Introduction

Brian C. Jenkins

Econ 126: Computational Macroeconomics

University of California, Irvine

February 27, 2019

Why Computational Methods for Macroeconomics

- Computational tools are indispensable for many areas of economic research and particularly for macroeconomics.
- Macroeconomists use computational tools to:
  1. Solve and simulate complex dynamic models (often impossible with paper and pencil)
  2. Manage and analyze data
- Undergraduate economic curriculum often shields students from this knowledge for a variety of reasons (e.g., pedagogical philosophy, perceived lack of math proficiency of students, cost to faculty of course design)

Why Computational Methods for Macroeconomics

Example: RBC Model

- A three-equation real business cycle (RBC) model:

Why Computational Methods for Macroeconomics

Example: RBC Model

- A three-equation real business cycle (RBC) model:

$$C_t^{-1} = \beta E_t \left[C_{t+1}^{-1} \left(\alpha A_{t+1} K_{t+1}^{\alpha-1} + 1 - \delta \right) \right] \tag{1}$$

$$C_t + K_{t+1} = A_t K_t^{\alpha} + (1-\delta) K_t \tag{2}$$

$$\log A_{t+1} = \rho \log A_t + \epsilon_{t+1} \tag{3}$$

Why Computational Methods for Macroeconomics

Example: RBC Model

- A three-equation real business cycle (RBC) model:

$$C_t^{-1} = \beta E_t \left[C_{t+1}^{-1} \left(\alpha A_{t+1} K_{t+1}^{\alpha-1} + 1 - \delta \right) \right] \tag{1}$$

$$\log A_{t+1} = \rho \log A_t + \epsilon_{t+1} \tag{3}$$

Why Computational Methods for Macroeconomics

Example: RBC Model

- A three-equation real business cycle (RBC) model:

$$C_t^{-1} = \beta E_t \left[C_{t+1}^{-1} \left(\alpha A_{t+1} K_{t+1}^{\alpha-1} + 1 - \delta \right) \right] \tag{1}$$

$$C_t + K_{t+1} = A_t K_t^{\alpha} + (1-\delta) K_t \tag{2}$$

$$\log A_{t+1} = \rho \log A_t + \epsilon_{t+1} \tag{3}$$

Why Computational Methods for Macroeconomics

Example: RBC Model

- A three-equation real business cycle (RBC) model:

$$C_t^{-1} = \beta E_t \left[C_{t+1}^{-1} \left(\alpha A_{t+1} K_{t+1}^{\alpha-1} + 1 - \delta \right) \right] \quad (1)$$

$$C_t + K_{t+1} = A_t K_t^{\alpha} + (1-\delta) K_t \quad (2)$$

$$\log A_{t+1} = \rho \log A_t + \epsilon_{t+1} \quad (3)$$

- Equation (1) implies that C_t depends on the *expectation* of C_{t+1}.

Why Computational Methods for Macroeconomics

Example: RBC Model

- A three-equation real business cycle (RBC) model:

$$C_t^{-1} = \beta E_t \left[C_{t+1}^{-1} \left(\alpha A_{t+1} K_{t+1}^{\alpha-1} + 1 - \delta \right) \right] \quad (1)$$

$$C_t + K_{t+1} = A_t K_t^{\alpha} + (1-\delta) K_t \quad (2)$$

$$\log A_{t+1} = \rho \log A_t + \epsilon_{t+1} \quad (3)$$

- Equation (1) implies that C_t depends on the *expectation* of C_{t+1}.

- But Equation (1) also implies that the *expectation* of C_{t+1} depends on the *expectation* of C_{t+2} and so on.

Why Computational Methods for Macroeconomics

Example: RBC Model

- So computing C_t requires also computing the expected path of $C_{t+1}, C_{t+2}, C_{t+1}, \ldots$ given A_t and K_t.

Why Computational Methods for Macroeconomics

Example: RBC Model

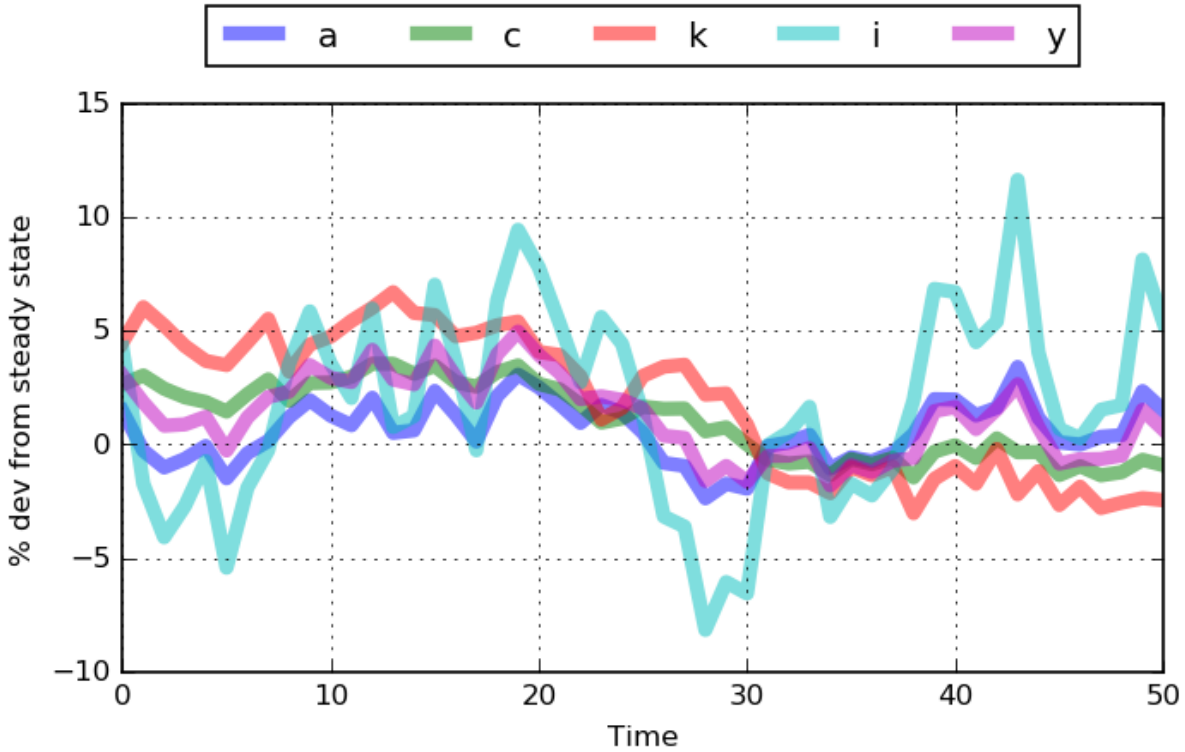
- So computing C_t requires also computing the expected path of $C_{t+1}, C_{t+2}, C_{t+1}, \ldots$ given A_t and K_t.
- Solving the problem *analytically* (i.e., exactly with paper and pencil) would take pages of calculations and is error prone.

Why Computational Methods for Macroeconomics

Example: RBC Model

- So computing C_t requires also computing the expected path of $C_{t+1}, C_{t+2}, C_{t+1}, \ldots$ given A_t and K_t.
- Solving the problem *analytically* (i.e., exactly with paper and pencil) would take pages of calculations and is error prone.
- Programmed properly, a computer can solve the problem *numerically* (i.e., approximately) in less than a second.

Figure 1: **Simulated RBC model.**



Why Python

- Macroeconomists have a lot of options:
  1. Lower level, compiled languages: Fortran, C/C++
  2. Higher level, scripting languages: Python, Matlab, Octave, Julia, R, Stata

Why Python

- Macroeconomists have a lot of options:
  1. Lower level, compiled languages: Fortran, C/C++
  2. Higher level, scripting languages: Python, Matlab, Octave, Julia, R, Stata

- The compiled languages execute more quickly, but the time investment to write code is greater.

Why Python

- Among the scripting languages, Python has a lot of advantages:

Why Python

- Among the scripting languages, Python has a lot of advantages:

  1. Free and open source

Why Python

- Among the scripting languages, Python has a lot of advantages:

  1. Free and open source
  2. Broad and active user base

Why Python

- Among the scripting languages, Python has a lot of advantages:

1. Free and open source
2. Broad and active user base
3. Many high quality libraries for numerical and statistical computing

Why Python

- Among the scripting languages, Python has a lot of advantages:

1. Free and open source
2. Broad and active user base
3. Many high quality libraries for numerical and statistical computing
4. Versatile: Numerical applications, web scraping, email management, web development

Why Python

- Among the scripting languages, Python has a lot of advantages:

  1. Free and open source
  2. Broad and active user base
  3. Many high quality libraries for numerical and statistical computing
  4. Versatile: Numerical applications, web scraping, email management, web development

- Versatility means that even if you don't become a researching macroeconomist or a financial engineer, you can still use Python.

Why Python

- I use regularly use Python for purposes other than research:
  - Updating graphs in lecture slides automatically

Why Python

- I use regularly use Python for purposes other than research:
  - Updating graphs in lecture slides automatically
  - Making animated videos for teaching purposes (e.g., https://youtu.be/SdAuHSKtpmk)

Why Python

- I use regularly use Python for purposes other than research:
  - Updating graphs in lecture slides automatically
  - Making animated videos for teaching purposes (e.g., https://youtu.be/SdAuHSKtpmk)
  - Sending personalized emails to students

Why Python

- I use regularly use Python for purposes other than research:
  - Updating graphs in lecture slides automatically
  - Making animated videos for teaching purposes (e.g., https://youtu.be/SdAuHSKtpmk)
  - Sending personalized emails to students
  - Completing the DMCA takedown request form on Course Hero website

Why Python

- I use regularly use Python for purposes other than research:
  - Updating graphs in lecture slides automatically
  - Making animated videos for teaching purposes (e.g., https://youtu.be/SdAuHSKtpmk)
  - Sending personalized emails to students
  - Completing the DMCA takedown request form on Course Hero website
  - Designing a mural for a wall in my house.

Figure 2: **Mural: design concept.** Colors are taken from the Sherwin Williams color catalog.

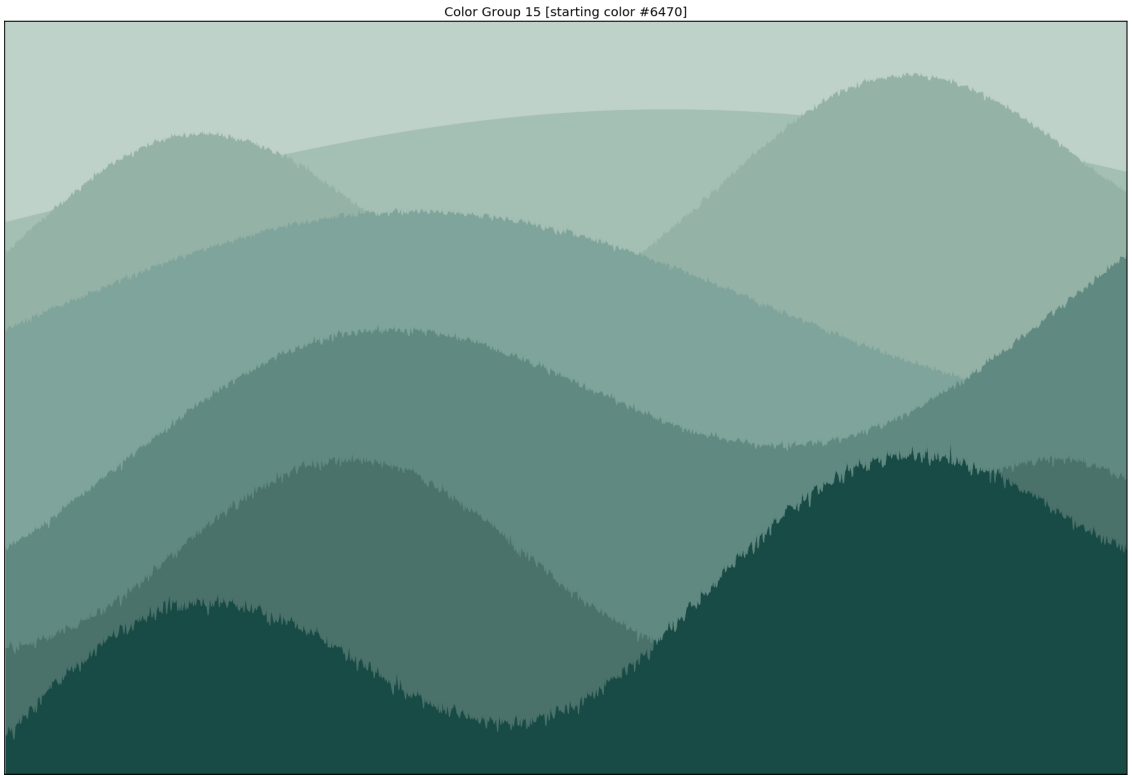


Figure 3: **Mural: key coordinates.**

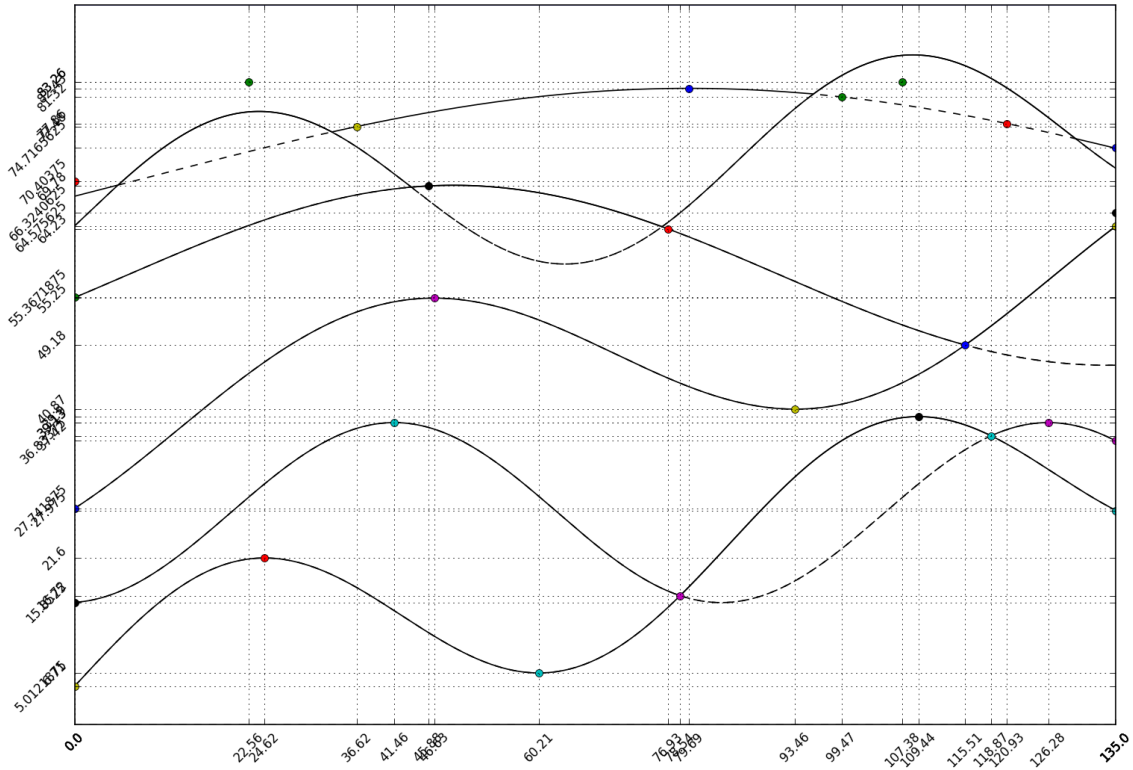


Figure 4: **Mural: final result.**

Overview of the Course

- In this course you will learn:

Overview of the Course

- In this course you will learn:
  - Basic Python programming skills

Overview of the Course

- In this course you will learn:
  - Basic Python programming skills
  - How to use Python to manage data and do basic statistics including linear regression

Overview of the Course

- In this course you will learn:
  - Basic Python programming skills
  - How to use Python to manage data and do basic statistics including linear regression
  - How to use Python to simulate linear dynamic models.

Overview of the Course

- In this course you will learn:
  - Basic Python programming skills
  - How to use Python to manage data and do basic statistics including linear regression
  - How to use Python to simulate linear dynamic models.
  - How to approximate nonlinear

Overview of the Course

- In this course you will learn:
  - Basic Python programming skills
  - How to use Python to manage data and do basic statistics including linear regression
  - How to use Python to simulate linear dynamic models.
  - How to approximate nonlinear
  - The basics of the real business cycle (RBC) and new Keynesian modeling approaches

Overview of the Course

- In this course you will learn:
  - Basic Python programming skills
  - How to use Python to manage data and do basic statistics including linear regression
  - How to use Python to simulate linear dynamic models.
  - How to approximate nonlinear
  - The basics of the real business cycle (RBC) and new Keynesian modeling approaches
  - Critiques and criticisms of both business cycle modeling approaches

Overview of the Course

- The course presumes no programming experience
- My philosophy is that coding is like cooking: it's often sufficient to learn just what you need to know in order to make what you're trying to make