



## ScPo-CompEcon Syllabus

- **Course:** Computational Economics for PhDs
- **Teacher:** Florian Oswald, [florian.oswald@sciencespo.fr](mailto:florian.oswald@sciencespo.fr)
- **Class Times:** Fridays 10:15-12:15 starting 29 Jan 2021
- **Class Location:** Zoom
- **Slack:** There will be a slack channel for all communication

### Course Description

This is a course for PhD students at the Department of Economics at Sciences Po in Computational Economics.

### Course Overview

In this course you will learn about some commonly used methods in Computational Economics. These methods are being used in all fields of Economics. The course has a clear focus on applying what you learn. We will cover the theoretical concepts that underlie each topic, but you should expect a fair amount of *hands on* action required on your behalf. In the words of the great Che-Lin Su:

Doing Computation is the only way to learn Computation. Doing Computation is the only way to learn Computation. Doing Computation is the only way to learn Computation.

True to that motto, there will be homeworks for you to try out what you learned in class. There will also be a term paper. It will be helpful to bring a laptop to the sessions if you have one.

Topics will be demonstrated through live-code examples/slides, available at <https://scpo-compecon.github.io/CoursePack/>.

### Prerequisites

1. You need a laptop.
2. You should be familiar with the material from *Introduction to Programming* taught by Clement Mazet in M1. Check out the materials [here](#)
3. You must sign up for a free account at [github.com](https://github.com). Choose a reasonable user name and upload a profile picture.
4. **Before** you come the first class, please do this:
  1. Download the latest stable `julia` release for your OS.
  2. Download the `VSCoDe` Editor

## Getting Programming Skills

1. Check out Clement Mazet's materials. You must know this level.
2. We will be using Julia for this course.
  - Noteworthy Differences from Other Languages
  - MATLAB, Python, Julia Syntax Comparison
3. Clement in his course will introduce you to things like the Unix Shell and the version control system Git. Both of those are very useful - for this course, and for the rest of your life as a scientist.
4. What is Version Control? watch this 5 minute video. and go back to Clement's stuff if unclear.

## Homeworks

There will be homeworks. They will be listed within the Course Outline.

## Term Project

This year your term project will be to replicate a paper published in an economics journal. Ideally this would be related to your field of interest. The requirements for choice of paper to replicate are:

1. Published version and replication kit is available online.
2. The paper to replicate must not use julia.
3. You must use julia for your replication.
  - Ideally your choice will involve at least some level of computational interest (i.e. more than an IV regression)
  - However, you can replicate a paper with an IV regression, but you have to go all the way to get the exact same results as in the paper. I.e. if the author typed the stata command `ivreg2 lw s expr tenure rns smsa _I* (iq=med kww age), cluster(year)` you will have to write (or find) julia code which will match all output from this, including standard errors. I do not recommend to do this.
4. You need to set up a public github repository where you will build a documentation website of your implementation. You'll learn how to do this in the course.
5. I encourage you to let the world know about your replication effort via social media and/or email to the authors directly. This is independent of whether you were able or not to replicate the results. Replication is not about finding errors in other peoples' work. If you are able to replicate some result in julia, this may be very interesting for others.

## Grade

Your grade will be 60% homeworks, 40% term project.

## Textbooks

There are some excellent references for computational methods out there. This course will use material from

- **Fackler and Miranda** (2002), Applied Computational Economics and Finance, MIT Press
- **Kenneth Judd** (1998), Numerical Methods in Economics, MIT Press
- **Nocedal, Jorge, and Stephen J. Wright** (2006): Numerical Optimization, Springer-Verlag
- **Kochenderfer and Wheeler** (2019), Algorithms for Optimization, MIT Press
- **A Gentle Introduction to Effective Computing in Quantitative Research** - What Every Research Assistant Should Know, Harry J. Paarsch and Konstantin Golyaev

## Course Schedule

### 1. Programming languages and why julia

1. Talk through homework requirements
2. Talk through term project requirements
3. Show where material is and do first set of slides.

### Homework

- Sign up to github.com.
- Sign up for introduction to github and send me a screenshot of all completed issues.
- Make a pull request.

### Optional Julia Workouts

- <https://exercism.io/> offers some simple coding exercises. I highly recommend those if you want to get some extra practice - particularly in the beginning!

---

### 2. julia setup and Getting Started

- Setup environment
- Tools and Editors
- Examples
- Types
- Essentials
- Speed
- Data and Statistical Packages

---

### 3. Integration and Function Approximation

1. Numerical Integration
    - Monte-Carlo integration
    - Gaussian Quadrature
    - Multidimensional Quadrature
      - Quadrature with correlated shocks
  2. Function Approximation
    - Polynomial Interpolation
      - Basis functions and Coefficients
    - Regression as Approximation
    - Colocation Methods
    - Multidimensional Approximation
      - The Smolyak Grid
- 

### 4. Optimisation 1

1. Intro
  2. Conditions for Optima
  3. Derivatives and Gradients
  4. Numerical Differentiation
  5. JuliaOpt
- 

### 5. Optimisation 2

1. Bracketing
  2. Local Descent
  3. First/Second Order and Direct Methods
  4. Constraints
- 

### 6. Numerical Dynamic Programming

- Review of DP theory
  - Different Solution methods for different cases
    - Discretization
    - Parametric approximation methods basically Function Approximation
    - The Endogenous Grid Method
    - Finite time vs infinite horizon models
  - Solving the Growth Model in 7 Different ways
-

## 7. Constrained Optimisation Applications as MPECs

- What is an MPEC?
- How can we cast constrained problems as MPECs?

Applications:

1. MPEC on John Rust's Bus Engine Replacement
  2. The Berry-Levinsohn-Pakes (BLP) paper as constrained optimization problems
- 

## 8. HPC julia

- Brief intro to parallel computing concepts
  - Parallel computing with julia
  - GPU computing with julia
- 

## 9. Rust Bus Model

---

## 10. Intro to Machine Learning with julia

---

## 11. The julia ML stack

---

## 12. julia ML applications

---

## Statements on Plagiarism

We will try to honour Science Po's anti-plagiarism policy:

Plagiarism occurs when a student submits work that does not allow one to distinguish the student's own thoughts from those of other authors: it can be characterised by the absence of citation of a group of consecutive words (five or more), by reformulation or translation, or by copying directly." (article on intellectual honesty)

## **Related to Plagiarism**

Reuse and building upon ideas or code are major parts of modern software development. As an economist writing code, you will (hopefully) never write anything from scratch. This class is structured such that all solutions are public. You are encouraged to learn from the work of your peers. As I said above, I won't hunt down people who are simply copying-and-pasting solutions, because without challenging themselves, they are simply wasting their time and money taking this class.

Please respect the terms of use and/or license of any code you find, and if you reimplement or duplicate an algorithm or code from elsewhere, credit the original source with an inline comment.

## **License**

I took the setup for the structure of this course from <https://github.com/advanced-js> taught by Aidan Feldman and team at NYU, and I would like to thankfully acknowledge making the materials useable to other teachers. The same license applies (below). The coursepack material is based on Chris Rackauckas' excellent Julia Intro. The license allows you to copy and use everything here, under the condition that you attribute the work (details in the license). The copyright notice to be included in any such copies and other derivative work is:

Copyright 2020 Florian Oswald, Sciences Po Paris, [florian.oswald@gmail.com](mailto:florian.oswald@gmail.com)

Thank you.

This work and all other materials under <https://github.com/ScPo-CompEcon> are licensed under a Creative Commons Attribution 4.0 International License.