

Modulverantwortliche/r Prof. Dr. Lena Oden

Dauer des Moduls
ein Semester

ECTS
10

Workload
300 Stunden

Häufigkeit
in jedem Sommersemester

Lehrveranstaltung(en) Fachpraktikum Scientific Programming in Python

Detaillierter Zeitaufwand Präsenzphase am Ende des Semesters: 25 Stunden
Vorbereitung und Einarbeitung: 50 Stunden
Bearbeitung der vorgegebenen Programmieraufgaben: 100 Stunden
Bearbeitung eines Projektes im Team: 100 Stunden
Dokumentation der Ergebnisse: 25 Stunden

Qualifikationsziele Nach dem Abschluss des Praktikums sind die Studierenden in der Lage, mathematische, vor allem numerische Probleme, effizient mit Python zu lösen. Dies umfasst das Arbeiten mit Matrizen und Vektoren, das Lösen von Gleichungssystemen, Datenanalyse und die ansprechende Visualisierung der Ergebnisse. Sie kennen die Werkzeuge, wie diese Probleme auf moderne Hardware effizient gelöst werden können, und sind in der Lage, die Lösung auch einfach zu optimieren und zu parallelisieren. Sie kennen die wichtigsten Pakete zum effizienten arbeiten mit Python, wie numba, scipy und Pandas. Sie haben gelernt, umfangreiche Programmieraufgaben in Team zu bearbeiten.

Inhalte Unter Scientific Computing versteht man die Umsetzung numerischer Algorithmen in eine Programmiersprache, um wissenschaftliche Probleme zu lösen. Für viele wissenschaftliche Algorithmen sind Geschwindigkeit und auch der Speicherverbrauch sehr wichtig. Reines Python hat jedoch einen hohen Speicherverbrauch und ist vergleichsweise - langsam, es bietet jedoch viele nützliche Erweiterungen an, mit deren Hilfe sich numerische Probleme effektiv lösen lassen.

In dem Praktikum soll die effektive Nutzung dieser Pakete für wissenschaftliches Arbeiten erlernt werden. Dabei wird es zunächst schwerpunktmäßig um Numpy, Scipy, Matplotlib und Pandas gehen. Dabei wird es um die Lösung numerischer Probleme, die Analyse großer Datenmengen und das Lösen von Simulations-Aufgaben gehen. Ein weiterer wichtiger Teil des Praktikums ist die ansprechende Visualisierung der Ergebnisse.

Im zweiten Teil des Praktikums geht es dann um Numba und mpi4py, welche eine effiziente Parallelisierung der Algorithmen ermöglichen.

Inhaltliche Voraussetzung Grundlegende Programierkenntnisse
Grundlegende Kenntnisse in Numerik und Analysis

Lehr- und Betreuungsformen Studententag/e
Betreuung und Beratung durch Lehrende
Zusatzmaterial

Anmerkung Für die Teilnahme an dieser Veranstaltung ist ein gesondertes Anmeldeverfahren im Vorsemester über folgenden Link erforderlich: <https://webregis.fernuni-hagen.de>.

Formale Voraussetzung mindestens zwei Wahlmodulprüfungen müssen bestanden sein

Verwendung des Moduls B.Sc. Informatik
M.Sc. Informatik
M.Sc. Praktische Informatik
M.Sc. Wirtschaftsinformatik

Prüfungsformen

Prüfung

Stellenwert
der Note 1/8

Art der Prüfungsleistung

benotete Praktikumsteilnahme:
Ausarbeitung und Vortrag

Voraussetzung

keine