

Inhaltsverzeichnis

1.1	Ziele und Lösungsvorschläge	1
1.2	Moodle-Kurs	2
1.3	Modulabschlussprüfung	4
1.4	Software	4
1.5	Zusätzliches Lehrmaterial	4
1.6	Quellen	5

Angewandte Mathematik im Sommersemester 2025

1.1 Ziele und Lösungsvorschläge

Das Ziel dieser Lehrveranstaltung ist mit Ihnen zusammen einen Werkzeugkasten mathematischer Algorithmen zu erarbeiten. Der Werkzeugkasten soll Sie für Ihr künftiges Berufsleben so ausstatten, dass Sie für verschiedenste mathematische Fragestellungen eine Lösungsmethode parat haben.

Über das Semester hinweg werden wir Modelle zur Beschreibung physikalischer Zustände oder Prozesse analysieren und anhand der Modelle konkrete mathematische Aufgabenstellungen formulieren. Die Aufgabenstellungen an sich sind Ihnen höchstwahrscheinlich aus den Grundvorlesungen in Mathematik bekannt, nämlich:

- Lösen von Gleichungen
- Lineare Gleichungssysteme
- Curve Fitting (Ausgleichsprobleme oder Interpolation)
- Optimierung (Extremwertberechnung)
- Integration
- Anfangswertprobleme

Neu ist vielleicht, dass die Problemstellung aus unserem realen Leben gegriffen ist. Neu ist auch, dass die Problemstellungen im allgemeinen so schwierig oder komplex sind, dass wir sie mit den gewöhnlichen Mitteln "Stift und Papier" einfach nicht mehr in den Griff bekommen - zwei Beispiele:

- Die Dimension der Matrizen, die wir in linearen Gleichungssystemen betrachten, ist viel, viel größer als 3 oder 4.
- Die Gleichungen, die wir lösen müssen, sind nicht mehr durch geschlossene Formeln lösbar, beispielsweise $\sin(x^2) = \sqrt{\ln(x)}$ oder $3 \cdot x^5 - \sqrt{2} \cdot x^3 - 7 = 0$.

Anstatt Stift und Papier, setzen wir den Computer ein, um die Aufgaben zu lösen. Lösen ist an der Stelle nicht ganz korrekt: die Aufgaben werden approximativ gelöst. "Computer" und "approximativ" sind Schlagworte, die Fragen aufwerfen, beispielsweise

- Wie lautet eine geeignete numerische Lösungsmethode?
- Wie kann man die Methode algorithmisch realisieren?
- Wie kontrolliert man, dass der Algorithmus auch funktioniert?
- Wie interpretiert man das Ergebnis?

Ein zentraler Baustein dieser Lehrveranstaltung ist das Programmieren und das kritische Analysieren von Algorithmen. Der andere Baustein ist das Nutzen von fertigen Software-Lösungen: für viele Fragestellungen gibt es fertige, numerisch robuste Software-Lösungen und die sollte man auch benutzen. Die Programmiersprache bzw. die Software, die wir während des Semesters einsetzen, ist **Matlab**.

Um die Lernziele zu erreichen, warten folgende Lehrangebote auf Sie:

- In den wöchentlichen Lehreinheiten werden wir die Algorithmen zusammen erarbeiten und beispielhaft programmieren.
- Es gibt ein vorlesungsbegleitendes Skript und Übungsmaterial, das Ihnen zur selbstständigen Vor- und Nacharbeit aller Lehreinheiten dient.

- Es gibt eine wöchentliche Leistungskontrolle in Form von zwei elektronischen Tests. Die Tests dienen dazu Sie auf die Modulabschlussprüfung vorzubereiten:
 1. Der **Theorie** Test dient dazu Ihre mathematischen Grundlagen zum aktuellen Thema zu trainieren: Sie rechnen Aufgaben und wenden so den aktuellen Stoff theoretisch an.
 2. Der **Coding** Test dient dazu Ihre Programmierkenntnisse zum aktuellen Thema zu trainieren: Sie setzen sich aktiv mit den Algorithmen auseinander und programmieren sie oder wenden sie an.

1.2 Moodle-Kurs

Moodle als zentrale Lernplattform der HTW Berlin gibt Ihnen Zugang zu dem modulbegleitenden Moodle-Kurs Angewandte Mathematik. Unser Moodle-Kurs ist Ihr Dreh- und Angelpunkt, um Materialien abzugreifen, Tests zu bearbeiten oder Kursnachrichten einzusehen.

1. Moodle-Kurs ist unser Informationsraum:
 - Über die Lernplattform Moodle sehen Sie alle Kursnachrichten, die ich an Sie versendet habe.
 - Über die Lernplattform Moodle erhalten Sie Überblick auf die Themen der aktuellen Woche.
 - Über die Lernplattform Moodle können Sie über "Nachrichten" mit mir Kontakt aufnehmen.
 - Über die Lernplattform Moodle sehen Sie den Bearbeitungsstatus aller zulassungsrelevanten Tests und Aufgaben.
2. Moodle-Kurs ist unser **Lernraum**:
 - Sie haben im Lernraum Zugang zu aktuellen Aufgaben, Kurzschriften und den Mitschriften aus der Lehreinheiten und zu Übungsblättern:

- Die **Kurzschriften** ermöglichen Ihnen die Lehrinhalte gezielt nachzulesen.
- Jedes Skript ist kurz und knackig gehalten und ersetzt kein Lehrbuch.
- Die Algorithmen sind nummeriert, so dass darauf Bezug genommen werden kann.
- Algorithmen sind **blau-umrahmt**.
- Wichtige Aussagen sind **grau-umrahmt**.



- Alle **Mitschriften** werden in einem OneNote-Notizbuch gesammelt.
- Sie haben während des Semesters Zugang zu unserem OneNote-Notizbuch.



- Zu jedem Thema gibt es ein passendes **Übungsblatt**.
- Die Aufgaben auf den Übungsblättern entsprechen inhaltlich den Aufgaben in den Leistungstests.
- Die Coding-Aufgaben auf den Übungsblättern sind prototypische Prüfungsaufgaben.



- Neben diesem Lehrmaterial gibt es Links und Hinweise zu weiteren kostenlosen Lehrangeboten.
3. Moodle-Kurs ist unser **Leistungsraum**: über die Lernplattform Moodle greifen Sie auf Ihre Tests zu.
 - Die **Aufgaben** sind prototypische Klausuraufgaben



- In jeder Woche gibt es während der Begleitübung ein bis zwei **Aufgaben**.
- Die **Aufgaben** sind kleine Programieraufgaben, in denen Matlab-Routinen angewendet werden.
- Die Aufgaben werden **während der Lehreinheit** von Ihnen bearbeitet.
- Die Bearbeitung und Abgabe erfolgt über einen VDI-Pool.
- Ihre Lösung soll als Datei (meist png) am Ende der entsprechenden Lehreinheit über die Moodle-Seite hochgeladen werden.
- Geben Sie die Aufgabe am Ende der Stunde ab, erhalten Sie einen Punkt.

- Die **Theorie Tests** testen, ob Sie die für das Thema relevanten mathematischen Grundlagen wissen.



- Pro Woche gibt es maximal einen **Theorie Test** mit einer Bearbeitungszeit von einer Woche.
- Jeder **Theorie Test** besteht aus maximal 8 Aufgaben (MC, Drag-And-Drop oder Stack-Plugin).
- Für jeden Test gibt es maximal 10 Punkte, wobei die Aufgaben unterschiedlich gewichtet sind.
- Innerhalb des Bearbeitungszeitraums haben Sie für jeden Test maximal zwei Versuche. Der bessere Versuch zählt.
- Nach Abgabe eines Versuchs wird die richtige Lösung angezeigt.
- Nachdem der Test geschlossen ist, wird für die meisten Aufgaben ein detaillierter Lösungsweg angezeigt.

- Die **Coding Tests** testen, ob Sie die vorgestellten numerischen Algorithmen programmieren und anwenden können.



- Pro Woche gibt es maximal einen **Coding Test** mit einer Bearbeitungszeit von einer Woche.
- Jeder **Coding Test** besteht aus einer Programieraufgabe, die das VPL-Plugin von Moodle nutzt.
- Die Programmieraufgabe lösen Sie zu hause. Sie können hierfür Matlab oder Octave nutzen - beides funktioniert.
- Für die richtige Bearbeitung der Aufgabe bekommen Sie maximal 100 Punkte.
- Nachdem Sie die Aufgabe über die Moodle-Maske hochgeladen haben, müssen Sie Ihren Code selbst evaluieren. Wichtig: das geschieht nicht automatisch!
- Ist der Code fehlerhaft, dann erhalten Sie eine entsprechende Information. Ihre Aufgabe ist es den Fehler in dem Code zu suchen und zu korrigieren.
- Sie können Ihren Code bis zu 10 Mal erneut hochladen und wieder evaluieren.
- Ab der zweiten Evaluation werden pro Evaluation 10 Punkte abgezogen.
- Läuft die Evaluation nicht fehlerfrei durch, nachdem der Test geschlossen ist, erhalten Sie keine Punkte.

1.3 Modulabschlussprüfung

In den Prüfungszeiträumen wird eine schriftliche Prüfung in Präsenz angeboten. Die Prüfungsdauer beträgt 90 Minuten. In Form eines Moodle-Tests erhalten Sie Programmieraufgaben und Aufgaben mit freier Texteingabe. Als Hilfsmittel zugelassen ist ein zwei-seitig handbeschriebenes Blatt Papier und das Benutzen der inline Matlab Hilfe. Die Prüfung ist nicht open-book. Um diese Rahmenbedingungen sicher zu stellen, wird an virtuellen Maschinen eines VDI-Pools unter Verwendung des Safe Exam Browsers durchgeführt.

Für die Teilnahme an der Modul-Abschlussprüfung gibt es eine **Zulassungsvoraussetzung**:

1. Es wird verlangt, dass Sie am Ende des Semesters 50% der **Aufgaben** abgegeben haben.
2. Es wird verlangt, dass Sie am Ende des Semesters 50% der maximal erreichbaren Punkte in den **Theorie** Tests haben.
3. Es wird verlangt, dass Sie am Ende des Semesters 50% der maximal erreichbaren Punkte in den **Coding** Tests haben.

Wichtig: Wenn Sie an einem Prüfungstermin durch Krankheit nicht teilnehmen können, dann wird für Sie kein Ersatztermin angeboten.

1.4 Software

Für die Veranstaltung nutzen wir Matlab. Um die Programmieraufgaben zu bearbeiten steht uns über das gesamte Semester hinweg der VDI-Pool **FB1 NAWI PoolA** zur Verfügung. Die Abkürzung VDI steht für Virtuelle Desktop Infrastruktur und VDI bedeutet, dass virtuelle Rechner zur Verfügung stehen, auf die man sich von dem eigenen PC oder Laptop oder einem der Rechner im Seminarraum aus einzuloggen und dann auf der virtuellen Maschine arbeiten kann. Matlab muss nicht auf dem eigenen Rechner installiert sein.

- Hinweise zur Nutzung des VDI-Pools finden Sie [hier](#).

Möchten Sie Matlab auf Ihrer eigenen Hardware installieren, dann ist das auch möglich:

- [Matlab HTW Berlin \(Software und Installationshinweise\)](#)

Eine kostenlose Alternative zu Matlab ist Octave. Die Funktionalitäten von Octave sind hinsichtlich dessen, was im Modul gebraucht wird, ausreichend:

- [Octave \(Software und Installationshinweise\)](#)

1.5 Zusätzliches Lehrmaterial

Angewandte Mathematik:

- [Numerische Mathematik](#) von Michael Knorrenschild erschienen im Hanser-Verlag.
- [Numerische Mathematik 1](#) von Josef Stoer erschienen im Springer-Verlag.
- [Programmierung numerischer Algorithmen](#) Vorlesungsskript von Prof. Dr. Steffen Börm.
- [Applied numerical Methods with Matlab for Engineers and Scientists](#) von Steven Chapra erschienen in McGraw-Hill Verlag.

Mathematik für Ingenieure:

- [Höhere Mathematik kompakt](#) von Georg Hoever erschienen im Springer-Verlag.
- [Mathematik für Ingenieure und Naturwissenschaftler Band 1](#) von Lothar Papula erschienen im Springer-Verlag.
- [Mathematische Methoden in der Physik](#) von Christian Lang und Norbert Pucker erschienen im Springer-Verlag.
- [Das Gelbe Rechenbuch Band 1](#) von Peter Furlan erschienen im Martina Furlan Verlag.

1.6 Quellen

Zur Erstellung des vorlesungsbegleitenden Materials wurden folgende Quellen genutzt:

1. Das vorlesungsbegleitende Skript ist von Lucy Weggler.
2. Autoren der Übungsaufgaben auf den Übungsblättern sind
 - Lucy Weggler (HTW Berlin)
 - Andreas Zeiser (HTW Berlin)
3. Autoren der Moodle-Aufgaben sind
 - Lucy Weggler (HTW Berlin)
 - Andreas Zeiser (HTW Berlin)
 - (ART - EST - GRP - DIG) HE STEM Programme
 - Calculus Refresher: Dr. A. Croft and Dr. A. Kay (Loughborough University)
 - Tony Barnard, A pocket map of algebraic manipulation