Aims

When solving problems in science, engineering or economics, a real-life situation is first converted into a mathematical model. This is often called the formulation of the problem and it is given in terms of mathematical equations. Only a handful of model equations can be solved in a neat analytical form. Hence we need numerical analysis, comprising a set of techniques for finding approximate solutions of these equations. This course provides an introduction to very basic methods in numerical analysis both from a theoretical and a practical perspective. It also provides an introduction to programming the scientific computing package Matlab.

Syllabus

**Computer arithmetics:** Number bases. Floating point numbers. Significant figures and digits. Round off errors. Absolute, relative and percentage errors. Floating point arithmetics. (3 lectures)

**The elements of Matlab:** The different Matlab environments. M-files and M-functions. Variables, data types and internal Matlab functions. Text strings. The “If” conditional and logical pipelines. “For” loops. (3 lectures)

**The basics of Matlab Programming:** The “While” loop. Vectors and matrices in Matlab. Operations with vectors and matrices in Matlab. Programming practices. Debugging, error messages and using the editor to change existing files. (3 lectures)

**Solving algebraic equations numerically:** The bisection method. Convergence of the bisection method. The regula falsi methods. Bisection and regula falsi algorithms with different stopping criteria in Matlab. (3 lectures)

**Solving smooth non-linear scalar equations:** Newton’s method. Fixed point iteration. Taylor’s Theorem. Order of convergence. Order of convergence for fixed point iteration. (3 lectures)

**Analysis of orders of convergence for fixed point iteration:** How to terminate fixed point iterations in practice. Multiple roots of a functional equation. The Newton method for multiple roots. The Newton method in Matlab. Fixed point iterations in Matlab and the logistic map. (3 lectures)

**Interpolation:** Polynomial interpolation. Direct method. Lagrange polynomials. Newton polynomials. Interpolation error and Chevyshev points. Interpolation in Matlab. (3 lectures)

**Numerical Integration:** Simple trapezoidal, midpoint and Simpson’s rules. Derivation of the simple integration rules. The composite trapezoidal, midpoint and Simpson’s rules. Matlab integration with applications to solving different practical problems. (3 lectures)
Convergence of the numerical integration methods: Convergence of the simple rules. Convergence of the composite rules. Minimal number of quadrature points for accuracy. Order of magnitude of the error in computing integrals. (3 lectures)

Numerical differentiation: Forward, backward, central and five-point differentiation. Error formulas for differentiation methods. Higher order derivatives. Differentiating in Matlab and applications. (3 lectures)

Teaching and Assessment

Contact Hours: 3 lectures + 1 tutorial or lab session per weeks
Assessment: 30% by class tests or other continuous assessment
70% by end of course 2-hour exam
Resit Type: 2 hours resit exam in semester 3
By the end of the course, students should be able to:

- Understand basic methods of numerical analysis and the numerical approximation of solution to mathematical problems.
- Become familiar with the concept of numerical approximation.
- Use mathematical techniques which are required to approximate the solution of single non-linear equations.
- Use mathematical techniques which are required to interpolate data.
- Use mathematical techniques which are required to approximate integrals.
- Use mathematical techniques which are required to approximate derivatives.
- Become familiar with the basics of the computer package Matlab.
- Understand the concept of conditionals in programming.
- Understand the concept of iteration in programming.
- Solve specific problems from the applied sciences using Matlab.

Content: January 2018