

# Einführung in Computational Engineering



## Grundlagen der Modellierung und Simulation

### 1. Vorlesung: Einführung

14. Oktober 2013

Prof. Dr. Jan Peters

produziert vom **HRZ**  
Hochschulrechenzentrum

# Motivation Computational Engineering: Beispiel

1. Problem Spezifikation: Wie kann man Tisch Tennis spielen?

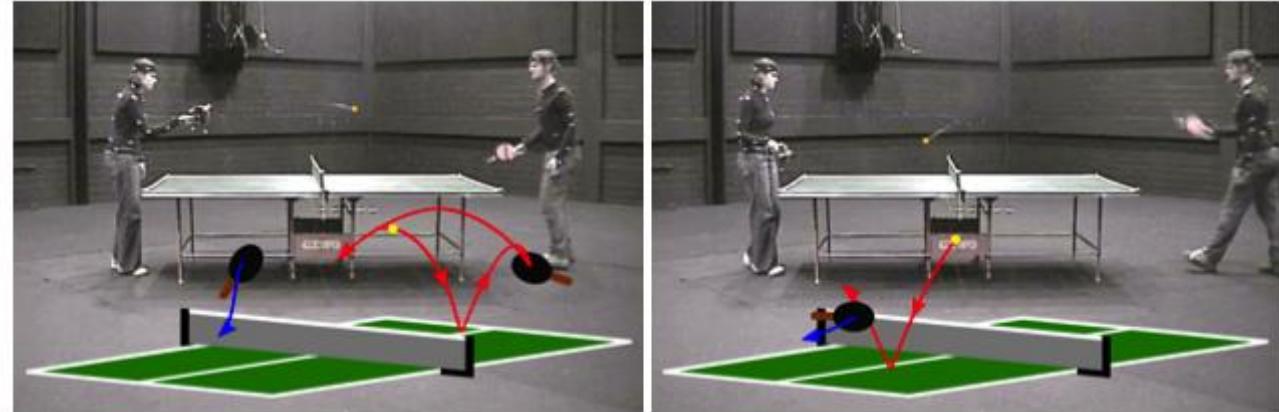


# Motivation Computational Engineering : Beispiel

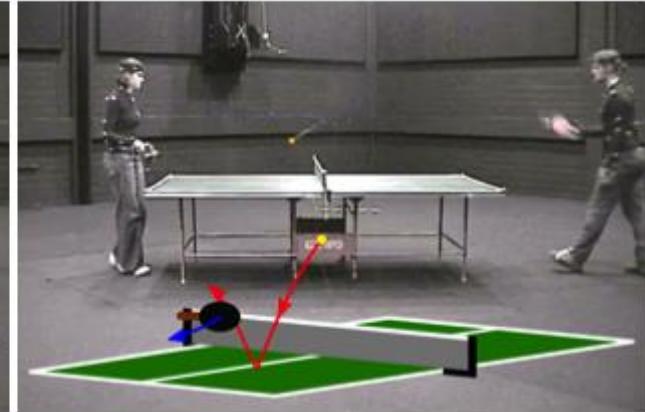


TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

2. Modellierung: Wie  
spielen Menschen  
Tisch Tennis?



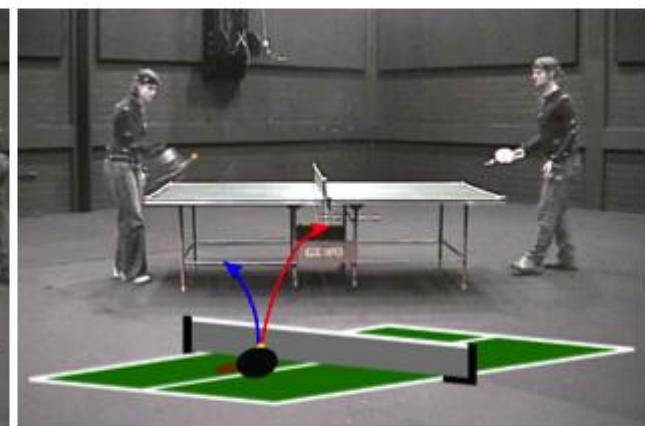
(a) Awaiting Stage



(b) Preparation Stage



(c) Hitting Stage



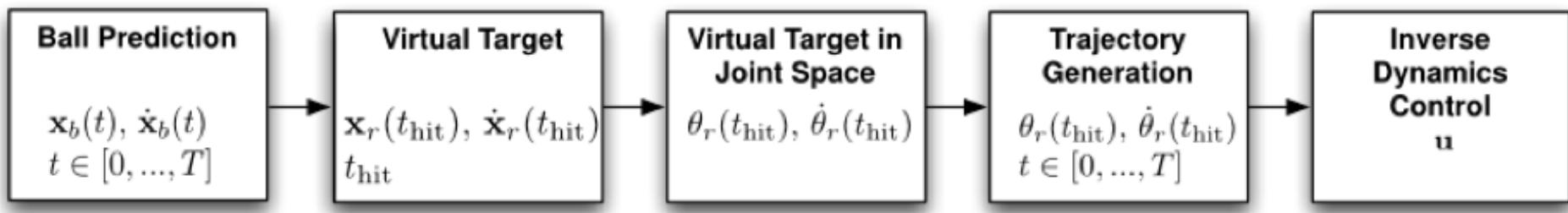
(d) Finishing Stage

# Motivation Computational Engineering : Beispiel



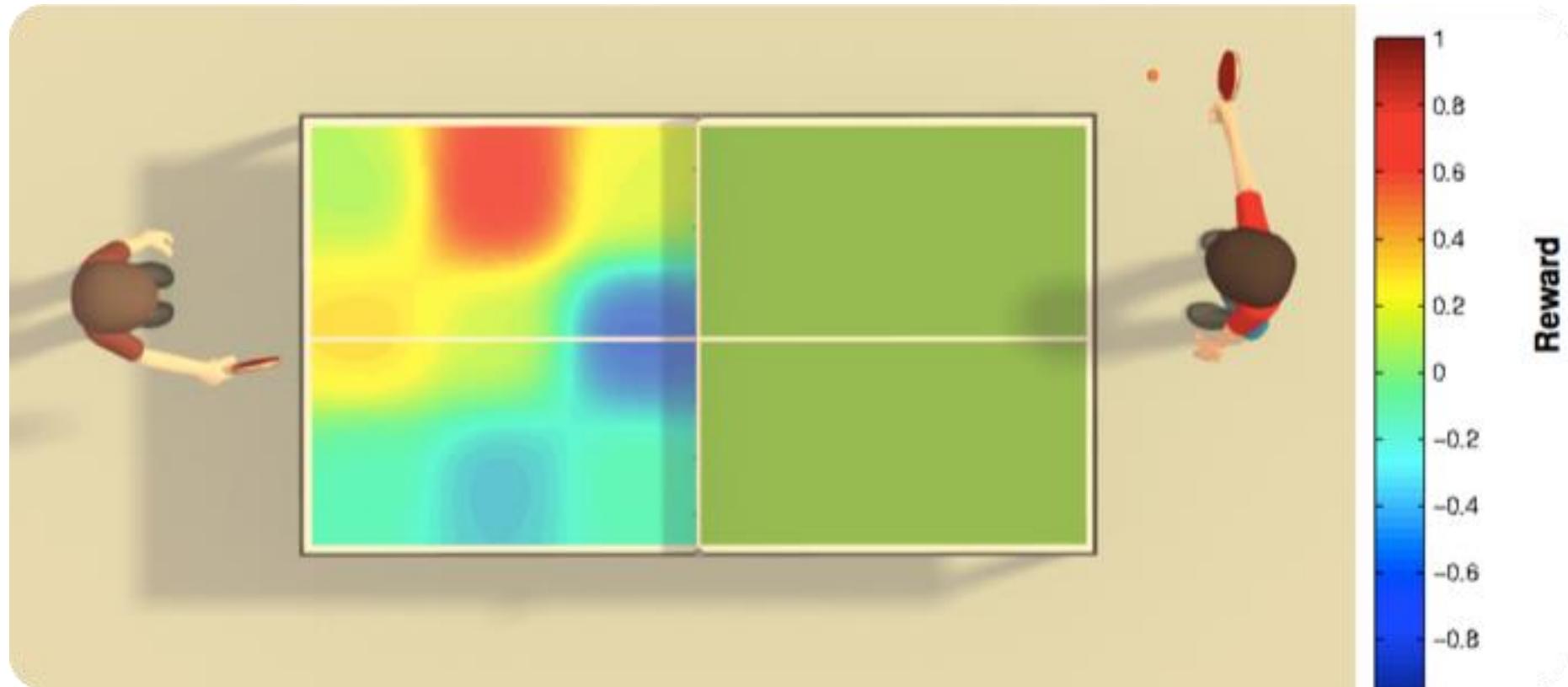
TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## 3. Implementierung: Model als Software Realisieren



# Motivation Computational Engineering : Beispiel

## 4. Validierung: Abgleich von Simulation und Realität



# Motivation Computational Engineering : Beispiel



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## 4. Anwendung: Tischtennis mit einem Roboter



AdaptiveBehavior2.mov

# Organisatorisches



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Vortragender:

Jan Peters

Betreuung:

Christian Daniel, Herke van Hoof

Termine:

Montag, 13:30-16:05 im Raum S202/C205

TU-CAN:

Übertragung in Raum S202/C110

Semesterwochenstunden:

20-00-0011-iv Einführung in Computational Engineering (Grundlagen der Modellierung und Simulation)

3

Exam:

Fr, 28. Feb. 2014 09:00-11:30

# Organisatorisches: Tutoren

---

## Termine der Tutorensprechstunden:

Tag	Uhrzeit	Datum	Raum
Dienstag	10:00	15.10.2013 - 11.2.2014	A213 (Rebecca, Linh, Johannes)
Dienstag	10:00	15.10.2013 - 11.2.2014	E302 (Andrej, Franklin & Thomas)
Mittwoch	16:30	16.10.2013 - 13.11.2013	E202 (Jan, Doro, Chinara)
Mittwoch	16:30	20.11.2013 - 4.12.2013	E203 (Jan S, Doro, Chinara)
Mittwoch	16:30	11.12.2013 - 12.2.2014	E202 (Jan S, Doro, Chinara)

# **Organisatorisches: Übungen**



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

---

Die Abgabe der Übungsaufgaben zur Korrektur und Bewertung stellt ein zusätzliches freiwilliges Angebot dar.

Die Bonuspunkteregelung wird maximal genutzt (volle Notenstufe).

Die Note 1,0 kann in der Klausur ohne Anrechnung im Übungsbetrieb erbrachter Leistungen erreicht werden.

# Organisatorisches: Übungen (2)



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Das erste bewertete Übungsblatt erscheint heute, am Montag, 14.Oktober 2014.

Das Abgabedatum ist auf jedem Übungsblatt angegeben

Abgabe der Übungsaufgaben am folgenden Montag bis spätestens 13:25 Uhr:

- Idealerweise in der Vorlesung!
- Im Notfall: im Briefkasten des Fachgebietes Intelligente Autonome Systeme neben dem Sekretariat E314 in S2|02 vor der Vorlesung.

Alle weiteren Übungsblätter erscheinen auch jeweils montags online auf dieser Kursseite zur Veranstaltung.

# Organisatorisches: Übungen (3)



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

---

Die Aufgaben dürfen in Gruppen von **bis zu zwei Personen** bearbeitet werden.

Es muss nur eine Lösung eingereicht werden, aber mit den Matrikelnummern beider Personen.

Die Beantwortung der Aufgaben hat stets mit **Begründung und nachvollziehbaren Lösungswegen** zu erfolgen.

In Ihrem eigene Interesse sollten Sie *alle* Aufgaben der Übung selbständig bearbeitet haben, nur so sind sie hinreichend auf die Klausur vorbereitet. In der Klausur können alle Typen von Aufgaben gestellt werden, nicht nur den Hausaufgaben entsprechende.

# Organisatorisches: Programmierpraktikum



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

Insgesamt werden 4 Programmieraufgaben gestellt

Bearbeitungszeit: mind. 2 Wochen.

Software: Matlab

Die Programmieraufgaben sind zum jeweils angegebenen Termin per Datei-Upload im Lernportal Informatik einzureichen.

Die Regelungen bzgl. Gruppenarbeit und Plagiarismus kommen bei den Programmieraufgaben analog zu den Übungsaufgaben zur Geltung.

# Webseite und Vorlesungsbegleitende Fragen



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- Webseite mit Wiki
  - <http://lehre.ias.informatik.tu-darmstadt.de/ComputationalEngineering>
- Moodle
  - <https://moodle.informatik.tu-darmstadt.de>



Bitte jetzt auf Moodle eine Frage  
beantworten!

- Mein Dr-Vater: “*There are no stupid questions! Just professors too stupid to answer questions.*”
- Trotzdem: Wenn Sie eine gute Frage stellen, dann riskieren Sie, dass Ihre KommilitonInnen diese lustig finden und “SNIGGERN” (engl. für ein unterdrücktes oder leises Auflachen).



Daher bekommen gute Fragen und Antworten  
von mir den Snigger Award!



- Ich beschäftige mich u.A., mit "Reinforcement Learning", also dem Lernen aus Versuch und Fehler. **Bitte geben Sie mir Feedback!**
- Mein Motto: ***Jeder Prof hat 'ne Meise! Meine dürfen Sie füttern!***

# Überblick der Vorlesungsinhalte



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

1. Einführung
2. Diskrete Modellierung und Simulation
3. Zeitkontinuierliche Modellierung und Simulation
4. Teilschritte einer Simulationsstudie
5. Interpretation und Validierung
6. Modulare und objektorientierte Modellierung und Simulation
7. Parameteridentifikation für Modelle



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

---

Grundlagen der Modellierung und Simulation

# 1. EINFÜHRUNG

# Heutige Lernziele: Kernfragen



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- Was sind die Grundbegriffe der Simulation?
- Was bedeuten Zustand, System und Modell?
- Wofür kann man Simulationen einsetzen?
- Wie läuft eine Simulationstudie an?
- Was bedeuten Problemspezifikation, Modellierung, Implementierung, Validierung und Simulationsanwendung?
- Wie leitet man Modelle her?
- Wie kann man Modelle klassifizieren?



## **Beispiel für eine Simulationsstudie: Schiffschaukel**

(Folien zu Schiffschaukel beruhen auf W. Wiechert, Uni Siegen)

# Problemspezifikation: Beispiel



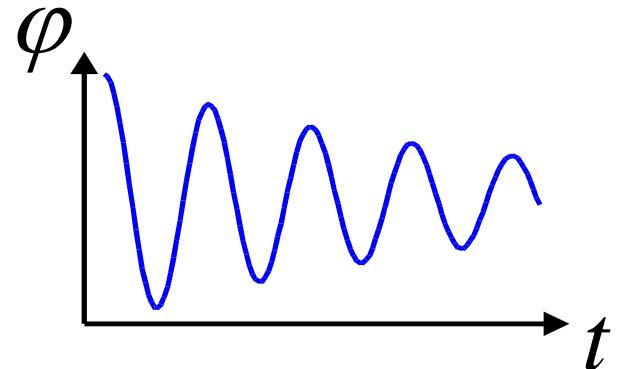
- Zweck der Simulation
  - Wartezeit und Durchsatz von Kunden
  - Belastung in den unterschiedlichen Tragestrukturen
- Die Zustandsvariablen der resultierenden Modelle ändern sich zeitdiskret (Kunden) und zeitkontinuierlich (Belastungen)
- Das Modell kann reale Bewegungen nur mit gewissen Einschränkungen wiedergeben

# Mathematische Modellierung: Beispiel

- resultierende Modelle sind Systeme gewöhnlicher Differentialgleichungen (Kapitel 3)

$$\dot{\varphi} = \omega$$

$$\ddot{\omega} = -\left(2\frac{i}{l} + \frac{d}{ml^2}\right)\omega - \frac{g}{l} \sin \varphi$$



- $\varphi$  und  $\omega$  sind Zustandsvariablen
  - gewöhnliche Differentialgleichungen beschreibt Verhalten

# 1.1 Begriffsbildung



- System
  - Ausschnitt der realen Welt
  - Definiert durch Zweck, Funktion, etc.
  - Gegliedert in unabhängige, interagierende Komponenten (Subsysteme)
- Modell
  - Gebildet unter Annahmen und Idealisierung
  - (vereinfachendes) Abbild einer (partiellen) Realität

# 1.1 Begriffsbildung: Beispiel



- Verkleinerung des Originals
- möglichst „naturgetreues“ Aussehen
- Funktionalität des Originals?



„klassisches“ Modell



Original des Modells

# 1.1 Begriffsbildung

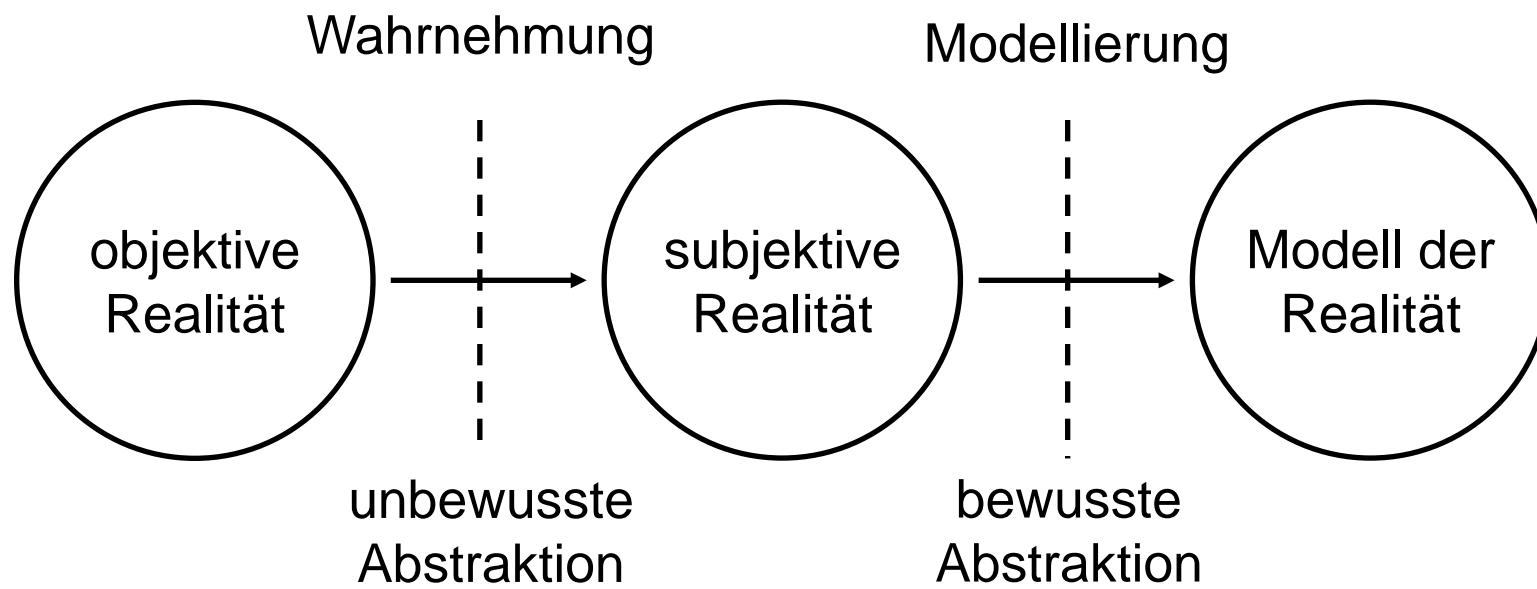


- Abstraktes Modell
  - Formale Beschreibung, typischerweise mit mathematischer Beschreibung
- Mathematische Modellierung
  - Prozess der formalen Herleitung und Analyse eines mathematischen Modells zur besseren Lösbarkeit

# 1.1 Begriffsbildung



- Modellierung als Abstraktionsprozess





# 1.1 Begriffsbildung

- Simulation
  - Ursprung: simulare (lat.) = nachahmen (dt.)
  - Virtuelles Experiment am Modell, weiterführendes Ziel der Modellierung
- Simulationsziele
  - Bekanntes Szenario verstehen bzw. nachvollziehen
  - Bekanntes Szenario optimieren
  - Unbekanntes Szenario vorhersagen

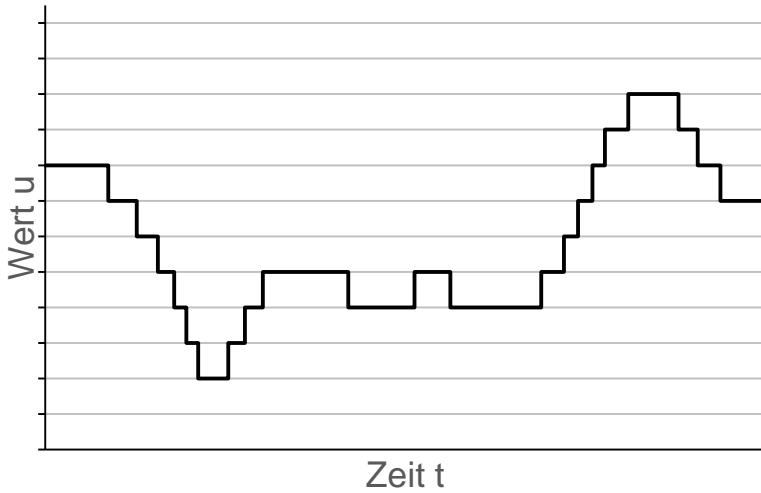
# 1.1 Begriffsbildung



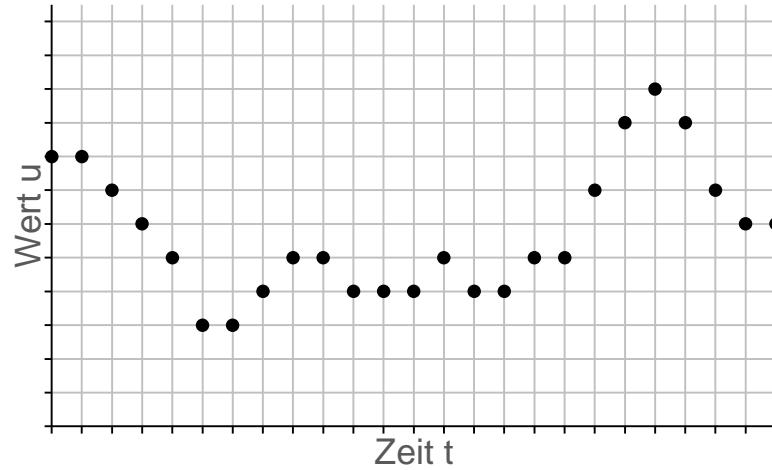
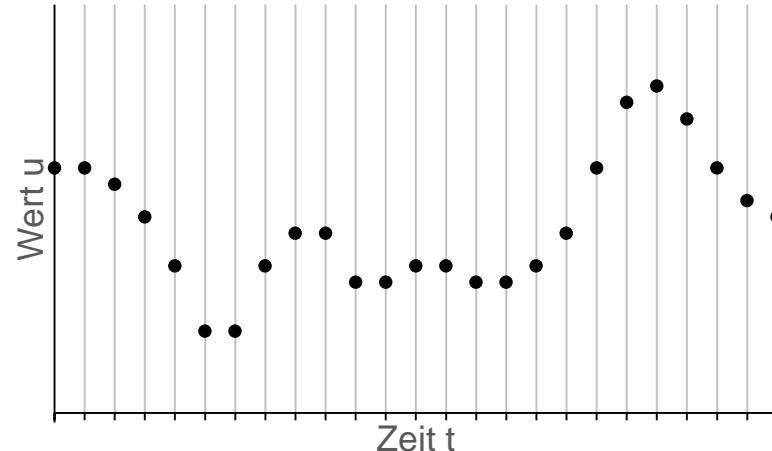
wertkontinuierlich



wertdiskret



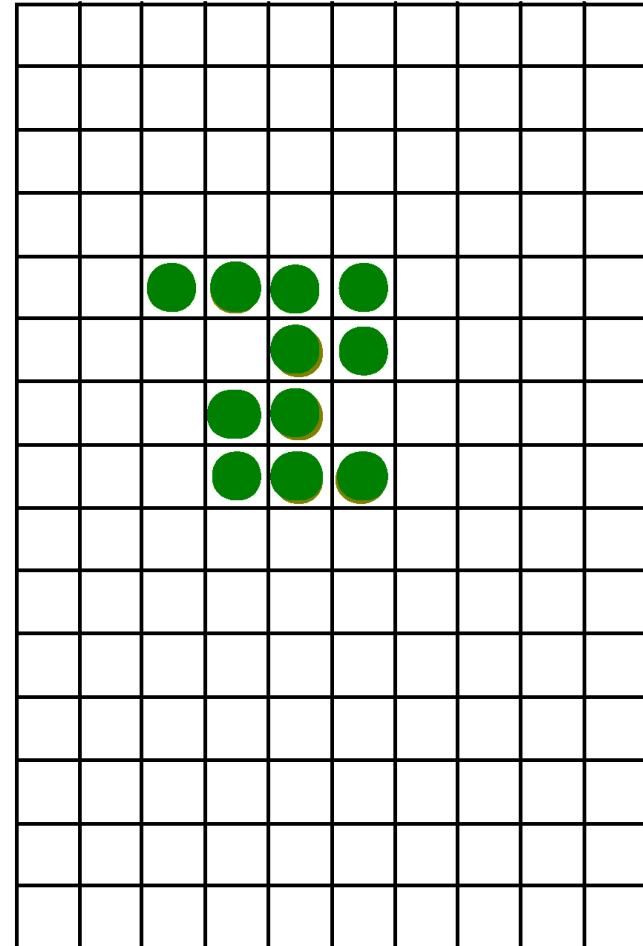
zeitdiskret



Bitte jetzt auf Moodle eine Frage  
beantworten!

# 1.1 Zeitdiskretes Modell: Game of Life Beispiel

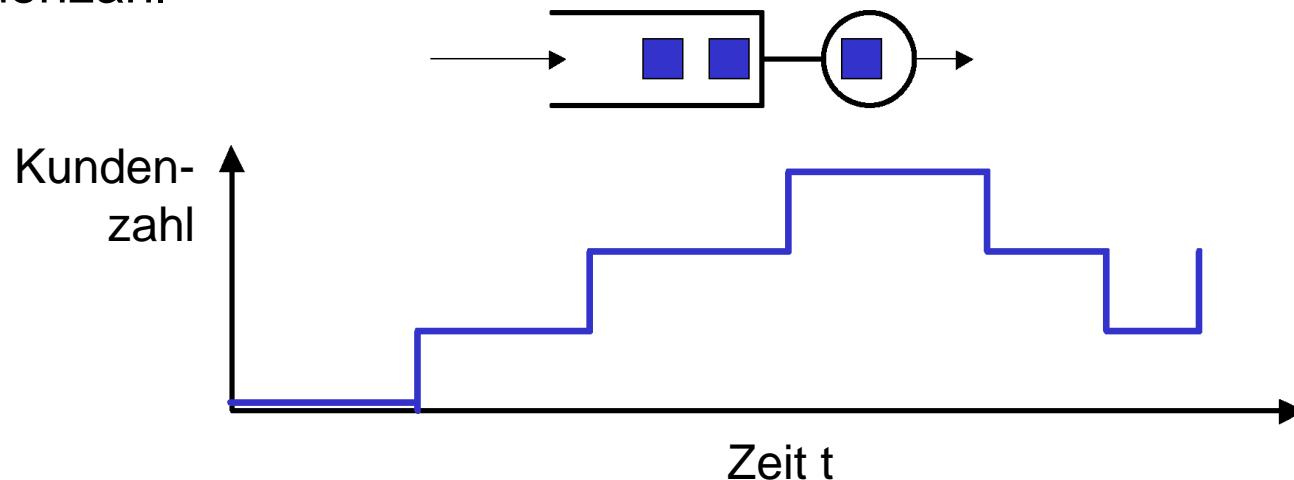
- Spielbrett mit quadratischen Zellen
- Jede Zelle ist tot oder lebendig
- Status der Zellen ändert sich von Generation zu Generation
  - wird lebendig, bei genau 3 lebendige Nachbarn
  - bleibt lebendig, wenn sie 2 oder 3 lebendige Nachbarn hat
  - in allen anderen Fällen ist die Zelle in der nächsten Generation tot



# 1.1 Ereignisdiskretes Modell: Bediensystem Beispiel



- Objekte
  - Kunden, Bedieneinrichtung, Warteschlange
- Attribute
  - Ankunftszeiten, Bedienzeiten, Kundenzahl
- Änderungen der Attribute
  - Bei Ankunft: Bediener belegen, Kundenzahl + 1
  - Bei Bedienende: nächster Kunde, Kundenzahl -1



## 1.2 Simulationsziele: Beispiel



Bekanntes Szenario		Unbekanntes Szenario
Nachvollziehen	Optimieren	Vorhersagen
<ul style="list-style-type: none"><li>▪ Naturkatastrophen</li></ul>	<ul style="list-style-type: none"><li>▪ Flugeinsatzplan einer Fluggesellschaft</li><li>▪ Wärmeabtransport eines Kühlsystems</li></ul>	<ul style="list-style-type: none"><li>▪ Klimaveränderung und Wettervorhersage</li><li>▪ Bevölkerungswachstum</li></ul>

## 1.2.1 Simulationszweck (Reales System)



- Zu groß, zu klein
  - Galaxien, Moleküle
- Zu schnell, zu langsam
  - Kernreaktion, Populationsdynamik, Klima
- Noch nicht gebaut
  - Flugzeug, integrierte Schaltung, Fertigungssteuerung
- Zu gefährlich
  - Kernreaktionen, Humanmedizin

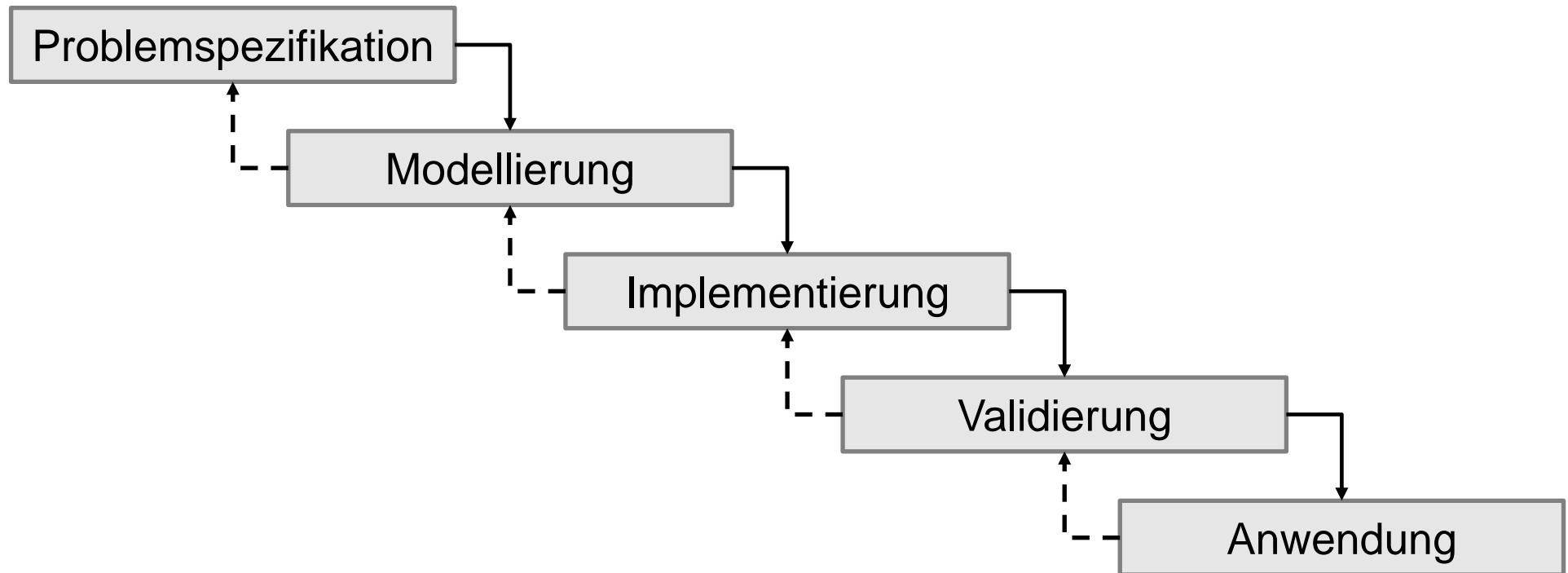
## 1.2.1 Simulationszweck (Reales System)



- Nicht experimentierbar
  - Volkswirtschaft, Ökosystem
- Zu teuer
  - Prototypen, Crash-Test
- Zu stark gestört
  - Börsenkurse, Wirtschaft

Fazit: Simulation, wenn reales System nicht oder nur unter hohem Aufwand verfügbar ist.

## 1.2.2 Ablauf der Simulationspipeline



## 1.2.2 Ablauf der Simulationspipeline



- Problemspezifikation
  - Aufgabenformulierung, Kriterienfeststellung, Datenerhebung
- Modellierung
  - Strukturfestlegung, Modellgleichung, -vereinfachung, -analyse
- Implementierung
  - Modellumsetzung / Berechnungsverfahren, Laufzeitoptimierung

## 1.2.2 Ablauf der Simulationspipeline



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- Validierung
  - Fehlersuche, Konsistenzprüfung, Daten- und Parameterabgleich
- Anwendung
  - Auswertung und Optimierung, Parameter- und Strukturvariationen

# 1.3 Anwendungsbeispiele



- Wo wird überall modelliert und simuliert?

- Halbleiterindustrie
- Computergraphik
- Logistik / Ablaufplanung
- Verkehrsszenarien
- Software Engineering

The screenshot shows an IDE interface with two panes. The left pane, titled 'JUnit', displays the test results: 'Finished after 0.023 seconds' with 'Runs: 2/2', 'Errors: 0', and 'Failures: 0'. It lists two tests: 'sort.QuickSortTest [Runner: JUnit 4] (0.000 s)' which includes 'sortSequenceSuccess (0.000 s)' and 'sortSequenceFail (0.000 s)'. The right pane shows the Java code for 'QuicksortTest.java':

```
1 package sort; 2 3 import org.junit.Assert; 4 import org.junit.BeforeClass; 5 import org.junit.Test; 6 7 public class QuicksortTest { 8 9     private static Quicksort qs; 10 11     @BeforeClass 12     public static void init() { 13         qs = new Quicksort(); 14     } 15 16     @Test 17     public void sortSequenceSuccess() { 18         int[] actuals = qs.sort(new int[] {13, 4, 21, 42, 1337, 27, 28, 666, 0}); 19         Assert.assertArrayEquals(new int[] {0, 4, 13, 21, 27, 28, 42, 666, 1337}, actuals); 20     } 21 22     @Test 23     public void sortSequenceFail() { 24         int[] actuals = qs.sort(new int[] {13, 4, 21, 42, 1337, 27, 28, 666, 0}); 25         Assert.assertNotEquals(new int[] {0}, actuals); 26     } 27 28 }
```

# 1.3 Anwendungsbeispiele



- Unterschiedliche Anwendungsgebiete
  - Daher „harte“ versus „weiche“ Modellierung
  - Unterschiedliche Zielsetzungen
  - Unterschiedliche Werkzeuge der Mathematik bzw. Informatik werden genutzt und benötigt
- Wie kommt man zu einem Modell?
- Welche Beschreibungsmittel nimmt man her?

# 1.4 Herleitung von Modellen



- Anforderungen an Modelle
  - Ausreichend genaue Abbildung bezüglich der relevanten Merkmale notwendig
  - Weglassen von Details (Abstraktion oder Idealisierung)
- Schwierigkeiten
  - Konflikt zwischen den Punkten der Anforderung an Modelle
  - Alle relevanten Merkmale oft „a priori“ nicht bekannt (Kapitel 7)
  - Auswirkungen von Idealisierung im Allgemeinem nicht klar

# 1.4 Herleitung von Modellen



- Was genau soll modelliert werden?
  - Bewegungsmöglichkeit eines Roboterarms (Geometrie, Kinematik) oder die dabei wirkenden / notwendigen Kräfte (Kinetik)
  - Durchsatz durch ein Rechnernetz oder die mittlere Durchlaufzeit eines einzelnen Pakets
- Welche Größen spielen dafür eine Rolle (qualitativ) und wie groß ist ihr Einfluss (quantitativ)?
  - Optimale Flugbahn des Space Shuttle (Gravitation des Mondes)
- Im Allgemeinen nicht offensichtlich (Expertise, Studien, Hypothesen)

# 1.4 Herleitung von Modellen



- Beziehung zwischen den als wichtig identifizierten Größen
  - Qualitativ
    - „wenn – dann“, Vorzeichen von Ableitungen, etc.
  - Quantitativ
    - Konkrete Größe der Abhängigkeiten

# 1.4 Herleitung von Modellen: Werkzeuge



- Algebraische Gleichungen und Ungleichungen

$$E = mc^2, \quad w^T x \leq 10$$

- (Systeme) gewöhnliche(r) Differentialgleichungen  
(eine unabhängige Variable, meist Zeit t)
  - $y''(t) + y(t) = 0$       Oszillation eines linearen Pendels
  - $y'(t) = y(t)$                 exponentielles Wachstum

# 1.4 Herleitung von Modellen: Werkzeuge



- (Systeme) partielle(r) Differentialgleichungen  
(mehr als eine unabhängige Variable)
  - $u_{xx} + u_{yy} = f$       für  $(x, y) \in \Omega$
  - $u = 0$                       für  $(x, y) \in \Delta\Omega$
- Verformung einer eingespannten Membran unter Last  $f$

# 1.4 Herleitung von Modellen: Werkzeuge

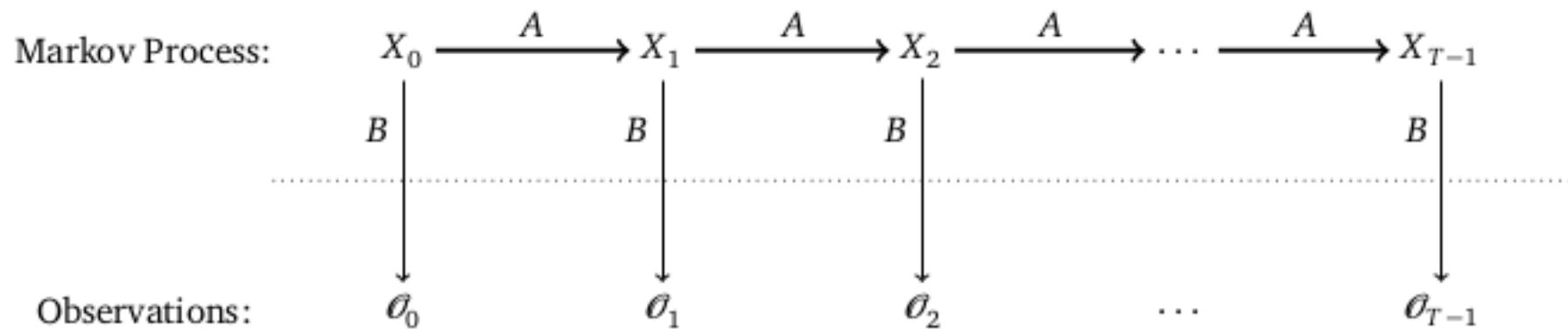


- Automaten, Zustandsübergangsdiagramme
  - Modellierung von Warteschlangen (Zustände: verschiedene Füllgrade; Übergänge: Ankunft bzw. Bearbeitungsende)
  - Modellierung von Texterkennung  
(Zustände: bisherige Struktur; Übergänge: neues Zeichen)
  - Modellierung von Wachstumsprozessen mit zellulären Automaten  
(Zustände: Gesamtbelegungssituation (Zellen voll, gefüllt, leer);  
Übergänge durch Regeln)

# 1.4 Herleitung von Modellen: Werkzeuge



- Beispiel für Automaten, Zustandsübergangsdiagramme
  - Hidden Markov Model



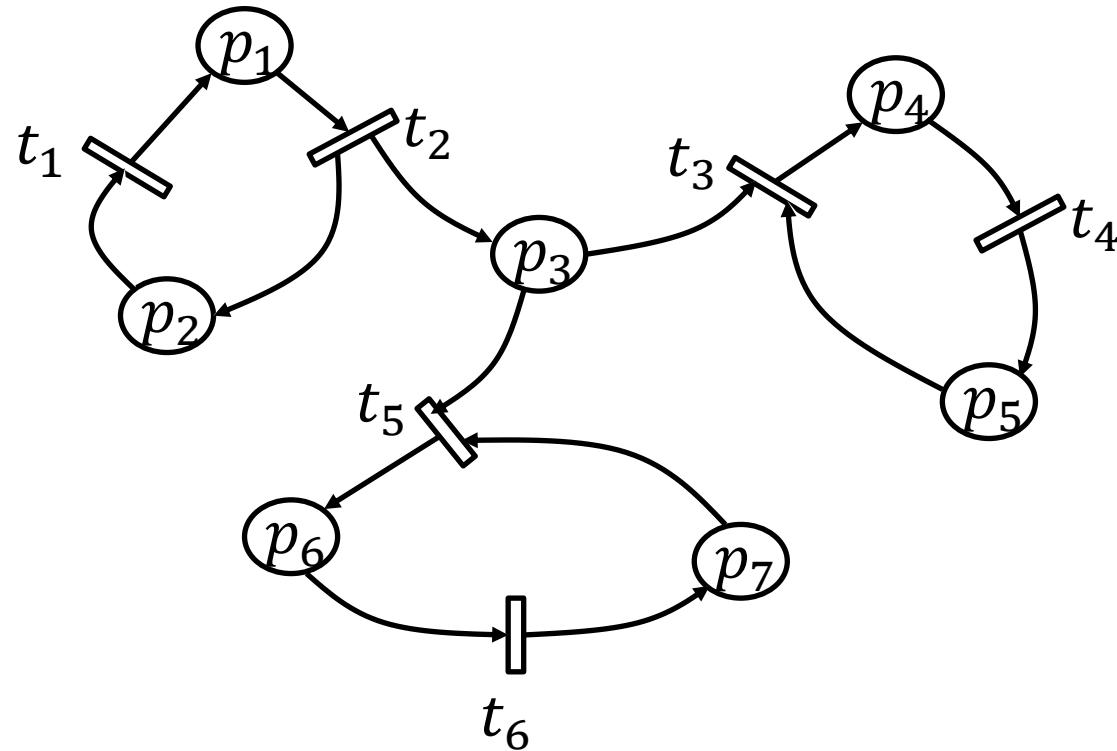
# 1.4 Herleitung von Modellen: Werkzeuge



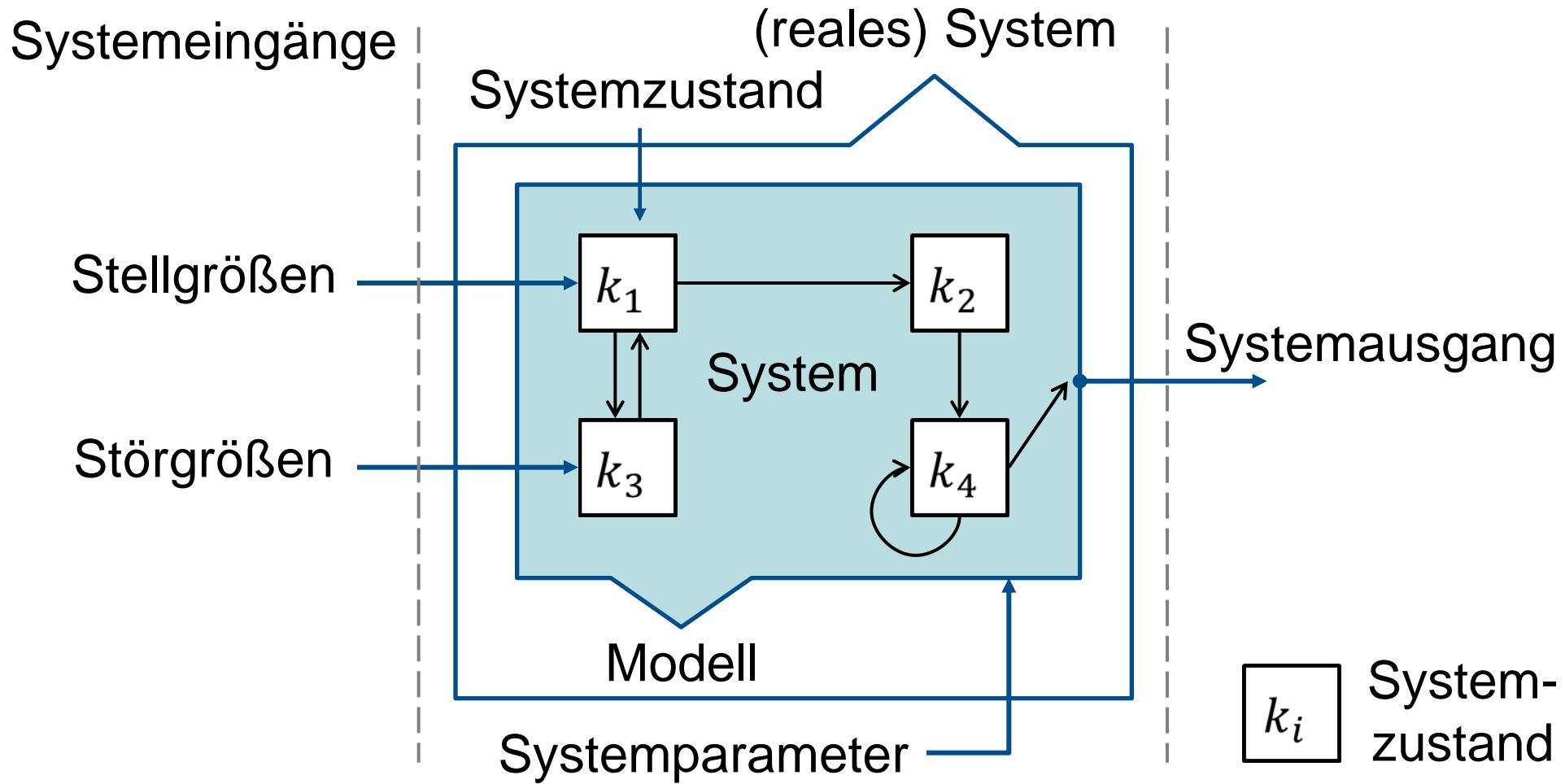
- Graphen
  - Modellierung von Rundreisen (5-Brücken Problem; Knoten: Orte; Kanten: Wege/Brücken)
  - Modellierung von Reihenfolgeproblemen (Knoten: Teilaufträge auf einer Maschine; Kanten: zeitliche Reihenfolge)
  - Modellierung von Rechensystemen (Komponenten und Kanäle)
  - Modellierung von Abläufen (Datenflüsse, work flow)

# 1.4 Herleitung von Modellen: Werkzeuge

- Beispiel für Graphen
  - Petrinetze (nächste Vorlesung mehr dazu)



## 1.4.1 (Reales) System und Modell



## 1.4.1 (Reales) System und Modell: Systemstruktur



- Menge der miteinander verbundenen Komponenten
- Komponenten zusammengesetzt aus Systemelementen ( $k_i$ )
- Systemelemente besitzen Attribute
- Änderungen der Attribute führen zu Änderungen in den Zustandsvariablen
- Systemzustand
  - Menge aller Zustandsvariablen

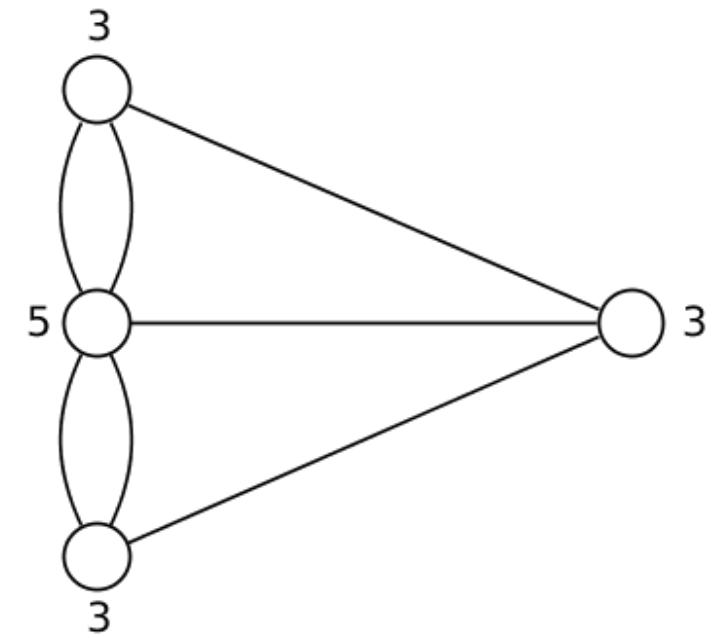
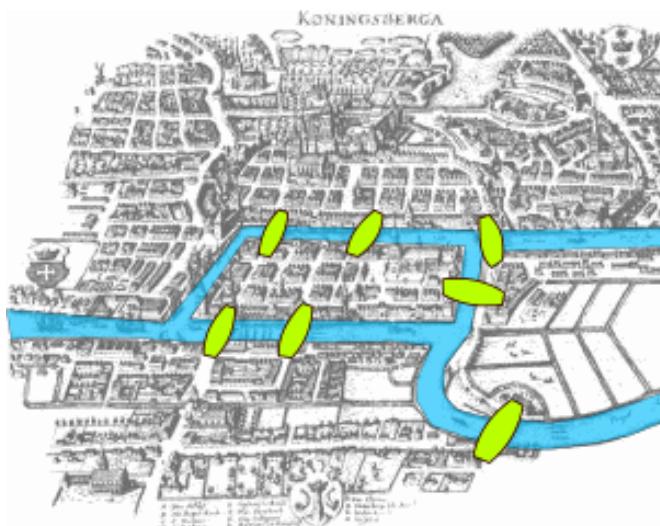
## 1.4.1 (Reales) System und Modell: Analyse

- Finde eine Lösung zu einem gegebenen Gleichungssystem
  - bestimmen einer gültigen Startlösung in der linearen Optimierung
- Finde die Lösung zu gegebenem Gleichungssystem
  - eindeutig lösbar gewöhnliche Differentialgleichung
- Löse beschränkte Extremalaufgabe
  - lineare Optimierung

## 1.4.1 (Reales) System und Modell: Analyse



- Löse Existenzaufgabe
  - eventuell nicht lösbar (Hamiltonscher Weg im Graphen)
- Ermittle Flaschenhals
  - Komponente maximaler Auslastung



## 1.4.1 (Reales) System und Modell: Modelleignung



- Verfügbarkeit der Eingabedaten (hinreichende Genauigkeit)
- Implementierungsaufwand
- Erforderlicher, absoluter Rechen- und Speicheraufwand
  - NP-vollständige Probleme
- Erforderlicher, relativer Rechen- und Speicheraufwand
  - konkurrenzfähiges Modell
- Empfindlichkeit
  - Verfälschung von Ergebnis durch (kleine) Störung
  - Rundungsfehler (bei Zwischenergebnissen)



Bitte jetzt auf Moodle eine Frage  
beantworten!

# 1.5 Lösungsansätze für mathematische Modelle

- Analytisch: Existenz- und Eindeutigkeitsnachweis sowie Konstruktion erfolgen formal / analytisch / direkt
  - Optimale Methode, da keine Vereinfachungen / Näherungen
  - Nur in einfachsten Spezialfällen möglich
- Heuristisch: „trial & error“ gemäß bestimmter Strategie
  - Nützlich vor allem bei Problemen der diskreten Optimierung
    - z.B. Rucksack-Problem: Greedy-Heuristik wählt immer lokal beste Alternative aus
  - Schwierigkeit: Konvergenz? Konvergenzgeschwindigkeit?

# 1.5 Lösungsansätze für mathematische Modelle

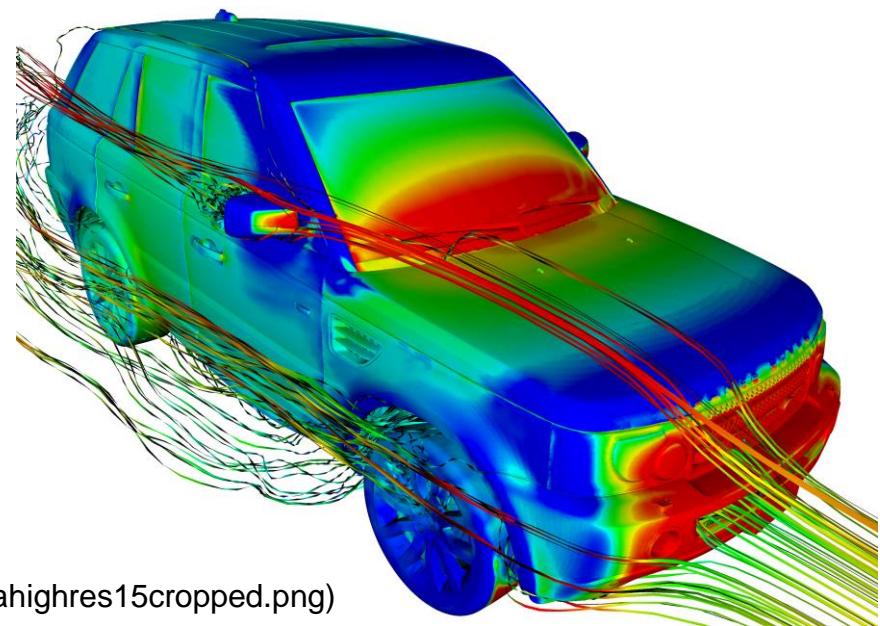


- Direkt-numerisch
  - Numerischer Algorithmus liefert exakte Lösung (mit Rundungsfehlern)
  - Keine Heuristik mehr, also erreichen des Ziels stets sichergestellt
    - z.B. direkte Lösung eines linearen Gleichungssystems ( $Ax = b$ )
- Approximativ-numerisch
  - (Iteratives) Näherungsverfahren für angenäherte (diskretisierte) Beziehungen (Gleichungen)
  - Erreichen einer (beliebig genauen) Approximation sichergestellt?
  - Hauptklassen bei der numerischen Simulation
    - Newton-Iterationsverfahren zur nichtlinearen Nullstellensuche

# 1.6 Bewertung von Modellen



- Validierung: „Stimmen Modell und Simulation?“
  - Vergleich mit Experimenten
    - „1:1 Experimente“ (Windkanal, Crashtest), ...
    - Laborexperimente an (verkleinerten) Prototypen;  
Problem: Skalierung sichergestellt?

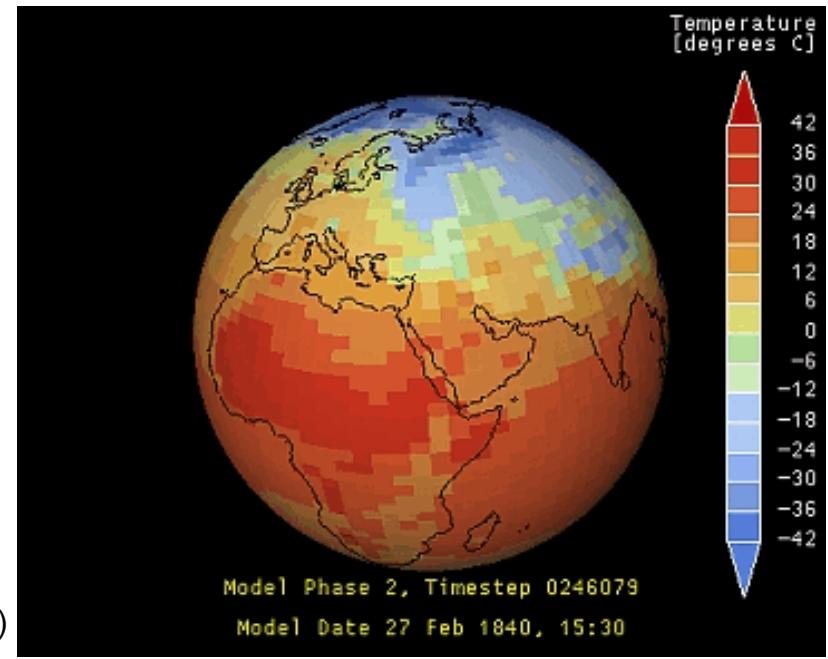


(Quelle Bild: <http://ww1.prweb.com/prfiles/2009/01/20/1126544/JaguarExahighres15cropped.png>)

# 1.6 Bewertung von Modellen



- Validierung: „Stimmen Modell und Simulation?“
  - A-posteriori Beobachtungen
    - Realitäts-Test (Wetter, Börse, militärische Szenarien)
    - Zufriedenheits-Test (Verkehrssteuerung, Beleuchtungsmodelle in der Computergraphik)



(Quelle Bild: <http://www.science-at-home.de/bilder/projekte/climateprediction.gif>)

# 1.6 Bewertung von Modellen



- Validierung: „Stimmen Modell und Simulation?“
  - Plausibilitäts-Test
    - Test der Simulationsergebnisse auf Konsistenz mit bestehenden Theorien (z.B. in Physik, Mechanik)
  - Modellvergleich
    - Vergleich der Ergebnisse mit auf unterschiedlichen Modellen basierenden Simulationen (Wettervorhersage)

# 1.6 Bewertung von Modellen



- Genauigkeit: „Wie präzise ist das Modell?“
  - Genauigkeit im Hinblick auf die Qualität der Eingabedaten
    - Bei Messdaten auf drei Stellen genau als Eingabe kann kein Resultat auf acht Stellen genau erwartet werden
  - Genauigkeit im Hinblick auf die Fragestellung
- Sicherheit: „worst case“ oder „average case“ Aussagen?

# 1.7 Klassifikation von Modellen



- Möglichkeit 1: diskret vs. kontinuierlich
  - Diskretes Modell nutzt diskrete / kombinatorische Beschreibung
    - Binäre oder ganzzahlige Größen
    - Zustandsübergänge in Graphen oder Automaten
  - Kontinuierliches Modell nutzt kontinuierliche / reellwertige Beschreibung
    - Reelle Zahlen, physikalische Größen
    - Algebraische Gleichungen, Differentialgleichungen

# 1.7 Klassifikation von Modellen



- Möglichkeit 1: diskret vs. kontinuierlich
  - beide Ansätze häufig sinnvoll
  - verwendeter Typ hängt von Fragestellung ab
  - Beispiel: Verkehrsfluss durch Darmstadt
    - diskret: Anzahl der Autos im System, an Ampeln, Warteschlagen
    - kontinuierlich: Dichten, Flüsse (Flüssigkeitsmodelle mit Kanälen, Sperren)

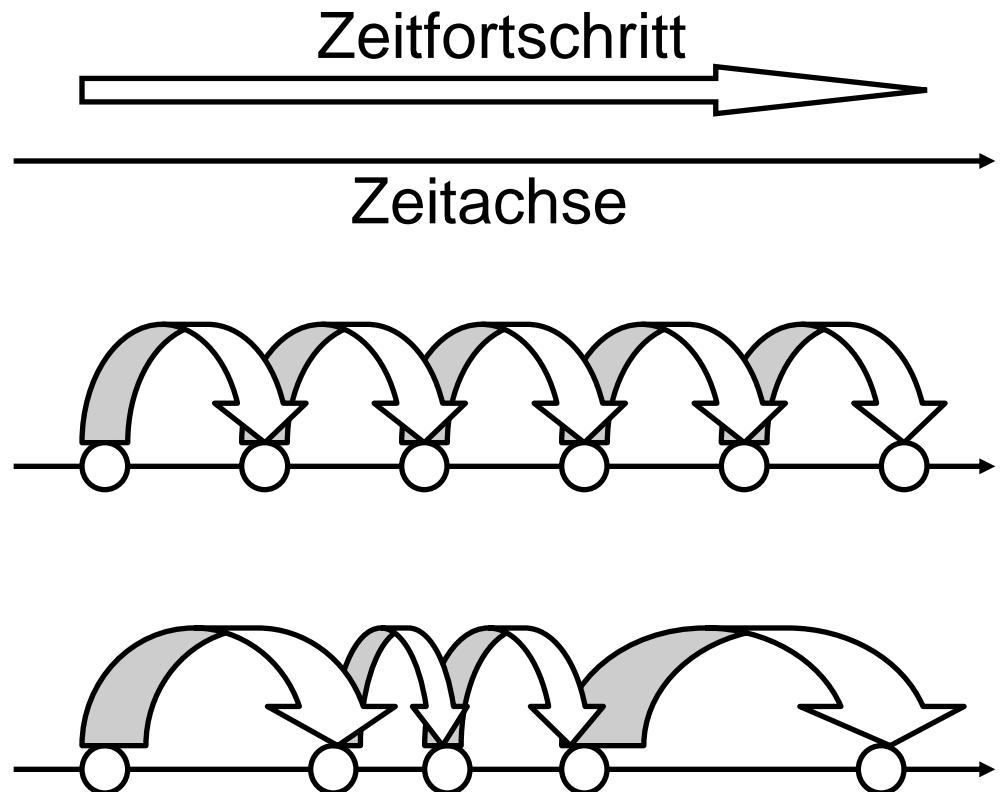
# 1.7 Klassifikation von Modellen



- Möglichkeit 2: deterministisch vs. stochastisch
  - Kein zwingender Bezug zwischen dem zu modellierenden Phänomen und dem Werkzeug
  - Beispiel: Würfeln
    - offensichtliches Zufallsexperiment (d.h. probabilistische Realität)
    - sinnvollerweise auch stochastisches Modell (Zufallsvariable)

# 1.7 Klassifikation von Modellen: Zeitachse

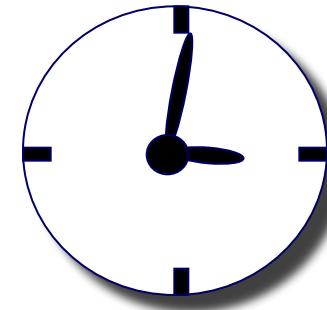
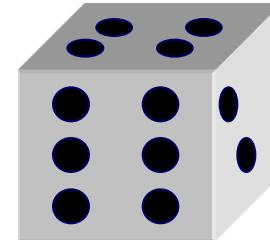
- kontinuierlich
- diskret äquidistant
- diskret nicht äquidistant



# 1.7 Klassifikation von Modellen: Zustandsraum



- diskrete Werteskala
- kontinuierliche Werteskala
- räumlich verteilte Variablen



# 1.7 Klassifikation von Modellen: Dynamik und Zustandsübergänge



- Dynamik
  - Instationär (dynamisch)
    - z.B. ein fahrendes Fahrzeug

$$x' = f(x)$$

- Stationär
  - 0 = f(x)
- Zustandsübergänge
  - deterministisch (z.B. Schiffschaukel)
  - stochastisch

# 1.7 Klassifikation von Modellen: Unterteilung dynamischer Modelle



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

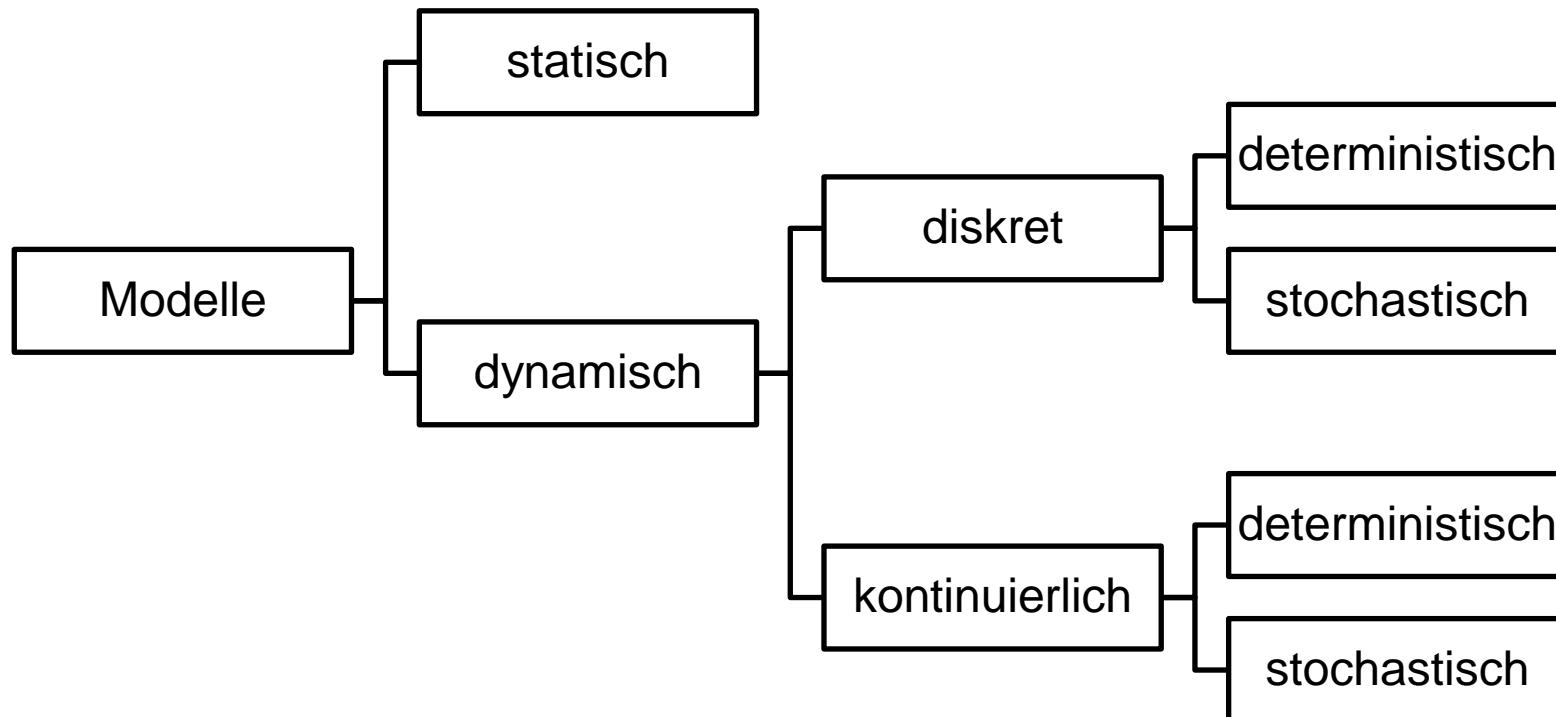
- Zeitdiskret
  - Werte der Zustandsvariablen ändern sich atomar alle  $\Delta\tau$  Zeiteinheiten
- Ereignisdiskret
  - Werte der Zustandsvariablen ändern sich atomar durch das Eintreten eines Ereignisses (nach Verweilzeit)
- Zeitkontinuierlich
  - Werte der Zustandsvariablen ändern sich kontinuierlich

# 1.7 Klassifikation von Modellen: Unterteilung dynamischer Modelle



- Zustandsübergänge erfolgen deterministisch oder stochastisch
- Zustandsübergang kann abhängig vom bisherigen Zustandsverlauf (Gedächtnis) oder nur vom aktuellen Zustand sein
- Verweilzeit: deterministisch oder stochastisch; u.U. mit Einfluss auf Zustandsübergang

# 1.7 Klassifikation von Modellen: Überblick



# 1.8 Ansätze zur Modellbildung

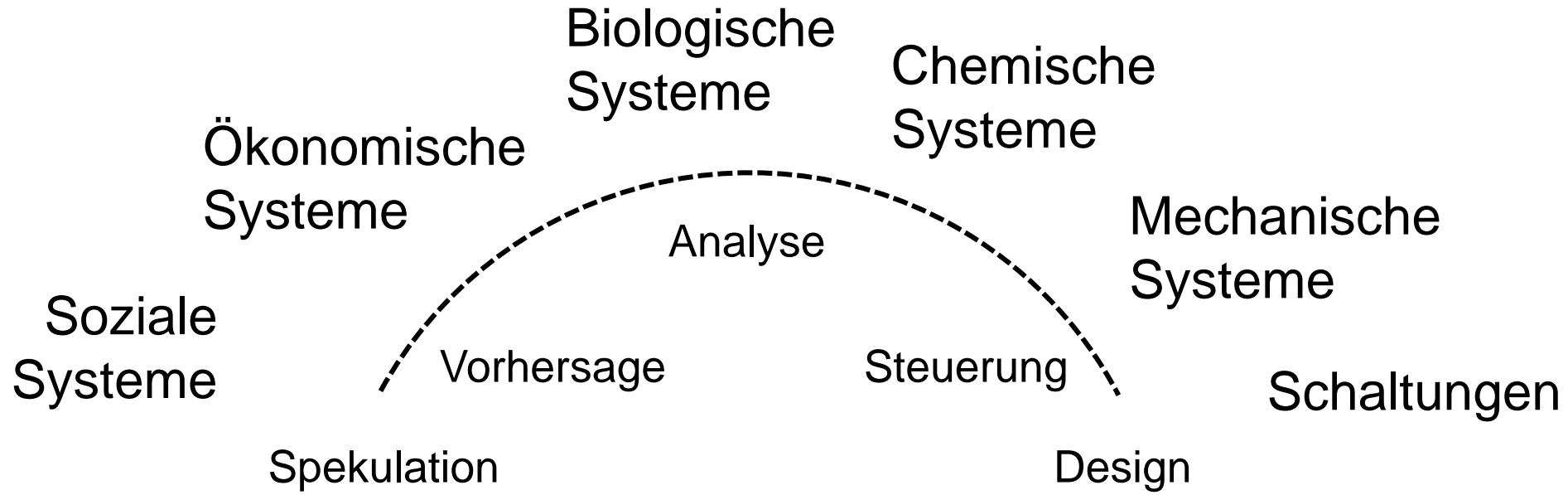


- Black-Box-Sichtweise
  - Modellbildung auf Basis des beobachteten Verhaltens am Ausgang (in Abhängigkeit vom Eingang)
  - Induktives Vorgehen bei der Modellierung
- White-Box-Sichtweise
  - Modellbildung auf Basis der Systemstruktur
  - Deduktives Vorgehen bei der Modellierung

# 1.8 Ansätze zur Modellbildung



- Regenbogen von Karplus



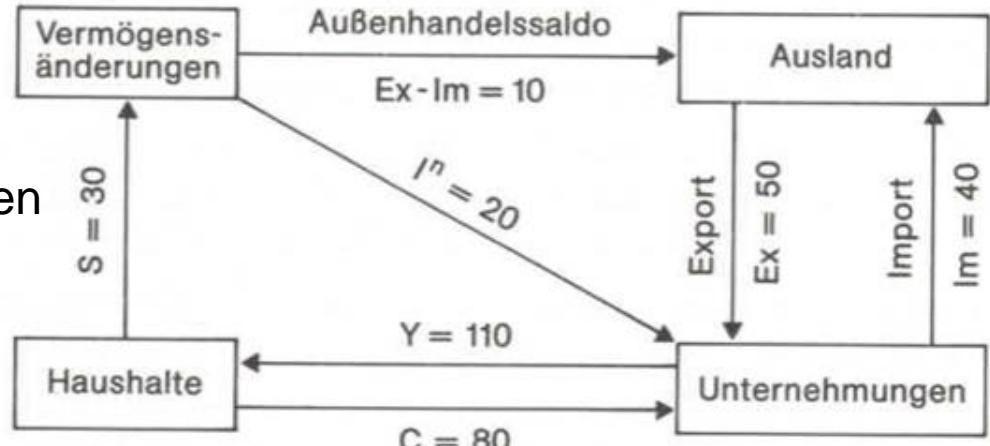
# 1.8 Ansätze zur Modellbildung



- Soziale Systeme
  - Modellbildung zu Verhaltensanalysen

- Ökonomische Systeme
  - Modellbildung von Wirtschaftssystemen
  - Finanzmodellen

- Chemische Systeme
  - Abstraktion von Elementen
  - Beispiel:  $2K Mn O_4 + 16H Cl \rightarrow 2Mn Cl_2 + 2K Cl_2 + 8H_2O$

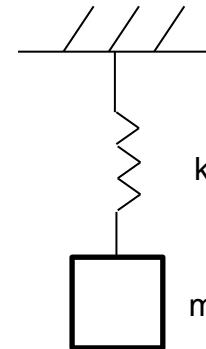


# 1.8 Ansätze zur Modellbildung



- Mechanische Systeme

- $m\ddot{x} + kx = 0$

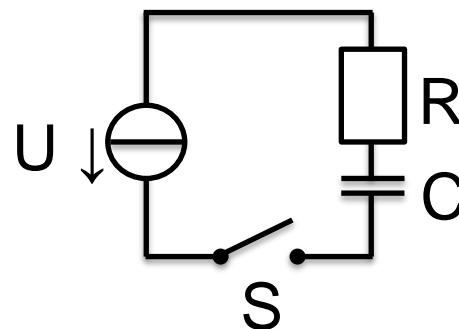


- Biologische Systeme

- Populationen von Tieren (z.B. Fuchs & Hase)
  - Regelkreisläufe im Körper (z.B. Insulin)
  - Neuronen (z.B. Hodgkin-Huxley Modell)

- Elektrischen Schaltungen

- $U = RI \text{ und } I = C\dot{U} \Rightarrow \dot{U} = \frac{U}{RC}$



Bitte jetzt auf Moodle eine Frage  
beantworten!

# Heutige Lernziele: Kernfragen



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

- Was sind die Grundbegriffe der Simulation?
- Was bedeuten Zustand, System und Modell?
- Wofür kann man Simulationen einsetzen?
- Wie läuft eine Simulationstudie an?
- Was bedeuten Problemspezifikation, Modellierung, Implementierung, Validierung und Simulationsanwendung?
- Wie leitet man Modelle her?
- Wie kann man Modelle klassifizieren?

**Selbsttest: Können Sie diese Fragen beantworten? Wenn nicht, schnell nochmal das Video anschauen!**