

COMPUTERORIENTIERTE MATHEMATIK I



Christof Schütte

Wintersemester 2018/19

Computer-orientierte Mathematik

Einführende Vorlesung - Christof Schütte

19.10.18

WILLKOMMEN!

http://numerik.mi.fu-berlin.de/wiki/WS_2018/CoMaI.php

Vorlesung	Fr, 12 - 14 Uhr	T9/Gr. Hörsaal	Prof. Dr. Christof Schütte
Übung	Mo, 10 - 12 Uhr	A3/SR 024	Raphael
Übung	Mo, 12 - 14 Uhr	A3/SR 119	Raphael
Übung	Di, 12 - 14 Uhr	A6/SR 032	Raphael
Übung	Mi, 10 - 12 Uhr	A3/SR 119	Felix
Übung	Mi, 12 - 14 Uhr	A3/SR 119	Adrian
Übung	Fr, 8 - 10 Uhr	A6/SR 007/008	Adrian
Übung	Fr, 14 - 16 Uhr	A3/SR 019	Felix
Klausur	Fr, 15.2.2019	tbd	
Nachklausur	tbd	tbd	

BSc und MSc-Studiengänge der FU:
Anmeldung im Campus-Management erforderlich

Bitte melden Sie sich im kommentierten Vorlesungsverzeichnis für
eine der Übungen an.

- ▶ Jede Woche nach der Vorlesung gibt es einen neuen Übungszettel
- ▶ Abgabe der Lösungen: Am Montag 10 Tage später
- ▶ Abgabegruppen: ≥ 2 und ≤ 3
- ▶ Theorie und Programmieraufgaben!
- ▶ Aktive Teilnahme
= mindestens 50 % der Übungspunkte + Vorrechnen
- ▶ Weitere Informationen auf der Vorlesungswebpage

- ▶ Aktive Teilnahme nachgewiesen → Klausurteilnahme
- ▶ Klausur (15.2.19) + Nachklausur
- ▶ Note ergibt sich allein aus dem Klausurergebnis
- ▶ Wer regelmäßig teilnimmt, hat 90 % Chance auf Bestehen der Klausur
- ▶ Weitere Informationen auf der Vorlesungswebpage

- ▶ Vorlesungsskript
- ▶ Folien, jeweils Freitags nach der Vorlesung
- ▶ Einführung in MATLAB

- ▶ Mut zur Lücke!
- ▶ COMA ist kein Schulfach!
- ▶ Stellen Sie Fragen!

Kondition eines Problems

Stabilität eines Algorithmus

Berechne das Polynom $f(x) = x^3 + 12a^2x - 6ax^2 - 8a^3 = (x - 2a)^2$
mit $a = 4\,999\,999$ an der Stelle $x_0 = 10\,000\,000$.

MATLAB: » $a = 4999999$;

» $x_0 = 10000000$;

» $f = x_0 * x_0 * x_0 + 12 * a * a * x_0 - 6 * a * x_0 * x_0 - 8 * a * a * a$

» $f = 393216$

Analytisch korrekt: $f(x_0) = 8$

Komplexität eines Problems

Theorem: Zu jedem Algorithmus gibt es eine Reihenfolge der n Zahlen, für die er mindestens $c n \log(n)$ Vergleiche benötigt!

Effizienz eines Algorithmus'

Algorithmus: bubbleSort:

Es gibt eine Reihenfolge der n Zahlen,
für die bubbleSort mindestens Cn^2 Vergleiche benötigt!

Algorithmus: mergeSort:

Für alle möglichen Reihenfolgen der n Zahlen, benötigt mergeSort
höchstens $Cn \log(n)$ Vergleiche.