Computer-aided methods to study functions and dynamical systems
(Programming in *Mathematica*)
Detailed program of the Ph.D. course
Mathematics doctoral school
University of Szeged

Length: 5x8 hours in computer cabinet

Lecturer: János Karsai Ph.D. associate professor, Department of medical Informatics, University of Szeged,

Goals: The student learn the basic functions and the programming tools of *Mathematica* to implement mathematical objects, algorithms and constructions emphasizing the differential equations and dynamic systems. The students also get acquainted with the concepts and basic methods of computer-aided experimentation and modeling.

Method: After the overview of the given topic, the teacher and students together (or simultaneously develop applications. Complete Mathematica courseware is available for the students.

Evaluation: The grade is based on the projects developed by the students during the course.

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, meshing.
- Dynamic features.

***Linear Algebra** in Mathematica*

Built-in Graphics and visualization in 3D:

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

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

<u>Part II: Elements of programming in *Mathematica*</u>

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 : Map, Apply, Thread, ...
- Rotating lists, and applications to numerical algorithm

Graphics programming I: structures and operations

- Graphics in 2D and 3D.
- Normal structures, GraphicsComplex, version 5 structures.
- 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, fixedpoints 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 Liapunov'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

Notes - handouts

[1] Karsai J.: Computer-algebra, TFH-Berlin, CD-ROM, 2007

[2] Karsai J.: *Mathematica*l and visualizaton packages: *Mathematica-6, CD-ROM, 2008.*

[3] Karsai J.: Computer-aided study of mathematical models, *CD-ROM, 2008.*

Recommended literature:

[4] Karsai J., Models of impulsive phenomena, mathematica experiments, Typotex Budapest, 2002, CD-Rom updated to Mathematica 6, 2008..

[5] Szili L. , Tóth J., Matematika and Mathematica, Eötvös Kiadó, 1996.

[6] Beltrami, E.: *Mathematics for Dynamic Modeling*, Academic Press, 1998.

[7] Davis B., Porta H., Uhl J.: *Calculus & Mathematica*, Addison-Wesley, 1994.

[8] Ghanza, V. G., Vorozhtsov E. V.: *Numerical Solutions for Partial Differential Equations. Problem Solving Using Mathematica*, CRC Press, 1996

[9] Gaylord R. J., Wellin P. R.: *Computer Simulations with Mathematica*, Telos-Springer, 1995.

[10] Gaylord R. J., Nishidate K., *Modeling Nature, Cellular Automata Simulations with Mathematica,* Telos-Springer, 1996.

[11] de Jong, M. L., *Mathematica For Calculus-Based Physics*, Addison-Wesley, 1999.

[12] Kaplan, D., Glass, L.: *Understanding Nonlinear Dynamics*, Springer, 1995.

[13] Maeder, R. E.: *The Mathematica Programmer I-II*, Academic Press, 1996

[14] Meerrschaert, M.M.: *Mathematical Modelling*, Academic Press, 1999

[15] Murray, D. J.: Mathematical Biology, Springer, 1997.

[16] W.T. Shaw, Complex Analysis with *Mathematica*, Cambridge Univ. Press, 2006.

[17] Wolfram, S.: *Mathematica*, Addison-Wesley Publishing Company, 2005.

[18] Wolfram, S.: *A New Kind of Science*, Wolfram Media, Inc., 2002.