Course syllabus

Course content. This course is aimed at giving a first introduction to classical topics in numerical analysis, including floating point arithmetics and round-off errors, the numerical solution of linear and nonlinear equations, iterative methods for eigenvalue problems, interpolation and approximation of functions, and numerical quadrature. If time permits, we will also cover numerical methods for solving ordinary differential equations.

Prerequisites. The course assumes a basic knowledge of linear algebra and calculus. Prior programming experience in Julia, Python or a similar language is desirable but not required.

Study goals. After the course, the students will be familiar with the key concepts of *stability*, *convergence* and *computational complexity* in the context of numerical algorithms. They will have gained a broad understanding of the classical numerical methods available for performing fundamental computational tasks, and be able to produce efficient computer implementations of these methods.

Education method. The weekly schedule comprises two lectures (2× 1h15 per week) and an exercise session (1h30 per week). The course material includes rigorous proofs as well as illustrative numerical examples in the Julia programming language, and the weekly exercises blend theoretical questions and practical computer implementation tasks.

Assessment. Computational homework will be handed out on a weekly or biweekly basis, each of them focusing on one of the main topics covered in the course. The homework assignments will count towards 70% of the final grade, and the final exam will count towards 30%.

Literature and study material. A comprehensive reference for this course is the following textbook: A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*, volume **37** of *Texts in Applied Mathematics*. Springer-Verlag, Berlin, 2007. Other pointers to the literature will be given within each chapter.