HARK: Open Source Tools for Heterogeneous Agents Modeling

Matthew N. White

University of Delaware

July 12, 2018

Elements of a dynamic economic model:

- Situations agent(s) can be in: the **state space**
- Choices agent(s) can make: the **control space**
- Constraints agent(s) face when making choice
- The nature of risks agent(s) face: the **shock distribution**
- How those combine to yield tomorrow's state: **dynamics**
- How agents feel about sequences of those: **preferences**
- How agents **process information** and/or **make choice**

Classification of Dynamic Economic Models

The nature of time:

- **Discrete** vs **continuous** time
- **Finite** vs **infinite** horizon

Representative vs Heterogeneous Agents

- In some models, one agent can stand in for many– the **representative consumer** or **representative firm**
- In others, shocks introduce *ex post* heterogeneity

Classification of Dynamic Economic Models

Equilibrium conditions:

- Agents solve their problem taking dynamics as **exogenous**
- But some aspects of dynamics are **endogenous** to controls and states of agents collectively
- **Information processing** includes **beliefs** about dynamics
- **Equilibrium:** beliefs "consistent" with actual dynamics

Required knowledge to enter heterogeneous agents macro:

- Economic modeling (and history thereof)
- Programming / software development
- Numeric methods: optimization, rootfinding, interpolation...
- Dynamic solution methodology
- "Trade secrets" or "occult knowledge"

Actually taught in economics PhD programs:

- Economic modeling (and history thereof)
- Programming / software development
- Numeric methods: optimization, rootfinding, interpolation...
- Dynamic solution methodology
- "Trade secrets" or "occult knowledge"

Current State of HA Macro Modeling

How programming HA models is done today:

- Written for today, for one particular purpose
- Everything is bespoke and handcrafted
- Documentation is missing or apocryphal
- Like econometrics circa 1970

Strategies for HA model programming:

1. Inherit codebase from your adviser('s adviser), repurpose
2. Re-reinvent the wheel / divine the secrets
3. Engage in "duct tape coding"

Current State of HA Macro Modeling

Why are things done this way?

- PhD program inertia / limited bandwidth
- Keep safe the secret of the secret sauce
- Hide the rat droppings
- Don't want to know how sausage is made

Solution: HARK

Heterogeneous Agents Resources & toolKit:

- Modular, extensible, interoperable, open source toolkit for solving discrete time heterogeneous agents models
- For “HA macro” and “structural micro” models
- Common framework for representing agents, environments, etc
- Dynamics, behavior, aggregation, and beliefs are modular
- “Model tree” based on class inheritance

Goals of HARK

HARK intended to make it much easier to...

- Enter the world of HA modeling: reduce barriers
- Teach HA methods and techniques (with hands-on exercises)
- Compare models to each other: interoperability
- Add new models and features: extensibility
- Mix-and-match components: modularity

Goals of HARK

Ultimate goals of HARK:

- Accelerate development of models for research and policymaking
- Bridge gaps among rational expectations, behavioral, "agent-based" modeling worlds
- And between "HA macro" and "structural micro"
- Make code audits feasible/expected in refereeing

Contents of HARK

Stuff in HARK, generally:

- Broadly useful numeric tools
- Superclass for representing agents: `AgentType`
- ...and environments they interact in: `Market`
- Ever-expanding “model tree” as subclasses
- In dev: documentation and tutorial notebooks

HARK's Superclass for Agents: `AgentType`

The `AgentType` class:

- General purpose class for representing economic agents
- Each model creates a subclass of `AgentType`
  - e.g. `PerfForesightConsumerType`
- Key method: `solve()`
- Just a universal backward induction loop...
- ...that lets different models "play nicely" together
- Complex models extend basic ones through class inheritance

Specifying a "Microeconomic" Model in HARK

Elements to specify a subclass of `AgentType`:

- What variables/objects agent needs to solve *one period* of his problem, and whether those things vary over time: `time_inv` and `time_vary`
- How to solve the one period problem, given current variables objects and solution to next period's problem: `solveOnePeriod`
- How to behave at the end of time, or as an initial solution guess: `solveTerminal`

Specifying a "Microeconomic" Model in HARK

Elements to specify an instance of an `AgentType` subclass:

- What is the nature of time? Lifecycle? Infinite horizon? Finitely repeated loop? `cycles` and `T_cycle`
- What values do variables named in `time_vary` take on in each period of the "cycle"?
- What values do variables named in `time_inv` take on?
- How many agents of this "type"? `AgentCount`

Example Model: Basic Consumption-Saving

Consumption-saving model with idiosyncratic permanent and transitory shocks to income (normalized format):

$$u(c) = c^{1-\rho}/(1-\rho).$$

$$v_t(m_t) = \max_{c_t} u(c_t) + \beta \not{D}_t \mathbb{E}_t \left[(\psi_{t+1}\Gamma_{t+1})^{1-\rho} v_{t+1}(m_{t+1}) \right] \text{ s.t.}$$

$$a_t = m_t - c_t, \quad a_t \geq \underline{a},$$

$$m_{t+1} = R/(\Gamma_{t+1}\psi_{t+1})a_t + \theta_{t+1},$$

$$\psi_{t+1}, \theta_{t+1} \sim F_{t+1}, \quad \mathbb{E}[\psi_{t+1}] = 1.$$

Example `AgentType` Subclass: `IndShockConsumerType`

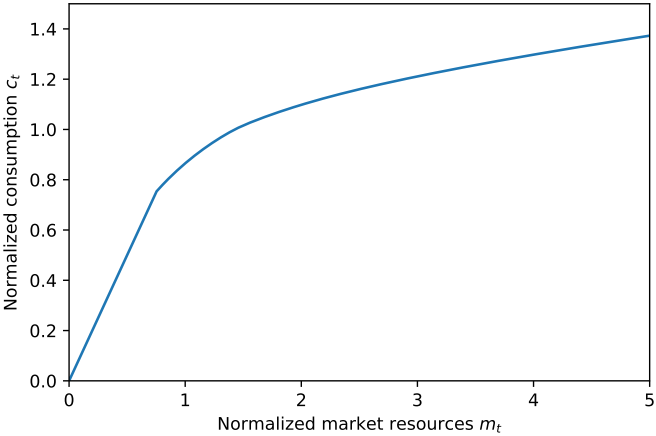
Mapping from model to attributes named in `time_inv`:

- Discount factor $\beta \longrightarrow$ `DiscFac`
- Relative risk aversion $\rho \longrightarrow$ `CRRA`
- Risk-free interest rate $R \longrightarrow$ `Rfree`
- Artificial borrowing constraint $\underline{a} \longrightarrow$ `BoroCnstArt`

Mapping from model to attributes named in `time_vary`:

- Survival probability $\not{D}_{t+1} \longrightarrow$ `LivPrb`
- Permanent income growth factor $\Gamma_{t+1} \longrightarrow$ `PermGroFac`
- Distribution of perm and trans income shocks $F_{t+1} \longrightarrow$ `IncomeDstn`

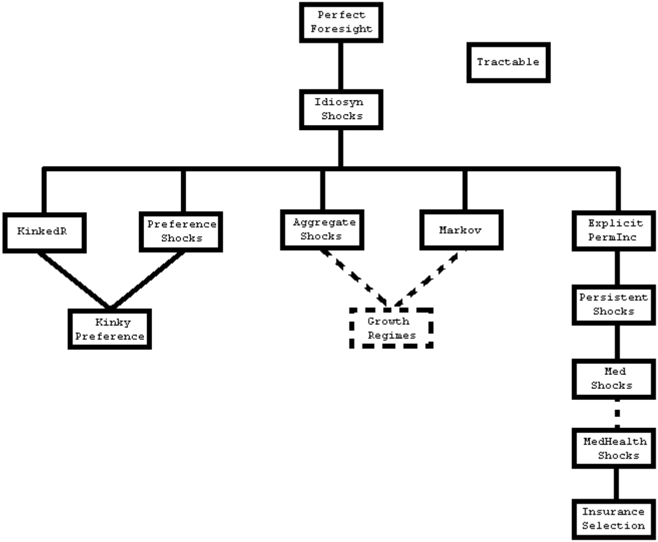
IndShockConsumerType Consumption Function



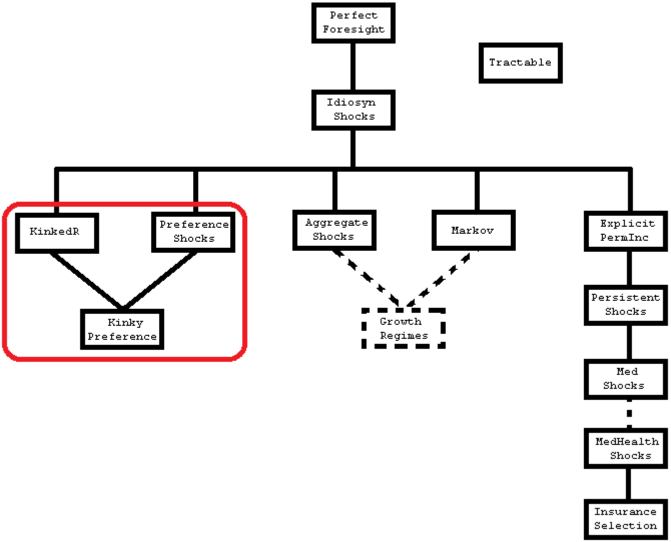
Object-Oriented Solution Methods

- "Parent" models in HARK are special cases of "child" models
- Solvers in HARK are object classes that act (a lot) like functions
- Child models inherit solver class from parent...
- ...and then add or change its methods as needed

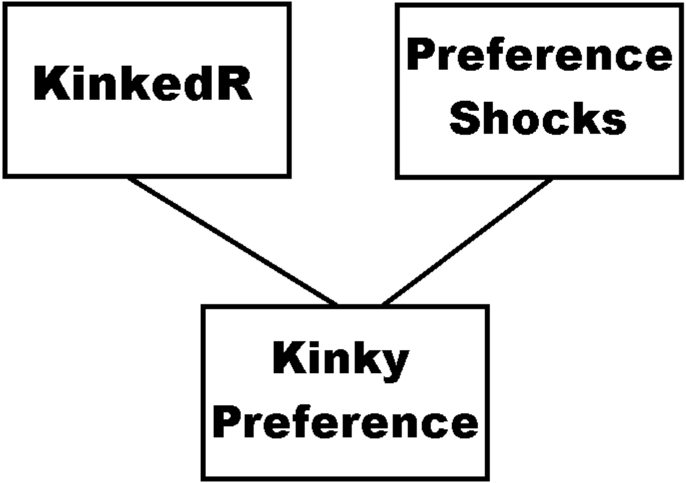
Consumption-Saving Model Tree



Consumption-Saving Model Tree



Consumption-Saving Model Tree



Kinked R: Costly Borrowing (1/3)

Make one small adjustment to idiosyncratic income shocks model: interest rate on borrowing is higher than rate on saving.

$$
\begin{aligned}
u(c) &= \frac{c^{1-\rho}}{1-\rho}, \\
v(m_t) &= \max_{c_t} u(c_t) + \beta \cancel{D}_{t+1} \mathbb{E}[v_{t+1}(m_{t+1})], \\
a_t &= m_t - c_t, \qquad a_t \geq \underline{a}, \\
m_{t+1} &= R/(\Gamma_{t+1}\psi_{t+1}) a_t + \theta_{t+1}, \\
\psi_{t+1}, \theta_{t+1} \sim F_{t+1}, & \qquad \mathbb{E}[\psi_{t+1}] = 1, \\
R &= \begin{cases} R_{boro} & \text{if } a_t < 0 \\ R_{save} & \text{if } a_t > 0 \end{cases}, \qquad R_{boro} \geq R_{save}.
\end{aligned}
$$

Kinked R: Costly Borrowing (2/3)

`ConsKinkedRsolver` inherits from `ConsIndShockSolver`

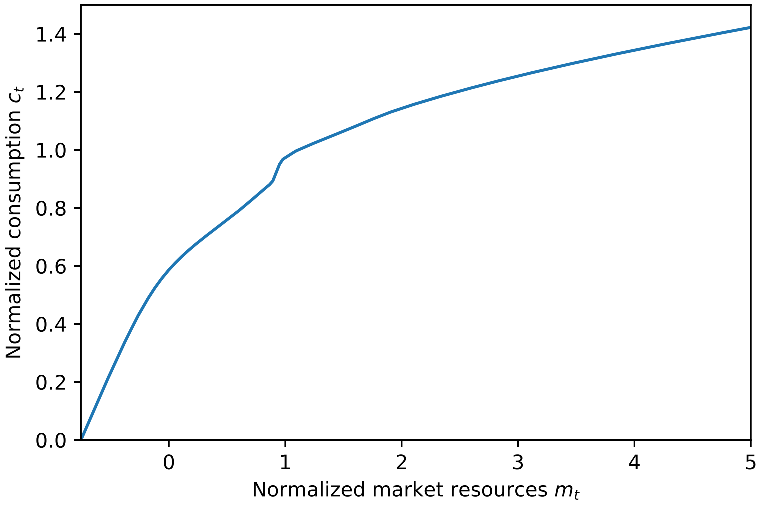
Additions to `__init__` method:

- 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 (3/3)



Marginal Utility Shocks (1/3)

Consider another small modification to `IndShockModel`:

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

$$
\begin{aligned}
u(c;\eta) &= \eta\frac{c^{1-\rho}}{1-\rho}, \qquad \eta_t \sim F_\eta, \\
v(m_t,\eta_t) &= \max_{c_t} u(c_t;\eta_t) + \beta \not{D}_{t+1}\mathbb{E}[v_{t+1}(m_{t+1})], \\
a_t &= m_t - c_t, \qquad a_t \geq \underline{a}, \\
m_{t+1} &= R/(\Gamma_{t+1}\psi_{t+1})a_t + \theta_{t+1}, \\
\psi_{t+1},\theta_{t+1} \sim F_{t+1}, &\qquad \mathbb{E}[\psi_{t+1}] = 1.
\end{aligned}
$$

Marginal Utility Shocks (2/3)

`ConsPrefShockSolver` inherits from `ConsIndShockSolver`

Additions to `__init__` method:

- 2 lines: Store preference shock distribution `PrefShkDstn`

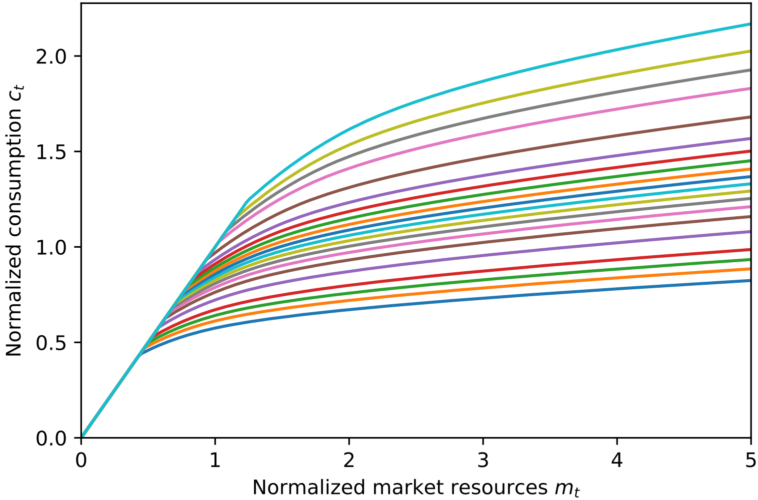
Replace `getPointsForInterpolation`

- 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 (2/3)



Combination Inheritance: “Kinky Preferences” (1/4)

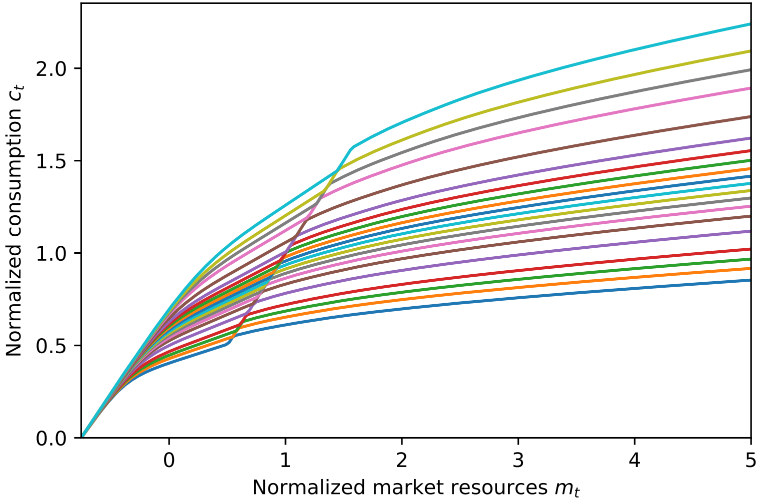
Combine those two extensions to `IndShockModel`:

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

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

Entirety of the code for the ConsKinkyPrefSolver:

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


Normalized consumption c_t
Normalized market resources m_t
0.0
0.5
1.0
1.5
2.0
0
1
2
3
4
5

Just One Animal on the Econ-ARK

Grand Vision: The Econ-ARK

- HA modeling is just one subfield in computational econ
- Econ-ARK is the umbrella for open source dynamic models
- We want to build frameworks for other areas:
    - Monetary economics: MARK in early stages at IMF
    - Continuous time models
    - Industrial organization models
- Frameworks will share top-level numeric tools

Organizations supporting the Econ-ARK

- Grant from Alfred P. Sloan Foundation
- Fiscal sponsorship from NumFocus
- Originally spawned by CFPB (support from IMF and OFR)
- Interest and potential support from central banks

Alfred P. Sloan FOUNDATION

NUMFOCUS
OPEN CODE = BETTER SCIENCE

- Installable package: `pip install econ-ark`
- Conda package on a private channel, stay tuned
- GitHub: `github.com/econ-ark`

Status Report on HARK and Econ-ARK

- Installable package: `pip install econ-ark`
- Conda package on a private channel, stay tuned
- GitHub: `github.com/econ-ark`
- Website: `www.econ-ark.org`
- Run live notebooks: examples, basic tutorial
- More tutorials and model documentation notebooks in dev

Status Report on HARK and Econ-ARK

- Installable package: `pip install econ-ark`
- Conda package on a private channel, stay tuned
- GitHub: github.com/econ-ark
- Website: www.econ-ark.org
- Run live notebooks: examples, basic tutorial
- More tutorials and model documentation notebooks in dev
- Written in Python 2.7, and will be forever

Status Report on HARK and Econ-ARK

- Installable package: `pip install econ-ark`
- Conda package on a private channel, stay tuned
- GitHub: github.com/econ-ark
- Website: www.econ-ark.org
- Run live notebooks: examples, basic tutorial
- More tutorials and model documentation notebooks in dev
- Written in Python 2.7, and will be forever
- j/k, we have a pull request updating to Py3

Status Report on HARK and Econ-ARK

- Installable package: `pip install econ-ark`
- Conda package on a private channel, stay tuned
- GitHub: github.com/econ-ark
- Website: www.econ-ark.org
- Run live notebooks: examples, basic tutorial
- More tutorials and model documentation notebooks in dev
- Written in Python 2.7, and will be forever
- j/k, we have a pull request updating to Py3
- Unit tests coming, we swear
- More models in development all the time

Who's Who in Econ-ARK

- Chris Carroll — Johns Hopkins University
- Matt White — University of Delaware
- Jackie Kazil — Capital One
- Nate Palmer — OFR $\longrightarrow$ Federal Reserve Board
- Dave Low — Consumer Financial Protection Bureau
- Josh Epstein — New York University
- Alex Kaufman — Woodrow Wilson School
- Patrick Mogensen — Copenhagen University
- Tiphanie Magne — University of Delaware
- Pablo Winant — Bank of England $\rightarrow$???
- Open Tech Strategies team members
- And YOU — looking for a project manager