

Projektarbeiten als Modulprüfung in der Computerphysik

Maher Valentino Chikho, Sebastian Meures

Universität zu Köln, Fachschaft Physik

Einleitung

Physiker*innen beschäftigen sich nicht nur mit dem Schubsen von Formeln und Lösen von Integralen oder mit dem Bauen von Apparaturen. Vor allem in der Theorie beschäftigen sich Physiker*innen auch mit Simulationen, welche eigenhändig konzipiert und programmiert werden. Ebenso erfordern Apparaturen in der heutigen Experimentalphysik eigens geschriebene Programme. In der heutigen Physik kommt ein weiterer Punkt hinzu, der über das klassische Programmieren hinaus geht: Methoden des Data-Science eröffnen in vielen Bereichen der Physik ganz neue Zugänge. An der Universität zu Köln lernt man dementsprechend nicht nur das reine Programmieren, sondern bringt es in einen Zusammenhang mit der Physik. Um dies zu ermöglichen, findet die Veranstaltung anders als an vielen anderen Unis auch erst im fortgeschrittenen Ba-Studium statt.

Es werden in den Übungen Aufgaben bearbeitet, die einen Bezug zur Physik haben. Diese werden mit Hilfe aus der Vorlesung bekannter Algorithmen, z.B. zum Lösen von Differentialgleichungen oder Eigenwertproblemen, bearbeitet. In den Sommersemestern 2021/22 wurde dabei ein Pilotversuch durchgeführt, bei dem es statt einer Klausur eine Projektarbeit gab, die in einem Zeitraum von zwei Wochen bearbeitet und danach abgegeben und bewertet wurde. Dies sollte dazu dienen, sowohl die Programmierkompetenzen abzuprüfen als auch das durch die Vorlesung erlangte theoretische Wissen. Gleichzeitig ist eine Projektarbeit natürlich mehr als nur eine Prüfung.

Das Modul

An der Universität zu Köln wird im Bachelorcurriculum vorgeschrieben, dass Studierende den Kurs Computerphysik zu absolvieren haben. Das Modul ist keine Voraussetzung für andere Module und besitzt auch selbst keine Voraussetzungen. Der Kurs ist für das vierte Semester angesetzt, wodurch sicher gestellt werden soll, dass die Studierenden grundlegende mathematische Methoden vor allem in der Linearen Algebra beherrschen. Der Kurs ist so konzipiert, dass Menschen ohne Programmierkenntnisse diesen absolvieren können. Das Modul setzt sich aus einer Vorlesung, einem Programmiertutorial und einer begleitenden Übung mit wöchentlichen Abgaben zusammen die auch zulassungsrelevant sind. Am Ende wurde das Modul zuvor von einer Klausur und während des Pilotprojekts mit einer Projektarbeit abgeschlossen, die auch benotet wurde.

Die Veranstaltung unmittelbar vor dem Start dieses Pilotprojektes fand Corona-bedingt online statt und ist vollständig öffentlich dokumentiert: <https://www.thp.uni-koeln.de/trebst/Lectures/2020-CompPhys.shtml>

Vorlesung

In der Vorlesung werden zum einen numerische Methoden zum Lösen von mathematischen Problemen, u.a. numerisches Lösen von Differenzialgleichungen, Matrixrechnung, also allgemeiner das Kennenlernen von diversen wichtigen und nützlichen Algorithmen thematisiert. Oft wird dabei

Bezug zur Physik genommen. Zum anderen erhält man einen Einblick in die Kontexte um das Programmieren. Gemeint sei so viel wie: Wie funktioniert ein Computer? Wie kann man Runtime verkürzen und wieso ist das sinnvoll? Was bedeutet Zufall in einem numerischen Sinne?

Tutorial

Neben Vorlesung und Übung gibt es ein „Programmirtutorial“, wo die Studierenden das Programmieren von Grund auf erklärt bekommen. Anfangs wird ihnen erklärt, welche Programmiersprache benutzt wird. In Köln wird die Sprache „Julia“ verwendet, da diese effizient für Berechnungen und Data-Science ist; außerdem hat sie starke Ähnlichkeiten zu Python, was in der Wissenschaft sehr verbreitet ist. Im Tutorial 0 wird den Studierenden gezeigt, wie man auf einem Endgerät Julia installiert. Dies wird als positiv empfunden, da man davon ausgeht, dass Menschen sich noch nie mit dem Programmieren auseinandergesetzt haben. Weiter geht es dann mit den Grundlagen wie dem Definieren von Variablen und Arrays oder einfachen Rechnungen. Darauf folgt das Programmieren von Funktionen, Einlesen und Plotten von Datensätzen und das Implementieren von einfacheren Algorithmen. Die verwendeten Beispiele sind hierbei oft nah an möglichen Anwendungsbereichen als Physiker*in.

Übungen

In den Übungen soll dann dieses Wissen durch Programmieraufgaben erprobt werden. Hier werden die numerischen Methoden, die in der Vorlesung behandelt wurden, in einen praktischen Kontext eingebettet. Die Übungen wurden in einer Jupyter-Notebook den Studierenden bereitgestellt. Die Studierenden müssen dann die Aufgaben mit der Programmiersprache Julia lösen. In den ersten Übungsaufgaben wird ein Großteil des Codes schon gegeben, den es dann zu verstehen und ergänzen gilt. Der Anteil der Algorithmen, der bereits durch die Aufgabenstellung gegeben ist, nimmt dann über das Semester hin ab, bis in den letzten Übungen eigentlich gar kein Code mehr gegeben ist. Die Übungen sollten in Gruppen mit bis zu drei Studierenden bearbeitet und abgegeben werden. Es wird auch eine Plattform zur Verfügung gestellt, mit der die Gruppenmitglieder gleichzeitig am gleichen Programm arbeiten können. Es gibt eine Klausurzulassung, für die die Studierenden 50% der Punkte aus allen Übungsaufgaben erreichen müssen. Falls einzelne Studierende kein Endgerät besitzen, gibt es hierfür auch verschiedene Lösungen, wie sie diese das Modul absolvieren können.

Klausuren

In der Modulabschlussprüfung bekommen Studierende üblicherweise eine Klausur, in der Vorlesungswissen abgefragt wird. „Welcher Algorithmus macht was?“ „Welcher Algorithmus lohnt sich hier mehr?“ oder: Vervollständigen Sie den folgenden Code.“ Wenn man sich nun sowohl das Prüfungsformat als auch den Inhalt der Vorlesung anschaut, könnte man sagen, dass man eine Diskrepanz zwischen dem, was geprüft werden soll, und der Prüfung erkennt: Die Übungen erzeugen im Laufe des Semesters eine Selbständigkeit in der Arbeit als Programmierer*in. Hier wird also vor allem das Wissen aus der Vorlesung und nur kaum bis gar nicht die Kompetenzen aus Übung und Programmier-Tutorial abgeprüft.

Pilotversuch Projektarbeit

In den Sommersemestern 2021 und 2022 wurde als Alternative zu einer „normalen“ Klausur eine Projektarbeit ausprobiert. Das Projekt beinhaltete eine Aufgabe im Kontext eines physikalischen Problems, bei der die Problemstellung vollständig gegeben war. Dieses Problem sollte man mit einer numerischen Methode, welche man in der Vorlesung gelernt hat, lösen. Desweiteren musste man über das Projekt ein Video machen. Darin erklärt man die Problemstellung und das Ergebnis in einer Powerpoint-Präsentation. Es waren zwei Wochen für das Projekt inklusive Video-Erstellung

vorgesehen, also waren Fleiß und verfügbare Zeit wichtige Faktoren. Der lange Prüfungszeitraum minimiert Stressfaktoren wie Zeitdruck. Gleichzeitig steigt der Anspruch an sich selbst.

Im Anschluss wurde dann ein Test geschrieben, welcher kurz das Vorlesungswissen abfragt. Der Test bestand aus offenen Fragen, welche ein Rundumschwung aus den theoretischen Methoden der Vorlesung waren. Für den Test war eine Stunde vorgesehen. Die Abschlussnote bestand dann zu 75% aus dem Projekt und zu 25% aus dem Test, wobei zum Bestehen beide Teile mit mindestens 4,0 abgeschlossen werden mussten. Eine nicht-bestandene „normale“ Klausur unbegrenzt häufig wiederholt werden; im Pilotprojekt konnten auch einzelne Teile der Prüfung, also Projekt oder Test, wiederholt werden.

Kompetenzabfragen in der Projektarbeit

In dieser Prüfungsform wird geprüft, wie gut die Studierenden eigenständig einfache Programmieraufgaben bewältigen können mit dem Zusatz, das Programmiererte auch erklären zu können. Die Prüfungen hatten damit einen starken Anwendungsfokus und bestehen, abgesehen von dem Test-Anteil, ausschließlich aus Problemlösung. Weiterhin fördert das abzugebende Video Kommunikation von Ergebnissen auf einem wissenschaftlichen Niveau. Somit wird nach der Fähigkeit selektiert, ob Studierende selbstständig ein Programm schreiben und präsentieren können, das einfache Probleme löst oder bearbeitet. Vortäuschen von Wissen ist bei diesem Prüfungsformat schwer möglich, da es um tatsächliche Problemlösekompetenz geht, zusätzlich mit dem Vorstellen des Gelernten. Programmieren erfordert immer kreative Problemlösung und kann nicht einfache Reproduktion sein. Die Problemstellungen waren teilweise sehr eng gefasst, was die Kreativität bei der Problemlösung etwas eingeschränkt hat. Somit wurde durch die Projektarbeit als Prüfungsform deutlich mehr Fokus auf die Kompetenz des Programmierens gelegt. Dies in einem Rahmen (2 Wochen Bearbeitungszeit), der deutlich näher an der alltäglichen Arbeitsweise in der Physik ist als eine künstlich erzeugte Stresssituation, wie es in einer Klausur der Fall ist.

Empfinden der Studierenden

Die Studierenden empfanden den Programmiereteil der Prüfung als fair, angemessen und auch als Bestätigung selbstständig programmieren zu können. Viele Studierende hatten dabei auch Spaß, ein Programm zu schreiben und sich ins Alltagsleben einer Programmierenden zu begeben. Es gab starke Unterschiede in der Schwierigkeit der (zufällig zugewiesenen) Projekte, was dazu führte, dass beim ersten Durchlauf die Studierenden sich im Nachhinein unfair behandelt gefühlt haben. Außerdem gab es auch für viele Studierende einen Angstfaktor sich selbst zu filmen. Viele Studierende fanden den Test-Anteil unangemessen anspruchsvoll, vor allem im Verhältnis zur Ankündigung vorher, wo anderes kommuniziert wurde. Kommuniziert wurde gegen Anfang, dass es ein Multiple-Choice-Test sein soll, doch überrascht wurden die Studierende von einem 12-Fragen-Test mit offenen Fragen. Im zweiten Durchlauf wurden alle Aspekte des Projektes stark verbessert, zum einen die Kommunikation und Organisation, zum anderen die Projekte und die Berücksichtigung der Klausurphase. Auch die Kommunikation in Bezug auf den Test wurde verbessert. Der gesamte Prüfungsaufwand durch die Projektarbeiten wurde in beiden Durchläufen als deutlich zu hoch wahrgenommen, da eine einzelne Klausur durch Projekt mit Präsentationsvideo und einen Test ersetzt wurde, was in Summe als deutlich mehr Aufwand wahrgenommen wurde. Dies war im zweiten Durchgang allerdings erheblich besser, da in den letzten zwei Semesterwochen keine Übungen mehr bearbeitet werden mussten und keine Vorlesungen stattfanden, um Zeit für die Projektarbeit zu schaffen.

Diese Eindrücke stammen aus verschiedenen Gesprächen von Fachschaftsmitgliedern mit Studierenden, die das Modul belegt haben, und auch aus persönlichen Meinungen von Fachschaftler*innen, die gemeinsam mit dem Dozenten Matteo Rizzi immer wieder ausgewertet und mit den standardmäßig stattfindenden Evaluationen, die aber leider nur bedingt Aussagen zu spezifischen Fragen zulassen, abgeglichen.

Fazit

Stellt man nun Klausur und Projektarbeit gegenüber, besteht der Hauptunterschied darin, welche Kompetenzen in den zwei verschiedenen Prüfungsformaten abgefragt werden. Wo die Klausur nur das Wissen der Vorlesung abprüft, prüft das Projekt die Fähigkeit selbstständig zu Programmieren. Da dies einer der essentiellen Bestandteile des Moduls ist, sollte es auch seinen Platz in der Modulprüfung finden. Weitergehend ist anzumerken, dass die Kompetenz Inhalte oder Ergebnisse wissenschaftlich zu präsentieren positiv ist, vor allem da diese Kompetenz in Köln nur selten bis gar nicht vor dem Bachelorkolloquium geübt wird.

Außerdem bietet die Projektarbeit eine gute Abwechslung zu dem sonst sehr klausurenlastigen Bachelorstudium. Der Vorteil der Projektarbeit, die Prüfungsangst zu reduzieren, wurde leider dadurch relativiert, dass immer noch ein Test stattgefunden hat, wenn auch mit deutlich geringerer Bedeutung und der Gesamtarbeitsaufwand im Vergleich zu einer herkömmlichen Klausur auch im zweiten Durchlauf von den Studierenden als zu hoch empfunden wurde. Leider wird eine Weiterentwicklung derzeit dadurch erschwert, dass das Kölner Justizariat einfordert, dass jede Prüfungsordnung die Prüfungsform eindeutig festlegen muss.