

**Modulhandbuch**  
**Master-Studiengang**  
**„Angewandte Informatik“**

**Ruprecht-Karls-Universität Heidelberg**  
**Fakultät für Mathematik und Informatik**

**Fassung vom 02.05.2016 zur Prüfungsordnung vom 22.07.2010  
mit letzter Änderung vom 07.02.2013**

**Studienform:** Vollzeit

**Art des Studiengangs:** Konsekutiv

**Regelstudienzeit:** 4 Semester

**Anzahl der im Studiengang zu erwerbenden Leistungspunkte:** 120

**Studienstandort:** Heidelberg

**Anzahl der Studienplätze:** Keine Zulassungsbeschränkung

**Gebühren/Beiträge:** Gemäß allgemeiner Regelung der Universität Heidelberg

# Präambel

## Einordnung und Gesamtdarstellung des Master-Studiengangs Angewandte Informatik

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 dem Curriculum und Modulen des Master-Studiengangs Angewandte Informatik 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.

Der Master-Studiengang Angewandte Informatik wird von der Fakultät für Mathematik und Informatik getragen. Der Master-Studiengang ist forschungsorientiert. Er vertieft und verbreitert die Fachkenntnisse, befähigt zum selbstständigen wissenschaftlichen Arbeiten, legt die Voraussetzungen zur Weiterentwicklung des Faches und bereitet auf eine anspruchsvolle Berufstätigkeit oder eine Promotion vor. AbsolventInnen sind qualifiziert für eigenverantwortliche und leitende Tätigkeiten. Der Master-Studiengang erlaubt eine sehr freie Gestaltung des Studiums um sowohl einen frühen Einstieg in forschungsnahen als auch innovative praktische Themengebiete zu ermöglichen. Weiterhin ermöglicht er eine Vertiefung in Themengebieten der Informatik, die insbesondere in den Natur-, Lebens- und Geisteswissenschaften zur Anwendung kommen.

### Qualifikationsziele des Master-Studiengangs Angewandte Informatik

Die Absolventen des Studiengangs sollen nach Abschluss des Studiums folgende grundlegende Kompetenzen überfachlicher Art im Kontext der Informatik besitzen.

- Sie besitzen Problemlösungskompetenz und beherrschen die Wissensanwendung im Bereich der Informatik und zusätzlich in einem breiteren fachlichen Zusammenhang oder verwandten Disziplinen. Darüber hinaus sind sie in der Lage, diese Kompetenzen auch in neuen, unvertrauten Situationen anzuwenden.
- Sie haben die Kompetenz zur Arbeit in einem Team sowie zur Übernahme von herausgehobener Verantwortung in einem Team (Teamleitung).

- Sie können eigene Schlussfolgerungen auf aktuellem Stand von Forschung und Anwendung vermitteln und sich fachbezogen auf wissenschaftlichem Niveau austauschen.
- Sie besitzen die Kompetenz zu selbständiger Informationssammlung, Urteilsfähigkeit und selbständiger Aneignung von Wissen im Bereich der Informatik sowie verwandten Disziplinen. Insbesondere sind sie befähigt zur Rezeption und Interpretation von Forschungsliteratur und zur Bewertung alternativer Lösungsansätze in fachlicher Hinsicht im Bereich der Informatik als auch fachübergreifend.
- Darüber hinaus beherrschen sie den effektiven Umgang mit komplexen Fachproblemen und Situationen, verfügen über Entscheidungsfähigkeit, sowie können selbständig forschungs- oder anwendungsorientierte Projekte durchführen.
- Sie können in fachlichen Angelegenheiten mündlich und schriftlich in deutscher und englischer Sprache effektiv kommunizieren.

In fachlicher Hinsicht beherrschen die AbsolventInnen des Master-Studiengangs Angewandte Informatik insbesondere die Kompetenzen der Bachelor-AbsolventInnen, im Detail:

- Sie verfügen über Kenntnisse der Praktischen, Theoretischen, Technischen und Angewandten Informatik und der Methoden der Mathematik und können diese zur Lösung von konkreten informatischen Problemen anwenden.
- Sie können eine informatische Aufgabe eigenverantwortlich planen, durchführen, dokumentieren und präsentieren.
- Sie können innerhalb einer vorgegebenen Frist ein Problem aus dem Bereich der Informatik mit wissenschaftlichen Methoden bearbeiten und Lösungsvorschläge entwickeln und präsentieren.
- Sie beherrschen wissenschaftlich fundierte Methoden der Programmierung und können diese in Projekten praktisch anwenden. Dazu gehören die wissenschaftlichen Methoden des Entwurfs, der Implementierung und des Debuggens von Software.
- Sie kennen die Konzepte für den Entwurf und die Analyse von effizienten Algorithmen und können diese bei der Erstellung von Software selbständig einsetzen.
- Sie kennen die Grundlagen der Verwendung von Betriebssystemen und Verwaltung von Ressourcen und sind in der Lage, diese Kenntnisse bei dem Entwurf, der Umsetzung und der Optimierung von informatischen Systemen einzusetzen.
- Sie kennen die Probleme und Bedeutung der Verlässlichkeit in modernen Computersystemen und Rechenverbunden und können diese Kenntnisse bei der Planung, Umsetzung als auch der Pflege solcher Systeme praktisch berücksichtigen.

Zusätzlich beherrschen die AbsolventInnen des Master-Studiengangs Angewandte Informatik folgende fachliche Qualifikationen über die Lernergebnisse des Bachelorstudiengangs hinaus.

- Sie sind in der Lage, umfangreiche informatische Systeme unter vorgegebenen technischen und ökonomischen Randbedingungen selbständig zu planen, zu entwerfen und zu evaluieren, sowie dazugehörige Softwareprojekte zu leiten.
- Sie verfügen über vertiefte Kenntnisse in einem oder mehreren Spezialgebieten der Informatik wie Datenanalyse, Requirements Engineering, Verteilte Systeme, Informationssysteme, und können diese Kenntnisse bei dem Entwurf und der Entwicklung von informatischen Systemen praktisch einsetzen.
- Sie können komplexe informatische Systeme in abstrakte Komponenten (Software und Hardware) zerlegen und dafür Realisierungsmöglichkeiten gemäß vorgegeben Randbedingungen ermitteln und bewerten, sowie diese Realisierung planen und umsetzen.
- Sie sind in der Lage, sich selbständig in zukünftige Techniken der Informatik also auch fachübergreifende Gebiete einzuarbeiten, diese in Projekten anzuwenden, sie fachlich zu kommunizieren, und in wissenschaftlicher Hinsicht zu entwickeln.

Aktuelle Forschungsschwerpunkte und Details zum Master-Studium Angewandte Informatik finden sich auf der Webseite [www.informatik.uni-heidelberg.de](http://www.informatik.uni-heidelberg.de).

# Inhaltsverzeichnis

<b>1 Pflichtbereich</b>	<b>8</b>
Wissenschaftliches Arbeiten	9
Seminar	10
Masterarbeit	11
Anwendungsgebiet	12
<b>2 Wahlpflichtbereich</b>	<b>13</b>
2.1 Gebietszuordnung der Module	13
2.2 Vertiefungen	16
Bildverarbeitung	16
Computergraphik und Visualisierung	17
Information Systems Engineering	19
Optimierung	21
Theoretische Informatik	23
Wissenschaftliches Rechnen	24
2.3 Module aus der Informatik	25
Big Data	26
Cloud Computing 1	28
Compilerbau	30
Computergraphik 1	32
Computergraphik 2	34
Computerspiele	35
Cybersicherheit und Internet Governance	36
Data Warehouses	40
Deep Vision	42
Effiziente Algorithmen 1	43
Effiziente Algorithmen 2	45
Einführung in das maschinelle Sehen in 3D	47
Formale Sprachen und Automatentheorie	48
Funddokumentation mittels optischer 3D-Scanner	51
Gaussian Processes for Machine Learning	52
Gemischt-ganzzahlige Programmierung und kombinatorische Optimierung	54
Geometric Modeling and Animation	56
Inverse Problems	58
IT-Projektmanagement	59
Knowledge Discovery in Databases	61
Künstliche Intelligenz	63

Machine Learning . . . . .	64
Fundamentals of Machine Learning . . . . .	65
Advanced Machine Learning . . . . .	66
Management und Analyse von Datenströmen . . . . .	67
Modellierung und Simulation in den Neurowissenschaften . . . . .	68
Modellierung, Optimierung und Regelung mechanischer Systeme . . . . .	70
Objekterkennung und automatisches Bildverstehen . . . . .	73
Objektorientiertes Programmieren im Wissenschaftlichen Rechnen . . . . .	75
Parallel Data Processing and Analysis . . . . .	76
Parallele Lösung großer Gleichungssysteme . . . . .	78
Paralleles Höchstleistungsrechnen . . . . .	79
Praktische Geometrie . . . . .	81
Projektseminar Biomedizinische Bildanalyse . . . . .	83
Qualitätsmanagement . . . . .	85
Randomisierte Algorithmen . . . . .	86
Räumliche Datenbanken . . . . .	88
Requirements Engineering . . . . .	90
Scientific Visualization . . . . .	91
Simulationswerkzeuge . . . . .	92
Social Network Analysis . . . . .	94
Software Evolution . . . . .	96
Software Ökonomie . . . . .	98
Software-Praktikum für Fortgeschrittene . . . . .	100
Verteilte Systeme I . . . . .	102
Visualisierung im Bereich Cultural Heritage . . . . .	104
Volumenvisualisierung . . . . .	105
Wissensmanagement und Entscheidungen im Software Engineering . . . . .	107
2.4 Module aus dem B.Sc./M.Sc. Mathematik . . . . .	109
Wahrscheinlichkeitstheorie . . . . .	110
Numerik . . . . .	111
Statistik . . . . .	112
Lineare Optimierung . . . . .	113
Nichtlineare Optimierung . . . . .	114
Wissenschaftliches Rechnen . . . . .	115
Computational Statistics . . . . .	116
Mathematische Logik . . . . .	118
Computeralgebra I . . . . .	119
Computeralgebra II . . . . .	121
Codierungstheorie . . . . .	122
Numerische Lineare Algebra . . . . .	123
Numerical methods for partial differential equations . . . . .	124
Numerische Optimierung bei Differentialgleichungen . . . . .	125
Statistik II . . . . .	126
Wahrscheinlichkeitstheorie II . . . . .	127
Berechenbarkeit und Komplexität I . . . . .	128

	Berechenbarkeit und Komplexität II . . . . .	129
	Algorithmische Optimierung I . . . . .	130
	Algorithmische Optimierung II . . . . .	131
	Mustererkennung . . . . .	132
	Mathematische Einführung in Compressed Sensing . . . . .	133
	Statistische Datenanalyse . . . . .	134
	Implementation of numerical methods for partial differential equations . .	136
2.5	Module aus dem M.Sc. Scientific Computing . . . . .	137
	Fundamentals of Computational Environmental Physics . . . . .	137
2.6	Module aus dem M.Sc. Technische Informatik . . . . .	139
2.7	Module aus dem B.Sc./M.Sc. Physik . . . . .	140
	Physics of Imaging . . . . .	140
	Image Processing . . . . .	141
	Introduction to Image Processing on the GPU . . . . .	142
	Fast Parallel Implementations of Image Labeling Problems . . . . .	144
2.8	Module aus dem B.S./M.Sc. Biologie . . . . .	146
	Model-based segmentation of biomedical images (Modellbasierte Segmen- tierung Biomedizinischer Bilder) . . . . .	146

# 1 Pflichtbereich

Im Master-Studiengang Angewandte Informatik ist das Modul *Wissenschaftliches Arbeiten* ein Pflichtmodul. Weiterhin müssen auch zwei Seminare absolviert werden. Nachfolgend werden beide Module sowie die Module *Masterarbeit* und *Anwendungsgebiet* beschrieben.



## Wissenschaftliches Arbeiten

<b>Code</b> IWA	<b>Name</b> Wissenschaftliches Arbeiten	
<b>Leistungspunkte</b> 2 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Wintersemester
<b>Lehrform</b> 2 SWS	<b>Arbeitsaufwand</b> 60 h; davon 30 h Präsenzstudium 30 h Selbststudium und praktische Übungen (eventuell in Gruppen)	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden</p> <p>kennen die wichtigsten Literaturquellen der Informatik; wissen, welche Tools und Techniken zur Verwaltung von Literatur existieren und wie diese verwendet werden; sind in der Lage, wissenschaftliche Texte (z.B. aus Tagungsbänden oder Journals) und Vorträge kritisch zu lesen und zu bewerten und sie kompakt zusammenzufassen; kennen die einschlägigen Techniken zur Präsentation eines wissenschaftlichen Vortrags; kennen die Möglichkeiten des wissenschaftlichen Publizierens und die Organisation von wissenschaftlichen Tagungen; kennen unterschiedliche Forschungsmethoden ; kennen die Möglichkeiten einer Tätigkeit im wissenschaftlichen Umfeld nach dem Studium; kennen aktuelle Forschungsarbeiten in der Informatik; haben einen Überblick über die Wege der Finanzierung von Forschungsarbeiten; kennen die Anforderungen an die Struktur von Anträgen zur Forschungsförderung.</p>	
<b>Inhalt</b>	<p>Literaturrecherche und -verwaltung Wissenschaftliches Vortragen, Schreiben, Publizieren und Begutachten Forschungsförderung über Drittmittel Forschungsmethoden und aktuelle Forschungsprojekte Wissenschaftliches Arbeiten nach dem Studium</p>	
<b>Voraussetzungen</b>	keine	
<b>Prüfungsmodalitäten</b>	Regelmäßige Teilnahme, Teilnahme an praktischen Übungen (u.a. durch Bearbeitung von Hausaufgaben und Projekten)	
<b>Nützliche Literatur</b>		

## Seminar

<b>Code</b> IS	<b>Name</b> Seminar	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Semester
<b>Lehrform</b> Seminar 2 + 2 SWS (Seminar/Tutorium)	<b>Arbeitsaufwand</b> 120 h; davon 30 h Präsenzstudium 90 h Vorbereitung Vortrag	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Kenntnis von Techniken des wissenschaftlichen Schreibens (insbesondere auch Literaturrecherche) Fähigkeit, komplexe wissenschaftliche Literatur zu erschließen</p> <p>Erweiterte Fähigkeit, komplexe wissenschaftliche Literatur in einem Vortrag zu präsentieren</p> <p>Erweiterte Fähigkeit, zu Vorträgen zu diskutieren und Feedback zu geben</p> <p>Fähigkeit, ein kurze wissenschaftliche Ausarbeitung zu einem komplexen Thema zu erstellen</p>	
<b>Inhalt</b>	<p>Einführung in und Einübung von Techniken des wissenschaftlichen Schreibens</p> <p>Vertiefte Einübung der Erschließung und Präsentation wissenschaftlicher Literatur</p> <p>Fortgeschritteneres Informatikthema</p>	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse im Themengebiet des Seminars	
<b>Prüfungsmodalitäten</b>	<p>Regelmäßige Teilnahme an den Seminarterminen und Mitwirkung in den Diskussionen,</p> <p>Ausarbeitung und Halten eines Vortrages von etwa 60 Minuten Dauer (inklusive Diskussion),</p> <p>schriftliche Ausarbeitung von ca. 10 Seiten</p>	
<b>Nützliche Literatur</b>		

## Masterarbeit

<b>Code</b> IMa	<b>Name</b> Masterarbeit	
<b>Leistungspunkte</b> 30 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Betreutes Selbststudium 2 SWS, Kolloquium 1 SWS	<b>Arbeitsaufwand</b> 900 h; davon 810 h Bearbeitung eines individuellen Themas (Forschungs- und Entwicklungsarbeiten) und schriftliche Ausarbeitung 90 h Ausarbeitung Vortrag und Präsentation und Mitwirkung Kolloquium	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik
<b>Lernziel</b>	Einsatz der erlernten Fachkenntnisse und Methoden zum selbstständigen Lösen einer komplexen Problemstellung aus der Informatik und ihren Anwendungen Fähigkeit, in großem Umfang selbstständig eine anspruchsvolle wissenschaftlichen Arbeit zu erstellen Fähigkeit, eine eigene, anspruchsvolle Arbeiten in einem wissenschaftlichen Vortrag darzustellen	
<b>Inhalt</b>	selbstständiges wissenschaftliches Bearbeiten einer beschränkten Aufgabenstellung aus der Informatik und ihren Anwendungen wissenschaftlicher Vortrag über die Problemstellung und die erarbeitete Lösung	
<b>Voraussetzungen</b>	nach Prüfungsordnung mindestens 45 LP; weiterhin sind empfohlen: Wahlpflichtvorlesungen und Module Seminar (IS) und Fortgeschrittenenpraktikum (IFP)	
<b>Pruefungs- modalitaeten</b>	regelmäßige Treffen mit der/dem BetreuerIn, schriftliche Ausarbeitung und Präsentation	
<b>Nuetzliche Literatur</b>		

## Anwendungsgebiet

<b>Code</b> IAG	<b>Name</b> Anwendungsgebiet	
<b>Leistungspunkte</b> 18 LP	<b>Dauer</b>	<b>Turnus</b>
<b>Lehrform</b> Vorlesung, Übung und / oder Praktikum	<b>Arbeitsaufwand</b> 540 h, Aufteilung in Präsenz- / Übungs- und Praktikumszeit in Absprache mit den Dozentinnen bzw. Dozenten	<b>Verwendbarkeit</b> M.Sc. Angewandte Informatik
<b>Lernziel</b>	Vertiefte Kenntnisse und Fähigkeiten in einem Anwendungsgebiet	
<b>Inhalt</b>	<p>Wahl eines Anwendungsgebietes nach den Regeln der Prüfungsordnung Festlegung von und Teilnahme an Modulen aus dem Anwendungsgebiet (die LP entsprechen dabei den Vorgaben aus dem Anwendungsgebiet). Dabei ist sicherzustellen, dass keine Module aus dem Anwendungsgebiet gewählt werden, die schon im Bachelor-Studium eingebracht wurden.</p> <p>(optional) Definition und Durchführung eines interdisziplinären Projektes, d.h. Festlegung eines Dozenten / einer Dozentin aus dem Anwendungsgebiet und der Informatik Gemeinsame Festlegung des Projektziels durch die Dozentinnen bzw. Dozenten und den/die Studierende. Das Projektziel umfasst eine informatische Leistung im Anwendungsgebiet.</p> <p>Festlegung des Arbeitsaufwandes und damit der LP für das Projekt Durchführung des Projekts entsprechend des Projektziels unter Anleitung der Dozentinnen bzw. Dozenten Dokumentation des Ergebnisses Erstellung eines Projektbericht Präsentation des Ergebnisses</p>	
<b>Voraussetzungen</b>	empfohlen ist das gleiche Anwendungsgebiet wie im Bachelor	
<b>Prüfungsmodalitäten</b>	<p>Von den 18 LP sind mindestens 8 LP durch einen nicht-informatischen Modul auf Masterniveau zu erbringen (Ausnahmegenehmigung möglich), die restlichen 10 LP können durch ein Projekt oder Module auf Bachelorniveau erbracht werden.</p> <p>Prüfungsleistungen in dem Anwendungsgebiet und (optional) Prüfungsleistungen für das interdisziplinäre Projekt analog zum Modul IFP, gewichtet nach dem jeweiligen Anteil der LP</p>	
<b>Nützliche Literatur</b>		

## 2 Wahlpflichtbereich

Im Folgenden sind die Wahlpflichtmodule des Masterstudiengangs Angewandte Informatik beschrieben. Es können (aber müssen nicht) Vertiefungen gewählt werden, dabei sind die nachfolgenden Anmerkungen zu beachten. Wie in der Prüfungsordnung beschrieben, sind bei der Wahl der Module Gebiete abzudecken. Die Zuordnung der Module zu den Gebieten ist im nachfolgenden Kapitel beschrieben. Anschließend folgen die Beschreibungen der Vertiefungen und dahinter die einzelnen Modulbeschreibungen.

### 2.1 Gebietszuordnung der Module

Entsprechend der in der Prüfungsordnung genannten Vorgaben ist eine Auswahl aus den folgenden Gebieten abzudecken:

Bildverarbeitung  
Computergraphik und Visualisierung (CGV)  
Datenbanksysteme (DB)  
Optimierung  
Parallele und Verteilte Systeme  
Software Engineering (SWE)  
Technische Informatik  
Theoretische Informatik  
Wissenschaftliches Rechnen

Nachfolgend werden die Module den einzelnen Gebieten zugeordnet.  
Alle nicht genannten Module sind keinem speziellen Gebiet zugeordnet.

#### **Bildverarbeitung**

Deep Vision (IDV)  
Einführung in das maschinelle Sehen in 3D (I3DCV)  
Funddokumentation mittels optischer 3D-Scanner (I3Doc)  
Projektseminar Biomedizinische Bildanalyse (IPBB)  
Physics of Image Processing (MWInf5)  
Bildverarbeitung (MWInf6 oder MH19)  
Statistische Mustererkennung (MWInf7 oder MH18)  
Objekterkennung und automatisches Bildverstehen (IOAB)  
Modellbasierte Segmentierung biomedizinischer Bilder (IMSBI)  
Introduction to Image Processing on the GPU (IGPU)

Fast Parallel Implementations of Image Labeling Problems (IMLP)

### **Computergraphik und Visualisierung (CGV)**

Computergraphik 1 (ICG1)  
Computergraphik 2 (ICG2)  
Computerspiele (ICS)  
Einführung in das maschinelle Sehen in 3D (I3DCV)  
Geometric Modeling and Animation (IGMA)  
Inverse Probleme (IIP)  
Künstliche Intelligenz (IKI)  
Praktische Geometrie (IPG)  
Scientific Visualization (ISV)  
Visualisierung im Bereich Cultural Heritage (IVCH)  
Volume Visualization (IVV)

### **Datenbanksysteme (DB)**

Data Warehouses (IDW)  
Knowledge Discovery in Databases (IKDD)  
Management und Analyse von Datenströmen (IMADS)  
Räumliche Datenbanken (IRDB)  
Social Network Analysis (ISNA)

### **Optimierung**

Effiziente Algorithmen 1 (IEA1)  
Effiziente Algorithmen 2 (IEA2)  
Mixed-Integer Programming and Combinatorial Optimization (IMIP)  
Lineare Optimierung (MD3)  
Nichtlineare Optimierung (MD4)  
Numerische Optimierung bei Differentialgleichungen (MH8)  
Numerische Methoden der Kontinuumsmechanik (MH9)  
Spezielle Themen der Numerik (MH10)  
Spezielle Themen der Optimierung (MH11)  
Algorithmische Optimierung I (MH16)  
Algorithmische Optimierung II (MH17)

### **Parallele und Verteilte Systeme**

Big Data (IBD)  
Cloud Computing 1 (ICC1)  
Parallel data Processing and Analysis (IPDPA)  
Verteilte Systeme I (IVS1)

### **Software Engineering (SWE)**

IT-Projektmanagement (IPM)

Software Evolution (ISWEvolv)  
Wissensmanagement und Entscheidungen im Software Engineering (ISWKM)  
Software Ökonomie (ISWÖk)  
Qualitätsmanagement (ISWQM)  
Requirements Engineering (ISWRE)

### **Technische Informatik**

Dieses Gebiet umfasst alle Grundlagen- und Vertiefungsmodule des Masters Technische Informatik.

### **Theoretische Informatik**

Formale Sprachen und Automatentheorie (IFSA)  
Randomisierte Algorithmen (IRA)  
Mathematische Logik (MB9)  
Berechenbarkeit und Komplexität I (MH14)  
Berechenbarkeit und Komplexität II (MH15)

### **Wissenschaftliches Rechnen**

Machine Learning (IML)  
Fundamentals of Machine Learning (IFML)  
Advanced Machine Learning (IAML)  
Modellierung, Optimierung und Regelung mechanischer Systeme (IMORMS)  
Modellierung und Simulation in den Neurowissenschaften (IMSN)  
Objektorientiertes Programmieren im Wissenschaftlichen Rechnen (IOPWR)  
Paralleles Höchstleistungsrechnen (IPHR)  
Parallele Lösung großer Gleichungssysteme (IPLGG)  
Simulationswerkzeuge (ISIMW)  
Numerik (MD1)  
Wissenschaftliches Rechnen (MD5)  
Numerische Lineare Algebra (MH5)  
Numerik Partieller Differentialgleichungen (MH7)

## 2.2 Vertiefungen

### Vertiefung Bildverarbeitung

Die Vertiefung wird gestaltet vom Heidelberger Zentrum für Bildverarbeitung (Heidelberg Collaboratory Image Processing). Sie kann im Master gewählt werden.

Sie umfasst 2 Seminare, Wahlpflichtveranstaltungen und die Masterarbeit. Es sind 2 Seminare und vertiefende Vorlesungen aus dem Gebiet der Bildverarbeitung zu wählen und die Masterarbeit in diesem Gebiet anzufertigen. Weitere Module in der Informatik sowie das Anwendungsgebiet können gemäß der Prüfungsordnung frei gewählt werden. Ansprechpartner für diese Vertiefung sind Professor Dr. Jähne, Professor Dr. Hamprecht und Professor Dr. Ommer. Vor Beginn der Vertiefung wird ein Beratungsgespräch bei einem der Ansprechpartner empfohlen.

Die Veranstaltungen aus dem Gebiet der Bildverarbeitung sind:

- Physics of Image Processing (MWInf5) 4 LP
- Bildverarbeitung (MWInf6 oder MH19) 8 LP
- Statistische Mustererkennung (MWInf7 oder MH18) 8 LP
- Objekterkennung und automatisches Bildverstehen (IOAB) 8 LP
- Multidimensionale Signalanalyse (TIMDS)

Dabei sind die Module MWInf6 und MWInf7 mehr auf die Physik ausgerichtet und die Module MH18 und MH19 mehr auf die Mathematik.

<b>Veranstaltung</b>	<b>Semester</b>	<b>LP</b>
Wissenschaftliches Arbeiten	1	2
Seminar - Bildverarbeitung	1 - 3	4
Seminar - Bildverarbeitung	1 - 3	4
Bildverarbeitung (MWInf6 oder MH19)	1 - 3	8
Statistische Mustererkennung (MWInf7 oder MH18)	1 - 3	8
Weitere Module aus dem Gebiet der Bildverarbeitung	1 - 3	20
Weitere Module aus mindestens 2 anderen Gebieten	1 - 3	26
Masterarbeit Bildverarbeitung	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		<b>120</b>



## Vertiefung Computergraphik und Visualisierung

Diese Vertiefung befähigt zur Entwicklung von Algorithmen und Anwendungsprogrammen für die visuelle Datenverarbeitung und -analyse. Hierzu gehören Kenntnisse im Bereich Mensch-Maschine-Interaktion, Computergraphik, Datenanalyse und wissenschaftliche Visualisierung. Die Vertiefung kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst mindestens ein Seminar, ein Praktikum, die Masterarbeit und zusätzliche Wahlpflichtveranstaltungen. Die Module sind aus dem Lehrgebiet Computergraphik und Visualisierung (CGV) zu wählen. Weitere Veranstaltungen sollten aus zwei ergänzenden Lehrgebieten gewählt werden.

Ansprechpartner für diese Vertiefung sind Herr Professor Dr. Sadlo und Frau Dr. Krömker. Vor Beginn der Vertiefung wird ein Beratungsgespräch bei einem der Ansprechpartner empfohlen.

Vertiefende Module aus dem Gebiet CGV sind:

- Computergraphik 1 (ICG1) 6 LP
- Computergraphik 2 (ICG2) 6 LP
- Computerspiele (ICS) 8 LP
- Einführung in das maschinelle Sehen in 3D (I3DCV) 2 LP
- Geometric Modeling and Animation (IGMA) 8 LP
- Künstliche Intelligenz (IKI) 6 LP
- Praktische Geometrie (IPG) 4 LP
- Scientific Visualization (ISV) 8 LP
- Visualisierung im Bereich Cultural Heritage (IVCH) 2 LP
- Volume Visualization (IVV) 8 LP

Für das ergänzende Lehrgebiet bieten folgende Bereiche eine gute Erweiterung.

Bildverarbeitung

Datenbanken

Optimierung

Software Engineering

Wissenschaftliches Rechnen

Weitere Bereiche können auf Antrag genehmigt werden.

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
Seminar - Lehrgebiet CGV	1 - 3	4
Seminar - CGV oder ergänzendes Lehrgebiet	1 - 3	4
Fortgeschrittenen-Praktikum CGV	1 - 3	8
Vertiefende Module aus dem Bereich CGV	1 - 3	20
Weitere Module aus mindestens 2 anderen Gebieten	1 - 3	34
Masterarbeit CGV	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		120

Bei Belegung der Vertiefung im Bachelor und Master müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

## Vertiefung Information Systems Engineering

Diese Vertiefung befähigt zu Entwicklung, Betrieb und Wartung von komplexen Informationssystemen. Sie kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst die Seminare, Praktika, Bachelor- bzw. Masterarbeiten und die Wahlpflichtveranstaltungen. Dabei sind die Module zu gleichen Teilen auf die beiden Lehrgebiete Datenbanken (DB) und Software Engineering (SWE) verteilt. Kernstück ist ein die beiden Lehrgebiete übergreifendes Praktikum (genannt ISE-Projekt) im Umfang von 16 LP, in dem im Team für externe Kunden ein Informationssystem mit ingenieurmäßigen Methoden unter Nutzung modernster Technologie entwickelt wird. Das ISE-Projekt wird immer im Wintersemester angeboten. Ansprechpartner für diese Vertiefung sind Herr Professor Dr. Gertz (DB) und Frau Professor Dr. Paech (SWE). Vor Beginn der Vertiefung wird ein Beratungsgespräch bei einem der Ansprechpartner empfohlen.

Vertiefende Module aus dem Gebiet SWE:

- Requirements Engineering (ISWRE) 8 LP
- Qualitätsmanagement (ISWQM) 8 LP

Vertiefende Module aus dem Gebiet DB:

- Data Warehouses (IDW) 4 LP
- Knowledge Discovery in Databases (IKDD) 8 LP
- Räumliche Datenbanken (IRDB) 8 LP

Weitere angebotene Module sind:

- DB: Management und Analyse von Datenströmen (IMADS) 4 LP
- SWE: IT-Projektmanagement (IPM) 3 LP
- SWE: Software-Ökonomie (ISWÖk) 3 LP
- SWE: Software-Evolution (ISWEvolv) 3 LP
- SWE: Wissensmanagement in der Softwareentwicklung (ISWKM) (3 LP)

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
Seminar - Lehrgebiet SWE	1 - 3	4
Seminar - Lehrgebiet DB	1 - 3	4
ISE-Projekt	1 - 3	16
2 Vertiefende Module Lehrgebiet SWE	1 - 3	16
2 Vertiefende Module Lehrgebiet DB	1 - 3	16
Weitere Module	1 - 3	6
Effiziente Algorithmen 1	1 - 3	8
Masterarbeit SWE oder DB	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		120

Wurde das ISE-Projekt schon im Bachelor gemacht, so sollten im Master je 1 Praktikum in DB und SWE gemacht werden. Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

## Vertiefung Optimierung

Diese Vertiefung hat als Inhalt Entwurf, Analyse und Implementierung von Algorithmen zur Lösung von kontinuierlichen und ganzzahligen Optimierungsproblemen. Sie kann im Bachelor- oder Master-Studium sowie durchgängig vom Bachelor- zum Master-Studium gewählt werden. Sie umfasst neben Vorlesungen, Seminaren und Praktika auch die Bachelorarbeit. Voraussetzungen sind gute bis sehr gute Kenntnisse aus den Pflichtmodulen Mathematik. Ansprechpartner für diese Vertiefung ist Herr Professor Dr. Reinelt. Vor Beginn der Vertiefung wird ein Beratungsgespräch empfohlen.

Den Kern der Veranstaltungen dieser Vertiefung bilden die folgenden Module aus dem Gebiet Optimierung:

- Effiziente Algorithmen 1 (IEA1) 8 LP
- Effiziente Algorithmen 2 (IEA2) 8 LP
- Algorithmische Optimierung 1 (MH16) 8 LP
- Algorithmische Optimierung 2 (MH17) 8 LP

sowie aus dem Gebiet Wissenschaftliches Rechnen:

- Numerische Optimierung bei Differentialgleichungen (MH8) 8 LP

Alle diese Module können dabei unabhängig voneinander absolviert werden.

Wird die Vertiefung sowohl im Bachelor- als auch im Masterstudium gewählt, sollten alle 5 Module absolviert werden.

Weitere erforderliche Module sind:

mindestens ein Seminar 4 LP,  
mindestens ein Fortgeschrittenen-Praktikum 8 LP.

Empfohlen werden, je nach aktuellem Angebot, weitere Module wie z.B. Kompaktkurs Gemischt-ganzzahlige und kombinatorische Optimierung (MIP) und Modul Numerik (MD1) (besonders für Optimierung bei Differentialgleichungen).

Möglich sind weiterhin:

- Lineare Optimierung (MD3)
- Nichtlineare Optimierung (MD4)
- Numerische Optimierung bei Differentialgleichungen (MH8)
- Numerische Methoden der Kontinuumsmechanik (MH9)
- Spezielle Themen der Numerik (MH10)
- Spezielle Themen der Optimierung (MH11)
- Mixed-Integer Programming and Combinatorial Optimization (IMIP)

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
Seminar Optimierung	1 - 3	4
Seminar (muss nicht Optimierung sein)	1 - 3	4
Fortgeschrittenen-Praktikum Optimierung	1 - 3	8
3 aus den oben genannten 5 Modulen	1 - 3	24
Weitere Module aus dem Gebiet Optimierung	1 - 3	8
Weitere Module aus mindestens 2 anderen Gebieten	1 - 3	22
Masterarbeit Optimierung	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		120

Für weitere Module werden besonders die Gebiete Wissenschaftliches Rechnen, Bildverarbeitung und Datenbanken empfohlen. Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.

## Vertiefung Theoretische Informatik

Die Vertiefung kann im Master gewählt werden. Sie umfasst Seminare, Wahlpflichtveranstaltungen und die Masterarbeit. Es sind 2 Seminare und 3 vertiefende Vorlesungen aus dem Gebiet der Theoretischen Informatik zu wählen und die Masterarbeit in diesem Gebiet anzufertigen. Weiter ist die Vorlesung Effiziente Algorithmen 1 verpflichtend. Weitere Lehrveranstaltungen in der Informatik sowie das Anwendungsgebiet können gemäß der Prüfungsordnung frei gewählt werden. Ansprechpartner für diese Vertiefung ist Herr Professor Dr. Ambos-Spies. Vor Beginn der Vertiefung wird ein Beratungsgespräch empfohlen.

Vertiefende Vorlesungen aus dem Bereich Theoretische Informatik:

- Berechenbarkeit und Komplexität 1 (MH14) 8 LP
- Berechenbarkeit und Komplexität 2 (MH15) 6 LP
- Formale Sprachen und Automatentheorie (IFSA) 8 LP

Auf Antrag kann eine dieser Vorlesungen durch eine vertiefende Vorlesung über effiziente Algorithmen ersetzt werden.

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
2 Seminare - Lehrgebiet Theoretische Informatik	1 - 3	8
3 Vertiefende Vorlesungen - Lehrgebiet Theoretische Informatik	1 - 3	22
Effiziente Algorithmen 1	1 - 3	8
Weitere Module Informatik (Wahlpflichtmodule)	1 - 3	32
Masterarbeit - Lehrgebiet Theoretische Informatik	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		120

## Vertiefung Wissenschaftliches Rechnen

Die Vertiefung kann im Bachelor, im Master und durchgängig vom Bachelor zum Master gewählt werden. Sie umfasst die Seminare, Praktika, Bachelor- und Masterarbeiten und die Wahlpflichtveranstaltungen. Ansprechpartner für diese Vertiefung ist Herr Professor Dr. Bastian. Vor Beginn der Vertiefung wird ein Beratungsgespräch empfohlen.

Vertiefende Vorlesungen aus dem Gebiet Wissenschaftliches Rechnen:

- Objektorientiertes Programmieren im Wissenschaftlichen Rechnen (IOPWR) 6 LP
- Paralleles Höchstleistungsrechnen (IPHR) 8 LP
- Parallele Lösung großer Gleichungssysteme (IPLGG) 8 LP
- Simulationswerkzeuge (ISIMW) 8 LP
- Numerik 1 (MD1) 8 LP
- Wissenschaftliches Rechnen (MD5) 8 LP
- Numerische Lineare Algebra (MH5) 8 LP
- Numerik gewöhnlicher Differentialgleichungen (MH6) 8 LP
- Numerik partieller Differentialgleichungen (MH7) 8 LP

Falls die Vertiefung noch nicht im Bachelor gewählt wurde, sollte das Modul MD1 gewählt werden.

Veranstaltung	Semester	LP
Wissenschaftliches Arbeiten	1	2
2 Seminare Wissenschaftliches Rechnen	1 - 3	8
Objektorientiertes Programmieren im Wissenschaftlichen Rechnen (IOPWR)	1 - 3	6
Numerik partieller Differentialgleichungen (MH7)	1 - 3	8
Numerische Optimierung bei Differentialgleichungen (MH8)	1 - 3	8
Paralleles Höchstleistungsrechnen (IPHR)	1 - 3	8
Algorithmische Optimierung I (MH16)	1 - 3	8
Eine weitere Veranstaltung aus: MH17, IPLGG, ISIMW, IMIP	1 - 3	8
Weitere Module aus mindestens 1 anderem Gebiet (empfohlen: SWE)	1 - 3	16
Masterarbeit Wissenschaftliches Rechnen	4	30
Anwendungsgebiet	1 - 3	18
<b>LP Summe</b>		120

Bei Belegung der Vertiefung im Bachelor müssen die im Master belegten Module disjunkt zu denen im Bachelor sein.



## 2.3 Module aus der Informatik

Nachfolgend sind die Module aus der Informatik in alphabetischer Reihenfolge beschrieben.

## Big Data

<b>Code</b> IBD	<b>Name</b> Big Data	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 90 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Hausaufgabenbearbeitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden</p> <p>kennen den prinzipiellen Aufbau einer Big Data Anwendung haben die erforderlichen Grundlagen gelernt (Technologien, Methoden und Konzepte)</p> <p>sind mit den Sicherheitsproblematiken von Big Data Anwendungen vertraut und können eine adäquate Architektur konzipieren</p> <p>können Big Data Konzepte in Anwendungen umsetzen sind in der Lage, öffentliche und private Daten in Anwendungen zu aggregieren</p> <p>können die erworbenen Kenntnisse auf verschiedene Anwendungsbereiche übertragen</p>	
<b>Inhalt</b>	<p>Die Vorlesung bietet eine praktische Einführung in das aktuelle Gebiet Big Data, wobei auch Cloud Ressourcen im Internet für eigene Anwendungen transparent genutzt werden können. Schwerpunkte sind u.a.:</p> <p>Fundamentals: Volume, Variety, Velocity, Veracity, Visualization, Value</p> <p>Architecture: Hadoop as a Service, Data Warehouse</p> <p>Data Management: Movement, Monitoring, Provenance, Preservation, SLAs</p> <p>Data Publication: Preparation, Curation, Discovery, Open Data, Open Access</p> <p>Data Security, Privacy &amp; Trust: Risk Management, Usage Control, Trusted Datastore</p>	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie im Modul ISW, IBN und ICC vermittelt werden. Modul IPDPA wird als Ergänzung bzw. Erweiterung empfohlen.	
<b>Prüfungs- modalitäten</b>	Teilnahme an der Vorlesung mit Hausaufgaben und Bestehen einer Klausur oder mündlichen Abschlussprüfung	

<p><b>Nuetzliche Literatur</b></p>	<p>BITKOM Leitfaden Big-Data-Technologien, 2014 (online)          Bücher aus dem O'Reilly Data Science Starter Kit, 2014 (online)          L.Barroso, J.Clidas, U.Hölzle, The Data Center as a Computer:          An Introduction to the Design of Warehouse-Scale Machines,          Morgan&amp;Claypool, 2013 (online)          A.Murthy, V.Vavilapalli, D.Eadline, J.Niemiec, Apache Hadoop          YARN: Moving beyond MapReduce and Batch Processing with          Apache Hadoop 2, Addison Wesley Data&amp;Analytics, 2014          C.Baun, M.Kunze, J.Nimis, S.Tai, Cloud Computing - Web          basierte dynamische IT-Services, Informatik im Fokus, Springer          Verlag 2011, 2. Auflage          C.Bengel, C.Baun, M.Kunze, U.Stucky, Masterkurs Verteilte und          Parallele Systeme, Springer Verlag 2008</p>
--	--

# Cloud Computing 1

<b>Code</b> ICC1	<b>Name</b> Cloud Computing 1	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 90 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Hausaufgabenbearbeitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden kennen den prinzipiellen Aufbau einer Computing Cloud haben die erforderlichen Grundlagen gelernt (Virtualisierung, Web Services) sind mit den Sicherheitsproblematiken von Cloud-Systemen vertraut und können eine adäquate Architektur konzipieren können Konzepte wie IaaS, PaaS und SaaS in Anwendungen umsetzen sind in der Lage, öffentliche und private Cloud-Systeme zu nutzen</p> <p>können die erworbenen Kenntnisse auf verschiedene Anwendungsbereiche zu übertragen</p>	
<b>Inhalt</b>	<p>Die Vorlesung bietet eine praktische Einführung in das aktuelle Gebiet Cloud Computing, bei dem Ressourcen im Internet für eigene Anwendungen transparent genutzt werden können. Schwerpunkte sind u.a.:</p> <ul style="list-style-type: none"> <li>Grundlagen</li> <li>Virtualisierung und Web Services</li> <li>Cloud Angebote im Internet (Taxonomie)</li> <li>Cloud Security</li> <li>Programmiermodelle</li> <li>Wiss. Rechnen in der Cloud</li> <li>High Performance Computing as a Service</li> <li>Big Data</li> </ul>	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in den Modulen Einführung in Software Engineering (ISW) sowie Betriebssysteme und Netzwerke (IBN) vermittelt werden	
<b>Prüfungsmodalitäten</b>	Teilnahme an der Vorlesung mit Hausaufgaben und Bestehen einer mündlichen Abschlussprüfung	

<b>Nuetzliche Literatur</b>	C.Baun, M.Kunze, J.Nimis, S.Tai, Cloud Computing - Web basierte dynamische IT-Services, Informatik im Fokus, Springer Verlag 2011, 2. Auflage C.Bengel, C.Baun, M.Kunze, U.Stucky, Masterkurs Verteilte und Parallele Systeme, Springer Verlag 2008
---------------------------------	---

## Compilerbau

<b>Code</b> ICOM	<b>Name</b> Compilerbau	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 20 h Prüfungsvorbereitung 100 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden kennen den prinzipiellen Aufbau eines Compilers haben die erforderlichen Grundlagen zu formalen Sprachen gelernt</p> <p>sind mit den Problematiken von lexikalischer, syntaktischer und semantischer Analyse vertraut kennen die Techniken zur Compilererstellung sind über Optimierungsmöglichkeiten informiert sind in der Lage, vorhandene Werkzeuge zum Compilerbau zu benutzen können die erworbenen Kenntnisse auf andere Anwendungsbereiche, wie z.B. Aufbereitung strukturierter Daten, übertragen</p>	
<b>Inhalt</b>	<p>Überblick über grundlegende Techniken Grundlagen der formale Sprachen Lexikalische Analyse Top-Down-Syntaxanalyse Bottom-Up-Syntaxanalyse Syntaxgesteuerte Übersetzung Semantische Analyse Die C-Maschine 3-Adress-Code Optimierung Datenfluss-Analyse</p>	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Prüfungs- modalitäten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50 % der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung	

<b>Nuetzliche Literatur</b>	z.B. Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques and Tools, Pearson - Addison-Wesley, 2006
---------------------------------	--

# Computergraphik 1

<b>Code</b> ICG1	<b>Name</b> Computergraphik 1	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Wintersemester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium, 100 h Aufgabenbearbeitung, 20 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	<p>Die Studierenden sind in der Lage, einfache Graphikfenster mit Inhalt zu belegen, sind mit 2D-Rastergraphik vertraut und in der Lage, mit Objekten in Weltkoordinaten über Graphikbibliotheken (OpenGL) zu interagieren, können mit homogenen Koordinaten umgehen und Projektions- sowie Transformationsmatrizen bestimmen. Sie kennen die verschiedenen Bufferkonzepte zur schnellen Darstellung realistischer Effekte, sind mit Farbkonzepten und Darstellungsmodellen vertraut, können lokale Lichtmodelle aus optischen Prinzipien herleiten und wissen, wie man Texturen für unterschiedliche Effekte einsetzt. Sie können die physikalischen Grundlagen für globale Lichtmodelle in Fremdsoftware abrufen. Das sichere Beherrschen der Programmierung kleinerer Graphikprogramme und die Generierung von Einzelbildern, 3D Grafiken und Animationen für die unterschiedlichen Zwecke des wissenschaftlichen Rechnens wird auf Basis der Programmierbibliothek OpenGL (Open Graphics Library) vermittelt.</p>	
<b>Inhalt</b>	<p>Grundlagen der Graphikprogrammierung          Koordinatensysteme          Projektionen, Transformationen          Zeichenalgorithmen          Bufferkonzepte (z-Buffer, Double-Buffer, Stereo-Buffer)          Shading und Lichtmodelle          Graphikbibliothek OpenGL          Direct Rendering (lokale Verfahren)          Texturen und Abbildungsverfahren          Globale Verfahren          Raytracing (Povray)          Volume Rendering</p>	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK)	



<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
<b>Nuetzliche Literatur</b>	J. D. Foley, A. Van Dam, S. K. Feiner, J . F. Hughes und R. L. Phillips: Computer Graphics: Principles and Practice. Addison-Wesley, 1996 D. Shreiner, M. Woo, J. Neider und T. Davis: OpenGL programming guide: the official guide to learning OpenGL, version 1.4. OpenGL Architecture Review Board, Addison-Wesley, 2004

## Computergraphik 2

<b>Code</b> ICG2	<b>Name</b> Computergraphik 2	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Sommersemester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium, 100 h Aufgabenbearbeitung, 20 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	Die Studierenden sind mit weiterführenden Lichtmodellen vertraut, können vertex- und fragmentbasierte Shader programmieren, kennen die Vor- und Nachteile verschiedener globaler photorealistischer Verfahren, können nichtphotorealistische Verfahren sinnvoll einsetzen und mit Splines als Linien und Flächen umgehen. Das sichere Beherrschen der Programmierung von Grafikkarten (Graphik-Hardware GPU) mit der Programmiersprache *C for graphics* und anderen *Application Programming Interfaces* (API) für globales Rendering wird unter Verwendung von numerischen Algorithmen zur Lösung nichtlinearer Gleichungssysteme vermittelt.	
<b>Inhalt</b>	Shading und Bidirectional Reflectance Distribution Function (BRDF), C for graphics (Cg), Radiosity, Photon Mapping, Nichtphotorealismus (NPR), Splines	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Computergraphik 1 (ICG1)	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung	
<b>Nützliche Literatur</b>	J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Hughes und R. L. Phillips: Computer Graphics: Principles and Practice. Addison-Wesley, 1996 Randima Fernando, Mark J. Kilgard: The Cg Tutorial. Addison-Wesley, 2003	

## Computerspiele

<b>Code</b> ICS	<b>Name</b> Computerspiele	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> Vorlesung 3 SWS, Übung 3 SWS	<b>Arbeitsaufwand</b> 240 h; davon 75 h Präsenzstudium 15 h Prüfungsvorbereitung 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Die Studierenden lernen die Konzepte von der informatischen Sicht kennen, was Computerspiele ausmacht und welche Herausforderungen damit verbunden sind. Sie lernen, wie man gute und effiziente Architekturkonzepte dafür entwickelt sowie wie man typische Probleme aus Graphik, Kollisionserkennung, Animation/Physik, Pfadplanung/KI umsetzt. Zudem lernen sie in den Übungen, wie man konkrete Spiele entwickelt, so dass sie in der Lage sind, eigene Spieleengines zu realisieren.	
<b>Inhalt</b>	Überblick über die Einteilung von Computerspielen Architektur von Game Engines Vorstellung von OGRE als einer open-source Game Engine Graphik und Computerspiele: ein Überblick Kollisionserkennungstechniken Animationstechniken und Physik bei Computerspielen mit Fokus auf der open source Bibliothek Bullet Pfadplanung und KI	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Pruefungs-modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung	
<b>Nuetzliche Literatur</b>	Gregory et al: Game Engine Architecture Ericson: Real-Time Collision Detection Eberly: Game Physics Millington: Artificial Intelligence for Games	

## Cybersicherheit und Internet Governance

<b>Code</b> ICIG	<b>Name</b> Cybersicherheit und Internet Governance	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> 1 Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 120 h; davon 30 h Präsenzstudium 60 h Vor- und Nachbereitung 30 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik

<p><b>Lernziel</b></p>	<p>Studierende</p> <ul style="list-style-type: none"> <li>* erwerben umfangreiches Wissen über die Funktionsweise und Verwundbarkeiten vernetzter Computersysteme, insbesondere der Internettechnologie und können somit Konzepte zur IT-Sicherheit bewerten und entwerfen</li> <li>* erlangen grundlegende Kenntnisse über die Verwaltung der kritischen Ressourcen des Internets (IP-Adressraum, DNS, Protokolle und Standards) und Entscheidungsfindungsmechanismen internationaler Internet Governance und können die Auswirkung von international festgelegten Vorschriften auf die Umsetzung von IT-Sicherheitskonzepten berücksichtigen</li> <li>* erwerben vertiefte Kenntnisse im Bereich Cybersicherheit; Studium und Nachvollzug typischer Cyberangriffsforme; umfassender Kompetenzerwerb zur Detektion von Cyberangriffen; grundlegender Kompetenzen im Feld der IT-/Cyber-Forensik und sind dadurch in der Lage aktive Maßnahmen im Sinne von Intrusion Detection Systemen für eine Einrichtung zu planen und rechtskonform umzusetzen</li> <li>* erhalten einen detaillierten Überblick über Felder gesellschaftspolitischer Regulierung (Netzneutralität, Datenschutz, Cybersicherheit), einschließlich normativ-rechtlicher Gestaltungsansprüche, konkreter politischer Regulierung (national, EU, international) sowie technischer Voraussetzungen und Umsetzungsmöglichkeiten (Sicherheits-Engineering) und können die Praktikabilität der jeweiligen Konzepte sowohl aus Sicht des gesellschaftlichen Diskurses als auch aus der Perspektive der Technik bewerten</li> </ul> <p>Möglichkeit zur Teilnahme an Aufbaukursen sowohl im technischen/computerwissenschaftlichen (als auch im politik-/sozialwissenschaftlichen Feld)</p> <p>Langfristiges Ausbildungsziel: Einsatz-/Beschäftigungsfähigkeit in der Breite des Arbeitsfeldes IT-Cybersicherheit; durch ganzheitlichen Ansatz der Ausbildung nicht allein in technischen Beschäftigungsfeldern, sondern auch in der Beratung von Unternehmen, Behörden und anderen Einrichtungen</p>
------------------------	---

<b>Inhalt</b>	<p>Die Digitalisierung und vernetzte Computersysteme prägen gesellschaftliche Austauschprozesse auf nationaler wie internationaler Ebene. Sie haben gravierende Effekte auf das alltägliche Leben der Nutzer, die Wirtschafts- und Arbeitswelt, den internationalen Handel und die Weltpolitik. Gerade die vergangenen Jahre haben gezeigt, wie verwundbar vernetzte Computersysteme sind. Cybersicherheit ist von einer technischen zu einer gesellschaftspolitischen Herausforderung ersten Grades geworden. Um ihr gerecht zu werden, ist es von großer Bedeutung, vernetzte Computersysteme und insbesondere das Internet nicht auf die technischen Aspekte zu reduzieren, sondern das Ineinandergreifen von Technologie auf der einen Seite, normativen und politischen Regulierungsansprüchen sowie vielfältigen Governance-Strukturen und -Prozessen auf der anderen Seite zu begreifen. Hierfür bietet die Vorlesung den teilnehmenden Studierenden einen interdisziplinären Ausgangspunkt, denn sie leistet einen Brückenschlag zwischen der Informatik und der Politischen Wissenschaft.</p> <p>Themen sind u.a.:</p> <ul style="list-style-type: none"> <li>* Angriffsszenarien und Verwundbarkeiten vernetzter Computersysteme;</li> <li>* Internet(un)sicherheit;</li> <li>* Sicherheits-Engineering;</li> <li>* Verwaltung kritischer Internetressourcen;</li> <li>* Netzneutralität;</li> <li>* Content-Regulierung und Zensur;</li> <li>* Cybersicherheit als Gegenstand politischer und rechtlicher Regulierung;</li> </ul>
<b>Voraussetzungen</b>	
<b>Pruefungs-modalitaeten</b>	regelmäßige Teilnahme (überprüft), Abschlussklausur (90 Min.)

<p><b>Nuetzliche Literatur</b></p>	<p>DeNardis, L. (2014). The global war for internet governance. New Haven/London: Yale University Press.</p> <p>Dunn Cavelty, Myriam (2011): Cyber-threats, in: Dunn Cavelty, Myriam/Mauer, Victor (Hg.): The Routledge handbook of security studies. Milton Park [et al.]: Routledge (Routledge handbooks), 180-189.</p> <p>Eckert, C. (2014). IT-Sicherheit: Konzepte, Verfahren, Protokolle (9. Aufl). De Gruyter Studium. Oldenbourg: de Gruyter.</p> <p>Harich, T. (2015). IT-Sicherheit im Unternehmen, MITP Taschenbuch</p> <p>Knake, Robert K. (2010): Internet governance in an age of cyber insecurity. New York, NY: Council on Foreign Relations (Council Special Report).</p> <p>Lange, H.-J., und Bötticher, A. Cyber-Sicherheit. Studien zur Inneren Sicherheit: Band 18.</p> <p>Nissenbaum, Helen (2005): Where Computer Security Meets National Security, in: Ethics and Information Technology 7: 2, 61-73.</p> <p>Weitere Literatur wird in der Vorlesung bekannt gegeben.</p>
------------------------------------	---

## Data Warehouses

<b>Code</b> IDW	<b>Name</b> Data Warehouses	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes 4. Semester
<b>Lehrform</b> Vorlesung 2 SWS, Übungen 1 SWS	<b>Arbeitsaufwand</b> 120 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 75 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	<p>Die Studierenden sind in der Lage, zu einer gegebenen Anwendung eine Data Warehouse Pipeline zu modellieren</p> <p>kennen die Komponenten der Architektur von Data Warehouse Systemen</p> <p>wissen, welche Arten von Indexe und Speicherstrukturen typische Data Warehouse Anfragen unterstützen</p> <p>kennen Verfahren der Anfragebearbeitung und (manueller) Optimierungstechniken</p> <p>wissen, wie materialisierte Sichten einzusetzen sind und wie diese zu pflegen sind</p> <p>sind mit dem Aufsetzen eines realen Data Warehouses und den administrativen Tasks vertraut</p> <p>kennen die wichtigsten Performance Benchmarks für Data Warehouse Anwendungen</p>	
<b>Inhalt</b>	<p>Einführung &amp; Grundbegriffe</p> <p>Data-Warehouse-Architekturen</p> <p>Modellierung von Data Warehouses</p> <p>Index- und Speicherungsstrukturen</p> <p>Anfragen an Data Warehouses</p> <p>Anfrageverarbeitung und -optimierung</p> <p>Materialisierte Sichten</p>	
<b>Voraussetzungen</b>	empfohlen sind: Algorithmen und Datenstrukturen (IAD), Datenbanken 1 (DB1)	
<b>Pruefungs-modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung	



<b>Nuetzliche Literatur</b>	<p>z. B.: Wolfgang. Lehner. Datenbanktechnologie für Data-Warehouse-Systeme. dpunkt.verlag, Heidelberg, 2003.</p> <p>Andreas Bauer, Holger Günzel. Data Warehouse Systeme ? Architektur, Entwicklung, Anwendung. 3. Auflage, dpunkt.verlag, Heidelberg, 2008.</p> <p>Gunter Saake, Andreas Heuer, Kai-Uwe Sattler. Datenbanken: Implementierungstechniken. 2. Auflage, mitp-Verlag, Bonn, 2005.</p>
---------------------------------	---

## Deep Vision

<b>Code</b> IDV	<b>Name</b> Deep Vision	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 105 h Selbststudium und Bearbeitung von theoretischen Aufgaben und Programmierübungen 15 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Die Studierenden haben gelernt Algorithmen zu entwickeln, die effizient aus großen Bildmengen robuste Objektmodelle lernen, kennen die Vorzüge/Einschränkungen von Deep Learning Ansätzen im Bereich des maschinellen Sehens, können Inferenz über Objekte in 3D durchführen, können die erworbenen Kenntnisse auf Anwendungsbereiche, wie z.B. die Erkennung von Objekten in 3D Szenen oder biomedizinischen Tomographien übertragen	
<b>Inhalt</b>	High-level Computer Vision 3D Vision and beyond Deep Learning Multi-view Approaches Applications	
<b>Voraussetzungen</b>	keine	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen und Bestehen einer Abschlussprüfung	
<b>Nützliche Literatur</b>		

## Effiziente Algorithmen 1

<b>Code</b> IEA1	<b>Name</b> Effiziente Algorithmen 1	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 4. Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übungen 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Die Studierenden verstehen die grundlegenden Konzepte der Graphentheorie können Fragestellungen als kombinatorische Optimierungsprobleme modellieren können die Komplexität von Optimierungsproblemen analysieren kennen die Methoden zum Beweis der Korrektheit kombinatorischer Algorithmen und der Analyse ihrer Laufzeit kennen die grundlegenden algorithmischen Ansätze sind mit den Fragen der effizienten Implementierung vertraut haben Einblick in die vielfältigen Anwendungsgebiete der kombinatorischen Optimierung	
<b>Inhalt</b>	Grundbegriffe der Graphentheorie Grundlegende Graphenalgorithmien Optimale Bäume und Branchings Kürzeste Wege Das Zuordnungsproblem Maximale Flüsse Minimale Schnitte in ungerichteten Graphen Flüsse mit minimalen Kosten Matchingprobleme	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Lineare Algebra 1	
<b>Prüfungs- modalitäten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung	

<b>Nuetzliche Literatur</b>	Korte, B., Vygen, J.: Kombinatorische Optimierung, Springer, 2007 Cook, W.J., Cunningham, W.H., Pulleyblank, W.R., Schrijver, A.: Combinatorial Optimization, Wiley, 1997
---------------------------------	--

## Effiziente Algorithmen 2

<b>Code</b> IEA2	<b>Name</b> Effiziente Algorithmen 2	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 4. Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übungen 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden können polynomiale und NP-schwere Optimierungsprobleme klassifizieren</p> <p>kennen die gesamte Bandbreite von Lösungsansätzen für schwierige Probleme</p> <p>können geeignete Modellierungen für Anwendungsprobleme selbst erstellen</p> <p>sind in der Lage, geeignete Lösungsverfahren auszuwählen</p> <p>können selbst Lösungsverfahren konzipieren und implementieren</p>	
<b>Inhalt</b>	<p>NP-schwere Optimierungsprobleme</p> <p>approximative Algorithmen und Heuristiken (Bin-Packing, Scheduling, Knapsack, Traveling Salesman)</p> <p>Relaxierungen (lineare, kombinatorische, semidefinite, Lagrange-Relaxierungen)</p> <p>Verfahren zur Bestimmung optimaler Lösungen (dynamische Optimierung, Branch-and-Bound, Branch-and-Cut)</p> <p>lineare 0/1- Optimierung (Modellierung, Schnittebenen)</p> <p>polyedrische Kombinatorik,</p> <p>Spaltengenerierung und Dekomposition</p> <p>Fallstudien (Traveling-Salesman_Problem, Max-Cut-Problem.)</p> <p>Es wird somit das gesamte Spektrum von Lösungsverfahren für schwierige kombinatorische Probleme vermittelt.</p>	
<b>Voraussetzungen</b>	<p>empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Lineare Algebra 1; Absolvierung des Moduls Effiziente Algorithmen 1 (IEA1) ist nützlich, aber nicht Voraussetzung</p>	
<b>Prüfungsmodalitäten</b>	<p>Erfolgreiche Teilnahme an den Übungen (mehr als 50 Prozent der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung</p>	

<b>Nuetzliche Literatur</b>	Korte, B., Vygen, J.: Kombinatorische Optimierung, Springer, 2007 Nemhauser, G.L., Wolsey, L.A.: Integer Combinatorial Optimization, Wiley, 1988
---------------------------------	---

## Einführung in das maschinelle Sehen in 3D

<b>Code</b> I3DCV	<b>Name</b> Einführung in das maschinelle Sehen in 3D	
<b>Leistungspunkte</b> 2 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Wintersemester
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 60 h; davon 30 h Präsenzstudium, 30 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Die Studierenden sind in der Lage 3D Erfassungs- und Auswertemethoden in praktischen, industriellen und geisteswissenschaftlichen Anwendungen einzusetzen. Sie kennen die Gesetze und Eigenschaften von optischen Linsen, grundlegende Bildfilter, interne und externe Kameraparameter, Stereorektifizierung bzw. Structure from Motion (SfM), Bündelausgleich, Berechnung von Tiefendaten, Datenformate für unstrukturierte Gitter, numerische Krümmungsberechnung auf diskreten Mannigfaltigkeiten. Sie können den Einsatz von sowohl strukturierten Licht als auch Time-of-flight Scannern planen.	
<b>Inhalt</b>	Bildaufnahme Kamerakalibrierung und Merkmalsextraktion Active Range Scanner(s) Shape from Monocular Images Shape from Multiple Images Time-of-flight-scanner (TOF) 3D Objektrepräsentationen 3D Anwendungen	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK)	
<b>Prüfungsmodalitäten</b>	Teilnahme an den Vorlesungen und Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Teilnehmerzahl)	
<b>Nützliche Literatur</b>	J. Bernd, Digital Image Processing., Springer, 6 edition, 2005. E. R. Davies, Machine Vision ? Theory, Algorithms, Practicalities, Elsevier, 2005.	

## Formale Sprachen und Automatentheorie

<b>Code</b> IFSA	<b>Name</b> Formale Sprachen und Automatentheorie	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mindst. jedes 4. Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übungen 2 SWS mit Hausaufgaben.	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 60 h Prüfungsvorbereitung (und Prüfung) 90 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik



<b>Lernziel</b>	<p>Die Studierenden kennen die Stufen der Chomsky-Hierarchie und deren äquivalente Beschreibung durch Grammatiken und Automatenmodelle, kennen die Charakterisierungen der Klasse der regulären Sprachen durch rechts- bzw. linkslineare Grammatiken, durch deterministische und nichtdeterministische endliche Automaten, durch reguläre Ausdrücke und durch Rechtskongruenzen mit endlichem Index, sind in der Lage, das Pumping-Lemma für reguläre Sprachen zu beweisen und dieses anzuwenden, um zu zeigen, dass bestimmte Sprachen nicht regulär sind, kennen den Nachweis der Eindeutigkeit des Minimalautomaten und können diesen in Form eines reduzierten Automaten oder eines Äquivalenzklassenautomaten konstruieren, können das Pumping-Lemma für kontextfreie Sprachen beweisen und dieses anwenden, um zu zeigen, dass bestimmte Sprachen nicht kontextfrei sind, können die Klasse der kontextfreien Sprachen durch kontextfreie Grammatiken und durch Kellerautomaten beschreiben und die Äquivalenz dieser Darstellungen beweisen, kennen den Satz von Chomsky und Schützenberger und seine anschauliche Bedeutung für die Struktur der Ableitungen in kontextfreien Sprachen, sind mit Normalformen, Abschlusseigenschaften, Entscheidbarkeits- und Komplexitätsaspekten insbesondere der regulären und kontextfreien Sprachen vertraut, können die Klasse der kontextsensitiven Sprachen durch kontextsensitive Grammatiken und Grammatiken vom Erweiterungstyp sowie durch linear beschränkte Automaten beschreiben und die Äquivalenz dieser Darstellungen beweisen.</p>
<b>Inhalt</b>	<p>Die Vorlesung gibt eine Einführung in die Theorie der formalen Sprachen und die dort verwendeten Begriffe und Methoden zur syntaktischen Sprachbeschreibung und -analyse unter besonderer Berücksichtigung algorithmischer Aspekte. Der Schwerpunkt der Vorlesung liegt auf den Sprachklassen der Chomsky-Hierarchie und deren Charakterisierung hinsichtlich der Beschreibbarkeit durch Grammatiken und der Erkennbarkeit durch Automaten.</p>
<b>Voraussetzungen</b>	<p>empfohlen ist: Einführung in die Theoretische Informatik (ITH)</p>
<b>Prüfungsmodalitäten</b>	<p>Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer mündlichen Abschlussprüfung.</p>

<b>Nuetzliche Literatur</b>	
---------------------------------	--

## Funddokumentation mittels optischer 3D-Scanner

<b>Code</b> I3DOK	<b>Name</b> Funddokumentation mittels optischer 3D-Scanner	
<b>Leistungspunkte</b> 2 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Übung 2 SWS	<b>Arbeitsaufwand</b> 60 h; davon 30 h Präsenzstudium 30 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Die Studierenden sind mit unterschiedlichen 3D-Nahbereichsscantechniken vertraut, können 3D-Modelle aufbereiten und kennen die Herangehensweise mit 3D Bildverarbeitung zur Erkennung von Merkmalen (Schrift).	
<b>Inhalt</b>	3D-Meßtechnik basierend auf dem Prinzip des Strukturierten Licht, Aufnahme und Verarbeitung von hochauflösenden 3D-Modellen *3D-image-processing pipeline*	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI) oder Computergraphik 1 (ICG1) oder Visualisierung im Bereich Cultural Heritage (IVCH) oder Vermessungskunde (UFG)	
<b>Pruefungs-modalitaeten</b>	Teilnahme an den einleitenden Vorlesungen und den praktischen Übungen. Bestehen einer praktischen Abschlussprüfung.	
<b>Nuetzliche Literatur</b>	Clive Orton: Mathematics in Archaeology. Cambridge, MA, Cambridge University Press, 1982 Katsushi Ikeuchi, Daisuke Miyazaki (editors): Digitally Archiving Cultural Objects. Springer, 2007 Christian Hörr. Algorithmen zur automatisierten Dokumentation und Klassifikation archäologischer Gefäße. Dissertation, TU Chemnitz, 2011, Hubert Mara, Multi-Scale Integral Invariants for Robust Character Extraction from Irregular Polygon Mesh Data, Dissertation, Uni Heidelberg 2012	

## Gaussian Processes for Machine Learning

<b>Code</b> IGPML	<b>Name</b> Gaussian Processes for Machine Learning	
<b>Leistungspunkte</b> 5 LP	<b>Dauer</b> one semester	<b>Turnus</b> irregular
<b>Lehrform</b> Lecture 2 SWS	<b>Arbeitsaufwand</b> 150 h; thereof 30 h lecture 100 h project 20 h report	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	<p>To build a solid background on both the theory of Gaussian processes (GPs) and how they are used in practice to build effective machine learning models.</p> <ul style="list-style-type: none"> <li>* Firm theoretical knowledge on how to use GPs for machine learning.</li> <li>* Knowledge on how big data can be modeled with GPs.</li> <li>* Practice on how to design, develop, and evaluate a powerful machine learning model.</li> </ul>	
<b>Inhalt</b>	<p>This module covers the following topics:</p> <ul style="list-style-type: none"> <li>* Introduction to and motivation for GPs.</li> <li>* Predicting real-valued and categorical output with GPs.</li> <li>* Approximate inference of GPs.</li> <li>* Modeling big data with GPs.</li> <li>* Exploratory data analysis and knowledge discovery with GPs.</li> <li>* Time series modeling with GPs.</li> <li>* Deep learning with GPs.</li> <li>* GPs for alternative learning setups.</li> </ul>	
<b>Voraussetzungen</b>	recommended are: basic background on probability and statistics, basic knowledge on machine learning and linear algebra	
<b>Pruefungsmodalitaeten</b>	<p>Successfully handing in the source code of a working GP-based machine learning model implementation and the related report (approximately four pages, single column). The students are free to choose the learning setup to study with GPs and the data set to evaluate the learning model on.</p>	

<b>Nuetzliche Literatur</b>	Carl E. Rasmussen, Christopher I. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006 (online) Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2007
---------------------------------	--

## Gemischt-ganzzahlige Programmierung und kombinatorische Optimierung

<b>Code</b> IMIP	<b>Name</b> Gemischt-ganzzahlige Programmierung und kombinatorische Optimierung	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, praktischer Teil (Übungen) 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 20 h Prüfungsvorbereitung 100 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>kennen die Modellierungsmöglichkeiten der gemischt-ganzzahligen Optimierung</li> <li>sind mit einer kommerziellen MIP-Software vertraut</li> <li>haben einen guten Überblick über die Anwendungsmöglichkeiten</li> <li>sind mit den polyedertheoretischen Grundlagen der Optimierung vertraut</li> <li>kennen die wichtigsten Optimierungstechniken</li> <li>haben die erforderlichen Kenntnisse zur Beurteilung von Möglichkeiten und Grenzen der praktischen Verwendbarkeit der erlernten Techniken</li> <li>haben Expertise in der Modellierung diskreter und ganzzahliger Optimierungsprobleme und dem Einsatz kommerzieller Optimierungssoftware</li> </ul>	

<b>Inhalt</b>	<p>Lineare Programmierung (Polyedertheorie, Dualität, Simplex-Algorithmus, postoptimale Analyse)</p> <p>gemischt-ganzzahlige Modellierungen</p> <p>Kombinatorische Optimierungsprobleme</p> <p>Bestimmung von Optimallösungen</p> <p>Polyhedrische Kombinatorik</p> <p>Kombinatorische Polytope</p> <p>Relaxierungen (kombinatorische, LP-, semidefinite , Lagrange-Relaxierungen)</p> <p>Branch-and-Cut Algorithmen</p> <p>Zulässige Ungleichungen, Schnittebenen</p> <p>Presolve</p> <p>Techniken für große Probleme (Spaltengenerierung, Benders- und Dantzig-Wolfe-Dekomposition)</p> <p>Anwendungen</p>
<b>Voraussetzungen</b>	empfohlen sind: Grundkenntnisse Mathematik, Programmierkurs (IPK)
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
<b>Nützliche Literatur</b>	<p>z.B. Kallrath, J. and Wilson, J.M.: Business Optimisation using Mathematical Programming, Macmillan Press, 1997</p> <p>Williams, H.P.: Model Building, Wiley, 1999</p>

## Geometric Modeling and Animation

<b>Code</b> IGMA	<b>Name</b> Geometric Modeling and Animation	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> at least every 4th semester
<b>Lehrform</b> Lecture 4 SWS, Exercise 2 SWS	<b>Arbeitsaufwand</b> 240 h; thereof 90 h on-campus program 15 h exam preparation 135 h independent study and exercises (possibly in groups)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	<p>The students</p> <p>know the mathematical foundations of geometric modeling</p> <p>know the mathematical and physical foundations of computer animation</p> <p>know the algorithms and implementation aspects</p> <p>are familiar with the basics of animated movies</p> <p>are able to apply existing tools for geometric modeling and animation</p>	
<b>Inhalt</b>	<p>Introduction to curves</p> <p>Interpolating curves</p> <p>Bézier curves</p> <p>B-Splines</p> <p>Rational curves</p> <p>Introduction to surfaces</p> <p>Tensor product surfaces</p> <p>Transfinite surfaces and extrusion</p> <p>Subdivision</p> <p>Subdivision surfaces</p> <p>Animation and simulation</p> <p>Rigid body kinematics</p> <p>Particle systems</p> <p>Mass-spring models</p> <p>Cloth modeling</p> <p>Numerical methods for differential equations</p> <p>Collision detection and handling</p> <p>Fluid simulation and natural phenomena</p>	
<b>Voraussetzungen</b>	recommended are: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Prüfungsmodalitäten</b>	Successful participation in the exercises (more than 50 % have to be scored) und passing a written or oral exam	



<b>Nuetzliche Literatur</b>	<ul style="list-style-type: none"> <li>- Curves and Surfaces for CAGD ? A Practical Guide, G. Farin, Morgan Kaufmann, 2002</li> <li>- Computer Animation ? Algorithms and Techniques, R. Parent, Morgan Kaufmann, 2002</li> <li>- 3D Game Engine Design: A Practical Approach to Real-Time Computer Graphics, D. Eberly, Morgan Kaufmann, 2000</li> <li>- Graphische Datenverarbeitung I, J. Encarnacao, W. Straßer, R. Klein, 4. Auflage, Oldenbourg 1996</li> <li>- Advanced Animation and Rendering Techniques, A. Watt, M. Watt, Addison-Wesley, 1992</li> <li>- Grundlagen der geometrischen Datenverarbeitung, J. Hoschek, D. Lasser, Teubner 1992</li> <li>- Numerical Recipes ? The Art of Scientific Computing, W.H. Press, P. Flannery, S.A. Teukolsky, W.T. Vetterling, Cambridge University Press, 1986</li> </ul>
---------------------------------	--

## Inverse Problems

<b>Code</b> IIP	<b>Name</b> Inverse Problems	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS und Hausarbeiten	<b>Arbeitsaufwand</b> 240 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 165 h Selbststudium und Aufgabenbearbeitung, Hausarbeiten	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	Die Studierenden verstehen, was inverse Probleme sind und warum sie schwer zu lösen sind. Sie lernen die Prinzipien kennen, wie man sowohl deterministische Probleme, als auch stochastische Probleme löst und dabei auch die Regularisierungsparameter geeignet wählt. Schließlich erfahren sie die neuesten Entwicklungen im Bereich compressed sensing. Alle Prinzipien werden an zwei ausgewählten Gebieten, der Tomographie und des Deblurrings dargestellt. Sie erhalten damit die Kompetenz komplexe Probleme zu lösen, die mit klassischen Techniken nicht stabil lösbar sind und sind damit in der Lage auch komplexe experimentelle Messungen adäquat auswerten zu können.	
<b>Inhalt</b>	Deterministische inverse Probleme Stochastische inverse Probleme Wahl der Regularisierungsparameter Compressed sensing Tomographie Deblurring	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD), Numerische Mathematik	
<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 60% der Punkte müssen erreicht werden) und Bestehen einer mündlichen Abschlussprüfung	
<b>Nuetzliche Literatur</b>	M. Bertero, P. Boccacci: Introduction to Inverse Problems in Imaging, IoP, 2002 web-Page and book: <a href="http://www.slaney.org/pct/pct-toc.html">http://www.slaney.org/pct/pct-toc.html</a>	

## IT-Projektmanagement

<b>Code</b> IPM	<b>Name</b> IT-Projektmanagement	
<b>Leistungspunkte</b> 3 LP als Spezialvorlesung Angewandte Informatik 6 LP als EPG	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Wintersemester
<b>Lehrform</b> Vorlesung+Übung 2 SWS für eine Anrechnung als EPG mit zusätzlichem Forschungsprojekt	<b>Arbeitsaufwand</b> Vorlesung + Übung 90 h insgesamt, davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)  für eine Anrechnung als EPG zusätzlich: 90 h Forschungsprojekt, davon 15 h Einarbeitung und Literaturrecherche 15 h Vorbereitung einer Studie, z.B. Interview-Training 20 h Durchführung einer Studie 20 h Auswertung 20 h Abschlusspräsentation + Berichterstellung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Die Teilnehmer/innen können ein Projekt planen und überwachen, verstehen, wie Projekte in Organisationen eingebettet sind und haben Grundkenntnisse in vertraglichen Themen. Die Teilnehmer/innen des EPG können ein Forschungsprojekt selbständig durchführen und kennen Forschungsergebnisse aus der Gender-Forschung.	

<b>Inhalt</b>	Projektplanung, Projektorganisation Kostenschätzung Angebot/ Vertrag, Verhandeln Vorgehensmodelle Risikomanagement Controlling IT-Vertragsrecht Änderungsmanagement Zeitmanagement Projektabschluss Verteilte Softwareentwicklung
<b>Voraussetzungen</b>	keine
<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Hausaufgaben (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer Abschlussprüfung. Für EPG: Zusätzlich erfolgreiche Teilnahme am Forschungsprojekt. Die Note für das Forschungsprojekt geht zur Hälfte in die Endnote ein.
<b>Nuetzliche Literatur</b>	PMI (Project Management Institute): A Guide to the Project Management Body of Knowledge (PM BOK ® Guide), 4. Ausgabe 2008

## Knowledge Discovery in Databases

<b>Code</b> IKDD	<b>Name</b> Knowledge Discovery in Databases	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Wintersemester
<b>Lehrform</b> Vorlesung 4 SWS, Übungen 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	<p>Die Studierenden sind in der Lage, die Daten zu einem Data Mining Verfahren mit Hilfe geeigneter (statistischer) Methoden vorzuverarbeiten</p> <p>kennen die grundlegenden Verfahren der statistischen Datenanalyse</p> <p>sind in der Lage, grundlegende Techniken des Data Mining auf Daten aus verschiedenen Anwendungsbereichen anzuwenden</p> <p>sind in der Lage, Data Mining Techniken im Kontext von Datenbanken und des KDD Prozesses zu realisieren und anzuwenden</p> <p>kennen die Techniken und Prinzipien, die den Algorithmen zur Klassifikation und dem Clustering von Daten zugrunde liegen</p> <p>wissen, wie welche Clustering Algorithmen im Kontext welcher Daten und Anwendungen zu verwenden sind</p> <p>kennen die Verfahren zum Finden häufiger Muster in Daten.</p> <p>kennen die wichtigsten Techniken zur Erkennung von Ausreißern</p> <p>sind vertraut mit den grundlegenden Algorithmen und Techniken zur Analyse von zeit- und raumbezogenen Daten</p>	
<b>Inhalt</b>	<p>Grundbegriffe: Statistik und Daten</p> <p>Datenaufbereitung</p> <p>Clustering</p> <p>Frequent Pattern Mining</p> <p>Klassifikationsverfahren</p> <p>Mining von Graphen</p> <p>Mining von räumlich und zeitlich veränderlichen Daten (z.B. Objekt-Trajektorien und Zeitreihen)</p>	
<b>Voraussetzungen</b>	empfohlen sind: Algorithmen und Datenstrukturen (IAD), Datenbanken 1 (DB1)	

<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung
<b>Nuetzliche Literatur</b>	z. B.: Jiawei Han und Micheline Kamber: Data Mining. Concepts and Techniques, Morgan Kaufmann Series in Data Management Systems, 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.

## Künstliche Intelligenz

<b>Code</b> IKI	<b>Name</b> Künstliche Intelligenz	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60h Präsenzstudium 15h Prüfungsvorbereitung 105h Selbststudium und Bearbeitung von theoretischen Aufgaben und Programmierübungen	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	Die Studierenden haben gelernt Algorithmen zu entwickeln die Probleme durch Suchen lösen können logische Inferenz zur Lösungssuche einsetzen sind in der Lage Inferenz im Fall von unsicherem Wissen über die Welt durchzuführen können lernende Agenten entwickeln können die erworbenen Kenntnisse auf Anwendungsbereiche, wie z.B. Computersehen oder Textmining, übertragen	
<b>Inhalt</b>	Problemlösen durch Suche KI und Spiele logische Inferenz Knowledge bases Aktionsplanung Schlussfolgerung unter Unsicherheit Maschinelles Lernen Neuronale Netze und Deep Learning Anwendungen	
<b>Voraussetzungen</b>	keine	
<b>Prüfungs- modalitäten</b>	Erfolgreiche Teilnahme an den Übungen und Bestehen einer Abschlussprüfung	
<b>Nützliche Literatur</b>	Russell & Norvig: Artificial Intelligence ? A Modern Approach	

## Machine Learning

<b>Code</b> IML	<b>Name</b> Machine Learning	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester (cannot be combined with *Fundamentals of Machine Learning* or *Advanced Machine Learning*)	<b>Turnus</b> in (irregular) alternation with *Fundamentals of Machine Learning* + *Advanced Machine Learning*
<b>Lehrform</b> 4 SWS lecture (in English), 2 SWS tutorial, homework assignments	<b>Arbeitsaufwand</b> 240h; thereof 90h lectures and tutorials 120h lecture wrap-up and homework 30h preparation for examination	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing, M.Sc. Physik (specialization Computational Physics)
<b>Lernziel</b>	Students understand a broad range of machine learning concepts, get to know established and advanced learning methods and algorithms, are able to apply them to real-world problems, and can objectively assess the quality of the results. In addition, students learn how to use Python-based machine learning software such as scikit-learn.	
<b>Inhalt</b>	This lecture is a compact version of the two-semester course *Fundamentals of Machine Learning* + *Advanced Machine Learning*: Classification (linear and quadratic discriminant analysis, neural networks, linear and kernelized support vector machines, decision trees and random forests), least squares and regularized regression, Gaussian processes, unsupervised learning (density estimation, cluster analysis, Gaussian mixture models and expectation maximization, principal component analysis, bilinear decompositions), directed probabilistic graphical models, optimization for machine learning, structured learning	
<b>Voraussetzungen</b>	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
<b>Pruefungsmodalitaeten</b>	successful homework solutions (at least 50% of total achievable points) and oral examination	
<b>Nuetzliche Literatur</b>	Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning (2nd edition), Springer, 2009; David Barber: Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012	



## Fundamentals of Machine Learning

<b>Code</b> IFML	<b>Name</b> Fundamentals of Machine Learning	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester (cannot be combined with *Machine Learning*)	<b>Turnus</b> in (irregular) alternation with *Machine Learning*
<b>Lehrform</b> 4 SWS lecture (in English), 2 SWS tutorial, homework assignments	<b>Arbeitsaufwand</b> 240h; thereof 90h lectures and tutorials 90h lecture wrap-up and homework 60h graded final report	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing, M.Sc. Physik (specialization Computational Physics)
<b>Lernziel</b>	Students understand fundamental concepts of machine learning (features vs. response, unsupervised vs. supervised training, regression vs. classification etc.), get to know established learning methods and algorithms, are able to apply them to real-world problems, and can objectively assess the quality of the results. In addition, students learn how to use Python-based machine learning software such as scikit-learn.	
<b>Inhalt</b>	The lecture, along with its sibling *Advanced Machine Learning*, offers an extended version of the one-semester course *Machine Learning*, with more room for regression methods, unsupervised learning and algorithmic details: Classification (nearest neighbor rules, linear and quadratic discriminant analysis, logistic regression, classical and randomized decision trees, support vector machines, ensemble methods); regression (linear and non-linear least squares, regularized and sparse regression, robust regression); unsupervised learning (hierarchical clustering, k-means algorithm, Gaussian mixture models and expectation maximization, principal component analysis, non-linear dimension reduction); evaluation (risk minimization, model selection, cross-validation)	
<b>Voraussetzungen</b>	recommended are: solid knowledge of basic calculus, statistics, and linear algebra	
<b>Pruefungs-modalitaeten</b>	successful homework solutions (at least 50% of total achievable points) and written examination (report on a 60h mini-research project)	
<b>Nuetzliche Literatur</b>	Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning (2nd edition), Springer, 2009	

## Advanced Machine Learning

<b>Code</b> IAML	<b>Name</b> Advanced Machine Learning	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester (cannot be combined with *Machine Learning*)	<b>Turnus</b> follows *Fundamentals of Machine Learning*
<b>Lehrform</b> 4 SWS lecture (in English), 2 SWS tutorial, homework assignments	<b>Arbeitsaufwand</b> 240h; thereof 90h lectures and tutorials 90h lecture wrap-up and homework 60h graded final report	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing, M.Sc. Physik (specialization Computational Physics)
<b>Lernziel</b>	Students get to know advanced machine learning methods that define the state-of-the-art and major research directions in the field. Students understand when these methods are called for, what limitations of standard solutions they address, and how they are applied to real-world problems. In addition, students learn how to use Python-based machine learning software such as scikit-learn, theano and OpenGM.	
<b>Inhalt</b>	The lecture, along with its sibling *Fundamentals of Machine Learning*, offers an extended version of the one-semester course *Machine Learning*: Multi-layered architectures (neural networks, deep learning); directed and undirected probabilistic graphical models (Gaussian processes, latent variable models, Markov random fields, structured learning); feature optimization (feature selection and learning, dictionary learning, kernel approximation, randomization); weak supervision (one-class learning, multiple instance learning, active learning, reinforcement learning)	
<b>Voraussetzungen</b>	recommended are: lecture *Fundamentals of Machine Learning* or similar	
<b>Pruefungsmodalitaeten</b>	successful homework solutions (at least 50% of total achievable points) and written examination (report on a 60h mini-research project)	
<b>Nuetzliche Literatur</b>	David Barber: Bayesian Reasoning and Machine Learning, Cambridge University Press, 2012 Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, 2006	

## Management und Analyse von Datenströmen

<b>Code</b> IMADS	<b>Name</b> Management und Analyse von Datenströmen	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. alle 2 Jahre
<b>Lehrform</b> Vorlesung 2 SWS, Übungen 1 SWS	<b>Arbeitsaufwand</b> 120 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 75 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	Dieses Modul vertieft die Grundkenntnisse und -fähigkeiten der Datenanalyse insbesondere im Umgang mit Datenströmen. Die Studierenden kennen die Herausforderungen der Datenstromverarbeitung und sind in der Lage, gängige Methoden anzuwenden, um Datenströme zu verwalten und auszuwerten. wissen, welche Mining Techniken für Datenströme im Kontext von verschiedenen Domänen anwendbar sind kennen die unter Inhalt angegebenen Methoden und Technologien	
<b>Inhalt</b>	Methoden des Managements und der Analyse von Datenströmen. Datenstrommodelle Datenstrommanagementsysteme Anfragesprachen und Anfrageverarbeitung Synopsenbildung und Approximation Data Mining Techniken wie beispielsweise Clustering, Frequent Pattern Mining und Erkennung von Ausreißern Mining und Analyse von Graph-Strömen	
<b>Voraussetzungen</b>	empfohlen sind: Algorithmen und Datenstrukturen (IAD), Datenbanken 1 (IDB1)	
<b>Pruefungs- modalitaeten</b>	Erfolgreicher Abschluss eines Projekts und Bestehen einer Abschlussprüfung	
<b>Nuetzliche Literatur</b>	Wissenschaftliche Veröffentlichungen. Details werden in der Vorlesung und auf der Moodle-Webseite bekannt gegeben.	

## Modellierung und Simulation in den Neurowissenschaften

<b>Code</b> IMSN	<b>Name</b> Modellierung und Simulation in den Neurowissenschaften	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 25 h Prüfungsvorbereitung 95 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden kennen den makro-/mikroskopischen Aufbau eines Gehirns und verstehen die Prinzipien neuronaler Signalverarbeitungsprozesse. sind in der Lage die Eigenschaften individueller Neuronenmodelle mit Methoden aus dem Bereich der dynamischen Systeme zu analysieren.</p> <p>können mathematische Modelle von Einzelneuronen selbst entwerfen und numerische Verfahren zur Lösung bestimmen.</p> <p>haben die Fähigkeit die erforderlichen numerischen Methoden in einer Programmiersprache (C/C++ ) eigenständig umzusetzen.</p> <p>haben die Kenntnis die Simulationsergebnisse mathematisch-informatisch zu analysieren und zu interpretieren.</p>	

<b>Inhalt</b>	<p>Grundkenntnisse in Neuroanatomie und -physiologie          Bioelektrizität: Biophysikalische Grundlagen von Ionenströmen, Elektrophysiologie, Nernstpotential          Biomembranmodelle          Modellierung von Signalverarbeitungsprozessen im extrazellulären Raum          Integrate-and-Fire Modelle / Neuronenmodelle mit einem Kompartiment          Numerische Lösungsmethoden für Einkompartimentmodelle          Dynamische Systeme: Begriffsbildungen, Phasenraumanalyse, Stabilität und Hysterese, Bifurkationen, Spiking, Bursting          Numerische Lösungsverfahren für steife Systeme          Modellierung und Simulation der passiven Kabelgleichung          Diskretisierung der Kabelgleichung          Das Hodgkin-Huxley Modell (HH) der aktiven Signalverarbeitung            Methoden zur nichtlineare Kopplung von HH und Kabelgleichung            Simulation stochastischer Prozesse</p>
<b>Voraussetzungen</b>	empfohlen sind: Grundlagen in Numerik wie z.B. aus MA7
<b>Pruefungsmodalitaeten</b>	Erfolgreiche Teilnahme an den Übungen und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
<b>Nuetzliche Literatur</b>	C. Koch: Biophysics of Computation: Information Processing in Single Neurons. Oxford Univ. Press, 1999, ISBN 0-19-518199-9

## Modellierung, Optimierung und Regelung mechanischer Systeme

<b>Code</b> IMORMS	<b>Name</b> Modellierung, Optimierung und Regelung mechanischer Systeme	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes 4. Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 50 h Prüfungsvorbereitung 100 h Selbststudium	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden können die grundlegenden Prinzipien der Modellierung, Optimierung und Regelung dynamischer Prozesse - im speziellen von mechanischen Systemen - darstellen und erklären. Sie kennen die Grundlagen der nichtlinearen Optimierung und können verschiedene mathematische Lösungsstrategien vergleichen und beurteilen. Die Studierenden können komplexe mechanische Systeme wie menschliche Geh- und Rennbewegung mit den erarbeiteten Methoden analysieren, klassifizieren und auf spezifische Eigenschaften wie Stabilität hin untersuchen .</p> <p>Die Studierenden können Softwarepakete auf der Basis von Lua und C++ einsetzen und zur Modellierung, Simulation, Optimierung und Visualisierung mechanischer Systeme nutzen. Sie sind in der Lage, die auftretenden Optimalsteuerungsprobleme numerisch zu lösen und die Güte der erreichten Lösung zu evaluieren.</p>	

<b>Inhalt</b>	<p>Ziel dieser Lehrveranstaltung ist es, Studierenden der Mathematik, Informatik und Physik einen anwendungsorientierten Einstieg in die Modellierung, Optimierung und Regelung von mechanischen Systemen zu bieten. Der Fokus der Vorlesung liegt auf komplexen Mehrkörpersystemen aus Robotik und Biomechanik.</p> <p>Während in der Vorlesung die notwendigen theoretischen Grundlagen vermittelt werden, dienen die Computerübungen dazu, den Umgang mit Softwarewerkzeugen für die Modellierung, Visualisierung, Simulation und Optimale Steuerung anhand einfacher Beispiele auszuprobieren.</p> <p>Inhalt der Vorlesung:</p> <ul style="list-style-type: none"> <li>Dynamische Prozessmodellierung</li> <li>Mechanische Grundbegriffe, Kinematik, Dynamik</li> <li>Modellierung von Mehrkörpersystemen</li> <li>Bewegungssimulation</li> <li>Nichtlineare Optimierung</li> <li>Direkte Methoden der Optimalen Steuerung</li> <li>Grundlagen der Regelungstechnik</li> <li>Grundlagen der Systemdynamik</li> <li>Steuerung und Regelung von Bewegungen</li> <li>Modellierung menschlicher Geh- und Rennbewegungen</li> <li>Modellierung von Laufbewegungen humanoider und zweibeiniger Roboter</li> <li>Stabilität von Bewegungen</li> </ul> <p>Inhalt der Computerübungen:</p> <ul style="list-style-type: none"> <li>Simulation und Visualisierung mechanischer Systeme</li> <li>Modellierung der Dynamik von Mehrkörpersystemen mit der RBDL (Rigid Body Dynamics Library)</li> <li>Implementierung und Lösen von Optimalsteuerungsproblemen mit MUSCOD-II für verschiedene mechanische Beispielsysteme.</li> </ul>
<b>Voraussetzungen</b>	<p>Dringend empfohlen sind: Programmierkenntnisse in C/C++, Einführung in die Numerik</p> <p>Vorteilhaft sind: Algorithmische Optimierung 1, Numerik 1, Kenntnisse in Matlab/Octave</p>

<b>Pruefungs- modalitaeten</b>	Bestehen einer 2-stündigen benoteten Klausur am Ende des Semesters. Zulassungsvoraussetzung für die Klausur ist eine erfolgreiche Teilnahme an den Computerübungen, d.h. regelmäßige Anwesenheit (Feststellung liegt im Ermessen des Übungsleiters) und Lösung von Programmieraufgaben mit anschließender Vorführung und Erklärung im CIP-Pool
<b>Nuetzliche Literatur</b>	D. Greenwood: Principles of Dynamics, Prentice Hall, 1987 I .Newton: Principia, 1687 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 Control, Springer 2008 Spong, Hutchinson, Vidyasagar: Robot modeling and control, Wiley, 2005 Perry, Burnfield: Gait Analysis - Normal and pathological function, Slack Inc., 2010 M. Raibert: Legged Robots that Balance, MIT Press, 2000



## Objekterkennung und automatisches Bildverstehen

<b>Code</b> IOAB	<b>Name</b> Objekterkennung und automatisches Bildverstehen	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jährlich
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden sind auf dem neuesten Stand der Forschung im Bereich von Mid- und High-Level Computer Vision kennen die relevantesten Ansätze zur Merkmalsextraktion sind vertraut mit den wesentlichen Methoden zur robusten Objektrepräsentation haben essentielle Algorithmen aus dem Bereich der Mustererkennung und des maschinellen Lernens verstanden und können diese auf neue Probleme übertragen und anwenden sind in der Lage aktuelle Forschungsarbeiten aus dem Themenbereich der Vorlesung zu verstehen und in den Kontext der klassischen Arbeiten einzuordnen können Objekterkennungsprobleme in neuen Anwendungen analysieren und die jeweils geeignete Algorithmik dazu abrufen um darauf aufbauend neue Lösungsansätze zu entwickeln sind fähig Objekterkennungsverfahren differenziert zu bewerten und die Validität einer experimentellen Evaluation zu überprüfen</p>	

<b>Inhalt</b>	<p>Methoden und Modelle aus dem Bereich von Mid- und High-Level Computer Vision.</p> <p>Insbesondere:</p> <ul style="list-style-type: none"> <li>Objektdetektion und -klassifikation</li> <li>lokale und globale Merkmalsextraktion</li> <li>Modell-basierte Ansätze</li> <li>Ansichten-basierte Methoden</li> <li>generative/diskriminative Verfahren</li> <li>Registrierung</li> <li>Formanalyse</li> <li>Voting Methoden</li> <li>Hashing Verfahren</li> <li>Hierarchische Objektrepräsentationen</li> <li>Erkennung von menschlichen Aktionen</li> <li>Lernverfahren für Objektrepräsentationen</li> <li>Deep Learning</li> </ul>
<b>Voraussetzungen</b>	keine
<b>Pruefungs-modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen und Bestehen einer Abschlussprüfung.
<b>Nuetzliche Literatur</b>	

## Objektorientiertes Programmieren im Wissenschaftlichen Rechnen

<b>Code</b> IOPWR	<b>Name</b> Objektorientiertes Programmieren im Wissenschaftlichen Rechnen	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS am Rechner.	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 105 h Selbststudium und Aufgabenbearbeitung 15 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Die Studierenden beherrschen die Programmiersprache C++ sind in der Lage Performanz von unterschiedlichen Lösungen zu beurteilen beherrschen Template Programmieretechniken können die Standard Template Library einsetzen sind in der Lage die gelehrtten Konzepte an ausgewählten Problemen des Wissenschaftlichen Rechnens praktisch umzusetzen	
<b>Inhalt</b>	Dieses Modul vertieft die in der Grundvorlesung Einführung in die Praktische Informatik vermittelten Kenntnisse in objektorientierter Programmierung mit spezieller Ausrichtung auf das Wissenschaftliche Rechnen: Klassenkonzept Dynamische Speicherverwaltung Ausnahmebehandlung Ressourcenallokierung und Initialisierung Benutzung von const Template Metaprogrammierung, statischer vs. dynamischer Polymorphismus Traits Policies Standard Template Library	
<b>Voraussetzungen</b>	empfohlen sind Kenntnisse einer objektorientierten Programmiersprache	
<b>Prüfungs- modalitäten</b>	Erfolgreiche Teilnahme an den Übungen (Erreichen von 50% der Punkte) und Bestehen einer Abschlussprüfung.	
<b>Nützliche Literatur</b>		

## Parallel Data Processing and Analysis

<b>Code</b> IPDPA	<b>Name</b> Parallel Data Processing and Analysis	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes 4. Semester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 105 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>* kennen ausgewählte Ansätze und Programmierparadigmen der parallelen Datenverarbeitung</li> <li>* können Tools zur parallelen Datenverarbeitung (u.a. Apache Hadoop und Spark) anwenden</li> <li>* wissen die Anwendungsbereiche der Analyse großer Datenmengen</li> <li>* kennen Methoden der parallelen Vorverarbeitung von Daten</li> <li>* kennen Analyseverfahren wie Klassifikation, Regression, Clustering sowie von deren parallelen Implementierungen</li> <li>* wissen die theoretischen und praktische Aspekte der Skalierbarkeit der parallelen Datenverarbeitung</li> </ul>	
<b>Inhalt</b>	<p>Zu den Inhalten zählen:</p> <ul style="list-style-type: none"> <li>* Programmierparadigmen für parallel-verteilte Datenverarbeitung, insbes. Map-Reduce und Spark-Programmiermodell</li> <li>* Praktische Kenntnisse von Apache Hadoop, Pig, und Hive, sowie Spark und ggf. anderer Frameworks für parallel-verteilte Datenverarbeitung</li> <li>* Anwendungsbereiche der parallelen Datenanalyse u.a. Clustering, Recommendation, Suche nach ähnlichen Objekten, Mining von Datenströmen</li> <li>* Verfahren zur parallelen Vorverarbeitung der Daten</li> <li>* Grundlagen der Analysetechniken wie Klassifikation, Regression, Clustering und Evaluation der Ergebnisse</li> <li>* Parallele Algorithmen für die Datenanalyse und ihre Umsetzung</li> </ul> <p>* Theorie und Praxis der Skalierbarkeit, Tuning der Algorithmen und Frameworks</p>	

<b>Voraussetzungen</b>	empfohlen sind: Programmierkenntnisse (z.B. in Java) und elementare Wahrscheinlichkeitsrechnung; Modul Knowledge Discovery in Databases (IKDD) ist empfohlen aber nicht notwendig. Modul Big Data (IBD) wird als Ergänzung bzw. Erweiterung empfohlen
<b>Pruefungsmodalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
<b>Nuetzliche Literatur</b>	<ul style="list-style-type: none"> <li>* Jure Leskovec, Anand Rajaraman, Jeffrey D. Ullman: Mining of Massive Datasets, Cambridge University Press, Version 2.1 von 2014 (<a href="http://www.mmms.org/">http://www.mmms.org/</a>)</li> <li>* Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009 (<a href="http://statweb.stanford.edu/~tibs/ElemStatLearn/">http://statweb.stanford.edu/~tibs/ElemStatLearn/</a>)</li> <li>* Ron Bekkerman, Misha Bilenko, John Langford: Scaling Up Machine Learning, Cambridge University Press, 2012</li> <li>* Jiawei Han, Micheline Kamber, Jian Pei: Data Mining: Concepts and Techniques, Morgan Kaufmann, (third edition), 2012</li> <li>* Bücher aus dem O'Reilly Data Science Starter Kit, 2014 (<a href="http://shop.oreilly.com/category/get/data-science-kit.do">http://shop.oreilly.com/category/get/data-science-kit.do</a>)</li> </ul>

## Parallele Lösung großer Gleichungssysteme

<b>Code</b> IPLGG	<b>Name</b> Parallele Lösung großer Gleichungssysteme	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes 4. Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS am Rechner mit Diskussion der Übungsaufgaben.	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	Die Studierenden beherrschen die Diskretisierung elliptischer Differentialgleichungen mittels Finiter Elemente verstehen das abstrakte Konzept der Teilraumkorrekturverfahren können dies im Rahmen von überlappenden und nichtüberlappenden Gebietszerlegungsverfahren sowie Mehrgitterverfahren umsetzen beherrschen die Konvergenztheorie zu diesen Verfahren können ausgewählte Verfahren praktisch auf Parallelrechnern umsetzen und der Leistungsfähigkeit bewerten	
<b>Inhalt</b>	Grundlagen der Diskretisierung elliptischer partieller Differentialgleichungen, Teilraumkorrekturverfahren überlappende und nichtüberlappende Gebietszerlegungsverfahren mit Konvergenztheorie geometrische Mehrgitterverfahren mit Konvergenztheorie algebraische Mehrgitterverfahren	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse einer objektorientierten Programmiersprache, MA7, Kenntnisse wie sie das Modul IPHR oder ICC1 vermittelt, Vorkenntnisse in Numerik, insbesondere Differentialgleichungen (MH7).	
<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (Erreichen von 50% der Punkte) und Bestehen einer Abschlussprüfung.	
<b>Nuetzliche Literatur</b>		

## Paralleles Höchstleistungsrechnen

<b>Code</b> IPHR	<b>Name</b> Paralleles Höchstleistungsrechnen	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS am Rechner mit Diskussion der Übungsaufgaben.	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	<p>Die Studierenden</p> <p>kennen die grundlegenden Architekturen paralleler Höchstleistungsrechner</p> <p>kennen die grundlegenden Synchronisations-mechanismen in parallelen Systemen inklusive Performanzaspekten</p> <p>beherrschen die wichtigsten Programmierparadigmen für parallele Systeme</p> <p>sind in der Lage grundlegende Synchronisationsaufgaben zu lösen</p> <p>beherrschen Algorithmen der parallelen linearen Algebra</p> <p>können die Qualität paralleler Algorithmen und deren praktische Implementierung bewerten</p>	
<b>Inhalt</b>	<p>Rechner mit globalem Adressraum</p> <p>Cache-Kohärenz</p> <p>Rechner mit lokalem Adressraum und Nachrichtenaustausch</p> <p>kritischer Abschnitt, Bedingungsynchronisation Semaphore</p> <p>Posix Threads</p> <p>Programmierung von Grafikkarten</p> <p>Nachrichtenaustausch</p> <p>MPI</p> <p>Client-Server Architekturen, entfernter Prozeduraufruf</p> <p>Bewertung paralleler Algorithmen</p> <p>Lastverteilung</p> <p>Algorithmen für vollbesetzte Matrizen</p> <p>Lösung dünnbesetzter Gleichungssysteme</p> <p>Partikelmethode</p> <p>Paralleles Sortieren</p>	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse einer Programmiersprache (Programmieren und Softwaretechnik) und grundlegender Algorithmen (Algorithmen und Datenstrukturen (IAD)).	

<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (Erreichen von 50%) und Bestehen einer Abschlussprüfung.
<b>Nuetzliche Literatur</b>	



## Praktische Geometrie

<b>Code</b> IPG	<b>Name</b> Praktische Geometrie	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, Übung 1 SWS	<b>Arbeitsaufwand</b> 120 h; davon 45 h Präsenzstudium 60 h Aufgabenbearbeitung 15 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Verständnis grundlegender geometrischer Konzepte zur Datenanalyse sowie effektive Punktsuche und Weiterverarbeitung von Messdaten Souveräner Umgang mit Projektionen und Beschreibungen jenseits der dreidimensionalen Erfahrungswelt Berechnung geometrischer Invarianten, Distanzen, Krümmungen aus Messdaten, rekonstruierten und generierten Flächen	
<b>Inhalt</b>	Grundlegende Gebiete der Geometrie mit Relevanz in Computergraphik, Bildverarbeitung, Mustererkennung, Computer Vision und Geometrischem Modellieren (i) Analytische Geometrie: Operationen auf Vektorräumen mit geeigneten Koordinaten und Abbildungen (Affinitäten, Kollinearitäten), geometrische Ausgleichsprobleme aus fehlerbehafteten Messdaten (ii) Projektive Geometrie: Zentralprojektion und inverse Rekonstruktion von 3D-Objekten aus ebenen Bildern (Computer Vision, Geodäsie), Unterschiede zwischen B-Spline-Kurven und -Flächen und der Klasse der NURBS, Freiformgeometrien in CAD-Systemen (iii) Differentialgeometrie: Parameterdarstellungen in der geometrischen Datenverarbeitung, implizite Darstellungen (level sets), Abschätzung von Invarianten aus diskreten Daten (Triangulierungen, Punktwolken)	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Mathematik für Informatiker (IMI1 und 2) oder Lineare Algebra (MA4)	
<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50 % der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung	

<b>Nuetzliche Literatur</b>	Geometrie für Informatiker, Skript TU Wien 2004, Helmut Pottmann Aktuelle Fachveröffentlichungen
---------------------------------	--

## Projektseminar Biomedizinische Bildanalyse

<b>Code</b> IPBB	<b>Name</b> Projektseminar Biomedizinische Bildanalyse	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> 2 Teile Seminar und Projekt, 4 SWS	<b>Arbeitsaufwand</b> 180 h (je zur Hälfte Seminar und Projekt) 60 h Präsenzstudium 120 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden erlangen vertiefte Kenntnisse und Fähigkeiten im Gebiet Biomedizinische Bildanalyse</p> <p>lernen fortgeschrittene Methoden und Algorithmen zur automatischen Analyse biomedizinischer Bilder</p> <p>lernen wie man Algorithmen und Software für automatische Bildanalyse entwickelt</p> <p>erweitern ihre Fähigkeiten Projektergebnisse mündlich zu präsentieren und schriftlich zu dokumentieren</p> <p>erweitern ihre Fähigkeiten zur Teamarbeit und zur Strukturierung von Projekten</p>	
<b>Inhalt</b>	<p>Die Studierenden arbeiten in Teams an ausgewählten fortgeschrittenen Themen der Biomedizinischen Bildanalyse. Der Schwerpunkt liegt auf der automatischen Analyse von Zellmikroskopiebildern und medizinischen tomographischen Bildern. Beispiele für Themen sind die Segmentierung und Verfolgung (Tracking) von Zellen in Mikroskopiebildern, die Segmentierung von Blutgefäßen in tomographischen Bildern sowie die Registrierung von Magnetresonanz (MR) Bildern des menschlichen Gehirns. Die Veranstaltung besteht aus einem Seminarteil (Einarbeitung in die relevante Literatur, Erarbeitung der theoretischen Grundlagen, Vortragspräsentation) und einem Projektteil (Spezifikation eines Softwaresystems, Entwurf von Algorithmen und Implementierung von Bildanalyseverfahren, Test und Evaluierung der Verfahren, Präsentation der Ergebnisse).</p>	
<b>Voraussetzungen</b>	empfohlen sind: Grundkenntnisse in Bildverarbeitung (Computer Vision, Image Analysis), Programmierkenntnisse, Kenntnisse in Software Engineering	

<b>Pruefungs- modalitaeten</b>	Vortragspraesentationen von Zwischen- und Endergebnissen (jeder Studierende 4 Vortraege je ca. 10 Min. und anschließender Diskussion) Schriftliche Ausarbeitung der theoretischen Grundlagen, der verwendeten Methoden und der Ergebnisse (jeder Studierende ca. 10 Seiten)
<b>Nuetzliche Literatur</b>	Bekanntgabe in der Lehrveranstaltung

## Qualitätsmanagement

<b>Code</b> ISWQM	<b>Name</b> Qualitätsmanagement	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Sommersemester
<b>Lehrform</b> Vorlesung 3 SWS, Übung 3SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung im Team	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Dieses Modul vertieft die Grundkenntnisse und Fähigkeiten des Software Engineering insbesondere im Bereich des Qualitätsmanagement. Kenntnis der unter Inhalt angegebenen Methoden, Prozess und Werkzeuge Fähigkeit, Qualitätssicherung, Qualitätsmanagement und Prozessverbesserung und -management einem Softwareentwicklungsprojekt durchzuführen bzw unter Anleitung angewandte Forschung dazu durchzuführen. Fähigkeit Teilaufgaben im Team durchzuführen (eventuell mit *echten* Kunden)	
<b>Inhalt</b>	Methoden, Prozesse und Werkzeuge für Qualitätssicherung Qualitätsmanagement Verbesserung von Softwareentwicklungsprozessen Management von Softwareentwicklungsprozessen	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen (durch Erwerb einer entsprechenden Punktzahl) und erfolgreiche Teilnahme an der mündlichen oder schriftlichen Abschlussprüfung.	
<b>Nützliche Literatur</b>	Wird jährlich aktualisiert	

## Randomisierte Algorithmen

<b>Code</b> IRA	<b>Name</b> Randomisierte Algorithmen	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mindst. jedes 4. Semester
<b>Lehrform</b> Vorlesung 3 SWS, Übung 1 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 40 h Prüfungsvorbereitung 80 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Auf der Grundlage der behandelten Anwendungsbeispiele aus verschiedenen Teilgebieten der Informatik können die Studierenden die probabilistische Betrachtungs- und Vorgehensweise anwenden bei der Konstruktion und Analyse von probabilistischen und deterministischen Algorithmen, auf kombinatorische Fragestellungen, um spieltheoretische Situationen zu analysieren, auf kryptographische Fragestellungen.	
<b>Inhalt</b>	Elementare Wahrscheinlichkeitsrechnung Das Tenure-Spiel Derandomisierungstechniken Die probabilistische Methode Byzantinische Übereinkunft Stabile Heiraten und der Gale-Shapley-Algorithmus Das Minimax-Prinzip von Yao Komplexitätsanalyse des randomisierten Sortierens Randomisierte Fehlersuche und -korrektur Das Local-Lemma von Lovasz PAC-Lernen und VC-Dimension Wahrscheinlichkeitsverstärkung und Fehlerschranken Lokale Suche für k-SAT Kryptographische Protokolle	
<b>Voraussetzungen</b>	empfohlen sind: elementare Grundkenntnisse in Algorithmen wie sie z.B. im Modul Algorithmen und Datenstrukturen (IAD) vermittelt werden.	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen und Bestehen einer mündlichen oder schriftlichen Abschlussprüfung	

<b>Nuetzliche Literatur</b>	R. Motwani und P. Raghavan, Randomized Algorithms, Cambridge University Press 1995. M. Mitzenmacher und E. Upfal, Probability and Computing, Cambridge University Press, 1995. N. Alon und J. H. Spencer, The Probabilistic Method, John Wiley and Sons, 2008.
---------------------------------	---

## Räumliche Datenbanken

<b>Code</b> IRDB	<b>Name</b> Räumliche Datenbanken	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Wintersemester
<b>Lehrform</b> Vorlesung 4 SWS, Übungen 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	<p>Die Studierenden</p> <p>kennen die Prinzipien von und Anforderungen an räumliche Daten und die Verwaltung solcher Daten in verschiedenen Anwendungsbereichen (z.B. in der Geographie, Kosmologie und Biologie)</p> <p>kennen die Konzepte und Anwendungen Geographischer Informationssysteme (GIS)</p> <p>sind in der Lage, Konzepte und Modelle für räumliche Daten und der Datenmodellierung anzuwenden</p> <p>sind vertraut mit der Unterstützung zur Verwaltung von und Anfrage an räumliche Datenbanken (z.B. PostGIS)</p> <p>kennen grundlegende Methoden der algorithmischen Geometrie und sind in der Lage, entsprechende Algorithmen und Techniken anzuwenden</p> <p>kennen die wichtigsten Vertreter von Indexstrukturen zu räumlichen Daten (Gridfile, kd-Tree, Quadtree, R-Tree)</p> <p>wissen, wie in existierenden räumlichen Datenbanksystemen (z.B. PostGIS) Datenbankschemata und Anwendungen erstellt werden</p>	



<b>Inhalt</b>	<p>Prinzipien und Anforderungen an die Verwaltung räumlicher Daten</p> <p>Anwendungsbereiche zur Verwaltung und Analyse räumlicher Daten</p> <p>Unterstützung zur Verwaltung räumlicher Daten in kommerziellen und Open-Source DBMS</p> <p>Konzepte und Modelle zur Repräsentation räumlicher Daten (2D, 3D); Tessellierung und Vektormodell; Gruppen von räumlichen Objekten</p> <p>Abstrakte Datentypen für räumliche Daten</p> <p>Grundlegende Techniken der algorithmischen Geometrie (z.B. Konvexe Hülle, Sweep-Line Methoden, Polygon-Partitionierung, Schnitte von Polygonen)</p> <p>Zugriffsstrukturen für räumliche Daten, insbesondere Grid-Files, kd-Tree, Quadtrees, R-Tree</p> <p>Algorithmen und Kostenmodelle für Zugriffsstrukturen zu räumlichen Daten</p> <p>Konzepte der Anfrageverarbeitung und -optimierung für räumliche Datenbanken, insb. Spatial Join</p> <p>Temporale Datenbanken und Indexstrukturen</p> <p>Moving Objects: Anwendungen, Anfragen, Indexstrukturen (B<sup>x</sup>-Tree und TPR-Tree)</p> <p>Einführung in das Mining räumlicher Daten (Clustering, Entdecken von Ausreißern)</p> <p>Überblick über Unterstützung zur Verwaltung räumlicher Daten in kommerziellen und Open-Source DBMS (PostgreSQL, MySQL, Oracle Spatial, GRASS)</p>
<b>Voraussetzungen</b>	empfohlen sind: Algorithmen und Datenstrukturen (IAD), Datenbanken 1 (IDB1)
<b>Pruefungs-modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung
<b>Nuetzliche Literatur</b>	<p>Spatial Databases ? With Applications to GIS. Philippe Rigaux, Michel Scholl, Agnes Voisard. Morgan Kaufmann, 2001.</p> <p>Computational Geometry: Algorithms and Applications Mark de Berg, Otfried Cheong, Marc van Kreveld, und Mark Overmars, Springer, Berlin, 2008.</p> <p>Forschungsartikel aus Tagungsbänden und Journals</p>

## Requirements Engineering

<b>Code</b> ISWRE	<b>Name</b> Requirements Engineering	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Sommersemester
<b>Lehrform</b> Vorlesung 3 SWS, Übung 3SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung im Team	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Dieses Modul vertieft die Grundkenntnisse und Fähigkeiten des Software Engineering insbesondere im Bereich des Requirements Engineering. Kenntnis der unter Inhalt angegebenen Methoden, Prozess und Werkzeuge Fähigkeit, Unternehmensmodellierung, Anforderungserhebung und -verhandlung und Requirements Management in einem Softwareentwicklungsprojekt durchzuführen bzw unter Anleitung angewandte Forschung dazu durchzuführen. Fähigkeit Teilaufgaben im Team durchzuführen (eventuell mit *echten* Kunden)	
<b>Inhalt</b>	Methoden, Prozesse und Werkzeuge für Unternehmensmodellierung Prozessverbesserung in Unternehmen Anforderungserhebung und -verhandlung Requirements Management (Verbreitung, Prüfung und Aktualisierung von Anforderungen)	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden	
<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (durch Erwerb einer entsprechenden Punktzahl) und erfolgreiche Teilnahme an der mündlichen oder schriftlichen Abschlussprüfung.	
<b>Nuetzliche Literatur</b>	Wird jährlich aktualisiert	

## Scientific Visualization

<b>Code</b> ISV	<b>Name</b> Scientific Visualization	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes 4. Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	Die Studierenden kennen fortgeschrittene Konzepte und Techniken der Wissenschaftlichen Visualisierung. kennen fortgeschrittene Algorithmen, mathematische Grundlagen, Datenstrukturen und Implementierungsaspekte. haben praktische Erfahrung mit Visualisierungswerkzeugen.	
<b>Inhalt</b>	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	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	
<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung	
<b>Nuetzliche Literatur</b>	C.D. Hansen, C.R. Johnson, The Visualization Handbook, 2005.	

## Simulationswerkzeuge

<b>Code</b> ISIMW	<b>Name</b> Simulationswerkzeuge	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium, 15 h Prüfungsvorbereitung 105 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>beherrschen grundlegende Verfahren zur Repräsentation komplexer Geometrien (CAD-Modelle)</li> <li>verstehen die Methoden zur Vernetzung komplexer Geometrien</li> <li>beherrschen standardisierte Datenformate zur Speicherung großer, wissenschaftlicher Datenmengen</li> <li>haben grundlegende Verfahren zur Visualisierung von Ergebnissen von Simulationsprogrammen kennengelernt</li> <li>können diese Verfahren im Kontext realistischer Anwendungsprobleme konkret einsetzen</li> </ul>	
<b>Inhalt</b>	<p>Einführung in die grundlegenden Methoden, Techniken und Softwarerealisierungen bei Entwurf und Repräsentation komplexer Geometrien und deren Vernetzung. Neben der reinen Simulation eines Vorganges, d.h. der numerischen Lösung des mathematischen Problems nehmen die Vor- und Nachverarbeitungsschritte einen zunehmend größeren Stellenwert bei der Durchführung von Simulationsstudien ein. Zur Vorverarbeitung zählen z.B. die Erstellung und Repräsentation komplexer, dreidimensionaler Geometrien mit Splines und Nurbs</p> <p>Zuordnung von Koeffizienten, z.B. Material-eigenschaften, und Randbedingungen</p> <p>Gittergenerierung mit Delauney und Advancing- Front-Methoden</p> <p>Speicherung größter Datenmengen, z.B. HDF5</p> <p>Datenaustausch mit Visualisierungssystemen und anderen Programmen</p> <p>Konkrete Benutzung von Simulationswerkzeugen an konkreten Beispielen, z.B. Salome, OpenCascade, Gmsh, Paraview/VTK</p>	
<b>Voraussetzungen</b>	empfohlen sind: Grundlagen der Numerik	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen (Erreichen von 50% der Punkte aus den Übungsaufgaben).	

<b>Nuetzliche Literatur</b>	
---------------------------------	--

## Social Network Analysis

<b>Code</b> ISNA	<b>Name</b> Social Network Analysis	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Wintersemester
<b>Lehrform</b> Vorlesung 4 SWS, Übungen 2 SWS	<b>Arbeitsaufwand</b> 240 h; davon 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (eventuell in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	<p>Die Studierenden sind in der Lage, charakteristische Eigenschaften sozialer Netzwerke zu formulieren</p> <p>sind in der Lage, wesentliche Graph-Algorithmen zu implementieren und anzuwenden</p> <p>können Netzwerkmodelle beschreiben und deren Parameter charakterisieren, berechnen und analysieren</p> <p>kennen die Techniken und Anwendungen des supervised und unsupervised Learning in Netzwerken</p> <p>wissen, wie Maße zu Netzwerken zu berechnen und zu interpretieren sind</p> <p>kennen verschiedene generative Modelle zur Entwicklung/Erzeugung von Graphen/Netzwerken</p> <p>kennen die grundlegenden Methoden zur Erkennung und der Evolution von Communities in Netzwerken</p> <p>sind mit den Begriffen des Einflusses und der Homophilie in Netzwerken vertraut</p> <p>kennen die Grundlagen von Recommender Systems</p> <p>wissen, wie das Verhalten von individuellen Benutzern und von Gruppen in Netzwerken bestimmt werden kann</p>	

<b>Inhalt</b>	<p>Graphentheorie und Graph-Algorithmen  Maße für Netzwerke (u.a, Zentralität, Transitivität, Reziprozität, Ähnlichkeiten)  Netzwerkmodelle (u.a. zufällige Graphen, Small-World Modell)  Grundlagen des Data Mining auf Graphen (Supervised und unsupervised Learning)  Analyse von Communities  Ausbreitung von Informationen in Netzwerken (Herdenverhalten, Kaskadierungen, Ausbreitung von Neuigkeiten, Epidemien)  Einfluss und Auswahl in Netzwerken  Berechnung von Empfehlungen in sozialen Medien  Verhaltensanalyse (von Individuen und Gruppen)  Erkennung und Entwicklung von Events in Netzwerken</p>
<b>Voraussetzungen</b>	<p>empfohlen sind: Algorithmen und Datenstrukturen (IAD), Knowledge Discovery in Databases (IKDD)</p>
<b>Pruefungs-modalitaeten</b>	<p>Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und erfolgreiche Teilnahme an einer schriftlichen Prüfung</p>
<b>Nuetzliche Literatur</b>	<p>Reza Zafarani, Mohammad Abbasi, Huan Liu: Social Media Mining-An Introduction, Cambridge University Press, 2014.  David Easley, Jon Kleinberg: Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.  Stanley Wasserman, Katherine Faust: Social Network Analysis-Methods and Applications, Cambridge University Press, 1994.</p>

## Software Evolution

<b>Code</b> ISWEvol	<b>Name</b> Software Evolution	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 90h; davon 30 h Vorlesung 35 Aufgabenbearbeitung und Aufarbeitung/Selbststudium 25 h Prüfungsvorbereitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden können nach der Vorlesung: ein Software-Reengineering-Projekt fachlich planen und beurteilen, bei der Ersterstellung von Software die Evolutionsfähigkeit konzeptuell sicherstellen, ein Wartungskonzept für eine erstellte Software aufbauen.</p> <p>Sie kennen die Unterschiede und Herausforderungen der Software-Weiterentwicklung versus der Softwareneuentwicklung ? und worauf die/der InformatikerIn hierbei achten muss, sowohl aus Sicht eines Softwareherstellers als auch aus der Sicht der NutzerInnen von Software, die klassischen Techniken der Softwaresanierung, die Typologie der Softwarewartung und das Management der Fehlerbehebung, die Relevanz der Thematik in der Praxis der industriellen Softwareerstellung.</p>	



<b>Inhalt</b>	<p>Dieses Modul vermittelt aus konzeptioneller Sicht die Grundlagen für ein erfolgreiches Lebenszyklusmanagement von Software nach ihrer Ersterstellung. Die Vorlesungsinhalte wurden aufgrund der 15-jährigen Erfahrung des Dozenten in der Praxis zusammengestellt, auf Basis aktueller Forschung und Lehre.</p> <p>Einleitung  Begriffsklärung, Grundlagen  Softwareevolution  Softwarewartung, Softwareerhaltung  Software-Reengineering  Evolution und Weiterentwicklung  Management der Softwareevolution  Zusammenfassung</p>
<b>Voraussetzungen</b>	<p>empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden</p>
<b>Pruefungs-modalitaeten</b>	<p>Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Anzahl der Teilnehmer)</p>
<b>Nuetzliche Literatur</b>	<p>Arnold, R. (Hrsg.): Software Reengineering. IEEE Computer Society Press, Los Alamitos 1993  Fowler, M.: Refactoring ? Improving the Design of Existing Code. Addison-Wesley, Reading, Massachusetts, 1999  von Hahn, E.: Werterhaltung von Software. DUV, Wiesbaden 2005  Müller, B.: Reengineering. Eine Einführung. Teubner, Stuttgart 1997  Sneed, H.M.; Hasitschka, M.; Teichmann, M.-T.: Software-Produktmanagement. Wartung und Weiterentwicklung bestehender Anwendungs-systeme. dpunkt, Heidelberg 2005  Smith, D.D.: Designing Maintainable Software. Springer, Heidelberg 1999</p>

## Software Ökonomie

<b>Code</b> ISWök	<b>Name</b> Software Ökonomie	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 90h; davon 30 h Vorlesung 35 Aufgabenbearbeitung und Aufarbeitung/Selbststudium 25 h Prüfungsvorbereitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	<p>Die Studierenden können nach der Vorlesung grob den Preis und die Lizenzierung einer erstellten Software ermitteln, die Vermarktung von Software planen und anstoßen, grob die Bilanz sowie die Gewinn- und Verlustrechnung eines Softwareherstellers verstehen, den Wert einer Software mit seinen verschiedenen Komponenten beurteilen, aus Sicht des Herstellers sowie aus Sicht der Nutzer, Preisverhandlungen zu Softwareprojekten planen. Sie kennen die Grundzüge der Kosten- und Leistungsrechnung (soweit sie für die Softwareerstellung relevant ist), die unterschiedlichen Vertragsarten, die im Umfeld der Softwareerstellung zum Einsatz kommen, die wichtigsten Verhandlungsstrategien bei der Verhandlung von Softwareverträgen, rechtliche Aspekte im Bereich der IT-Kriminalität, die Relevanz der Vorlesungsthemen in der Praxis der industriellen Softwareerstellung.</p>	

<b>Inhalt</b>	<p>Dieses Modul vermittelt aus konzeptioneller Sicht die Grundlagen der Wirtschaftsinformatik, wie sie für die Softwareerstellung relevant sind. Die Vorlesungsinhalte wurden aufgrund der 15-jährigen Erfahrung des Dozenten in der Praxis zusammengestellt, auf Basis aktueller Forschung und Lehre.</p> <p>Einleitung          Begriffsklärung, Grundlagen          Softwareökonomie          Management von Softwareprojekten          Wertermittlung von Software          Bepreisung von Software          Software-Marketing          Verhandlungen und Verträge          Bilanzierung und Rechnungslegung          IT-Kriminalität          Schadensabwendung          Zusammenfassung</p>
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse und Fähigkeiten wie sie in Modul Einführung in Software Engineering (ISW) vermittelt werden
<b>Pruefungs-modalitaeten</b>	Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Anzahl der TeilnehmerInnen)
<b>Nuetzliche Literatur</b>	<p>Buxmann, P.; Diefenbach, H.; Hess, T.: Die Softwareindustrie. Ökonomische Prinzipien, Strategien, Perspektiven. Heidelberg, 2008</p> <p>Herzwurm, G.; Pietsch, W.: Management von IT-Produkten. Heidelberg, 2009</p> <p>Mertens, P. (Hrsg.): Lexikon der Wirtschaftsinformatik, 3. Auflage, Heidelberg 2001</p> <p>Versteegen, G.: Marketing in der IT-Branche. Heidelberg 2003</p> <p>Wöhe, G.: Einführung in die Allgemeine Betriebswirtschaftslehre. München 2010</p>

## Software-Praktikum für Fortgeschrittene

<b>Code</b> IFM	<b>Name</b> Software-Praktikum für Fortgeschrittene	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Praktikum 6 SWS	<b>Arbeitsaufwand</b> 240 h; davon mind. 25 h Präsenzzeit 10 h Vorbereitung Vortrag	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik  Fachübergreifende Kompetenzen: Bachelor Mathematik, Master Mathematik
<b>Lernziel</b>	<p>Die Studierenden erlangen vertiefende Problemlösungskompetenz für komplexe Entwurfs- und Implementierungsaufgaben können Problemanalyse- und Beschreibungstechniken klar darstellen, differenzieren und anwenden vertiefen Programmierkenntnisse in der jeweiligen für das Projekt erforderlichen Programmiersprache sind in der Lage, das Projekt mit Hilfe einer Softwareentwicklungsumgebung durchzuführen</p> <p>Zusätzlich werden die projektypischen Kompetenzen vertieft, insbesondere das Arbeiten im Team (von bis zu drei Studierenden): Durchführung und Evaluation von Projekten und ihrer Phasenstruktur Planung und Durchführung von Projekt- und Teamarbeit. Zu den zu trainierenden Softskills zählen somit insbesondere Teamfähigkeit, Verfeinerung von Präsentationstechniken, etwaige Erschließung wissenschaftlicher Literatur sowie eigenverantwortliches Arbeiten.</p>	
<b>Inhalt</b>	<p>Domänenkenntnisse abhängig von den DozentInnen; allgemeine Lerninhalte sind: Vertiefung in die Projektarbeit Eigenständige Entwicklung von komplexer Software und deren Dokumentation</p>	
<b>Voraussetzungen</b>	keine	
<b>Prüfungsmodalitäten</b>	Bewertung der dokumentierte Software, des Projektberichts und des Vortrag	

<b>Nuetzliche Literatur</b>	
---------------------------------	--

## Verteilte Systeme I

<b>Code</b> IVS1	<b>Name</b> Verteilte Systeme I	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes 4. Semester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 180 h; davon 60 h Präsenzstudium 15 h Prüfungsvorbereitung 105 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	<p>Verständnis der unterschiedlichen parallelen Architekturen und Besonderheiten von verteilten Systemen</p> <p>Kenntnis der grundlegenden theoretischen Probleme und Algorithmen in verteilten Systemen (z.B. Datenaufteilung auf Prozessoren)</p> <p>Kenntnis über Anwendbarkeit diverser paralleler Frameworks (Pthreads, MPI, Map-Reduce) für eine gegebene Architektur</p> <p>Fähigkeit zur Erstellung von parallelen und verteilten Programmen in Java (z.B. mit Threads, RMI) oder Groovy (GPars)</p> <p>Fähigkeit der praktischen Verwendung von *modernen* Ansätzen der parallelen/verteilten Programmierung (Map-Reduce, Actors, datenstromorientierte Ansätze)</p> <p>Vertrautheit mit Praxisfällen (Web-Dienste, parallele Verarbeitung von Daten)</p>	

<b>Inhalt</b>	Das Modul behandelt die grundlegenden Prinzipien und Techniken der parallelen und verteilten Systeme im Kontext von Netzwerkanwendungen und der parallelen Verarbeitung von Daten. Es werden Konzepte aus den Bereichen Architekturen, Protokolle, Programmierung, Softwareframeworks und Algorithmen vorgestellt. Ein wesentlicher Teil der Vorlesung widmet sich der parallelen und verteilten Programmierung, insbesondere moderneren Ansätzen wie Map-Reduce, Actors, datenstromorientierte Programmierung. Ergänzende Themen umfassen Fehlertoleranz, effiziente Protokolle und Skalierbarkeit. Die Umsetzung in die Praxis erfolgt an Beispielen aus den Bereichen Web-Dienste oder Verarbeitung großer Datenmengen. Das Modul soll die Studierenden befähigen, Spezifika und Probleme der verteilten Systeme zu verstehen, verteilte Anwendungen zu programmieren und Softwareframeworks wie Hadoop effizient einzusetzen.
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse in Java (z.B. durch Einführung in Software Engineering (ISW)) sowie in Betriebssystemen (z.B. IBN)
<b>Pruefungs-modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer Abschlussprüfung.
<b>Nuetzliche Literatur</b>	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.

## Visualisierung im Bereich Cultural Heritage

<b>Code</b> IVCH	<b>Name</b> Visualisierung im Bereich Cultural Heritage	
<b>Leistungspunkte</b> 2 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> unregelmäßig
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 60 h; davon 30 h Präsenzstudium, 30 h Prüfungsvorbereitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, Lehramt Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Die Studierenden sind mit unterschiedlichen Scantechniken vertraut und können Georadardaten interpretieren. Sie beherrschen den Umgang mit 3D Scan-, Georadar- und Magnetfelddaten, geophysikalischer Prospektion und weiteren Untersuchungen von Messdaten und kennen die Herangehensweise mit 2D und 3D Bildverarbeitung zur Erkennung von Merkmalen (Schrift). Sie wissen um die ethischen Grundsätze bei der Rekonstruktion, Befund und Hypothese (London Charter).	
<b>Inhalt</b>	Weißlicht- und Time-of-flight-Scanner, Rekonstruktionen von Gefäßen und Gebäuden, 3D-Puzzle, Skelettierung, ethische Grundsätze	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Computergraphik 1 (ICG1)	
<b>Prüfungsmodalitäten</b>	Teilnahme an den Vorlesungen und Bestehen einer mündlichen oder schriftlichen Abschlussprüfung (je nach Teilnehmerzahl)	
<b>Nützliche Literatur</b>	Clive Orton: Mathematics in Archaeology. Cambridge, MA, Cambridge University Press, 1982 Katsushi Ikeuchi, Daisuke Miyazaki (editors): Digitally Archiving Cultural Objects. Springer, 2007 Christian Hörr. Algorithmen zur automatisierten Dokumentation und Klassifikation archäologischer Gefäße. Dissertation, TU Chemnitz, 2011	



## Volumenvisualisierung

<b>Code</b> IVV	<b>Name</b> Volumenvisualisierung	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes Sommersemester
<b>Lehrform</b> Vorlesung 2 SWS, Übung 3 SWS	<b>Arbeitsaufwand</b> 240 h; davon 75 h Präsenzstudium 15 h Prüfungsvorbereitung 150 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik, M.Sc. Scientific Computing
<b>Lernziel</b>	<p>Die Studierenden verstehen, wie experimentelle Daten zustande kommen und welche prinzipiellen Informationselemente diese enthalten, die für eine wissenschaftliche Visualisierung notwendig sind. Sie lernen die Prinzipien kennen, wie man zwischen kontinuierlichen (wirkliche Verteilung) und diskreten Signalen (Darstellung im Computer) wechseln kann. Zudem werden sie eingeführt in Methoden der Konversion von Oberflächendaten in Volumendaten und umgekehrt. Schließlich werden ausgehend von theoretischen physikalischen Prinzipien sie in der Lage sein, verschiedene Visualisierungstechniken abzuleiten und die Näherungen, die hierfür gemacht werden, zu verstehen.</p> <p>Sie sind damit in der Lage, komplexe volumenorientierte Daten adäquat mathematisch zu repräsentieren, zu transformieren und die wesentlichen Strukturen mit adäquat darauf angepassten Verfahren der Visualisierung darzustellen.</p>	
<b>Inhalt</b>	<p>Einführung in die Visualisierung wissenschaftlicher Daten aus den Natur- und Lebenswissenschaften.</p> <p>Diskrete und kontinuierliche Repräsentation von Daten sowie numerische und computergraphische Methoden der Interpolation.</p> <p>Methoden der Konversion von Oberflächenrepräsentationen in Volumenrepräsentationen und umgekehrt, Optimierungstechniken für effiziente Algorithmen</p> <p>Ableitung und Varianten von Volumenvisualisierungstechniken</p> <p>Beschleunigungsverfahren und Parallelisierungsmethoden für Volumenvisualisierung</p> <p>Programmiertechnik: GPU-Programmierung</p>	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Algorithmen und Datenstrukturen (IAD)	

<b>Pruefungs- modalitaeten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen oder mündlichen Abschlussprüfung
<b>Nuetzliche Literatur</b>	Engel et al: Real-Time Volume Graphics <a href="http://www.real-time-volume-graphics.org">www.real-time-volume-graphics.org</a> , Schroeder et al: VTK Textbook <a href="http://www.kitware.com/products/books/vtkbook.html">http://www.kitware.com/products/books/vtkbook.html</a>

## Wissensmanagement und Entscheidungen im Software Engineering

<b>Code</b> ISWKM	<b>Name</b> Wissensmanagement und Entscheidungen im Software Engineering	
<b>Leistungspunkte</b> 3 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> jedes 2. Wintersemester
<b>Lehrform</b> Vorlesung+Übung 2 SWS	<b>Arbeitsaufwand</b> 90 h; davon 30 h Präsenzstudium 15 h Prüfungsvorbereitung 45 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Die Teilnehmer/innen kennen vertiefende Software Engineering Methoden, die Entscheidungen unterstützen bei Anforderungspriorisierung, Entwurf, Managemententscheidungen und Risikomanagement. Sie wissen, wie man im Arbeitsalltag Wissen verwaltet und haben eine Einführung in die Entscheidungstheorie erhalten.	
<b>Inhalt</b>	Wissensmanagement Ontologien Rationale Re-engineering learning organization Entscheidungen Management-Entscheidungen, Business Case Risikomanagement Anforderungspriorisierung Entscheidungen im Entwurf: ATAM, SAAM, CBAM Entscheidungstheorie Entscheiden unter Ungewissheit Mathematical Economics Entscheidung mit mehreren Parteien: Harvard-Konzept, Verhandlungen Spieltheorie Fehlentscheidungen/ Decision Traps/ Biases	
<b>Voraussetzungen</b>	empfohlen sind: Vorlesung und Übung Einführung in Software Engineering (ISW) oder vergleichbare Vorkenntnisse	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Hausaufgaben (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer Abschlussprüfung.	

<b>Nuetzliche Literatur</b>	Raiffa, Howard; Richardson, John; Metcalfe, David: Negotiation analysis - the science and art of collaborative decision making, Belknap, Cambridge, 2002
---------------------------------	--

## 2.4 Module aus dem B.Sc./M.Sc. Mathematik

Die Module mit den Kürzeln MC4, MD1, MD2, MD3, MD4, MD5, MD6 und ME3 wurden aus dem Modulhandbuch des Bachelor-Studiengangs Mathematik (Fassung vom 10.10.2014) entnommen.

Die Module mit den Kürzeln MG19, MG20, MG21, MH5, MH7, MH8, MH12, MH13, MH14, MH15, MH16, MH17, MH18, MH21 und MH27 wurden aus dem Modulhandbuch des Master-Studiengangs Mathematik (Fassung vom 27.01.2015) entnommen. Das Modul MH19 wurde zum Sommersemester 2016 aktualisiert.

## Wahrscheinlichkeitstheorie

<b>Code</b> MC4	<b>Name</b> Wahrscheinlichkeitstheorie	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes zweite Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> BA Mathematik, LA Mathematik ab dem 4. Semester
<b>Lernziel</b>	Grundlagen für alle Gebiete der Wahrscheinlichkeitstheorie und Statistik	
<b>Inhalt</b>	<p>I. Maß- und Integrationstheorie: <math>\sigma</math>-Algebren, Borel-<math>\sigma</math>-Algebra, messbare Abbildungen, Konstruktion von Wahrscheinlichkeitsmaßen, Produkträume. Erwartungswert als Maßintegral, Sätze von Lebesgue, Beppo Levi, Fubini und Radon-Nikodym.</p> <p>II. Konvergenz von Zufallsvariablen: <math>L^p</math>-Räume, Zusammenhang zwischen fast sicherer, stochastischer und <math>L^p</math>-Konvergenz, Starkes Gesetz der großen Zahlen, Konvergenz in Verteilung, charakteristische Funktionen, zentraler Grenzwertsatz.</p> <p>III. Bedingte Verteilungen: Bedingte Erwartungen, Markov-Kerne, Martingale in diskreter Zeit.</p> <p>IV. Stochastische Prozesse: Brownsche Bewegung, Poisson-Prozess, Empirischer Prozess.</p>	
<b>Voraussetzungen</b>	empfohlen sind: Analysis I und II (MA1, MA2) , Lineare Algebra I und II (MA4, MA5), Höhere Analysis (MA3), Einführung in die Wahrscheinlichkeitstheorie und Statistik (MA 8)	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur. Art und Zeitpunkt einer Wiederholungsprüfung wird vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.	
<b>Nützliche Literatur</b>	<p>Bauer, H.: Wahrscheinlichkeitstheorie, de Gruyter.</p> <p>Billingsley, P.: Probability and Measure, Wiley.</p> <p>Dudley, R.N.: Real Analysis and Probability</p> <p>Durrett, R.: Probability: Theory and Examples, Duxbury Press</p> <p>Jacod, J. and Protter, P.: Probability Essentials, Springer</p> <p>Shiryaev, A.: Probability, Springer.</p>	

## Numerik

<b>Code</b> MD1	<b>Name</b> Numerik	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes zweite Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> für Studiengänge BA Mathematik, BA Informatik, BA Physik, LA Mathematik, LA Informatik jeweils ab dem 3. Studiensemester
<b>Lernziel</b>	Kenntnisse der numerischen Lösung von Anfangswert- und Randwertaufgaben gewöhnlicher Differentialgleichungen und einfacher partieller Differentialgleichungen Abstraktes und algorithmisches Denken, Anwendung von Techniken der Analysis und linearen Algebra	
<b>Inhalt</b>	I. Theorie von Anfangs- und Randwertaufgaben II. Einschrittmethode: Konsistenz, Stabilität, Konvergenz. III. Numerische Stabilität und steife Anfangswertaufgaben IV. Andere Verfahrensklassen: Lineare Mehrschrittmethoden, Extrapolationsmethoden, Galerkin-Methoden (optional). V. Lösung von Differentiellalgebraischen Aufgaben VI. Lösung von Randwertaufgaben: Schießverfahren, Differenzen und Galerkin-Verfahren (optional). VII. Differenzenverfahren für elliptische partielle Differentialgleichungen, Laplace-Gleichung, 5-Punkte-Approximation. VIII. Iterative Lösungsverfahren für diskretisierte Probleme.	
<b>Voraussetzungen</b>	empfohlen sind: Analysis I (MA1), Lineare Algebra I (MA4), Einführung in die Numerik (MA7)	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben, mit benoteten 2-stündigen Klausuren, Wiederholungsmöglichkeit mit der Vorlesung im Folgejahr.	
<b>Nützliche Literatur</b>	Bekanntgabe in der Vorlesung (Vorlesungsskriptum)	

## Statistik

<b>Code</b> MD2	<b>Name</b> Statistik	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes zweite Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> BA Mathematik, LA Mathematik ab dem 4. Semester
<b>Lernziel</b>	Prinzipien der mathematischen Statistik	
<b>Inhalt</b>	I. Entscheidungstheorie: Dualität von Tests und Konfidenzbereichen, Neyman-Pearson-Theorie, allgemeine Entscheidungsverfahren, Risikofunktionen, Bayes- und Minimaxoptimalität II. Asymptotische Statistik: Verteilungsapproximation, Fisher-Information, relative asymptotische Effizienz von Tests und Schätzern, Likelihood-basierte Verfahren, nichtparametrische Verfahren.	
<b>Voraussetzungen</b>	empfohlen sind: Analysis I (MA1), Lineare Algebra I (MA4), Einführung in die Wahrscheinlichkeitstheorie u. Statistik (MA8), Wahrscheinlichkeitstheorie (MC4)	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur. Art und Zeitpunkt einer Wiederholungsprüfung wird vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.	
<b>Nützliche Literatur</b>	Bickel, P. J. and Doksum, K. A.: Mathematical Statistics, Prentice Hall Lehmann, E. L.: Testing Statistical Hypotheses, Springer Verlag Van der Vaart, A. W.: Asymptotic Statistics, Cambridge University Press	



## Lineare Optimierung

<b>Code</b> MD3	<b>Name</b> Lineare Optimierung	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> BA Mathematik, LA Mathematik ab dem 2. Semester
<b>Lernziel</b>	Probleme, Theorie, Methoden und Algorithmen der Linearen Optimierung	
<b>Inhalt</b>	Die Vorlesung behandelt die folgenden Themen: Formulierung von linearen Optimierungsproblemen Dualitätstheorie Struktur von Polyedern Die Simplexmethode, Grundversion und Varianten Der duale Simplex-Algorithmus Postoptimale Analyse und Re-Optimierung Polynomiale Algorithmen zur Linearen Optimierung Innere-Punkte-Methoden	
<b>Voraussetzungen</b>	empfohlen sind: Lineare Algebra I, Programmierkenntnisse	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.	
<b>Nützliche Literatur</b>	Padberg: Linear Optimization and Extensions Chvátal: Linear Programming Wright: Primal-Dual Interior-Point Methods	

## Nichtlineare Optimierung

<b>Code</b> MD4	<b>Name</b> Nichtlineare Optimierung	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> BA Mathematik, LA Mathematik ab dem 3. Semester
<b>Lernziel</b>	Probleme, Theorie, Methoden und Algorithmen der Nichtlinearen Optimierung	
<b>Inhalt</b>	Die Vorlesung behandelt die folgenden Themen: Endlich-dimensionale, glatte, kontinuierliche, nichtlineare Optimierungsprobleme, Optimalitätsbedingungen für unbeschränkte und beschränkte Optimierungsprobleme, Gradientenverfahren, Konjugierte Gradienten-(CG-) Verfahren, Line Search, Newton- und Quasi-Newton-SQP-Verfahren, Gauß-Newton-Verfahren, Behandlung von Ungleichungsbeschränkungen, Trust-Region- Verfahren, Automatische Differentiation	
<b>Voraussetzungen</b>	empfohlen sind: Lineare Algebra I, Analysis I und II, Programmierkenntnisse	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben. Benotete Klausur bzw. mündliche Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.	
<b>Nützliche Literatur</b>	Nocedal, Wright: Numerical Optimization Gill, Murray, Saunders, Wright: Practical Optimization Geiger, Kanzow: Numerik (un)restringierter Optimierung Jarre, Stoer: Optimierung	

## Wissenschaftliches Rechnen

<b>Code</b> MD5	<b>Name</b> Wissenschaftliches Rechnen	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes vierte Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> BA und MA Mathematik, LA Mathematik
<b>Lernziel</b>	Grundlegende Methoden der Modellbildung und Simulation. Schwerpunktartig werden die Prozesse betrachtet, die sich mit Hilfe partieller Differentialgleichungen beschreiben lassen.	
<b>Inhalt</b>	Hauptthemen sind: I. Modellbildung: Modellierungssystematik, Diskrete Systeme, kontinuierliche Prozesse, Standardansätze zur Modellierung, Erhaltungsgleichung. II. Simulationsmethoden: Grundlegende Diskretisierungsverfahren, elementare Lösertechniken. III. Anwendungsbeispiele: Hier kommen einfache Anwendungen aus Biologie, Medizin, Physik u. a. zur Diskussion.	
<b>Voraussetzungen</b>	empfohlen sind: Mathematische Grundvorlesungen MA1 bis MA8	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitpunkt einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.	
<b>Nützliche Literatur</b>	Es wird ein Skriptum angeboten.	

## Computational Statistics

<b>Code</b> MD6	<b>Name</b> Computational Statistics	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung mit Übungen	<b>Arbeitsaufwand</b> 240	<b>Verwendbarkeit</b> BA Mathematik Bachelor, LA Mathematik, Diplom Mathematik, BA/Master Ang. Informatik, Master Scientific Computing
<b>Lernziel</b>	Statistische Modellierung; praktische Anwendung statistischer Verfahren am Computer; Output-Interpretation und Analyse; Modell- und Datendiagnostik; Programmierung in R. Anwendung eines Statistik-Systems (als Beispiel R); Output-Analyse und Diagnostik; Entwurf und Implementierung einfacher stochastischer Simulationen.	

<b>Inhalt</b>	<p>In diesem Kurs soll die Anwendung statistischer Verfahren am Computer eingeübt werden. Statistische Grundkenntnisse werden vorausgesetzt. Der Hintergrund der im Kurs verwendeten Methoden wird bei Bedarf wiederholt. Verwendet wird die speziell für die Statistik entwickelte Programmiersprache R. Vorkenntnisse über R sind nicht erforderlich. Eine Einführung in R ist Teil des Kurses. Dieser Teil wird evtl. als Blockkurs angeboten. Es wird empfohlen, diesen Teil vorab zu besuchen.</p> <p>'Computational Statistics' ist der Zweig der Statistik, der von den heutigen rechnerischen Möglichkeiten ausgeht. Neben effizienter Implementierung klassischer Verfahren stehen oft neue bis hin zu experimentellen Ansätzen. Die Vorlesung stellt typische Konzepte der Statistik vor und illustriert ihre praktische Anwendung.</p> <p>Themenbereiche sind:</p> <ul style="list-style-type: none"> <li>- Diagnostik und Anpassungstests für univariate Verteilungen</li> <li>- Lineare Modelle, incl. Residuenanalyse und Regressionsdiagnostik</li> <li>- Allgemeine Zwei-Stichproben-Vergleiche</li> <li>- Nichtparametrische Verfahren</li> <li>- Monte-Carlo-Verfahren, Resampling-Verfahren, Simulation</li> <li>- Beispiele für multivariate Methoden, wie z.B. multidimensionale Skalierung, Hauptkomponenten-Analyse, Projection Pursuit</li> </ul>
<b>Voraussetzungen</b>	empfohlen sind: Grundkenntnisse in der Statistik, z.B. MA8 Einführung in die Wahrscheinlichkeitstheorie und Statistik, MD2 Statistik (kann parallel besucht werden)
<b>Prüfungsmodalitäten</b>	Programmieraufgaben: Implementierung statistischer Auswertung für gegebene Datensätze und schriftliche Analyse der Ergebnisse.
<b>Nützliche Literatur</b>	<p>John M. Chambers: Computational Methods for Data Analysis</p> <p>G. Sawitzki: Computational Statistics: An Introduction to R</p>

## Mathematische Logik

<b>Code</b> ME3	<b>Name</b> Mathematische Logik	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b> mind. jedes vierte Semester
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> BA Mathematik, BA Informatik, LA Mathematik, LA Informatik ab dem 3. Semester.
<b>Lernziel</b>	Einführung in die verschiedenen Teilgebiete der Mathematischen Logik.	
<b>Inhalt</b>	I. Prädikatenlogik: Untersuchung der in der Mathematik üblichen logischen Schlussweisen. II. Mengenlehre: Grundagentheorie der Mathematik sowie Theorie der Ordinal- und Kardinalzahlen. III. Modelltheorie: Zusammenhang zwischen axiomatischen Theorien und ihren Modellen mit Beispielen aus der Algebra. IV. Berechenbarkeitstheorie: Eigenschaften des Begriffes der berechenbaren Funktion. V. Beweistheorie: Grenzen der Formalisierbarkeit, Unvollständigkeit und Unentscheidbarkeit.	
<b>Voraussetzungen</b>	empfohlen sind: Lineare Algebra I (MA4), Einführung in die Praktische Informatik (IPI)	
<b>Prüfungs- modalitäten</b>	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitpunkt einer Wieder- holungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.	
<b>Nützliche Literatur</b>	Bekanntgabe in der Vorlesung	

## Computeralgebra I

<b>Code</b> MG19	<b>Name</b> Computeralgebra I	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master, Informatik Master
<b>Lernziel</b>	Methodenkenntnis in Computeralgebra, Selbständiges Lösen von Aufgaben, Umgang mit CA-Systemen	
<b>Inhalt</b>	<p>Die Vorlesung Computeralgebra befasst sich mit der Theorie und der Komplexität grundlegender mathematischer Algorithmen und deren Implementierungen in Computeralgebrasystemen.</p> <p>Hauptthemen sind:</p> <p>I. Schnelle Arithmetik: Komplexität der elementaren Grundoperationen, diskrete Fouriertransformation, schnelle Multiplikation und schneller Euklidischer Algorithmus, Subresultanten und Polynomrestfolgen, modulare Algorithmen, Rechnen mit algebraischen Zahlen, schnelle Matrizenmultiplikation</p> <p>II. Primzerlegung und Primzahltests: Primzahltest von Solovay-Strassen und Miller-Rabin, der AKS-Primzahlentest, RSA- Schema, elementare Primzahlzerlegungsverfahren, quadratisches Sieb, Irreduzibilitätstest für Polynome, Berlekamp-Algorithmen, Zassenhaus-Algorithmus, Gitter-Basis-Reduktion, Faktorisierung multivariater Polynome</p> <p>III. Gröbnerbasen-Algorithmen: Gröbnerbasen und reduzierte Gröbnerbasen, Buchberger-Algorithmus, Eliminationstheorie, Algorithmen für elementare Idealoperationen, Berechnung der Dimension eines Ideals.</p>	
<b>Voraussetzungen</b>	empfohlen ist: Algebra I (MB1)	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekanntgegeben.	

<b>Nützliche Literatur</b>	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 (Skriptum, in Vorbereitung)
--------------------------------	---



## Computeralgebra II

<b>Code</b> MG20	<b>Name</b> Computeralgebra II	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master, Informatik Master
<b>Lernziel</b>	Vertiefte Kenntnisse in Computeralgebra, Selbständiges Lösen von Aufgaben, Umgang mit CA-Systemen	
<b>Inhalt</b>	Die Vorlesung Computeralgebra II behandelt eines oder mehrere Gebiete aus dem folgenden Themenkatalog: I. Algorithmische Zahlentheorie II. Algorithmische kommutative Algebra III. Algorithmische Gruppentheorie IV. Algorithmische Invariantentheorie V. Algorithmische Arithmetische Geometrie	
<b>Voraussetzungen</b>	empfohlen sind: Algebra I und II (MB1, MB2), Computeralgebra I (MG19)	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.	
<b>Nützliche Literatur</b>	Wird in der Vorlesung bekannt gegeben	

## Codierungstheorie

<b>Code</b> MG21	<b>Name</b> Codierungstheorie	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master, Informatik Master
<b>Lernziel</b>	Grundkenntnisse in Codierungstheorie	
<b>Inhalt</b>	<p>Die Vorlesung Codierungstheorie behandelt theoretische Grundlagen und Algorithmen für fehlerkorrigierende Codes. Hauptthemen sind:</p> <p>I. Elementare Codierungstheorie: Übertragungswahrscheinlichkeiten und Satz von Shannon, lineare Codes und Gewichtspolynom, Reed-Solomon-Codes und MDS-Codes, perfekte Codes und Golay-Codes, zyklische Codes und BCH-Codes, quadratische Reste-Codes, Reed-Muller-Codes und Gruppencodes, Schranken für Codes, klassische Goppa-Codes</p> <p>II. Arithmetische Codes: Geometrische Goppa-Codes, rationale Codes und Symmetrien, elliptische und hyperelliptische Codes, Teilkörpercodes, Decodierung arithmetischer Codes, Hermitesche Codes, Codes in Artin-Schreier-Türmen, asymptotische Schranken für Codes, Satz von Drinfeld-Vladut, Darstellung linearer Codes als arithmetische Codes</p>	
<b>Voraussetzungen</b>	empfohlen sind: Algebra I (MB1), für Teil II: Algebraische Geometrie I (MG3)	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekanntgegeben.	
<b>Nützliche Literatur</b>	<p>W. C. Huffman, V. Pless: Fundamentals of Error-Correcting Codes</p> <p>S. A. Stepanov: Codes on Algebraic Curves</p> <p>H. Stichtenoth: Algebraic Function Fields and Codes</p> <p>B. H. Matzat: Codierungstheorie (Skriptum)</p>	

## Numerische Lineare Algebra

<b>Code</b> MH5	<b>Name</b> Numerische Lineare Algebra	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master, Physik Master
<b>Lernziel</b>	Kenntnis der gebräuchlichen Methoden zur numerischen Lösung von Aufgaben der Linearen Algebra, Analytisches und algorithmisches Denken, Anwendung von Techniken der Analysis und Linearen Algebra	
<b>Inhalt</b>	I. Lineare Gleichungssysteme und Eigenwertaufgaben II. Iterative Verfahren, Fixpunktiterationen III. Krylowraum-Methoden IV. Iterative Verfahren für Eigenwertaufgaben V. Singulärwertzerlegung VI. Anwendungen auf Systemmatrizen bei der Diskretisierung partieller Differentialgleichungen	
<b>Voraussetzungen</b>	empfohlen ist: Einführung in die Numerik	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur (und einer Freischussmöglichkeit), Wiederholungsmöglichkeit mit der Vorlesung in den Folgejahren.	
<b>Nützliche Literatur</b>	Bekanntgabe in der Vorlesung (Vorlesungsskriptum)	

## Numerical methods for partial differential equations

<b>Code</b> MH7	<b>Name</b> Numerical methods for partial differential equations	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> one semester	<b>Turnus</b> annually
<b>Lehrform</b> Lecture 4 SWS, Exercise session 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Master Scientific Computing, Mathematics, Physics; advanced bachelor students
<b>Lernziel</b>	Foundations of finite element methods and their analysis, Ability to use typical analytical techniques from finite element analysis in order to design and analyze discretization schemes	
<b>Inhalt</b>	Introduction to elliptic partial differential equations; construction of the finite element method; a priori error estimates in energy and weaker norms; iterative solvers; multigrid and domain decomposition methods; a posteriori error estimation; adaptive mesh refinement; mixed finite element methods for saddle point problems	
<b>Voraussetzungen</b>	recommended are: Introduction to numerical analysis, Höhere Analysis (Lebesgue integration, Gauß theorem) Participation in the class „Implementation of numerical methods for partial differential equations“ in this semester is recommended, but not required.	
<b>Prüfungs- modalitäten</b>	Solution of homework exercises and a final oral exam. Details will be given by the lecturer at the beginning of the course.	
<b>Nützliche Literatur</b>	Grossmann, Roos(, Stynes): Numerical Treatment of Partial Differential Equations, English edition / deutsche Ausgabe	

## Numerische Optimierung bei Differentialgleichungen

<b>Code</b> MH8	<b>Name</b> Numerische Optimierung bei Differentialgleichungen	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master, Informatik Master
<b>Lernziel</b>	Parameterschätzung sowie optimale nichtlineare Versuchsplanung bei Differentialgleichungen	
<b>Inhalt</b>	Das Modul behandelt Grundlagen und numerische Methoden der optimalen Steuerung	
<b>Voraussetzungen</b>	empfohlen sind: Kenntnisse, wie sie in den Vorlesungen Einführung in die Numerische Mathematik und Numerische Mathematik I vermittelt werden sowie Grundkenntnisse der linearen Algebra und Analysis sowie einer Programmiersprache, Vorlesung Gewöhnliche Differentialgleichungen (MC1)	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen (Erreichen einer Mindestpunktzahl) und Bestehen einer Abschlussprüfung.	
<b>Nützliche Literatur</b>		

## Statistik II

<b>Code</b> MH12	<b>Name</b> Statistik II	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master
<b>Lernziel</b>	Vertiefte Behandlung einer Auswahl statistischer Methoden	
<b>Inhalt</b>	<p>Mögliche Themen sind:</p> <p>I. Multivariate Statistik: Wishart-Verteilung, multipler Korrelationskoeffizient, Hotellings T<sup>2</sup>-Verteilung, Hauptkomponentenanalyse, kanonische Korrelationen, grafische Modelle</p> <p>II. Zeitreihenanalyse: Lineare Filter, ARMA-Modelle, Prädiktion, State-Space Modelle, Spektraldarstellung, Periodogramm, Whittle-Likelihood, nichtlineare Zeitreihenmodelle</p> <p>III. Nichtparametrik: Dichteschätzung und nicht-parametrische Regression, Kernschätzer, lokal polynomiale Schätzer, Orthogonalreihenschätzer, Adaptivität, Risikoabschätzung, nichtparametrische Tests</p>	
<b>Voraussetzungen</b>	empfohlen sind: Wahrscheinlichkeitstheorie I (MC5), Statistik I (MD3) Analysis I, Lineare Algebra I	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter 2-stündiger Klausur. Art und Zeitrahmen einer Wiederholungsprüfung wird vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.	
<b>Nützliche Literatur</b>	<p>Anderson, T. W.: An Introduction to Multivariate Statistical Analysis, John Wiley</p> <p>Jørgensen, Bent: The Theory of Linear Models, Chapman&amp;Hall, New York, 1993.</p> <p>Brockwell, P. J. and Davis R. A.: Time Series: Theory and Methods, Springer-Verlag</p> <p>Wasserman, L.: All of Nonparametric Statistics, Springer-Verlag</p>	

## Wahrscheinlichkeitstheorie II

<b>Code</b> MH13	<b>Name</b> Wahrscheinlichkeitstheorie II	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master
<b>Lernziel</b>	Ausgewählte Themen zu Stochastischen Prozessen und zur Stochastischen Analyse.	
<b>Inhalt</b>	<p>I. Theorie Stochastischer Prozesse: Endlich-dimensionale Verteilungen, Existenzsatz von Kolmogorov, stetige Pfade, Konstruktion und Eigenschaften der Brownschen Bewegung, Gaußprozesse</p> <p>II. Ergodentheorie: Stationäre und ergodische Prozesse, Ergodensätze</p> <p>III. Invarianzprinzipien: Straffheit, schwache Konvergenz im Raum der stetigen Funktionen, Invarianzprinzip von Donsker, Theorie der Empirischen Prozesse</p> <p>IV. Stochastisches Integral: Martingale in stetiger Zeit, Itô-Integral, Itô-Formel</p>	
<b>Voraussetzungen</b>	empfohlen sind: Analysis I, Lineare Algebra I, Wahrscheinlichkeitstheorie I	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben	
<b>Nützliche Literatur</b>	<p>Durrett, S.: Probability: Theory and Examples, Duxbury Press</p> <p>Karlin, S. and Taylor, H.: A First/Second Course in Stochastic Processes, Academic Press</p> <p>Karatzas, I. and Shreve, S.: Brownian Motion and Stochastic Calculus, Springer</p>	

## Berechenbarkeit und Komplexität I

<b>Code</b> MH14	<b>Name</b> Berechenbarkeit und Komplexität I	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master, Informatik Master
<b>Lernziel</b>	Grundkenntnisse über Berechenbarkeit und Komplexität	
<b>Inhalt</b>	Die Berechenbarkeitstheorie liefert den formalen Rahmen, die Lösbarkeit algorithmischer Probleme zu untersuchen, die Komplexitätstheorie stellt Methoden und Konzepte zur Analyse des erforderlichen Aufwands algorithmischer Problemlösungen zur Verfügung. Ziel des Moduls ist es die Studierenden mit den zentralen Konzepten und Methoden der Berechenbarkeits- und der Komplexitätstheorie vertraut zu machen. In der Berechenbarkeitstheorie stehen Methoden zum Nachweis der Unentscheidbarkeit im Mittelpunkt, in der Komplexitätstheorie liegt der Schwerpunkt auf dem Vergleich und der strukturellen Analyse der polynomiell beschränkten Komplexitätsklassen. Insbesondere werden das P-NP- Problem und die NP-Vollständigkeit behandelt.	
<b>Voraussetzungen</b>	empfohlen sind: Grundkenntnisse aus der Theoretischen Informatik	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben	
<b>Nützliche Literatur</b>		



## Berechenbarkeit und Komplexität II

<b>Code</b> MH15	<b>Name</b> Berechenbarkeit und Komplexität II	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 3 SWS, Übung 1 SWS	<b>Arbeitsaufwand</b> 180 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master, Informatik Master
<b>Lernziel</b>	Vertiefte Kenntnisse über Berechenbarkeit und Komplexität	
<b>Inhalt</b>	In diesem Modul werden ausgewählte fortgeschrittene Themen aus dem Bereich der Berechenbarkeits- und Komplexitätstheorie behandelt.	
<b>Voraussetzungen</b>	empfohlen sind: Berechenbarkeit und Komplexität I	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.	
<b>Nützliche Literatur</b>		

## Algorithmische Optimierung I

<b>Code</b> MH16	<b>Name</b> Algorithmische Optimierung I	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master, Informatik Master
<b>Lernziel</b>	Grundkenntnisse über algorithmische Optimierung	
<b>Inhalt</b>	Das Modul behandelt moderne Verfahren der unbeschränkten und beschränkten Optimierung. Die Studierenden werden in die Lage versetzt moderne Verfahren des Gebietes anzuwenden, zu beurteilen und zu entwickeln.	
<b>Voraussetzungen</b>	empfohlen sind: Mathematische Grundvorlesungen MA1-MA8	
<b>Prüfungsmodalitäten</b>	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben.	
<b>Nützliche Literatur</b>	Bekanntgabe in der Vorlesung	

## Algorithmische Optimierung II

<b>Code</b> MH17	<b>Name</b> Algorithmische Optimierung II	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master, Informatik Master
<b>Lernziel</b>	Grundlagen der linearen und ganzzahligen Optimierung	
<b>Inhalt</b>	I. Dualitätstheorie II. Simplexalgorithmus und Varianten III. Innere-Punkte-Verfahren IV. Schnittebenen-Verfahren	
<b>Voraussetzungen</b>	empfohlen ist: Algorithmische Optimierung I (MH16)	
<b>Prüfungs- modalitäten</b>	Lösung von Übungsaufgaben mit benoteter Klausur bzw. mündlicher Prüfung. Art und Zeitrahmen einer Wiederholungsprüfung werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben	
<b>Nützliche Literatur</b>	Bekanntgabe in der Vorlesung	

## Mustererkennung

<b>Code</b> MH18	<b>Name</b> Mustererkennung	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Master, Mathematik Lehramt, Scientific Computing Master, Informatik Master
<b>Lernziel</b>	Mathematische Methoden und algorithmische Verfahren zur überwachten und unüberwachten Klassifikation empirischer Daten. Selbständiges computergestütztes Lösen von Problemen der Statistischen Mustererkennung.	
<b>Inhalt</b>	Euklidische Einbettungen, Multidimensionale Skalierung, Bayes Klassifikator, Fehlerschranken und -abschätzungen, Entscheidungsbäume, Kombination und Performanzsteigerung einfacher Klassifikatoren, Klassifikation mit Kernfunktionen, Gaußsche Prozesse und Klassifikation, Klassifikation mit Mischungsverteilungen, nichtparametrische Klassifikation, Merkmalsauswahl und -extraktion, Ballungsanalyse mit Prototypen, Ähnlichkeitsgraphen und unüberwachtes Lernen.	
<b>Voraussetzungen</b>	empfohlen sind: Mathematische Grundvorlesungen, Wahrscheinlichkeitstheorie I, Numerische Lineare Algebra (MH5), Algorithmische Optimierung I (MH16)	
<b>Prüfungsmodalitäten</b>	Bearbeiten von Übungsblättern und Übungen am Computer, Semesterbegleitende Prüfung, Art und Zeitrahmen werden vom Dozenten festgelegt und zu Beginn der Vorlesung bekannt gegeben	
<b>Nützliche Literatur</b>	Bekanntgabe in der Vorlesung und auf der entsprechenden WWW-Seite.	

## Mathematische Einführung in Compressed Sensing

<b>Code</b> MH19	<b>Name</b> Mathematische Einführung in Compressed Sensing	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> Mathematik Bachelor/Master, Scientific Computing (Wiss. Rechnen) Master, Angewandte Informatik Master
<b>Lernziel</b>	Mathematische Modellierung und computergestütztes Lösen zentraler Probleme der sparsen Rekonstruktion.	
<b>Inhalt</b>	Theorie: Dünnbesetzte (sparse) Rekonstruktion via Minimierung; Grundannahmen: mutual incoherence; nullspace property, restricted isometry property; Rekonstruktion mit Zufallsmatrizen; Phasendiagramme; Grundlagen der konvexen Analysis, der Wahrscheinlichkeitstheorie und der Integralgeometrie. Algorithmen: Orthogonal matching pursuit; Thresholding basierte Verfahren; Primal-duale Verfahren. Anwendungen: Sparse Approximation; Bildverarbeitung (Tomographische Inversion, Entfalten, etc. ); Low-Rank Completion.	
<b>Voraussetzungen</b>	empfohlen sind: Lineare Algebra, Analysis, Umgang mit Matlab. Weitere Kenntnisse (Optimierung, Wahrscheinlichkeitstheorie) wären vorteilhaft, werden aber nicht vorausgesetzt.	
<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer mündlichen Abschlussprüfung	
<b>Nützliche Literatur</b>	S. Foucart, H. Rauhut: A Mathematical Introduction to Compressive Sensing, Springer, 2013 S. Boyd, L. Vandenberghe: Convex Optimization, Cambridge University Press, 2004 M. Ledoux: The Concentration of Measure Phenomenon American Mathematical Society, 2005 R. Schneider, W. Weil: Stochastic and Integral Geometry, Springer, 2008 J.-L. Starck, F. Mutagh, J.M. Fadili: Sparse Image and Signal Processing, Cambridge University Press, 2010	

## Statistische Datenanalyse

<b>Code</b> MH21	<b>Name</b> Statistische Datenanalyse	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 2 SWS und Projekt-Arbeit. Die Vorlesung geht problemorientiert vor: Anhand von konkreten Fallbeispielen und Datensätzen werden unterschiedliche Methoden und Strategien der Analyse diskutiert.	<b>Arbeitsaufwand</b> 240 h	<b>Verwendbarkeit</b> BA Mathematik Bachelor, Mathematik Master, LA Mathematik, Diplom Mathematik, BA/Master Ang. Informatik, Master Scientific Computing
<b>Lernziel</b>	Statistische Modellierung und Modelldiagnostik; grundlegende Methoden und Strategien der Datenanalyse. Konkrete Anwendung statistischer Methoden in der Datenanalyse. Diagnostik und Modellierung. Präsentation und Kommunikation der Diagnostik an Beispielen.	
<b>Inhalt</b>	<p>Datenanalyse ist ein Teil der Statistik, der die klassischen Zugänge ergänzt. Zum einen spielt sie als „explorative Datenanalyse“ bei der Modellbildung eine wesentliche Rolle; zum anderen bietet sie als „Residuenanalyse“ Ansätze, die Gütigkeit von formalen statistischen Ergebnissen im Anwendungsfall datenbezogen zu prüfen. Die Datenanalyse ist eine neuere Entwicklung in der Statistik und geht in großen Teilen auf Arbeiten von J. Tukey zurück. In einzelnen Bereichen, so der Residuenanalyse für lineare Modelle, ist ein weitgehend abgeschlossener Stand erreicht. In anderen Bereichen liegt keine geschlossene Theorie vor. Deshalb muss hier auf Beispiele und Fallstudien zurückgegriffen werden.</p> <p>Themenbereiche:</p> <ul style="list-style-type: none"> <li>- Visualisierung in der Statistik.</li> <li>- Statistik für höherdimensionale Probleme; Dimensionsreduktion.</li> <li>- Multi-Resolutionsanalyse.</li> </ul>	

<b>Voraussetzungen</b>	empfohlen sind: MA8 Einführung in die Wahrscheinlichkeitstheorie und Statistik, MD2 Statistik
<b>Prüfungsmodalitäten</b>	Schriftliche Ausarbeitung einer Fallstudie
<b>Nützliche Literatur</b>	Tukey, J.W.; The Collected Works of John W. Tukey: Vol. III. Philosophy and Principles of Data Analysis : 1949-1964 Vol. IV. Philosophy and Principles of Data analysis : 1965-1986 Vol. V. Graphics : 1965-1985 Aktuelle Literatur, im wesentlichen aus den Zeitschriften „Journal of Computational and Graphical Statistics“ und „Statistics and Computing“

## Implementation of numerical methods for partial differential equations

<b>Code</b> MH27	<b>Name</b> Implementation of numerical methods for partial differential equations	
<b>Leistungspunkte</b> 6 LP	<b>Dauer</b> one semester	<b>Turnus</b> yearly
<b>Lehrform</b> 2 SWS lecture, 2 SWS exercise session	<b>Arbeitsaufwand</b> 180 h	<b>Verwendbarkeit</b> Master Scientific Computing, Mathematics, Computer Science, Physics, advanced bachelor students
<b>Lernziel</b>	Learn to use the software deal.II to numerically solve a wide range of partial differential equations. Ability to modify existing deal.II codes to solve the partial differential equations and to write new deal.II based programs.	
<b>Inhalt</b>	This course serves as an introduction to the use of deal.II with an emphasis on the practical implementation of the finite element methods.	
<b>Voraussetzungen</b>	recommended are: 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 prerequisite.	
<b>Prüfungsmodalitäten</b>	Grade based on assigned tasks including a final project with an oral presentation.	
<b>Nützliche Literatur</b>	The lectures will be based on the available online documentation provided on the webpage <a href="http://www.dealii.org">http://www.dealii.org</a> .	



## 2.5 Module aus dem M.Sc. Scientific Computing

### Fundamentals of Computational Environmental Physics

<b>Code</b> FCEP	<b>Name</b> Fundamentals of Computational Environmental Physics	
<b>Leistungspunkte</b> 8 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 4 SWS, Übung 2 SWS	<b>Arbeitsaufwand</b> 240 h 90 h Präsenzstudium 15 h Prüfungsvorbereitung 135 h Selbststudium und Aufgabenbearbeitung (evtl. in Gruppen)	<b>Verwendbarkeit</b>
<b>Lernziel</b>	Die Studenten <ul style="list-style-type: none"> <li>– Lernen die Modellierung fundamentaler Prozesse der Umweltphysik mit Methoden der Kontinuumsmechanik</li> <li>– Lernen die Simulation dieser Modelle mit modernen Methoden der numerischen Mathematik kennen</li> <li>– Wenden diese Methoden auf konkrete umweltphysikalische Fragestellungen an</li> </ul>	
<b>Inhalt</b>	<p>Elementare, lineare Modelle</p> <ol style="list-style-type: none"> <li>1) Einphasen-Strömung in porösen Medien / Elliptische partielle Differentialgleichungen (PDGLn)</li> <li>2) Skalarer Transport / Hyperbolische Gleichung erster Ordnung</li> <li>3) Wärmeleitung / parabolische PDGLn</li> <li>4) Wellenausbreitung / Hyperbolische PDGLn zweiter Ordnung</li> </ol> <p>Nichtlineare Modelle:</p> <ol style="list-style-type: none"> <li>5) Gekoppelte elementare Modelle</li> <li>6) Strömungsmechanik / Stokes- und Navier-Stokes-Gleichung</li> </ol>	
<b>Voraussetzungen</b>	empfohlen sind: Einführung in die Praktische Informatik (IPI), Programmierkurs (IPK), Einführung in die Numerik, Numerik	

<b>Prüfungsmodalitäten</b>	Erfolgreiche Teilnahme an den Übungen (mehr als 50% der Punkte müssen erreicht werden) und Bestehen einer schriftlichen Abschlussprüfung
<b>Nützliche Literatur</b>	

Entnommen aus den Modulbeschreibungen MA Mathematik und MA Scientific Computing (Wissenschaftliches Rechnen) vom 11.07.2013, ins Deutsche übersetzt

## 2.6 Module aus dem M.Sc. Technische Informatik

Alle fachlichen Module (nicht Fachübergreifende Kompetenzen) aus dem Master Technische Informatik können – entsprechend der inhaltlichen Voraussetzungen – auch im Master Angewandte Informatik belegt werden. Das Angebot findet sich im jeweils gültigen Modulhandbuch des Masters Technische Informatik.

## 2.7 Module aus dem B.Sc./M.Sc. Physik

### Physics of Imaging

<b>Code</b> MWInf5	<b>Name</b> Physics of Imaging	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b>	<b>Turnus</b>
<b>Lehrform</b> Lecture 4 SWS	<b>Arbeitsaufwand</b> 120 h	<b>Verwendbarkeit</b>
<b>Lernziel</b>	Basics of the Physics of Imaging; common principles and techniques of imaging for atomic to astronomical scales.	
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>– Projective geometry, optics, wave optics, Fourier optics and lens aberrations</li> <li>– Radiometry of imaging</li> <li>– Methods of imaging: scanning electron microscopy, X-ray, EDX, FLIM, FRET, fluorescence imaging, near-field imaging</li> <li>– CCD and CMOS technology</li> <li>– Holography, ultrasound imaging, CT-computer tomography, magnetic resonance imaging</li> <li>– Satellite imaging, synthetic aperture radar, radio astronomy</li> </ul>	
<b>Voraussetzungen</b>	recommended are: UKInf2, PEP1 - PEP4	
<b>Prüfungsmodalitäten</b>	Defined by lecturer before beginning of course	
<b>Nützliche Literatur</b>		

Entnommen aus dem Modulhandbuch MScPhysik, Version 2012.2/v3.

## Image Processing

<b>Code</b> MWInf6	<b>Name</b> Image Processing	
<b>Leistungspunkte</b> 7 LP	<b>Dauer</b>	<b>Turnus</b>
<b>Lehrform</b> Lecture 4 SWS, Exercises 2 SWS	<b>Arbeitsaufwand</b> 210 h	<b>Verwendbarkeit</b>
<b>Lernziel</b>	Learn how to analyze signals from time series, images, and any kind of multidimensional signals and to apply it to problems in natural sciences, life sciences and technology.	
<b>Inhalt</b>	<ul style="list-style-type: none"> <li>- Continuous and discrete signals, sampling theorem, signal representation</li> <li>- Fourier transform</li> <li>- Random variables and fields, probability density functions, error propagation</li> <li>- Homogeneous and inhomogeneous point operations</li> <li>- Neighbourhood operations, linear and nonlinear filters, linear system theory</li> <li>- Geometric transformations and interpolation</li> <li>- Multi-grid signal presentation and processing</li> <li>- Averaging, edge and line detection, local structure analysis, local phase and wave numbers</li> <li>- Motion analysis in image sequences</li> <li>- Segmentation</li> <li>- Regression, globally optimal signal analysis, variation approaches, steerable and nonlinear filtering, inverse filtering</li> <li>- Morphology and shape analysis, moments, Fourier descriptors</li> <li>- Bayesian image restoration</li> <li>- Object detection and recognition</li> </ul>	
<b>Teilnahme-Voraussetzungen</b>		
<b>Voraussetzungen</b>	recommended are: UkInf1	
<b>Prüfungs-modalitäten</b>	Defined by lecturer before beginning of course	
<b>Nützliche Literatur</b>	B. Jähne, Digital Image Processing, 6th edition, Springer	

Entnommen aus dem Modulhandbuch MScPhysik, Version 2012.2/v3

## Introduction to Image Processing on the GPU

<b>Code</b> IGPU	<b>Name</b> Introduction to Image Processing on the GPU	
<b>Leistungspunkte</b> 2 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Four day block course, equivalent of 2 SWS	<b>Arbeitsaufwand</b> 60 h, thereof 30 h Lectures and lab exercises 30 h Preparation and home exercises	<b>Verwendbarkeit</b>
<b>Lernziel</b>	<p>For certain computation tasks which are colloquially called „embarrassingly parallel“, and which occur quite frequently for example in image analysis, an implementation on a GPU (i.e. graphics card) can be orders of magnitude faster than a similar CPU implementation.</p> <p>In this short course, the students learn:</p> <ul style="list-style-type: none"> <li>- what kind of code can and can not be parallelized on a graphics processing unit (GPU)</li> <li>- the structure of a GPU, the different types of memory and how and when to use them</li> <li>- how to program the GPU using nVidia CUDA</li> <li>- how to perform various image analysis tasks using the GPU</li> <li>- ways to optimize the speed of GPU code</li> </ul>	
<b>Inhalt</b>	<p>The four days will be broken up into theoretical and practical segments. In the (short) theoretical lectures, I will introduce the concepts necessary for GPU programming and basic image analysis, which you will immediately try out afterwards in lab sessions. This way, the course will be highly practical and interactive. Students will work in small groups, each of which will have the task to write a small program solving an image analysis problem of their choice. Note that while image analysis tasks are a focus of the course because they give immediately visible results, the acquired techniques can be of course be employed in other fields as well, e.g. for solving large linear algebra problems or PDEs.</p>	
<b>Teilnahme-Voraussetzungen</b>	none	
<b>Voraussetzungen</b>	<p>recommended are: Solid knowledge of C programming, in particular no fear of pointers and direct memory access Some basics of image processing are helpful, but not strictly necessary</p>	

<b>Prüfungs- modalitäten</b>	Active participation in the sessions and completion of the assigned program.
<b>Nützliche Literatur</b>	e.g. nVidia CUDA Programming Guide, available in the „CUDA zone“ on <a href="http://www.nvidia.com">www.nvidia.com</a>

## Fast Parallel Implementations of Image Labeling Problems

<b>Code</b> IMLP	<b>Name</b> Fast Parallel Implementations of Image Labeling Problems	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> 2 SWS Lecture, 1 SWS Exercise	<b>Arbeitsaufwand</b> 120 h; thereof 45 h Lectures and lab exercises 60 h Revision and home exercises 15 h Exam preparation	<b>Verwendbarkeit</b>
<b>Lernziel</b>	<p>Image labeling problems are a fundamental class of problems appearing in image analysis, which dominate tasks in low-level computer vision like depth and motion estimation. Recently, many algorithms have been developed to solve this kind of problems in a variational framework, which allows for fast parallel implementations on the GPU.</p> <p>In this short course, the students learn:</p> <ul style="list-style-type: none"> <li>- Theoretical background for solving labeling problems in a variational framework</li> <li>- Efficient algorithms to solve the related class of optimization problems with parallel algorithms, which can be implemented on the GPU</li> <li>- How to implement these algorithms using nVidia CUDA</li> <li>- Techniques and tricks to make the implementations efficient</li> </ul>	
<b>Inhalt</b>	<p>The lecture is aimed at students who either participated in my CUDA course at the beginning of the semester and want to learn more about image analysis and the theoretical background, or participated in a theoretical course on variational image analysis and want to learn more about state-of-the art labeling algorithms and the practical side of their implementation.</p> <p>If you are new to both topics, you might still take part in the course, but must be prepared to take (potentially a lot of) additional time learning the prerequisites.</p>	
<b>Teilnahme-Voraussetzungen</b>	none	
<b>Voraussetzungen</b>	recommended are: either „Introduction to Image Processing on the GPU“ offered at the beginning of the semester, or an introduction to Variational Image Analysis from e.g. last semester.	



<b>Prüfungsmodalitäten</b>	Active participation in lecture and exercises, oral exam.
<b>Nützliche Literatur</b>	e.g. Chambolle et al. 2010 „An Introduction to Total Variation for Image Analysis“ for theoretical background, and nVidia CUDA Programming Guide, available in the „CUDA zone“ on <a href="http://www.nvidia.com">www.nvidia.com</a> , for the practical one.

## 2.8 Module aus dem B.S./M.Sc. Biologie

### Model-based segmentation of biomedical images (Modellbasierte Segmentierung Biomedizinischer Bilder)

<b>Code</b> IMSBI	<b>Name</b> Model-based segmentation of biomedical images (Modellbasierte Segmentierung Biomedizinischer Bilder)	
<b>Leistungspunkte</b> 4 LP	<b>Dauer</b> ein Semester	<b>Turnus</b>
<b>Lehrform</b> Vorlesung 2 SWS	<b>Arbeitsaufwand</b> 120h insgesamt, davon 30h Präsenzstudium 30h Prüfungsvorbereitung 60h Selbststudium und Aufgabenbearbeitung	<b>Verwendbarkeit</b> B.Sc. Angewandte Informatik, M.Sc. Angewandte Informatik
<b>Lernziel</b>	Ziel der Vorlesung ist es, einen Überblick über 2D und 3D deformierbare Modelle und deren mathematische Grundlagen für die modellbasierte Segmentierung biomedizinischer Bilder zu geben.	
<b>Inhalt</b>	<p>In der Vorlesung werden drei wesentliche Klassen von deformierbaren Modellen (aktive Konturen, statistische Formmodelle, analytische parametrische Modelle) sowie starre Template Modelle behandelt. Die verschiedenen Segmentierungsverfahren werden an aktuellen Beispielen aus der 2D und 3D biomedizinischen Bildverarbeitung veranschaulicht. Geplante Themen sind unter anderem:</p> <ul style="list-style-type: none"> <li>- Introduction and overview of image analysis, image segmentation, and biomedical images</li> <li>- Template models, linear transformations, Hough transform</li> <li>- Active contour models: snakes, level sets, and geodesic active contours</li> <li>- Statistical shape models: active shape models, active appearance models, and extensions</li> <li>- Shape correspondence estimation: landmark-based, intensity-based, and hybrid</li> <li>- Analytic parametric models: contour-based and intensity-based</li> <li>- Performance evaluation and applications</li> </ul>	

<b>Voraussetzungen</b>	empfohlen sind: Grundkenntnisse Signal- oder Bildverarbeitung (z.B. Vorlesung Bildverarbeitung oder Bioinformatik)
<b>Prüfungsmodalitäten</b>	Bestehen einer Abschlussprüfung
<b>Nützliche Literatur</b>	Bekanntgabe in der Vorlesung