

Computational Engineering

Description

The course is a hands-on introduction to the numerical solution of linear and nonlinear systems, optimization, and ordinary and partial differential equations. The emphasis is on understanding fundamental aspects of numerical methods and the challenges associated with their implementation and validation. After the course, the students are able to find the methods in standard libraries that are more suitable for a particular problem, and they can assess the trade-offs between accuracy and cost. The course is taught in English. The evaluation is based on take-home assignments, where the students have to write python codes to solve specific problems with the algorithms discussed in class.

Contents

1. Introduction.

- Computational engineering.
- Python3. Optionally, Spyder3 as programming environment.

2. Linear systems.

- Direct methods. Operation count. LU factorization. Fill-in.
- Iterative methods. Stationary methods (Jacobi, Gauss-Seidel) and non-stationary methods (steepest descent, conjugate gradient).

3. Nonlinear systems and optimization.

- Non-linear systems. Newton's method.
- Optimization. Line-search methods.

4. Ordinary differential equations.

- Finite difference approximations. Taylor tables. Error analysis. Boundary value problems.
- Time-marching schemes. Explicit and implicit schemes. Error and stability analysis. Initial value problems.

5. Partial differential equations.

- Finite difference methods. Semi-discrete approach (method of lines). Error and stability analysis: CFL number and diffusion number.
- Multidimensional problems. Operator splitting.
- Introduction to finite volume methods.
- Introduction to finite element methods.