

Research-Oriented Teaching with bwJupyter

Jasmin Hörter ^{*}, Paul Hoger, Stephanie Hofmann, and Admir Obraliija

October 10, 2026

Abstract

As the influence of artificial intelligence and computational modeling in all scientific fields grows, profound skills in computational and data science are more important than ever. University education must reflect this development and incorporate state-of-the-art research tools and infrastructures into teaching and course work. *bwJupyter* provides a platform for all universities in Baden-Württemberg to access powerful computing resources and to engage with these new technologies in a low-threshold way. Special features that are adapted to user needs upgrade classical Jupyter Notebooks to a versatile tool in modern, research-oriented higher education.

1 Introduction

Today, computer-aided modeling and simulations are not only used in the natural sciences and technology, but in almost all areas of research, from medicine to the social sciences and humanities. With the rapidly growing field of artificial intelligence, in particular, the possible applications are endless. This places new demands on the skills of graduates in all disciplines. To use the numerous tools effectively, modern university education must not only include subject-specific topics but also promote knowledge of computational and data scientific principles as well as their expert application to address a wide range of issues. For this to succeed, the entry level to computing environments for both students and teachers should be low and the handling should be simple.

The open-source project Jupyter satisfies all these requirements. Jupyter Notebooks enable interactive and visual representations of research processes, allowing students to engage with scientific methods and data analysis in a practical way. By combining code, text, graphics and interactive elements, complex concepts are understandably communicated and directly applied to real research scenarios. The flexible integration of programming languages such as Python, Julia, or R further promotes the development of coding skills.

To provide access to this state-of-the-art research tool, the ‘bwJupyter for Teaching’ project, a collaboration between the Karlsruhe Institute of Technology (KIT) and the University of Stuttgart, established a state-wide JupyterHub for university teaching in Baden-Württemberg (BW). In the following, we will first introduce the functionality of Jupyter Notebooks, and then focus on the objectives and components of bwJupyter. During the 21 month development phase, various features were implemented to support the low-threshold use of Jupyter within the context of education and teaching. Lecturers can use those to create tailor-made profiles for their courses. Finally, we introduce the technical setup, present selected case studies and look to future applications.

^{*}Karlsruhe Institute of Technology

2 The Jupyter Notebook Environment

Jupyter Notebooks [5] are interactive documents that consist of multiple building blocks including code, text or visualizations (see Fig.1). Via the JupyterHub, a multi-user server, Jupyter Notebooks can be launched directly on the computing nodes of any underlying compute resource, including large infrastructures like High-Performance Computing (HPC) or AI clusters.

Every Notebook consists of code and text cells. Coding cells contain source code in a predefined programming language. Jupyter supports multiple programming languages, including **Julia**, **Python**, and **R**. Each code cell can be executed individually, enabling step-by-step learning. The output of a code cell is displayed within the Notebook in text form or as a visualization (e.g., image, plot, graph). Outputs can also command further input or interaction (e.g., through sliders or check boxes). Text cells contain Markdown, $\text{\LaTeX}$, or HTML to explain content, pose exercises and document code. This makes Jupyter Notebooks an accessible, user-friendly interface compared to many plain coding environments.

Given their versatility, Jupyter Notebooks can be used in a wide range of teaching scenarios, from guided learning to exercises and independent project work [3, 8, 1]. For the beginner level, lecturers can create interactive lecture notes that guide students through the course content. Students with more advanced programming skills may use Jupyter Notebooks as a tool to complete exercises or research projects (see Sec. 7).

Given their versatility, Jupyter Notebooks can be used in a wide range of teaching scenarios, from guided learning to exercises and independent project work [3, 8, 1]. For the beginner level, lecturers can create interactive lecture notes that guide students through the course content. Students with more advanced programming skills may use Jupyter Notebooks as a tool to complete exercises or research projects (see Sec. 7).

3 Objectives

bwJupyter ...

- enables research-oriented teaching and promotes digital skills,
- improves digital sovereignty,
- provides centralized, accessible and energy-efficient computing resources.

Jupyter Notebooks are the de-facto standard for utilizing large-scale data centers and HPC clusters and thus ideally suited to research-oriented teaching and learning. With the growing influence of artificial intelligence in almost all areas of life, it becomes more important to enable the proper use of these new technologies and strengthen digital competencies.

Research codes from a wide variety of disciplines can be made available via the centralized bwJupyterHub environment and accessed through any web browser independent of device and

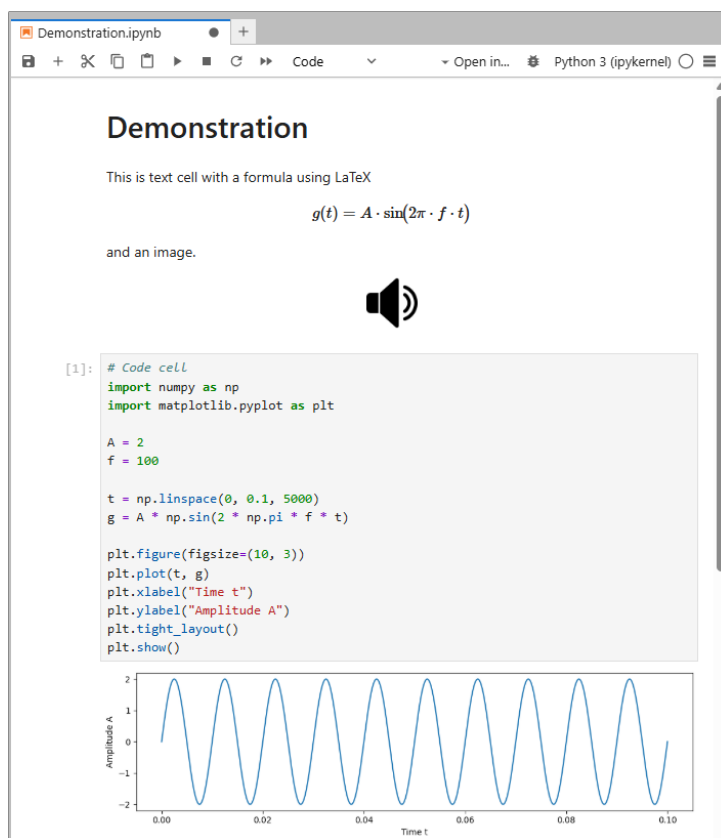


Figure 1: Example of a Jupyter Notebook

location. No local installation or downloads are needed. Hence, bwJupyter provides a low-threshold way for students to access large compute clusters from their own laptops or tablets. As admission is managed through the federated identity management bwIDM [6], anyone affiliated with a BW university can access bwJupyter without additional registration.

Moreover, bwJupyter is a further step towards digital sovereignty for universities, and hosting a JupyterHub on state-owned hardware offers significant advantages over relying on commercial providers. Furthermore, Jupyter supports over 40 programming languages and software packages, including many open source solutions like GNU Octave¹ which is comparable to the closed-source software MATLAB. Thus, the new state service also has the potential to replace costly commercial solutions.

Many lecturers already use Jupyter Notebooks, typically via a department-owned server. This situation, involving many small, low-performance, and independently operating systems, is extremely inefficient in terms of both energy consumption and management. For example, each server operator must individually resolve integration issues and more resource-intensive computing jobs (e.g., heavy GPU usage for AI models) cannot be executed. At the same time, smaller universities in particular often lack sufficient resources to operate their own JupyterHub. Hence, the goal of bwJupyter is to replace the extremely fragmented landscape with a centralized state-level service, thereby enabling the use of Jupyter at all universities. In addition, bwJupyter leverages many existing state services and infrastructures like bwIDM, bwSupport-Portal and ZOERR. This is not only resource- and cost-effective, but also significantly speeds up implementation, as existing expertise can be put to efficient use. A comparable initiative on a state-level is JupyterHub.nrw² in North Rhine-Westphalia, which we were in contact with throughout our development phase.

4 bwJupyter Features

The design of bwJupyter aims to support a variety of teaching scenarios. To achieve this, numerous features were developed and functionalities are constantly evolving.³

bwJupyter Notebooks are launched within a bwJupyter profile that determines which environment, i.e., programming language, libraries, and software is being used. Users can select from a set of predefined profiles that are either provided and maintained by the official Jupyter project⁴, or created based on the bwJupyter user survey conducted during the project (see Fig. 4).

Similar to learning environments like ILIAS and Moodle, lecturers have the option to create their own customized profiles for their courses by choosing any of the provided profiles and activating any of the available features. Alternatively, they can provide their own Docker image to base their profile on. Docker images enable the provision of additional programming languages and libraries, as well as specialized software packages, data sets and Jupyter add-ons. All profiles

Create a Profile

Profile Name

 The profile name will be shown in the profile list.

Profile Description

 The description states when the students should use the profile.

End Date

 We reserve the right to delete the profile and its associated data for all users after this date. This also affects the associated service and directory-based submissions, and groups, including the group accounts and files.

Access Control
☐ Restrict access to members of my university
 If this option is selected, only users from your university can start the profile. Otherwise, an error page is shown. Please note that you cannot change this option after the profile has been created.

Shared Directories
☒ Share files with students
 By default, every profile contains the directory `__shared` with 10 GB capacity, which is used to distribute course material. You can grant individual users write access after creating the profile by adding them as a profile collaborator.

☐ Share files among collaborators
 If this option is enabled, the profile owner and collaborators have write access to the shared directory `__collaborators`. You can add collaborators after the profile has been created by using the "Manage" button on the respective profile page.

Data
☐ Provide a Git repository
 By enabling this option, nbgitpuller is getting installed and automatically clones the Git repository into the provided sub-directory. The configuration cannot be changed once the profile is created.

Submissions
☐ Include submission and correction directories
 If this option is selected, two additional directories `__submission` and `__correction` are created. Students can submit their work to the `__submission` directory. If they start the profile, they find a sub-directory for their work in the `__submission` directory. Once a sub-directory is created, they can move their work into the `__correction` directory.

Figure 2: Creating a bwJupyter profile

¹<https://octave.org>

²<https://www.jupyterhub.nrw/de/>

³<https://hub.bwjupyter.de/hub/changelog>

⁴<https://jupyter-docker-stacks.readthedocs.io/en/latest/using/selecting.html>

are shared via a link and added to the personal profile list. To test the chosen features, lecturers can switch to student mode to interact with bwJupyter in a temporary student account. Figure 2 shows the workflow to create a new profile. Enabling a certain feature is as simple as ticking the corresponding checkbox.

Each profile contains a folder called `__shared`, where teaching material and data sets can be stored and which students can access. Since storage space is limited, students should not copy larger files to keep personal folders small and avoid redundant data storage.

Another way to share data is to provide a Git repository. Using the extension `nbgitpuller`, the local directory is automatically kept in sync with the remote repository. This enables lecturers to seamlessly create bwJupyter profiles from existing Git repositories and any changes in the repository are automatically transferred to the profile.

In order to keep storage available for active users, a deprovisioning strategy is applied to custom profiles that are used in courses and expire after 1.5 years.

- **Collaborative Work**

- **in real time**

- Coding projects are often done in groups, thus working together in the same Notebook is useful. The Real-Time Collaboration (RTC) feature, based on the package `jupyter-collaboration`, allows students to form their own groups and work simultaneously in real-time on the same Notebook.

- **among lecturers**

- If several lecturers are responsible for the same course, they can use the directory `__collaborators` to share files only between them and jointly manage course material in the profile.

- **Submissions**

To implement a simple submission system, lecturers can add additional `__submission` and `__correction` folders. Students can submit their files by copying them into the `__submission` folder. The profile owner and the profile collaborators can view all student submissions. Once the Notebook is corrected, it can be stored in the `__correction` folder and accessed by the individual student. If the collaboration feature is enabled, group submissions are also possible.

- **Grader Service**

To execute the entire exercise workflow (i.e., create, distribute, edit, submit, correct, give feedback) within bwJupyter, we integrated the grader service⁵ developed by the Technical University of Vienna. Lecturers can annotate Notebooks with tasks and points via a Jupyter extension, and public or hidden tests can be added to each task.

Once the assignment has been published, students can download the Notebook and work on their solutions. As soon as the Notebook is handed in, the tests are automatically carried out and points are allocated. After this automatic correction, it is possible to correct the submission manually. Finally, a Notebook containing the achieved points, the lecturer's comments and the tests carried out, is generated and made available to the student.

- **AI Assistant**

To support code development and guide novices, AI-supported tools including in-line completion, code correction and the Jupyter AI chat bot (see Fig. 3) are available in the bwJupyter - Jupyter AI profile. It is based on the official Jupyter profile Datascience and common packages for Python, Julia and R are already preinstalled. The external Large

⁵<https://grader-service.readthedocs.io/en/latest/index.html>

Language Models (LLMs) must be provided through API keys. Free versions are available through the KI-Toolbox⁶.

• ZOERR Integration

A wide range of teaching material is collected in the Central Repository for Open Educational Resources in Baden-Württemberg (ZOERR) [7]. All educational resources that contain Jupyter Notebooks can be accessed through bwJupyter. It ranges from introductory courses to the Python programming language⁷ to demonstrating how AI models work internally⁸. We encourage lecturers to publish their teaching materials through ZOERR.

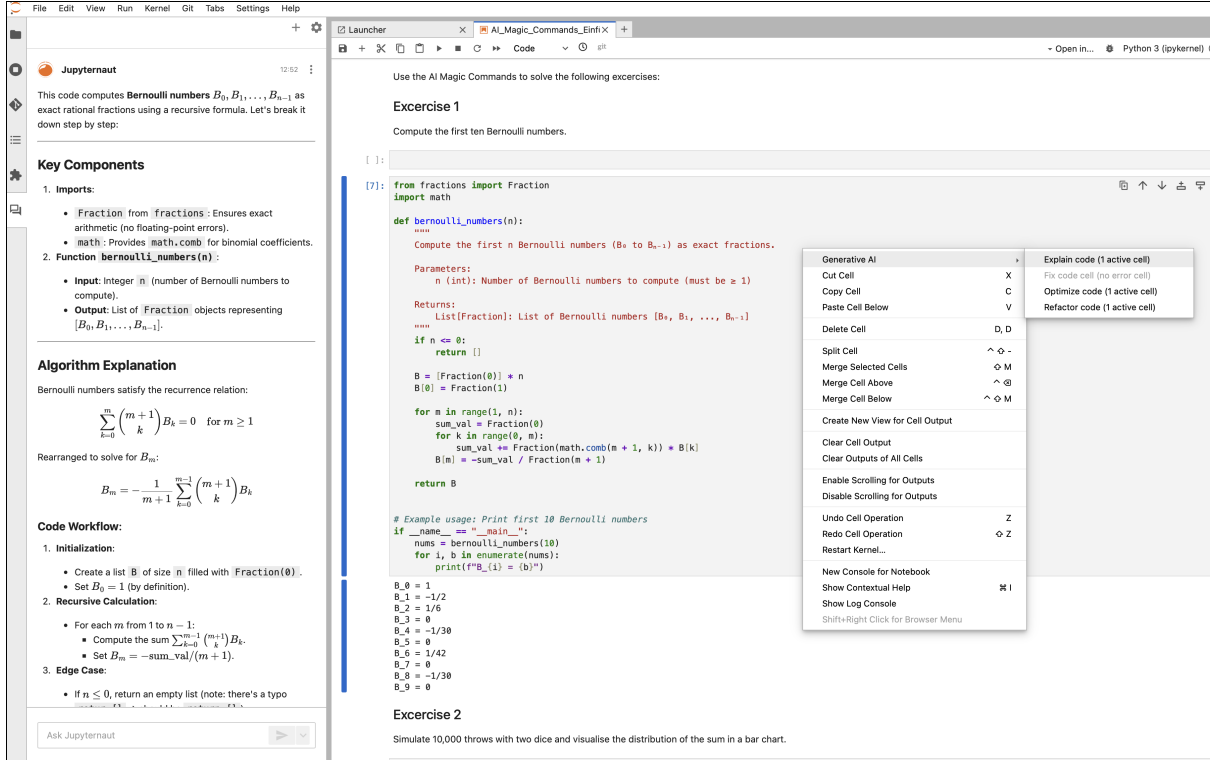


Figure 3: AI-supported tools in bwJupyter

5 Project Components and Implementation

The idea to develop a state-wide JupyterHub originated during the dialogue process *Future Lab: Universities in the digital world* in 2022.⁹ This round table, initiated by the Ministry for Science, Research and Arts Baden-Württemberg (MWK), and consisting of stakeholders from academia and politics, work-shopped numerous ideas to advance digitization in all areas of academic life. The realization of *bwJupyter for Teaching*, including the necessary hardware as well as 2 FTE (full time equivalent) software engineer positions, was subsequently funded by the MWK for the period from April 2024 until the end of 2025.



⁶<https://www.zml.kit.edu/english/ki-toolbox-en.php>

⁷<https://www.oerbw.de/edu-sharing/components/render/85288b70-1908-4da0-be11-282b8561c5e5>

⁸<https://www.oerbw.de/edu-sharing/components/render/2e67e51c-d2e5-4dcd-a7e5-1cd2e57dcd73>

⁹<https://mwk.baden-wuerttemberg.de/de/hochschulen-studium/dialogprozess-zukunfts-labor-hochschulen-in-der-digitalen-welt>

Starting in the winter term 2023/2024, a first pilot was implemented on a small partition of the bwUniCluster¹⁰, the joint high-performance computing system of Baden-Württemberg’s Universities. It was available to selected courses at KIT. Three notebook profiles (basic Python, Data Science with Python, R and Julia, GPU-Jupyter with TensorFlow and PyTorch) were accessible to approximately 700 students. During the following summer term, the pilot was extended to the universities in Stuttgart, Tübingen and Ulm, as well as the University of Applied Sciences Karlsruhe.

Simultaneously, new hardware was procured and a wide ranging survey to assess requirements and preferences of lecturers and students was conducted. The responses of the over 200 participants from 21 different universities revealed, that the majority of lecturers wants to use bwJupyter to demonstrate code examples during lectures, and as a platform to encourage students to study independently. Creating assignments and evaluating submissions in an automated manner as well as a collaborative work environment for students were further requests. Moreover, the survey asked which programming languages and software packages lecturers would like to use, and what computational resources would be required. Based on these insights, features to support the teaching and learning process were developed and implemented (see Section 4).

From April 2025 to May 2026, bwJupyter has been accessible to all university members in Baden-Württemberg. In its first year of operation, over 7,700 active users from 40 universities were registered. To advertise the new service and showcase the many possible applications, biannual community events and supplementary materials are provided. At the end of the funding period, the project transitioned into a self-financing state service operated by KIT. Based on the number of enrolled students, participating universities cover future hardware investments, operating expenses (e.g., energy), and staff costs.

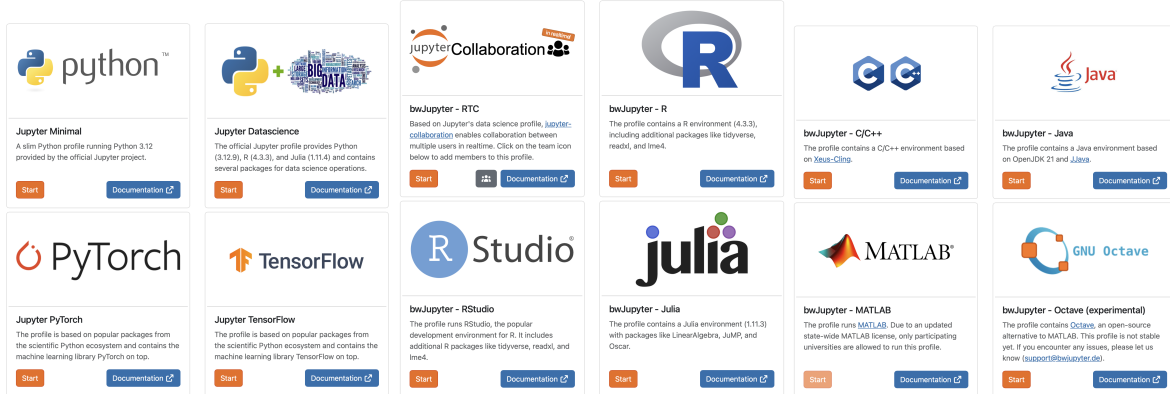


Figure 4: bwJupyter’s profile list, supporting multiple programming languages with preinstalled packages and tools

6 Technical Setup

Starting in 2014, Project Jupyter [2] is a nonprofit organization that develops open-source software and tools for interactive computing across different programming languages. It consists of several sub-projects that cover different aspects like user interfaces, kernels or notebook features. Regarding bwJupyter, the JupyterHub project [4] is the central platform users interact with. It is open-source, actively maintained, and widely used across education, research and industry. Its main capabilities are user authentication and user environment management. Due to the highly configurable framework architecture, plugins can be used to fully customize the logic of both of these tasks without interfering with JupyterHub’s source code.

¹⁰<https://wiki.bwhpc.de/e/BwUniCluster3.0>

- **Login:** For bwJupyter, we use the OAuth authenticator to integrate bwIDM¹¹, a state-wide identity management that facilitates access to all BW services. This allows any member of a BW university to sign in using their bwIDM university account without any further manual activation or registration.

In addition, temporary guest accounts can be generated for non-university users, e.g., outreach projects with schools or workshops for the wider public.

- **User Environment:** JupyterHub supports a wide range of technical implementations, from a single process on a local machine up to isolated containers on distributed systems and jobs in an HPC cluster. For bwJupyter, we schedule user environments as pods in a Kubernetes cluster. Thus, each environment is isolated and users cannot interfere with one another. In addition, each environment can be assigned the same amount of compute power, storage, and memory, so that every user receives the same resources.

Moreover, bwJupyter supports not just the native Notebook environment, but also popular data science and mathematical software tools such as RStudio. Users at universities participating in the statewide MATLAB funding can also run MATLAB using the state license. As an alternative, bwJupyter provides GNU Octave, an open-source MATLAB replacement. They can be accessed through dedicated bwJupyter profiles and launched on the JupyterHub, without any installation on the user's computer.

- **Compute Resources:** JupyterHub and the user environments run in Kubernetes¹², which combines multiple machines into a single cluster. The open-source project Zero to JupyterHub¹³ (Z2JH) provides a pre-defined configuration to deploy the JupyterHub on the Kubernetes cluster.

bwJupyter's cluster consists of 8 worker nodes, each containing 64 cores with 2 Intel CPUs, 1 TB RAM, as well as one GPU (NVIDIA L40S) with 48 GB vRAM. Since GPU resources are both valuable and limited, they are replicated using a time-slicing process. Currently, each user receives 2 GB of personal storage, 2 CPUs, and 8 GB of RAM. This allows for a total of 500 simultaneous users. Every bwJupyter profile receives an additional 10 GB storage within the `__shared` directory. All compute resources are subject to a fair-share policy. This concept proves sufficient, as 96% of users currently require 1 GB or less of personal storage. Due to the modular design of Jupyter Notebooks, CPU usage spikes only when code cells are executed. However, once execution is complete, CPU usage drops almost immediately back to 0 (idle).

- **Features:** JupyterHub is used by a broad range of companies and institutions, serving as a general-purpose application for many different use cases. Since bwJupyter's focus is on teaching and education, we extend JupyterHub with custom-developed plugins, to meet the students' and lecturers' requirements. The plugins, called hub-managed services, allow running custom applications alongside JupyterHub without modifying JupyterHub's source code (see Sec. 4).

7 Conclusion and Outlook

In just over a year, bwJupyter was able to establish itself as a new state-wide service with a diverse customer base. As of May 2026, bwJupyter has over 13,200 registered users and nearly 200 active custom profiles. These range from courses in Computational and Data Science to Physics and Engineering, as well as Biology, Chemistry, and Medicine - underlining the wide

¹¹<https://www.bwidm.de>

¹²<https://kubernetes.io>

¹³<https://z2jh.jupyter.org/en/stable/>

spread application fields. 13 of these profiles have more than 100 participants, indicating the usage in large, introductory courses. In the following we outline some bwJupyter use cases:

• Interactive Lecture Notes

The lecture Uncertainty Quantification at KIT provides the entire course material in form of Jupyter Notebooks (see Fig. 5). By combining mathematical notation, computations as code, and corresponding visualizations, an interactive learning environment is created. Students must fill gaps or solve exercises, and instant feedback guides them to the solution.

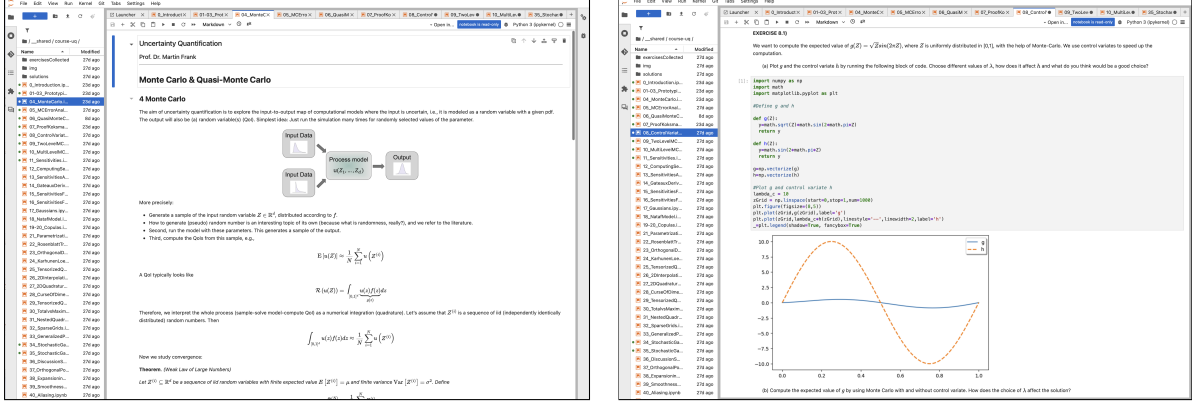


Figure 5: Lecture notes and exercises in Uncertainty Quantification, KIT

• Training and Exercises

With the grader service, lecturers can provide interactive exercise sheets with built-in tests. As described in Section 4, the entire exercise workflow is executed in bwJupyter.

The Department of Physics at KIT has been using Jupyter since 2021 by operating their own small JupyterHub. They designed Notebooks to support foundational training in the first and second semesters of their Bachelor's program and provided students with a standardized platform for data visualization and analysis in physics labs, and for their Bachelor's theses. Assignments in the form of Jupyter Notebooks replaced the previously required reports in text form. With the start of bwJupyter, the department moved their materials to the central bwJupyterHub, using a dedicated Docker image containing the relevant libraries and notebooks.

• Projects

In open teaching scenarios like seminars, workshops, or theses, Jupyter Notebooks are used as a blank space to realize own ideas and simulate real research scenarios. If necessary, Notebooks can be transferred to more powerful HPC clusters like bwUniCluster.

The Computational & Data Science Hackathon at KIT utilizes bwJupyter as a tool to access compute resources and document findings in form of a computational essay. During the week-long course, students work in teams on open-ended modeling problems. The real-time collaboration feature allows for simultaneous work on the same Notebook by multiple team members and the AI-supported programming assistant supports novice programmers.

While the development phase was characterized by establishing the framework of bwJupyter, we are now in normal operation and focus on maintaining the service long-term. This includes extending and refining our documentation, as well as reaching out to users on a regular basis.

Based on feedback and suggestions from lecturers and students, we maintain a small backlog. Besides the profiles we currently provide, we plan to release additional options covering new programming languages (e.g., C#) and applications (e.g., Robot Operating System (ROS))

and marimo). In addition to ZOERR, we are integrating the learning materials of the high-school outreach project CAMMP¹⁴ into bwJupyter, allowing teachers and students access via temporary guest accounts.

A frequently requested feature is the use of bwJupyter for digital examinations. Apart from legal issues, this raises interesting technical questions regarding availability, access and security. Taking into account the results of the *Partnership for Innovative E-Exams: Consortium of Universities in Baden-Württemberg (PePP)*¹⁵, we plan to examine these questions in the future.

Acknowledgements

The authors thank the Ministry of Science, Research and Arts Baden-Württemberg for funding the project. Moreover, the authors thank Ulrich Betzler, Jörn Beutner, Ulf Bischoff, David Boehringer, Martin Frank, Philip Hoyer, Matthias Leander-Knoll, Martin Nußbaumer, Günter Quast, Klaus Scheibenberger, Pascal Schuhmacher, Anett Seeland, Pascal Seeland and Thorsten Tüllmann for their valuable contributions to the project.

References

- [1] Lorena A. Barba, Lecia J. Barker, Douglas S. Blank, Jed Brown, Allen B. Downey, Timothy George, Lindsey J. Heagy, Kyle T. Mandli, Jason K. Moore, David Lippert, Kyle E. Niemeyer, Ryan R. Watkins, Richard H. West, Elizabeth Wickes, Carol Willing, and Michael Zingale. Teaching and learning with Jupyter. In <https://jupyter4edu.github.io/jupyter-edu-book>, December 2019. Accessed: 2026-05-27. URL: <https://jupyter4edu.github.io/jupyter-edu-book/>.
- [2] Brian E. Granger and Fernando Pérez. Jupyter: Thinking and storytelling with code and data. In *Computing in Science & Engineering*, volume 23, pages 7–14, 2021. doi:10.1109/MCSE.2021.3059263.
- [3] Jeremiah W. Johnson. Benefits and pitfalls of jupyter notebooks in the classroom. In *Proceedings of the 21st Annual Conference on Information Technology Education*, SIGITE '20, page 32–37, New York, USA, 2020. Association for Computing Machinery. doi:10.1145/3368308.3415397.
- [4] Project Jupyter, Matthias Bussonnier, Jessica Forde, Jeremy Freeman, Brian Granger, Tim Head, Chris Holdgraf, Kyle Kelley, Gladys Nalvarte, Andrew Osherooff, M Pacer, Yuvi Panda, Fernando Perez, Benjamin Ragan-Kelley, and Carol Willing. Binder 2.0 - reproducible, interactive, sharable environments for science at scale. In *SciPy 2018*, 2018. doi:10.25080/Majora-4af1f417-011.
- [5] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, and Kyle Kelley. Jupyter notebooks - a publishing format for reproducible computational workflows. In *Positioning and power in academic publishing: players, agents and agendas: proceedings of the 20th International Conference on Electronic Publishing*, page 87. IOS press, 2016.
- [6] Jens Köhler, Sebastian Labitzke, Michael Simon, Tobias Dussa, Martin Nussbaumer, and Hannes Hartenstein. bwidm – federated access to it-based services at the universities of the

¹⁴<https://www.cammp.online>

¹⁵<https://www.hnd-bw.de/projekte/pepp-2/>

- state of baden-württemberg. In *PIK - Praxis der Informationsverarbeitung und Kommunikation*, volume 37, pages 15–21, 2014. URL: <https://doi.org/10.1515/pik-2013-0025> [cited 2026-05-27], doi:doi:10.1515/pik-2013-0025.
- [7] Peter Rempis. *Open Educational Resources (OER) und das Zentrale OER-Repository (ZOERR) der Hochschulen in Baden-Württemberg—zu Präsentation und Nutzung zeitgemäßer Lehr-/Lernmaterialien*. Deutsche Nationalbibliothek, 2019.
- [8] Sarah Schönbrodt, Kirsten Wohak, and Martin Frank. Digital tools to enable collaborative mathematical modeling online. In *Modelling in Science Education and Learning*, volume 15, page 151–174, Apr. 2022. URL: <https://polipapers.upv.es/index.php/MSEL/article/view/16269>, doi:10.4995/msel.2022.16269.