Lehrveranstaltungen im WS 2019/20

Info-Veranstaltung

Mittwoch, den 26. Juni 2019
14.15 bis ca. 15.45 Uhr
HörsaalC01

Institut für Informatik

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

JG|U

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

JG|U

Modellierung mit Differentialgleichungen

(Gewöhnliche Differentialgleichungen, 3V+2Ü)

Peter Spichtinger

Institut für Physik der Atmosphäre, JGU Mainz

26. Juni 2019

3 Säulen der Naturwissenschaften

- Theorie/Modell
- Experiment
- (Numerische) Simulation

3 Säulen der Naturwissenschaften

- Theorie/Modell
- Experiment
- (Numerische) Simulation

Differentialgleichungen zur Modellbildung

Viele physikalische Prozesse können mit Differentialgleichungen beschrieben werden.

Einfache Beispiele:

- Schwingendes Pendel

$$\frac{\mathrm{d}^2 x}{\mathrm{d}t^2} + \kappa \frac{\mathrm{d}x}{\mathrm{d}t} + \omega x = 0 \tag{1}$$

- Wärmeleitung in einem Stab

$$\frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial x^2} \tag{2}$$

Spezialisierung auf gewöhnliche Differentialgleichungen (ODEs)

Themen

- Nutzen und Grenzen von Modellen
- Qualitative Theorie und Analyse von ODEs
- Asymptotische Analyse
- Numerische Methoden und Simulationen

Organisatorisches

Vorlesungstermin Montag/Dienstag 12:30 -14:00 Uhr

Geeignet für Studierende Ende Bachelor oder Master

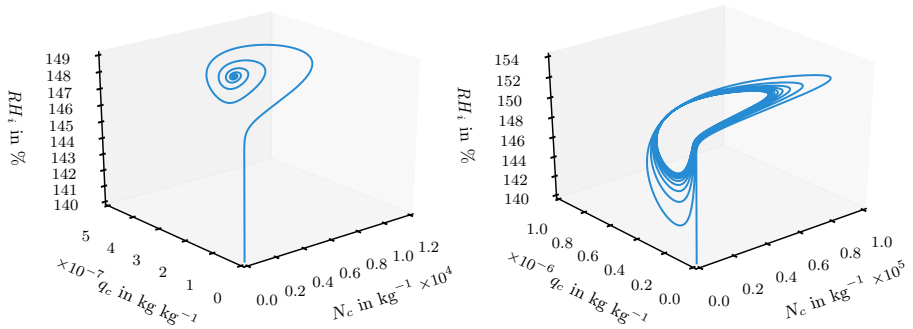
bei weiterem Interesse Fortsetzung im Sommersemester durch Modellierung mit partiellen Differentialgleichungen

Viele Anwendungsbeispiele aus Physik/Chemie/Biologie

- Räuber-Beute-Modelle
- Wachstum von Populationen
- Chemische Reaktionen
- Einfache Atmosphärenmodelle

Analysetechniken aus Theorie dynamischer Systeme

- Fixpunkte
- Grenzzyklen
- Übergang zum Chaos



Approximationstechniken mit Asymptotik

- Reguläre/singuläre Störungstheorie
- Mehrskalenasymptotik
- Matched asymptotics

Numerische Methoden

- Explizite/implizite Verfahren
- Stabilität etc.

Nichtlinearer Oszillator (Schwingkreis)

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \mu(x^2 - 1)\frac{\mathrm{d}x}{\mathrm{d}t} + x = 0 \tag{3}$$

Räuber-Beute-Modell

$$\frac{\mathrm{d}x}{\mathrm{d}t} = ax - bxy \tag{4}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = -cy + bxy \tag{5}$$

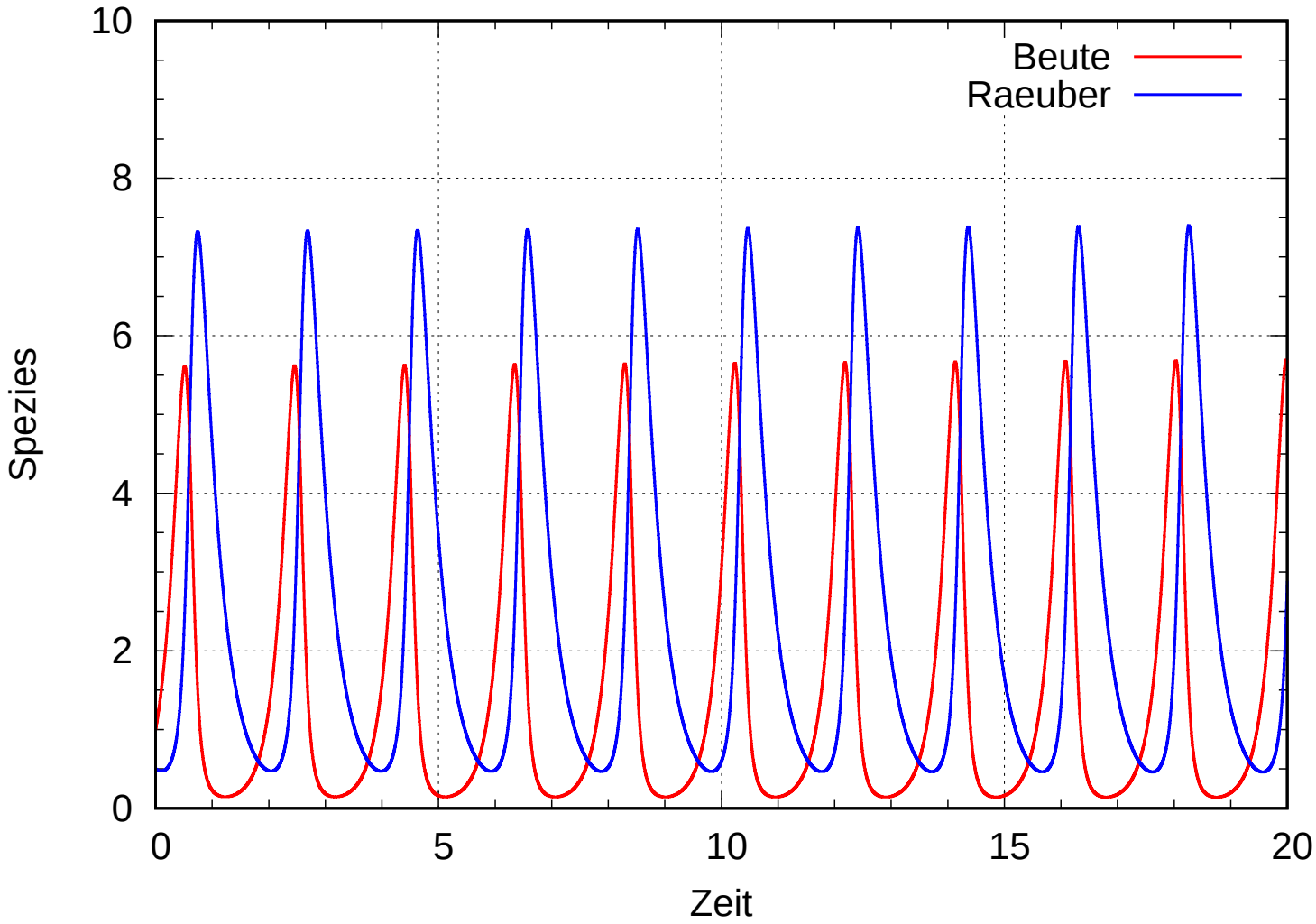
Lorenz-System (chaotisches Atmosphärenmodell)

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y - x) \tag{6}$$

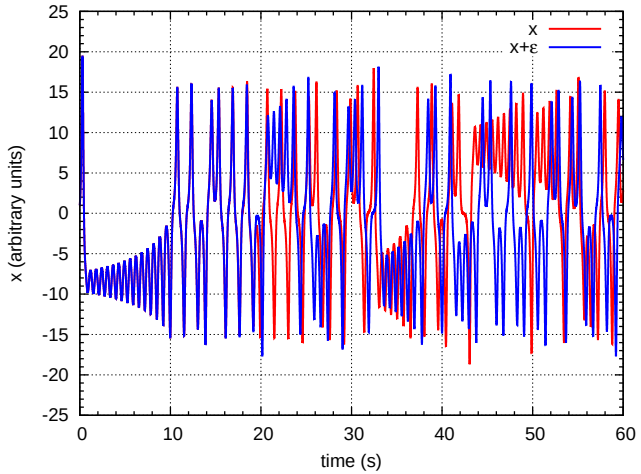
$$\frac{\mathrm{d}y}{\mathrm{d}t} = (r - z)x - y \tag{7}$$

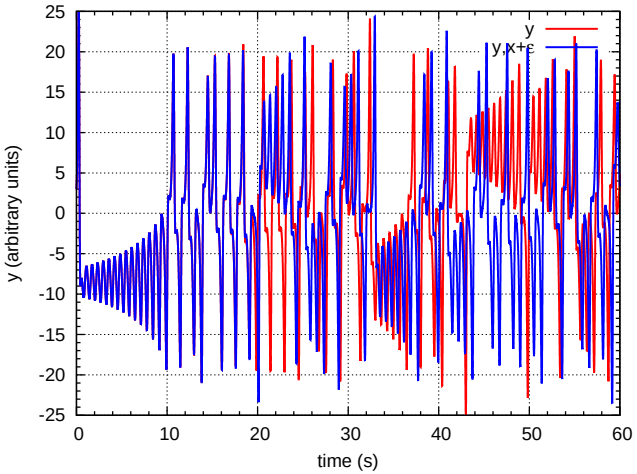
$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - bz \tag{8}$$

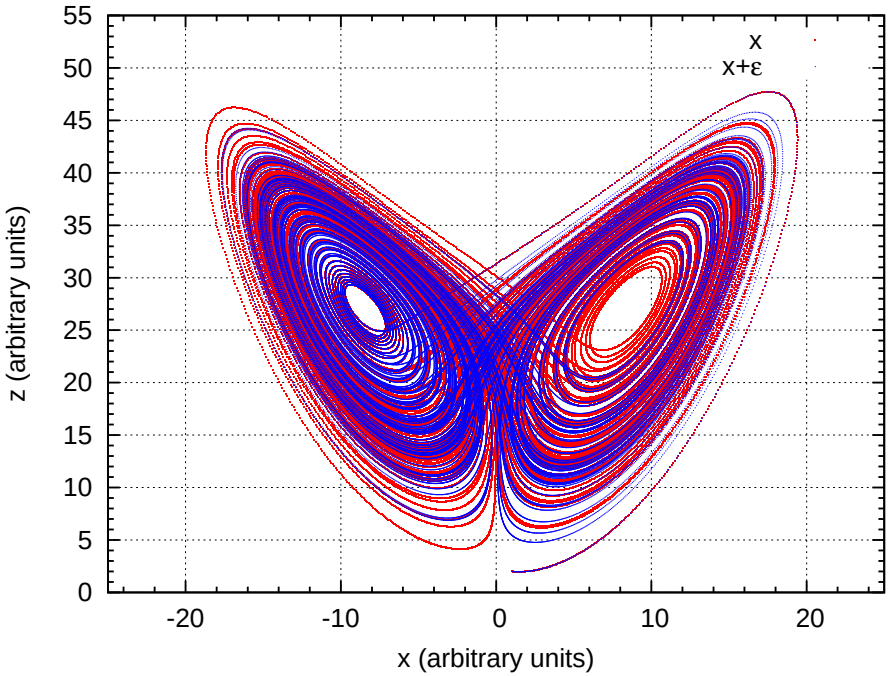
Beispiele - Räuber-Beute-System



Beispiele - Lorenz-System



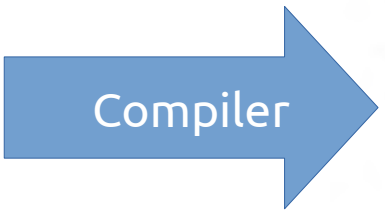




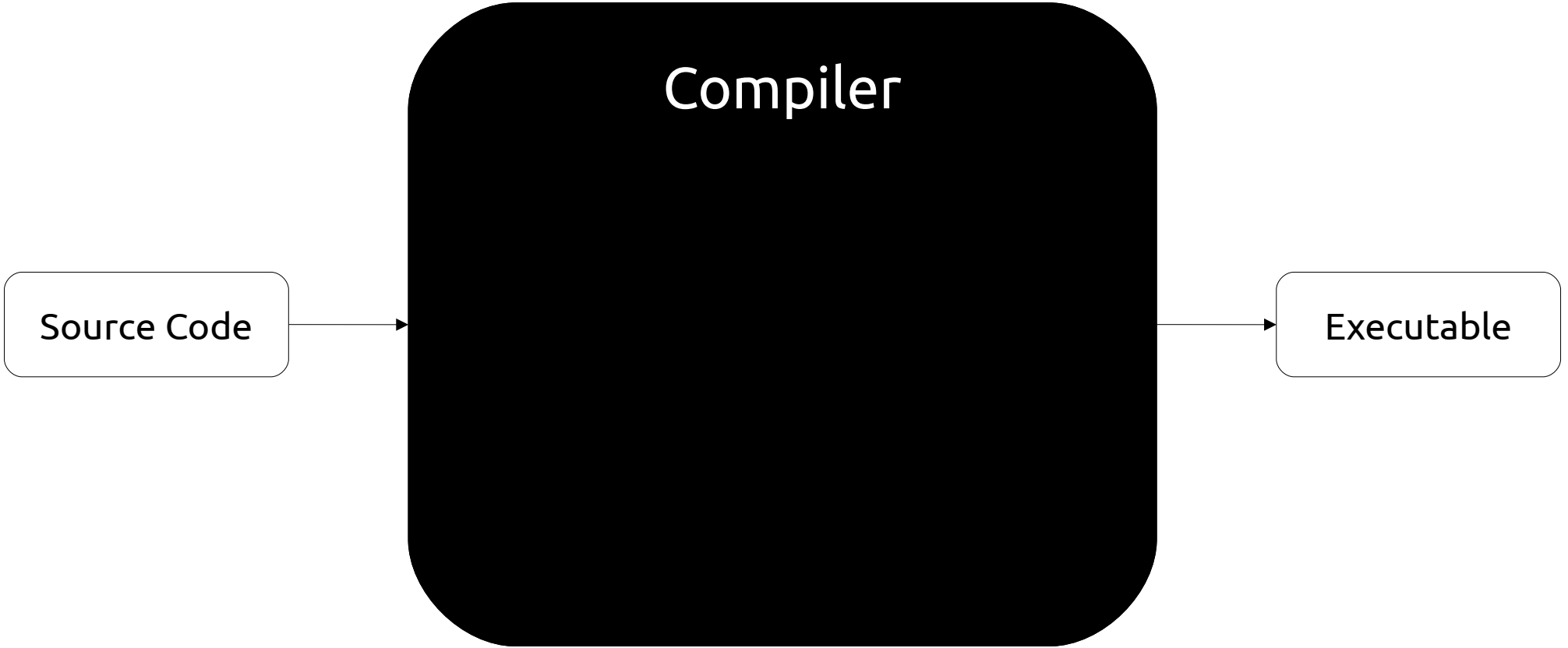
Vorlesung und Praktikum Sprach- und Compilerbau

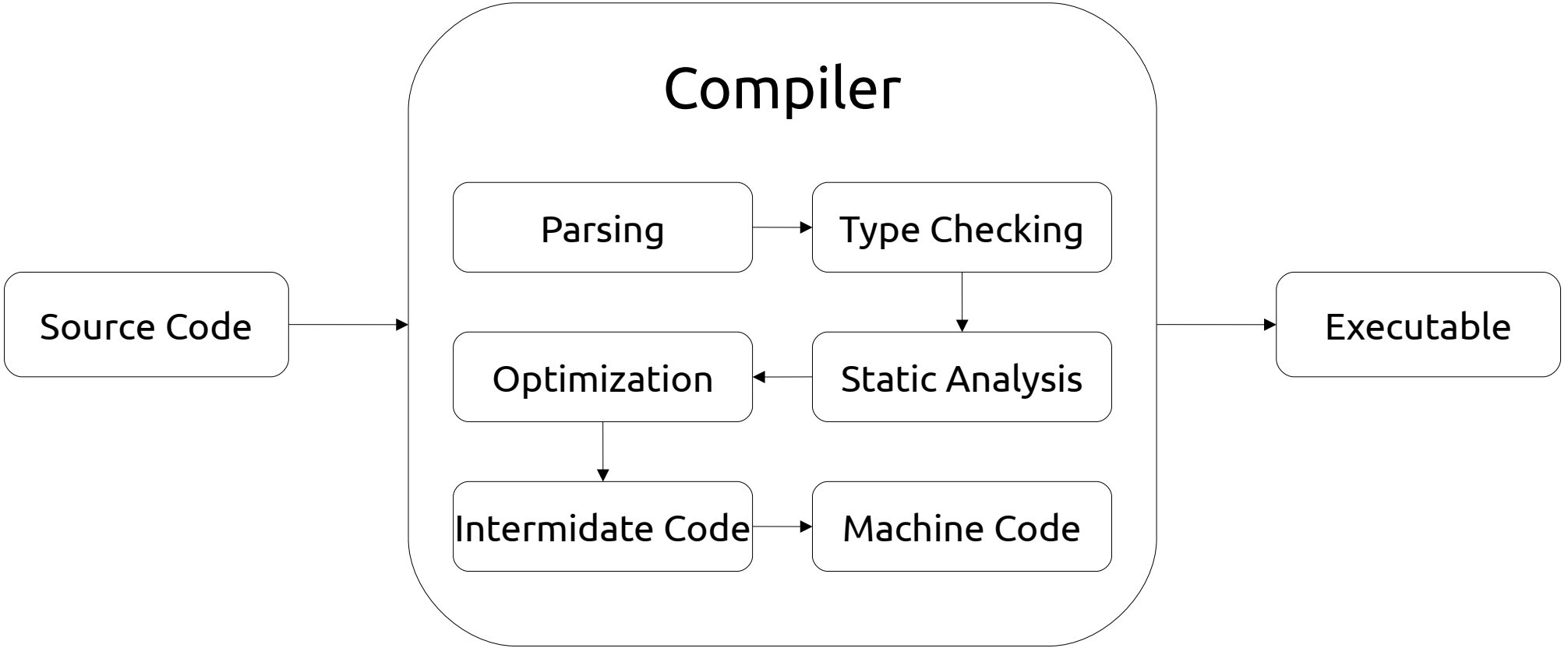
```java
public int cipher(ByteBuffer input, ByteBuffer output) {
    // if GCM-DECRYPT, we have to handle the buffered input
    // and the retrieve the trailing tag from input
    byte[] inputBytes = new byte[input.remaining()];
    input.get(inputBytes, 0, inputBytes.length);
    inBuffer.write(inputBytes, 0, inputBytes.length);
    byte[] inputFinal = inBuffer.toByteArray();
    inBuffer.reset();

    len = OpenSslNative.updateByteArrayByteBuffer(context,
            output, output.position(), output.remaining());

    // retrieve tag
    tag.put(inputFinal, inputFinal.length - getTagLen());
    tag.flip();
}
```

```java
public int cipher(ByteBuffer input, ByteBuffer output) {
    // if GCM-DECRYPT, we have to handle the buffered input
    // and the retrieve the trailing tag from input
    byte[] inputBytes = new byte[input.remaining()];
    input.get(inputBytes, 0, inputBytes.length);
    inBuffer.write(inputBytes, 0, inputBytes.length);
    byte[] inputFinal = inBuffer.toByteArray();
    inBuffer.reset();

    len = OpenSslNative.updateByteArrayByteBuffer(context,
            output, output.position(), output.remaining());

    // retrieve tag
    tag.put(inputFinal, inputFinal.length - getTagLen());
    tag.flip();
}
```




```
cipher(II)I
  ALOAD 0
  GETFIELD Main.input
  ILOAD 1
  ILOAD 2
  IADD
  ICONST_2
  IDIV
  IALOAD
  ISTORE 5
  ILOAD 1
  ICONST_1
  ISUB
  ISTORE 3
  ILOAD 2
  ICONST_1
  IADD
  ISTORE 4
```


Compiler
Source Code
Executable


Compiler
Source Code
Parsing
Type Checking
Static Analysis
Optimization
Intermidate Code
Machine Code
Executable

Domain-Specific Languages

DecisionSample

```
    Money discount;
    discount = create(createPerson());

    if (discount > 400 USD || discount >= 350 EUR) {
      discount = 300 EUR;
    }

    System.out.println("Your name: " + createPerson());
    System.out.println("Your discount: " + discount);

  }

  public Money create(map<string, Object> person) {
    return Money Default: 0 EUR
```

	isLevel_1(person)	*isLevel_2*(person)
isChild(person)	500 EUR	1000 EUR
isAdult(person)	50 EUR + this.seasonalBonus()	100 EUR + this.seasonalBonus()
isRetired(person)	200 EUR	250 EUR + (person["name"] == "Susan" ? this.seasonalBonus() : 0 EUR)

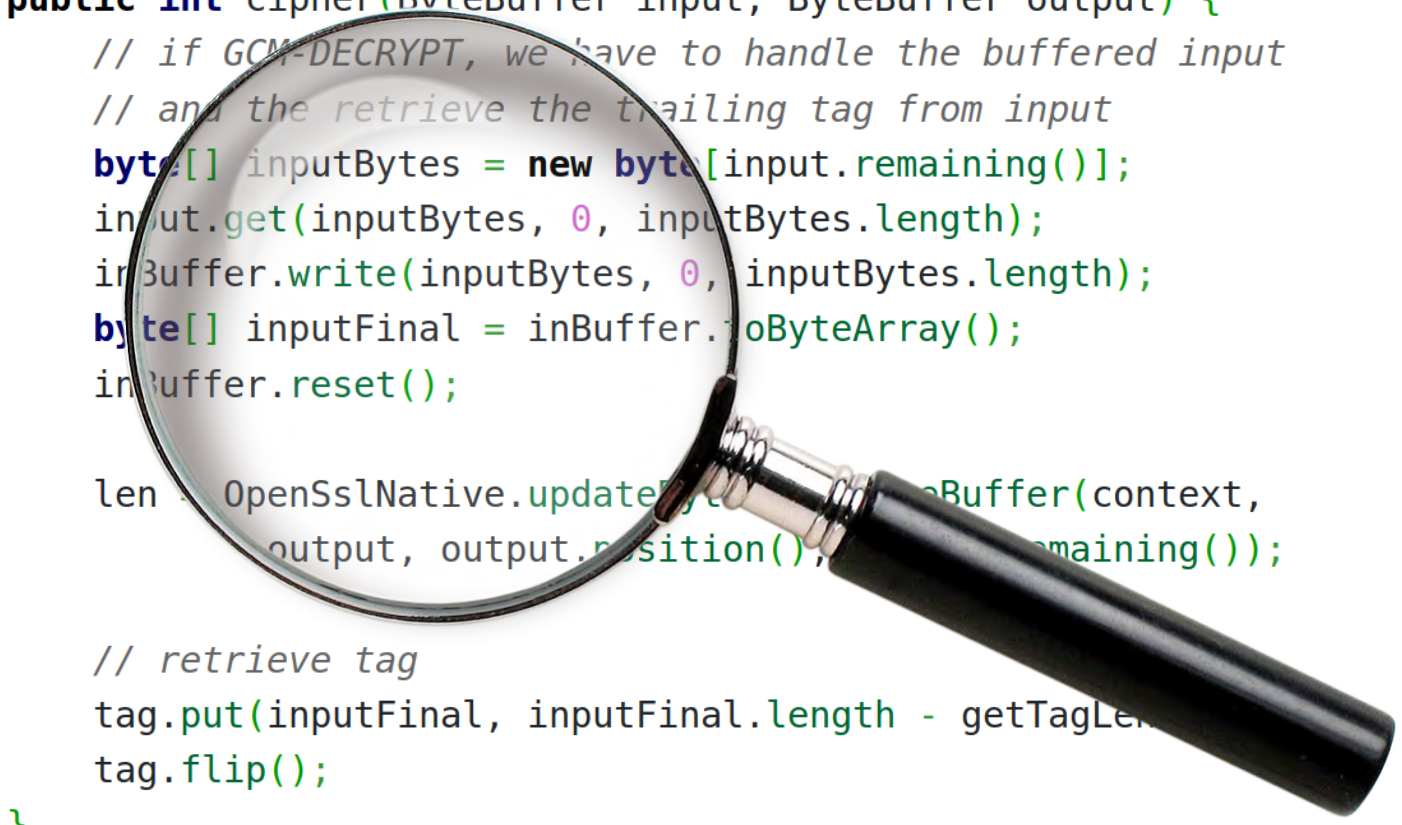
;

```
  }

  private Money seasonalBonus() {
    return 100 EUR;
  }
```

```
System.out.println(String.valueOf((∑ [[1  k  0]
                                      [0 1.0 0]
                                      [0  0  1]])));
System.out.println(exp(a + i * b) - exp(a) * (cos(b) + i * sin(b)));
```

$$\texttt{matrix<Double> s} = \begin{bmatrix} \begin{bmatrix} 3.0 \\ 3^2 \\ 0 \\ 4 \end{bmatrix} & \begin{bmatrix} \sin(1) \\ 1 \\ 2^{7 - \frac{1.0}{2} + 1} \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 3 + \frac{1.0}{2} \\ \exp(1) \\ 0 \end{bmatrix} & \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \end{bmatrix} \end{bmatrix};$$

Seminar Programanalyse



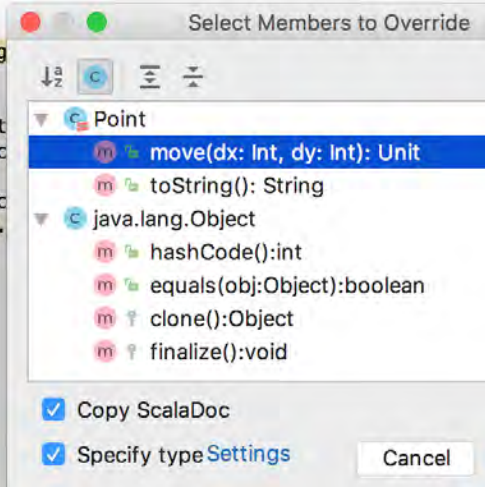
```
x = 1
while(x < 10) {
  y = x * a + b
  x = x + 1
}
```

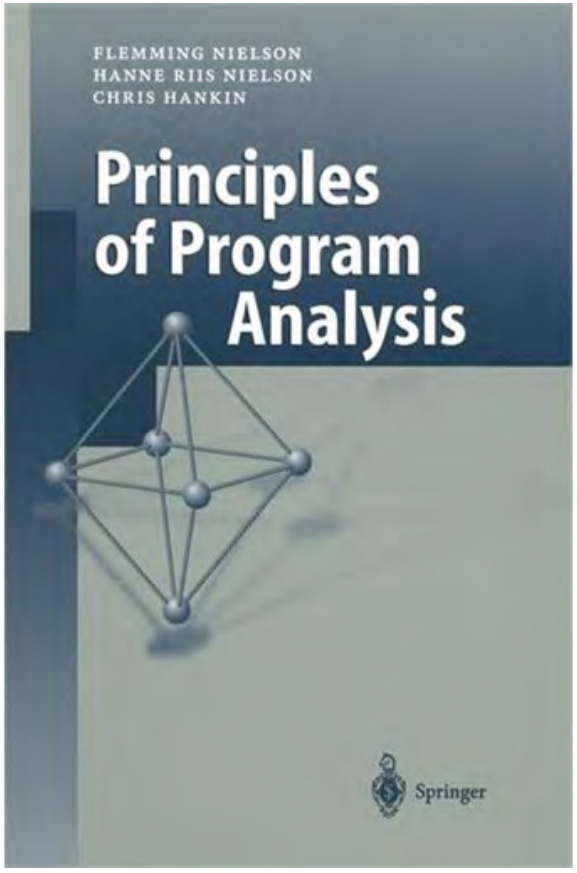
```
x = 1
c = a + b
while(x < 10) {
  y = x * c
  x = x + 1
}
```

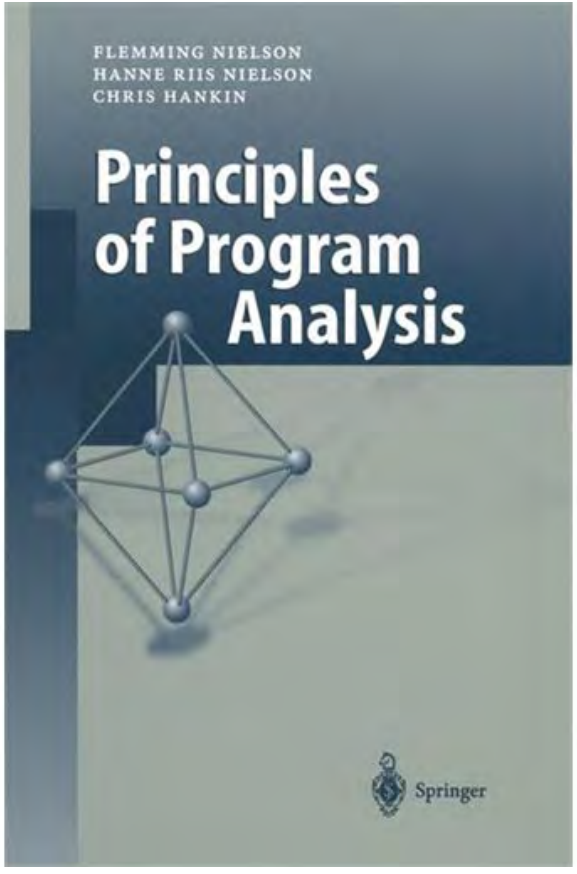
```
x = 1
while(x < 10) {
  y = x * a + b
  x = x + 1
}
```

```
x = 1
c = a + b
while(x < 10) {
  y = x * c
  x = x + 1
}
```

```
class Point(xc: Int, yc: Int) {
  var x: Int = xc
  var y: Int = yc
  def move(dx: Int, dy: Int) : Unit = {
    x = x + dx
    y = y + dy
  }
  override def toString
}

class colorPoint(u: Int
  val color: String = c

  def compareWith(pt: c
    (pt.x == x) && (pt.
}
```




FLEMMING NIELSON
HANNE RIIS NIELSON
CHRIS HANKIN
Principles of Program Analysis
Springer




```
matchTermAgainstConstructor matchSubterms = proc (c,ps,
  case (c,ps,s) of
    (_,_,Bottom) -> bottom -< ()
    ("Cons",[_,_],_) | isLi
        ss <- matchSubterms -< (ps,[getListElem t, t])
        cons -< ("Cons",ss))
        <⊔>
        (throw -< ())
    ("Cons",_,_) -> typeMismatch -< ("List",show s)
    ("Nil",[],_) | isList t -> (returnA -< Term (List B
    ("Nil",_,_) -> throw -< ()
    ("",_,Tuple ss)
      | eqLength ss ps -> do
        ss' <- matchSubterms -< (ps,sortsToTerms ss ctx
        cons -< (c,ss')
      | otherwise -> throw -< ()
    ("",_,Top) -> (do
        ss <- matchSubterms -< (ps,[Term Top ctx | _ <
        cons -< ("",ss))
        <⊔>
        (throw -< ())
    (_,_,Top) -> do
      (| joinList (typeError -< printf "cannot find co
```

Vorlesung und Praktikum Sprach- und Compilerbau

Seminar Programanalyse

```
public int cipher(ByteBuffer input, ByteBuffer output) {
    // if GCM-DECRYPT, we have to handle the buffered input
    // and the retrieve the trailing tag from input
    byte[] inputBytes = new byte[input.remaining()];
    input.get(inputBytes, 0, inputBytes.length);
    inBuffer.write(inputBytes, 0, inputBytes.length);
    byte[] inputFinal = inBuffer.toByteArray();
    inBuffer.reset();

    len = OpenSslNative.updateByteBuffer(context,
            output, output.position(), output.remaining());

    // retrieve tag
    tag.put(inputFinal, inputFinal.length - getTagLen
    tag.flip();
}
```

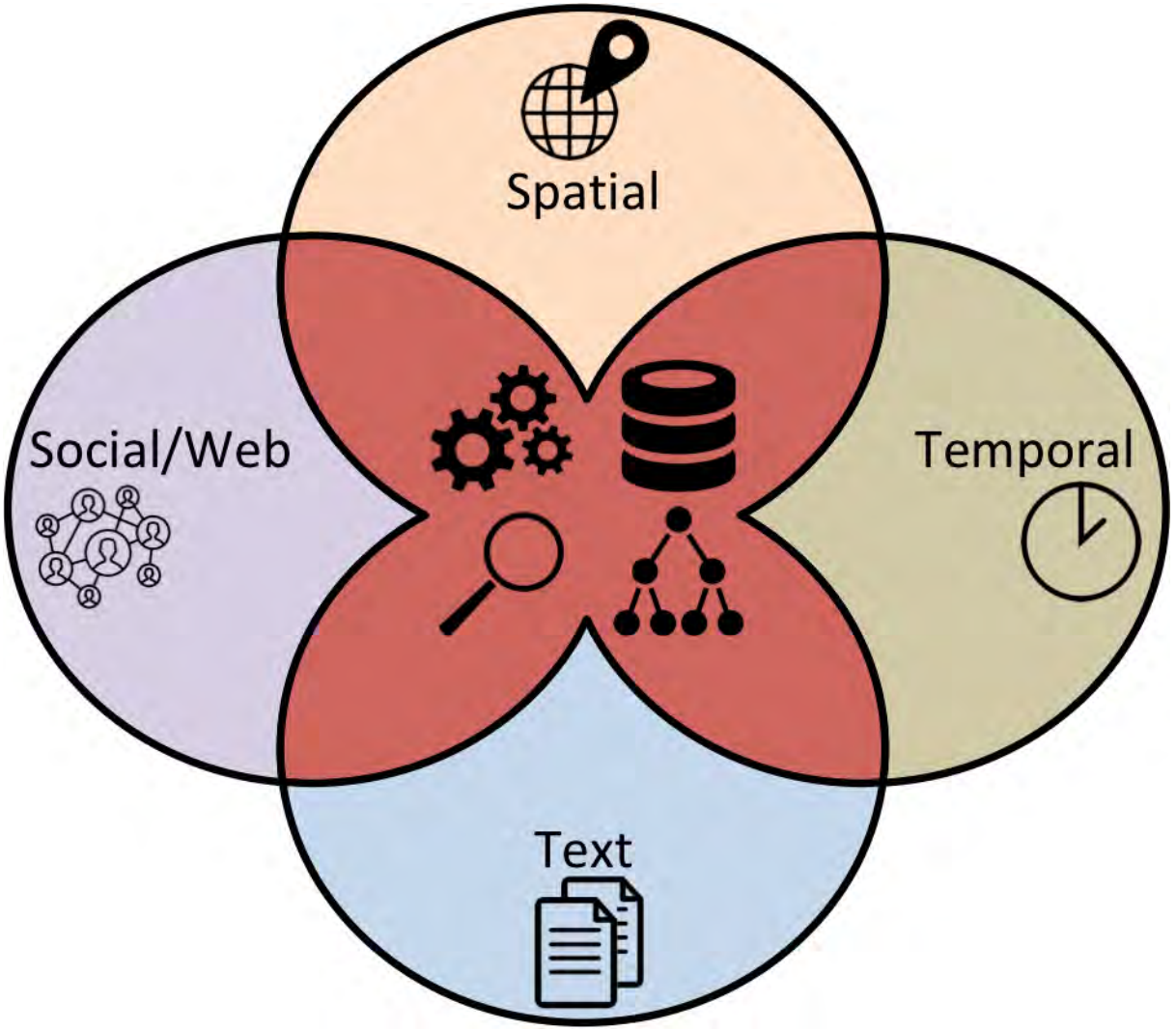
AG Data Management

Storing

Indexing

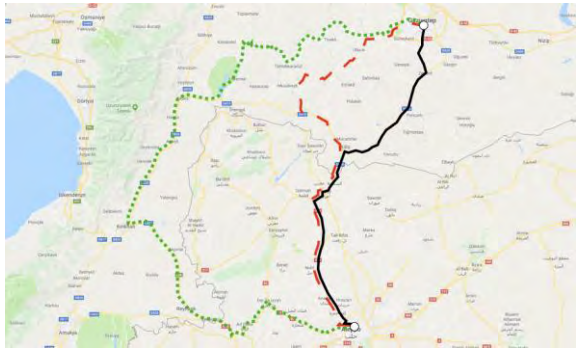
Processing

Querying



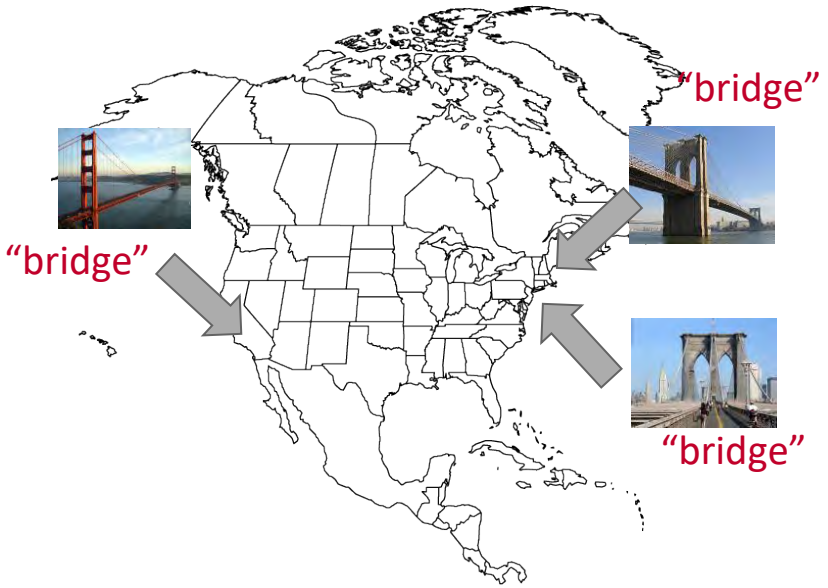
AG Data Management

Querying graphs

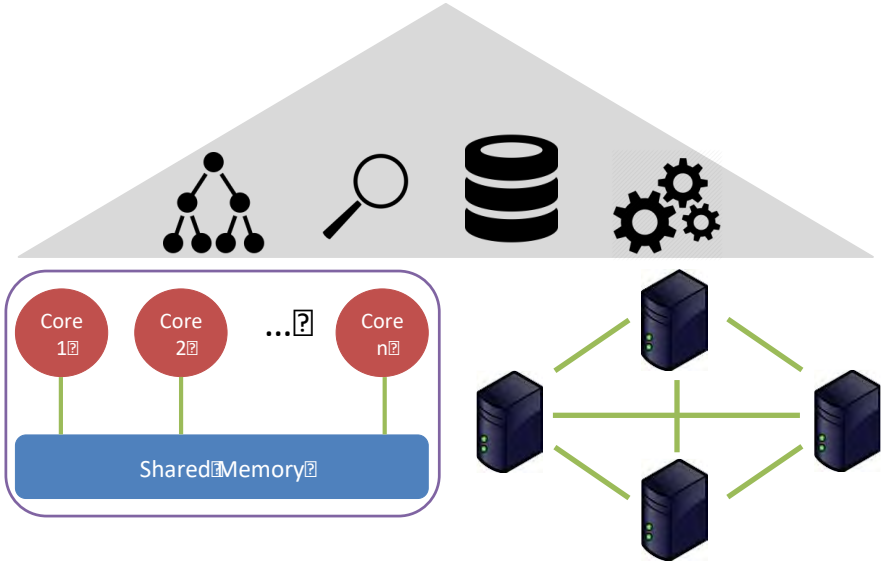


Location-based Social Networks

Data de-duplication



Modern Hardware and the Cloud



Datenbanken Praktikum

- Duration: two weeks
- Dates: 31.09 – 11.10, TBD
- Language: English
- Description:
  - Hands-on experience, practical aspects
  - Building a functional and practical UI on top of a DBMS
  - Web programming, Java, C++

Datenbanken II

- Lectures
  - Tuesdays 10:00 – 12:00, Rm 03-428
- Exercise sessions
  - Deepen your understanding, gain practical experience
  - Weekly exercises (handins assignments)
  - Appointments to be determined
- Language
  - English
- Level of studies
  - B. Sc. / M. Ed.

Datenbanken II

- Description
  - Advanced topics in data management
  - Multi-dimensional data
    - Data warehouses
    - Indexing
    - Ranking
  - Complex data management
    - Geographical - spatial
    - Text
    - Social
  - NoSQL/NewSQL

Lehrveranstaltungen der theoretischen Informatik

- VL: Komplexit¨atstheorie (VL: Di 10-12)
- VL: Formale Sprachen und Berechenbarkeit
  (Friederike Schmid und Markus Blumenstock, VL: Mi 10-12)
- VL: Graphalgorithmen, VL: Mo. 10-12
- Seminar: Algorithmen
- Praktikum: Approaching Programming Contests (Domenico Mosca)

Graphalgorithmen

Inhalt:

- Matchings
- Lowest Common Ancestors
- Randomisierte Graphalgorithmen
- Parametrisierte Algorithmen
- Planare Graphen

Formales:

- Vorlesung: Mi 14-16
- Übung: tba
- Organisation der Übung: Sarah Ziegler

Seminar: Algorithmen

- Zeit und Ort: Blockseminar im März (genauer Termin wird per doodle bestimmt)
- Vorbesprechung: Anfang Oktober (genauer Termin wird noch bekannt gegeben)
- Voraussetzungen: DSeA oder äquivalent
- Scheinvergabe:
  - Vortrag von ca. 30min
  - Ausarbeitung in LaTeX
- Inhalte:
  - Vorstellung eines Algorithmus ähnlich denen aus DSeA
  - Genauere Themenvorschläge gibt es in der Vorbesprechung

Approaching Programming Contests (Praktikum)

- Zeit und Ort: wird mit den Teilnehmern vereinbart
- Betreuung: Domenico Mosca
- Voraussetzungen: EiP, EiS, DSEA

Inhalt:

- Lösen von Problemen, wie sie typischerweise in Programmierwettbewerben auftreten
- Eingesetzte Algorithmen: z. B. Breitensuche, Berechnen von minimalen Spannbäumen, Max-Flow-Algorithmen, dynamische Programmierung, Divide-And-Conquer
- Teilnahme am GCPC oder Wintercontest (Termin noch unbekannt)

Erwünscht:

- Bereitschaft, am NWERC teilzunehmen

NWERC 2018 in Eindhoven, NL

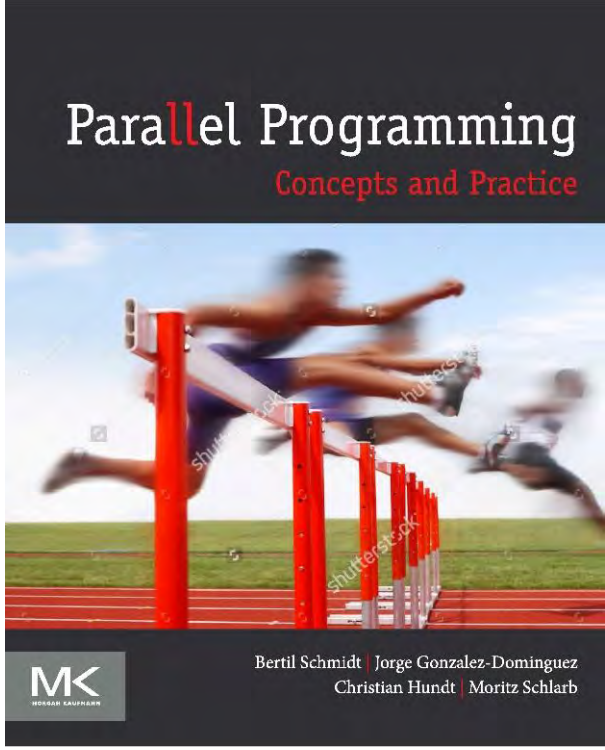
Parallele und Verteilte Architekturen – Lehrangebot: WiSe 2019/20

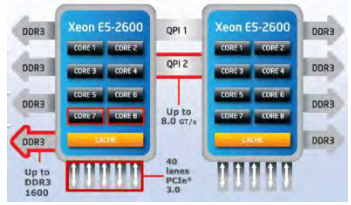
- **Vorlesungen**
  - Einführung in die Bioinformatik
  - High Performance Computing (HPC)
- **Praktikum**
  - Paralleles Programmieren (mit CUDA)
- **Seminar**
  - Paralleles Rechnen

Prof. Bertil Schmidt

High Performance Computing (HPC)

- Background
  - Parallel Hardware
  - Parallel Software
- Shared Memory Programming
  - OpenMP
  - C++11-Multithreading
- Distributed Memory Programming
  - MPI
  - Unified Parallel C (UPC)
- Parallel Program Development
- Practical Parallel Programming Exercises
- Interactive/Blended Learning Approach





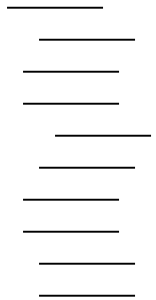
Multi-core CPUs

Manycores

Super-computers

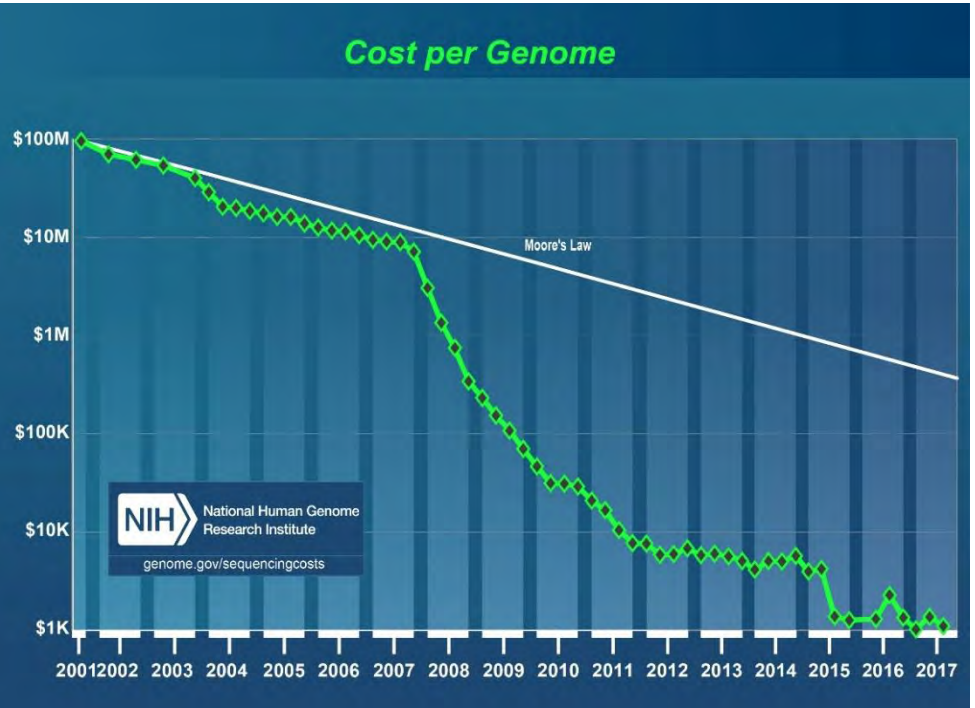
Einführung in die Bioinformatik: Explosive Growth of Sequencing Data

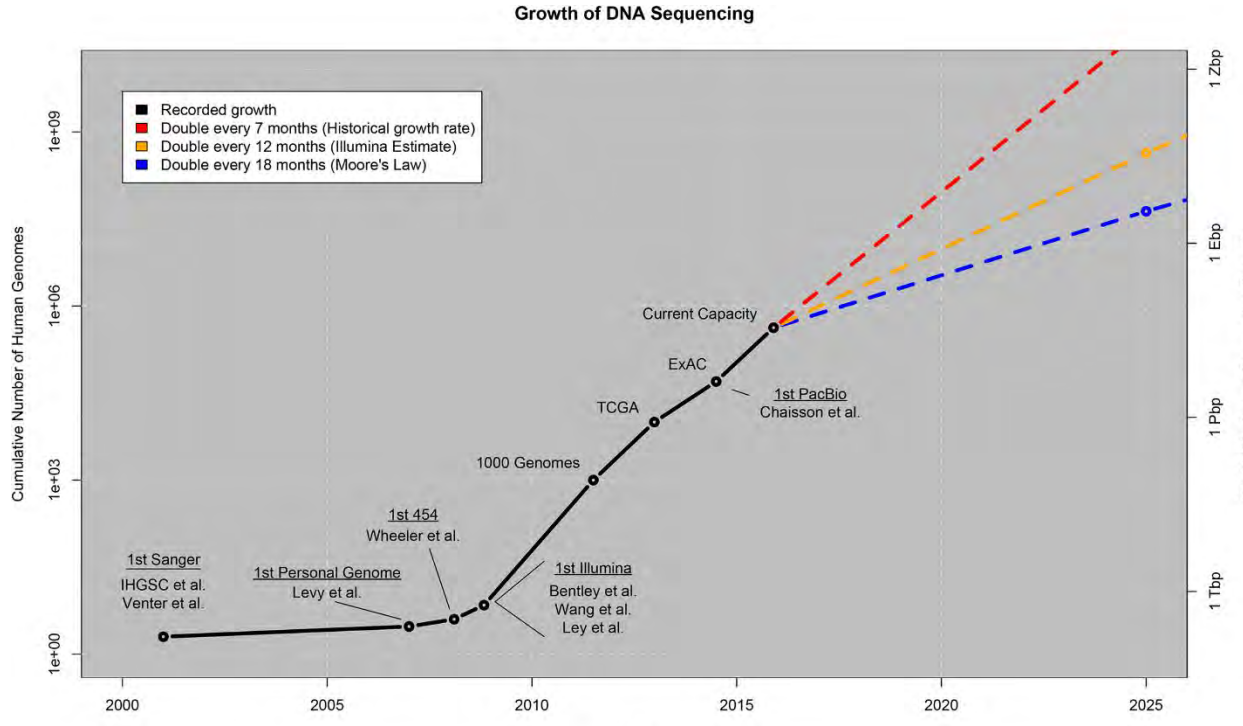
DNA



Read-sequences

DNA-sequence





Einführung in die Bioinformatik

- Study algorithmic techniques that have been successfully applied to bioinformatics problems
- Learn how to design algorithms to solve bioinformatics problems efficiently
  - Optimal sequence alignment (global, local, semi-global alignment, linear & affine gap penalties)
  - Algorithms and data structures for large-scale text indexing and exact search (Suffix Trees/Arrays)
  - Heuristics for read alignment and mapping (mapping DNA-seq and RNA-seq reads)
  - Genome assembly (k-mers, De Brujin graph construction, long-read technology, read-overlap graph assembly)
  - Phylogenetic Trees
  - Clustering
  - Small Programming Exercises with Python and Big Data (Map Reduce)

Block-Praktikum: Paralleles Programmieren (CUDA)

- Paralleles Programmieren mit CUDA
  - **30.9. – 14.10.2019**
  - Voraussetzung: Erfolgreiche Teilnahme an PAA

Seminarvorbesprechung: Paralleles Rechnen

Prof. Bertil Schmidt

Seminar: **Paralleles Rechnen – Themen**

1. Accelerating Reduction and Scan Using Tensor Core Units
2. diBELLA: Distributed Long Read to Long Read Alignment
3. Performance extraction and suitability analysis of multi- and many-core architectures for next generation sequencing secondary analysis
4. Adaptive Sparse Matrix-Matrix Multiplication on the GPU
5. Hardware-conscious Hash-Joins on GPUs
6. Harnessing GPU tensor cores for fast FP16 arithmetic to speedup iterative refinement solvers
7. GRASShopPER—An algorithm for de novo assembly based on GPU alignments
8. Exascale Deep Learning for Climate Analytics
9. Efficient Architecture-Aware Acceleration of BWA-MEM for Multicore Systems
10. A Modular Benchmarking Infrastructure for High-Performance and Reproducible Deep Learning
11. UPC++: A High-Performance Communication Framework for Asynchronous Computation
12. Large-Scale Hierarchical *k*-means for Heterogeneous Many-Core Supercomputers
13. A Framework for the Automatic Vectorization of Parallel Sort on x86-based Processors
14. **Selber vorgeschlagenes Thema** (muss aber von mir genehmigt werden)

Themenvergabe

- E-Mail mit **2** bevorzugten Themen (mit Präferenz) an Prof Schmidt **bis spätestens 12.7.2019**
- Ich werde dann versuchen die Themen an Studenten zuzuordnen
- Neue Themen können auch vorgeschlagen werden (müssen dann aber von mir genehmigt werden)

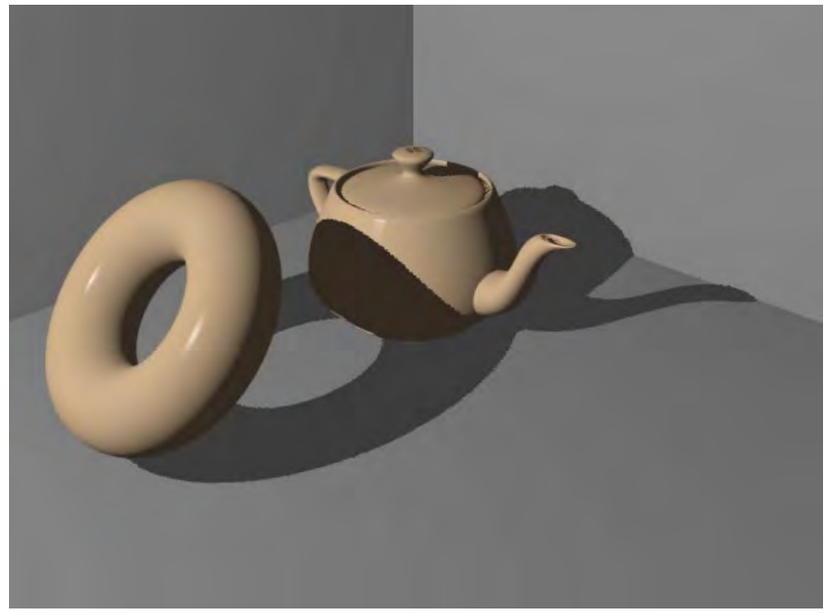
Scheinkriterien und Organisation

- **Voraussetzung:** Erfolgreiche Teilnahme an PAA oder HPC (mindestens Klausurzulassung)
- Vortrag von ca. 45min (inkl. Q&A)
  - Termine: werden noch bekanntgegeben (voraussichtlich Freitags, 12-14Uhr)
- Abgabe der Vortragsfolien
  - **zwei Wochen vor dem Vortrag einzureichen per Email**
  - Danach persönliche Vorbesprechung mit Professor Schmidt
- Abgabe einer ausführlichen Ausarbeitung als Basis für die Bewertung des schriftlichen Teils
  - Ausarbeitung im IEEE CS Format (Umfang mindestens 5-7 Seiten)
  - **Abgabe bis spätestens vier Wochen nach dem Vortrag!**
- Teilnahme am Seminar (Anwesenheitspflicht)

Veranstaltungen WS 19/20 der Arbeitsgruppe

Computational Geometry

Prof. E. Schömer



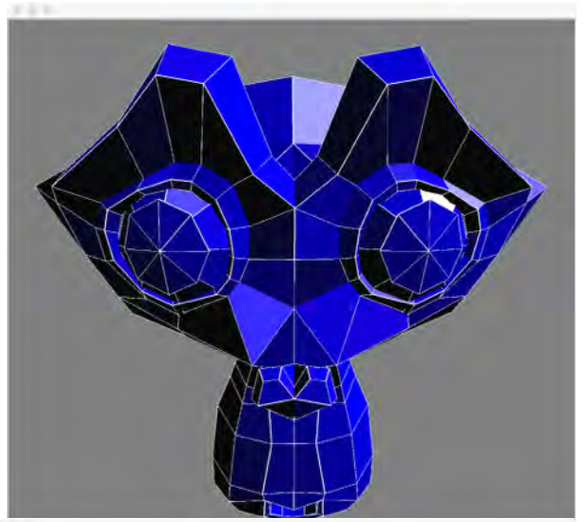
- **Vorlesung: Computergrafik und VR (FR 12-14 04 422)**
- **Seminar: TBA**
- **Praktikum: nach dem Semester**

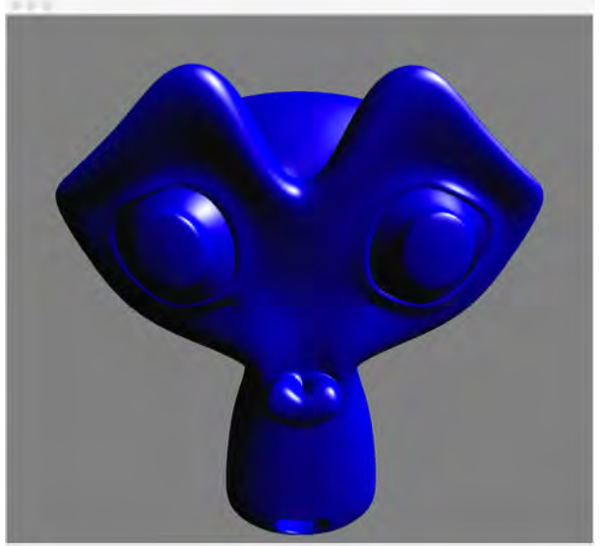
Beschreibung: Typische Einsatzgebiete der Computergrafik sind die grafische Aufbereitung von Mess- und Simulationsdaten und die Schaffung und Animation von virtuellen Welten für Spiele, Filme oder für die Entwicklung technischer Produkte.

Vorkenntnisse: C++ (erwünscht)

Themen der Vorlesung:

- Aufbau und Rendern einer Szene mit OpenGL
- lokale Beleuchtungsmodelle
- Grafik-Pipeline und Shader
- Stereoscopic Rendering
- Subdivision Surfaces





- Prof. Dr. Ulrich Schwanecke
RheinMain University of Applied Sciences Wiesbaden Rüsselsheim Geisenheim
13. April 2011

2D Vision and Deep Learning

RheinMain University of Applied Sciences Wiesbaden
Rüsselsheim

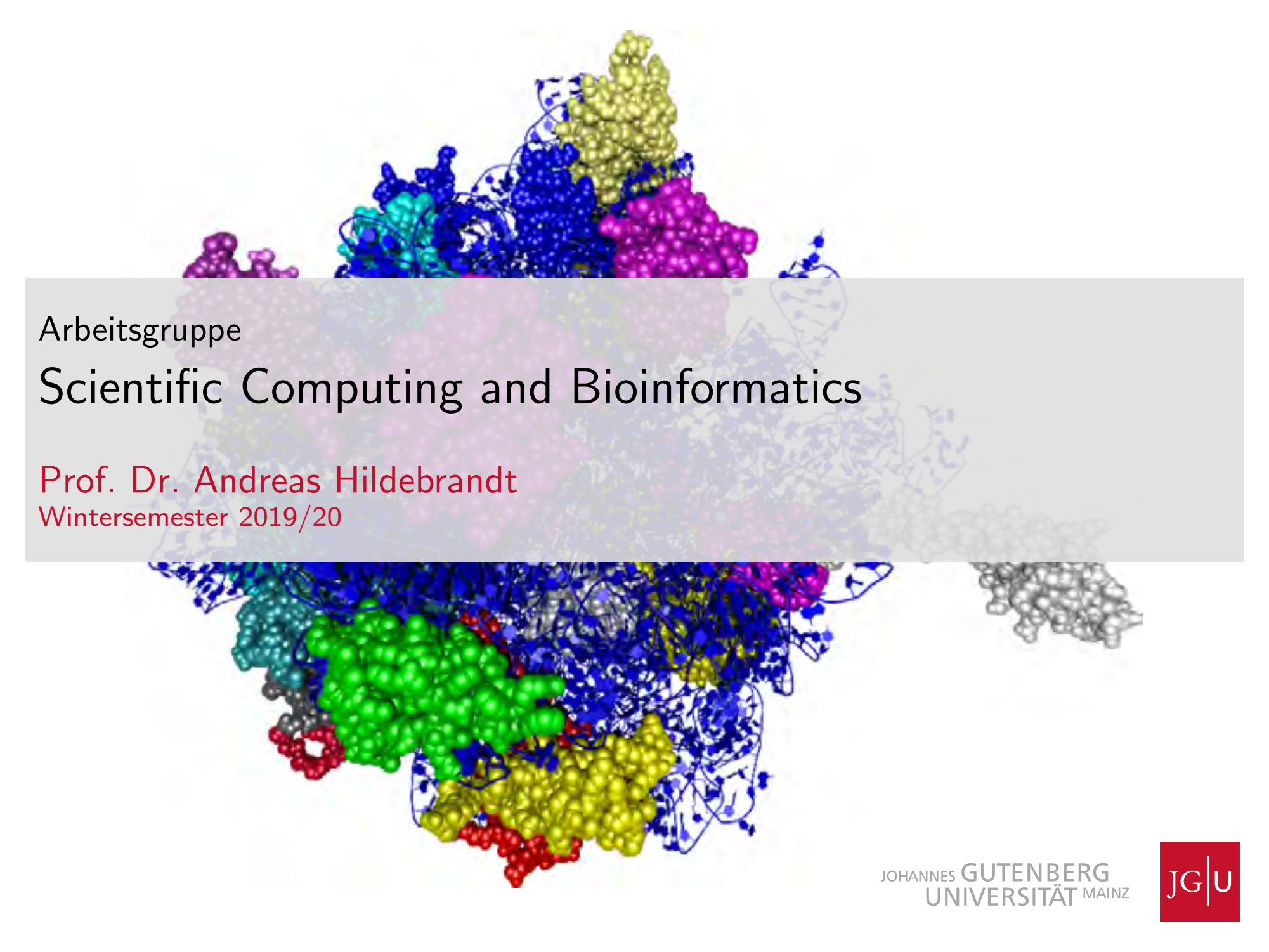
2019/06/26

Content

- History of Computer Vision
- Photometric Image Formation
-
  - The digital camera, point operators, linear filtering, Fourier transform
- Feature Detection and Matching
-
  - Points, edges, lines, SIFT
- Recognition
-
  - Object detection, face recognition
- Machine Learning Basics
-
  - k-NN classifier, Naive Bayes classifier, Multilayer Perceptron
- Convolutional Neural Networks
-

Prerequisites and Literature

- Prerequisites
-
  - Basic knowledge of linear algebra and analysis
  - Programming exercises will be done using Python and Pytorch
-
- Literature
-
  - Richard Szeliski, *Computer Vision: Algorithms and Applications*, Springer 2011
  - Aurélien Néron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques for Building Intelligent Systems*, O'Reilly UK Ltd. 2017
  - Ian Goodfellow, Yoshua Bengio, Aaron Courville, *Deep Learning (Adaptive Computation and Machine Learning)*, MIT Press 2016



Arbeitsgruppe

Scientific Computing and Bioinformatics

Prof. Dr. Andreas Hildebrandt

Wintersemester 2019/20

JOHANNES GUTENBERG
UNIVERSITÄT MAINZ

JG|U

Vorlesung

Big Data

- Termin: Dienstag, 12-14 Uhr
- Sprache: Deutsch
- Inhalt
  - Verteilte Dateisysteme
  - Map/Reduce
  - Komplexität von Map/Reduce-Programmen
  - Frameworks: Hadoop, Spark (Java, Scala)
  - Use Cases: PageRank, Clustering, Recommender Systems
- Gruppenprojekte am Ende des Semesters

Seminar

Vertiefungsseminar Bioinformatik

- Termin: TBA
- Sprache: Englisch
- Themenbereiche (Auswahl):
  - Proteinstrukturaufklärung
  - Maschinelle Lernverfahren in der Bioinformatik
  - Molekülbewegungen
  - Docking
  - Wirkstoffdesign
- Themenwahl voraussichtlich im September
- Keine Themenvergabe ohne Anmeldung in Jogustine

Praktikum

Vertiefungspraktikum Bioinformatik

- Dauer: 2 Wochen, ganztägig
- Termin: März 2020
- Sprache: Deutsch oder Englisch
- Inhalte:
  - Softwarewerkzeuge in der Bioinformatik
  - Molekulardynamicsimulationen
  - Use Cases: Computational Genomics (Python)

Algorithmen und Techniken der Optimierung

Frank Fischer

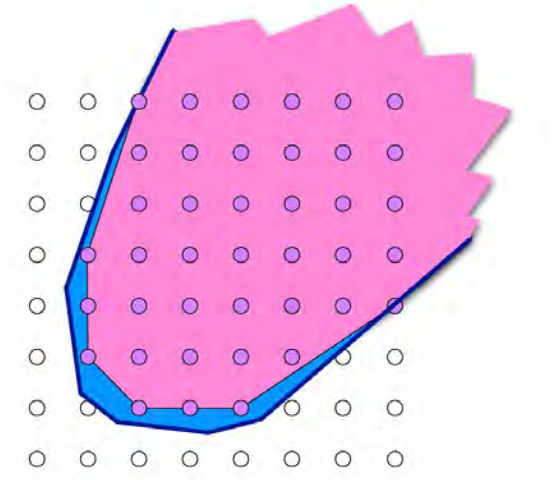
JOHANNES GUTENBERG UNIVERSITÄT MAINZ

JG|U

Überblick

Veranstaltungen

- Teil des Moduls „Algorithmen und Techniken der Optimierung“
  - Theoretische Informatik
- Im Sommersemester:
  - Vorlesung
- Im Wintersemester:
  - Seminar
  - Praktikum (2 Wochen)



- Kenntnis über **Lineare- und ganzzahlige Optimierung** wird vorausgesetzt
  - Besuch der Vorlesung im SS

Inhalt

Seminar

- Thema: **Approximationsalgorithmen**
  - Basierend auf Techniken der Linearen/Ganzzahligen Optimierung
- Die Teilnahme ist ggf. möglich, wenn themenverwandte Vorlesungen besucht wurden (Fortgeschrittene Algorithmen, Graphenalgorithmen, …)
  - Persönliche Vorbesprechung

Praktikum

- Umsetzung/Implementierung fortgeschrittener Techniken an ausgewählten Beispielen
  - Spaltengenerierung, Dantzig-Wolfe-Dekomposition
  - Schnittebenenverfahren
  - Lagrange-Dekomposition, …
  - ...

Termine

Seminar

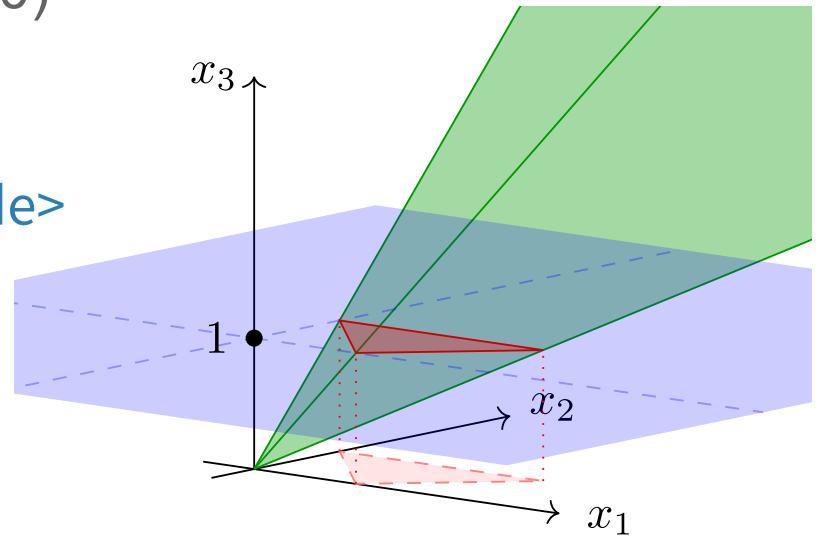
- Semesterbegleitend
- Vorbesprechung in erster Semesterwoche (14. – 18. Oktober)

Praktikum

- 2 wöchig
- Voraussichtlich in der vorlesungfreien Zeit zwischen WS19/20 und SS 20 (08. Februar – 14. April 2020)

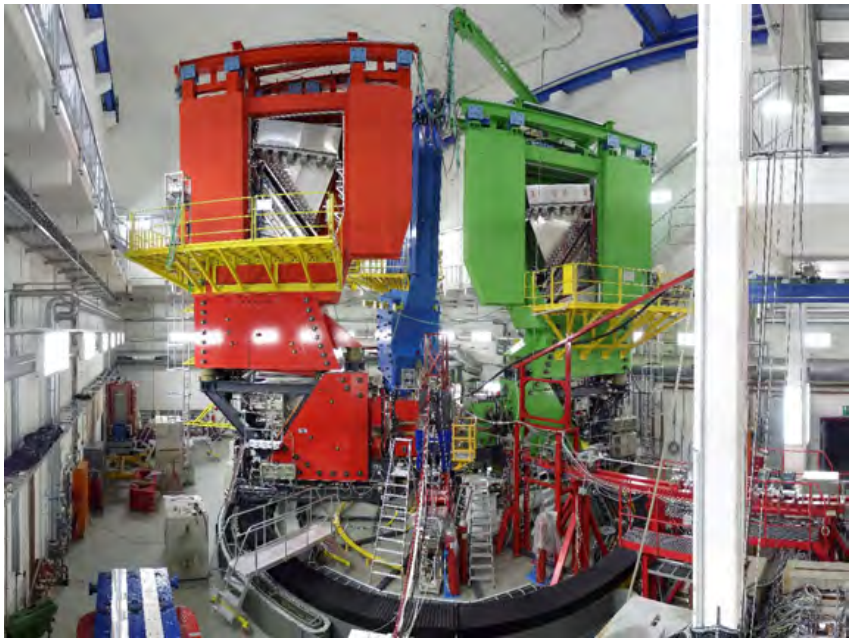
Fragen

- Frank Fischer <frank.fischer@uni-mainz.de>



08.079.212 Betriebssysteme - WiSe 2019/20

Dr. Michael O. Distler <distler@uni-mainz.de>



Mainz, 26. Juni 2019

Gliederung der Vorlesung - Worum geht es?

- Programme, die der internen Verwaltung des Betriebs eines Rechensystems dienen
- Organisation und Koordination von (nebenläufigen) Abläufen
- Optimale oder effiziente Verwaltung von Betriebsmitteln
- Bekannte Beispiele: UNIX, Windows NT

Lehrbuch zur Vorlesung

- A. S. Tanenbaum: Moderne Betriebssysteme,
  3. aktualisierte Auflage, Pearson Studium, 2009
  4. Auflage in englisch

Organisation der Veranstaltung (1)

Vorlesung (2-stündig)

- Vorstellung von Betriebssystem-Konzepten
  - Prozessverwaltung
  - Interprozesskommunikation
  - Speicherverwaltung
  - Dateiverwaltung
  - Input / Output
  - Deadlocks (Systemverklemmungen)

Übungen (Praktikum) (2-stündig)

- Praktikum am Rechner
  - Kennenlernen der Programmierschnittstelle (API) des Betriebssystems UNIX
  - Programmieraufgaben unter Nutzung von Systemdiensten
  - Erwartete Vorkenntnisse: C, gcc, Linux
- Eingestreute Papierübungen

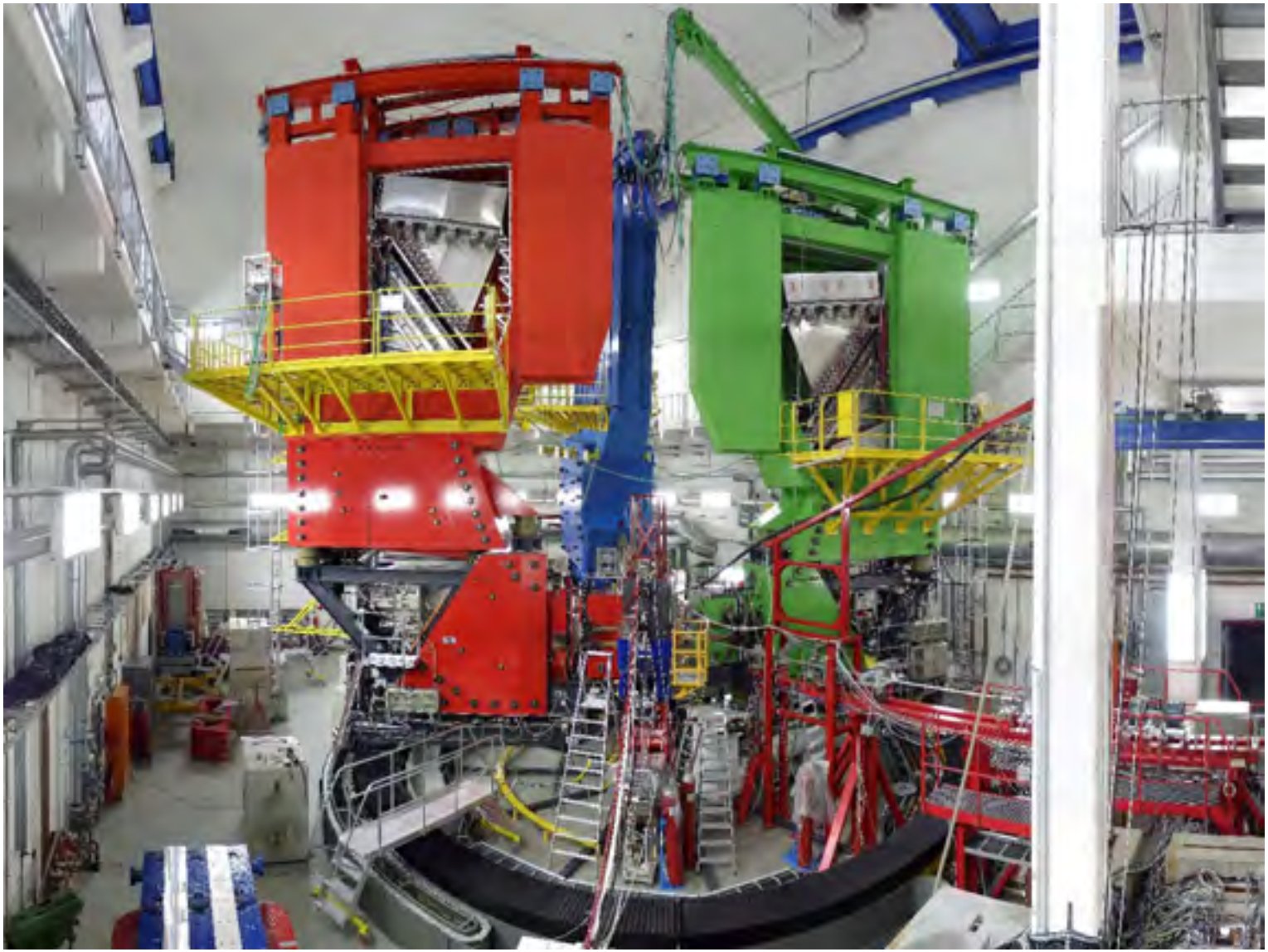
08.079.580 Vertiefungspraktikum Betriebssysteme

- Vertiefungspraktikum Betriebssysteme als Blockpraktikum
- Termin: März 2020
- Ort: Institut für Kernphysik, IT Gruppe
- Teilnehmerzahl ist aus organisatorischen beschränkt auf maximal 6 Studierende

Dr. Michael O. Distler (Vita)

ab 1985	Studium der Physik in Mainz *Vorlesungen bei Prof. Perl:* *Algorithmen, Software-Engineering, KI, . . .*
7/1990	Diplomarbeit: “Aufbau und Test einer vertikalen Driftkammer”
ab 10/1990	Promotionsstudium *dabei: Entwicklung eines Analyse-Programms für Streuexperimente*
1/1997	Promotion: “Elektroproduktion von neutralen Pionen am Wasserstoff an der Schwelle”
1997/1998	PostDoc am M.I.T. in Cambridge/MA
seit 1999	wiss. Mitarbeiter am Institut für Kernphysik *Entwicklung eines Datenaufnahmesystems für die A1 Kollaboration @ MAMI*
seit 2016	Leitung der IT-Gruppe in der KPH *Ausbildung von Fachinformatikern (Systemintegration)*

Die 3-Spektrometer-Anlage @ MAMI



Quanteninformation für Informatiker - Prof. Jochen Walz

- **Qualifikationsziele / Lernergebnisse / Kompetenzen:**
- Die Studierenden erlernen die Grundlagen und den Formalismus der Quantenphysik - soweit für Quanteninformation und Quantencomputing relevant. Sie können mit Qubits umgehen und verstehen die Besonderheiten im Vergleich zu konventionellen Bits. Sie werden mit verschränkten Quantenszuständen vertraut und wissen um die zentrale Bedeutung von Verschränkung (Entanglement). Die Funktionsweise von Quantenkommunikation, Quantenteleportation und von einigen Quantenalgorithmen wird erarbeitet. Die Studierenden verstehen, bei welchen Problemen Quantencomputing vorteilhaft sein kann, und warum. Sie lernen einige experimentelle Plattformen für Quanteninformation und Quantencomputing kennen.

Quanteninformation für Informatiker - Prof. Jochen Walz

- Inhalte:

  - Elementare Quantenphysik (Dirac-Notation, Axiome der Quantenmechanik, abrupte Quantendynamik des Messprozesses, kontinuierliche Quantendynamik der unitären Zeitentwicklung)

  - Qubits (Eigenschaften und Besonderheiten, "no cloning"-Theorem, Quantengatter, Quantenschaltkreise)

  - Verschränkung (Eigenschaften, Nichtlokalität, "no-signaling"-Theorem, Quantenteleportation)

  - Quanteninformations-Verarbeitung (Deutsch-Josza-Algorithmus - ist eine Funktion "balanced" oder nicht?, Quanten-Fourier-Transformation, Shor-Algorithmus - Faktorisieren, Grover-Algorithmus - Suche in einer unsortierten Liste, Quanten-Fehler-Korrektur)

Vorlesung: Montag 10-12 Uhr
Übung: Dienstag 10-12 Uhr

Softwaretechnik

Praktikum	**Web & Mobile Development**
Dozent	Prof. Dr. C. Wille (Lehrauftrag)

Praktikum zur Vorlesung aus dem SS19

- Konzepte und technische Grundlagen von Webservern und Java basierter Webtechnologien
- Konzeption und Entwicklung von Servlets /JSP / JSF
- Rolle von Scriptsprachen wie Javascript im modernes Webdesign
- Konzepte und Entwicklungsschritte mobiler Applikationen
- Software Plattform Android und ihre Komponenten
- GUI-Programmierung für mobile Endgeräte
- Datenverwaltung und Content Provider
- Datenzugriff über Asynchrone Task / Webservice mit SOAP und REST
- Entwicklung von Anwendungen mit Sensoren und Ortsbezogenheit
- Netzwerkprogrammierung für mobile Geräte
- Sicherheit webbasierter und mobiler Anwendungen
- Plattformübergreifende Programmierung.

Studiengang	B.Sc.
Termin	Ende September 2019

Softwaretechnik

Vorl. + Üb.	**Mensch-Maschine-Interaktion**
Dozent	Prof. Dr. Volker Luckas
Inhalte	….
Studiengang	B.Sc. / M.Ed.
Termin	Do. 10.00 – 14.00 Uhr (Vorlesung + Übung)

Fachdidaktik

Vorl. + Üb.	**Fachdidaktik I**
Dozent/en	Prof. Dr. Jens Gallenbacher / Erich Messner
Inhalte	
Studiengang	B.Ed.
Termin	VL + Üb Fr.. 9-13 Uhr

Fachdidaktik

Vorl. + Üb. + Seminar	**Fachdidaktik II**
Dozent/en	Dr. Jens Gallenbacher
Inhalte	
Studiengang	M.Ed. / B.Ed. (neue PO)
Termin	Freitag, 12 - 15 Uhr, Seminar n.V.

Fachdidaktik

Praktikum	**Projektpraktikum**
Dozent/en	Dr. H.-J. Schröder
Inhalte	
Studiengang	M.Ed.
Termin	n.V.

Datenbanken

Seminar	**Datenbanken**
Dozent	Dr. H.-J. Schröder
Inhalte	XML Security Engineering Redesign ….
Studiengang	B.Sc. / M.Ed.
Termin	n.V.

IT-Sicherheit

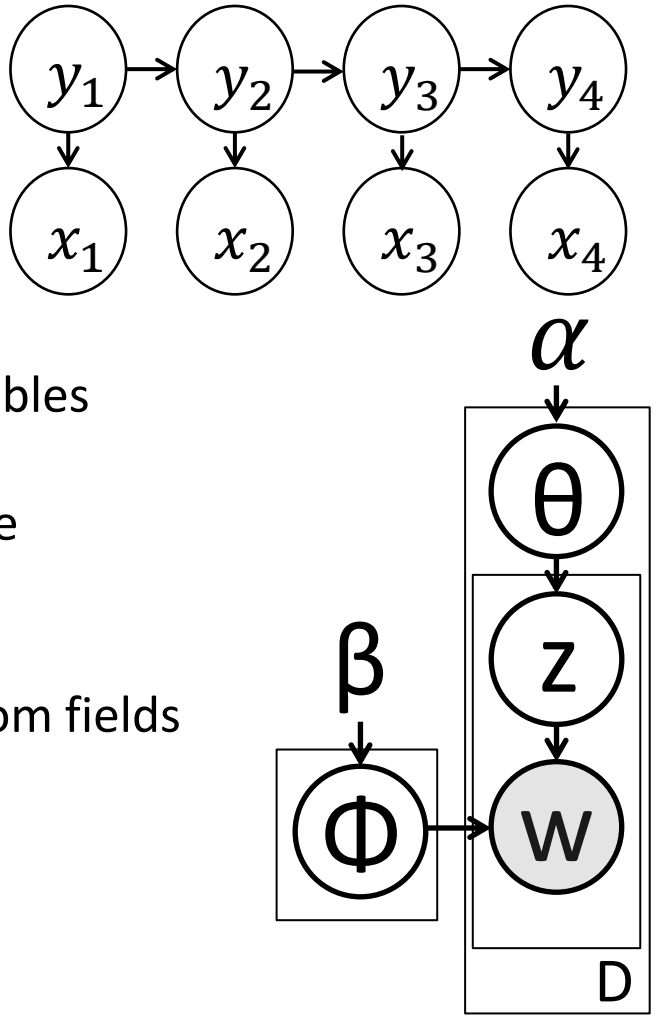
Seminar	**IT-Sicherheit**
Dozent/en	Prof. Dr. Nicolai Kuntze
Inhalte	
Studiengang	B.Sc., M.Ed.
Termin	n.V.

Datenbanken

Seminar	**Design-Pattern**
Dozent	Dr. Stefan Endler
Inhalte	
Studiengang	B.Sc.
Termin	VB 26.09.2019 12.00 Raum 03-424

Probabilistic Graphical Models and Deep Learning

- What are graphical models?
  - Probabilistic models
  - Graph is used to express conditional dependencies between random variables
  - Undirected or directed graphs encode different sets of dependencies
  - Bayesian networks and Markov random fields



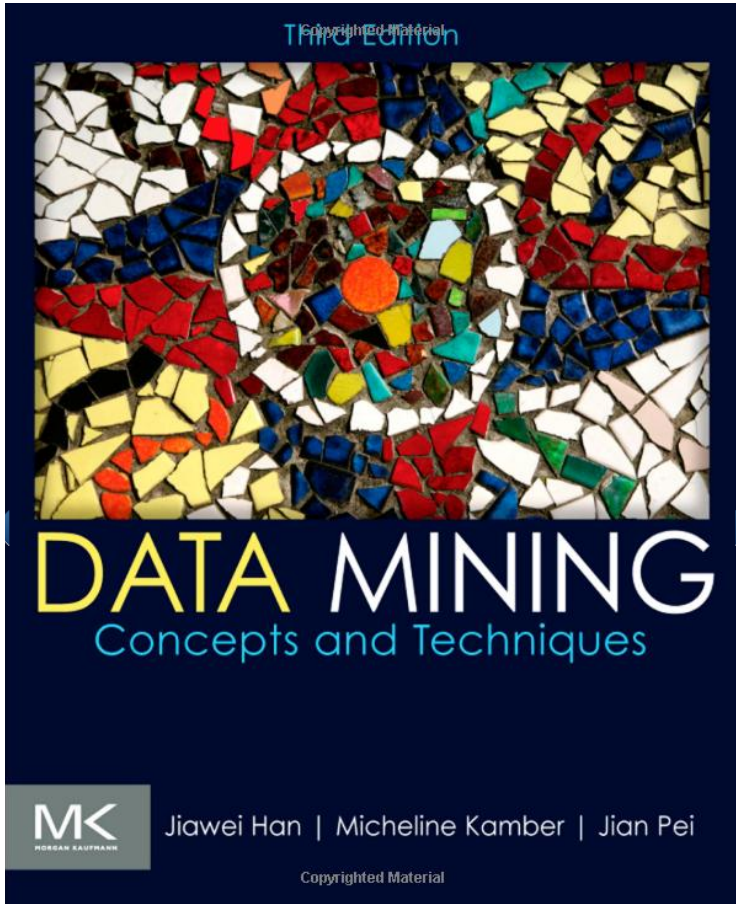
Topics

- Probability Theory, Bayesian Networks
- Inference in graphical models (how to answer queries using the model)
- Learning in graphical models (how to learn the parameters and fit the model to data)
- Sampling methods and variational methods
- Applications in machine learning with a focus on text and NLP
- Deep neural networks that are Bayesian (e.g. variational autoencoders)

Lecture: Monday at 12:00
Tutorial: tba

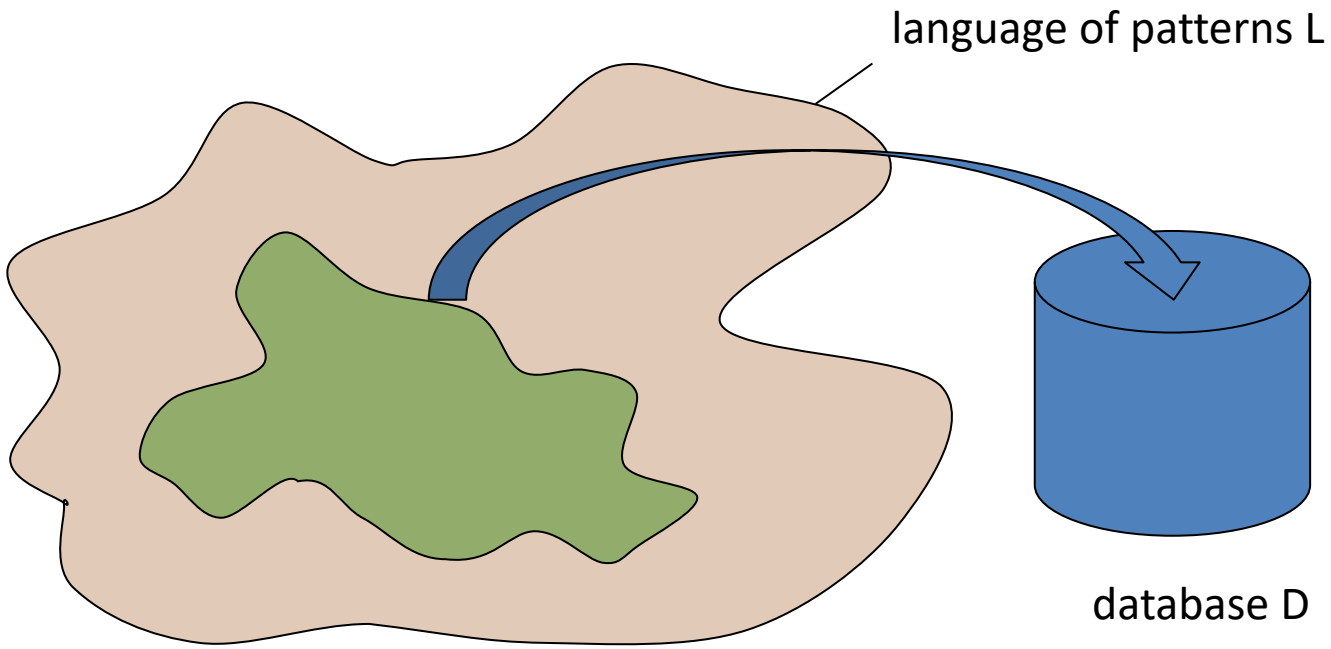
Ablauf

- Vorlesung und Übung
- Vorlesung:
  - Donnerstag, 14-16 Uhr, Raum 03.428
  - Material auf Englisch
- Hands-on
  - Programmierkenntnisse erforderlich
  - Interesse
- Prüfung:
  - schriftlich



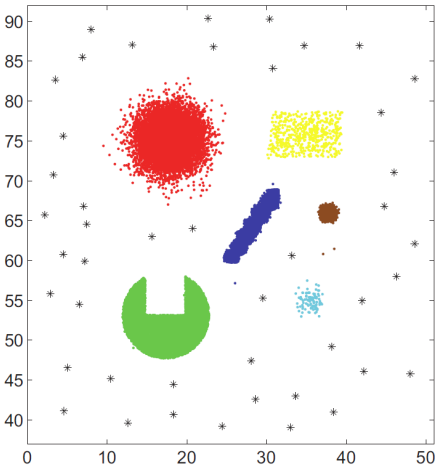
Content / Inhalt der Vorlesung

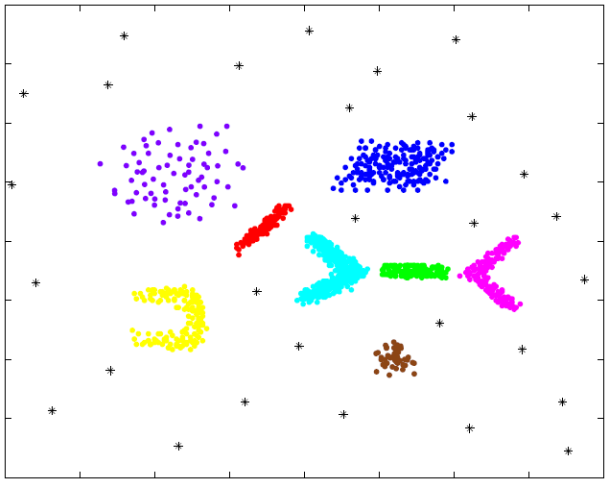
- Pattern Mining



Content / Inhalt der Vorlesung

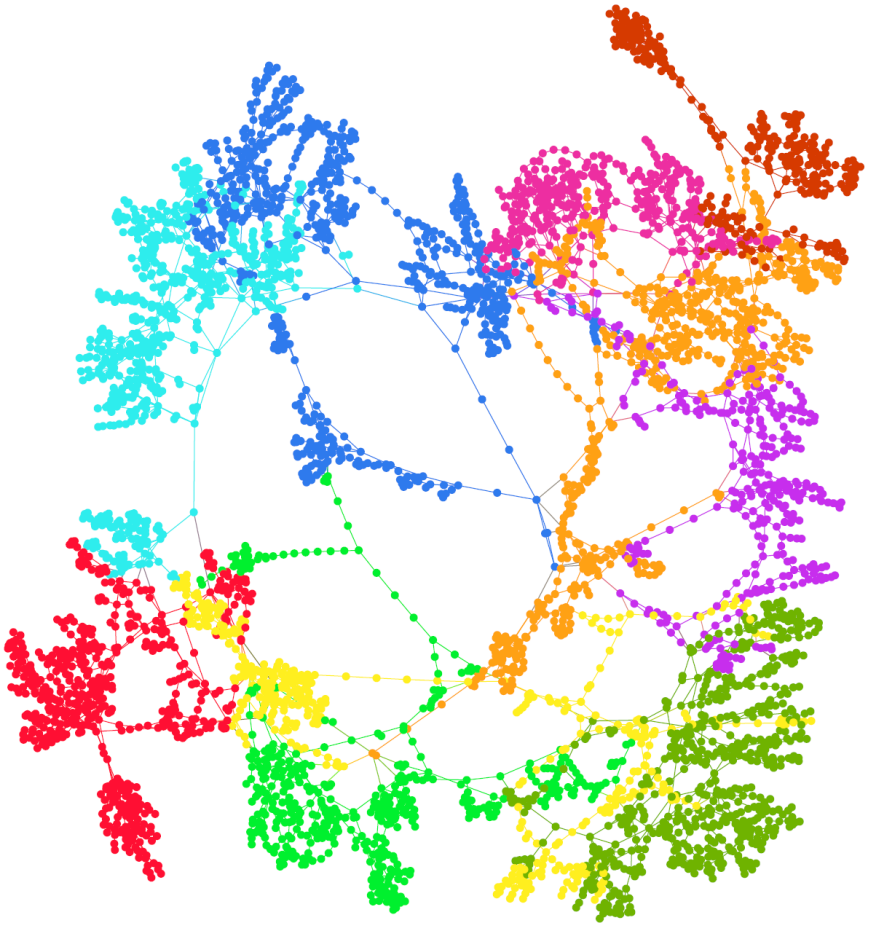
- Pattern Mining
- Clustering





Content / Inhalt der Vorlesung

- Pattern Mining
- Clustering
- Graph Mining



Content / Inhalt der Vorlesung

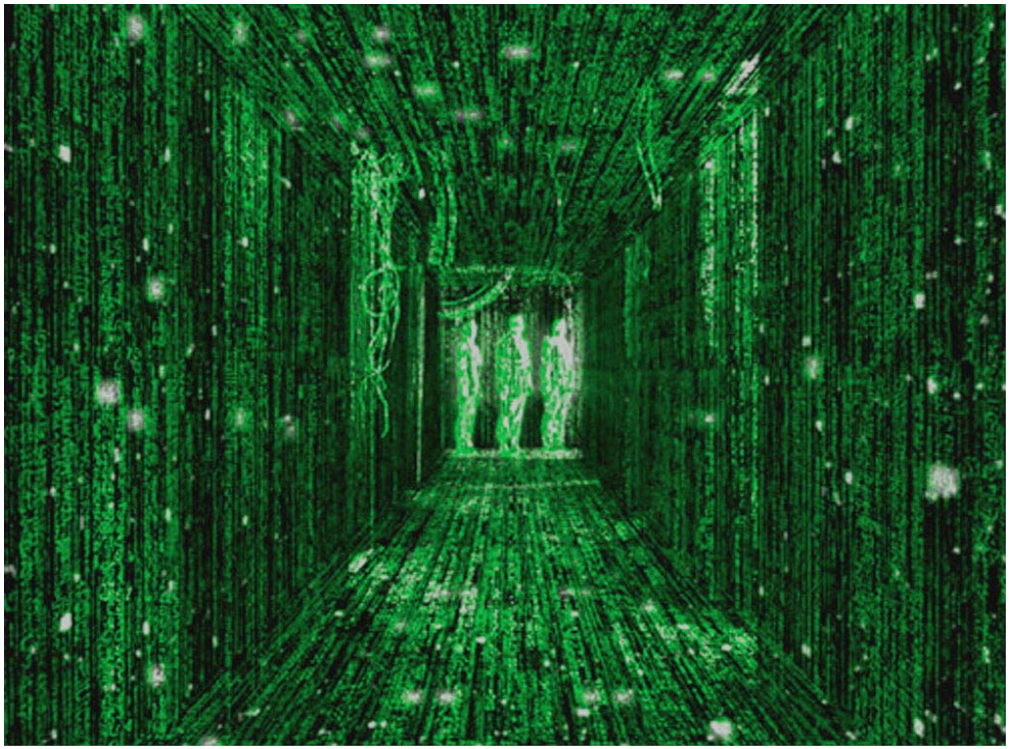
- Pattern Mining
- Clustering
- Graph Mining
- Matrix-faktorisierungen, Dimensions-reduktion, Autoencoder

$$\mathcal{Y} = \overbrace{\begin{pmatrix} 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}}^{q} \Bigg\} m$$

$$\Rightarrow \mathcal{Y}' = \overbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}}^{q'} \Bigg\} m \quad , \mathcal{M} = \overbrace{\begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}}^{q} \Bigg\} q'$$

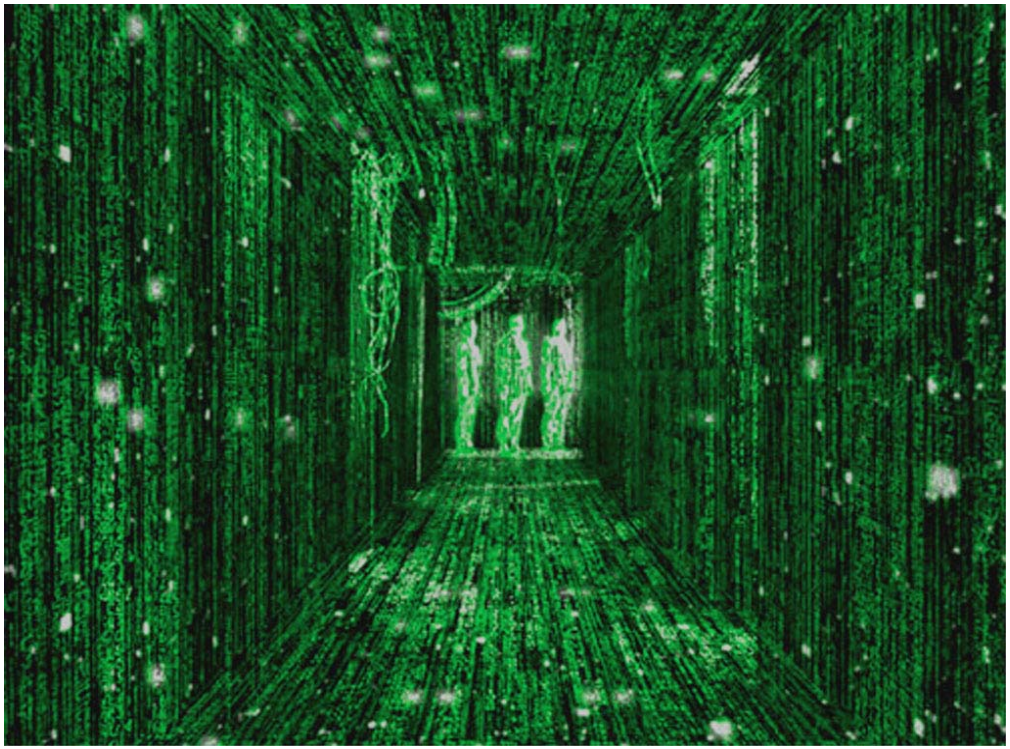
Content / Inhalt der Vorlesung

- Pattern Mining
- Clustering
- Graph Mining
- Matrix-faktor., Dimensions-reduktion, Autoencoder
- Stream Mining



Content / Inhalt der Vorlesung

- Pattern Mining
- Clustering
- Graph Mining
- Matrix-fakor., Dimensions-reduction, Autoencoder
- Stream Mining



Machine Learning Praktikum

- Fokus auf Reinforcement Learning (Verstärkungslernen)
- *Format* ähnlich zu Data Mining Praktikum
- Vorab eine Vorlesung als Brücke

Machine Learning Seminar

- Inkl. *Computational Learning Theory*

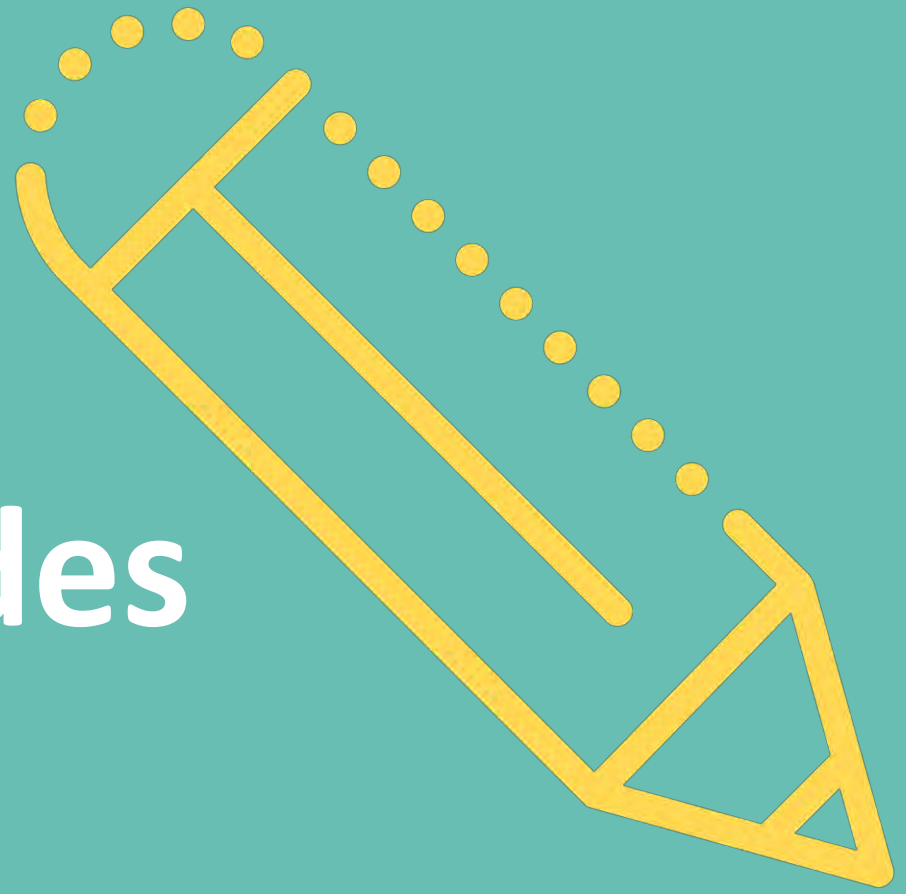
Vorbesprechung zu Praktikum und Seminar:

- Zeit: 5. Juli oder 12. Juli 2019, 11:00
- Ort: TBA

LOB

WISSENSCHAFTLICHES SCHREIBEN

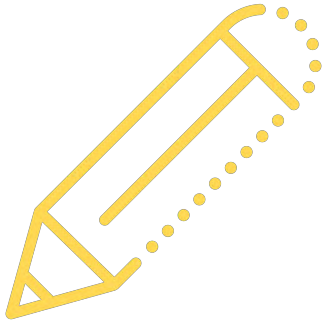
Schreibwerkstatt des Fachbereichs 08



jeden Dienstag von 13 – 16 Uhr in Raum 01-227 (Staudingerweg 9, mittlerer Kreuzbau)

www.studium.fb08.uni-mainz.de/helpdesk/schreibwerkstatt

Schreibwerkstatt



- Du bist Student/Studentin der Physik, Mathematik, Informatik oder Meteorologie?
- Du arbeitest gerade an deiner Bachelor- oder Masterarbeit? Und hast allgemeine Fragen zum wissenschaftlichen Schreiben, zum Aufbau deiner Arbeit, zur Literaturrecherche, etc. ….?

→ **Dann komm in unsere SCHREIBWERKSTATT des Fachbereichs 08**

Jeden Dienstag von 13 – 16 Uhr

Wo: Seminarraum D 01-227 (Staudingerweg 9, mittlerer Kreuzbau)

Start: 16.04.2019

Ohne Anmeldung, einfach vorbeikommen und hier an deiner wissenschaftlichen Arbeit schreiben, zusammen mit anderen Studis und in Begleitung eines Tutors!

Kontakt: Esther Reineke (ereineke@uni-mainz.de)

LOB Lehren Organisieren Beraten

Bundesministerium für Bildung und Forschung

FB 08 PHYSIK, MATHEMATIK UND INFORMATIK

JG|U

Dieses Vorhaben wird aus Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01PL17055 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

ZQ – Kollegiales Audit * (Re-)Akkreditierung

Gruppengespräch mit Studierenden zur Weiterentwicklung der Studiengänge am Institut für Informatik

Liebe Studierenden,
im Rahmen der regelmäßigen Qualitätssicherung und -entwicklung der Studiengänge an der JGU – die sogenannte (Re-)Akkreditierung – möchten wir Sie herzlich zu einem Gruppengespräch zu unterschiedlichen Aspekten Ihres Studiums der Mathematik einladen. Dabei interessieren uns Ihre Erfahrungen und Einschätzungen u.a. zu folgenden Themen:

1. **Studienvoraussetzungen und -motivation**
2. **Studienorganisation und Studieninhalte**
3. **Leistungs- und Prüfungsanforderungen**
4. **Akademische Gemeinschaft (Kontakt zu Lehrenden etc.)**
5. **Berufliche Perspektiven**
6. **Rahmenbedingungen**

ZQ – Kollegiales Audit * (Re-)Akkreditierung

Wir möchten Sie daher bitten, sich die Zeit für ein ca. 1½ -stündiges Gruppengespräch zu nehmen, das von Frau Alina Franz vom Zentrum für Qualitätssicherung und -entwicklung (ZQ) moderiert wird. Das ZQ ist eine unabhängige Einrichtung an der JGU, welche im Rahmen von (Re-)Akkreditierungen

Terminvorschläge:

12.08.19 – 22.08.19 oder 08.07.19 – 16.07.19

Bei einer grundsätzlichen Bereitschaft zu einem solchen Gespräch tragen Sie bitte Ihre Kontaktdaten unter dem passenden Studiengang ein.
Bezüglich der weiteren Gesprächsorganisation werden Sie dann vom ZQ per Mail kontaktiert und erhalten weiter Informationen.

Vielen Dank für Ihr Mitwirken
Alina Franz
Die Kontaktdaten werden nach dem Verfahren gelöscht.