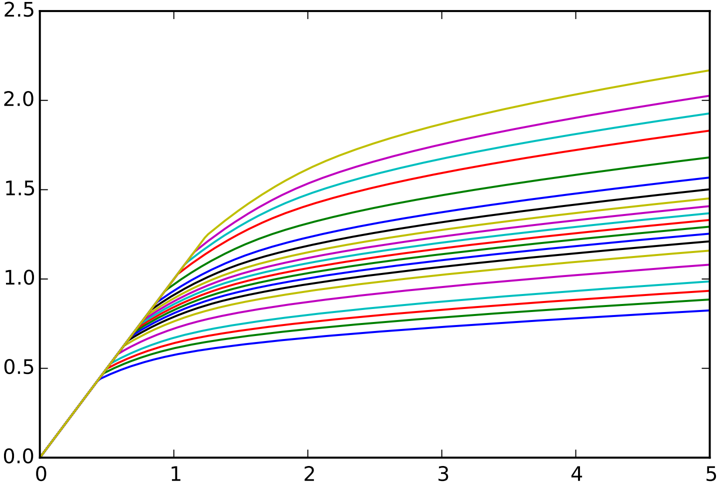
Replace `getPointsForInterpolation`

- 8 lines: Values of c_t and m_t for each η_t in `PrefShkDstn`

Replace `usePointsForInterpolation`

- 6 lines: Construct `cFunc` as a `LinearInterpOnInterp1D`
- 6 lines: Make `vPfunc` by integrating marginal utility across η_t

Marginal Utility Shocks



Combination Inheritance: “Kinky Preferences”

Combine those two extensions to `IndShockModel`:

- Borrowing has higher interest rate than saving...
- ...and there are shocks to marginal utility
- HARK makes this pretty easy

Combination Inheritance: "Kinky Preferences"

$$\begin{aligned}
v(m_t) &= \max_{c_t} u(c_t) + \beta \not{D}_{t+1} \mathbb{E}[v_{t+1}(m_{t+1})], \\
a_t &= m_t - c_t, \\
a_t &\geq \underline{a}, \\
m_{t+1} &= R/(\Gamma_{t+1}\psi_{t+1})a_t + \theta_{t+1}, \\
\theta_t \sim F_{\theta t}, & \quad \psi_t \sim F_{\psi t}, \quad \mathbb{E}[F_{\psi t}] = 1, \\
u(c) &= \eta_t \frac{c^{1-\rho}}{1-\rho}, \qquad \eta_t \sim F_{\eta t}, \\
R &= \begin{cases} R_{boro} & \text{if } a_t < 0 \\ R_{save} & \text{if } a_t > 0 \end{cases}, \qquad R_{boro} \geq R_{save}.
\end{aligned}$$

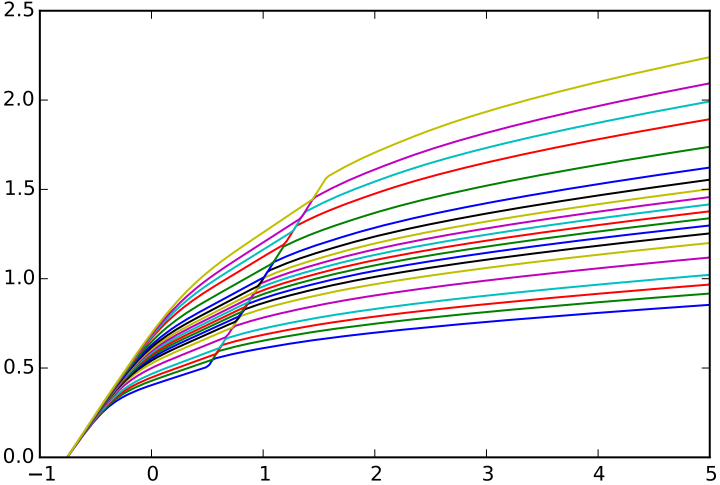
Combination Inheritance: "Kinky Preferences"

`ConsKinkyPrefSolver` inherits from two parent classes

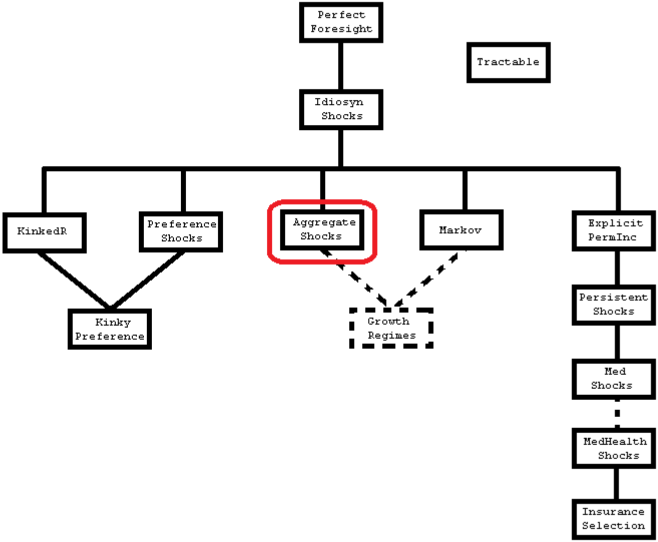
Entirety of `ConsKinkyPrefSolver` code:

```
class ConsKinkyPrefSolver(ConsPrefShockSolver,ConsKinkedRsolver):
    def __init__(self,solution_next,...):
        ConsKinkedRsolver.__init__(self,solution_next,...)
        self.PrefShkPrbs = PrefShkDstn[0]
        self.PrefShkVals = PrefShkDstn[1]
```

Combination Inheritance: “Kinky Preferences”



Consumption-Saving Model Tree



Aggregate Productivity Shocks

Consider a model with *aggregate* productivity shocks:

- Two aggregate shocks: fully permanent or fully transitory
- Interest rate and wage rate depend on capital-to-labor ratio
- And that ratio evolves over time according to some rule
- Where does that “rule” come from? Maybe general equilibrium

Aggregate Productivity Shocks

$$
\begin{aligned}
v_t(m_t, k_t) &= \max_{c_t} u(c_t) + \beta \not{D}_{t+1} \mathbb{E}[v_{t+1}(m_{t+1}, k_{t+1})], \\
a_t &= m_t - c_t, \\
a_t &\geq 0, \\
m_{t+1} &= \frac{R_{t+1}}{\Gamma_{t+1}\psi_{t+1}\Psi_{t+1}} \cdot a_t + W_{t+1}\theta_{t+1}, \\
R_{t+1} = \mathbf{R}(k_{t+1}/\Theta_{t+1}), & \quad W_{t+1} = \mathbf{W}(k_{t+1}/\Theta_{t+1}), \\
k_{t+1} &= \mathbf{k}(k_t), \\
\theta_t \sim F_{\theta t}, & \quad \psi_t \sim F_{\psi t}, \quad \mathbb{E}[F_{\psi t}] = 1, \\
\Theta_t \sim F_{\Theta}, & \quad \Psi_t \sim F_{\Psi}, \quad \mathbb{E}[F_{\Psi}] = \mathbb{E}[F_{\Theta}] = 1.
\end{aligned}
$$

Aggregate Productivity Shocks

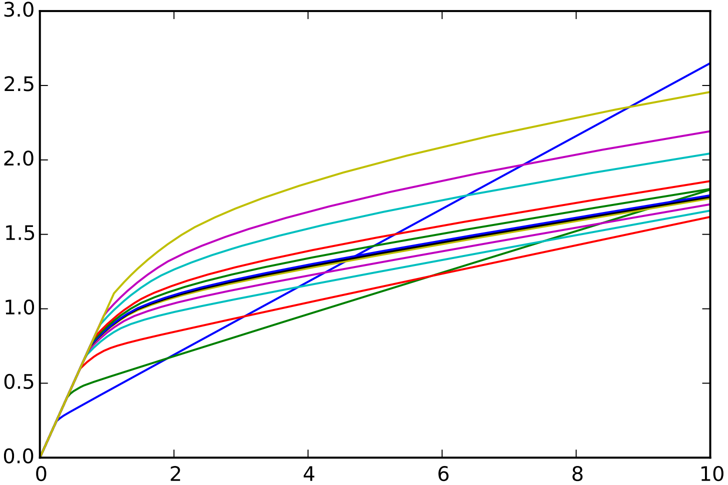
Some totally new inputs for an `AggShockConsumerType`:

- `Rfunc` and `wFunc`: Factor payments as function of k_t
- `kGrid`: Grid of k_t state values (sort of constructed)
- `kNextFunc`: $\mathbb{E}[k_{t+1}] = \mathbf{k}(k_t)$

`IncomeDstn` combines idiosyncratic and aggregate shocks:

- Discrete approximation to aggregate shock distribution constructed like idiosyncratic shocks: `PermShkAggStd`, `TranShkAggStd`, `PermShkAggCount`, `TranShkAggCount`
- `IncomeDstn` has five elements: probs, idio shocks, agg shocks

Aggregate Productivity Shocks



Macroeconomics and General Equilibrium

- In AggShocksModel, where did `kNextFunc` come from?
- Some objects or processes treated as exogenous at the micro level are actually endogenous at the macro level
- HARK calls these “dynamic rules” (even if they’re static)
- Want consistency between agent beliefs about dynamic rules and actual outcomes when agents act on those beliefs

Market Classes for Aggregate Shocks Model

Small Open Economy

- Wage and interest rates exogenous, don't depend on k_t
- `kNextFunc` is irrelevant or trivial
- But the market still generates aggregate shocks

Cobb-Douglas Economy

- Wage and interest rates from marginal products of factors
- Solve micro model, generate simulated history of k_t
- Make new `kNextFunc` from history, iterate to convergence

Cobb-Douglas Economy in the `Market` Framework

Farming metaphor for aggregate productivity shocks model:

- `sow`: Distribute $(k_t, R_t, W_t, \Theta_t, \Psi_t)$ to consumers
- `cultivate`: Consumers draw (θ_t, ψ_t), choose C_t
- `reap`: Collect assets A_t from consumers
- `mill`: Calculate $K_{t+1} \rightarrow k_{t+1}$, draw $(\Theta_{t+1}, \Psi_{t+1})$, get (R_{t+1}, W_{t+1})
- `store`: Record k_{t+1} in its history

Loop that process for (say) 1000 periods

- `calcDynamics`: Regress $log(k_t)$ on $log(k_{t-1})$, make new $\mathbf{k}$
- Distribute new **k** to consumers as `kNextFunc`

Other Consumption-Saving Models in HARK

But wait, there's more!

- `TractableBufferStock`: Highly specialized idiosyncratic shocks

And even more to come...

Other Consumption-Saving Models in HARK

But wait, there's more!

- `TractableBufferStock`: Highly specialized idiosyncratic shocks
- `MarkovModel`: $\cancel{D}$, Γ, F_θ, F_ψ, R vary by discrete state

And even more to come...

Other Consumption-Saving Models in HARK

But wait, there's more!

- `TractableBufferStock`: Highly specialized idiosyncratic shocks
- `MarkovModel`: $\cancel{D}$, Γ, F_θ, F_ψ, R vary by discrete state
- `ExplicitPermInc`: Same as `IndShock`, but without normalization
- `PersistentShock`: "Permanent" shocks not fully permanent

And even more to come...

Other Consumption-Saving Models in HARK

But wait, there's more!

- `TractableBufferStock`: Highly specialized idiosyncratic shocks
- `MarkovModel`: $\cancel{D}$, Γ, F_θ, F_ψ, R vary by discrete state
- `ExplicitPermInc`: Same as `IndShock`, but without normalization
- `PersistentShock`: "Permanent" shocks not fully permanent
- `MedShock`: 2nd consumption good with random marginal utility

And even more to come...

Other Consumption-Saving Models in HARK

But wait, there's more!

- `TractableBufferStock`: Highly specialized idiosyncratic shocks
- `MarkovModel`: $\cancel{D}$, Γ, F_θ, F_ψ, R vary by discrete state
- `ExplicitPermInc`: Same as `IndShock`, but without normalization
- `PersistentShock`: "Permanent" shocks not fully permanent
- `MedShock`: 2nd consumption good with random marginal utility
- `MedHealthShock`:* Medical shocks plus discrete health states
- `DynInsSel`:* ...plus choice over medical insurance contracts

And even more to come...

Combination Inheritance in the Future

Future goal: ability for any solver to inherit `MarkovModel`

- As is, `MarkovModel` extends only `IndShockModel`...
- ...but `MedHealthShockModel` also uses Markov discrete states!

Combination Inheritance in the Future

Future goal: ability for any solver to inherit `MarkovModel`

- As is, `MarkovModel` extends only `IndShockModel`...
- ...but `MedHealthShockModel` also uses Markov discrete states!
- Want to fully generalize it so it acts like a wrapper
- Make a new solver class that inherits from `FooSolver` and `MarkovSolver`, generates `FooMarkovSolver` with no work

Combination Inheritance in the Future

Future goal: ability for any solver to inherit `MarkovModel`

- As is, `MarkovModel` extends only `IndShockModel`...
- ...but `MedHealthShockModel` also uses Markov discrete states!
- Want to fully generalize it so it acts like a wrapper
- Make a new solver class that inherits from `FooSolver` and `MarkovSolver`, generates `FooMarkovSolver` with no work
- E.g. Markov states in aggregate shocks model: growth regimes?

HARK in the Future: Durable Goods

$$u(C_t, D_t) = \frac{(C_t^{\alpha} D_t^{1-\alpha})^{1-\rho}}{1-\rho}, \qquad D_{t+1} = (1-\delta)(D_t + i_t).$$

- Surge in durables (housing and autos) before Great Recession
- Could be labeled “spent-up demand”; *i* likely to decrease whether or not GR hits, but recession makes it greater...

HARK in the Future: Durable Goods

$$u(C_t, D_t) = \frac{(C_t^{\alpha} D_t^{1-\alpha})^{1-\rho}}{1-\rho}, \qquad D_{t+1} = (1-\delta)(D_t + i_t).$$

- Surge in durables (housing and autos) before Great Recession
- Could be labeled “spent-up demand”; *i* likely to decrease whether or not GR hits, but recession makes it greater...
- ...and RA models would likely underestimate durables decrease!
- Some pre-GR durables purchases were enabled only by excessive/imprudent lending practices

HARK in the Future: Durable Goods

$$u(C_t, D_t) = \frac{(C_t^{\alpha} D_t^{1-\alpha})^{1-\rho}}{1-\rho}, \qquad D_{t+1} = (1-\delta)(D_t + i_t).$$

- Surge in durables (housing and autos) before Great Recession
- Could be labeled “spent-up demand”; *i* likely to decrease whether or not GR hits, but recession makes it greater...
- ...and RA models would likely underestimate durables decrease!
- Some pre-GR durables purchases were enabled only by excessive/imprudent lending practices
- Heterogeneous agents approach captures this

HARK in the Future: Costly Employment Search

$$u(C_t, s_t) = \frac{((1 - s_t)^\alpha C_t)^{1-\rho}}{1-\rho}, \qquad \text{Prob}(e_{t+1} = 1 | e_t = 0) = s_t.$$

- What if re-employment probability is endogenous choice?
- Searching for job is costly, enters utility function
- How does search intensity change as benefits approach expiry?
- How does unemp spell length change with benefits duration?

HARK in the Future: Other Markets

Labor Markets

- `CobbDouglasEconomy` has exogenous fixed L_t; can endogenize
- Populated by agents with costly employment search
- Or endogenous ℓ_t on the extensive margin: LFP
- Difficulty of search depends on macro state

Medical Insurance Markets

- Add to medical shocks model: choice over insurance contract
- Premiums depend on actuarial constraints...
- ...and maybe some government regulations: ACA