Lessons Learned 2, 2 (2022)
Submitted: 21.10.2022
Accepted: 05.12.2022
DOI: https://doi.org/10.25369/ll.v2i2.52
ISSN: 2749-1293 (print); 2749-1307 (online).

Blended Learning with Jupyter Notebooks

J. Brose*

Faculty of Physics, TU 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

This article was originally submitted in German.

1. Course conception

Up to and including the winter semester 2019, the bachelor's program in physics at the TU Dresden included a one-semester course in programming with two semester hours per week each of lecture and exercise in the PC pool. As part of the revision of the study regulations, the course was to be converted into a two-week optional block course during the lecture-free period between the winter and summer semesters. This gave rise to a fundamental revision of the methodology of the course:

- Knowledge transfer in self-learning units in the inverted classroom based on Jupyter notebooks with interactive elements instead of face-to-face lectures
- Work on concrete physical tasks in small groups in the PC pool under the guidance of tutors

In the self-study units, knowledge is acquired through multimedia online material in individual work [1] instead of in the traditional lecture format. The presence phase in the programming exercises in the PC pool is additionally used to clarify questions that have arisen during preparation. Thus, this combination of teaching-learning scenarios corresponds to the inverted classroom model [2],[3]. Together with the actual programming activity in presence in the PC pool, a form of integrated learning (blended learning, [4],[5]) is created as a combination of online and presence phases. The scope of the self-learning units and the programming tasks was chosen so that both forms require approximately the same amount of time.

Python [6] is used as the programming language, since on the one hand the entry barriers for programming beginners are particularly low, and on the other hand all purposes relevant to physics are covered by a large number of scientific modules. Jupyter Notebooks [7], which are based on Python, can be used anywhere as a web application and, in addition to the interactive execution of program code, offer the possibility of displaying equations, visualizations and references as well as formatted text. They therefore form an ideal tool for unifying programs, their results, and their description and documentation, as well as for performing data analysis in real time.

Students always find the same work interface, whether in the PC pool or on their private end device. This does not require any software installation. The notebooks are delivered in the web browser via a Jupyterhub [8], which was set up as a virtual server in the TU Dresden Enterprise Cloud.

In the corresponding course of the learning management system OPAL used at the TU Dresden, the participants will find links that automate the individual login to the Jupyterhub with the OPAL access data. The links are realized via the OPAL LTI tool (Learning Tool Interoperability) [9]. When using the corresponding link, an initially identical but individually editable and storable Jupyter notebook is delivered to each course participant by the Gitlab version control system [10] of the Department of Mathematics and Natural Sciences of the TU Dresden.

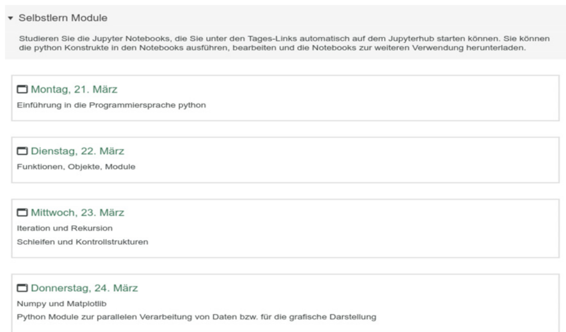

Fig. 1: Selection of Jupyter notebooks in the OPAL course.

In the notebook, program code can be written in input cells and executed directly. Students have the opportunity to independently check the correctness of the input.

In addition to the daily self-study units with approx. 1.5 hours of processing time, ninety-minute exercises supervised by tutors are also carried out daily in small groups (maximum twelve participants) in PC pools. There, two to three students come together to solve a programming problem together.

Fig. 2: Documentation and code cells in the Jupyter Notebook.

Fig. 3: Concept of the programming course.

Due to the wide range of prior knowledge of the participants, the course was divided into two independent strands, each consisting of self-study units and adapted exercises. Students with no prior knowledge of programming acquired the basics of the Python programming language in the first week and learned the modules for graphics output and numerical calculations on data arrays in the second week. They applied the acquired skills in the exercises to typical problems in the analysis of physics lab courses, e.g. linear fitting, random numbers and statistics, and Monte Carlo methods.

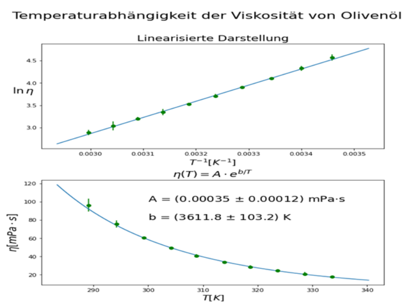

Fig. 4: Sample solution of the linear fitting task for programming beginners

The focus of the advanced group was on scientific data processing, understanding (through own implementation) elementary numerical methods for differentiation, integration, determination of zeroes of functions and solution of ordinary differential equations as well as getting acquainted with the standard modules for working on these problems.

Fig. 5: Numerical zero search task for participants with previous knowledge (part 1).

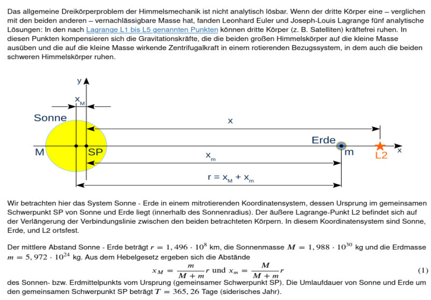

Fig. 6: Numerical zero search task for participants with previous knowledge (part 2).

2. Technical implementation

The self-learning units and exercises consist of individual notebook fragments that are assembled into the final notebook using a simple graphical interface for the Jupyter extension nbmerge [11].

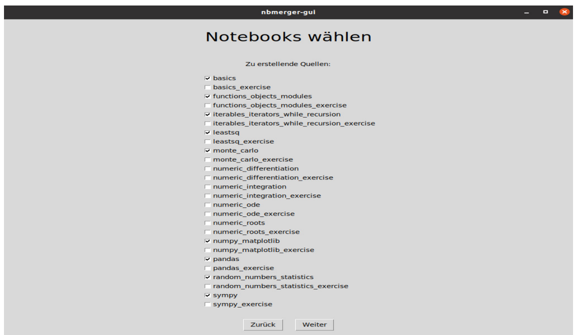

Fig. 7: Selection of notebooks for compilation using nbmerger-gui.

This approach allows a fast and flexible exchange of content blocks of the notebooks.

Fig. 8: Assembling notebook fragments for a self-learning unit using nbmerger-gui.

The connection between the Git repository with the prepared notebooks and the students' notebook instances on the Jupyterhub is established using the Jupyter extension nbgitpuller [12] in the OPAL LTI tool.
Since the version of the LTI tool used in OPAL so far does not provide for a return channel, a file upload for edited notebooks has been set up (though not used in the course, since no assessment of the exercises is done).
The entire implementation of the project based on the content of the programming course until 2019 was funded by the Multimedia Fund of TU Dresden in the period 2019/20.

3. Lessons learned

As of March 2022, 68 students from the target group of approximately one hundred -Bachelor of Physics, 1st semester- have participated in at least nine of the ten face-to-face exercises. The optional nature of the course and the absence of performance pressure, compared to the compulsory course until 2019 where credit points had to be earned in the exercises to pass the module, resulted in a much more lively, cooperative and productive character of the exercises. Practice-oriented exercise assignments established a connection to the major subject. The cooperation in small teams of two to three persons on a problem promoted the gain of knowledge.

On the last two days of the course, an anonymous online survey for the evaluation of the course was released to the participants. This contained 24 single or multiple choice questions about the course, which were based on the questions of the usual teaching evaluation for semester course events (supplemented by some technical questions about the concrete implementation). Forty-eight of the 68 participants took the survey.

Jupyter notebooks were considered to be a valuable tool due to their interactive character. The course component of the self-study units, designed as an inverted classroom, combined with individual tests with response was very positively received.

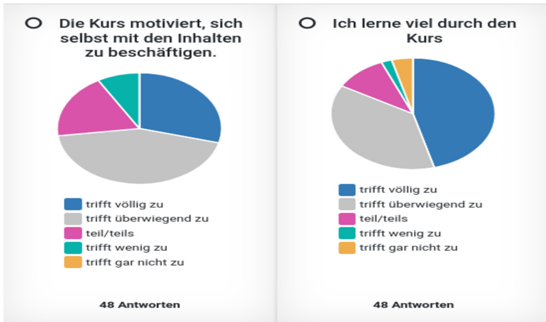

Fig. 9: Evaluation of the programming course.

The increased use of Python in experiment evaluation in the physics lab course, as well as the use of this programming language in more advanced courses (e.g., computational physics, numerics, and computer simulations in soft condensed matter), results in the practical relevance of the programming course.

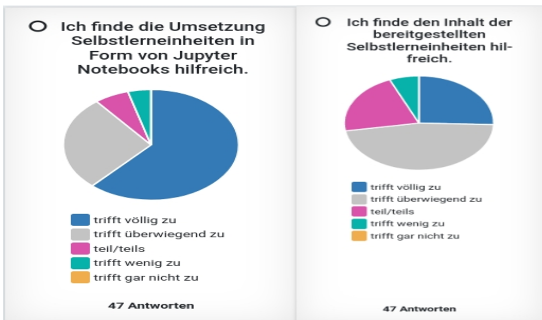

Fig. 10: Evaluation: Jupyter notebooks as work tools.

Fig. 11 compares the identical evaluation questions for both course types (blue - winter semester 2019/20 (compulsory course with classical face-to-face lecture and assessed exercises); red - block course 2022 (optional during the lecture-free period with self-learning units (inverted classroom) and supervised face-to-face exercises)).
While the requirements or the severity of the material as well as the amount of material were assessed as optimal in each case, there

are differences in some of the questions to be evaluated on the grading scale of 1 to 5. For example, although the content of both types of courses was nearly identical, the blended learning course 2022 was rated as better aligned with students' prior knowledge. Significant positive differences for the blended learning format occurred for the statements "I learn a lot through the course." (1.81 +- 0.14) vs. (2.77 +- 0.13) and "Overall, I am satisfied with the course." (1.77 +- 0.11) vs. (2.25 +- 0.14).

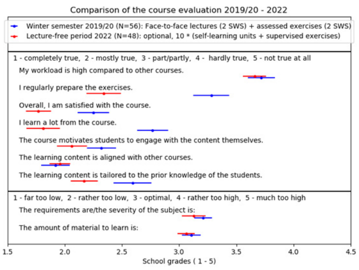

Fig. 11: Comparison of course evaluations: Blue - winter semester 2019/20 (compulsory course with classic face-to-face lecture and graded exercises, 56 survey participants); Red - block course 2022 (optional during the lecture-free period with self-learning units (inverted classroom) and supervised face-to-face exercises, 48 survey participants). Mean values and standard deviations of the questions evaluated according to school grades (1-5) are shown.

The result is particularly surprising for the statement "I regularly prepare the exercises." (2.34 +- 0.15) vs. (3.28 +- 0.16), since in the case of the worse result in the winter semester 2019/20, the exercises had to be handed in and assessed. When assessing the evaluation results, however, it should be taken into account that the cohort winter semester 2019/20 included all students of a compulsory course, while in the comparison with the blended learning course 2022, the participants came voluntarily during the semester break and can therefore be assumed to have a higher motivation on average.
The course was awarded the TU Dresden 2021 teaching prize at the suggestion of the students of the Physics Faculty.

Acknowledgement

I would like to thank the Multimediafonds 2019/20 for funding the project and Konstantin Köhring, Faculty of Computer Science, who was the student assistant responsible for the conception and implementation of the course design.

Literature

[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 (eds.): Das Inverted Classroom Model. Oldenbourg, Munich 2012, ISBN 978-3-486-71652-8.
[4] A. Sauter, W. Sauter: Blended Learning. Efficient Integration of E-Learning and Classroom Training. 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