

COMPUTATIONAL ECONOMICS FOR PHD STUDENTS

Quarter 2, November-December 2021

Instructor:	Florian Oswald	Time:	Tue+Thu 10:20-11:50
Email:	florian.oswald@sciencespo.fr	Place:	Università Bocconi

Course Page: <https://floswald.github.io/NumericalMethods/> This website has all relevant info and required material, so please have a look at it to get a good overview of what we do in this course.

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.

Course Objective: *To take the fear out of computation.* In this course I want to help you to develop your computational skills. I will give you tools that are relatively easy to use, sometimes even fun (!) to use. Key to this endeavour is the julia programming language, which is both performant (you can use it for *real* work), and at the same time easier to use than more traditional languages like FORTRAN or C++. While we will spend a good amount of time on the *usual suspects* like Dynamic Programming and Optimization, I want to give a good overview of the available methods out there.

Office Hours: By appointment

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

Term Project: 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:

- Published version and replication kit is available online.
- The paper to replicate must not use the julia language.
- You must use the julia language for your replication.

- Ideally your choice will involve at least some level of computational interest.
- 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.

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.

There is more detail and resources on the course website at https://floswald.github.io/NumericalMethods/#term_project

Prerequisites:

1. You should be familiar with the material from Introduction to Programming taught by Clement Mazet in M1. Check out his material at <https://cms27.github.io/teaching/>
2. You must sign up for a free account at github.com. Choose a reasonable user name and upload a profile picture.
3. Before you come to the first class, please download the latest stable julia release from <https://julialang.org>
4. You **must** know what the UNIX shell (or windows command line) is. Clement's course above, if not.
5. You **must** know what version control is. Watch [this video](#) and go to Clement's course above, if not.

It is natural that some students have better programming skills than others because of previous exposure and inclination. This course takes this into account. For example at SciencesPo, where I usually teach this course, the entire cohort of PhDs signed up, with a lot of heterogeneity in skills. Feel free to send me an email for specific questions in this regard.

Tentative Course Schedule: Please consult https://floswald.github.io/NumericalMethods/#course_schedule

Grading: Homeworks (60%), Final Project (40%).