

# PhD-teach-PhD: Approaching science with AI

## An introduction to scientific machine learning

Guilherme Zagatti

June 2021

Dynamical systems are a common way of describing phenomena in science. From harmonic oscillators, infectious diseases, ecology to market dynamics, all of these phenomena can be described with sets of ordinary equations that detail how the different variables in the system change with time and affect one another. To develop good models that explain and extrapolate empirical data, scientists need lots of observation and good intuition, as well as a deep understanding of their field of expertise. Conversely, machine learning poses an alternative approach to research: data in, predictions out. This black-box approach eschews generalization, leaving little room for explanation and understanding.

Recently, a new paradigm that blends scientific methods with machine learning called scientific machine learning has emerged. Using the fact that neural networks are universal function approximators, we can blend neural networks with ordinary differential equations to facilitate explainable model discovery from data. In particular, a new breed of neural network called NeuralODE can help us make sense of non-linear dynamics. This course will introduce scientific machine learning using the Julia programming language and its SciML environment.

First, the workshop will introduce scientific machine learning, dynamical systems and Julia. It will show the prevalence of dynamical system in sciences with illustration of simple but classical models from a variety of fields. After motivating our workshop, we will provide a brief introduction to Julia to equip the students with the essential tools which will allow them to follow the workshop and complete the exercises. At the end of this section, the students should know basic applications of dynamical systems in science and be able to setup a Julia working environment.

The second part of the course will dig deeper in the topic of ordinary dynamic equations (ODEs). We will formally introduce the initial value problem and characterize the solution of this problem. We will then discuss classical methods that are employed to fit observations to models of dynamical systems. At the end of this part, the students should have a basic understanding of ODEs and know the difference between solving the initial value problem and fitting data to it.

After presenting classical optimization methods, the workshop will move on to neuralODEs. We will re-formulate the initial value problem in terms of a neural network. We will discuss the properties of neural networks that make it ideal for solving dynamical systems. Next, we will introduce neuralODEs. At the end of this part, the students should know when it might be appropriate to use neural networks in an ODE. The students should also be able to describe basic application of neuralODEs to other problems in machine learning and know the main limitations of this method.

In the third part of the course the workshop will proceed to practical exercises. The students will be invited to download a Jupyter notebook with pre-filed instructions. Given the limited amount of time and the on-line format of the workshop, the instructor will allow some time

for the student to attempt the exercise in groups. They will then present to others. By the end of the course, the students should be able to follow along a basic problem and know where resources are available to go further in this field.

Given these objectives the lesson plan will proceed as following:

1. Part I, duration 1h: Motivation and Julia. Lecture with audience interaction (eg. questions and discussions).
2. Part I, duration 30min: Julia environment setup.
3. Part II, duration 1.5h: Initial value problem and classical fitting. Lecture with audience interaction (eg. questions and discussions).
4. Part II, duration 1.5h: NeuralODEs. Lecture with audience interaction (eg. questions and discussions).
5. Part III, duration 2h: Hands-on exercise. The students will be split in two groups. A Jupyter notebook with pre-filled code will be provided with a motivating example. Given the limited amount of time, the students will only be required to make minimal input that will illustrate particular features of scientific machine learning. The students will then present to others.

This course will borrow from the large number of references available online, among them:

- Julia Computing. SciML Scientific machine learning software. 2021. <https://sciml.ai>
- Julia Computing. SciMLTutorials.jl: Tutorials for scientific machine learning and differential equations. 2021. <https://tutorials.sciml.ai>
- Rackauckas, Christopher. MIT 18.337J/6.338J: Parallel computing and scientific machine learning. Online Course. 2020. <https://github.com/mitmath/18337>
- Rackauckas, Christopher. Doing scientific machine learning (SciML) with Julia. Julia COn2020. 2020. <https://www.youtube.com/watch?v=QwV00Xh2Hbg>
- Rackauckas, Christopher. Universal Differential Equations for SciML. University of Arizona Modeling and Computation Seminar. 2020. <https://www.youtube.com/watch?v=5zaB1B4h0nQ>
- Sanders, David P. and Edelman, Alan. MIT 6.S083/18.S190: Introduction to computational thinking with Julia, with applications to modelling the COVID-19 pandemic. Online course. 2020. <https://github.com/mitmath/6S083>