

## **Julia for the working scientist**

**Speaker: Mathieu Besançon (The Zuse Institute Berlin)**

### **Part I: motivation and introduction**

Scientific programming comes with a set of unique challenges, with the need for flexibility and interactivity on one hand, and for performance to tackle large-scale problems. Julia is a modern programming language bridging the gap between interactivity and performance, making it well-suited for scientists. We will make a tour of the features of the language and ecosystem, covering the tools for scientific-oriented development, the programming model, and some applications in optimization and engineering. This tutorial will not require prior knowledge of the language, beginners are welcome.

## **Julia for the working scientist**

**Speaker: Mathieu Besançon (The Zuse Institute Berlin)**

### **Part II: structured optimization in JuMP**

JuMP is an Algebraic Modelling Languages (AMPL, PuLP, Pyomo) embedded in Julia as a library. It brings a syntax close to the mathematical expression of the optimization problem while benefiting from the flexibility of the host language. We will cover some examples from linear, mixed-integer, conic optimization, and see how JuMP can be extended to create new sorts of constraints.