Computer Algebra

Compact course

Beuth Hochschule für Technik Berlin

Computational Engineering

Lecturer: dr. János Karsai associate professor, University of Szeged

Length: 5x8 hours in computer cabinet

Language: English

Program

<u>Part I: Application based introduction</u>

Basic concepts and tools

- The basic concepts of *Mathematica*: Front-End, Kernel, typesetting, cells, formulas, Help
- Basic structures and operations numerical and symbolic operations, variables, algebraic manipulations
- Lists, vectors and arrays
- Value setting (immediate and delayed), substitution rules, patterns, functions
- Solving equations
- *Notebook operations:* styles, stylesheets.

Graphics and visualization in 2D:

- Plots for functions and lists
- Parametric curves
- Elements of visualization: Animations, coloring, zooming

Built-in Graphics and visualization functions in 3D:

- Functions of two variables
- 3D parametric curves and surfaces
- Scalar fields in 3D
- Vector fields

Dynamic Evaluation, interactive graphics

Linear Algebra *in Mathematica*

Calculus 1D to nD:

- Graphical, experimental study of the properties of functions, parametric lines and surfaces, scalar fields, vector fields
- Limits, derivatives, integral, series expansions
- Tangent and normal vectors, planes
- Maxima-minima, zeros

Elementary data handling, statistics (optional)

- Experimental data, plotting data, data transformations
- Curve fitting
- Presentation graphics

Differential equations

- ODE summary, simple tools to solve and visualize ODE’s

Part II: Elements of programming in *Mathematica*

Object types, assignments, functions

- Structures, types, Head, Head operations, type check, logical functions
- Assignments, substitution rules (immediate and delayed)
- Function definitions in details
- Patterns: parameter type-check
- Piecewise or conditional definition of functions, recursions

List programming

- Rule-based programming
- Structure operations on lists: Map, Apply, Thread, ...
- Rotating lists, and applications to numerical algorithm

Graphics programming I: structures and operations

- Graphics in 2D and 3D
- How the built-in plots work
- Applications of structure and rule-based programming to graphics objects
- Some advanced applications to scientific and engineering visualizations: functions, vector fields and scalar fields

Iteration, nesting

- Recursion vs. iterations
- Iterations, fixed points of mappings
- Numerical applications: Newton iteration, gradient method, Euler method to solve ODE's, etc,

Graphical programming II: Iterative forms, fractal constructions (optional)

- Simple constructions
- Generating trees, Sierpinsky triangles, the midpoint rule

Programming paradigms in Mathematica: a systematic overview

- Procedural programming
- Functional programming
- Rule-based programming

Applications to difference systems

- Solving, visualizations
- cobweb diagram, bifurcation diagram
- Discretization of ODE's, PDE's

Advanced applications to differential equations

- The phasemap and Ljapunov's methods: a visual approach
- Differential systems with Dirac delta
- Poincare maps

Writing packages

- Package design, a general overview
- Structure-based functional programming
- Using variable names as parameters
- Handling options

Additional topics

- Advanced notebook operations: options, option inspector
- Stylesheet design, automatic numbering, hyperlinks, ...
- Export, import: HTML, XML, MathML, TeX, XLS, …