

# Introducing the Econ-ARK: Economics “Algorithmic Repository and toolKit”

Presentation by Chris Carroll at  
CEF Milan

June 21, 2018

# Goals: Like DYNARE's, but for Models With Heterogeneity

Operationally (for now): More like STATA than DYNARE or HetSol

# Goals: Like DYNARE's, but for Models With Heterogeneity

Operationally (for now): More like STATA than DYNARE or HetSol

State-of-the-art tools for:

# Goals: Like DYNARE's, but for Models With Heterogeneity

Operationally (for now): More like STATA than DYNARE or HetSol

State-of-the-art tools for:

- 1 Solving microeconomic dynamic stochastic optimization problems

# Goals: Like DYNARE's, but for Models With Heterogeneity

Operationally (for now): More like STATA than DYNARE or HetSol

State-of-the-art tools for:

- 1 Solving microeconomic dynamic stochastic optimization problems
  - 'Hard' Bellman problems with uncertainty, 'kinks,' nonconvexities

# Goals: Like DYNARE's, but for Models With Heterogeneity

Operationally (for now): More like STATA than DYNARE or HetSol

State-of-the-art tools for:

- ① Solving microeconomic dynamic stochastic optimization problems
  - 'Hard' Bellman problems with uncertainty, 'kinks,' nonconvexities
- ② Simulating populations of agents

# Goals: Like DYNARE's, but for Models With Heterogeneity

Operationally (for now): More like STATA than DYNARE or HetSol

State-of-the-art tools for:

- ① Solving microeconomic dynamic stochastic optimization problems
  - 'Hard' Bellman problems with uncertainty, 'kinks,' nonconvexities
- ② Simulating populations of agents
  - Whether or not they are solving DSOP

# Goals: Like DYNARE's, but for Models With Heterogeneity

Operationally (for now): More like STATA than DYNARE or HetSol

State-of-the-art tools for:

- ① Solving microeconomic dynamic stochastic optimization problems
  - 'Hard' Bellman problems with uncertainty, 'kinks,' nonconvexities
- ② Simulating populations of agents
  - Whether or not they are solving DSOP
  - Allows disciplined exploration of deviations from RE



# Goals: Like DYNARE's, but for Models With Heterogeneity

Operationally (for now): More like STATA than DYNARE or HetSol

State-of-the-art tools for:

- ① Solving microeconomic dynamic stochastic optimization problems
  - 'Hard' Bellman problems with uncertainty, 'kinks,' nonconvexities
- ② Simulating populations of agents
  - Whether or not they are solving DSOP
  - Allows disciplined exploration of deviations from RE
- ③ Finding equilibria for markets/economies populated by such agents

# Goals: Like DYNARE's, but for Models With Heterogeneity

Operationally (for now): More like STATA than DYNARE or HetSol

State-of-the-art tools for:

- ① Solving microeconomic dynamic stochastic optimization problems
  - 'Hard' Bellman problems with uncertainty, 'kinks,' nonconvexities
- ② Simulating populations of agents
  - Whether or not they are solving DSOP
  - Allows disciplined exploration of deviations from RE
- ③ Finding equilibria for markets/economies populated by such agents
  - Definition of eqbm is user-specified

# Goals: Like DYNARE's, but for Models With Heterogeneity

Operationally (for now): More like STATA than DYNARE or HetSol

State-of-the-art tools for:

- ① Solving microeconomic dynamic stochastic optimization problems
  - 'Hard' Bellman problems with uncertainty, 'kinks,' nonconvexities
- ② Simulating populations of agents
  - Whether or not they are solving DSOP
  - Allows disciplined exploration of deviations from RE
- ③ Finding equilibria for markets/economies populated by such agents
  - Definition of eqbm is user-specified
    - e.g.: Stock Price Equilibrates Shares bought = Shares sold

# Goals: Like DYNARE's, but for Models With Heterogeneity

Operationally (for now): More like STATA than DYNARE or HetSol

State-of-the-art tools for:

- ① Solving microeconomic dynamic stochastic optimization problems
  - 'Hard' Bellman problems with uncertainty, 'kinks,' nonconvexities
- ② Simulating populations of agents
  - Whether or not they are solving DSOP
  - Allows disciplined exploration of deviations from RE
- ③ Finding equilibria for markets/economies populated by such agents
  - Definition of eqbm is user-specified
    - e.g.: Stock Price Equilibrates Shares bought = Shares sold
    - Expectations can be – but need not be – rational

# Who Has Produced It?

Name	TLA	Affiliation
<i>Christopher D Carroll</i>	CDC	JHU, CFPB
<i>David C Low</i>	DCL	CFPB
<i>Nathan M Palmer</i>	NMP	OFR
<i>Matthew N White</i>	MNW	UDel, CFPB
<i>Alex Kaufman</i>	ABK	CFPB → Princeton

Nothing herein may be interpreted as reflecting opinions of

- CFPB - United States Consumer Financial Protection Bureau
- JHU - Johns Hopkins University
- IMF - International Monetary Fund
- OFR - Office of Financial Research, U.S. Treasury
- UDel - University of Delaware

# Big Grant from Alfred P. Sloan Foundation!

Three Years

# Big Grant from Alfred P. Sloan Foundation!

## Three Years

- Hire Programmers, RA's, Open Source Project Managers, etc etc

# Why: Goals

Make it *much* easier:



# Why: Goals

Make it *much* easier:

- To get started doing structural Heterogeneous Agent modeling

# Why: Goals

Make it *much* easier:

- To get started doing structural Heterogeneous Agent modeling
- To teach newcomers how to use such models

# Why: Goals

Make it *much* easier:

- To get started doing structural Heterogeneous Agent modeling
- To teach newcomers how to use such models
- To *compare* models to each other

# Why: Goals

Make it *much* easier:

- To get started doing structural Heterogeneous Agent modeling
- To teach newcomers how to use such models
- To *compare* models to each other
- To add new capabilities

# Why: Goals

Make it *much* easier:

- To get started doing structural Heterogeneous Agent modeling
- To teach newcomers how to use such models
- To *compare* models to each other
- To add new capabilities
- To mix-and-match components/modules/agent types

# Why: Goals

Make it *much* easier:

- To get started doing structural Heterogeneous Agent modeling
- To teach newcomers how to use such models
- To *compare* models to each other
- To add new capabilities
- To mix-and-match components/modules/agent types

# Why: Goals

Make it *much* easier:

- To get started doing structural Heterogeneous Agent modeling
- To teach newcomers how to use such models
- To *compare* models to each other
- To add new capabilities
- To mix-and-match components/modules/agent types

Remove the excuse 'Structural model was not worth the effort'

# Where Is It?

Browse without installing:

- Browse on our webpage at [econ-ark.org](http://econ-ark.org)



# Where Is It?

Browse without installing:

- Browse on our webpage at [econ-ark.org](http://econ-ark.org)
- Browse our code at <http://github.com/econ-ark>

# Where Is It?

Browse without installing:

- Browse on our webpage at [econ-ark.org](http://econ-ark.org)
- Browse our code at <http://github.com/econ-ark>
- Browse our talks at <http://github.com/econ-ark/PARK>

# Where Is It?

Browse without installing:

- Browse on our webpage at [econ-ark.org](http://econ-ark.org)
- Browse our code at <http://github.com/econ-ark>
- Browse our talks at <http://github.com/econ-ark/PARK>
- Overview paper at <http://github.com/econ-ark/PARK/SciPy2018.pdf>

# Where Is It?

Browse without installing:

- Browse on our webpage at [econ-ark.org](http://econ-ark.org)
- Browse our code at <http://github.com/econ-ark>
- Browse our talks at <http://github.com/econ-ark/PARK>
- Overview paper at <http://github.com/econ-ark/PARK/SciPy2018.pdf>
- Browse our live [notebooks](#)

# Where Is It?

Browse without installing:

- Browse on our webpage at [econ-ark.org](http://econ-ark.org)
- Browse our code at <http://github.com/econ-ark>
- Browse our talks at <http://github.com/econ-ark/PARK>
- Overview paper at <http://github.com/econ-ark/PARK/SciPy2018.pdf>
- Browse our live [notebooks](#)
- Browse our [documentation](#)

# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)

# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:

# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:
  - ① **Anaconda2** - adds many packages useful for scientific computing



# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:
  - 1 **Anaconda2** - adds many packages useful for scientific computing
  - 2 Python 2.7

# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:
  - 1 **Anaconda2** - adds many packages useful for scientific computing
  - 2 Python 2.7
    - **On Mac or Linux** to download and install it

# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:
  - 1 **Anaconda2** - adds many packages useful for scientific computing
  - 2 Python 2.7
    - **On Mac or Linux** to download and install it
    - **On Windows**

# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:
  - 1 **Anaconda2** - adds many packages useful for scientific computing
  - 2 Python 2.7
    - **On Mac or Linux** to download and install it
    - **On Windows**
    - Install **Jupyter**

# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:
  - 1 **Anaconda2** - adds many packages useful for scientific computing
  - 2 Python 2.7
    - On Mac or Linux to download and install it
    - On Windows
    - Install Jupyter
    - Make sure you have **pip** installed

# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:
  - 1 **Anaconda2** - adds many packages useful for scientific computing
  - 2 Python 2.7
    - On Mac or Linux to download and install it
    - On Windows
    - Install Jupyter
    - Make sure you have **pip** installed
  - 3 Install the 'econ-ark' package:

# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:
  - 1 **Anaconda2** - adds many packages useful for scientific computing
  - 2 Python 2.7
    - On Mac or Linux to download and install it
    - On Windows
    - Install Jupyter
    - Make sure you have **pip** installed
  - 3 Install the 'econ-ark' package:
    - `pip install econ-ark`

# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:
  - 1 **Anaconda2** - adds many packages useful for scientific computing
  - 2 Python 2.7
    - On Mac or Linux to download and install it
    - On Windows
    - Install Jupyter
    - Make sure you have **pip** installed
  - 3 Install the 'econ-ark' package:
    - `pip install econ-ark`



# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:
  - 1 **Anaconda2** - adds many packages useful for scientific computing
  - 2 Python 2.7
    - On Mac or Linux to download and install it
    - On Windows
    - Install Jupyter
    - Make sure you have **pip** installed
  - 3 Install the 'econ-ark' package:
    - `pip install econ-ark`

# Installing It On Your Local Computer

- You Need Python 2.7 (Python 3 target is July)
- If you don't have Python 2.7 on your computer, install either:
  - 1 **Anaconda2** - adds many packages useful for scientific computing
  - 2 Python 2.7
    - On Mac or Linux to download and install it
    - On Windows
    - Install Jupyter
    - Make sure you have **pip** installed
  - 3 Install the 'econ-ark' package:
    - `pip install econ-ark`
- Get our **our demonstration notebooks** from **DemARK**