Module Handbook

Ruprecht-Karls-Universität Heidelberg
Fakultät für Mathematik und Informatik
International Master Scientific Computing

Fassung vom 26.01.2015 zur Prüfungsordnung vom 16.03.2009
mit letzter Änderung vom 22.04.2013
Version as of 26.01.2015 corresponding to examination regulations of 16.03.2009
with changes of 22.04.2013

Studienform/form of study: Vollzeit/full time

Art des Studiengangs/type of study: konsekutiv/consecutive

Regelstudienzeit/regular period of study: 4 Semester/4 semesters

Einführungsdatum/date of begin: 22.04.2013

Studienstandort/location of study: Heidelberg

Anzahl der Studienplätze/Number of places: derzeit keine Begrenzung/currently no limitation

Gebühren/Beiträge/fee: gemäß allgemeiner Regelung der Universität Heidelberg/according to general regulations of Heidelberg University

Weitere Informationen/Additional information:

`http://www.scicomp.uni-hd.de`

Präambel
Qualitätsziele der Universität Heidelberg in Studium und Lehre

Anknüpfend an ihr Leitbild und ihre Grundordnung verfolgt die Universität Heidelberg in ihren Studiengängen fachliche, fachübergreifende und berufsfeldbezogene Ziele in der umfassenden akademischen Bildung und für eine spätere berufliche Tätigkeit ihrer Studierenden. Das daraus folgende Kompetenzprofil wird als für alle Disziplinen gültiges Qualifikationsprofil in den Modulhandbüchern aufgenommen und in den spezifischen Qualifikationszielen sowie den Curricula und Modulen der einzelnen Studiengänge umgesetzt:

- Entwicklung von fachlichen Kompetenzen mit ausgeprägter Forschungsorientierung;
- Entwicklung transdisziplinärer Dialogkompetenz;
- Aufbau von praxisorientierter Problemlösungskompetenz;
- Entwicklung von personalen und Sozialkompetenzen;
- Förderung der Bereitschaft zur Wahrnehmung gesellschaftlicher Verantwortung auf der Grundlage der erworbenen Kompetenzen.

Fachliche und überfachliche Qualifikationsziele des internationalen Masterstudiengangs Scientific Computing

Der internationale Masterstudiengang Scientific Computing hat das Ziel einer Erweiterung der mathematischen und informatischen Grundkenntnisse sowie einer Vertiefung, die bis zum Kontakt mit aktueller Forschung in einem der in Heidelberg vertretenen Gebiete reicht. AbsolventInnen des Masterstudiengangs sind in der Lage, Methoden der Mathematik und Informatik anzuwenden, diese selbständig weiterzuentwickeln und effizient zu implementieren. Der Einsatz mathematischer und informatischer Methoden für konkrete Anwendungsprobleme stellt einen wesentlichen Teil der Qualifikation dar. Durch die Anfertigung einer Masterarbeit werden in sehr großem Maße die Fähigkeiten zur selbständigen wissenschaftlichen Arbeit, zur Problemanalyse und -lösung sowie auch zur Organisation von Arbeit gestärkt.

Preamble
Qualification Goals in Teaching of Heidelberg University

Following its mission statement and its charter Heidelberg University pursues in its degree courses disciplinary, interdisciplinary and professional goals in the comprehensive academic education and for the subsequent professional career of its students.

The resulting profile of competence is to be included in the module handbooks of all degree programs and will be implemented in all specific qualification goals as well as the curricula and modules of the individual degree programs:

- Development of disciplinary competence with specific orientation towards research;
- Development of transdisciplinary competence for dialogue;
- Buildup of practical problem solving competence;
- Development of personal and social competence;
- Promotion of the willingness to take on social responsibility on the basis of the acquired competences.

Disciplinary and interdisciplinary goals of qualification of the international master program “Scientific Computing”

The international master program “Scientific Computing” aims at expanding proficiency in mathematics and computer science towards topics relevant in cutting-edge research at Heidelberg University. Graduates of the master program are able to apply methods of mathematics and computer science, advance these methods independently as well as implement them efficiently. Using methods and techniques of mathematics and computer science to solve practical applications represents a major part of the qualification. Preparation of a master thesis fosters to a very high degree the ability to conduct independent research, problem analysis and solving as well as the organisation of work.

Module Description
International Master Scientific Computing

1 Complex Analysis I

Module	Code MG9	Name Complex Analysis I		
	Credit Points 8	**Workload** 240 h	**Duration** 1 term	**Cycle** WS|SS
Program	Master Scientific Computing			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	T.B.A.			
Objectives	Basic knowledge about complex spaces			
Content	• Local theory of complex spaces: differential forms, Hodge decomposition, Dolbeault theory • Basic facts about complex functions of several variables: analytic functions, Cousin problem, local rings of analytic functions, Oka's lemma, Weierstraß preparation theorem • Abelian functions			
Learning outcomes	• Ability to solve problems about complex spaces and several complex variables and to present these solutions in problem sessions			
Prerequisites	None			
Suggested previous knowledge	Complex Functions I+II (MB3+MB4)			
Assessment(s)	Homework, assignments, written or oral exam. Modalities for make-up exams to be determined by the lecturer			
Literature				

2 Complex Analysis II

Module	**Code** MG10	**Name** Complex Analysis II		
	Credit Points 8 CP	**Workload** 270 h	**Duration** 1 term	**Cycle**
Program	Master Scientific Computing			
Methods	Methods 4 h + Exercise course 2 h			
Lecturer	T.B.A.			
Objectives	Profound knowledge about complex spaces			
Content	• Analytic sheaves: theorem of Oka, coherence theorems • Finiteness and vanishing theorems for coherent sheaves and applications • Theorem A+B • Applications: mapping theorems			
Learning outcomes	• Ability to solve problems about complex spaces and several complex variables and to present these solutions in problem sessions			
Prerequisites	Complex Analysis I (MG9)			
Suggested previous knowledge				
assessments	(s. Complex Analysis I)			
Literature				

3 Differential Geometry I

Module	**Code** MG15	**Name** Differential Geometry I		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle**
Program	Master Scientific Computing			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	T.B.A.			
Objectives	Command of basic notions and methods in differential geometry. Ability to apply methods from analysis and algebra to problems in geometry.			
Content	Differentiable manifolds, (Semi-)Riemannian manifolds, connections, geodesics, curvature			
Learning outcomes	Ability to solve problems in differential geometry and to present these solutions in problem sessions			
Prerequisites	None			
Suggested previous knowledge	Mathematics Bachelor			
Assessment(s)	Homework assignments, written or oral exam. Modalities for make-up exams are to be determined by the lecturer and will be announced at the beginning of the course.			
Literature	Do Carmo: Riemannian Geometry Gallot-Hulin-Lafontaine: Riemannian Geometry Gromoll-Klingenberg-Meyer: Riemannsche Geometrie im Großen Kobayashi-Nomizu: Foundations of Differential Geometry Petersen: Riemannian Geometry Spivak: Differential Geometry			

4 Differential Geometry II

Module	Code MG16	Name Differential Geometry II		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle**
Program	Master Scientific Computing			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	T.B.A.			
Objectives	Command of certain advanced topics in differential geometry. Connection of geometric problems with wider mathematical context.			
Content	Relations between geometry and topology, symmetries Potential topics include: • geometry of spaces with upper or lower curvature bounds • symmetric spaces and homogeneous spaces • hyperbolic geometry • other topics in geometry			
Learning outcomes	Ability to solve problems in differential geometry and to present these solutions in problem sessions			
Prerequisites	None			
Suggested previous knowledge	Differential Geometry I (MG15)			
Assessment(s)	Homework assignments, written or oral exam. Modalities for make-up exams are to be determined by the lecturer and will be announced at the beginning of the course.			
Literature	Literature will be announced in the course.			

5 Computer Algebra I

Module	**Code** MG19	**Name** Computer Algebra I		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle**
Program	Master Mathematics, Master Scientific Computing			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	T.B.A.			
Objectives	To have a firm command of basic notions and methods in Computer Algebra			
Content	This lecture attends to the theory and complexity of basic mathematical algorithms and their implementations in computer algebra systems. Main topics are: • Fast arithmetic: complexity of basic operations, discrete Fourier transformation, fast multiplication, fast Euclidean algorithm, subresultants and polynomial remainder sequences, modular algorithms, computations with algebraic numbers, fast matrix multiplication • Prime factorisation and primality tests: Solovay-Strassen and Miller-Rabin primality test, AKS primality test, RSA algorithm, elementary factorisation methods, quadratic sieve, irreducibility tests for polynomials, Berlekamp algorithm, Zassenhaus algorithm, lattice basis reduction, factorisation of multivariate polynomials • Groebner Basis algorithms: Groebner Bases and reduced Groebner Bases, Buchberger algorithm, elimination theory, algorithms for elementary operations on ideals, computation of the dimension of an ideal			
Learning outcomes	Ability to solve problems in Computer Algebra and to present these solutions in problem sessions, ability to work with computer algebra systems			
Prerequisites	None			
Suggested previous knowledge	Mathematics Bachelor			
Assessment(s)	Homework assignments, written or oral exam. Modalities for make-up exams are to be determined by the lecturer and will be announced at the beginning of the course.			

Literature	J. von zur Gathen, J. Gerhard: Modern Computer Algebra O. Geddes, S. R. Czapor, G. Labahn: Algorithms for Computer Algebra D. Cox, J. Little, D. O'Shea: Ideals, Varieties and Algorithms B. H. Matzat: Computeralgebra (lecture notes)

6 Computer Algebra II

<table>
<tr><td>Module</td><td>Code
MG20</td><td colspan="3">Name
Computer Algebra II</td></tr>
<tr><td></td><td>Credit Points
8</td><td>Workload
240 h</td><td>Duration
1 Semester</td><td>Cycle</td></tr>
<tr><td>Program</td><td colspan="4">Master Mathematics, Master Scientific Computing</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 4 h + Exercise course 2 h</td></tr>
<tr><td>Lecturer</td><td colspan="4">T.B.A.</td></tr>
<tr><td>Objectives</td><td colspan="4">Command of certain advanced topics in Computer Algebra</td></tr>
<tr><td>Content</td><td colspan="4">This lecture attends to one or several of the following topics:
Algorithmic Number Theory
Algorithmic Commutative Algebra
Algorithmic Group Theory
Algorithmic Invariant Theory
Algorithmic Arithmetic Geometry</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">Ability to solve problems in Computer Algebra and to present these solutions in problem sessions, ability to work with computer algebra systems</td></tr>
<tr><td>Prerequisites</td><td colspan="4">None</td></tr>
<tr><td>Suggested previous knowledge</td><td colspan="4">Mathematics Bachelor, Computer Algebra I (MG19)
To attend the theme Algorithmic XXX, some familiarity with XXX is expected.</td></tr>
<tr><td>Assessment(s)</td><td colspan="4">Homework assignments, written or oral exam. Modalities for make-up exams are to be determined by the lecturer and will be announced at the beginning of the course.</td></tr>
<tr><td>Literature</td><td colspan="4">T.B.A. at the beginning of the course</td></tr>
</table>

7 Nonlinear Functional Analysis

Module	Code	Name		
	MH1	Nonlinear Functional Analysis		
	Credit Points	**Workload**	**Duration**	**Cycle**
	8	240 h	1 Semester	
Program	Master Scientific Computing			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	Prof. Dr. Claus Gerhardt			
Objectives	To have a firm command of the main topics in Nonlinear Functional Analysis.			
Content	• *Existence theorems*: inverse function theorem, implicit function theorem, Sard's theorem • *Fredholm operators*: Fredholm operators, Smale's version of Sard's theorem and applications. • *Topological degree theory*: Brouwer's degree theory, Leray–Schauder's degree theory, applications.			
Learning outcomes	• The capability of solving problems in the underlying field. • The capability to present the solutions in the exercise courses.			
Prerequisites	none			
Suggested previous knowledge	Partial Differential Equations, Functional Analysis			
Assessment(s)	Homework consisting of solving exercises and a written or oral examination at the end of the term.			
Literature	Louis Nirenberg: Topics in Nonlinear Functional Analysis Melvyn S. Berger: Nonlinearity and Functional Analysis Claus Gerhardt: Analysis II Claus Gerhardt: Curvature Problems			

8 Partial Differential Equations II

Module	**Code** MH3	**Name** Partial Differential Equations II		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle**
Program	Master Scientific Computing			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	Prof. Dr. Claus Gerhardt			
Objectives	To have a firm command of the existence and regularity theories for solutions of nonlinear partial differential equations of second order.			
Content	• *H"older continuity of weak solutions of linear partial differential equations*: Sobolev spaces, Harnack inequality, De Giorgi–Nash theory, Moser's iteration technique. • *H"older continuity of the gradient of a solution*: Application of the De Giorigi–Nash estimates to the gradient of the solution of an elliptic quasilinear differential equation, Morrey's lemma. • *Existence theory for quasilinear elliptic differential equations*: Continuity method, fixed point theorems, Leray-Schauder's fixed point theorem.			
Learning outcomes	• The capability of solving problems in the underlying field. • The capability to present the solutions in the exercise courses.			
Prerequisites	none			
Suggested previous knowledge	Partial Differential Equations I, Schauder theory.			
Assessment(s)	Homework consisting of solving exercises and a written or oral examination at the end of the term.			
Literature	David Gilbarg, Neal Trudinger: Elliptic partial differential equations of second order R.A. Adams: Sobolev spaces			

9 Numerical Linear Algebra

Module	**Code** MH5	**Name** Numerical Linear Algebra		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle** WS|SS
Program	Master Scientific Computing			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	R. Rannacher et al.			
Objectives	To have a firm command of advanced numerical methods for problems in linear algebra			
Content	• Systems of linear equations and eigenvalue problems • Iterative methods of fixed point type • Krylov space methods • Singular value decomposition • Application to matrices with special structures			
Learning outcomes	• Analytical and algorithmical thinking • Mastering of methods from analysis and linear algebra • Capability of problem solving			
Prerequisites	None			
Suggested previous knowledge	Courses on Linear Algebra and Introductory Numerical Analysis			
Assessment(s)	Homework, exercise course, written or oral exam			
Literature	Announcement in the course (Lecture Notes)			

10 Numerics of Ordinary Differential Equations

Module	**Code** MH6	**Name** Numerics of Ordinary Differential Equations		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle** WS|SS
Program	Master Scientific Computing			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	Various			
Objectives	To have a firm command of advanced numerical methods for ordinary differential equations			
Content	• Theory of initial and boundary value problems • One-step methods: consistency, stability and convergence • Linear multi-step methods, extrapolation, Galerkin methods • Differential-algebraic problems • Shooting method • Sturm-Liouville problems			
Learning outcomes	• Analytical and algorithmical thinking • Mastering of methods from analysis and linear algebra • Capability of problem solving			
Prerequisites	None			
Suggested previous knowledge	Course on Introductory Numerical Analysis			
Assessment(s)	Homework, exercise course, written or oral exam			
Literature	Announcement in the course (Lecture Notes)			

11 Numerical Solution of Partial Differential Equations

<table>
<tr><td>Module</td><td>Code
MH7</td><td colspan="3">Name
Numerical Solution of Partial Differential Equations</td></tr>
<tr><td></td><td>Credit Points
8</td><td>Workload
240 h</td><td>Duration
1 Semester</td><td>Cycle
WS|SS every 2 years</td></tr>
<tr><td>Program</td><td colspan="4">Master Scientific Computing Mathematics, Computer Science, Technical Computer Science</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 4 h + Exercise course 2 h</td></tr>
<tr><td>Lecturer</td><td colspan="4">Various</td></tr>
<tr><td>Objectives</td><td colspan="4">To have a firm command of the most relevant methods for the numerical solution of initial-value and initital-value/boundary-value problems: Poisson, heat and wave equation</td></tr>
<tr><td>Content</td><td colspan="4">Theory of partial differential equations, type classification and properties of solutions
Finite difference method for elliptic boundary-value problems
Finite element method for elliptic boundary-value problems: discretization, a-priori and a-posteriori error analysis, grid adaptation
Solution methods for large linear systems: fixed-point iterations, Krylow methods and geometric multigrid methods
Methods for parabolic initial-value/boundary-value problems (heat equation)
Methods for hyperbolic problems (wave equation)</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">Analytic and algorithmic thinking
Application of methods from analysis and linear algebra
Independent solution of excercises with presentation</td></tr>
</table>

Prerequisites	[Names/codes of other modules]
Suggested previous knowledge	Analysis I (MA1), Linear Algebra I (MA4), Introduction to Numerical Methods (MA7), Partial Differential Equations I (MC2)
Assessment(s)	weekly excercises with presentation, written or oral exam
Literature	Announced in the lecture

12 Optimization with Ordinary Differential Equations

Module	Code	Name		
	MH8	Optimiz. with Ordinary Differential Equations		
	Credit Points	**Workload**	**Duration**	**Cycle**
	6	240 h	1 Semester	
Program	Master Scientific Computing (and Computer Science)			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	Various			
Objectives	To have a firm command of numerical and algorithmic techniques for solving boundary value problems, optimal control problem, parameter estimation problems, and optimum experimental design problems constrained by ordinary differential equations.			
Content	**Modeling of Dynamic Processes** Process models built from ordinary differential equations and differential-algebraic equations **Boundary Value Problems** Existence and uniqueness of solutions, shooting methods, condensing **Optimal Control Problems** Problem formulations, Pontryagin's maximum principle, direct methods (collocation, single and multiple shooting), condensing **Parameter Estimation Problems** nonlinear and linear least-squares problems, Moore-Penrose and generalized inverses, generalized Gauß-Newton method, local contraction theory, statistical interpretations of incompatibility, covariance analysis and confidence regions **Optimum Experimental Design Problems** Problem formulation, covariance objective functions, matrix derivative generation, structure exploitation, binary controls			
Learning outcomes	The students know how to model dynamic processes by ordinary differential equations, how to design experiments for obtaining real-world data such that the amount of information contained is maximized, how to fit the model behavior to this real-world data, and how to control a dynamic process to minimize/maximize a performance criterion.			
Prerequisites	None.			
Suggested previous knowledge	Numerics of Ordinary Differential Equations (MH6), Calculus, Linear Algebra, basic knowledge of one mathematical programming language (e.g. Matab, Octave, C/C++, Fortran)			
Assessment(s)	Successful participation in the exercises (50% of the points for exercises have to be gained) and a written exam or oral exam			
Literature	H.G. Bock, M. Diehl, C. Kirches, K. Mombaur, S. Sager: lecture notes, 2014. J.T. Betts: Practical methods for optimal control using nonlinear programming. SIAM, 2009. M. Gerdts: Optimal Control of ODEs and DAEs. De Gruyter, 2012.			

13 Numerical Methods in Continuum Mechanics

Module	**Code** MH9	**Name** Numerical Methods in continuum mechanics		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle**
Program	Master Scientific Computing and Mathematics and Physics			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer				
Objectives	Knowledge of the mathematical theory and of numerical approximation techniques for continuum mechanical problems			
Content	• Models in continuum mechanics: Lamé-navier, Euler-, Navier-Stokes equations • Finite elements for structural mechanics • Finite elements for fluid mechanics: stokes-elements, inf-sup condition, stabilization techniques • Numerical approximation of the discretized algebraic problems • Time discretization • Fluid-structure-interactions			
Learning outcomes	Analytical and algorithmic skills, mathematical modeling, application of techniques from calculus and linear algebra, independent work on problem sets, presentation in tutorials			
Prerequisites	N/A			
Suggested previous knowledge	Introduction to numerical mathematics [MA7], partial differential equations [MC2], numerical methods for partial differential equations [MH7], functional analysis [MC3]			
Assessment(s)	homeworks & presentation, final exam (written or oral)			
Literature	Will be announced			

14 Numerical Methods in Fluid Dynamics

Module	**Code**	**Name**		
	MH9a	Numerical Methods in fluid dynamics		
	Credit Points	**Workload**	**Duration**	**Cycle**
	5	140 h	1 Semester	
Program	Master Scientific Computing and Mathematics and Physics			
Methods	Lecture 2 h + Exercise course 2 h			
Lecturer				
Objectives	Knowledge of the finite element method for problems in fluid dynamics			
Content	• Models in fluid dynamics • Mathematical theory of the Navier-Stokes equations • Finite elements for the Navier-Stokes equations • Time discretization • Numerical approximation of the discretized algebraic problems			
Learning outcomes	Analytical and algorithmic skills, mathematical modeling, application of techniques from calculus and linear algebra, independent work on problem sets, presentation in tutorials			
Prerequisites	N/A			
Suggested previous knowledge	Introduction to numerical mathematics [MA7], partial differential equations [MC2], numerical methods for partial differential equations [MH7], functional analysis [MC3]			
Assessment(s)	homeworks & presentation, final exam (written or oral)			
Literature	Will be announced			

15 Special topics in Numerics

Module	**Code** MH10	**Name** Special topics in Numerics		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle**
Program	Master Scientific Computing and Mathematics and Physics			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	T.B.A.			
Objectives	Knowledge of methods for the modeling, analysis, simulation and optimization of complex problems from different applications			
Content	The class discusses selected topics related to the numerical treatment of complex problems arising from different applications: • Numerical stability analysis • Numerical methods for conservation laws • Discontinuous finite element methods • Numerics for pde-constraint optimization problems • Numerics for risk analysis • Non-standard Galerkin methods • Numerics for the Schr"odinger equation • Numerics for multi-scale problems • Numerics for multi-physics problems			
Learning outcomes	Analytical and algorithmic skills, mathematical modeling, skills for selecting problem-adapted numerical methods, independent work on problem sets, presentation in tutorials			
Prerequisites	N/A			
Suggested previous knowledge	Introduction to numerical mathematics [MA7], numerical methods for partial differential equations [MH7]			
Assessment(s)	homeworks & presentation, final exam (written or oral)			
Literature	Will be announced			

16 Statistics II

Module	Code	Name		
	MH12a	Statistics II		
	Credit Points	**Workload**	**Duration**	**Cycle**
	8 CP	240 h	1 Semester	
Program	Master Scientific Computing, Master Mathematics, Master Computer Science			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	Prof. Dr. Mark Podolskij			
Objectives	To have a firm understanding of the advanced statistical methods for iid and non-iid data			
Content	• Estimation and prediction methods for linear models • Order statistics; estimation of quantiles • Empirical processes; hypothesis testing • Asymptotic theory for U-Statistics • Non-parametric estimation methods; kernel density estimator			
Learning outcomes	• firm understanding of the advanced statistical methods for iid and non-iid data			
Prerequisites	MC4 or equivalent; MD2 or equivalent			
Suggested previous knowledge				
Assessment(s)	TBD (typically, homework and written exam)			
Literature	H. David and H. Nagaraja: Order Statistics. Wiley, 2003. D. Pollard: Convergence of stochastic processes. Springer, 1984. A.W. van der Vaart: Asymptotic Statistics. Cambridge University Press, 2000.			

17 Statistical Forecasting

Module	Code	Name		
	MH12c	Statistical Forecasting		
	Credit Points	**Workload**	**Duration**	**Cycle**
	8 CP	240 h	1 Semester	
Program	Master Scientific Computing, Master Mathematics			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	Prof. Dr. Tilmann Gneiting			
Objectives	To have a firm understanding of the statistical theory of forecasting, and the ability to design, implement and evaluate prediction techniques			
Content	• Basic notions: statistical decision theory, probabilistic and point forecasts, prediction spaces, information bases, calibration and sharpness • Proper scoring rules and consistent scoring functions • Forecasts combinations • Times series forecasts and spatial prediction • Statistical postprocessing of ensemble forecasts; combining numerical and statistical approaches • Applications and case studies in meteorology, economics and other disciplines			
Learning outcomes	• Firm theoretical understanding of the measure theoretic, probabilistic and statistical foundations of forecasting • Design and implementation of statistical forecasting algorithms, along with associated assessment techniques			
Prerequisites	MC4 or equivalent; MD2 or equivalent			
Suggested previous knowledge	Programming in R			
Assessment(s)	TBD (typically, homework and written exam)			

Literature	Gneiting, T.: Making and evaluating point forecasts. *Journal of the American Statistical Association* **106** (2011), 746–762. Gneiting, T. and Raftery, A. E.: Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* **102** (2007), 359–378.

18 Probability Theory II

Module	Code MH13a	Name Probability Theory II		
	Credit Points 8 CP	**Workload** 240 h	**Duration** 1 Semester	**Cycle**
Program	Master Scientific Computing and Master Mathematics			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	Prof. Dr. Mark Podolskij			
Objectives	To have a firm understanding of stochastic calculus for continuous processes			
Content	• Definition and properties of the Brownian motion; martingale property of the Brownian motion • Markovian properties of the Brownian motion; law of iterated logarithm • Smoothness properties of the Brownian motion • Definition and properties of stochastic integrals; Itô isometry • Itô formula and exponential martingales • Stochastic differential equations; existence and uniqueness of weak/strong solutions			
Learning outcomes	• A firm understanding of stochastic calculus for continuous processes			
Prerequisites	MC4 or equivalent			
Suggested previous knowledge	MD2 or equivalent			
Assessment(s)	TBD (typically, homework and written exam)			
Literature	I. Karatzas and S. Shreve: Brownian motion and stochastic calculus. Springer, Berlin 1988. D. Revuz and M. Yor: Continuous martingales and Brownian motion. Springer, Berlin 1991.			

19 Spatial Stochastic Processes

Module	**Code** MH13c	**Name** Spatial Stochastic Processes		
	Credit Points 8 CP	**Workload** 240 h	**Duration** 1 Semester	**Cycle**
Program	Master Scientific Computing and Master Mathematics			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	Prof. Dr. Tilmann Gneiting			
Objectives	To have a firm understanding of the theory of stochastic processes on Euclidean spaces and spheres			
Content	• Basic notions: random functions, Kolmogorov's theorem, Gaussian processes, stationarity, Bochner's theorem, Cramér's theorem, spatial prediction • Second order theory: positive definite functions on Euclidean spaces and spheres, non-stationary schemes, spectral theory and orthogonal expansions • Gaussian random fields and Lévy bases: Properties of the sample paths, stochastic integrals • Models building and model fitting for spatial and spatio-temporal data			
Learning outcomes	• Firm theoretical understanding of spatial stochastic processes • Ability to build basic stochastic models for spatial data			
Prerequisites	MC4 or equivalent			
Suggested previous knowledge	MD2 or equivalent			
Assessment(s)	TBD (typically, homework and written exam)			
Literature	Yaglom, A. M.: Correlation Theory of Stationary and Related Random Functions, Springer 1987 Gneiting, T. and Guttorp, P.: Continuous-parameter stochastic process theory, in Handbook of Spatial Statistics (Gelfand, A. E., Diggle, P. J., Fuentes, M. and Guttorp, P., eds.), Chapman & Hall 2010, pages 17–28			

20 Implementation of numerical methods for partial differential equations

Module	**Code** MH27	**Name** Implementation of numerical methods for partial differential equations		
	Credit Points 6 CP	**Workload** 180h	**Duration** 1 semester	**Cycle** yearly
Program	Master Scientific Computing, Mathematics, Computer Science, Physics, advanced bachelor students			
Course type	2 SWS lecture + 2 SWS exercise session			
Objectives	Learn to use the software deal.II to numerically solve a wide range of partial differential equations.			
Course description	This course serves as an introduction to the use of deal.II with an emphasis on the practical implementation of the finite element methods.			
Skills to be acquired	Ability to modify existing deal.II codes to solve the partial differential equations and to write new deal.II based programs.			
Prerequisites	none			
Helpful previous knowledge	Knowledge in C/C++ particularly in classes, pointers, references, templates. Basic knowledge of numerical analysis. Prior knowledge of implementating finite element methods to solve differential equations is helpful. It would be beneficial to simultaneously attend the course "Numerical methods for partial differential equations" although not a mandatory prerequiste.			
Grading policy	Grade based on assigned tasks including a final project with an oral presentation.			
Suggested literature	The lectures will be based on the available online documentation provided on the webpage http://www.dealii.org.			

21 Fundamentals of Computational Environmental Physics (FCEP)

Module	Code	Name		
	MH30	Fundamentals of Computational Environmental Physics		
	Credit Points	**Workload**	**Duration**	**Cycle**
	8 CP	240h	1 semester	
Program	Master Scientific Computing, Mathematics, Computer Science, Physics, advanced bachelor students			
Course type	4 SWS lecture + 2 SWS exercise session			
Objectives	Learn how to model fundamental processes in environmental physics with the continuum mechanical approach and learn how to simulate such models with state-of-the-art numerical methods.			
Course description	**Elementary linear models:** • Flow in porous media / elliptic partial differential equations (PDEs) • Scalar transport / first-order hyperbolic PDEs • Heat Transport / parabolic PDE • Wave Propagation / first and second order hyperbolic PDEs **Nonlinear models:** • Coupled elementary models • Fluid dynamics / Stokes and Navier-Stokes equation			
Skills to be acquired	Ability to apply models and numerical methods to environmental problems			
Prerequisites	none			
Helpful previous knowledge	Programming experience in particular in C++. Basic knowledge of numerical methods is useful.			
Grading policy	Solution of exercises and a final exam in written or oral form. Details will be given by the lecturer at the beginning of the course.			
Suggested literature				

22 Mixed finite element methods

Module	**Code** MH30	**Name** Mixed finite element methods		
	Credit Points 6 CP	**Workload** 180	**Duration** 1	**Cycle**
Program	Mathematics Master, Scientific Computing Master, Physics Master, Advanced Bachelor			
Course type	4 hr lecture			
Objectives	Foundations of stability theory for mixed finite element methods and Hilbert cochain complexes. Understanding of the relation between matching function spaces and stability. Knowledge of particular discretization techniques for incompressible flow. Foundations and applications of discontinuous Galerkin methods			
Course description	Hilbert cochain complexes, finite element cochain complexes, stability of mixed finite element methods, multigrid methods for mixed finite elements, incompressible flow, discontinuous Galerkin methods			
Skills to be acquired	Ability to understand and analyze non-standard finite element methods			
Prerequisites	Numerical methods for partial differential equations			
Helpful previous knowledge				
Grading policy	Classroom participation and oral exam. Details will be given by the lecturer at the beginning of the course.			
Suggested literature	**Grossmann, Roos(, Stynes)**: Numerical Treatment of Partial Differential Equations, English edition/ deutsche Ausgabe **Arnold, Falk, Winther**: Finite element exterior calculus: from Hodge theory to numerical stability, Bulletin of the AMS 2010, `http://www.ima.umn.edu/ arnold//papers/bulletin.pdf` B. Rivière: Discontinuous Galerkin methods for solving elliptic and parabolic equations, SIAM, 2008			

23 Optimization with PDEs: Parameter Estimation and Optimal Experimental Design

Module	Code	Name		
	MH31	Optimization with PDEs: Parameter Estimation and Optimal Experimental Design		
	Credit Points	**Workload**	**Duration**	**Cycle**
	8 CP	240h	1	
Program	Mathematics Master, Scientific Computing Master and Interested Students of other Disciplines			
Course type	Lecture course 4 hours + exercise session 2 hours			
Objectives	Learn the basic concepts to solve parameter estimation and optimal experimental design problems with models based on PDEs			
Course description	The lecture gives an introduction to the theory and numerics of optimization problems with PDEs. The following topics are covered: Estimation of parameters in elliptic and parabolic PDEs; Optimal experimental design with PDEs; Optimal control of PDEs.			
Skills to be acquired	Ability to numerically solve optimization problems with PDEs.			
Prerequisites	No prerequisites			
Helpful previous knowledge	Basic concepts of numerical methods for ordinary and partial differential equations (ODEs and PDEs) are advantageous. Knowledge of optimization methods is not mandatory.			
Grading policy	Solution of exercises and a final exam in written or oral form. Details will be given by the lecturer at the beginning of the course.			
Suggested literature	Lecture notes: Optimierung mit partiellen Differentialgleichungen, T. Carraro, 2012 M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich, Optimization with PDE Constraints, Springer, 2008 D. Uciński, Optimal Measurement Methods for Distributed Parameter System Identification, Crc Pr Inc (2005) H.W. Engl, M. Hanke, A. Neubauer, Regularization of Inverse Problems, Kluwer, 2008 F. Tröltzsch, Optimale Steuerung partieller Differentialgleichungen, Vieweg, 2009 D.G. Luenberger, Linear and Nonlinear Programming, Springer, Berlin, 2008			

Courses in Computer Science

24 Computational Geometry

<table>
<tr><td>Module</td><td>Code
IAGM</td><td colspan="3">Name
Computational Geometry</td></tr>
<tr><td></td><td>Credit Points
4</td><td>Workload
120 h</td><td>Duration
1 Semester</td><td>Cycle
WS/SS</td></tr>
<tr><td>Program</td><td colspan="4">Master Scientific Computing and Computer Science</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 2 h + Exercise course 1 h</td></tr>
<tr><td>Lecturer</td><td colspan="4">JProf.Dr. Heike Leitte</td></tr>
<tr><td>Objectives</td><td colspan="4">The students have a firm command of the fundamental algorithms and data structures of computational geometry and can implement them.</td></tr>
<tr><td>Content</td><td colspan="4">• Fundamental concepts of computational geometry such as convex hull and polygon triangulation.
• Efficient point location
• Voronoi diagrams
• Delaunay triangulation
• Search structures
• Algorithmic complexity</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">The students
• know and understand central algorithms from computational geometry.
• can implement efficient algorithms for geometrical problems.
• can weigh up different strategies and choose appropriate ones.</td></tr>
<tr><td>Prerequisites</td><td colspan="4"></td></tr>
<tr><td>Suggested previous knowledge</td><td colspan="4">Basic geometrical knowledge, programming experience</td></tr>
<tr><td>Assessment(s)</td><td colspan="4">assignments, written or oral exam</td></tr>
<tr><td>Literature</td><td colspan="4">M. de Berg, O. Cheong, M. van Kreveld, M. Overmars: Computational Geometry - Algorithms and Applications, 3rd edition, Springer, 2008.</td></tr>
</table>

25 Computer Graphics 1

<table>
<tr><td>Module</td><td>Code
ICG1</td><td colspan="3">Name
Computer Graphics 1</td></tr>
<tr><td></td><td>Credit Points
6</td><td>Workload
180 h</td><td>Duration
1 Semester</td><td>Cycle
WS (odd years)</td></tr>
<tr><td>Program</td><td colspan="4">Master Scientific Computing (and Computer Science)</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 2 h + Exercise course 2 h</td></tr>
<tr><td>Lecturer</td><td colspan="4">Dr. Susanne Krömker</td></tr>
<tr><td>Objectives</td><td colspan="4">To have a firm command of programming of still images, 3D graphics and animations for various purposes in scientific computing based on the framework of the OpenGL graphics library</td></tr>
<tr><td>Content</td><td colspan="4">1. Basics in Computer Graphics
Projections and Transformations in Homogeneous Coordinates
Drawing Algorithms
Buffer Concepts including Double and Stereo Color Buffer, Depth Buffer, Stencil Buffer, Accumulation Buffer
2. OpenGL Direct Rendering
Color Spaces, Shading and Light Models
Textures
3. Global Illumination
Raytracing with PovRay
Volume Rendering</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">The students have a profound understanding of computer graphics from the 2D raster graphics basics to objects in 3D world coordinates. They are able to choose between fast rendering of realistic effects with local light models or global methods for rendering of special effects like raytracing and volume rendering for scientific purposes.</td></tr>
<tr><td>Prerequisites</td><td colspan="4">None</td></tr>
<tr><td>Suggested previous knowledge</td><td colspan="4">Introduction to Applied Computer Science (IPR), Programming Course (IPK)</td></tr>
<tr><td>Assessment(s)</td><td colspan="4">Successful participation in the exercises (50% of the exercise testings have to be gained) and a written exam or oral exam</td></tr>
<tr><td>Literature</td><td colspan="4">J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Hughes and R. L. Phillips: Computer Graphics: Principles and Practice. Addison-Wesley, 1996
D. Shreiner, M. Woo, J. Neider and T. Davis: OpenGL programming guide: the official guide to learning OpenGL, version 1.4.OpenGL Architecture Review Board, Addison-Wesley, 2004</td></tr>
</table>

26 Computer Graphics 2

Module	**Code** ICG2	**Name** Computer Graphics 2		
	Credit Points 6	**Workload** 180 h	**Duration** 1 Semester	**Cycle** SS (even years)
Program	Master Scientific Computing (and Computer Science)			
Methods	Lecture 2 h + Exercise course 2 h			
Lecturer	Dr. Susanne Krömker			
Objectives	To have a firm command of programmable graphics hardware with *C for graphics* and advanced packages for global rendering methods using numerical algorithms for solving nonlinear systems of equations			
Content	**1. Shading** Shaders based on Bidirectional Reflectance Distribution Function (BRDF) Programmable Graphics Hardware with *C for graphics (Cg)* **2. Advanced Methods for Global Rendering** Radiosity versus Raytracing Photon Mapping **3. Rendering of Large Data Sets** Non-Photorealism (NPR) Data Reduction and Splines			
Learning outcomes	The students know about advanced light models, and they can distinguish between vertex based and fragment based shaders. They are able to judge the pros and cons of various global photorealistic methods. They can sensibly insert non-photorealistic methods for scientific visualization.			
Prerequisites	None			
Suggested previous knowledge	Introduction to Applied Computer Science (IPR), Programming Course (IPK), Computer Graphics 1 (ICG1)			
Assessment(s)	Successful participation in the exercises (50% of the exercise testings have to be gained) and a written exam or oral exam			
Literature	J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Hughes and R. L. Phillips: Computer Graphics: Principles and Practice. Addison-Wesley, 1996 Randima Fernando, Mark J. Kilgard: The Cg Tutorial. Addison-Wesley, 2003			

27 Data Mining - Algorithms and parallel Techniques

<table>
<tr><td>Module</td><td>Code

IDM</td><td colspan="3">Name
Data Mining - Algorithms and
parallel Techniques</td></tr>
<tr><td></td><td>Credit Points
6</td><td>Workload
180 h</td><td>Duration
1 Semester</td><td>Cycle
-</td></tr>
<tr><td>Program</td><td colspan="4">Master Scientific Computing</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 2 h + Exercise course 2 h</td></tr>
<tr><td>Lecturer</td><td colspan="4">Prof. Dr. Artur Andrzejak</td></tr>
<tr><td>Objectives</td><td colspan="4">To have a firm knowledge of applications, sequential algorithms and their parallel counterparts within data mining.</td></tr>
<tr><td>Content</td><td colspan="4">This module covers the sequential and parallel algorithms within data mining together with their programming and applications. This comprises the following topics:
pre-processing of data
feature generation and selection
classification and regression techniques
clustering
time series analysis
Bayesian networks
evaluation of results.
An essential part of the course is devoted to the parallel and distributed data mining, for example under the Map-Reduce programming model. The practical aspects are gained via programming examples in Matlab / GNU Octave and usage of libraries / tools such as Weka and KNIME.</td></tr>
</table>

Learning outcomes	• Familiarity with applications of data mining • Understanding of the methods of data preprocessing (normalization, discretization, dimensionality reduction) • Knowledge of approaches for classification, regression, clustering and their parallel and incremental implementations • Familiarity with methods of evaluating results • Understanding overfitting phenomena and methods of their prevention • Practical knowledge of data mining with Matlab / GNU Octave and with Java libraries and frameworks (Weka, KNIME) • Proficiency in parallel mining of large datasets with MapReduce and Matlab
Prerequisites	None
Suggested previous knowledge	Knowledge of Java (eg via Introduction to Software Engineering (ISW)) and in elementary probability theory / statistics
Assessment(s)	Successful participation in the exercises with homework (achieving a minimum score) and passing a final exam.
Literature	Ethem Alpaydin, Maschinelles Lernen, Oldenbourg Verlag, 2008. Stephen Marsland, Machine Learning: An Algorithmic Perspective, CRC Press Inc., 2009. Robert Nisbet, John Elder IV, Gary Miner, Handbook of Statistical Analysis and Data Mining Applications, Academic Press, 2009. Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification (2nd edition), Wiley-Interscience, 2000.

28 Efficient Algorithms 1

Module	**Code** IEA1	**Name** Efficient Algorithms 1		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle** WS|SS
Program	Master Scientific Computing			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	Prof. Dr. Gerhard Reinelt			
Objectives	To acquire the knowledge for designing and implementing algorithms for polynomially solvable combinatorial optimization problems and to gain experience for applying optimization models.			
Content	• Basic graph theory and algorithms • Optimal trees and branchings • Shortest paths • Assignment problem • Maximum flows and minimum cuts • Network flows • Matching problems			
Learning outcomes	The participants • understand basic graph-theoretical concepts, • can analyze the complexity of problems and algorithms, • have knowledge in modelling optimization problems, • know essential algorithms for polynomial problems, • are familiar with application areas.			
Prerequisites	IP, IPK, IAD, MA4			
Suggested previous knowledge	MA1			
Assessment(s)	Solution of 50% of the assignments and written exam			

Literature	Korte, Vygen: Combinatorial Optimization, Springer, 2007 Cook, Cunningham, Pulleyblank, Schrijver: Combinatorial Optimization, Wiley, 1997

29 Efficient Algorithms 2

Module	**Code** IEA2	**Name** Efficient Algorithms 2		
	Credit Points 8	**Workload** 160 h	**Duration** 1 Semester	**Cycle** WS|SS
Program	Master Scientific Computing			
Methods	Lecture 2 h + Exercise course 2 h			
Lecturer	Prof. Dr. Gerhard Reinelt			
Objectives	Learn the full spectrum of algorithms for finding good approximate solutions to difficult combinatorial optimization problems or for solving them to proven optimality.			
Content	• NP-hard optimization problems • Approximative algorithms and heuristics • Relaxations • Branch-and-bound algorithms • Dynamic programming • Integer programming • Polyhedral combinatorics • Column generation and decomposition • Traveling salesman problem • Max-cut problem			
Learning outcomes	The participants • can classify easy and hard problems, • known the full spectrum of algorithms in combinatorial optimization, • are able to model and solve difficult application problems with appropriate algorithms.			
Prerequisites	IP, IPK, IAD, MA4			

Suggested previous knowledge	MA1, IEA1
Assessment(s)	Solution of 50% of the assignments and written exam
Literature	Korte, Vygen: Combinatorial Optimization, Springer, 2007 Cook, Cunningham, Pulleyblank, Schrijver: Combinatorial Optimization, Wiley, 1997 Wolsey: Integer Programming, Wiley, 1998

30 Introduction to Scientific Visualization

<table>
<tr><td>Module</td><td>Code
IGWV</td><td colspan="3">Name
Introduction to Scientific Visualization</td></tr>
<tr><td></td><td>Credit Points
8</td><td>Workload
240 h</td><td>Duration
1 Semester</td><td>Cycle
WS</td></tr>
<tr><td>Program</td><td colspan="4">Master Scientific Computing and Computer Science</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 4 h + Exercise course 2 h</td></tr>
<tr><td>Lecturer</td><td colspan="4">JProf.Dr. Heike Leitte</td></tr>
<tr><td>Objectives</td><td colspan="4">The students have a firm command of the fundamental concepts and algorithms of scientific visualization and can implement them.</td></tr>
<tr><td>Content</td><td colspan="4">Human vision and how it affects the intake of visual information.
Visualization pipeline
Statistical graphics
Visualization of scalar and vector fields
Interaction and data exploration
Fundamentals of graph drawing</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">The students
know and understand central concepts and algorithms from scientific visualization.
can implement algorithms and interaction mechanisms.
can weigh up different strategies and choose appropriate ones.</td></tr>
<tr><td>Prerequisites</td><td colspan="4"></td></tr>
<tr><td>Suggested previous knowledge</td><td colspan="4">Programming experience</td></tr>
<tr><td>Assessment(s)</td><td colspan="4">assignments, written or oral exam</td></tr>
<tr><td>Literature</td><td colspan="4">A. C. Telea: Data Visualization - Principles and Practice, A K Peters Ltd., 2007.
R. Spence: Information Visualization, Addison Wesley, 2000.</td></tr>
</table>

31 Knowledge Discovery in Databases

<table>
<tr><td>Module</td><td>Code
IKDD</td><td colspan="3">Name
Knowledge Discovery in Databases</td></tr>
<tr><td></td><td>Credit Points
8</td><td>Workload
240 h</td><td>Duration
1 Semester</td><td>Cycle
every other WS</td></tr>
<tr><td>Program</td><td colspan="4">Master Scientific Computing</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 4 h + Exercise course 2 h</td></tr>
<tr><td>Lecturer</td><td colspan="4">Prof.Dr. Michael Gertz</td></tr>
<tr><td>Objectives</td><td colspan="4">Students should get a deep understanding of concepts, models, and techniques underlying the KDD-process and different data mining techniques, including data preprocessing, clustering, frequent pattern analysis, classification, outlier detection and graph mining. They should also be able to apply different data mining techniques to analyse and explore real-world datasets.</td></tr>
<tr><td>Content</td><td colspan="4">
The KDD process
Basics: Data, statistics, and probability theory
Data preprocessing: data quality, duplicate detection, data integration
Data clustering: k-means, hierarchical clustering, density-based clustering, cluster evaluation
Frequent pattern mining: frequent itemsets, mining sequential data, mining with constraints
Classification: decision trees, bayes classifier, overfitting, support vector machines
Outlier detection: statistical approaches, clustering-based techniques, density-based techniques
Graph mining: subgraph patterns, graph clustering, indexing
Mining spatial and spatio-temporal data
</td></tr>
</table>

Learning outcomes	• Knowing the requirements and methods underlying the different steps of the KDD process • Knowing the concepts and techniques related to different clustering approaches • Knowing the concepts and techniques related to different classification approaches • Knowing the concepts and techniques related to different frequent pattern mining approaches • Be able to apply different data preprocessing and mining techniques for analyzing and exploring data • Be familiar with the support of data mining approaches using relational database management system • Knowing fundamental methods underlying graph mining
Prerequisites	Algorithms and data structures (IAD)
Suggested previous knowledge	Introduction to databases (IDB1)
Assessment(s)	Assignments; at least 50% of the credit points for the assignments need to be obtained to be eligible to participate in the final written exam
Literature	Jiawei Han und Micheline Kamber: Data Mining. Concepts and Techniques, Morgan Kaufmann Series in Data Management Systems (2nd Edition), 2006. Martin Ester und Jörg Sander: Knowledge Discovery in Databases: Techniken und Anwendungen, Springer, 2000. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining. Addison Wesley, 2005.

32 Artificial Intelligence

Module	**Code** IKI	**Name** Artificial Intelligence		
	Credit Points 6	**Workload** 180 h	**Duration** 1 Semester	**Cycle**
Program	Master Scientific Computing (and Mathematics) (and Computer Science)			
Methods	Lecture 2 h + Exercise course 2 h			
Lecturer	Prof.Dr. Björn Ommer			
Objectives	To be able to develop algorithms for solving problems by searching. To know how to apply logical inference for finding solutions. To have a firm command of applying inference under uncertainties. To be able to develop learning agents. To know how to apply the developed expertise to different application areas such as Computer Vision, text mining.			
Content	• Solving problems by searching • Game playing • Inference using logic • Knowledge bases • Planning and acting • Reasoning under uncertainty • Machine Learning Applications			
Learning outcomes	Being able to solve problems in the field of artifical intelligence.			
Prerequisites				
Suggested previous knowledge				
Assessment(s)	homework assignments, final examination			
Literature	Russell & Norvig: Artificial Intelligence: A Modern Approach			

33 Mixed-Integer Programming

Module	**Code** IMIP	**Name** Mixed Integer Programming		
	Credit Points 3	**Workload** 90 h	**Duration** 1 Week	**Cycle** WS|SS
Program	Master Scientific Computing			
Methods	Lecture 4 h / day + practical exercises 4 h / day			
Lecturer	Prof. Dr. Gerhard Reinelt			
Objectives	To become familiar with the potential of mixed-integer programming and learn how to model problems and solve them with commercial software.			
Content	• Linear programming and duality • Polyhedral theory • Postoptimal analysis • Mixed-integer modelling • Computation of optimal solutions • Polyhedral combinatorics and combinatorial polytopes • Implementation of branch-and-cut algorithms • Valid inequalities and cuts			
Learning outcomes	The participants • know the modelling possibilities of (linear) mixed-integer programming, • aer familiar with all solution techniques, • have basic knowledge of commercial solvers, • are able to solve mixed-integer optimization problems, • know how to model practical problems.			
Prerequisites	IP, IPK, IAD, MA4			

Suggested previous knowledge	MA1, IEA1, IEA2
Assessment(s)	Solution of 50% of the assignments and written exam
Literature	Kallrath, Wilson: Business Optimisation using Mathematical Programming, Macmillan Press, 1997 Williams: Model Building, Wiley, 1999

34 Numerical Simulation of Transport Processes in Porous Media

<table>
<tr><td>Module</td><td>Code
INTPM</td><td colspan="3">Name
Numerical Simulation of Transport Processes in Porous Media</td></tr>
<tr><td></td><td>Credit Points
6</td><td>Workload
180 h</td><td>Duration
1 Semester</td><td>Cycle
WS</td></tr>
<tr><td>Program</td><td colspan="4">Master Scientific Computing, Mathematics and Computer Science</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 2 h + Exercise course 2 h</td></tr>
<tr><td>Lecturer</td><td colspan="4">Dr. Olaf Ippisch</td></tr>
<tr><td>Objectives</td><td colspan="4">To obtain a thorough introduction of both theory and application of models for flow and transport in porous media and their numerical solution.</td></tr>
<tr><td>Content</td><td colspan="4">Classification of partial differential equations
Spatial discretisation schemes
Simple and advanced iterative solvers for linear equation systems
Solution of parabolic partial differential equations
Solution of hyperbolic partial differential equations
Solution of non-linear partial differential equations
Modelling of Groundwater flow, heat transport, solute transport, sorption and unsaturated water transport</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">The students
can model saturated and unsaturated water transport in porous media, solute and heat transport
know the assumptions and limitations of the models.
can apply the most important numerical schemes to solve the resulting partial differential equations.
can apply the simulations to solve sample problems of soil remediation.</td></tr>
</table>

Prerequisites	
Suggested previous knowledge	C++-Programming
Assessment(s)	successful participation in the exercises (get at least 50% of the points), written or oral exam (depending on the number of participants)
Literature	

35 Object Recognition and Image Understanding

Module	Code	Name		
	IOAB	Object Recognition and Image Understanding		
	Credit Points	**Workload**	**Duration**	**Cycle**
	8	240 h	1 Semester	
Program	Master Scientific Computing (and Mathematics) (and Computer Science)			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	Prof.Dr. Björn Ommer			
Objectives	To have reached understanding of the state-of-the-art in mid- and high-level Computer Vision. To know the most relevant methods for robust object representation. To be able to apply essential algorithms from pattern recognition and machine learning to new problems in machine vision. To be capable of understanding the latest publications in Computer Vision and to evaluate their strengths and weaknesses. To have a firm command of the algorithmic basics and to be able to analyze and solve object recognition problems in novel application areas.			

Content	• Methods in mid- and high-level Computer Vision, esp.: • object detection and classification • local and global feature extraction • model based approaches • view based approaches • generative/discriminative methods • registration • shape analysis • voting methods • hashing approaches • hierarchical object representations • recognition of human actions • machine learning approaches for object representation
Learning outcomes	Being able to solve problems in the field of Computer Vision.
Prerequisites	
Suggested previous knowledge	
Assessment(s)	homework assignments, final examination
Literature	

36 Object-Oriented Programming for Scientific Computing

Module	Code	Name		
	IOPWR	Object-Oriented Programming for Scientific Computing		
	Credit Points	**Workload**	**Duration**	**Cycle**
	6	180 h	1 Semester	SS
Program	Master Scientific Computing, Mathematics and Computer Science			
Methods	Lecture 2 h + Exercise course 2 h			
Lecturer	Prof. Dr. Peter Bastian, Dr. Olaf Ippisch			
Objectives	Knowledge of advanced object-oriented programming techniques and experience in their practical application			
Content	This module deepens the skills in object-oriented programming obtained in the basic lecture "Introduction to practical informatics" with special emphasis on scientific computing. • Class concepts • Dynamic memory allocation • Exception handling • Safe allocation and deallocation of resources • Reasonable use of const • Template programming • Static versus dynamic polymorphism • Traits and Policies • Standard Template Library • Template Metaprogramming			

Learning outcomes	The students • are proficient in the programming language C++ • can assess the performance of different programming techniques • know template programming techniques and can use the Standard Template Library • can apply their new skills to solve selected problems of scientific computing
Prerequisites	
Suggested previous knowledge	Basic knowledge of an object-oriented programming language
Assessment(s)	successful participation in the exercises (get at least 50% of the points), written or oral exam (depending on the number of participants)
Literature	

37 High-Performance Computing

<table>
<tr><td>Module</td><td>Code
IPHR</td><td colspan="3">Name
High-Performance Computing</td></tr>
<tr><td></td><td>Credit Points
8</td><td>Workload
240 h</td><td>Duration
1 Semester</td><td>Cycle
WS|SS every 2 years</td></tr>
<tr><td>Program</td><td colspan="4">Master Scientific Computing Mathematics, Computer Science, Technical Computer Science</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 4 h + Exercise course 2 h</td></tr>
<tr><td>Lecturer</td><td colspan="4">Prof. Dr. Peter Bastian, Dr. S. Lang</td></tr>
<tr><td>Objectives</td><td colspan="4">To have a firm command of relevant programming models for parallel high-performance computing and a knowledge of parallel algorithms in several application areas.</td></tr>
<tr><td>Content</td><td colspan="4">
Introduction to parallel computer architectures
Shared memory programming models, basic algorithms for shared memory: condition synchronization, locks, semaphores, cache coherence, multithreading, GPGPU programming
Message-passing programming model: asynchronous vs. synchronous communication, message Passing Interface (MPI), basic algorithms for global communication: broadcast, reduce, barrier
Remote procedure call and its object oriented extensions
Performance analysis of parallel algorithms
Applications: dense linear algebra, solution of sparse linear systems, particle methods, sorting
</td></tr>
</table>

Learning outcomes	• Know the basic design of current parallel high-performance computers • Understand the different programming models with their strengths and weaknesses • Formulate and analyze parallel algorithms from several application areas
Prerequisites	
Suggested previous knowledge	Programming in C++, numerical methods
Assessment(s)	weekly excercises, programming projects, written or oral exam
Literature	V. Kumar, A. Grama, A. Gupta and G. Karypis: Introduction to Parallel Computing – Design and Analysis of Algorithms. Benjamin Cummings, 1994. Lecture notes are provided

38 Parallel Solution of Large Linear Systems

<table>
<tr><td rowspan="2">Module</td><td>Code
IPLGG</td><td colspan="3">Name
Parallel Solution of Large Linear Systems</td></tr>
<tr><td>Credit Points
8</td><td>Workload
240 h</td><td>Duration
1 Semester</td><td>Cycle
WS|SS every 2 years</td></tr>
<tr><td>Program</td><td colspan="4">Master Scientific Computing Mathematics, Computer Science, Technical Computer Science</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 4 h + Exercise course 2 h</td></tr>
<tr><td>Lecturer</td><td colspan="4">Prof.Dr. Peter Bastian</td></tr>
<tr><td>Objectives</td><td colspan="4">To have a firm command of state-of-the-art methods for the parallel solution of large linear systems arising from the discretization of elliptic partial differential equations</td></tr>
<tr><td>Content</td><td colspan="4">Finite element method for the discretization of elliptic partial differential equations
Abstract subspace correction method
Overlapping Schwartz method
Geometric and algebraic multigrid method
Non-overlapping domain decompostion method
Convergence theory based on subspace correction method
Implementation aspects and scalability of these methods</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">Students are able to
understand the structure and properties of linear systems arising from the discretization of elliptic partial differential equations
formulate and analyse a variety of different subspace correction methods for the solution of these linear systems
understand the algorithmic details of a software implementation of these methods in a message passing environment</td></tr>
<tr><td>Prerequisites</td><td colspan="4"></td></tr>
</table>

Suggested previous knowledge	Programming in C++, parallel computing, numerical solution of partial differential equations
Assessment(s)	weekly excercises, programming projects, written or oral exam
Literature	B. Smith and P. Bjørstad and W. Gropp: Domain Decomposition Methods. Cambridge University Press, 1996. A. Toselli and B. Widlund: Domain Decomposition Methods – Algorithms and Theory. Springer-Verlag, 2010. Lecture notes are provided

39 Computational Geometry and Grid Generation

Module	Code	Name		
	ICGGG	Computational Geometry and Grid Generation		
	Credit Points	**Workload**	**Duration**	**Cycle**
	6	180 h	1 Semester	WS
Program	Master Scientific Computing, Mathematics, Computer Science and Technical Computer Science			
Methods	Lecture 2 h + Exercise course 2 h			
Lecturer	Dr. Stefan Lang			
Objectives	To obtain thorough theoretical knowledge and to learn practical methods in the field of computational geometry and grid generation.			
Content	• Introduction to numerical simulation methods • Approximation and interpolation • Bernstein polynomials, Bezier curves and surfaces • Splines, B-Splines, Non-uniform Rational B-Splines (NURBS) • Computational methods for NURBS • Introduction to computational mesh generation • Theory and methods for Delauney triangulation • Advancing-front mesh generation			

<table>
<tr><td>Learning outcomes</td><td>The students

can represent simulation domains with adequate methods of computational geometry.
have knowledge about individual advantages of the presented modeling techniques and their theoretical limitations.
can select appropriate tools to construct own geometries.
have the knowledge about important mesh generation methods and practical software implications.
are able to generate computational domains and discrete computational surface and volume representations.
</td></tr>
<tr><td>Prerequisites</td><td></td></tr>
<tr><td>Suggested previous knowledge</td><td>C++-Programming, Linear Algebra</td></tr>
<tr><td>Assessment(s)</td><td>successful participation in the exercises (get at least 50% of the points), written or oral exam (depending on the number of participants)</td></tr>
<tr><td>Literature</td><td>Announced in the lecture</td></tr>
</table>

40 Spatial Databases

Module	Code IRDB	Name Spatial Databases		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle** every other WS
Program	Master Scientific Computing			
Methods	Lecture 4 h + Exercise course 2 h			
Lecturer	Prof.Dr. Michael Gertz			
Objectives	Students should get a deep understanding of concepts, models, and techniques underlying the modeling, management, processing and querying of diverse types of spatial data, and they should be able to apply suitable techniques, such as spatial index structures in different application settings.			

Content	• Principles and requirements of managing spatial data • Application domains of managing and analyzing spatial data • Support of managing spatial data using commercial and open-source DBMS • Concepts and methods for representing spatial data in 2d and 3d; tessellation and vector model, groups of spatial objects • abstract data types for spatial data • Fundamentals of computational geometry (e.g., convex hull, sweep-line techniques, polygon partitioning, cut of polygons) • Access structures for spatial data, in particular grid-files, kd-tree, quad-tress, and R-trees • Algorithms and cost models for using spatial index structures • Principles of spatial query processing and optimization, in particular spatial join techniques • Temporal databases and index structures • Moving objects: applications, querying, and index structures • Introduction to mining spatial data (clustering, outlier detection)

Learning outcomes	• Knowing the principles and requirements underlying spatial data and the management of such data in different application domains (e.g., geography, biology, cosmology) • Knowing the concepts and applications related to geographic information systems • Be able to apply concepts and techniques for modeling spatial data • Be familiar with the management and querying spatial data using a spatial database management system (e.g., PostGIS) • Knowing fundamental methods of computational geometry • Knowing important index structures for spatial data such as the grid-file, kd-tree, Quad-tree, and R-tree.
Prerequisites	Algorithms and data structures (IAD)
Suggested previous knowledge	Introduction to databases
Assessment(s)	Assignments; at least 50% of the credit points for the assignments need to be obtained to be eligible to participate in the final written exam
Literature	Spatial Databases - With Applications to GIS. Philippe Rigaux, Michel Scholl, Agnes Voisard. Morgan Kaufmann, 2001. Computational Geometry: Algorithms and Applications Mark de Berg, Otfried Cheong, Marc van Kreveld, und Mark Overmars, Springer, Berlin, 2008.

41 Visualization in Cultural Heritage

Module	Code IVCH	Name Visualization in Cultural Heritage		
	Credit Points 3	**Workload** 60 h	**Duration** 1 Semester	**Cycle** irregular
Program	Master Scientific Computing (and Computer Science)			
Methods	Lecture 2 h			
Lecturer	Dr. Susanne Krömker			
Objectives	To have a firm command of dealing with 3D scanners, georadar and magnetic field survey, of further investigations of measured data, and of obeying ethical principals in reconstruction and presentation of archaeological records.			
Content	**1. Acquisition and Processing of Data** Structured Light Scanner Time-of-flight-Scanner Dealing with Point Clouds Investigation of Meshes **2. Reconstruction and Puzzling** Reconstructions of Vessels Methods for 3D Stone Puzzle for Reconstructions of Ancient, Medieval, Islamic, Hinduistic and Buddhistic Architecture Transcriptions of various Scripts from Cuneiform to Greek, and Hebrew, Chinese and Khmer Vector Data, Skelettonization and Unicode **3. Good Practice in Archiving and Presenting** European and Non-European Cultures Ethical Principals London Charter			
Learning outcomes	The students can judge on the appropriate technical equipment for 3D data acquisition. They have a survey of various techniques to deal with these point clouds or meshes. They are sensitive to ethical principles in the reconstruction of artifacts.			
Prerequisites	None			
Suggested previous knowledge	Introduction to Applied Computer Science (IPR), Programming Course (IPK), Computer Graphics 1 (ICG1)			
Assessment(s)	Participation in the lecture and a written exam or oral exam (depending on the number of participants)			

Literature	Clive Orton: Mathematics in Archaeology. Cambridge, MA, Cambridge University Press, 1982 Katsushi Ikeuchi, Daisuke Miyazaki (editors): Digitally Archiving Cultural Objects. Springer, 2007

42 Visualization in Natural Sciences and Engineering

Module	Code	Name		
	IVNT	Visualization in Natural Sciences and Engineering		
	Credit Points	**Workload**	**Duration**	**Cycle**
	4	120 h	1 Semester	SS
Program	Master Scientific Computing and Computer Science			
Methods	Lecture 2 h + Exercise course 1 h			
Lecturer	JProf.Dr. Heike Leitte			
Objectives	The students have a firm command of advanced concepts and algorithms of scientific visualization and can implement them. The focus is on problems that occur in natural sciences and engineering.			
Content	• Data acquisition, simulation, storage and processing in natural sciences and engineering • Volume visualization • Vector field and flow visualization • Topology in visualization • Visualization of multivariate data • Visualization systems			
Learning outcomes	The students • know and understand advanced concepts and algorithms from scientific visualization. • can implement algorithms and interaction mechanisms. • can weigh up different strategies and choose appropriate ones.			
Prerequisites				
Suggested previous knowledge	Programming experience, Introduction to Scientific visualization			
Assessment(s)	assignments, written or oral exam			
Literature	C. D. Hansen, C. R. Johnson: The Visualization Handbook, Academic Press, 2004.			

43 Distributed Systems I

<table>
<tr><td>Module</td><td>Code
IVS1</td><td colspan="3">Name
Verteilte Systeme I / Distributed Systems I</td></tr>
<tr><td></td><td>Credit Points
6</td><td>Workload
180 h</td><td>Duration
1 Semester</td><td>Cycle
SS</td></tr>
<tr><td>Program</td><td colspan="4">Master Scientific Computing</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 2 h + Exercise course 2 h</td></tr>
<tr><td>Lecturer</td><td colspan="4">Prof. Dr. Artur Andrzejak or Dr. Sascha Hunold</td></tr>
<tr><td>Objectives</td><td colspan="4">To have a firm command of the basic principles, techniques and programming approaches within parallel and distributed systems and to apply them in practice.</td></tr>
<tr><td>Content</td><td colspan="4">The module covers the basic principles and techniques of parallel and distributed systems in the context of network applications and the parallel processing of data. It will enable students to understand the specifics and problems of distributed systems, to program distributed applications and to use software frameworks such as Hadoop.The main contents are:
concepts from the fields of architecture, protocols, programming, software frameworks and algorithms
parallel and distributed programming, especially more modern approaches such as Map-Reduce, Actors, stream-oriented programming
(as additional topics): fault tolerance, scalability and efficient protocols.
Examples from the domain of web services or data processing will illustrate the practical implementation of these concepts.</td></tr>
</table>

Learning outcomes	• Understanding of the different parallel architectures and characteristics of distributed systems • Knowledge of basic theoretical problems and algorithms in distributed systems (eg data distribution on processors) • Knowledge of applicability of various parallel programming frameworks (Pthreads, MPI, Map-Reduce) for a given architecture • Ability to create parallel and distributed programs in Java (eg threads, RMI) or Groovy (GPars) • Practical knowledge of modern approaches to parallel / distributed programming (Map-Reduce, Actors, dataflow programming) • Familiarity with application cases (web services, parallel processing of data)
Prerequisites	None
Suggested previous knowledge	Knowledge of Java (eg via Introduction to Software Engineering (ISW)) and in operating systems (eg via IBN)
Assessment(s)	Successful participation in the exercises with homework (achieving a minimum score) and passing a final exam.
Literature	Andrew S. Tanenbaum, Maarten Van Steen: Distributed Systems: Principles and Paradigms, Prentice Hall, 2006. George Coulouris, Jean Dollimore, Tim Kindberg: Distributed Systems: Concepts and Design (4th ed.), Addison-Wesley, 2005. Kurose, James F., Keith W. Ross: Computernetzwerke : der Top-Down-Ansatz, Pearson Studium, 2008. Grama, A., Gupta, A., Karypis, G., Kumar V.: Introduction to Parallel Computing, Addison-Wesley, 2. Auflage, 2003. Casanova, H.; Legrand, A.; Robert, Y.:Parallel Algorithms, Chapman & Hall/CRC, 2008.

44 Volume Visualization

<table>
<tr><td>Module</td><td>Code
IVV</td><td colspan="3">Name
Volume Visualization</td></tr>
<tr><td></td><td>Credit Points
8</td><td>Workload
240 h</td><td>Duration
1 Semester</td><td>Cycle
WS|SS</td></tr>
<tr><td>Program</td><td colspan="4">Master Scientific Computing</td></tr>
<tr><td>Methods</td><td colspan="4">Lecture 2h + Exercise course 3h</td></tr>
<tr><td>Lecturer</td><td colspan="4">Prof. Dr. Jürgen Hesser</td></tr>
<tr><td>Objectives</td><td colspan="4">To have a firm command of methods to represent complex scientific data, to interpolate and visualize it with volume rendering. In addition, a thorough understanding of accelerating and parallelizing the algorithms should be acquired.</td></tr>
<tr><td>Content</td><td colspan="4">Introduction of the visualization of scientific data of natural sciences and bio-sciences
Discrete and continuous representation of data and methods of interpolation
Methods of conversion between surface- and volume-representations and their efficient realizations
Theory of volume rendering and their different realizations
Accelerating and parallelization of volume rendering
Programming technique: GPU-programming</td></tr>
<tr><td>Learning outcomes</td><td colspan="4">The students
learn to understand how to use techniques of volume visualization to render complex scientific data.
This consists of the representation of data by surface or volume elements, the conversion of different representations and techniques of interpolation.
They further understand the physical principles of volume rendering, the different strategies of their realization with advantages and disadvantages – they should critically assess different techniques – and their parallelization.</td></tr>
<tr><td>Prerequisites</td><td colspan="4">Introduction into computer science I, programming course, algorithms & data structures;</td></tr>
</table>

Suggested previous knowledge	computer graphics I (ICG1), computer graphics II (ICG2), basics of scientific visualization (IGWV)
Assessment(s)	Successful participation in exercises (50% correct solutions), successful examination (written or oral)
Literature	Engel et al.: Real-Time Volume Graphics www.real-time-volume-graphics.org, Schroeder et al.: VTK Textbook http://www.kitware.com/products/books/vtkbook.html

45 Introduction to High Performance Computing

Module	Code MScTI_INTROHPC	Name Introduction to High Performance Computing		
	Credit Points 6 CP	**Workload** 180h	**Duration** 1 semester	**Cycle**
Program	Master Technische Informatik, Master Scientific Computing			
Course type	3 SWS lecture + 1 SWS exercise session			
Objectives	Students ... • Know the most important approaches of how to solve large scale computing problems. This involves hardware as well as software architectures • Have seen some real world examples in some detail • Have gained practical experience in implementing a high performance computing solution			
Course description	• The challenges of HPC • Hardware architectures • Software frameworks • Practical problems and their solutions			
Skills to be acquired				
Prerequisites	none			
Helpful previous knowledge	Basic knowledge of computer architecture			
Grading policy	15 - 30 min. oral exam. Details will be given by the lecturer at the beginning of the course.			
Suggested literature	none			

46 Modeling, Optimization and Control of Mechanical Systems

Module	**Code** MORMS	**Name** Modeling, Optimization and Control of Mechanical Systems		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle** every two years
Program	Master Scientific Computing (and Mathematics) (and Computer Science)			
Methods	Lecture 4 h + Computer exercises 2 h			
Lecturer	Prof. Dr. Katja Mombaur			
Objectives	- To give an application oriented introduction to modeling, optimization and control of mechanical systems with a focus on complex multibody systems in robotics and biomechanics. - To introduce computational tools to perform these tasks and apply them to examples in the computer exercises.			
Content	While the lecture focuses on the theoretical and mathematical foundations of the field, the computer exercises serve to teach the usage of software tools for modeling, visualization, simulation and optimal control treating different example problems. *Contents of the lecture*: Dynamic process modeling; Mechanical basics, kinematics, dynamics; Multibody system modeling; Simulation of motions; Nonlinear optimization; Direct methods for optimal control problems; Elementary control principles; Basics of system dynamics; Open-loop and closed loop control of motions; Modeling human-like walking and running motions; Modeling locomotion of humanoid and bipedall robots; Stability of motions; *Contents of the computer exercises*: Simulation and visualization of mechanical systems; Modeling multibody system with RBDL (Rigid Body Dynamics Library) ; Implementation and solution of optimal control problems with MUSCOD-II for different mechanical example systems;			

Learning outcomes	The students know principles of modeling, optimization and control of dynamic processes, in particular mechanical systems, and can explain and apply them. They are familiar with nonlinear optimization and optimal control methods and can compare and evaluate different mathematical approaches. The students can model, classify and analyze complex motions of mechanical systems, e.g. in robotics or biomechanics, and investigate specific properties such as stability. The students know how to use software tools based on C++ and Lua for the application domain They are capable to solve optimal control problems numerically and to evaluate the quality of the solution.
Prerequisites	Programing skills in C/C++, Lecture "Introduction to Numerics"
Suggested previous knowledge	Lectures "Algorithmic optimization", "Numerical Mathematics 1"; Knowledge of Matlab, Octave
Assessment(s)	Successful participation in the exercises (50% of the exercise testings have to be gained) and succesful completion of a written exam
Literature	D. Greenwood: Principles of Dynamics, Prentice Hall, 1987 I .Newton: Principia, 1687 (bzw. englische Version, University of California, 1999) J. T. Betts: Practical Methods for Optimal Control Using Nonlinear Programming. SIAM, Philadelphia, 2001 J. Craig: Introduction to Robotics - Mechanics and Control. Prentice Hall, 2003 J. Nocedal, S. Wright: Numerical Optimization, Springer, 2000 B. Siciliano, et al: Robotics - Modeling, Planning and Con-trol, Springer 2008 Spong, Hutchinson, Vidyasagar: Robot modeling and con-trol, Wiley, 2005 Perry, Burnfield: Gait Analysis - Normal and pathological function, Slack Inc., 2010 M. Raibert: Legged Robots that Balance, MIT Press, 2000

47 Scientific Visualization

Module	**Code** ISV	**Name** Scientific Visualization		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle** every two years
Program	Master Scientific Computing (and Computer Science)			
Methods	Lecture 4 h + Computer exercises 2 h			
Lecturer	Dr. Filip Sadlo			
Objectives	To give an introduction to mathematical and computer science aspects of scientific visualization techniques.			
Content	Introduction: Visualization Process; Data Sources and Representation; Interpolation and Filtering; Approaches for Visual Mapping; Scalar Field Visualization: Advanced Techniques for Contour Extraction, Classification, Texture-Based; Volume Rendering, Volumetric Illumination, Advanced Techniques for Volume Visualization, Pre-Integration, Cell Projection, Feature Extraction; Vector Field Visualization: Vector Calculus, Particle Tracing on Grids, Vector Field Topology, Vortex Visualization, Feature Extraction, Feature Tracking; Tensor Field Visualization: Glyphs, Hue-Balls and Lit- Tensors, Line-Based Visualization, Tensor Field Topology, Feature Extraction;			
Learning outcomes	The students know advanced concepts and techniques of Scientific Visualization. know advanced algorithms, mathematical fundamentals, data structures, and implementation aspects. are experienced in using visualization tools.			
Prerequisites	Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen, Grundlagen der wiss. Visualisierung			
Suggested previous knowledge				
Assessment(s)	Successful participation in the exercises (more than 50% have to be scored) und passing a written or oral exam.			

Literature	C.D. Hansen, C.R. Johnson, The Visualization Handbook, 2005.

48 Geometric Modeling and Animation

Module	Code IGMA	Name Geometric Modeling and Animation		
	Credit Points 8	**Workload** 240 h	**Duration** 1 Semester	**Cycle** every two years
Program	Master Scientific Computing (and Computer Science)			
Methods	Lecture 4 h + Computer exercises 2 h			
Lecturer	Dr. Filip Sadlo			
Objectives	To give introduce the students to the methods of geometric modeling and animation			
Content	Introduction to curves Interpolating curves Bézier curves B-Splines Rational curves Introduction to surfaces Tensor product surfaces Transfinite surfaces and extrusion Subdivision Subdivision surfaces Animation and simulation Rigid body kinematics Particle systems Mass-spring models Cloth modeling Numerical methods for differential equations Collision detection and handling Fluid simulation and natural phenomena			
Learning outcomes	The students know the mathematical foundations of geometric modeling know the mathematical and physical foundations of computer animation know the algorithms and implementation aspects are familiar with the basics of animated movies are able to apply existing tools for geometric modeling and animation			

Prerequisites	
Suggested previous knowledge	Einführung in die Praktische Informatik, Programmierkurs, Algorithmen und Datenstrukturen
Assessment(s)	Successful participation in the exercises (more than 50% have to be scored) und passing a written or oral exam.
Literature	- Curves and Surfaces for CAGD ? A Practical Guide, G. Farin, Morgan Kaufmann, 2002 - Computer Animation ? Algorithms and Techniques, R. Parent, Morgan Kaufmann, 2002 - 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, D. Eberly, Morgan Kaufmann, 2000 - Graphische Datenverarbeitung I, J. Encarnacao, W. StraSSer, R. Klein, 4. Auflage, Oldenbourg 1996 - Advanced Animation and Rendering Techniques, A. Watt, M. Watt, Addison-Wesley, 1992 - Grundlagen der geometrischen Datenverarbeitung, J. Hoschek, D. Lasser, Teubner 1992 - Numerical Recipes ? The Art of Scientific Computing, W.H. Press, P. Flannery, S.A. Teukolsky, W.T. Vetter- ling, Cambridge University Press, 1986

49 Randomized algorithms

Module	**Code** IRA	**Name** Randomized algorithms		
	Credit Points 6	**Workload** 180 h	**Duration** 1 Semester	**Cycle** every two years
Program	Master Scientific Computing (and Computer Science)			
Methods	Lecture 3 h + Exercises 1 h			
Lecturer	PD Dr. Wolfgang Merkle			
Objectives	Introduction to randomization techniques in the design and analysis of algorithms.			
Content	Basic probability theory The tenure game Derandomization techniques The probabilistic method Byzantine agreement Stable marriages and the Gale-Shapley algorithm Yao's minimax principle Complexity analysis of randomized sorting Randomized error detection and correction The Lovasz local lemma PAC-learning and VC-dimension Probability amplification and error bounds Local search for k-SAT Cryptographic protocols			
Learning outcomes	Based on application examples taken from various subareas of computer science, the students are able to apply the probabilistic viewpoint and method - with the construction and analysis of probabilistic and deterministic algorithms, - to combinatorial problems and constructions, - for the analysis of game-theoretic situations, - to cryptographic problems.			
Prerequisites	None			

Suggested previous knowledge	Basic knowledge of algorithms as can be obtained by the module IAD.
Assessment(s)	Solution of 50% of the assignments and oral or written exam
Literature	R. Motwani und P. Raghavan, Randomized Algorithms, Cambridge University Press 1995.

General Modules

50 Advanced Software Practical

Module	Code	Name		
	PR1	Advanced Software Practical		
	Credit Points	**Workload**	**Duration**	**Cycle**
	8 CP	240h	1 semester	
Program	Master Scientific Computing			
Course type	6 SWS Practical			
Objectives	Students acquire problem solving competence for complex design and implementation tasks and deepen their programming ability.			
Course description	Specific contents depends on the problem to be solved. General approach of a software practical includes • Execution and evaluation of projects and their phase structure • Planing of team work • Presentation of project results			
Skills to be acquired	Ability to carry out a project from concept to execution and presentation			
Prerequisites	none			
Helpful previous knowledge	Programming experience			
Grading policy	Grading of the documented software, the project report and the presentation of the results			
Suggested literature				

51 Field of Application (Anwendungsgebiet)

Module	**Code** SCAP	**Name** Field of Application		
	Credit Points 18	**Workload** 540	**Duration** 2 semesters	**Cycle**
Program	International Master Scientific Computing			
Methods	Lecture, excercise and/or practical			
Responsible	Dean of studies			
Objectives	Advanced proficiency and skills in an application field			
Content	The field of application is chosen according to the rules stated in the examination regulations. This includes that the same field of application as in the bachelor's degree should be chosen and that no modules can be credited that have been credited in the bachelor's degree. The collection of courses in the application field should be discussed preferably with the mentor of the specialization and has to be confirmed by the chairman of the examination board.			
Learning outcomes	The students have acquired knowledge that allows them to apply methods of scientific computation to their chosen field of application.			
Prerequisites	no prerequisites			
Suggested previous knowledge	Same field of application as in bachelor's degree			
Assessment(s)	Determined by the courses comprising the field of application			
Literature				

52 Master Thesis

Module	**Code**	**Name**		
	SCMa	Master Thesis		
	Credit Points	**Workload**	**Duration**	**Cycle**
	30	900 h: 810 h working on a thesis topic (research and development tasks) and 90 h for preparing and giving a thesis presentation	1 Semester	
Program	International Master Scientific Computing			
Methods				
Lecturer				
Objectives	Supervised research work in a specific research area of Scientific Computing			
Content	• Research work on a specific topic in Scientific Computing			
Learning outcomes				
Prerequisites				
Suggested previous knowledge				
Assessment(s)	Composite grade: 90% thesis (grade given by the two thesis examiners), 10% presentation (grade given by the two examiners)			
Literature	Given by supervisor.			