



Blended Learning mit Jupyter Notebooks

J. Brose*

Fakultät Physik, Technische Universität Dresden

Abstract

Seit 2021 findet jeweils in der vorlesungsfreien Zeit im Frühjahr der fakultative Blockkurs „Python in der Physik“ als Blended Learning Programmierkurs mit Wissensvermittlung im Inverted Classroom basierend auf Jupyter Notebooks mit interaktiven Elementen sowie im gleichen Umfang Arbeit an konkreten physikalischen Aufgaben in Kleingruppen im PC-Pool in Präsenz und unter Anleitung durch Tutoren statt. Im Beitrag werden die organisatorisch-technische Umsetzung sowie die gewonnenen Erkenntnisse aus dieser Veranstaltungsform präsentiert.

Since 2021, the optional block course "Python in Physics" has been held during the spring semester break as a blended learning programming course with knowledge transfer via inverted Classroom based on Jupyter notebooks with interactive elements and to the same extent work on concrete physical tasks in small groups in the PC pool in presence and under guidance by tutors. The article presents the organizational and technical implementation as well as the insights gained from this type of event.

*Corresponding author: jens.brose@tu-dresden.de

1. Konzeption des Kurses

Bis einschließlich Wintersemester 2019 umfasste der Bachelor-Studiengang Physik an der TU Dresden eine einsemestrige Lehrveranstaltung Programmierung mit je zwei Semesterwochenstunden Vorlesung und Übung im PC-Pool. Im Rahmen der Überarbeitung der Studienordnung sollte die Lehrveranstaltung in einen zweiwöchigen fakultativen Blockkurs in der vorlesungsfreien Zeit zwischen Winter- und Sommersemester überführt werden. Damit ergab sich der Anlass einer grundsätzlichen Überarbeitung der Methodik der Lehrveranstaltung:

- Wissensvermittlung in Selbstlerneinheiten im Inverted Classroom basierend auf Jupyter Notebooks mit interaktiven Elementen an Stelle von Präsenzvorlesungen
- Arbeit an konkreten physikalischen Aufgaben in Kleingruppen im PC-Pool unter Anleitung durch Tutoren

In den Selbstlerneinheiten erfolgt die Wissensaneignung durch multimediales Online-Material in individueller Arbeit [1] statt im traditionellen Vorlesungsformat. Die Präsenzphase in den Programmierübungen im PC-Pool wird zusätzlich zur Klärung von Fragen, die in der Vorbereitung aufgekommen sind, genutzt. Damit entspricht diese Kombination von Lehr-Lern-Szenarien dem Modell des Inverted Classroom [2],[3]. Zusammen mit der konkreten Programmierfähigkeit in Präsenz im PC-Pool entsteht eine Form des integrierten Lernens (Blended Learning, [4],[5]) als Kombination aus Online- und Präsenzphasen. Der Umfang der Selbstlerneinheiten und der Programmieraufgaben wurde so gewählt, dass beide Formen ungefähr den gleichen Zeitaufwand benötigen.

Als Programmiersprache kommt Python [6] zum Einsatz, da hier einerseits die Einstiegshürden für Programmieranfänger besonders niedrig sind, andererseits durch eine Vielzahl von wissenschaftlichen Modulen alle für die Physik relevanten Einsatzzwecke abgedeckt werden. Auf Python beruhende Jupyter Note-

books [7] können als Webanwendung überall verwendet werden und bieten neben der interaktiven Ausführung von Programmcode die Möglichkeit, Gleichungen, Visualisierungen und Verweise als auch formatierten Text darzustellen. Sie bilden daher ein ideales Werkzeug, um Programme, ihre Ergebnisse sowie ihre Beschreibung und Dokumentation zu vereinen sowie Datenanalysen in Echtzeit durchzuführen.

Die Studierenden finden immer die gleiche Arbeitsoberfläche vor, sei es im PC-Pool oder auf dem privaten Endgerät. Dazu ist keinerlei Softwareinstallation notwendig. Die Auslieferung der Notebooks erfolgt im Webbrowser über einen Jupyterhub [8], der als virtueller Server in der TU Dresden Enterprise Cloud eingerichtet wurde.

Im entsprechenden Kurs des an der TU Dresden verwendeten Lernmanagementsystems OPAL finden die Teilnehmenden Links, die die individuelle Anmeldung am Jupyterhub mit den OPAL-Zugangsdaten automatisieren. Die Links werden über das OPAL LTI-Tool (Learning Tool Interoperability) [9] realisiert. Bei der Verwendung des entsprechenden Links wird an jeden Kursteilnehmer ein anfangs identisches, aber individuell bearbeit- und speicherbares Jupyter Notebook vom Gitlab Versionsverwaltungssystem [10] des Bereichs Mathematik und Naturwissenschaften der TU Dresden ausgeliefert.

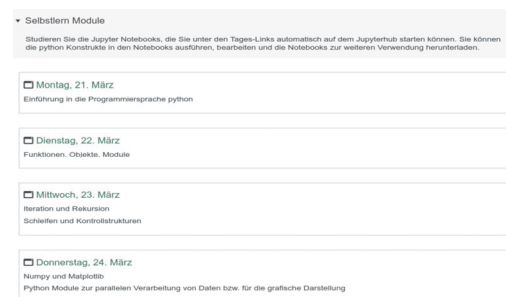


Abb. 1: Auswahl von Jupyter Notebooks im OPAL Kurs.

Im Notebook kann Programmcode in Eingabezellen geschrieben und direkt ausgeführt werden. Für die Studierenden besteht die Möglichkeit, die Richtigkeit der Eingaben selbständig zu überprüfen.

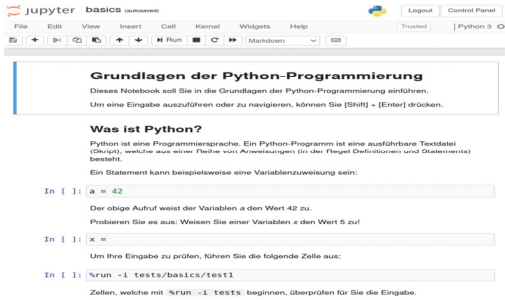


Abb. 2: Dokumentations- und Codezellen im Jupyter Notebook.

Neben den täglichen Selbstlerneinheiten mit ca. 1,5 Stunden Bearbeitungszeit werden ebenfalls täglich durch Tutoren betreute neunzig-minütige Übungen in Kleingruppen (maximal zwölf Teilnehmende) in PC-Pools durchgeführt. Dort finden sich zwei bis drei Studierende zusammen, um ein Programmierproblem gemeinsam zu lösen.



Abb. 3: Konzeption des Programmierkurses.

Wegen der großen Spannweite der Vorkenntnisse der Teilnehmenden wurde der Kurs in zwei voneinander unabhängige, jeweils aus Selbstlerneinheiten und angepassten Übungen bestehende Stränge unterteilt. Studierende ohne Vorkenntnisse in der Programmierung eigneten sich in der ersten Woche Grundlagen der Programmiersprache Python an und lernten in der zweiten Woche die Module für Grafik-Ausgabe und numerische Berechnungen auf Datenarrays kennen. Die erworbenen Fähigkeiten wendeten Sie in den Übungen auf typische Problemstellungen in der Auswertung physikalischer Praktika, z.B. Lineare Anpassung, Zufallszahlen und Statistik sowie Monte Carlo Methoden an. Der Fokus der Gruppe der Fortgeschrittenen lag auf wissenschaftlicher Datenverarbeitung, Verstehen (durch eigene Implementierung) elementarer numerischer Methoden für Differentiation, Integration, Nullstellensuche und

Lösung gewöhnlicher Differentialgleichungen sowie Kennenlernen der Standardmodule zur Bearbeitung dieser Problemstellungen.

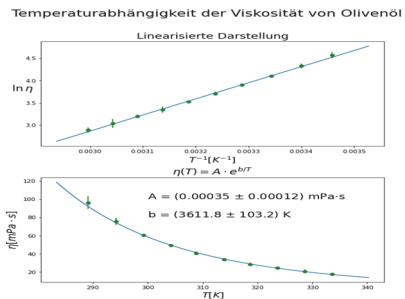


Abb. 4: Musterlösung der Aufgabe zur Linearen Anpassung für Programmieranfänger



Abb. 5: Aufgabe zur numerischen Nullstellensuche für Teilnehmende mit Vorkenntnissen (Teil 1).

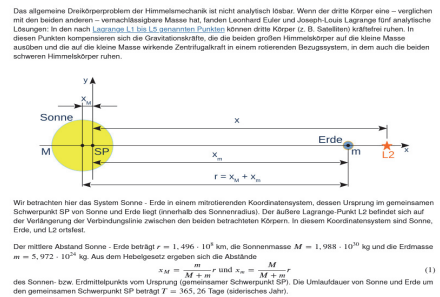


Abb. 6: Aufgabe zur numerischen Nullstellensuche für Teilnehmende mit Vorkenntnissen (Teil 2).

2. Technische Umsetzung

Die Selbstlerneinheiten und Übungen bestehen aus einzelnen Notebookfragmenten, die mit Hilfe einer einfachen grafischen Oberfläche für die Jupyter Erweiterung nbmerge [11] zum finalen Notebook zusammengesetzt werden. Diese Vorgehensweise erlaubt einen schnellen und flexiblen Austausch von Inhaltsblöcken der Notebooks. Die Verbindung zwischen dem Git Repository mit den vorbereiteten Notebooks und den Notebook Instanzen der Studierenden auf

dem Jupyterhub wird mit der Jupyter Erweiterung nbgitpuller [12] im OPAL LTI-Tool hergestellt.

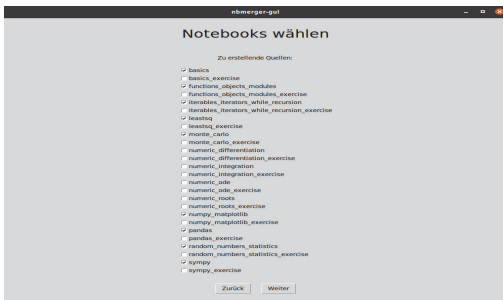


Abb. 7: Auswahl der Notebooks für die Zusammenstellung mit Hilfe von nbmerger-gui.



Abb. 8: Zusammenstellen der Notebookfragmente für eine Selbstlerneinheit mit Hilfe von nbmerger-gui.

Da die bisher in OPAL verwendete Version des LTI-Tools keinen Rückkanal vorsieht, wurde ein Dateiupload für bearbeitete Notebooks eingerichtet (allerdings im Kurs nicht verwendet, da keine Bewertung der Übungen vorgenommen wird).

Die gesamte Umsetzung des Projekts basierend auf den Inhalten des Programmierkurses bis 2019 wurde in der Periode 2019/20 durch den Multimediafonds der TU Dresden gefördert.

3. Gewonnene Erkenntnisse

Im März 2022 haben 68 Studierende aus der ca. hundertköpfigen Zielgruppe - Studiengang Bachelor Physik, 1. Semester - an mindestens neun der zehn Präsenzübungen teilgenommen. Der fakultative Charakter des Kurses und das Fehlen eines Leistungsdrucks führten, verglichen mit dem Pflichtkurs bis 2019, bei dem in den Übungen Leistungspunkte erworben werden mussten, um das Modul zu bestehen, zu einem deutlich lebendigeren, kooperativen und produktiven Charakter der Übungen. Für die Physik Studierenden praxisnahe Übungs-

aufgaben stellten dabei den Bezug zum Hauptfach her. Die Zusammenarbeit in kleinen Teams von zwei bis drei Personen an einer Problemstellung förderte den Erkenntnisgewinn.

An den letzten beiden Kurstagen wurde eine anonyme Online-Umfrage zur Evaluation der Veranstaltung für die Teilnehmenden freigeschaltet. Diese enthielt 24 single bzw. multiple Choice Fragen zur Lehrveranstaltung, die sich an den Fragen der üblichen Lehrevaluation für Semester-Kursveranstaltungen orientierten (ergänzt um einige technische Fragen zur konkreten Umsetzung). An der Umfrage beteiligten sich 48 der 68 Teilnehmenden.

Jupyter Notebooks wurden auf Grund ihres interaktiven Charakters als Gewinn bringendes Arbeitsmittel eingeschätzt. Die als Inverted Classroom gestaltete Kurskomponente der Selbstlerneinheiten, verbunden mit individuellen Tests mit Response wurde sehr positiv angenommen.

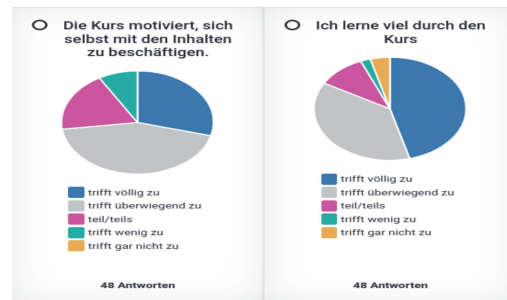


Abb. 9: Evaluation des Programmierkurses.

Durch den verstärkten Einsatz von Python in der Versuchsauswertung im Physikalischen Praktikum sowie der Verwendung dieser Programmiersprache in weiterführenden Lehrveranstaltungen (z. B. Computational Physics, Numerik und Computersimulationen in der weichen kondensierten Materie) ergibt sich die Praxisrelevanz des Programmierkurses.

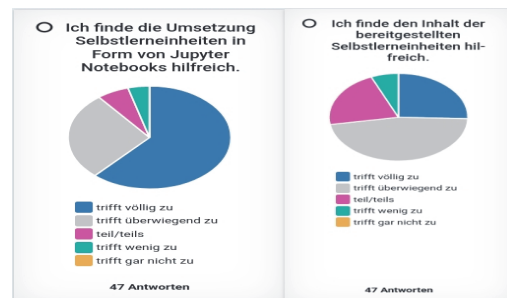


Abb. 10: Evaluation: Jupyter Notebooks als Arbeitsmittel.

In Abb. 11 werden die für beide Lehrveranstaltungsformen (Blau - Wintersemester 2019/20 (Pflichtveranstaltung mit klassischer Präsenz-Vorlesung und bewerteten Übungen); Rot - Blockkurs 2022 (fakultativ in der vorlesungsfreien Zeit mit Selbstlerneinheiten (inverted classroom) und betreuten Präsenz-Übungen)) identischen Evaluationsfragen verglichen.

Während die Anforderungen bzw. die Schwere des Stoffes sowie die Stoffmenge jeweils als optimal eingeschätzt wurden, ergeben sich bei einigen der auf der Notenskala von 1 bis 5 zu bewertenden Fragen Unterschiede. Obwohl die Inhalte beider Lehrveranstaltungsformen nahezu identisch waren, wurden beispielsweise beim Blended Learning Kurs 2022 die Lerninhalte als besser auf das Vorwissen der Studierenden abgestimmt eingeschätzt. Signifikante positive Unterschiede für das Blended Learning Format ergeben sich bei den Aussagen „Ich lerne viel durch den Kurs.“ (1,81 +- 0,14) vs. (2,77 +- 0,13) sowie „Insgesamt bin ich mit dem Kurs zufrieden.“ (1,77 +- 0,11) vs. (2,25 +- 0,14).

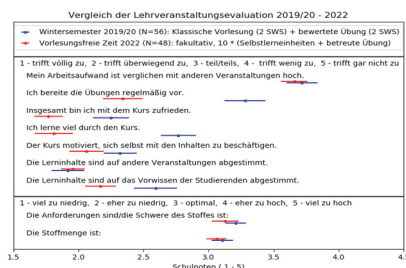


Abb. 11: Vergleich der Lehrveranstaltungsevaluation: Blau - Wintersemester 2019/20 (Pflichtveranstaltung mit klassischer Präsenz-Vorlesung und bewerteten Übungen, 56 Umfrageteilnehmer); Rot - Blockkurs 2022 (fakultativ in der vorlesungsfreien Zeit mit Selbstlerneinheiten (inverted classroom) und betreuten Präsenz-Übungen, 48 Umfrageteilnehmer). Dargestellt sind Mittelwerte und Standardabweichungen der nach Schulnoten (1-5) bewerteten Fragen.

Besonders überraschend ist das Ergebnis bei der Aussage "Ich bereite die Übungen regelmäßig vor." (2,34 +- 0,15) vs. (3,28 +- 0,16), da ja beim schlechteren Ergebnis im Wintersemester 2019/20 die Übungsaufgaben abgegeben werden mussten und bewertet wurden. Bei der Beurteilung der Evaluationsergebnisse sollte man allerdings berücksichtigen, dass die

Kohorte Wintersemester 2019/20 alle Studierenden einer Pflichtveranstaltung umfasste, während im Vergleich beim Blended Learning Kurs 2022 die Teilnehmenden freiwillig in der Semesterpause kamen und man ihnen daher eine im Mittel höhere Motivation unterstellen kann.

Die Lehrveranstaltung wurde auf Vorschlag der Studierenden der Fakultät Physik mit dem Lehrpreis der TU Dresden 2021 ausgezeichnet.

Danksagung

Ich möchte mich beim Multimediafonds 2019/20 für die Förderung des Projekts und bei Konstantin Köhring, Fakultät Informatik, der als studentische Hilfskraft die Konzeption und Umsetzung der Kursgestaltung übernahm, bedanken.

Literatur

- [1] <https://dbs-lin.ruhr-uni-bochum.de/lehreladen/e-learning-technik-in-der-lehre/inverted-classroom/was-ist-inverted-classroom/>
- [2] Lage, M. J., Platt, G. J., & Treglia, M. (2000). Inverting the classroom: A gateway to creating an inclusive learning environment. *The Journal of Economic Education*, 3 (1), 30 – 43, <https://doi.org/10.2307/1183338>
- [3] J. Handke, A. Sperl (Hrsg.): *Das Inverted Classroom Model*. Oldenbourg, München 2012, ISBN 978-3-486-71652-8.
- [4] A. Sauter, W. Sauter: *Blended Learning. Effiziente Integration von E-Learning und Präsenztraining*. Luchterhand, Neuwied 2004, ISBN 3-472-05592-8.
- [5] <https://www.ecademy-learning.com/ausbildung-digital/blended-learning/>
- [6] Python: <https://www.python.org/>
- [7] Jupyter Notebook: <https://jupyter.org/>
- [8] Jupyterhub: <https://jupyter.org/hub>
- [9] OPAL LTI: <https://www.bps-system.de/help/display/LMS/LTI-Tool>
- [10] Gitlab: <https://gitlab.mn.tu-dresden.de/>
- [11] nbmerge: <https://github.com/jbn/nbmerge>
- [12] nbgitpuller: <https://github.com/jupyterhub/nbgitpuller>