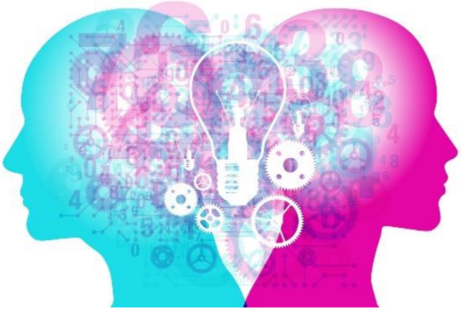


B.SC. THESIS

Jupyter Notebook as an ICT Tool in Education

Using Jupyter Notebook to Teach about Performance in Parallel Programming

MARUNA DERIEG

3rd of October 2021

Supervisor: Prof. Dr. Lena Oden ∗ Degree Program: B.Sc. Computer Science ∗ Matriculation Nr.: 6558020
FernUniversität in Hagen ∗ Faculty of Mathematics and Computer Science ∗ Chair of Technical Computer Science

Maruna Derieg

Jupyter Notebook as an ICT Tool in Education - Using Jupyter Notebook to Teach about Performance in Parallel Programming

Abstract

In this thesis the ICT tool Jupyter Notebook is evaluated for its suitability to teach students about performance in parallel programming. A set of Jupyter notebooks has been developed at Fern-Universität in Hagen to teach students about this topic. This project was evaluated with a semi-structured questionnaire, applying a mixed-method approach. In total, the notebooks have been tested and evaluated by 28 research participants. Most of the participants are students at the faculty of mathematics and computer science at FernUniversität in Hagen. The results showed an overall positive attitude towards the use of Jupyter Notebook in teaching. The participants liked especially the interactivity that the use of Jupyter Notebook enables. However, the participants also expressed conflicting opinions about some of the applied pedagogical methods, namely the conversational language style and the usage of speech-bubble images.

A literature review has shown that many educational institutions are already using Jupyter Notebook for various subjects. It has been noted, however, that most universities apply Jupyter Notebook complementary to traditional teaching methods and not as a medium for instructions. In the Jupyter notebooks that have been developed at FernUniversität in Hagen, Jupyter Notebook is not only used as an environment for exercises but also as a medium for teaching. For this purpose, multiple pedagogical methods have been applied, some of which can only be implemented with an ICT tool such as Jupyter Notebook. The developed notebooks interleave textual instructions with code sections, allowing students to test, discover and explore the learner theory. Further, integrated self-test questions with immediate and dynamic feedback were implemented, as well as dynamic graphics that can be adjusted by the learners. One feature that has been offered but not extensively used yet by the research participants is the ability to test various benchmarks for performance.

Maruna Derieg. *Jupyter Notebook as an ICT Tool in Education - Using Jupyter Notebook to Teach about Performance in Parallel Programming.* Faculty of Mathematics and Computer Science, FernUniversität in Hagen, 2021.

Author: Maruna Derieg
Cover Design: Maruna Derieg
Date: 3rd October 2021

Table of Contents

1 Introduction

This B.Sc. thesis is about Jupyter Notebooks as an ICT tool in education, and how we have used Jupyter Notebook to teach students about performance in parallel programming. Jupyter Notebook is a web based open-source application that was released in 2014. It allows users to *"create and share documents that contain live code, equations, visualizations and narrative text"*[1].

In the 21st century 'Information and Communication Technology' (ICT) is playing a crucial role in daily life [2]. The use of ICT tools in formal and non-formal education is gaining popularity and in many cases it has become indispensable. This holds especially true in the light of the Covid-19 pandemic, where millions of students ended up studying from home [3]. With most educational institutions being unprepared for this unexpected challenge, the importance of ICT in education has become quite evident [3], [4]. Digital learning is not just limited to students who study from a distance. Instead, the usage of ICT in education can be a valuable extension to traditional teaching methods for any student, regardless of educational level and discipline [5]. To make the most out of the potential digital learning has to offer, it is crucial to do further research and development (R&D) in this field.

Pedagogical methods should be based on research and backed-up by theory [6]. R&D in the field of education is often not a prioritized field of interest by governments and other institutions. In the year 2018, the government of the United States designated less than 0.15% of all R&D funding to the department of education [7]. In Germany, on the other hand, 4.7% of all R&D funding went to the sector of education in the year 2018 [8]. Evidence suggests that education has a positive influence on national wealth, the quality of economic and political institutions, and criminality rates [9]–[11]. Considering the important role education plays in the maintenance of a successful society, it is justified to invest in a good educational system. This includes R&D in the field of education, which serves as a basis for the ongoing evaluation and development of pedagogical methods and tools.

One area of education that could especially benefit from more R&D is the field of e-learning and digital learning. A study by Maia et al. [12] has shown that even though educators are interested in ICT tools to support teaching and learning processes, these tools rarely get adopted. There seems to be a discrepancy between what technology could make possible in the field of education and standard practices in teaching. There are many factors that determine the decision of educators to explore new methods in teaching [12]. One such factors is the availability of research results [13]. It is crucial to perform research in the field of education, and especially in the field of digital learning and e-learning. This will serve as a necessary decision basis for educational institutions to adjust current pedagogical practices and to adopt new ICT tools.

At FernUniversität in Hagen, a module called 'Parallel Programming' is offered to undergraduate students in each autumn semester. In 2020 the course has adopted Jupyter Notebook as a programming interface for programming exercises. The implementation with JupyterHub allows students to easily access computational resources located at the university.

Even though Jupyter Notebook is made available to the students as a programming environment, as per 2020, Jupyter notebooks were not used as a medium for instructions in the module 'Parallel Programming'. Since FernUniversität in Hagen is a university where students learn entirely by distance, it makes sense to also consider Jupyter notebooks as a medium for course instruction. The project of this thesis consists in the development of a set of interactive Jupyter notebooks which introduce students into the topic of performance in parallel programming. Depending on the success of the project, the developed material could become an integral part of the currently taught module about parallel programming.

It is worth noticing that Jupyter Notebook was not specifically developed for the purpose of digital learning [14]. Thus, there is currently no documentation which demonstrates to educators how Jupyter Notebooks could be used as a medium for interactive instructions. The objective of the developed Jupyter notebooks is not only to teach students about performance in parallel programming, but also to demonstrate some of the salient Jupyter Notebooks features that can be applied to teaching.

A first version of the developed Jupyter notebooks has been distributed to students who were willing to test and evaluate the learning material. The participants' opinion about the developed notebooks and their learning experience has been evaluated with a mixed-approach questionnaire. A long-term study about how the developed notebooks influence learning outcome is recommended for future works but would have been beyond the scope of this thesis.

1.1 Aim, Objective, and Sub-Objectives

The aim of this thesis is to shed light on the potential use of Jupyter Notebook as an ICT tool in education. To achieve this aim, we define the following objective:

Objective: To test and evaluate the ICT tool Jupyter Notebook for its suitability as a medium for teaching.

In order to achieve this objective, the following sub-objectives were taken into consideration:

Sub-objective 1: Analyze how Jupyter Notebook has been applied to teaching by other researchers and institutions.

Sub-objective 2: Demonstrate salient Jupyter Notebook features which are relevant when using Jupyter Notebook as a medium for instruction.

Sub-objective 3: Develop a set of Jupyter notebooks which teaches students about performance in parallel programming.

Sub-objective 4: Evaluate the students' learning experience and their opinion about the developed Jupyter notebooks.

Sub-objective 5: Identify areas where the current version of Jupyter notebooks can be further improved.

2 Background

Using Jupyter Notebook as an ICT tool for instruction can be considered non-traditional teaching. In this section we will first define commonly used terms with respect to non-traditional education and analyze how the presented learning modes relate to each other. Then will follow an introduction about computational notebooks in general and Jupyter Notebook in specific. The chapter ends with an overview about Project Jupyter and some of its subprojects that are relevant to education.

2.1 Definitions of Different Learning Modes and their Relationship

Numerous terms are used to describe non-traditional teaching methods, the most common of which are e-learning, digital learning, online learning, multimedia learning, and distance learning. These terms are often used interchangeably and without precision [15]. To avoid misunderstandings it is indispensable to clearly define and differentiate these terms. Further, we will analyze how different modes of learning differ from each other and where they overlap.

2.1.1 E-Learning

There is little consensus about what e-learning actually means [2], [15]. One definition is given by Clark and Mayer, who define e-learning as *"instruction delivered on a digital device that is intended to support learning"* [5, p. 8].

Clark and Mayer further differentiate between synchronous and asynchronous e-learning. Instruction-led learning that is presented at a fixed time is classified as synchronous e-learning, whereas learning designed for self-study that is available upon demand is considered asynchronous e-learning [4, p. 7].

The definition given by Clark and Mayer [4, p. 8] is not very precise, as it is unclear whether certain scenarios fall under the category of e-learning or not. Assuming a professor is presenting a face-to-face lecture with the help of a Jupyter Notebook, would this be considered e-learning according to Clark and Mayer [5]? Throughout this thesis we will only use the term e-learning if the instruction is delivered entirely on a digital device. Thus, the just mentioned scenario where a digital tool is used complementary to face-to-face instruction would not be classified as e-learning. Another situation that could raise questions is a student who studies a course with an e-book that is read on a 'kindle e-book reader'. Would this be considered e-learning? The instruction is delivered on a digital device, and the instruction is intended to support learning. The ICT tool 'e-book reader', however, has not contributed to any additional support in learning, since the student could also use a print-out version of the book without compromising on any learning support. Clark and Mayer [5] do not specify whether it is the instruction or the digital device that has to support learning. Considering this, an extended definition for e-learning will be used in this thesis. We define e-learning as *"instruction that is entirely delivered on a digital device, where the applied ICT is intended to support learning".*

2.1.2 Digital Learning

Basak et al. [2] state that the term e-learning is increasingly getting replaced by the term 'digital learning' and they consider e-learning to be a subset of digital learning (see Figure 1). In the 'Every Student Succeeds' Act (Public Law 114 of the United Stated) digital learning is defined as *"any instructional practice that effectively uses technology to strengthen a student's learning experience and encompasses a wide spectrum of tools and practices, including […]"* [16]. According to this definition,

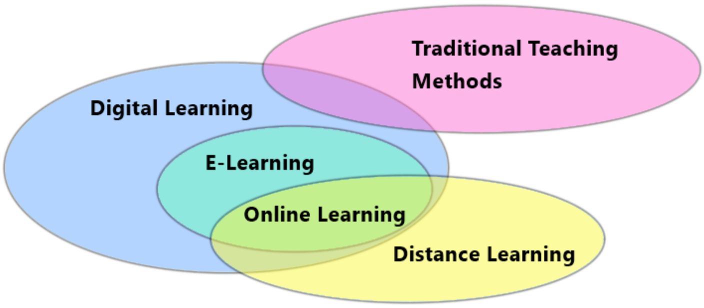


FIGURE 1: VENN DIAGRAM ABOUT DIFFERENT LEARNING MODES

the complementary use of Jupyter Notebook during a face-to-face lecture can be considered as digital learning, under the condition that the students' learning experience is strengthened due to the use of Jupyter Notebook. What is interesting with the given definition of digital learning is the emphasis that the used ICT needs to have a positive influence on the students' learning experience [16]. Whether or not an ICT tool can be considered a digital learning tool depends on the features that are offered. Learning with an e-book can only be classified as digital learning if the used software or device offers special features for which there is no 'paper-pen' alternative.

2.1.3 Distance Learning

Another term that seems to have a different meaning for different authors is 'distance learning' [15]. In this thesis we define the term 'distance learning' as, *"any type of learning where student and teacher are physically apart from each other during the learning process".*

According to this definition, distance learning can also happen without the use of digital devices and therefore it cannot be considered a synonym of e-learning or digital learning (see Figure 1). Distance learning can be synchronous as well as asynchronous [15].

2.1.4 Online Learning

The term 'online' indicates that a digital device is required and that the student and teacher are physically apart. Therefore, we define 'Online Learning' as *"a subset of e-learning and distance learning, where the instruction is delivered over a network"*.

2.1.5 Traditional Teaching Methods

This term is challenging to define because it can vary depending on historical time frame and cultural context [17]. We broadly define traditional teaching methods as *"synchronous face-to-face teaching, where teacher and student are physically present in a classroom environment".*

2.1.6 Multimedia Learning

Mayer defines Multimedia Learning as *"learning from words and pictures"* [5, p. 5]. He further specifies that pictures include static as well as dynamic visual content, such as videos and animations [5, p. 6]. According to this definition, the process of learning with a Jupyter notebook can be classified as multimedia learning, if the notebook complements instructional text with pictures, videos, animations, simulations, or any other type of graphic.

2.1.7 Relationship Between Different Learning Modes

Figure 1 displays the relationship between e-learning, digital learning, online learning, distance learning, and traditional teaching methods in the way these terms are referred to throughout this thesis. Multimedia learning is not included in this graphic, since it can be applied in all five learning modes.

Many more terms are used to refer to non-traditonal teaching methods, such as virtual learning, technology-mediated learning, online collaborative learning, web-based learning, blended learning, computer-based learning or computer-aided learning [15], [18]. There seems to be no consensus in literature about how exactly these terms are to be defined and differentiated from one another. It would be beyond the scope of this thesis to further elaborate on these terms.

2.2 Computational Notebooks

A computational notebook is a file that can be *"read like a journal paper and run like a computer program"* [19]. Working with computational notebooks allows programmers to quickly create and share code that can be written and executed in separate code cells, and interweaved with other media such as text, pictures, or videos. Thus, computational notebooks can be considered a suitable environment for interactive computing, allowing users to present their code in the form of a computational narrative [20].

The idea of interactive computing has its roots in applications such as IDL(1977), Maple(1982) and MATLAP(1984) [20]. The first application to provide a front-end GUI in the style of a Notebook was Mathematica, which was introduced in 1988 by Macintosh [21]. As the notebook interface became more popular, similar applications were developed for a wide range of programming languages, including Python, MATLAP and SQL [21]. As of 2021 there are over 60 providers of computational notebook systems that differ in various ways, such as how code is imported, how code and prose is edited, how code can be executed and how the notebook output can be published [22]. Lau et al. [22] have created an overview of the most popular environments for computational notebooks, and how they differ in the available features.

In general, a computational notebook offers the following features [21]:

- It allows to add code in separate cells that can be executed independently from another. This can facilitate debugging and make programming easier.
- It can display visual representations of data.
- It allows users to add other media such as text in between code cells.
- The different sections of the notebook can easily be rearranged.
- It can be used for various purposes, such as live presentations or interactive reports on collected data and data analysis results.

2.3 Jupyter Notebook

With over 10 million of public Jupyter notebooks available on GitHub in early 2021, Jupyter Notebook is arguably the most popular computational notebook environment as of today [20]. Many authors consider Jupyter Notebook the de facto standard for Data Science and Artificial Intelligence [23]–[25]. The question arises why Jupyter Notebook has become the computational notebook of choice for a majority computer scientist? Jeffrey M. Perkel [23] suggests that the reasons for this are recent improvements of the software such as integration with GitHub and Google Drive, the general maturation of scientific Python and Data Science, as well as the easy access to remote data from within the Jupyter Notebook environment.

It is important to understand that the term 'Jupyter Notebook' can refer to three different meanings:

1. A software called 'Jupyter Notebook', which was the first software released by Project Jupyter.
2. A computational notebook, created with one of the software released by Project Jupyter.
3. A programming environment provided by one of the software released by Project Jupyter.

The context usually makes it clear what meaning the term 'Jupyter Notebook' refers to. In this thesis the capitalized term 'Jupyter Notebook' stands for to the Jupyter Notebook environment, and the non-capitalized version 'Jupyter notebook' refers to a document, i.e. a computational notebook.

In addition to the general characteristics of computational notebooks, a Jupyter notebooks has the following key features [21]:

- It is internally represented in Java Script Object Notation (JSON).
- It is saved with the '.ipynb' extension.
- It can be converted to other filetypes, such as HTML, LaTeX, pdf, markdown or a Python script [26].
- It can be viewed outside of the Jupyter Notebook environment on platforms such as GitHub, Jupyter Notebook Viewer or Google CoLaboratory [27].
- It can easily be shared through mediums such as E-mail, GitHub or Moodle [21].
- It is created with software that is, as per 2021, completely free. Due to the modified BSD license under which Project Jupyter software has been released users have very few limitations with respect to usage and distribution [21].
- It can be created with a broad range of additional functionalities in the form of widgets and extensions [1]. Jupyter Notebook also allows third party extensions, making it possible to develop and add utilities designed for individual needs.
- It can support a wide range of programming languages. As of early 2021 Jupyter Notebook supports more than 145 kernels in over 40 different programming languages [28].
- Its kernel does not have to be running on the local machine. This opens possibilities to set up a system where users can access remote resources, such as a supercomputer located at a university [23].

Despite the many functionalities, working with Jupyter notebooks also has its drawbacks. Some authors have voiced concern that learning programming in the Jupyter Notebook environment could instill bad programming habits in students [29]. Programming with computational notebooks in general can be accompanied by bad practices regarding naming, versioning, testing and modularizing of code [29]. Another issue arises because code cells can be executed in any arbitrary order, which can lead to unexpected behavior and hidden states. Also the reproducibility of notebooks is not always guaranteed since the notebook format does not encode library dependencies with their associated versions [29]. In a paper called *"Ten simple rules for writing and sharing computational analyses in Jupyter Notebook"* Rule et al. [30] summarize recommended practices to avoid the just mentioned issues.

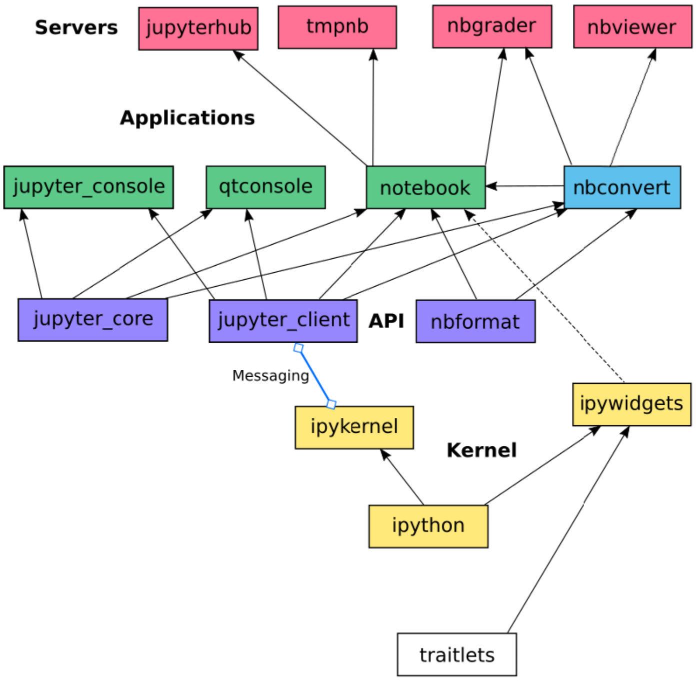


FIGURE 2: PROJECT JUPYTER OVERVIEW [31]

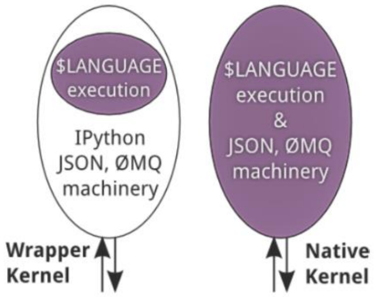

$LANGUAGE execution
IPython JSON, ØMQ machinery
Wrapper Kernel
$LANGUAGE execution & JSON, ØMQ machinery
Native Kernel

FIGURE 3: TWO TYPES OF KERNELS [31]

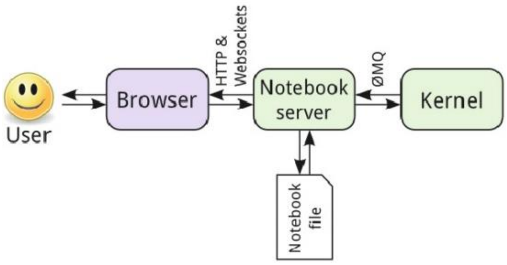


FIGURE 4: JUPYTER NOTEBOOK INTERFACE [31]

2.4 Project Jupyter

Jupyter Notebook has emerged out of 'Project Jupyter' which was initiated by Fernando Pérez [21]. The aim of this community-driven project is to *"to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages"* [1]. The name Jupyter stands for 'Julia', 'Python' and 'R', as these are the programming languages initially supported.

2.4.1 History

Project Jupyter submerged out of the IPython project, which originally was just a better console-based 'read-evaluate-print' loop for Python [19]. Conveniences of IPython included command-line history, command completion and built-in macros called 'magics', which have also found their way into the Jupyter Notebook environment. The big limitation of IPython was that it only supported the programming language Python. As a response to this, IPython was expanded to support multiple languages, which was the beginning of 'Project Jupyter'. As per 2021 the Jupyter Notebook environment can support over 40 different programming languages.

2.4.2 System Architecture

In the backend, Jupyter is still using the IPython kernel (see Figure 2). Frontend interfaces like Jupyter Notebook, JupyterLab or Qt console communicate with the IPython kernel through JSON messages that are sent over zeroMQ sockets [31]. One special feature of the IPython kernel is that it can be connected to multiple frontends simultaneously. In this case, the variables are shared amongst the different frontends. The motivation for this design was that it should be easy to develop different frontends based on the same kernel [31]. Another advantage of this architecture is that it allows to support new languages in the same frontends after developing a kernel of those languages.

New kernels for additional languages can be developed in two different ways: as wrapper kernels or as native kernels (see Figure 3) [31]. For languages that have good Python wrappers, such as Octave, it is easiest to write a wrapper kernel. Native kernels on the other hand, are more likely to be maintained by the community using them. Guidelines about how to develop kernels for Jupyter Notebook can be found in the documentation of Project Jupyter [1].

2.4.3 The Jupyter Notebook Interface

Project Jupyter offers three language agnostic clients, which serve as an interface for users to create, run and edit code (see Figure 2) [19]. The 'jupyter_console' is a plain-text frontend, and a graphical frontend is available with 'qtconsole'. The third interface, which is the most relevant one for this thesis, is the web-based notebook frontend.

Jupyter Notebook was the first notebook interface released by Project Jupyter in 2014, which replaced the IPython Notebook [1]. The next generation of Jupyter Notebook became available in 2018 with JupyterLab. One benefit of using the web-based notebook interface is the ability to store code and output together with markdown notes as a notebook file. When the user requests to save a notebook the data is sent from the browser to the notebook server, which then saves it on the drive as a JSON file [31]. Since it is the Notebook server that loads and saves notebooks and not the kernel, it is possible to read and edit notebooks even without having the specific kernel for the used programming language installed. In this case the code cells can be edited but not run. The kernel is only responsible for running code cells, it does not know anything about the notebook document.

Figure 4 displays the relationship between user, browser, notebook server and kernel.

2.4.4 JupyterHub

One sub-project that sets Project Jupyter apart from other providers of computational notebooks is JupyterHub. Since the Jupyter Notebook interface is developed as a web-based application, the kernel to run programs does not necessarily need to be located on the local machine. Instead it can be deployed on a remote server, allowing users to run programs over the web. JupyterHub is a cloud-based implementation of the Jupyter Notebook environment, providing a platform where multiple users can log-in and access a personal Jupyter Notebook workspace [32]. With JupyterHub universities and other institutions can provide remote access to computational resources such as supercomputers, allowing users to run computationally intensive programs. Another advantage of JupyterHub is that users can start coding right away after login, without having to deal with installation and maintenance tasks. The JupyterHub administrator can set up an identical Jupyter Notebook workspace for all users, with the required dependencies already installed. This guarantees that all users are working in the same environment.

JupyterHub is composed of four subsystems:

1. The Hub
2. A configurable HTTP proxy
3. Multiple single-user Jupyter Notebook servers
4. An authentication class

Requests from the client's browser are sent to the HTTP proxy. The Hub, also called tornado process, is the most crucial component of the JupyterHub system. At start-up it launches the HTTP proxy, which

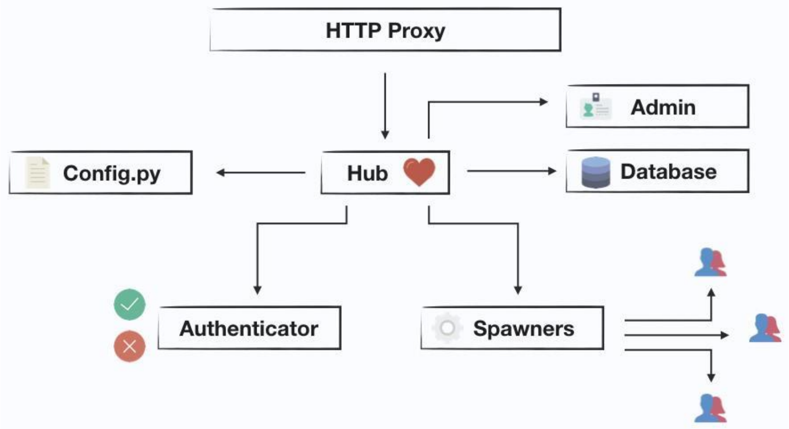


FIGURE 5: JUPYTERHUB COMPONENTS [32]

will then forward all request to the Hub. After successful login of the user the Hub spawns a single-user server that is monitored by a spawner. The Hub configures the proxy so that all URL prefixes are forwarded to the single-user notebook server. Optional configurations for the JupyterHub system can be defined in the `'config.py'` file and user kernels can be managed on the admin panel.

Figure 5 gives an overview about the different components of the JupyterHub system.

2.4.5 Other Software

Project Jupyter continues to further develop the offered functionalities which leads to many additional software components being released on a regular basis. In addition to the 4 user interfaces (Jupyter Notebook, JupyerLab, Jupyter Console and Qt Console) and JupyterHub there are IPyWidgets, extensions developed by Project Jupyter and third-party extensions. It would be beyond the scope of this thesis to present all these subprojects, but we will briefly summarize some which we consider as relevant for the topic 'Jupyter Notebook in education'. The subprojects mentioned here were not applied in the first version of the developed Jupyter notebooks but might be utilized in future versions.

Nbgrader

Specifically developed for education, nbgrader is a tool for creating notebook based assignments with options for manual grading, auto-grading and collection of students' feedbacks [33]. It also supports separate versions of the notebook for instructors and students, distribution of notebooks to students and collection of completed notebooks. Nbgrader is primarily deployed together with JupyerHub but can also be used without it. The automated distribution and collection of notebooks, however, is only supported with JupyerHub.

Nbgitpuller

Another extension that is often used together with JupyerHub, but also working on a local system, is nbgitpuller [34]. It offers an easy way to distribute git repositories without the recipients having to know anything about git. With nbgitpuller, teachers can distribute directories containing Jupyter notebooks and other files to students with a simple URL. When a student clicks on the link the git repository will be cloned into the student's home directory. Automated conflict resolution ensures the recipients do not have to deal with merge conflicts, in case there already exists a copy of the repository.

Voilà

The software Voilà was developed to facilitate communication of results [1]. One of the advantages of Jupyter Notebook is the interactive interface. When sharing results, however, a more simple and restricted presentation might be desirable. Voilà transforms Jupyter notebooks into stand-alone web applications that can be shared. It provides a secure and customizable interactive dashboard, that allows the author to control how the Jupyter notebook is presented to the reader [35]. In the context of education this could be a more user-friendly way of presenting educational material. Some students might feel intimidated by the Jupyter Notebook interface, especially if they do not have previous experience in programming. Depending on the subject taught and the functionalities needed, exposing students to Jupyter notebooks through Voilà instead of Jupyter Notebook or JupyerLab could be an alternative that allows students to focus on the content of the notebooks without being distracted by the interface.

3 Literature Review

Out of 39 scientific articles related to Project Jupyter we have identified thirteen which discuss how Jupyter Notebook has been used for teaching specific courses at various educational institutions (see Table 1). All these papers were published between 2018 and 2021, which indicates that the use of Jupyter Notebook in education is a newer phenomenon. It should be considered, however, that the Jupyter Notebook interface only exists since 2014 and that not everyone that starts using Jupyter Notebooks in teaching will publish a research paper about it. By May 2018 there were over 10'000 nbgrader-based Jupyter notebooks on GitHub [36]. This indicates that a large amount of Jupyter notebooks has been developed for educational purposes.

Our literature review revealed that Jupyter Notebook has successfully been used in subjects of computer science, engineering, physics, biology, chemistry, and earth science. Unfortunately, in many papers we have reviewed, it remains unclear to what extent Jupyter Notebook is used to replace traditional teaching methods. We also could not identify research results about the pedagogical advantages and disadvantages of learning in the Jupyter Notebook environment. A long-term study with a control group would be necessary to get a deeper understanding about how Jupyter Notebook can influence learning outcome. To our knowledge, as per early 2021, no such study has been published. Possible reasons for this could be lack of research interest, lack of funding, lack of time resources for a long-term study and/or lack of collaboration between faculties, researchers, professors, and board of directors. Much less resources are needed to evaluate students' opinion about a new teaching methodology, especially since general course evaluations are already a standard practice at most universities. Students' attitude towards the use of Jupyter Notebook in education was evaluated in 6 out of 13 publications discussed in this section.

3.1 Students' Attitude Towards Jupyter Notebook

At the University of Johannesburg, a Jupyter notebook was developed to teach 71 students about power systems in engineering [37]. 45 students have responded to a questionnaire evaluating their perception about the use of Jupyter Notebook in that course. The results revealed that the students had a positive attitude towards Jupyter Notebooks in teaching. They did, however, not consider it a useful tool for future work life.

Because of Covid-19 the Sonoma State University has replaced face-to-face labs in a course about quantitative analysis in chemistry with Jupyter notebooks [38]. To virtualize the laboratory learning experience, a HPLC simulator was written in the Jupyter Notebook environment. This allowed students to get some additional insights that they usually do not get in the face-to-face version of the exercise. Out of 18 students, 11 responded to a questionnaire which evaluated their opinion about the virtual labs. Several students reported that the virtual labs were more convenient for them because they could be accessed at any time. All the participants agreed to the statement that the HPLC simulator should be used in future as well, complementary to the in person HPLC experiment. Further, the students stated that they enjoyed the exercises, and 6 out of 11 students felt like they wanted to learn more about Python and computing in chemistry. Two students reported that they felt intimidated by the exercises because of Python. The notebooks were deployed on 'Chem Compute', a server that gives students from any university access to a remote Jupyter Notebook environment [39].

At the Lewis & Clark College Portland an ecosystem of tools specially tailored for education was developed [40]. With the usage of JupyterHub the authors set up an instance of Jupyter Notebook on a server, which enables users to run Jupyter notebooks directly on the universities high performance computing (HPC) system. In addition, the authors developed a Jupyter Notebook extension that makes it easy for teachers to assign notebook assignments or tests to selected students. Once the set deadline for completion is reached, a copy of the updated notebook is automatically copied back into the teacher's private directory. This system has been used at Lewis & Clark College by multiple professors, in courses about biology, machine learning, computer and network security, and in a CS1 course. In these courses, the Jupyter Notebook environment was used complementary to traditional teaching methods. Further, the authors have created a set of Jupyter notebooks which is designed as an interactive self-guided course about HPC. An evaluation revealed that students perceived the Jupyter notebooks about HPC as more useful than a traditional textbook. In the survey one student said: *"I really liked having the cells throughout the chapter to explain and showcase the material from that section"*. Another student said that it was nice to be able to run cells, as this allowed to change values and see how this effects the output.

At the Nayarbeyev University in Kazakhstan, Jupyter Notebooks were developed to teach students about the mechanism of chemical reactions [41]. The notebooks were used in computer-based lab-sessions, complementary to face-to-face lectures in a course about Advanced Chemical Reaction Engineering. According to the authors the students gave very positive feedback with respect to the content and implementation of the lab-sessions. The students felt that experimenting with Jupyter Notebook, and analyzing results, helped them better understand the theory and get practical experience as well. Further, the students liked the Jupyter Notebook interface, the interactivity, and the ability to visualize results.

An engineering course at the University of Vigo in Spain has integrated Jupyter Notebook into a course about manufacturing processes in mechanical engineering [42]. Jupyter notebooks were used for lab works and problem discussion sessions. A questionnaire surveyed the students' opinion about the use of Jupyter Notebook in the course; 26 students responded to the questions. The results of the evaluation showed that students highly agreed with the effectiveness of Jupyter Notebook as an ICT tool in education. The authors also noticed that the average score in the exam was higher that year compared to the previous year where Jupyter Notebook was not used yet. Thus, the authors conclude that the use of Jupyter Notebook had a positive effect on academic performance.

One study with mixed results regarding students' attitude towards Jupyter Notebooks was conducted at the University of California Merced [43]. The authors have developed a set of Jupyter notebooks that introduce students into the programming language Python while applying this language to solve chemical problems. The notebooks were used complementary to an undergraduate course about analytical chemistry. Compared to the evaluations we have just discussed, this university did not only measure the students' opinion after working with Jupyter Notebook, but also before. The researchers compared the students' opinion about ICT in education before and after working with Jupyter Notebook. The results showed that the average attitude towards computers in education slightly shifted to more negative after working with Jupyter Notebook. Unfortunately, the authors did not present further research results explaining why this was the case. It would have been interesting to know if the average attitude shifted because of a few individual students who might have developed a strong aversion towards programming and/or Jupyter Notebook, or because most students developed a slightly worse attitude towards it. A qualitative analysis of why the shift happened with different students would also have been insightful.

3.2 Jupyter Notebooks in University Courses

In addition to the six publications just summarized, we identified five articles that report the usage of Jupyter Notebook in certain university courses without an evaluation being done. Further, we found two articles about open-source projects that aim at making Jupyter notebooks about a certain topic available to the public.

At the University of Victoria in Canada the programming language for a CS1 course has changed from Java to Python [14]. The author decided that Jupyter Notebook would be the most suitable programming environment for the course. Jupyter Notebook was used in lab-sessions, as well as in lectures to demonstrate code examples.

In response to the Covid-19 pandemic many universities had to switch to online classes. The University of KwaZulu-Natal in South Africa was one of them [24]. Jupyter Notebook was chosen to replace face-to-face labs in a course about molecular biology. According to the author, the students were able to analyze different data sets in the Jupyter Notebook environment without having any prior coding experience.

The University of Coimbra in Portugal used Jupyter Notebook in an undergraduate course about data analysis and transformation [44]. The authors suggest that computational notebooks *"can contribute to explore innovative approaches and improve teaching and learning activities in different high educational courses, especially in engineering subjects"*. The authors also report improved students' participation and performance in the course where Jupyter Notebook was used.

A new paradigm of teaching was tried out at the University of Madeira in Portugal. The idea of using an agile approach supported with the use of Jupyter Notebook was implemented in a CS2 course [45]. The agile approach in education that is presented in the published article includes four principles: 'students over traditional processes and tools', 'working projects over comprehensive documentation', 'student and instructor collaboration, over rigid course syllabi', and 'responding to feedback rather than following a plan'. The authors consider Jupyter Notebook a suitable environment to support this agile approach of teaching.

Most universities mentioned so far used Jupyter Notebook only in lab-sessions. This is not the case for the Technical University of Hamburg in Germany, who also used Jupyter Notebook to partially replace paper-based exams [46], [47]. In a course about 'Machine Learning in Logistic' Jupyter Notebook was applied in labs and as a medium for exam completion. The author points out that special consideration must be taken to ensure no laws are violated with the use of Jupyter Notebook in exams. In order to conform to the law of the district, the exam results had to be converted into a specific format and archived for a certain amount of time. The students accessed the exams through their account on JupyterHub. Special testing was performed prior to the exam date to ensure that the server would not be overloaded.

3.3 Open-Source Projects

The Internet has made it possible that knowledge is widely available to everyone. Open educational resources have become increasingly available and popular [48]. We have identified two articles that discuss how Jupyter Notebook has become a medium to create and share educational material.

PyRosetta is a Python package developed by Johns Hopkins University that offers an interactive platform for protein structure prediction and design [27]. A series of Jupyter notebooks has been developed that introduces users into the fundamental principles of PyRosetta and its features. On the GitHub page where the Jupyter notebooks are made available to the public the users are encouraged to contribute to the current collection [49]. Thus, PyRoesetta can be considered a community driven project. The Jupyter notebooks are designed to be used either in a classroom context or for individual self-study, and they have already been used in multiple courses and workshops.

Another community driven project is 'Seismo-live', a public collection of educational Jupyter notebooks for Seismology [50]. The goal of this project is to provide high-quality interactive programming tutorials and exercises for training in seismology. Topics that are covered so far include data processing, computational seismology, earthquake physics and reproducible papers and graphics. The founders of 'Seismo-live' suggest that the notebooks can be used in workshops, in academic training and as seeds for seismological research projects. Early usage has shown a global interest in the project, which is publicly available on GitHub [51].

3.4 Conclusion

The literature review has shown that there are many ways how educational institutions can integrate Jupyter Notebook as an ICT tool in teaching. All the literature presented is summarized in Table 1.

One of the differences we have identified are the ways how students access the Jupyter Notebook environment, which could either be locally on their personal system, or through a server from the university with the usage of a JupyterHub account. Further, it seems that the courses taught with the help of Jupyter Notebook mostly belong to the departments of Computer Science, Engineering, Chemistry, Life Science and Earth Science.

Even though Jupyter Notebook is a tool that can be used to completely replace traditional teaching methods, most universities do not seem to use this feature. More likely, universities will stick to the traditional way of delivering a course, which is teaching the theory in form of lectures and applying the theory in exercises during separate lab-sessions. The literature review has revealed that Jupyter Notebook is most likely used in lab-sessions, which could be held face-to-face or by distance. Most articles published about the topic have not disclosed how much textual instructions and theory is present in the developed Jupyter notebooks. This makes it difficult to estimate whether Jupyter Notebook is merely used as a programming environment, or as an innovative tool for teaching.

Table 1 gives an overview about the literature that has been discussed in this section.

Educational Institution	Year	Title of Publication	Topic Taught	Usage of Jupyter Notebook	Access	Student Evaluation	Open-source	Refe-rence
University of Johannesburg, South Africa	2021	Student's Perception about the Use of Jupyter Notebook in Power Systems Education	Engineering - Power Systems in Engineering	Unknown	Unknown	Yes	No	[37]
Sonoma State University, USA	2020	Online Data Generation in Quantitative Analysis: Excel Spreadsheets and an Online HPLC Simulator Using a Jupyter Notebook on the Chem Compute Website	Chemistry - Quantitative Analysis	- Lab-sessions	JupyterHub	Yes	No	[38]
Lewis & Clark College Portland, USA	2019	Jupyter notebooks and user-friendly HPC access	- Biology - Machine Learning - Computer and Network Security - CS1 - High Performance Computing	- Complementary to course work - Self-study	JupyterHub	Yes	No	[40]
Nazarbayev University, Kazakhstan	2019	A set of Jupyter notebooks for the analysis of transport phenomena and reaction in porous catalyst pellet	Chemical and Materials Engineering - Advanced Chemical Reaction Engineering	- Lab-sessions	Unknown	Yes	No	[41]
University of Vigo, Spain	2018	Teaching optimization of manufacturing problems via code components of a Jupyter Notebook	Mechanical Engineering - Manufacturing Processes	- Problem-discussing sessions - Lab-sessions	Local	Yes	No	[42]
University of California Merced, USA	2020	Series of Jupyter Notebooks Using Python for an Analytical Chemistry Course	Chemistry and Chemical Biology - Analytical Chemistry	- Complementary to course work	Local	Yes	No	[43]
University of Victoria, Canada	2019	Jupyter notebook in CS1: An experience report	Computer Science - CS1	- Lab-sessions - Demonstration of Python in lectures	Local	No	No	[14]
University of KwaZulu-Natal, South Africa	2020	Analyzing biological models and data sets using Jupyter notebooks as an alternate to laboratory-based exercises during COVID-19	Biology - Molecular Biology	- Lab-sessions	Local	No	No	[24]
University of Coimbra, Portugal	2019	Using the Jupyter Notebook as a Tool to Support the Teaching and Learning Processes in Engineering Courses	Engineering – Data Analysis and Transformation	unknown	Unknown	No	No	[44]
University of Madeira, Portugal	2019	Agile approach to a CS2-based course using the Jupyter notebook in lab classes	Computer Science - CS2	- Lab-sessions	Unknown	No	No	[45]
Technical University of Hamburg, Germany	2019	Mit Jupyter Notebooks prüfen	Interdisciplinary - Machine Learning in Logistic	- Lab-sessions - Exams	JupyterHub	No	No	[46]
Johns Hopkins University, USA	2020	PyRosetta Jupyter Notebooks Teach Biomolecular Structure Prediction and Design	Computational Molecular Biophysics - PyRosetta	Designed to be used in - Classroom context - Self-study	Flexible	No	Yes	[27]
Seismo-live	2018	Seismo-live: An Educational Online Library of Jupyter Notebooks for Seismology	Quantitative Earth Science - Seismology	Designed to be used in - Classroom context - Self-study - Seeds for seismological research projects	Flexible	No	Yes	[50]

TABLE 1: LITERATURE REVIEW

4 Project

A set of Jupyter notebooks has been developed, which introduce students into the topic of performance in parallel programming. The challenge of this project was not only to introduce the students into the theory of the topic, but also to leverage and demonstrate Jupyter Notebook's potential as an ICT tool in e-learning.

In this chapter we will first elaborate on the objectives that this project aims to achieve. The theoretical foundations that the students should understand after working through the notebooks are summarized in section 4.2. Presenting theory without applying didactics, however, does not maximize a student's potential for learning. Therefore, we have applied a combination of pedagogical principles which are discussed in section 4.3. The chapter ends with a detailed description of how different features of the notebooks were implemented.

4.1 Objective

In general, a university course about parallel programming not only teaches the chosen parallel programming language, but also performance related know-how. Students should be aware in what ways the parallelization of programs influences performance. Topics related to performance are measurements such as speed-up and efficiency, as well as strong and weak scalability. The later can be theorized with Amdahl's law and Gustafson's law, respectively. A summary of these themes is given in section 4.2. One of the sub-objectives of this thesis is to teach students about these topics in an explorative and interactive teaching style. Not only should the students gain a deep understanding of the topic, but they should also retain the knowledge well and be able to apply it. To address this challenge, multiple pedagogical methods have been applied, which will be discussed in section 4.3.

Another sub-objective of this thesis is to demonstrate how Jupyter notebooks can be used as the main medium of instruction in e-learning. The literature review has revealed that many universities are using Jupyter Notebook as a programming interface in lab works. Teaching an entire course by distance with Jupyter Notebook seems to be less common. One reason for this could be the fact that Jupyter Notebook was not primarily developed for this purpose [14]. We identified a few books which introduce the reader into the usage of Jupyter Notebook as a programming interface [21], [26], [52], but no literature could be found about how to use Jupyter Notebook specifically for teaching. Thus, one of the sub-objectives of this thesis is to demonstrate some of the salient Jupyter Notebook features that are relevant when creating e-learning material.

4.2 Theoretical Framework

The main notebook starts with a discussion about the motivation behind parallel programming, and then introduces the students into the two most used metrics for performance, which are speed-up and efficiency. Parallelizing a program will not always increase performance, due to parallel overhead. The students are taught about different kinds of overheads and how this influences performance. When analyzing the performance of a parallel program it is also important to know how the program scales. We introduce the students into the concept of strong and weak scalability, and commonly used terms related to scalability analysis. An upper-bound for speed-up can be calculated with Amdahl's law in strong scaling, and with Gustafson's law in weak scaling. The students are introduced into the concept and mathematical formulas of these laws, as well as their limitations.

In addition to learning the theory, students are also encouraged to experiment with the learned concepts themselves. For this purpose, a series of benchmarks together with additional notebooks was made available to the students. We also developed some notebooks which discuss topics that only a subgroup of students might be interested in. There is, for example, a notebook which introduces students into different methods for measuring run-time in Jupyter Notebook.

In this section we will summarize the theoretical basics that the students should understand after working through the notebooks.

4.2.1 Motivation for Parallel Programming

The main motivation behind parallel programming is to increase performance [53, p. 3]. Performance is most commonly associated with speed, but can also refer to power consumption [54, p. 4]. With parallel computing it is possible to either:

1. Execute a program in the same amount of time, but with reduced power consumption,
2. or execute a program with the same amount of power consumption, but with reduced execution time.

The developed notebooks only discuss the aspect of performance with respect to execution time. However, a link to an external video hosted on YouTube[1] is given in the notebook, allowing interested students to get further information about how parallel programming relates to power consumption.

4.2.2 Metrics for Performance: Speed-up and Efficiency

Two of the most used metrics for performance in parallel computing are speed-up and efficiency. For the calculation of speed-up in the context of parallel programming the run-time of the serial program is compared to the run-time of the parallelized version of the program. Speed-up is a metric to describe how much faster the parallelized program runs compared to the serial version of the program.

The formula for speed-up is [53, p. 58]:

$$\text{Speed-up} = \frac{W_1}{W_p} \tag{1}$$

with
$W_1 =$ wall-time of the serial program
$W_p =$ wall-time of the parallel program, run with p parallel processing units

In most cases, W_1 is calculated by measuring the run-time of the serial version of the algorithm, that has been parallelized [53, p. 59]. An alternative option is to measure the run-time of the fastest serial algorithm available for a specific problem [6, p. 139].

Another commonly used metric for performance is efficiency which gives a measurement for how much each parallel processing unit is contributing to the speed-up.

The formula for efficiency is [53, p. 58]:

$$\text{Efficiency} = \frac{\text{Speed-up}}{p} = \frac{\frac{W_1}{W_p}}{p} = \frac{W_1}{p \cdot W_p} \tag{2}$$

[1] https://youtu.be/cMWGeJyrc9w (accessed on 5th June,2021)

with
p = number of parallel processing units

4.2.3 The Effect of Overhead on Performance

In general, parallelization only makes sense for larger problem sizes. The reason for this is the overhead caused by parallelization.

The total execution time of a program can be divided into the following blocks [55, p. 339]:

- t_{cpu} = time spent for calculations
- t_{com} = time spent for exchanging data between processes
- t_{wait} = time spent waiting
- t_{syn} = time spent for the actual synchronization of the used processes
- t_{place} = time spent for allocating tasks to individual processes
- t_{start} = time spent to start the parallel tasks on all processes

Bengel et al. [55, p. 340] define setup time and overhead as:

- $\text{setup time} = t_{place} + t_{start}$
- $\text{overhead} = t_{com} + t_{wait} + t_{syn}$

Other authors count the setup time as a part of the parallel overhead. Pacheco [53, p. 139] defines parallel overhead as *"the part of the parallel run-time that's due to any additional work that isn't done by the serial program"*.

In addition to the overhead caused by parallelization, there are other types of overheads which can distort calculations of speed-up and efficiency. The use of the Jupyter Notebook interface leads to a small overhead caused by the communication time between the kernel, Jupyter Notebook server and browser (see Figure 4). Also, the use of the Python module 'timeit' for time measurements leads to an overhead and there is an operation system overhead as well [54, p. 206]. The larger the execution time of the program, the less significant these kinds of overheads become [55, p. 340].

4.2.4 The Concept of Strong and Weak Scalability

In the context of computer science the term 'scalability' is generally used to indicate the *"ability of hardware and software to deliver greater computational power when the amount of resources is increased"* [56]. Applied to parallel programming one can say that a program scales, if an increase in speed-up can be observed upon increasing the number of parallel tasks. The terms 'poorly scalable' and 'highly scalable' are often used to describe the degree of scalability [54].

Students who are new to the topic of HPC could easily assume that the expression 'poorly scalable' refers to 'weak scalability'; and 'highly scalable' to 'strong scalability'. This would be a wrong association, since the terms 'weak' and 'strong' do not indicate the degree of scalability, but the type of scaling analysis performed.

Strong Scalability

If speed-up or efficiency are calculated for an increasing number of parallel processing units while keeping the problem size fixed; then a program is analyzed for strong scalability. The question asked is: *"How does speed-up/efficiency change if the number of threads is increased while keeping the problem size constant?"*

A program is considered 'strongly scalable' if it is possible to increase the number of parallel processing units without increasing the problem size, while achieving the same efficiency for all measurements [53, p. 62].

In strong scalability analysis, an upper-bound for speed-up is given with Amdahl's law.

Weak Scalability

If speed-up or efficiency are calculated for an increasing number of parallel processing units while simultaneously the problem size is increased at a constant rate, then the program is analyzed for weak scalability. The question asked is: *"How does speed-up/efficiency change if the number of threads is increased while correspondingly also the problem size is increased?"*

A program is considered 'weakly scalable' if it is possible to increase the number of parallel processing units while correspondingly increasing the problem size at a constant rate, achieving the same efficiency for all measurements [53, p. 62].

In strong scalability analysis, an upper-bound for speed-up is given with Gustafson's law.

4.2.5 Amdahl's Law

Amdahl's law allows to calculate the maximum speed-up possible for a specific program and a given number of processors, assuming the problem size remains fixed.

Serial and Parallelizable Part of a Program

When analyzing the performance of a parallel program the run-time is usually measured as 'wall-time', which is the elapsed time between the start and the end of the program [53, p. 65]. The wall-time of the parallelized program can be divided into two parts: the program part that can benefit from parallelization and the part that cannot.

$$\mathrm{W_1} = \mathrm{W_{ser}} + \mathrm{W_{par}} \tag{3}$$

with
$\mathrm{W_1}$ = total wall-time of the parallelized program, run with one parallel processing unit
$\mathrm{W_{ser}}$ = total wall time of the program parts which are inherently serial
$\mathrm{W_{par}}$ = total wall time of the program parts which can be parallelized

With this notation it is possible to give a lower-bound for the wall-time of the parallelized program:

$$\mathrm{W_p} \geq \mathrm{W_{ser}} + \frac{\mathrm{W_{par}}}{\mathrm{p}} \tag{4}$$

with
W_p = total wall-time of the parallelized program, run with p parallel processing units

Upper-bound for Speed-Up

To get the maximum possible speed-up we can just insert the equations (3) and (4) into the speed-up function (1), and we get [57, p. 32]:

$$\text{Speed-up} \;=\; \frac{W_1}{W_p} \leq \frac{W_{ser} + W_{par}}{W_{ser} + \frac{W_{par}}{p}} \tag{5}$$

It is possible to further simply this equation by replacing the absolute values of W_{ser} and W_{par} with a fraction value:

$$W_{ser} = W_1 \cdot f \tag{6}$$

$$W_{par} = W_1 \cdot (1 - f) \tag{7}$$

with
$f \;=\;$ fraction of the program which is inherently serial
$1 - f \;=\;$ fraction of the program which can be parallelized

Amdahl's Law

By inserting equation (6) and (7) this into the formula (5) for 'maximum speed-up' we get Amdahl's law [57, p. 32], which is:

$$\text{Speed-up} \;\leq\; \frac{W_{ser} + W_{par}}{W_{ser} + \frac{W_{par}}{p}} \;=\; \frac{W_1 \cdot f + W_1 \cdot (1-f)}{W_1 \cdot f + \frac{W_1 \cdot (1-f)}{p}} \;=\; \frac{f + (1-f)}{f + \frac{1-f}{p}} \;=\; \frac{1}{f + \frac{1-f}{p}} \tag{8}$$

From Amdahl's law we can also derive a formula for the maximum possible efficiency, which is:

$$\text{Efficiency} \;\leq\; \frac{\frac{1}{f + \frac{1-f}{p}}}{p} \;=\; \frac{1}{f \cdot p + (1-f)} \tag{9}$$

If the number of processors is infinite, then the upper-bound for speed-up is:

$$\lim_{p \to \infty} \frac{1}{f + \frac{1-f}{p}} = \frac{1}{f} \tag{10}$$

The conclusion we can draw from Amdahl's law is that no matter how powerful a machine is, the speed-up will never be more than $\frac{1}{f}$.

Limitations of Amdahl' Law

- Amdahl's law fails to consider the effects of varying problem sizes.
- Amdahl's law assumes that all parallel regions use the same number of threads. In practice this is not always the case. Each parallel region might have a different number of parallel processing units. Another common scenario is to have a loop, and a different number of parallel processing units for each run of the loop. The upper-bound for speed-

up calculated with Amdahl's law is still correct in these cases because it only gives an 'upper-bound'. However, one could get a more precise upper-bound for a certain number of parallel processing units by adjusting the formula of Amdahl's law. Trobec et al. [58, p. 42] presented a generalization of Amdahl's law which considers different numbers of parallel processing units for different parallel regions.

- Amdahl's law does not give an approximation for the true speed-up. The true speed-up can vary greatly from the upper-bound for speed-up calculated with Amdahl's law, due to parallel overhead.
- Amdahl's law assumes that f is fixed. In practice it can be observed that the ratio of the parallelizable part of the program to the non-parallelizable one changes with a change of the problem size [59].

4.2.6 Gustafson's Law

Amdahl's law does not consider any change in problem size, but keeps it fixed instead. What can be often observed, however, is a general trend for the problem size of applications to adjusts to the hardware capacities. A lot of computer games, for example, leverage hardware to its maximum and would often run poorly on hardware that was developed five years ago. This can be observed not only with computer games but with a wide range of applications [60, p. 61]. As the hardware becomes more powerful, the programs developed to run on that hardware are designed to handle accordingly large problem sizes. Further, the additional memory that becomes available with the usage of multiple CPU's allows to run parallel programs which would not be runnable otherwise.

These observations have caused Gustafson to suggest that *"speed-up should be measured by scaling the problem to the number of processors, not by fixing the problem size"* [59]. Gustafson's law provides a formula for calculating an upper-bound for the 'scaled speed-up'. That is, the speed-up achieved if the problem size is increased at the same rate as the number of processors.

Scaling Function of the Serial and Parallelizable Part of a Program

When the problem size is increased it can often be observed that the run-time of the parallelizable part of the program increases faster than the run-time of the inherently serial part of the program. Thus, with a change of the problem size the value for f changes as well; as the problem size increases, f decreases. To what degree this is the case will depend on the program.

Schmidt et al. [57, p. 33] define the scaling functions α and β as:

α = *"scaling function of the part of the program that does not benefit from parallelization with respect to the complexity of the problem size"*

β = *"scaling function of the part of the program that benefits from parallelization with respect to the complexity of the problem size"*

The scaling functions α and β take as input the problem size and return the factor by which the inherently serial part and parallelizable part decrease or increase.

With the use of α and β we can give a more generalized formula for the calculation of the serial runtime, which accounts for different problem sizes [57, p. 33]:

$$W_1 = \alpha \cdot W_{ser} + \beta \cdot W_{par} = \alpha \cdot f \cdot W_1 + \beta \cdot (1 - f) \cdot W_1 \tag{11}$$

This formula allows to calculate the run-time of a serial program for various problem sizes. The problem size will be the input of the scaling functions α and β. The output of these functions will determine the factor by which the run-time of the two program parts have to increase to account for the increased problem size.

Equation 11 can be adjusted to calculate the wall-time of a parallelized program with various problem sizes by dividing the parallelizable part with p. The result is only a lower-bound for the wall-time since the effect of parallel overhead is not accounted for:

$$W_p \geq \alpha \cdot W_{ser} + \frac{\beta \cdot W_{par}}{p} = \alpha \cdot f \cdot W_1 + \frac{\beta \cdot (1-f) \cdot W_1}{p} \tag{12}$$

Scaled Speed-Up

Dividing W_1 by W_p results in a upper-bound for the scaled speed-up [57, p. 34]:

$$\begin{aligned} \text{Scaled Speed-up } &= \frac{W_1}{W_p} \leq \frac{\alpha \cdot W_{ser} + \beta \cdot W_{par}}{\alpha \cdot W_{ser} + \frac{\beta \cdot W_{par}}{p}} = \frac{\alpha \cdot f \cdot W_1 + \beta \cdot (1-f) \cdot W_1}{\alpha \cdot f \cdot W_1 + \frac{\beta \cdot (1-f) \cdot W_1}{p}} \\ &= \frac{\alpha \cdot f + \beta \cdot (1-f)}{\alpha \cdot f + \frac{\beta \cdot (1-f)}{p}} \end{aligned} \tag{13}$$

This equation can be further simplified. Instead of using two separate scaling functions α and β it is possible to only consider the ration of the two:

$$\text{Scaled Speed-up} \leq \frac{f + \gamma \cdot (1-f)}{f + \frac{\gamma \cdot (1-f)}{p}} \tag{14}$$

$$\text{with}$$
$$\gamma = \frac{\alpha}{\beta}$$

The value for γ can either be [57, p. 34]:

- $\gamma = 1$
- $\gamma = p$
- $\gamma =$ any other function depending on p

The first case of $\gamma = 1$ is true if α=β, which means that f does not change with a change of the problem size. In this case equation (14) is exactly Amdahl's law.

The second case where $\gamma = p$ is where Gustafson's law can be applied.

Gustafson's Law

In case of $\gamma = p$ we can simplify equation (14) by replacing γ with p. This results in Gustafson's law [57, p. 34], which is:

$$\text{Scaled Speed-up} \leq \frac{\mathrm{f} + \mathrm{p} \cdot (1-\mathrm{f})}{\mathrm{f} + \frac{\mathrm{p} \cdot (1-\mathrm{f})}{\mathrm{p}}} = \mathrm{f} + \mathrm{p} \cdot (1-\mathrm{f}) \tag{15}$$

$$\text{Efficiency} \leq \frac{f + p \cdot (1-f)}{p} = \frac{f}{p} + (1-f) \tag{16}$$

Limitations of Gustafson's Law

- Gustafson's law only gives an upper-bound for scaled speed-up. It will not give an approximation to the true scaled speed-up of a program. This is the case because Gustafson's law does not account for parallel overhead.
- Gustafson's law only holds true if $\gamma = p$.
- Gustafson's law can only be applied to applications where the problem size can be increased. Further, it does not make sense to apply Gustafson's law if the end user of the parallelized program has no interest in an increased problem size.

4.2.7 Calculating the Serial Fraction of a Program

To apply Amdahl's law or Gustafson's law it is necessary to know the serial fraction f of a program. In the notebooks we discuss two possible methods to get a value for f:

1. Measuring the wall-time of different program parts
2. Analysing the speed-up progression with strong scaling

One possible method to get f is by measuring the wall-time of all the program parts which are non-parallelizable, and the wall-time of the parts which can be parallelized. This gives the measurements needed to calculate an approximation to the serial fraction of the program. The challenge with this approach is to identify all the serial parts hidden within the parallel regions. It can be difficult and time consuming to find all the code blocks which belong to the inherently serial part of the program.

An alternative approach for approximating f is given with Amdahl's law. We know from Amdahl's law that if we keep the problem size fixed, and we measure the run-time with an increasing number of parallel processing units, then the maximum speed-up will converge towards $\frac{1}{f}$ for an unlimited number of parallel processing units. After calculating the speed-up progression of a parallelized program with an increasing number of threads, it is sometimes possible to see what value the calculated speed-ups are converging to. This value can be interpreted as an approximate number for $\frac{1}{f}$. The serial fraction can then be calculated with $f = \frac{1}{1/f}$.

Instead of doing the analysis visually, it is also possible to fit a graph to the set of data points, for example by using the 'least square regression method' [61]. The limit of the fitted graph for an infinite number of p would then be the approximate value for $\frac{1}{f}$. The issue with this approach is that the data points of the calculated speed-ups will not always resemble a curve in the shape how it is assumed by Amdahl's law (see Figure 10). Thus, the fitted graph might not converge to an upper limit. Reasons for this are random noise in the time-measurements and the fact that true speed-up is not equal to the upper-bound of speed-up. Due to parallel overhead the true speed-up is often much smaller than the upper-bound for speed-up calculated with Amdahl's or Gustafson's law.

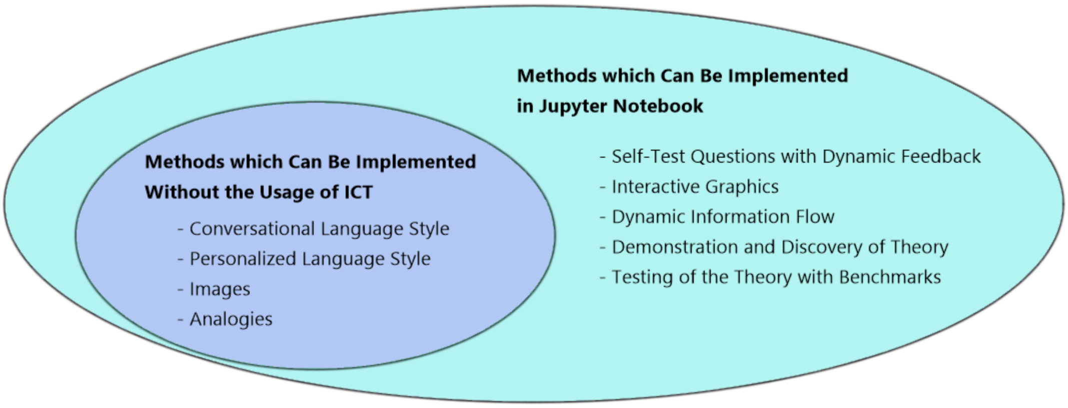

Methods which Can Be Implemented in Jupyter Notebook
- Self-Test Questions with Dynamic Feedback
- Interactive Graphics
- Dynamic Information Flow
- Demonstration and Discovery of Theory
- Testing of the Theory with Benchmarks
Methods which Can Be Implemented Without the Usage of ICT
- Conversational Language Style
- Personalized Language Style
- Images
- Analogies

FIGURE 6: VENN DIAGRAM OF APPLIED PEDAGOGICAL METHODS

4.3 Pedagogical Principles

In the previous section we presented the theory without the use of any pedagogical methods. This is not suitable for teaching since the students' benefit from the instructions would not be maximized. To fully leverage a student's potential for learning it is necessary to apply pedagogical principles. Educational science is a young research field, which is why many of the traditional as well as contemporary teaching methods are not extensively researched yet. As of today, there is no consensus amongst professionals in terms of which pedagogical principles should be used in teaching or not. This makes it more challenging to choose a suitable combination of pedagogical methods.

The pedagogical methods which we applied can be divided into two groups:

1. General pedagogical methods which can be implemented without the usage of ICT
2. Pedagogical methods which can only be implemented with an ICT tool such as Jupyter Notebook

Figure 6 gives an overview about the applied methods, and how the two groups of methods relate to each other. The pedagogical methods that can be applied without the usage of ICT can also be implemented in a Jupyter notebook, due to the markdown cells which allow to present content such as text, images, or equations. On the other hand, Jupyter notebooks allows to implement certain pedagogical methods which could not be implemented without the use of ICT. Thus, the two groups of methods can be displayed as a Venn Diagram (see Figure 6).

4.3.1 Pedagogical Methods which do not Depend on ICT

Personalized and Conversational Language Style

Research results suggest that a personalized and conversational writing style results in substantially better learning outcome compared to a formal writing style [5, p. 186], [62]–[64].

Mayer et al. [63], for example, made three experiments where they replaced sentences like *"During exhaling, the diaphragm moves up creating less room for the lungs, air travels through the bronchial tubes and throat to the nose and mouth where it leaves the body"* with a personalized version. In a personalized script, this sentence would be changed to: *"During exhaling, your diaphragm moves up creating less room for your lungs, air travels through your bronchial tubes and throat to your nose and mouth where it leaves your body."* Simply exchanging 'the' with 'your' resulted in better transfer of knowledge with a median effect size of 0.79.

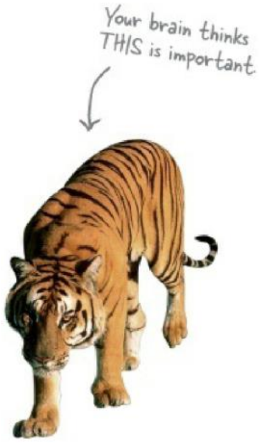


FIGURE 7: KATHY SIERRA'S LEARNING THEORY [14, P. XXIII]

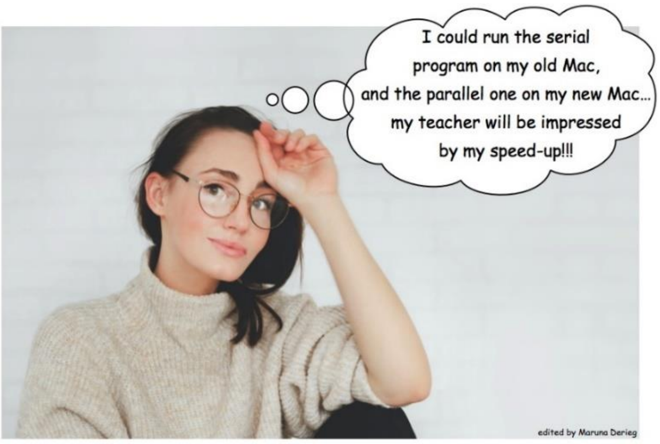

I could run the serial program on my old Mac, and the parallel one on my new Mac... my teacher will be impressed by my speed-up!!!
edited by Maruna Derieg

FIGURE 8: EXAMPLE OF A SPEECH-BUBBLE PICTURE USED IN THE NOTEBOOK

In the textual instructions of the notebooks, we applied a personalized language style, directly addressing the students and choosing active voice over passive voice. Further, we applied a conversational writing style that guides the reader through the instructions by asking questions. Instead of writing *"In the next section we will discuss how the serial fraction of a program can be identified"* we wrote *"Have you wondered how you would be able to find out what fraction of the program is inherently serial?"*. Also, other casual comments such as *"Let's find out how we could do this*" were included in the instructions, which gives the text the appearance of a narrative. Moreno and Mayer [64] who taught students about the formation of lightening used comments like: *"Now that your cloud is charged up, let me tell you the rest of the story."* In their research, Moreno and Mayer found that students who were exposed to this language style performed substantially better than the control group.

Images

The use of images in educational instructions has become a standard in modern teaching material. Many research experiments have shown that using meaningful pictures in addition to textual instructions has a positive influence on learning outcome [5, p. 77]. Multiple types of images were used in the notebooks, including graphics displaying results of performance calculations, pictures visualizing analogies, images displaying theoretical concepts, and speech-bubble pictures. An example of a speech-bubble image can be seen in Figure 8. In this section we will only discuss the speech-bubble pictures because this is the only image type that we feel might rise controversies.

One of the reasons why the usage of speech-bubble pictures is not common practice in teaching could be the lack of research. To our knowledge, there is currently no research about the effect of speech-bubble pictures on learning. Thus, we are left with assumptions and research results that only indirectly relate to the topic.

The idea to use speech-bubble pictures in our notebooks is inspired by the teaching style of Kathy Sierra, who has created the award-winning book series 'Head First'. All the books in this series teach about programming. Sierra has developed a unique style to teaching which applies many pedagogical principles, one of which is the use of speech-bubble pictures [65]. Though Sierra claims that her approach to teaching is based on research results, but she does not mention sources. Sierra's theory is that certain images can trigger biochemical reactions in the brain which will facilitate learning. Sierra gives the example of a tiger. If humans are confronted with a life-threatening situation, such as a tiger, the experience will be retained in the brain much longer than other situations (see Figure 7). According

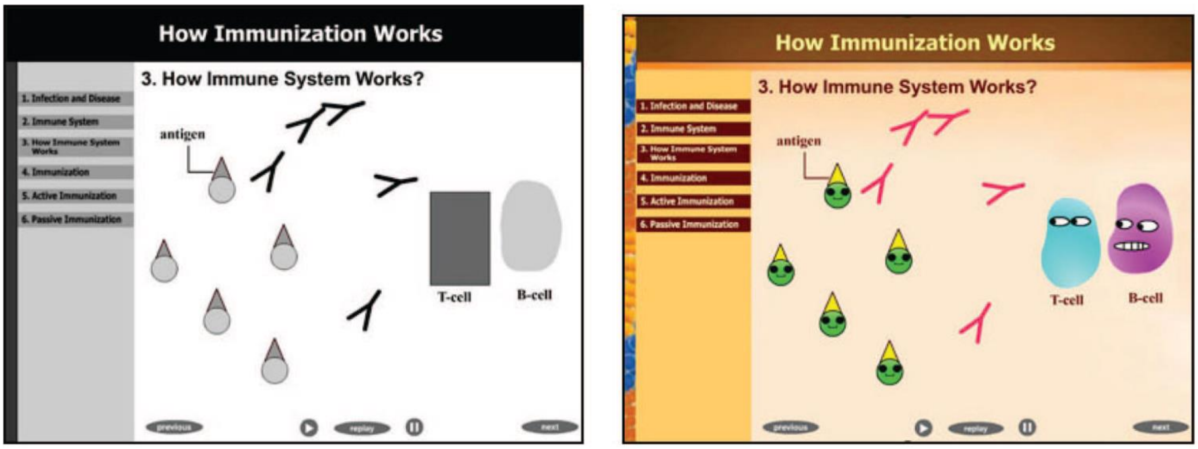


FIGURE 9: EXAMPLE IMAGE BEFORE AND AFTER AN EMOTIONAL REDESIGN [67]

to Sierras theory, interleaving textual instructions with images that the brain classifies as important will have a positive effect on retainability [14, p. XXIII].

We were not able to find research that confirms Sierra's theory but found a few other studies that are related to the topic. A research by Nittono et al. [68] could show that looking at cute animal pictures before performing a task increased attentional focus. This suggests that images, even when completely unrelated to the task, can influence a human's performance. Whether or not an image has a positive influence on learning seems to depend on the content the image displays. Research results indicate that including unrelated images in textual instructions can have a negative influence on learning outcome, regardless of whether the images are perceived as interesting [5, p. 164], [69], [70]. On the other hand, it could be shown that images following an 'emotional design' positively influence learning [5, p. 167]. Um et al. [67] conducted a study where they redesigned images about immunology with the intention to raise emotions in the learner. Figure 9 shows an example of how the researchers applied 'emotional design' principles. The results of the study suggest that applying emotional design principles leads to better performance in terms of transfer and comprehension of knowledge.

Using speech-bubble pictures allows to convey information in a different manner, thus allowing to look at a topic from a different angle. An example for this is given with Figure 8. Instead of explaining to students that it is important to perform run-time measurements of serial and parallelized programs in same conditions, this message can also be conveyed in a speech-bubble picture (see Figure 8). What might be inappropriate to mention in textual instructions can sometimes more easily be communicated through a medium such as a speech-bubble picture. Another advantage of speech-bubble pictures is that they allow to combine theory with humor. Research results suggest that humor has a positive influence on learning [71]–[74]. The difficulty with humor is that it is highly subjective. Different people find different things funny. Also, teachers might fear that the students will not appreciate their humor or that they are perceived as less professional if they are funny.

Analogies

Research has shown that analogies are a useful tool to facilitate students understanding of a concept [75]. Whether a suitable analogy can be found depends on the topic to be taught. With respect to Amdahl's law there are many possible analogies that could be used to exemplify the general concept.

It was decided to first discuss the principles of Amdahl's law with a practical example, before presenting any mathematical formulas. Understanding the concept behind Amdahl's does not require advanced mathematical skills if a suitable analogy is used to explain it. A student that already understands the core principles of Amdahl's law will find it easier to also grasp the meaning behind the mathematical formulas that Amdahl's law contains.

To explain Amdahl's law without the use of mathematical formulas we created a story of a project which allows to apply parallelism. The imaginary project aims at building a beach at home, for which sand is to be taken from the beach, loaded on a truck, driven home, and then unloaded to build the new beach at home. There are three phases in this project: loading the truck, driving the truck home, and unloading the truck. For each phase the students are told how long it takes to complete the task if no one is helping. It follows a series of self-test questions and instructions, which guide the students step by step through the parallelization of the project, and the effect on performance.

Using an analogy makes it possible to teach the basics of Amdahl's law without the necessity to understand any of the mathematical formulas. Research results suggest that this approach increases the students' comprehension of the topic [75].

4.3.2 Pedagogical Methods which Depend on ICT

Self-Test Question with Dynamic Feedback

Prompting students to answer questions will keep them more engaged with the topic and cause them to reflect on the learned material [12, p. 231]. In total, the notebooks contain nineteen self-test questions. Many instructional textbooks in print format contain self-test questions as well, but the answers to these questions will be static and usually located at the end of the chapter or book. In Jupyter Notebook it is possible to implement self-test questions which return immediate feedback in dependence of the provided answer. Thus, the feedback to the given answer can be considered dynamic. The advantage of dynamic feedback is that the students do not see the correct answer after submitting a wrong one. Instead, the students are given a hint or some encouragement to try again. This increases the chances that the students will eventually find the right solution on their own, which is arguably an important factor for motivation. Further, it is more convenient for the user to immediately receive feedback without having to change the current reading position.

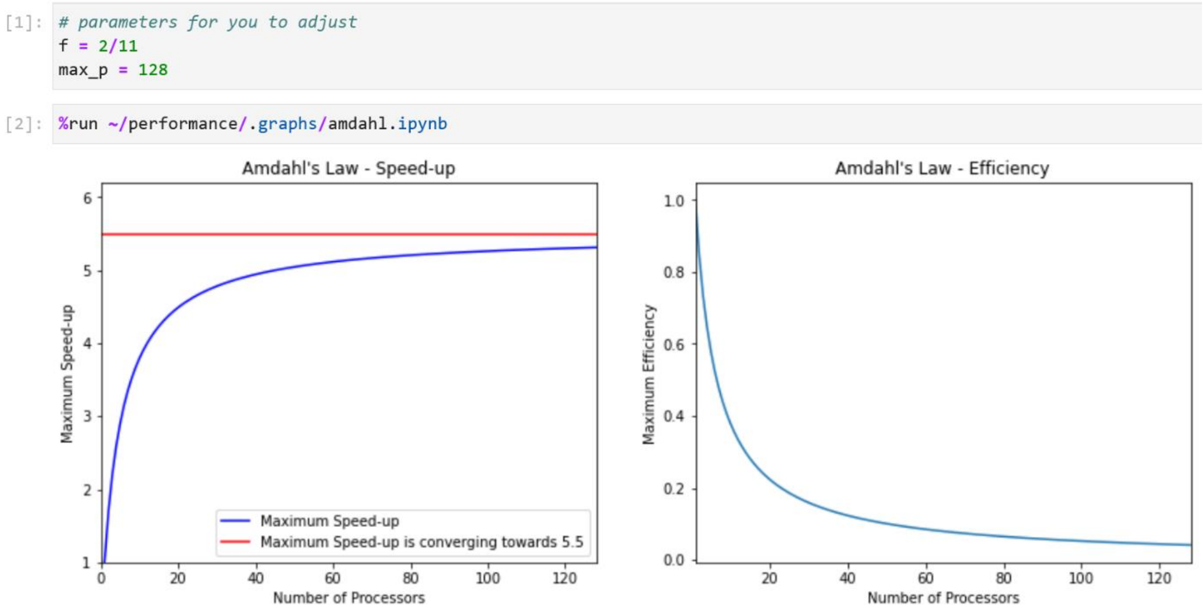
The disadvantage with dynamic feedback is that the questions must be formulated in a way that allow to categorize the possible answers. This limits the type of questions that can be asked.

Interactive Graphics

One of the advantages of computational notebooks is that graphics do not need to be static. Figure 10 shows an interactive graphic, which displays the upper-bound of speed-up in dependence of the serial fraction of the program. In a static graphic, the author needs to decide one or multiple serial fractions, for which the maximum possible speed-up should be displayed. Also, the author needs to set some other parameters, such as the highest number of parallel processing units. In an interactive graphic, it can be left to the user to set a value for these parameters. Thus, the user can change the graphic according to personal interests. This provides a basis for active learning, which is defined as *"anything that involves students in doing things and thinking about the things they are doing"* [76, p. 19].

Dynamic Information Flow

In Jupyter Notebook knowledge can be presented in a tree like structure. In traditional teaching media, knowledge is introduced in a linear order, one topic following the other. The markdown cells in Jupyter Notebook make it possible to include links, which open additional notebooks in a new tab when clicked upon. Instead of presenting all sections in a linear order, it is possible to fork to additional notebooks



```
[1]: # parameters for you to adjust
     f = 2/11
     max_p = 128
```

```
[2]: %run ~/performance/.graphs/amdahl.ipynb
```

Figure 10: Interactive Graphic about Amdahl's Law

wherever appropriate. Like this the main notebook can remain much more compact. The dynamic information flow also enables a more dynamic style of knowledge presentation, allowing to address students with diverse educational backgrounds and interests. In case a topic can only be understood with previous knowledge about how run-time is measured, for example, then a link can be provided which opens another notebook explaining these basics. Students who already know about this topic can simply ignore the link and keep reading. Also, topics which might only interest a subgroup of students can be placed in additional notebooks.

Demonstration and Discovery of Theory with Code Sections

The fact that textual instructions can be interleaved with code cells in Jupyter Notebook allows to directly demonstrate certain aspects of the theory. The students do not need to take the word of the teacher for granted but can convince themselves of certain concepts by running prewritten code cells.

An example of how we applied this approach is given with the introduction to overhead. Before introducing the students into the topic of overhead it was demonstrated with a code cell that the parallelized quicksort program runs slower than the serial quicksort program, in case the list to be sorted consists of 2'500 random numbers. Discovering that parallelization does not always increase performance will lead to the question of why this is the case. In a next step the students are presented with the theory that explains the reasons for the observed behavior.

Another possible usage scenario of code sections is that the instructor can create the code cells in a way that allows the student to discover the theory prior to reading about it. Schwartz and Brandford [77] conducted a study which could show that discovering a certain phenomenon before being taught about it results in better transfer of knowledge. In that experiment the students participated in two separate exercises, first they tried to discover certain phenomena, and then they attended a lecture which taught the theory related to the discoveries. Jupyter Notebook makes it possible to combine these two phases. Students can be guided to try out certain exercises in code cells which might be prewritten or partially prewritten. These code cells can immediately be followed by textual instructions that explain to the student the theory behind the made observations.

Testing of the Theory with Benchmarks in Additional Notebooks

One of the goals of this project was to create an environment where the students not only read about the topic of performance but are also able to test the learned theory. With Jupyter Notebook the performance of programs can be evaluated in the same environment as the theory is taught. This is convenient for the students because they do not have to switch between different ICT tools. Another benefit is that the time gap between learning a concept and applying or testing it can be minimized. At most universities it is custom to have special lab-sessions in addition to lectures. Thus, theory and practice are kept separate, and there is a time gap between learning and applying. Research has shown that newly acquired knowledge is only retained for a short amount of time [78]. Arguably, it makes more sense for students to apply knowledge when the memory about the new information is still fresh. Jupyter Notebook provides an environment where instructors can create learning materials that interleave theory with practice, thus minimizing the time gap between learning and applying.

The students are given the possibility to measure and calculate the performance of various benchmarks that were specifically written for this purpose. When programs are analyzed for performance, the results are often not as expected. It is crucial that students are aware of this, and that they understand why certain programs perform in a certain way. Testing different programs for performance will help students to get a more profound understanding of the theory. Further, it can help students to identify knowledge gaps, which can then be discussed with professors or fellow students.

4.4 Implementation

4.4.1 Images

In Jupyter Notebook two different methods for displaying images are possible. First, images can be integrated in a markdown cell with the use of a HTML image tag. Second, images can be the output of a code cell. In the notebooks there are many images which display the speed-up or efficiency in dependence of the number of parallel processing units or problem size. These graphics can be created at the time the code cell is run, meaning that the run-time measurements, performance calculations and output of the graphic are only calculated at the time when the user runs the code cell. Thus, the resulting image can be considered dynamic since it will look different with every run of the code cell.

Instead of dynamic graphics based on code cells we mostly used static pictures in the main Jupyter notebook. This has the advantage that the reader does not have to wait for the time measurements to be completed. Further, it is easier to write textual instructions referring to a static picture, since the exact performance results are known at the time of writing. This is not the case with dynamic graphics, where the time measurements and calculations are performed after the textual instructions are already written. Wherever a static picture displaying performance is presented, the student is also given a link which opens an additional notebook containing the code used to calculate the static picture. Thus, the graphics can easily be reproduced and adjusted by the user.

Static images have been integrated into the notebooks directly inside the markdown cells with the `<img>` HTML tag:

```
<img src='figures/myPicture.png' width='400' art='Description of Picture'>
```

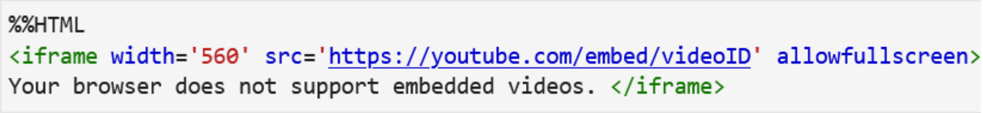
In Jupyter notebooks pictures in markdown cells cannot be accessed from hidden directories or files. Thus, the folder `'figures'` which contains all the pictures must be a visible directory.

4.4.2 Videos

Except for a very short video in the introduction about the project, no videos were used. Originally it was planned to include an eight-minute-long YouTube video[2] by Tim Mattson which introduces the audience into the relevance of performance and parallel programming.

We will briefly discuss two methods how videos can be integrated in Jupyter Notebook and why we have decided against it.

A video can be published on a platform such as YouTube or Vimeo which allows it to be embedded in a Jupyter Notebook code cell with the `%%HTML` magic command and the `<iframe>` tag:

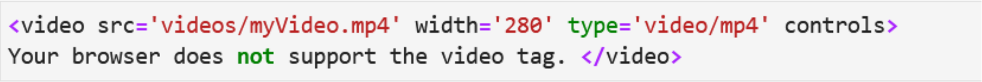


```
%%HTML
<iframe width='560' src='https://youtube.com/embed/videoID' allowfullscreen>
Your browser does not support embedded videos. </iframe>
```

For security reasons, the `<iframe>` tag is not supported in markdown cells in Jupyter Notebook. Thus, the code for an embedded video must be placed in a separate code cell. The `%%HTML` magic command ensures that the code in the code cell is interpreted as HTML.

The issue with embedded videos is that a video can get deleted from the publisher and that the publisher does not always allow the video to be embedded. In case of the video by Tim Mattson the later was the case.

To get around this problem, the video was downloaded from YouTube and saved locally. This allows to integrate the video with the `<video>` tag in a markdown cell:



```
<video src='videos/myVideo.mp4' width='280' type='video/mp4' controls>
Your browser does not support the video tag. </video>
```

This solution worked but there are two concerns with this approach. First, we have not obtained permission by the publisher to use the video in this way, and second, we noticed that the JupyterHub account takes much longer to load at login if videos are saved on the JupyterHub server. When a video is embedded, it is streamed once the play button is clicked. In case the video is saved on the JupyterHub server the video is loaded at login, which results in long wait times for the user when the personal account is loaded. For these reasons we decided against the usage of videos in the developed Jupyter notebooks. Instead, we provided a link to the video by Tim Mattson, allowing interested students to watch it on YouTube.

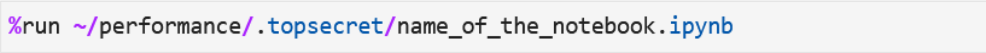
We conclude that in case JupyterHub is used, the most optimal way for integrating videos is by hosting them on an external platform such as YouTube or Vimeo. The videos can then be embedded into the notebook with code cells using the `%%HTML` magic command. This will ensure that the JupyterHub account does not take additional time to be loaded at login. Further, it is recommended to self-create the videos, which will avoid issues with usage rights and solve the problem that videos by other publishers can get deleted.

[2] https://youtu.be/cMWGeJyrc9w (accessed on 5th June,2021)

4.4.3 Self-Test Questions

For each self-test question a separate notebook consisting of a single code cell was written. The code, if run, will first collect a user input and then print a response in dependence of the given input. In case the provided answer was correct, the output message will be something like *"well done"*, sometimes followed by an explanation of why the given answer was correct. Otherwise, the output message will state that the answer is wrong, followed by an encouragement to try again. Sometimes a hint is given if the answer was not correct.

To maximize the students' benefit of these self-test questions, the code needs to be hidden from the user as otherwise it would be possible to get the correct answer by analyzing the code content. To achieve this, the notebook containing the code of the self-test question is saved in a hidden directory named `'.topsecret'`. This notebook can then be run from within the main notebook which is saved in the `'performance'` directory, with the command:



```
%run ~/performance/.topsecret/name_of_the_notebook.ipynb
```

Running this command will run all code cells of the notebook located at the specified path and display the output in the notebook from where the command is run. This approach allows to collect user input and to provide feedback in dependence of the input all from within the main notebook, without the user being exposed to the code located in the `'.topsecret'` directory. The code in the external notebook will be run with the kernel session that has been started in the main notebook. Thus, the code in the external notebook needs to be written in the same programming language as is used in the main notebook.

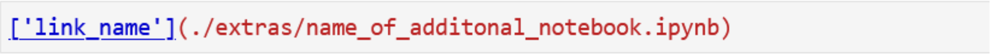
A skilled user will be able to get access to the code which contains the answers to the questions by making the `'.topsecret'` directory visible. With self-test questions this should not give any reasons for concerns since the submitted answers are ungraded.

4.4.4 Interactive Graphics

Interactive graphics are integrated into the notebook in a similar way as are the self-test questions. The code which calculates the graphics is located in a separate notebook, which is saved in a hidden directory named `'.graphs'`. This has the advantage that the reader is not distracted by the code which calculates the graphics. In case of Amdahl's law, the parameters which the user can adjust are the serial fraction of a program f, and the maximum number of processing units for which the upper-bound of speed-up should be calculated. These values can be set by the user in the main notebook simply by assigning a value to these variables (see Figure 10). In the next step, the external notebook which contains the code for the calculations is run from within the main notebook with the `%run` command. This results in the graphic being displayed in the main notebook. This approach works because the variables which are initialized within the main notebook are visible by the external notebook `'amdahl.ipynb'` due to the shared kernel.

4.4.5 Dynamic Information Flow

The following expression, when placed somewhere in a markdown cell, will display a link to another notebook:



```
['link_name'](./extras/name_of_additonal_notebook.ipynb)
```

In our set of notebooks, we placed the following topics in additional notebooks.

- How to measure the execution time of a program in Jupyter Notebook
- Performance comparison with the two presented methods for measuring run-time
- Two different approaches to calculate the run-time of the serial program
- The threshold from when onwards the benefit of parallelization outweighs overhead
- Calculating the graphics which display speed-up and efficiency progressions for strong and weak scaling
- Analyzes of how problem size influences performance

Since the topic of performance in parallel programming contains a lot of potential for exploration, there are quite a few additional notebooks which analyze certain subtopics more in detail. It was decided to place most explorative content in separate notebooks, as to keep the presentation of the theoretical foundations more compact.

4.4.6 Testing of the Theory with Benchmarks

Possibly the biggest challenge of the project was to create reusable code that allows to evaluate the performance of different programs. The goal was to write code that can be used to analyze the performance of any program, without having to do major adjustments to the code.

The performance of a program can be visualized with scatterplot diagrams that display run-time, speed-up or efficiency. All the code for these graphics was written in Python, using the library Matplotlib. The metrics for this graphics, however, can only be calculated if the run-time of the serial and parallelized program is known.

Measuring Run-Time

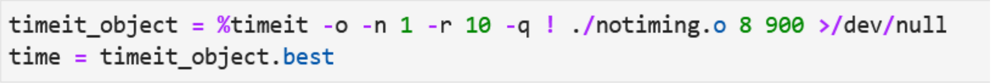
The first step when calculating speed-up and efficiency is to measure the run-time of the serial and parallelized program version. We identified two methods for measuring the execution time of programs:

1. With the magic command `%timeit` from within the Jupyter Notebook environment
2. From within the program, with whatever method is suitable in the used programming language

In both options, the UNIX terminal is accessed from within a code cell in the Jupyter notebook. This can be done by placing an exclamation mark in front of a bash command. Thus, the serial and parallelized program can be run from within a code cell in Jupyter Notebook, by executing a bash command. The difference between the two methods for measuring run-time lies at the start and end point of the time measurement.

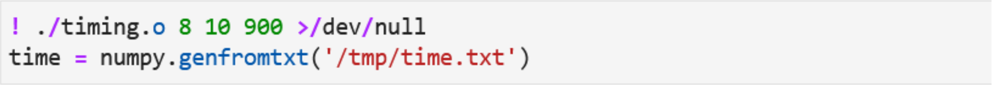
In the first option, the time measurement starts right before the program is run, and it ends after the program has terminated. In this method there is a large overhead in the measurement, since the time needed to load the program and its libraries is included in the measurement. The second approach measures the execution time from within the program itself and saves it into a file located in the `/tmp` folder. Thus, when the program is run from within the Jupyter Notebook code cell, the program writes its own run-time into a text file. In a next step the current value saved in this text file can be read and saved in a Python variable. The second method gives the developer more control over which sections in the program should be included in the time measurement, and there is less overhead since the time measurement starts after the program is already loaded and ends before the program terminates.

The following code examples demonstrate how run-time can be measured with the two options. When measuring the execution time with the magic command `%timeit` the option `-o` allows to save the result as an object. Further, the options `-r` and `-n` define the number of repeats and loops per repeat respectively. In the following code examples the thread number is set to 8, the problem size to 900, the number of repeats to 10, and the option `>/dev/null` is used to repress the output.



```
timeit_object = %timeit -o -n 1 -r 10 -q ! ./notiming.o 8 900 >/dev/null
time = timeit_object.best
```

In case we run a program which saves its own run-time into a text file, the code could be like this:



```
! ./timing.o 8 10 900 >/dev/null
time = numpy.genfromtxt('/tmp/time.txt')
```

We decided to use both measuring methods in the notebooks for two reasons:

- In case students want to test their own programs for performance, they can do this more easily with the first method since less code adjustments are needed.
- Using both methods allows to demonstrate how overhead influences speed-up and efficiency. We discussed this topic in two of the additional notebooks.

No Shared Variables Between the Python Kernel and the Terminal

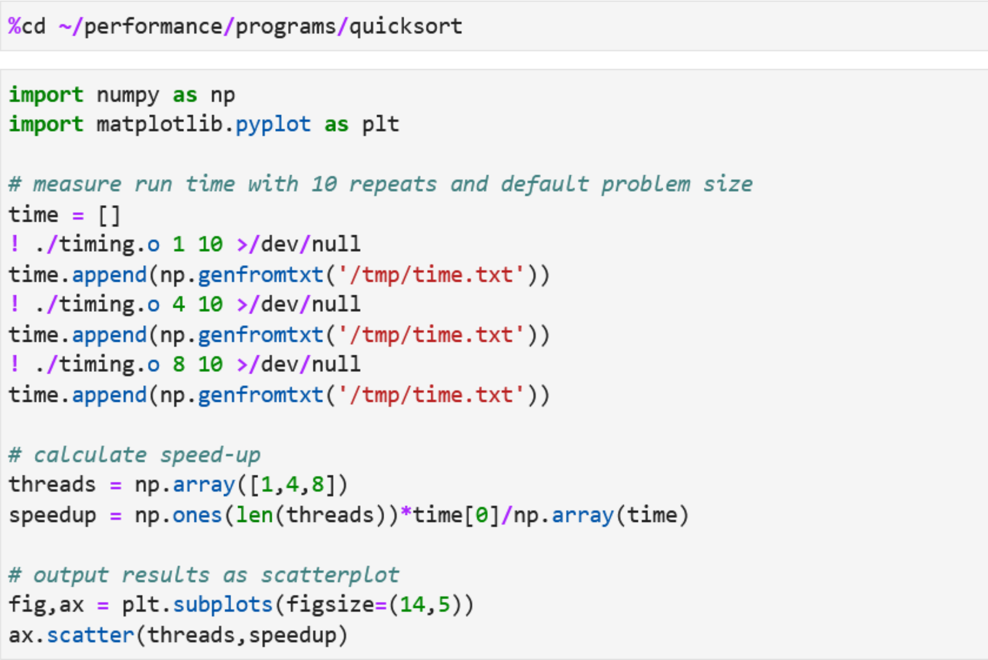
Some of the basic principles of code writing could not be applied, because the terminal and the Python kernel do not have shared variables. Ideally, a developer would implement a for-loop where in each round the run-time of the program is measured with a different number of threads. This does not work in our usage scenario because it is not possible to define Python variables and then pass their value to a terminal command that is called from within Jupyter Notebook. Unfortunately, because of this, there is a line of code for each time measurement. This makes it more time consuming for the user to change the parameters of the time measurements.

Reusable Code

It would not be convenient to have a separate notebook for each benchmark to be analyzed. We came up with a solution that allows to reuse the same notebook for any program, as long as the program follows certain requirements. Which ones these are will be discussed in the next section.

The idea is to use relative path names when running the program with a bash command from within the notebook. Further, each compiled program needs to be given the same name. In case the program is measuring its own run-time internally the binary needs to be named `'timing.o'`, and otherwise `'notiming.o'`. The only way to keep the binaries of the different benchmarks apart from each other is by placing them in separate subdirectories. Following this naming convention and using relative path names when calling the programs allows to reuse the code of the performance calculations with any program. To do this, the user only needs to change the present working directory to the one where the program of interest is located at.

The following code gives an example:



```
%cd ~/performance/programs/quicksort
```

```
import numpy as np
import matplotlib.pyplot as plt

# measure run time with 10 repeats and default problem size
time = []
! ./timing.o 1 10 >/dev/null
time.append(np.genfromtxt('/tmp/time.txt'))
! ./timing.o 4 10 >/dev/null
time.append(np.genfromtxt('/tmp/time.txt'))
! ./timing.o 8 10 >/dev/null
time.append(np.genfromtxt('/tmp/time.txt'))

# calculate speed-up
threads = np.array([1,4,8])
speedup = np.ones(len(threads))*time[0]/np.array(time)

# output results as scatterplot
fig,ax = plt.subplots(figsize=(14,5))
ax.scatter(threads,speedup)
```

This code will first change to the directory of interest, and then measure the run-time of the `'timing.o'` binary with the number of threads set to one, four and eight. Next, the speed-up for each number of threads is calculated and displayed in a scatterplot. The students do not need to understand the Python code in the second code cell. All they need to do is change the present working directory with the first code cell and run the second code cell.

Testing a New Program for Performance

The developed additional notebooks allow to test any program for performance which can be run on the terminal, regardless of the programming language used. This is very convenient as it makes the notebooks usable for a more diverse audience. It does not matter whether a student implements concurrency with multithreading, openMP or some other technique; as long as the program follows certain requirements the notebooks will calculate and display the performance of that program.

To make this work, the user will have to adjust the code so that the program accepts the number of threads as the first argument and the problem size as the second. Also, the compiled program must be named `'notiming.o'` and there needs to be a default value for the number of threads and the problem size.

Little more adjustments are needed if the user wishes to measure the run-time only on a certain part of the program. In this case, the user also needs to extend the program so that it measures its own execution time and saves it to a text file named `'time.txt'` in the `'/tmp'` folder. Further, it is necessary to implement multiple repeats of the time measurement, as to reduce random noise in the measurement. The program needs to accept the thread number as the first argument, the number of repeats as the second argument, and the problem size as the third argument. The compiled program must be named `'timing.o'`.

In case the compiled program satisfies these criteria, all that is left to do for the user is to navigate to the directory where the program is saved at. This needs to be done before any of the other code cells are run. No adjustments in the Python code are necessary to calculate and output the graphics which visualize the performance of the program. Thus, a student does not need to understand Python to be able to evaluate the performance of a program. However, the student must be able to adjust the program in the way we have just discussed.

Benchmarks

When selecting the benchmarks we focused on algorithms which the students have likely been exposed to in the past. It makes little sense to compare the performance of programs that the students don't understand. For students to be able to interpret why different programs perform in a certain way it is necessary to have some knowledge about the used algorithms. In most undergraduate programs of computer science students take a compulsory course in algorithms and data structures. It can be expected that the learners know about sorting algorithms such as Quicksort and Mergesort. Further, it is likely that the students will read the notebooks in the context of a course about parallel programming. We have mostly chosen algorithms that are often used to teach students about parallel programing, thus increasing the likelihood that the user has already been exposed to the used algorithms.

In weak scaling the problem size of a program is increased simultaneously with an increase of the number of parallel processing units. It is therefore crucial that the used benchmarks can be run not only with different numbers of threads but also with different problem sizes. Only if this is guaranteed can the benchmarks be used in weak scaling analyzes.

In total, ten programs have been provided to the students for performance testing. These are:

- Four sorting algorithms (Quicksort, Mergesort, Bubblesort and Selectionsort)
- Calculation of Pi
- Calculation of the Fibonacci series
- Finding all prime numbers lying in a given interval
- Calculation of the Jacobi matrix
- Calculation of the Mandelbrot set
- Matrix-matrix multiplication

All the benchmarks are written in C and parallelized with openMP. Since we are not experts in parallel programming, we mostly relied on code that has already been parallelized by other computer scientists. The original source is mentioned in the comments of each program. It was necessary to adjust all programs so that the number of threads and problem size can be passed as parameters. For each benchmark we created two versions, one that does not measure run-time internally and one that does. With the later, also the internal measurement of run-time with multiple repeats had to be implemented. Further, it was necessary to allocate memory dynamically to avoid a stack overflow in the case of large problem sizes.

5 Evaluation

New teaching material is usually developed in multiple cycles, during which the current version of the teaching material is reevaluated and updated accordingly. An evaluation of the teaching material will give the insights needed to make meaningful adjustments. Thus, an evaluation is crucial for quality assurance. In case new pedagogical methods are applied, it is also recommended to research the influence of these methods on learning outcome. Since this would be beyond the scope of this thesis, we limited the evaluation to a self-report of the students' learning experience and their opinion about the developed notebooks.

Instructors and students usually have a different level of education and expertise. Therefore, it can be difficult for instructors to understand how students perceive the learning material. An instructor cannot know the students' experience with the learning material unless some form of feedback takes place. An evaluation of the students' learning experience and opinions will give the necessary insights to further improve the learning material.

A first draft of the developed Jupyter Notebook has been made available to students interested in the topic. The participants' learning experience and opinion about the teaching material was evaluated with a semi structured questionnaire. In this chapter we will first define the research questions and then present the methodology and the results. The chapter ends with a short discussion about factors which might have impacted the validity and reliability of the evaluation.

5.1 Research Questions

One of the sub-objectives of this thesis is to evaluate the participants' learning experience and their opinion about the developed Jupyter notebooks, which will help identify areas where the current version of Jupyter notebooks can be further improved.

To reach this sub-objective, the following research questions were taken into consideration:

Question 1: Who are the research participants?

1.1 What educational program are they enrolled in?
1.2 How much previous knowledge about the topic do they have?
1.3 Have they worked with Jupyter Notebook before?

Question 2: What was the learners' behavior when working through the developed notebooks?

2.1 Which sections of the main notebook did they work through?
2.2 Did they look at any of the additional material?
2.3 Did they test any self-written programs for performance?
2.4 How much time did they spend working through the notebooks?

Question 3: Were the learners able to work with the developed Jupyter notebooks?

3.1 How challenging was it for them to work in the Jupyter Notebook environment?
3.2 Did they understand how to submit answers to the self-test questions?
3.3 Did they understand how to work with the code sections?
3.4 Did they understand how to work with the additional notebooks?
3.5 Did they face any technical issues when testing their own programs?

Question 4: What is the learners' opinion about the developed Jupyter notebooks?

4.1 What is their opinion about the applied pedagogical methods with respect to quality and the ability to foster learning?

4.2 What is their teaching medium of choice when learning about performance in parallel programming?
4.3 What did they like and dislike about the notebooks?
4.4 What are their suggestions for future changes on the notebooks?

Question 5: Are there correlations and dependencies amongst the measured variables?

5.2 Methodology

In this section we will present the methods that were applied to answer the research questions of the evaluation. First, the research design is presented, followed by a description of the measuring instrument and the used variables and measurements. Next, we will present the data collection process, the sample and sampling technique, as well as the research methods of the quantitative and qualitative data analysis. The section ends with a summary of how we dealt with ethical concerns.

5.2.1 Research Design

The evaluation of the developed notebooks was conducted using a mixed method approach. A concurrent research design has been applied, collecting quantitative and qualitative data with the same measuring instrument [79, p. 175]. The participants' learning experience and their opinion about the developed notebooks is evaluated with a 'self-report', measuring what the participants experience, think and feel [80, p. 65].

5.2.2 Measurement Instrument

A semi-structured questionnaire was considered as most suitable for the research purpose, choosing a mixture of closed and open-ended questions [81, p. 474]. The questionnaire was designed with the intention to gather a broad range of data while at the same time minimizing the amount of time participants need to spend completing the questionnaire.

The questionnaire starts with a question asking for the participants' email address, in case they wish to be included in the prize drawing. Then follow three questions about the participants background and six questions which collect general information about the experience and behavior when working with the Jupyter notebook. Multiple question types were chosen for this first set of questions, namely five open questions, two multiple-choice questions, and two rating scale questions. The questionnaire as well as the collected answers can be found in the appendix of this thesis.

The participants' opinion about the pedagogical methods was evaluated with multiple-choice questions. Out of a list of statements the participants were asked to choose the ones they agree with. This question type was chosen because it allows to collect a broad range of data from the participants without using too much of their time [81, p. 477]. Compared to the open-ended questions asked at the end of the questionnaire, this more structed question format ensures to collect the opinion of all participants towards a specific topic.

The questionnaire ends with three open-ended questions, allowing students to give more detailed feedback about what they liked and disliked, and to communicate suggestions.

5.2.3 Measurements and Variables

An often-used approach to evaluate the acceptance towards a software product is given with the 'technology acceptance model' (TAM) [2, p. 87]. With this method, the perceived 'ease of use' and 'usefulness' is evaluated, which together gives a measurement for the 'technology acceptance' [82, p. 61].

Variable Group	Variable Name	Variable Description	Level of Data	Question Type
Descriptive	D1	Pursued level of education	Ordinal	Open
	D2	Current educational program	Nominal	Open
	D3	Previous knowledge about performance in parallel programming	Ordinal	Rating scale 1-5
	D4	Previous exposure to Jupyter Notebook	Ordinal	Multiple-choice
General	G1	Perceived difficulty level of working with Jupyter Notebook	Ordinal	Rating scale 1-5
	G2	Completion of additional notebooks	Binary	Open
	G3	Testing of own programs	Binary	Dichotomous
	G4	Preferred medium for learning about the topic	Nominal	Multiple-choice
Sections Read	R1	Total number of sections read	Ratio	Multiple-choice
	R2 - R11	Each variable describes whether the according section was read	Binary	Multiple-choice
Time	T1	Time spent working on the main notebook	Ratio	Open
	T2	Time spent to work on the extra notebook	Ratio	Open
Pedagogical Methods	Q1 - Q10	Each variable represents a statement about the self-test questions	Binary	Multiple-choice
	S1 - S10	Each variable represents a statement about the speech-bubble pictures	Binary	Multiple-choice
	C1 - C9	Each variable represents a statement about the code sections	Binary	Multiple-choice
	E1 - E9	Each variable represents a statement about the extra notebooks	Binary	Multiple-choice

TABLE 2: OVERVIEW ABOUT THE VARIABLES

In this evaluation we do not only evaluate Jupyter Notebook as a software tool in general, but also the specific notebooks we have developed. Evaluating the project with the TAM approach alone would not consider students' opinion about applied pedagogical methods and other project specific dimensions. We have therefore defined our own variables, which partly overlap with the measurements of the TAM approach. Table 2 gives an overview about all the variables measured and their properties.

Out of all the applied pedagogical methods discussed in section 4.3 the following were chosen to be included in the evaluation:

- Speech-Bubble Pictures
- Integrated Self-Test Questions
- Integrated Code Sections
- Additional Notebooks for Further Exploration

There are multiple reasons why exactly these four pedagogical methods were selected. First, a selection had to be done since it would have been too time consuming for the participants to evaluate all applied pedagogical methods. The speech-bubble pictures are of special interest since we could not identify research done in the past about this approach. The other three methods were chosen because they depend on a medium such as Jupyter Notebook for their implementation.

For each of these four pedagogical methods we evaluated the participants' opinion about the method's influence on learning outcome. To measure the dimension 'ability to foster learning' we defined the following five subdimensions:

- Influence on Retainability

- Influence on Understandability
- Influence on Engagement
- Ability to Provide a Different Perspective
- Risk for Distraction

To measure the participants opinion about these subdimensions, they were presented with a list of statements which each represent one subdimension. The participants were asked to mark all statements they agree with. This resulted in a binary variable for each statement. It can be assumed that participants only marked a statement if they fully or mostly agree with it, leaving it blank if they feel neutral towards that statement or if they disagree. Thus, the value that the binary variables take cannot be interpreted as 'agree' or 'disagree', but rather as 'agree' and 'everything else'.

In addition to the opinion about the influence on learning outcome, we also measured the participants' opinion about quality, ease of use and usage in general. These dimensions were only measured where applicable. Table 9 in the attachment of this thesis gives a detailed overview about each dimension and subdimensions that the different statements represent.

An alternative approach could have been to use a Likert scale for each statement, which would have allowed to measure a more precise degree of agreement. We decided against a Likert scale because it takes much more time for a participant to carefully estimate a degree of agreement for each statement. Considering the research background, we propose that the evaluation is more insightful if a higher number of dimensions is measured with multiple-choice questions, instead of collecting less data but with a Likert scale.

In Table 2 only the variables associated with quantitative data are listed. In addition, also qualitative word-based data was collected. The qualitative data contains information about:

- Problems encountered when testing own programs
- What the participants liked and disliked about the notebooks
- Suggestions about future changes on the notebooks

5.2.4 Data Collection Process

The biggest challenge in the data collection process was to find students willing to spend time reading and evaluating the notebook. An email invitation to work through the notebook was sent twice to all students participating in the undergraduate course 'Computer Systems' and the graduate course 'Advanced Parallel Programming' during Spring Semester 2021 at FernUniversität Hagen. In addition, an announcement has been posted in three different channels on the Discord chat room, a platform where students from the faculty of mathematics and computer science discus study related topics. Anyone interested in the notebooks was requested to write an email to the supervisor of this thesis who in turn created an account for the specific student on the JupyterHub. In total, 54 accounts were created, but only 28 students have logged into their account at least once. A few participants were also given access to the notebooks through a GitHub link, which allowed them to download the directory and work through the notebooks locally.

The Jupyter notebook contains a link which opens a questionnaire that was created and managed with Google Forms. In this way everyone who read the notebook automatically had access to the questionnaire. To give an incentive for completing the notebooks and the questionnaire, a prize drawing amongst the participants was announced. Everyone who completed the questionnaire and left an email-address had the chance to win one of six prizes. In total, 28 participants completed the questionnaire between the 26th of April and 15th of June.

The rate of return has been much larger than originally suspected. An exact number cannot be given, since it is impossible to say how many students read the announcements about the project. Considering only the students who have at least once logged into their JupyterHub account, the rate of return is higher than 82% if we assume that five or less participants worked through the notebooks locally.

5.2.5 Sample

The sample consists of 28 research participants. Due to initial low expectations with regards to the rate of return a 'convenience sampling technique' was applied [83, p. 145]. The Jupyter Notebook has been made available to whomever showed interest in it. As a result, the sample is rather diverse and not representative for a specific subgroup of the population.

All participants who provided information about their current study program are studying either computer science and/or mathematics. The pursued level of education ranges from 'Akademiestudium'[3] to 'Ph.D.' Most participants are students at FernUniversität in Hagen which means that they are studying by distance. Thus, the research participants might have a more positive attitude towards e-learning than students at a traditional university would have. Students at FernUnviersität in Hagen also tend to be more diverse with respect to age, compared to traditional universities. The age and gender of the sample are unknown due to anonymity reasons (see section 5.2.7).

5.2.6 Data Analysis

Quantitative Data Analysis

For the quantitative data analysis, we differentiate between categorical and non-categorical data. To understand how the different variables interconnect and possibly influence each other, we performed a correlation analysis on all the variables. The level of significance for all correlation calculations was set to $\alpha=0.05/5\%$.

Depending on the variable type we used different calculation methods. The correlation between non-categorical variables was quantified with the Spearman's Rho correlation coefficient [84, p. 31]. The correlation between all ordinal non-categorical and dichotomous categorical variable pairs was calculated with the Point-Biserial correlation coefficient [85, p. 280]. Further, we calculated the Phi coefficient for all dichotomous variable pairs. Section 5.2.3 gives an overview about all the variables and their data type [86, p. 301].

Qualitative Data Analysis

All text-based data from the open questions was descriptively analyzed, using a qualitative approach. A content analysis was chosen as most suitable to extract information from the data and the analytical system was derived with an inductive approach [87, p. 258]. The first step in this process was an initial immersion into the data where descriptive coding was applied to organize the data into themes. In the next step we counted the number of codes for each theme, sorted the themes in descending order and summarized the opinions of the participants for each theme.

[3] A program for students who do not qualify to study on Bachelor's or Master's level

5.2.7 Ethical Considerations

The ensure that the evaluation is conducted under ethical conditions, the participants have been informed about the purpose of the study and they have given consent to the usage of provided data [80, p. 40]. In addition, special care was taken not to ask questions that could compromise the participants' anonymity.

Purpose of the Study is Known

The first section of the main Jupyter notebook is an introduction about the project, explaining that the notebook was developed as part of a Bachelor thesis project and that an evaluation is part of that project. Thus, the participants have been informed about the purpose of the study, which is to evaluate the participants' opinion about the notebooks.

Consent for the Use of Provided Data

The questionnaire ends with the following statement: *"By submitting your answers, you give your electronic consent that your answers (except for your email address) will be used and evaluated for research purposes"*. With this, the participants have agreed to the usage of the provided data for research purposes when they submitted their answers.

Anonymity

To guarantee the anonymity of the participants, the questionnaire does not collect data that reveal the identity of the participants [81, p. 129]. The only exception to this is a question asking for the email-address, which the participants could choose to answer if they wanted to participate in the prize drawing. The collected email-addresses have been used exclusively for the distribution of the prizes and have been permanently deleted from the data set. Due to the low number of females at the faculty of mathematics and informatics at FernUniversität Hagen, the gender has not been evaluated. Also, there was no question about age, since an unusual high or low age can reveal someone's identity. The only question which could be problematic with respect to anonymity is question 1, which is an open question about the current study program. Participants could choose not to reveal precise information about the current study program; some participants simply stated that they are students at FernUniversität Hagen.

5.3 Results

Since the study follows a mixed-method approach, we have conducted a quantitative as well as a qualitative analysis. In the first part of this section the results from the quantitative analysis will be presented, followed by the qualitative results.

5.3.1 Quantitative Analysis

First, we will present the results that describe the participants, such as their educational background, previous knowledge about the topic and previous experience with Jupyter Notebook. Then follow the results about the participants' behavior while working with Jupyter Notebook, including the total time that was spent reading the notebooks, the number of sections read and the perceived difficulty level of working with Jupyter Notebook. A correlation analysis will evaluate the relationship between these non-categorical variables. Then follow the quantitative results regarding the participants' opinion about the applied pedagogical methods. The quantitative analysis ends with the results about which teaching medium the participants most prefer when learning about performance in parallel programming.

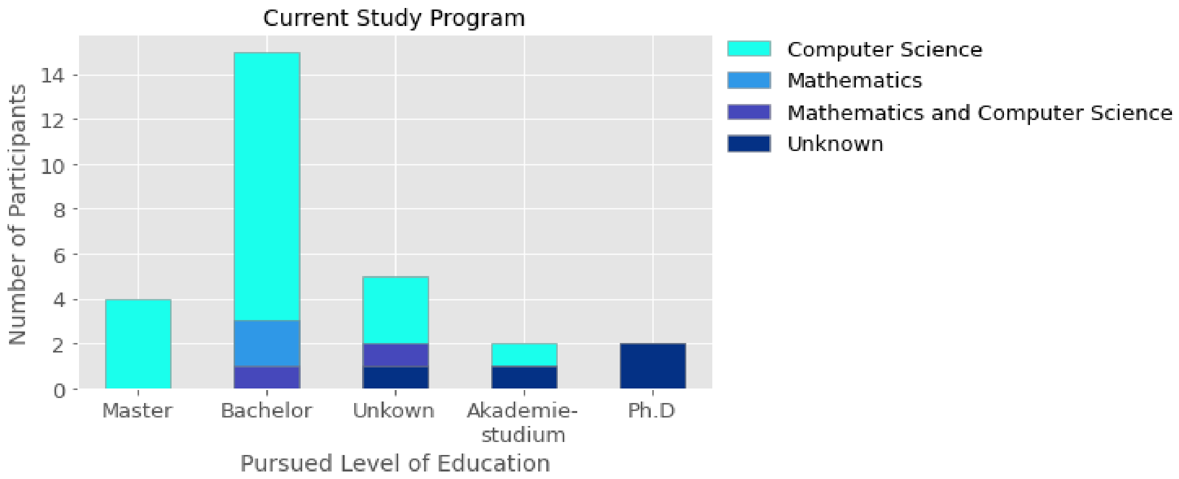


FIGURE 11: CURRENT STUDY PROGRAM

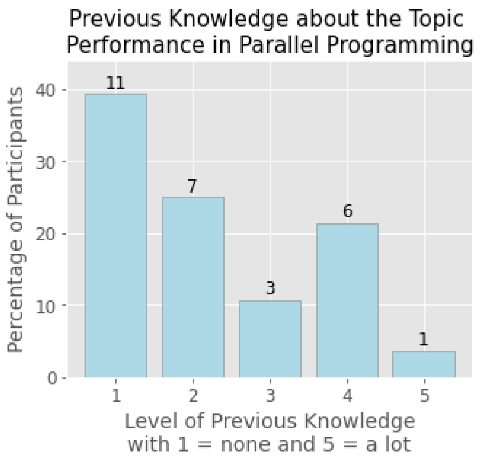


FIGURE 12: PREVIOUS KNOWLEDGE ABOUT THE TOPIC PERFORMANCE IN PARALLEL PROGRAMMING

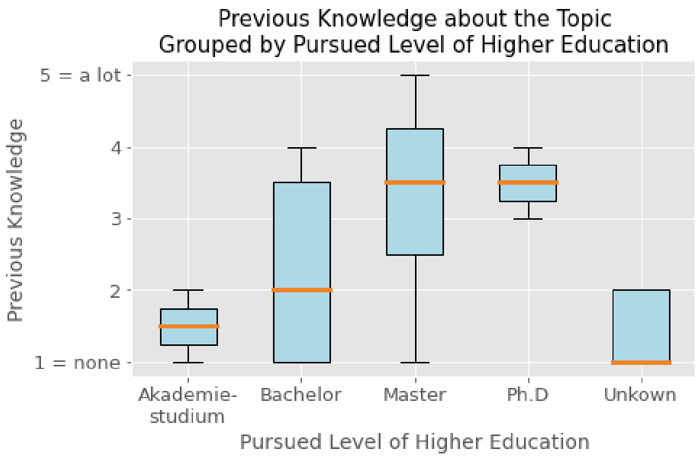


FIGURE 13: PREVIOUS KNOWLEDGE ABOUT THE TOPIC GROUPED BY PURSUED LEVEL OF HIGHER EDUCATION

Current Study Program of the Participants

Out of the 28 participants, more than half are undergraduate students enrolled at FernUniversität Hagen with the majority studying Computer Science. One participant stated to be enrolled at FH Achen studying Applied Mathematics and Computer Science, and another participant stated not be studying anymore. FernUniversität Hagen offers a special program called 'Akademiestudium' which enables people who do not fulfil the entry qualifications for a Bachelor's or Master's program to enrol in eligible courses. Two participants stated to be enrolled in that program.

Figure 11 displays the current study program and the level of education that is pursued. Since this information was collected with an open question, there are some missing values which are encoded as 'unknown'.

Previous Knowledge about Performance in Parallel Programming

About 64% of all participants reported to have had no or very little knowledge about performance in parallel programming prior to working through the notebooks. On a five-point Likert scale where one equals 'none' and five equals 'a lot', the medium lies as two and the average at 2.25. Figure 12 displays

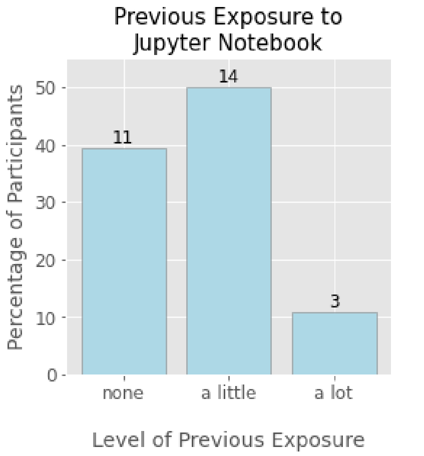


FIGURE 14: PREVIOUS EXPOSURE TO JUPYTER NOTEBOOK

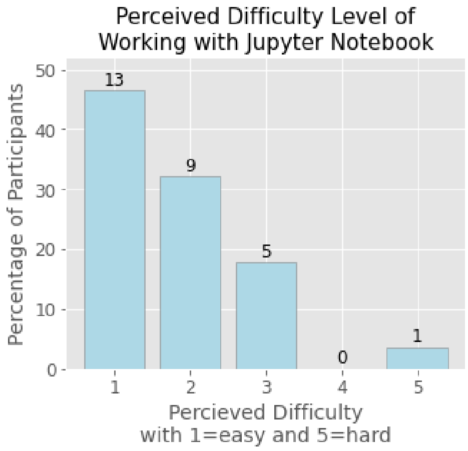


FIGURE 15: PERCEIVED DIFFICULTY LEVEL OF WORKING WITH JUPYTER NOTEBOOK

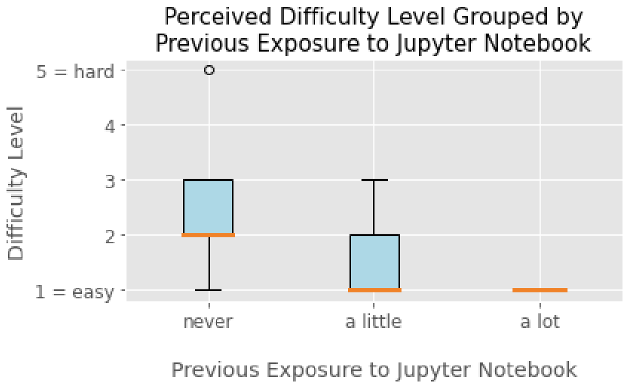


FIGURE 16: PERCEIVED DIFFICULTY LEVEL GROUPED BY PREVIOUS EXPOSURE TO JUPYTER NOTEBOOK

the distribution of the given answers. Even though most participants were new to the topic there have also been a few participants with advanced prior knowledge about performance in parallel programming. Only one participant stated to have 'a lot' of previous knowledge about the topic. Figure 13 reveals that participants who pursue a degree on a higher educational level, such as Master's or Ph.D., were also more likely to report higher levels of previous knowledge about performance in parallel programming.

Previous Exposure to Jupyter Notebook

Only 11% of the participants reported that they have worked 'a lot' with Jupyter Notebook in the past (see Figure 14). The other 89% of participants have either worked with Jupyter Notebook a little bit in the past (50%) or never (39%).

Perceived Difficulty Level of Working with Jupyter Notebook

On a five-point Likert scale where one equals 'easy' and five 'equals' hard, only one participant reported that it was hard to work with Jupyter Notebook. All other participants gave a rating between one and three. Figure 15 displays the distribution of the given ratings and Figure 16 shows a boxplot diagram of the perceived difficulty level grouped by reported previous exposure to Jupyter Notebook. The medium reported difficulty level lies at two for the participants who never worked with Jupyter Notebook

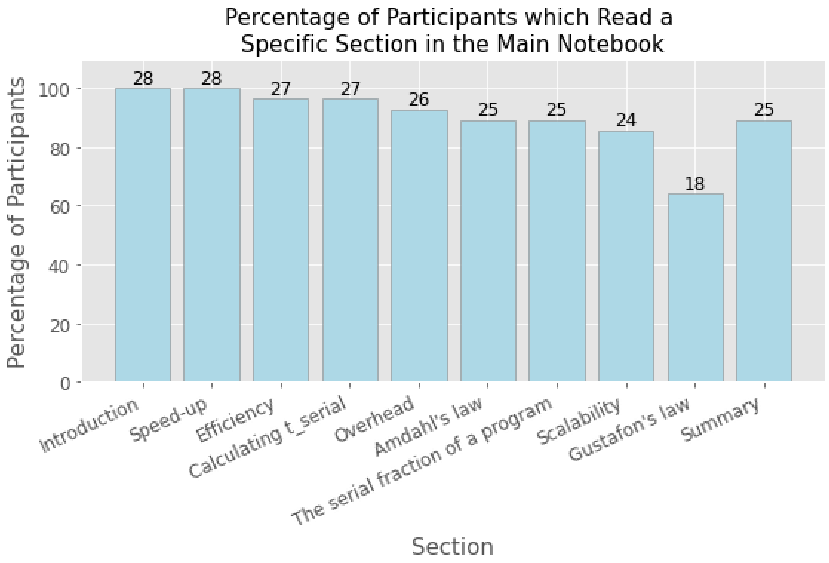
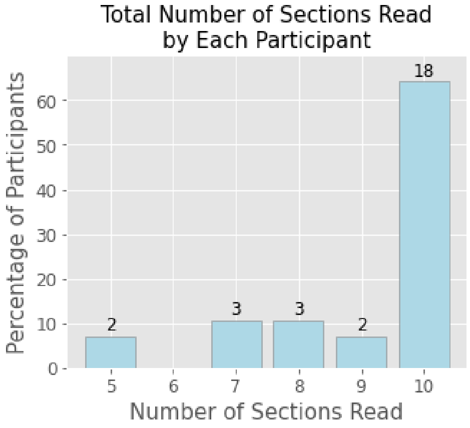


FIGURE 17: PERCENTAGE OF PARTICIPANTS WHICH READ A SPECIFIC SECTION IN THE MAIN NOTEBOOK



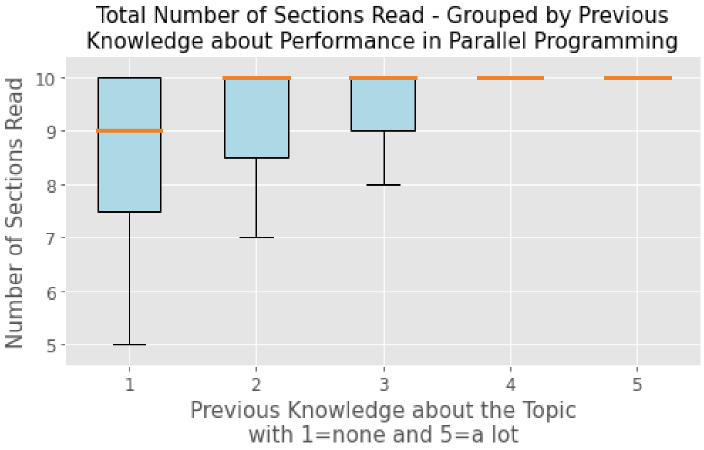


FIGURE 18: TOTAL NUMBER OF SECTIONS READ BY EACH PARTICIPANT

FIGURE 19: TOTAL NUMBER OF SECTIONS READ - GROUPED BY PREVIOUS KNOWLEDGE ABOUT PERFORMANCE IN PARALLEL PROGRAMMING

before. For all other participants the medium reported difficulty level lies at one which means 'easy'.

We also specifically evaluated the ease of use with respect to the self-test questions, the code sections and the usage of the additional notebooks. The results showed that most participants understood how to work with these components. All participants understood how to submit answers to the self-test questions. Only one participant did not understand how to work with the additional notebooks. The same participant also did not understand how to work with the code sections. In total, two participants stated that they were not able to work with the code sections and yet another participant felt confused by both the code sections and the additional notebooks.

Sections of the Main Notebook that the Participants Read

Not all participants read the entire notebook. Figure 17 displays for each section in the main notebook the percentage of participants which read that section. The graphic displays the different sections in the same order as they were presented in the notebook. We can observe a relationship between where

Variable Pair	Variable Description	Corre-lation	Depen-dency	P-Value
G1 – D4	Perceived difficulty level of working with Jupyter Notebook – Previous exposure to Jupyter Notebook	- 0.57	0.325	0.001
G1 – D3	Perceived difficulty level of working with Jupyter Notebook – Previous knowledge about performance in parallel programming	- 0.463	0.214	0.013
G1 – R1	Perceived difficulty level of working with Jupyter Notebook – Total number of sections read	- 0.438	0.192	0.02
D3 – D4	Previous knowledge about performance in parallel programming – Previous exposure to Jupyter Notebook	0.55	0.299	0.003
D3 – R1	Previous knowledge about performance in parallel programming – Total number of sections read	0.492	0.242	0.007
D4 – R1	Previous exposure to Jupyter Notebook – Total number of sections read	0.507	0.257	0.006

TABLE 3: SPEARMAN'S RHO COEFFICIENT WHERE P<0.05

in the notebook a section is placed and the likelihood that the section was read. The earlier a topic was introduced, the more likely that section was read. The only exception to this trend is the last section which is the summary.

Figure 18 shows the distribution of the number of sections each participant read. In total, 64% of the participants read the entire notebook, which contains 10 sections. It can also be observed that the more prior knowledge about performance in parallel programming the participants reported, the more sections of the notebook they read (see Figure 19).

Time Spent Working on the Notebooks

The time that the participants spent working through the notebooks varied a lot amongst the participants. For the main notebook the reported times range between half an hour and six hours, with the medium at two hours. For the additional notebooks the participants reported to have spent up to four hours. Nine participants did not look at the extra notebooks at all. Considering only the participants that looked at the additional notebooks, the average lies at 38 minutes and the median at 30 minutes.

Only three participants reported to have looked at every additional notebook. Thus, the reported times refer to the amount of time the participants were willing to spend on the extra notebooks, and not to the time it takes to completely work through all of them.

Correlation Analysis

To get an overview about possible dependencies between the ordinal as well as ratio variables, the Spearman's Rho coefficient was calculated for each pair [80, p. 151]. Table 3 summarizes the calculated correlations, dependencies, and p-values for all correlations with a p-value smaller or equal to $\alpha = 0.05$.

The correlation coefficients show a moderate correlation between each of the four variables G1[4], R1[5],

[4] G1 = Perceived difficulty level of working with Jupyter Notebook

[5] R1 = Total number of sections read

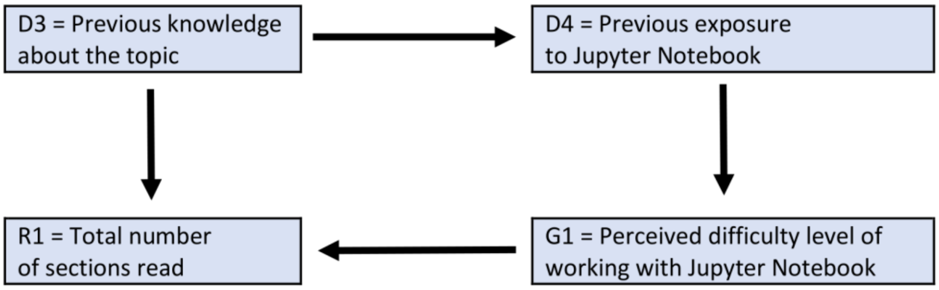

D3 = Previous knowledge about the topic
D4 = Previous exposure to Jupyter Notebook
R1 = Total number of sections read
G1 = Perceived difficulty level of working with Jupyter Notebook

FIGURE 20: HYPOTHESIS ABOUT RELATIONSHIP BETWEEN NON-CATEGORICAL VARIABLES

D3[6] and D4[7]. The following suggestions about how these variables might influence each other are merely hypotheses which need to be confirmed with further research [88, p. 56].

The relationship between the variables G1 and D4 is negative. Variable D4 is the independent variable. The more previous experience with Jupyter Notebook the participants reported, the less they found it difficult to work with Jupyter Notebook. Out of all correlations listed in Table 3 this is the strongest one and the one where causation is most trivial (see Figure 16).

G1 also negatively correlatives with D3; the more previous knowledge about parallel programming the participants reported, the less they found it difficult to work with Jupyter Notebook. This could be explained by the positive relationship between the variables D3 and D4; participants who reported previous knowledge about the topic were more likely to also report previous exposure to Jupyter Notebook. We propose a chain of causation, where D3 causes D4 and D4 causes G1. Thus, D3 indirectly causes G1 which explains the negative relationship between these two variables (see Figure 20). In other words, if students are likely to get exposed to Jupyter Notebook during studies about performance in parallel programming then they will find it easier to work in the Jupyter Notebook environment.

Another negative correlation can be observed with the variables G1 and R1. We assume that the perceived level of difficulty acts as the independent variable in this relationship. The more difficult it was for the participants to work in the Jupyter Notebook environment, the less sections did they read.

Variable R1 also positively correlates with the variables D3 and D4. Again, we propose a chain of causation. If D3 causes D4, D4 causes G1 and G1 causes R1, then D3 and D4 will indirectly cause R1 (see Figure 20). In other words, if students are likely to get exposed to Jupyter Notebook during studies about performance in parallel programming, then they will find it easier to work in the Jupyter Notebook environment which in turn will cause them to read more sections of the notebook. Another plausible causation could be a direct one from D3 to R1; the more previous knowledge about the topic the participants have, the more sections of the notebook they complete. Figure 19 reveals that the number of 'sections read' grows continuously in dependence of the reported previous knowledge about performance parallel programming.

Non-categorical variables which did not show any significant correlations with other non-categorical

[6] D3 = Previous knowledge about performance in parallel programming
[7] D4 = Previous exposure to Jupyter Notebook

Dimension	Subdimension	Agree-ments	Percen-tage	Statement
Opinion about the Ability to Foster Learning	Engagement	22	78.6 %	The questions made the notebook more interesting
	Understandability	18	64.3 %	The questions made it easier for me to understand the topic
	Retainability	17	60.7 %	The questions helped me to better remember what I learned
	Different Perspective	9	32.1%	The questions made me look at the topic from a different perspective
	Distraction	0	0 %	The questions distracted from the main content of the notebook
Opinion about Quality	Level of difficulty	5	17.8 %	The questions were too easy
		0	0 %	The questions were too difficult
	Frequency	3	10.7 %	There were not enough questions
		0	0 %	There were too many questions

TABLE 4: AGREEMENTS WITH THE STATEMENTS ABOUT THE SELF-TEST QUESTIONS

variables are D1[8], T1[9] and T2[10]. The pursued level of education and the time spent working on the notebooks did not correlate with each other or with any of the other variables displayed in Figure 20.

In addition to the correlation analysis of the non-categorical variables, we also calculated the correlation coefficients of other variable pairs where one variable is ordinal and the other dichotomous, or where both variables are dichotomous. Unfortunately, we were not able to derive meaningful results from these calculations. The reason for this could be the restricted range of dichotomous variables, which can be especially problematic if the sample size is small [80, p. 155]. Thus, the results of the correlations which involved dichotomous variables had to be discarded and will not be presented in this thesis.

Opinion about Self-Test Questions

The participants were given a list of 10 statements, out of which they were supposed to mark the ones they agree with. Table 4 summarizes the given answers.

The given answers indicate a positive attitude towards the ability of the self-test questions to foster learning. Almost 80% of all participants agreed, that the questions make the notebook more interesting and over 60% of the participants believed that the questions have a positive influence on understandability and retainability. Further, 32% felt that the questions made them look at the topic from a different perspective. None of the participants perceived the questions as distracting.

With respect to the difficulty level, about 20% found the questions too easy and nobody believed them to be too difficult. Thus, it can be assumed that about 80% of the participants perceived the difficulty level as adequate. Similar results can be seen with the frequency. About 10% said that there should be more questions, and no one believed that there should be less. Therefore, we assume that about 90% of the participants estimated the number of questions as suitable.

[8] D1 = Pursued level of education

[9] T1 = Time spent working on the main notebook

[10] T2 = Time spent working on the additional notebooks

Dimension	Subdimension	Agree-ments	Percen-tage	Statement
Opinion about the Ability to Foster Learning	Engagement	16	57.1 %	The pictures made the notebook more interesting
		7	25 %	The first thing I did was look at all the pictures
	Retainability	5	17.8 %	The pictures helped me to retain the information better
	Distraction	5	17.8 %	I felt distracted by the pictures
	Different Perspective	3	10.7 %	The pictures made me look at the topic from a different perspective
	Understandability	2	7.1 %	The pictures made it easier for me to understand the topic
Opinion about Quality	Professional Appearance	8	28.6 %	Pictures like this make the notebook look less professional
	Frequency	8	28.6 %	There were too many speech-bubble pictures
		2	7.1 %	I was hoping there would be more speech-bubble pictures
	Adequacy	3	10.7 %	Pictures like this should not be used in a university course

TABLE 5: AGREEMENTS WITH THE STATEMENTS ABOUT THE SPEECH-BUBBLE PICTURES

Opinion about the Speech-Bubble Pictures

Overall, the participants did not feel that the speech-bubble pictures had a great influence on their learning outcome. Even though almost 60% agreed that the pictures make the notebook more interesting, only 18% suspected a positive influence on retainability, about 11% felt that the pictures gave them a different perspective on the topic and 7% agreed that they understood the topic more easily due to the pictures. Further, about 18% felt distracted by the pictures.

With respects to quality, statements about frequency, adequacy and professional appearance were asked. Almost 30% think that the speech-bubble pictures make the notebook look less professional and about 10% perceive these types of questions as inadequate for a university course. Mixed feedback was given regarding the frequency of the pictures. Almost 30% think there were too many speech-bubble pictures, and 7% would have liked more. Thus, it can be assumed that about 60% of the participants perceived the frequency of speech-bubble pictures as adequate.

Table 5 summarizes the given answers.

Opinion about the Code Sections

Amongst all statements about the pedagogical methods, the statement that the notebook became more interesting thanks to the code sections received the highest number of agreements, with 82% of all participants agreeing. The other subdimensions related to learning outcome received good feedback as well. About 50% of the participants think that the code sections had a positive influence on retainability and understandability and nobody felt distracted by the code sections. Only 14% think that the code sections helped them get a different perspective on the topic.

With respect to the frequency, 14% would like more code sections and nobody thinks there were too many. Thus, it can be concluded that 86% considered the number of code sections as suitable.

Table 6 summarizes the given answers.

Dimension	Subdimension	Agree-ments	Percen-tage	Statement
Opinion about the Ability to Foster Learning	Engagement	23	82.1 %	The notebook became more interesting thanks to the code sections
	Retainability	15	53.6 %	The code sections helped me retain the learned information better
	Understandability	13	46.4 %	The code sections made it easier for me to understand the topic
	Different Perspective	4	14.3 %	The code sections helped me look at the topic from a different perspective
	Distraction	0	0 %	The code sections distracted from the main content
Opinion about Quality	Frequency	4	14.3 %	I would have liked more code sections inside the main notebook
		0	0 %	There were too many code sections

TABLE 6: AGREEMENTS WITH THE STATEMENTS ABOUT THE CODE SECTIONS

Dimension	Subdimension	Agree-ments	Percen-tage	Statement
Ability to Foster Learning	Understandability	9	47.4 %	I gained a deeper understanding of the topics thanks to the extra notebooks
	Engagement	7	36.8 %	The extra notebooks made the topic more interesting
	Retainability	5	26.3 %	The extra notebooks helped me retain the learned information better
	Different Perspective	4	21 %	The extra notebooks made me look at the topic from a different perspective
	Distraction	2	10.5 %	The extra notebooks distracted from the main topic

TABLE 7: AGREEMENTS WITH THE STATEMENTS ABOUT THE ADDITIONAL NOTEBOOKS

Opinion about the Additional Notebooks

In total, 32% reported not to have looked at the extra notebooks. The percentages for all other statements about the additional notebooks are calculated for the remaining 19 participants who looked at the extra notebooks. Table 7 summarizes the given answers.

Only half of the participants who looked at the additional notebooks also tested different programs for performance. Almost 50% considered the extra notebooks to have a positive impact on their understanding of the topic. Further, about 37% found the notebook more interesting due to the extra notebooks. A positive impact on retainability was assumed by 26%, and 21% felt that the additional notebooks gave them a different perspective on the topic. In addition, 10% stated that they felt distracted by the extra notebooks.

Comparison of the Opinions with Respect to the Pedagogical Method's Ability to Foster Learning

Figure 21 displays for each one of the evaluated pedagogical methods how many percent of the participants felt that this method had an influence on their learning outcome.

Overall, the self-test questions and the code section are the methods that count the highest numbers of agreements with respect to the ability to foster learning. These are also the two methods that no

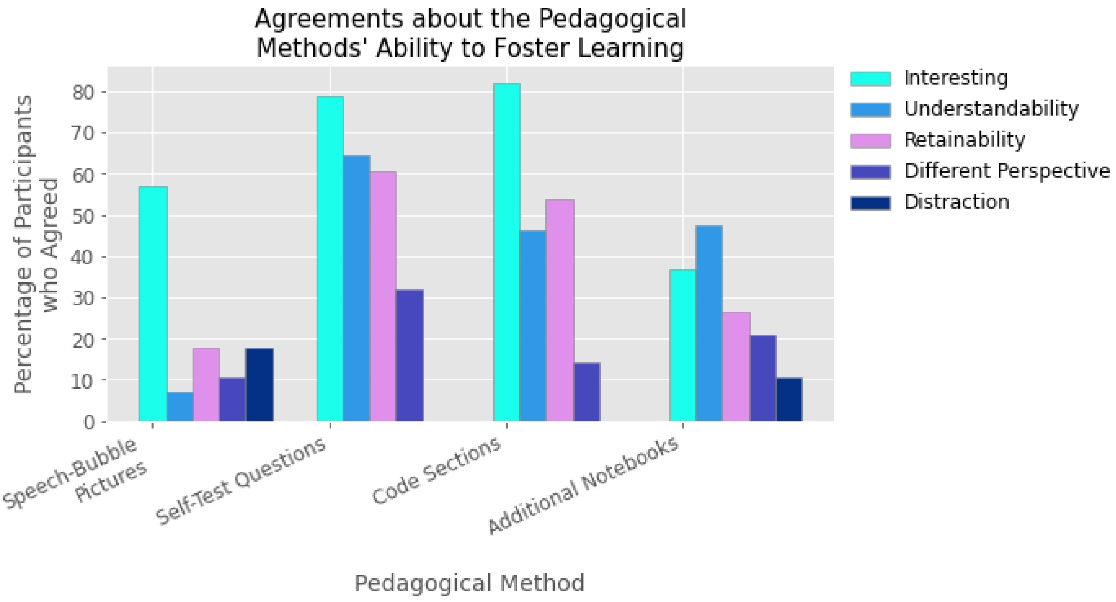


FIGURE 21: AGREEMENTS ABOUT THE PEDAGOGICAL METHOD'S ABILITY TO FOSTER LEARNING

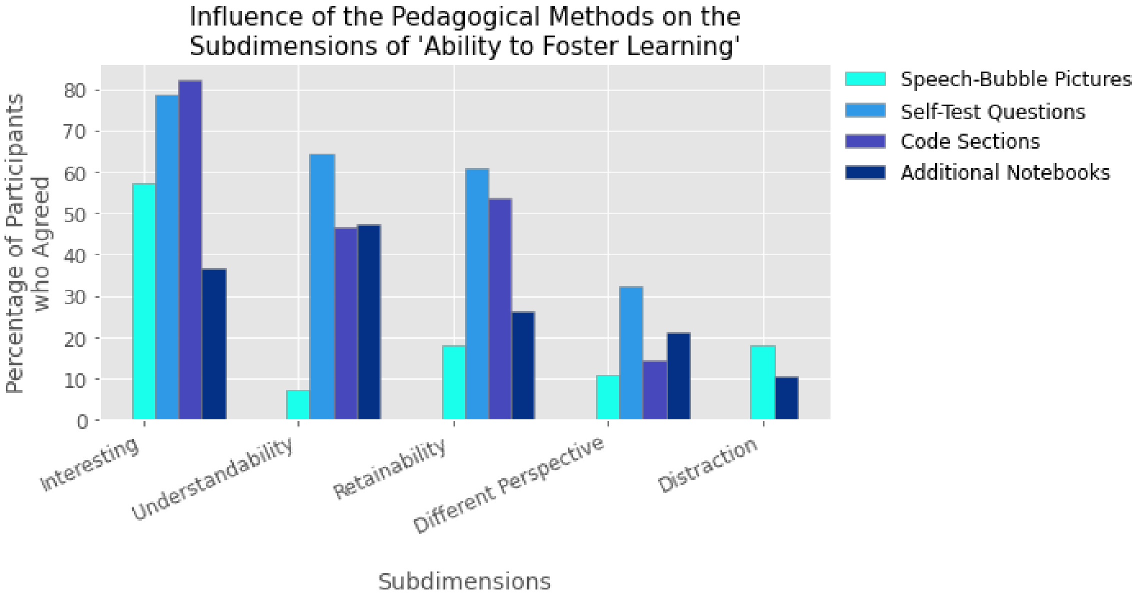


FIGURE 22: INFLUENCE OF THE PEDAGOGICAL METHODS ON THE SUBDIMENSIONS OF 'ABILITY TO FOSTER LEARNING'

participant perceived as distracting. About 80% of the participants felt that the code sections and the self-test questions made the notebooks more interesting. Further, about 50 to 60% of the participants felt that the code sections and self-test questions had a positive influence on the understandability and retainability of the learned theory. Only about 30% and 15% of the participants felt that the self-test questions and code sections helped them get a different perspective on the topic, respectively. Moderate influence on learning outcome was attributed to the additional notebooks. The speech-bubble pictures received the smallest number of agreements. Even though almost 60% of the participants agreed that the speech-bubble pictures make the notebook more interesting, less than 20% attributed a positive influence to the other subdimensions of learning outcome.

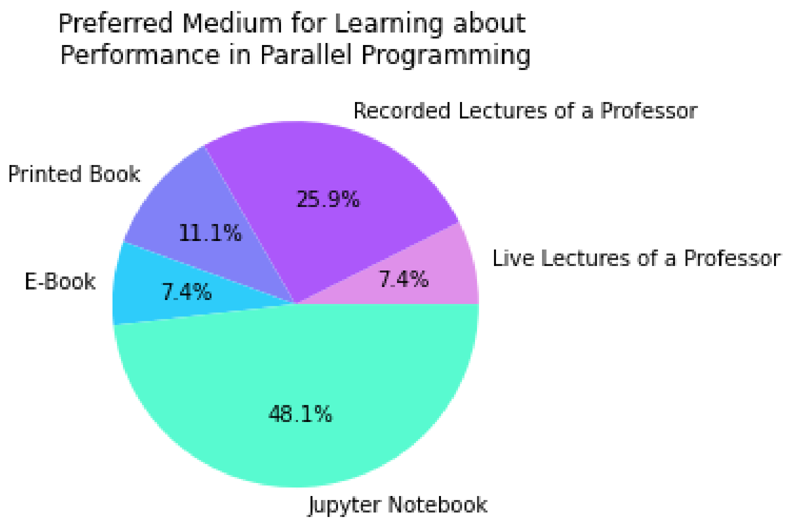


FIGURE 23: PREFERRED MEDIUM FOR LEARNING ABOUT PERFORMANCE IN PARALLEL PROGRAMMING

Preferred Medium for Learning about the Topic

The participants were given a list of different learning mediums and were asked to select the one they would prefer the most when learning about performance in parallel programming. Figure 23 gives an overview about the given answers. One option that is not displayed in Figure 23 is the learning medium 'PowerPoint Slides', since this option was not chosen by any of the participants.

At most universities content is delivered through live lectures by a professor. Only 7.4% would choose this traditional way of learning when it comes to the topic of performance in parallel programming. Recorded lectures, on the other hand, were chosen by 25.9 % of the participants as their favourite medium for learning about the topic. Thus, in total 33.3% would like to learn with lectures.

Learning with a book would be the preferred choice by 28.5% of the participants, out of which 11.1% would want a printed book and the other 7.4% an e-book.

Jupyter Notebook was chosen by 48.1% as the preferred medium for learning about performance in parallel programming. If given the choice between PowerPoint slides, lectures, books, and Jupyter Notebook, almost half of the research participants would choose Jupyter Notebook.

5.3.2 Qualitative Analysis

Testing of own programs

Only two participants reported to have tested one or more of their own programs. One of the participants stated that it was sometimes difficult to explain the achieved result. The other participant said that *"most problems are too small to really see the speed-ups"* and that larger speed-ups can be observed on a 'Raspberry Pi'. Neither one of the two participants mentioned technical issues with the process of testing own programs in the provided Jupyter Notebook environment.

What the Participants Liked About the Notebook

Out of 28 participants, 23 have answered question 15: *"What did you like about the notebook?"*

The keyword that was mentioned most by the participants was 'interactivity'. Participant 1 said: *"The notebook is a good method to learn interactively […]"* and participant 26 liked the interactive components which allowed to 'explore the phenomena on a real system'. In total, ten participants communicated their liking for the interactivity in the notebook.

The code sections too have received good feedback. Seven participants mentioned that they appreciated the ability to 'play around with code' and to 'try out code'. Out of these seven, four participants specifically mentioned the fact that in Jupyter Notebook the code can be directly interleaved with textual instructions, which allows to run code 'directly' and 'immediately', without having to switch between different programs.

Six participants commented on the explanations given in the notebooks, which are described as 'easily understandable', 'profound', 'practical', 'demonstrative' and 'detailed'. Further, participant 28 remarked on the 'detailed and thorough approach', participant 27 liked the 'mix of detail and overview', participant 3 mentioned the 'nice introduction to central terms' and participant 21 liked the 'compact forwarding of information'.

Another point that has been mentioned by five participants is the presence of test questions. Participant 5 said that the questions resulted in a 'more active role during the learning process' and participant 3 considered the test question design a good starting point for 'deeper user interaction'.

A liking for the graphics has been mentioned by five participants. Out of these, participants 1 and 4 also expressed their appreciation for the ability to change the code of the graphics and to immediately see the resulting changes.

Five participants said that they liked the given examples, out of which participant 6 and 7 specifically mentioned the 'sand story' example. Also the structure of the instructions was liked by five participants.

Four participants expressed a liking for the pictures, out of which participant 24 specifically mentioned a liking for the 'unusual way with the speech-bubbles'. It is not clear, which type of pictures the other three participants were referring to. Since participant 4 and 12 also mention a liking for the graphics, it can be argued that they meant the speech-bubble pictures, when they expressed a liking for the 'pictures'. Further, participant 6 stated: *"I liked the pictures. They made me smile"*. This too can be interpreted as referring to the speech-bubble pictures. Thus, we assumed that in total four participants communicated a liking for the speech-bubble pictures.

The keyword 'interesting' was mentioned by two participants. Participants 7 said that the text was written in an interesting manner, and participants 12 stated that the pictures, graphics and code sections made it more interesting to work with the notebooks.

Two participants expressed their liking for the casual writing style and participant 9 described the notebook as 'friendly and motivating'.

Participant 12 appreciated the additional notebooks, saying that they inspire to deepen the gained understanding without being overwhelmed by 'too much text'.

Finally, participants 22 stated: *"It is another approach towards the learning [of the] topic".*

Table 8 gives an overview about the likes and dislikes of the participants with respect to the developed notebooks. The column 'count' specifies the number of participants who communicated the specific like or dislike.

Likes	Count
Interactivity	10
Code sections	7
Explanations	6
Self-test questions	5
Graphics	5
Examples	5
Structure of the text	5
Speech-bubble pictures	4
Interesting	2
Writing style	2
Additional notebooks	1
New approach to learning	1

Dislikes	Count
Writing style	4
Speech-bubble pictures	3
Having to run the code cell first in order to see the input field of self-test questions	2
Lack of portability	2
Difficulty level of more advanced sections	2
Errors	2
Missing introduction about how to work with Jupyter Notebook	2
Unprofessional appearance	1
Video clip in introduction	1
Not enough practical content	1
Issue with LaTeX integration	1
Too much theory, not enough practice	1
Losing the current reading position when clicking on links that point to other locations in the notebook	1
Inability to test self-written programs due to lack of knowledge	1

TABLE 8: OVERVIEW ABOUT LIKES AND DISLIKES

What the Participants did not Like About the Notebook

Out of 28 participants, 21 have answered question 16: *"What did you not like about the notebook?"*

Compared to the previous section, the answers given to question 16 were more individual with less similarities to each other.

The biggest agreement could be identified with respect to the writing style. Four participants expressed their dislike for the chosen style of writing, describing it as 'casual', 'too colloquial for a scientific course', 'sometimes inappropriate' and 'somewhat flippant'. One of these four participants also did not like the usage of active voice.

Another area where some consensus could be identified was with respect to the speech-bubble pictures. Only participant 3 specifically mentioned the speech-bubble pictures, but participant 21 stated not to like the 'silly images' and participant 23 stated that the 'unrelated pictures' were 'distracting' and 'a bit too ADHDy'. Based on the word choice it can be argued that the participants 21 and 23 were referring to the speech-bubble pictures. Thus, we suggest that three participants communicated a dislike for this picture type. Further, participants 3 also expressed a dislike for the short movie clip used at the very beginning of the notebook.

Two participants have mentioned that they did not like that the input field for questions only appears once the corresponding code cell is run.

Issues with portability were mentioned by two participants, with participants 9 saying that *"it took some time to make it work on Windows"* and participants 24 stating that some adjustments had to be done before the notebook could be run locally.

Participants 6 and 27 both mentioned that they found it hard to understand the later parts of the notebook, with participant 27 specifically naming Amdahl's law and Gustafson's law.

Participant 18 said that an introduction about how to use the Jupyter Notebook was missing and participant 7 suggested that the explanation about how to work with Jupyter Notebook should be better.

Participant 2 did not like the spelling errors, which are described as 'disturbing for the reading flow' and participants 12 expressed a dislike for grammatical errors. Participant 2 also remarked that the appearance of the notebook seems at times 'unprofessional'.

Participant 14 did not like the amount of time it took to work through the notebooks, and participant 23 felt that there was 'too much theory' and not enough 'practical info'.

Further, participant 17 mentioned that the LaTeX formulas stopped working at one point, participant 25 communicated a dislike for the inability to highlight text in Jupyter Notebook and participant 4 mentioned the problem that it is not possible to navigate back to the current reading position after clicking on a notebook internal reference link. Participants 5 did not like that she/he was not able not test any own programs, explaining that it was not possible due to being a 'beginner'.

Suggestions Given by the Participants

Out of 28 participants, 16 have given suggestions about how the first version of the notebooks should be further adjusted. In total 2042 words were submitted, out of which 75.6% were given by three participants who provided a very detailed list with spelling errors and other content-related errors and suggestions.

Five participants have made suggestions with respect to the speech-bubble pictures, asking to either leave them out completely or to put less of them. Participant 1 suggested *"a professional look without speech-bubble pictures".* Also, participants 2 suggested not to use these types of pictures, stating that they are 'nice to look at', but often seem 'rather childish'. Participants 19 and 21 suggested using less speech-bubble pictures. Participant 23 did not specify if there should be less pictures, or none at all, but gave a detailed opinion about the pictures instead*: "[…] it seems to me that the images used here were a desperate attempt at making the participants funny. It doesn't seem that they help with the subjects or fit in naturally. And while I do like that there is a high number of women represented, […] I think the selection of stereotypical young, sexy women in the pictures doesn't really help with representing a diverse crowd or the average (female) student at FernUniversität Hagen".*

In total, five participants made suggestions regarding spelling errors. The participants either pointed out specific spelling errors or gave a general suggestion that the notebooks need proofreading.

Regarding the test questions, participants 4 and 18 both felt that the answer of '0.2' should evaluate as correct in the test question 18. Further, participants 2 suggested to adjust the code for the test questions so that answers can be case insensitive and participants 18 suggested that it should be possible to use a comma instead of a dot to represent the decimal point in numeric answers.

Two participants made suggestions with respect to the writing style, with participant 19 proposing not to use the active voice and participant 1 recommending more formal wording to increase the acceptance as an academic teaching material.

The first version of the notebooks uses absolute path names to change between directories. Two participants propose to use a different solution instead. Participant 22 suggested using the 'actual notebook directory'. Participants 18 felt that all the 'cd' commands should be removed, for example by saving the present working directory in the beginning and restoring it at the end of the scripts, or by getting the path at load time.

Several participants suggested content related adjustments. Participants 3 and 18 mentioned that it was unclear whether the truck driver in the sand beach analogy is also helping with loading and unloading the truck. Participants 4 and 23 pointed out that p has not been defined yet when it was used the first time in mathematical formulas. Participant 4 also mentioned some unclarities with regards to strong and weak scaling and proposed to explain better why there was no weak scaling observed for higher thread numbers in the matrix-matrix multiplication example. Further, participant 4 felt that efficiency would better be explained as average degree of capacity utilization, that the term 'Gustafson's law' should not be used before it is introduced, and that some of the points about the limitations of Amdahl's law are too much alike. Both, participants 4 and 2, suggested a more detailed explanation about whether setting the thread number to one for the calculation of the serial run-time is acceptable or not. Participant 2 also proposed to give a more detailed explanation about the used sorting algorithms, the caching effects and the different timing methods.

Other suggestions made on individual basis are:

- Making the notebook runnable on Windows
- Deleting the video in the introduction
- A final test about all sections
- Placing the different sections of the notebook into separate notebooks
- A handout summarizing crucial information such as vocabulary and formulas
- More highlighting of the text using tools such as colors and bold/italic font
- Different versions of the notebooks for beginners and advanced students
- More information about Jupyter Notebook itself
- Adding the exact page numbers to the references
- More complex data structures in the benchmarks
- Recreating the picture about the sand beach analogy by using a timeline
- Changing the structure of the text
- Displaying all mathematical terms with the usage of LaTeX
- More precise labeling of the graphics
- Indenting Python comments

5.4 Limitations

In this section we will briefly discuss some factors which might have negatively impacted the validity and reliability of the presented results.

5.4.1 Order and Recency Effect in Multiple-Choice Questions

The multiple-choice question come with the risk of some undesired effects which negatively influence reliability. One such effect is the order effect which results in a tendency of participants to judge later statements in terms of earlier ones [81, p. 478]. Another negative influence on reliability is the recency effect, which says that items later in a list will be more likely remembered by the participants than items earlier in a list [81, p. 478]. According to Cohen et al. [81, p. 478] nothing can be done to prevent the order and recency effect from happening. We would possibly get different research results, were we to repeat the research with the same questionnaire, but with a different ordering of the statements. Thus, the reliability of the research instrument is negatively impacted by the order and recency effect.

5.4.2 Completeness of the Multiple-Choice Questions

For the multiple-choice questions the participants were asked to mark all the statements they agree with. There is a risk that participants who are in a rush to complete the questionnaire might not carefully read each statement. This is a thread to validity, as some participants might not have marked all the statements they agree with.

To better understand whether this was indeed an issue in the evaluation, we took a closer look at all answers given by the participants who did not answer any of the open questions. We wanted to know if the participants who did not take the time to answer any of the open questions might have also spent less time on the multiple-choice questions.

In total, there were four participants who did not answer any of the optional open questions. In average, these four participants marked 19.7% of all multiple-choice statements whereas the other 24 participants marked in average 22.9%. Considering the rather small difference, it did not seem justified to eliminate the answers of the four participants who did not answer any of the open questions.

5.4.3 Accuracy and Honesty in Self-Reports

One disadvantage of self-reports is that results only reflect the participants' subjective point of view, which can be biased and distorted by many factors [89, p. 65]. No objective conclusion about the influence of the notebooks on learning outcome can be drawn based on a self-report. Nonetheless, a self-report study can provide valuable information about the participants' attitude towards the learning material and areas where the notebook can be further improved.

Another issue with self-reports is that the researcher needs to rely on the participants truthfulness [80, p. 62]. It is not possible to estimate the level of honesty that the participants had when answering the questions. Since the participation in the evaluation was anonymous, it can be argued that the participants had no reason for dishonesty.

5.4.4 Size and Representativeness of the Sample

With the 'convenience sampling technique' no consideration has been taken to select a representative sample of a certain population [83, p. 145]. For this reason, and also due to the rather small sample size, the results of the evaluation cannot be generalized to a specific group of individuals [80, p. 214]. Instead, the results need to be interpreted as relating to the studied participants only.

6 Discussion of the Results

In this chapter we will discuss the presented results. First, some of the results related to the participants' behavior when working through the notebooks are discussed. Then we will elaborate on the feasibility of working with Jupyter Notebook and discuss the results with respect to the applied pedagogical principles. The chapter ends with a discussion about the most preferred mediums for learning about performance in parallel programming, and suggestions for future versions of the notebooks.

6.1 Behavior of the Participants

6.1.1 Time Spent on the Notebooks

The results have shown that there is no correlation between the number of sections read and the time spent working through the main notebook. This indicates that the participants were willing to dedicate a certain amount of time to working on the notebooks, regardless of how many sections they would be able to complete in that time frame.

6.1.2 Number of Sections that the Participants Worked Through

The correlation analysis revealed positive relationship between previous knowledge about the topic and the total number of sections read. We see two plausible reasons why the participants with less previous knowledge about the topic were less likely to read the entire notebook. First, these participants might have felt overwhelmed by the more advanced topics and thus decided to skip them. Second, participants might not have been willing to continue reading the notebook after a certain amount of time was spent on that task.

6.1.3 Exploring the Theory with Additional Material

The results indicate that the opportunity to test different programs for performance was not leveraged by most of the participants. About two thirds of the participants looked at the some of the additional notebooks, and one third tested some of the provided benchmarks for performance. Only three participants worked through all of the additional notebooks and the opportunity to test a self-written program was used by two participants.

We explain this rather decent interest in the additional material with the fact that working through the notebooks was an 'out of curriculum' activity for the research participants. It can be expected that students would spend more time on explorative content if the notebooks were part of a university course. Also, there were no exercises that prompted the learners to solve certain tasks with the help of the additional notebooks. Providing specific exercises would probably increase the learners' engagement with the explorative content.

6.2 Feasibility of Working with Jupyter Notebook

The results indicate that working with Jupyter Notebook was a feasible task. Most participants who have already worked with Jupyter Notebook in the past found it easy to work with Jupyter Notebook. Also, the participants who were new to Jupyter Notebook perceived the level of difficulty as moderate, giving a median rating of two on a rating scale where one equals easy and five equals hard. Only two participants reported that they did not understand how to work with the code sections, out of which one participant also did not understand how to work with the additional notebooks. Two different participants stated that they would have liked more explanations about how to work with Jupyter

Notebook. We suggest that providing students with more detailed information about Jupyter Notebook would lower the perceived difficulty level of working in this environment.

With respect to the ability to test self-written programs for performance, we cannot make conclusions about the difficulty level of this task. Only two participants reported to have tested some of their own programs for performance and they did not mention any technical issues with this. More data is needed to confirm that the students are indeed able to test their self-written programs for performance with the provided notebooks.

6.3 Applied Pedagogical Methods

We will first discuss the methods which do not depend on ICT for their deployment, followed by the methods that can only be implemented with an ICT tool such as Jupyter Notebook.

6.3.1 Language Style and Speech-Bubble Pictures

The open questions have revealed that the participants have a wide range of opinions about the developed Jupyter notebooks, which sometimes oppose each other. This is especially true with respect to the used language style and the speech-bubble pictures, which are both methods that are traditionally not used in teaching material at university level.

Conflicting Opinions

While four participants explicitly expressed their appreciation for the speech-bubble pictures, another six[11] participants communicated their dislike for these methods and/or suggested not to use them. Similar results can be observed with the language style. Two participants explicitly stated that they liked the language style, and four[12] participants said they did not.

Three of the four participants who communicated their dislike for the language style also mentioned their dislike for the speech-bubble pictures. Thus, we can observe that the participants who did not like the conversational language style also were more likely to disagree with the usage of speech-bubble pictures. We propose that these participants display an overall more conservative attitude towards non-traditional teaching methods. On the other hand, there were no participants who expressed both, a liking for the speech-bubble pictures as well as a liking for the language style. Thus, it cannot be assumed that some learners in general just liked all the methods that are non-traditional.

The conflicting preferences make it difficult to estimate a level of general acceptance towards the speech-bubble images or used language style. Ultimately, it is almost impossible to create teaching material that satisfy every student's learning preference. The usage of ICT in education could potentially allow to offer personalized learning, meaning that the style and content of learning material can be dynamically adjusted to the learners' ability and preference. The only way to partially implement this in Jupyter Notebook, as of today, would be by creating multiple versions of the notebooks that the learners can choose from.

Since the participants do not agree about whether speech-bubble images and conversational language

[11] The speech-bubble pictures were mentioned in the section about what was disliked by one participant, in the section about suggestions by three participants, and in both of these two sections by two participants.

[12] The language style was mentioned in the section about what was disliked by four participants. Out of these four, two also mentioned the language style in the suggestions.

should be used, another criterium to decide about the future inclusion of these methods could be the influence on learning outcome.

Influence on Learning Outcome

The quantitative questions revealed that most participants do not attribute much influence on retainability, understandability and different perspective to the speech-bubble pictures. Almost 60%, however, feel that the notebooks became more interesting due to the speech-bubble pictures. It should be considered, however, that this type of pictures can also be perceived as distracting, which was affirmed by 17.8% of participants.

The question arises whether students know which pedagogical methods are most suitable for their learning outcome. Just because a student has a certain preference, it does not mean that this preference is the right choice from a pedagogical point of view. To give an example, there was one participant who said: *"I didn't like that I cannot highlight any for me important parts of the text. So I felt a little bit lost while reading".* Highlighting text is a very common technique many students rely on, but research suggest that this method has a negative influence on learning outcome [90].

Research has shown that unrelated pictures can negatively influence learning, even if the pictures are perceived as interesting [5, p. 164]. On the other hand, there was also research that showed an increase in learning outcome with the usage of pictures that follow an 'emotional design' [5, p. 167]. Based on the assumption that speech-bubble pictures can trigger emotions, we make the hypothesis that the speech-bubble pictures have a positive influence on learning. More research is needed to confirm or deny this hypothesis.

Trade-Off Between Educational Benefit and Professional Appearance

We suggest a trade-off between the educational advantage of speech-bubble pictures and professional appearance. Almost 30% agreed, that the speech-bubble pictures made the notebooks look less professional and about 10% felt that pictures like this are not adequate for a university course.

A similar trade-off is proposed for the language style. According to research results a conversational language style has a positive influence on learning outcome [5, p. 185]. Four participants, however, did not perceive a casual language style as adequate for a university course.

Whether speech-bubble pictures and informal language should be used in educational material depends on the priorities that an educational institution sets. A university that highly values a professional appearance should maybe not use these methods, while an institution that prioritizes learning outcome might want to consider them.

6.3.2 Pedagogical Methods that Depend on ICT for their Deployment

In the quantitative analysis we evaluated the participants' opinion about the integrated self-test questions, the code sections, and the additional notebooks. A comparison of these three methods has revealed that the participants attributed most influence on learning outcome to the integrated self-test questions and the code sections, and not so much influence to the additional notebooks.

We see two possible reasons for this. First, the participants might have perceived the additional notebooks as less influential on learning outcome, compared to the self-test questions and the code sections. Second, the participants might have not spent enough time on the additional notebooks to form a clear opinion about whether they agree with certain statements. It has already been discussed that most participants did not leverage the opportunity to test different programs for performance. This naturally also results in the participants marking less of the statements. The participants can only fully agree with certain statements about the additional notebooks if they spent sufficient time working with them.

In addition to the methods evaluated in the quantitative analysis, the participants also mentioned most of the applied pedagogical methods in the answers about what was liked about the notebooks. Overall, the feedback with respect to the code sections and self-test questions was very positive. Also, the interactive graphics and the additional material for exploration were mentioned. On the other hand, none of the participants communicated a dislike for any of the pedagogical methods that depend on ICT for their deployment. Thus, we propose that the participants had a positive attitude towards our usage of Jupyter Notebook as a medium for teaching.

We suggest that the biggest advantage of using Jupyter Notebook in education is the interactivity aspect, which has also been the most liked feature of the developed notebooks. All of the applied pedagogical methods that depend on an ICT tool such as Jupyter Notebook essentially enhance interactivity.

Some of the participants also mentioned the advantage of being able to run code right away in the same environment as the theory is taught. At most universities, there is a time gap between learning the theory and applying it. With Jupyter Notebook, this time gap can be minimized, which could prove as beneficial in the learning process. More research is needed to confirm this hypothesis.

6.4 Preferred Medium for Learning about the Topic

At FernUniversität Hagen the most used medium for teaching are printed scripts which have been written by a professor. Interestingly, only 11.1% of the participants would choose printed books as their favourite medium for learning about performance in parallel programming. Another 7.4% would choose an e-book instead, so in total 28.5% prefer to learn about the topic with books. It should be considered that the students had to choose one single option. At the FernUniversität Hagen, however, courses are usually taught with a combination of different learning mediums. Further, the question was specifically asked with respect to the topic 'performance in parallel programming'. Asking the same question for a different topic might have resulted in very different results.

Instead of learning with books, students at traditional universities usually learn with live lectures. Only 7.4% prefer this synchronous teaching style and 25.9% would choose asynchronous recorded lectures instead. In total 33.3% would like to learn about performance in parallel programming in lectures. It makes sense that students who are enrolled at a distance learning university prefer asynchronous learning.

At traditional universities students often prepare for exams with the PowerPoint slides that were presented during live lectures. None of the participants would choose PowerPoint slides as the preferred option for learning about performance in parallel programming.

If the research participants had to choose one single teaching medium for learning about the topic, 48.1% would choose Jupyter Notebook. Thus, almost half of all participants prefer Jupyter Notebook over other more traditional teaching mediums when learning about performance in parallel programming.

These results clearly indicate the diverse learning preferences of the research participants. Jupyter Notebook makes it possible, to some extent, to address these different preferences. One third of the participants would most like to learn about the topic with lectures. A recorded lecture could easily be integrated into a Jupyter notebook, by interleaving interactive components of the notebook with embedded video instructions. Further, it is possible to export a Jupyter notebook to pdf format, providing a printable version of the learning material. The interactive components and videos will obviously be lost in a printable version of the notebook. A printable version, however, could be used complementary to the actual notebooks by students who prefer to learn this way.

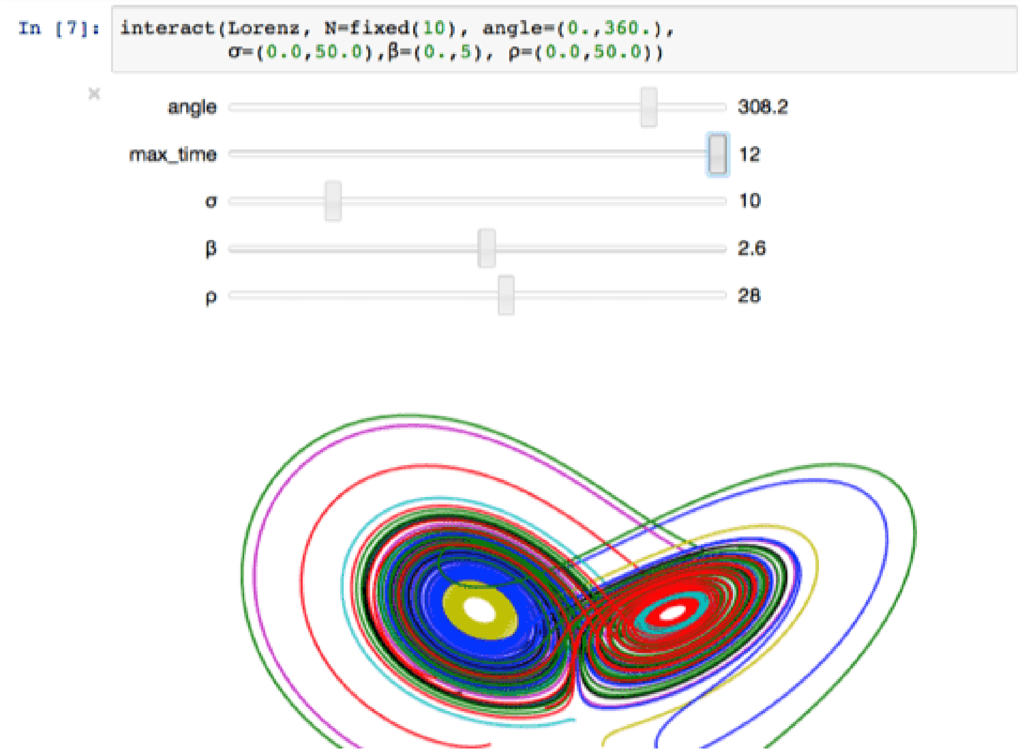
6.5 Suggestions for Future Versions of the Notebooks

The first draft of the notebooks has already been updated since their first release. The made updates include:

- Correction of spelling and grammar errors
- Improved LaTeX integration in textual instructions
- Adjustments of the naming of variables in the section about Gustafson's law
- Adjustments in the explanation of the sand beach analogy

Based on the provided feedback, personal reflection, and recent ideas we suggest the following further adjustments:

- Placing the references in a separate notebook, and link from the main notebook to that notebook whenever a reference is cited. This has the advantage that the learner does not lose the current reading position in the main notebook.
- Creating a notebook which explains the basics of how to work with Jupyter Notebook. At the very beginning of the main notebook, the students can be informed about the existence of this notebook and given the link which will open it in a new tab. Even though instructions about how to work with Jupyter Notebook are already given in the readme file, it is easy to overlook that this file exists.
- Placing the different sections in separate notebooks. Considering that there is no navigation menu, it can be hard to find recent reading positions. Separating the content into different notebooks will make it easier for the learners to navigate between the sections.
- Adjusting the picture of the sand beach analogy so that it looks more professional. One participant has mentioned that the picture would be more insightful if it was created based on a timeline. This could be a reasonable alternative to consider.
- Diversify the speech-bubble pictures, so that people of different ethnicities, gender and age groups are presented. This point obviously only applies if the decision is made to use this type of images. While special care has been taken to choose an equal number of women and men of different age groups, it seems equally important to include people of different ethnicities. One participant mentioned that the women who are displayed in the speech-bubble pictures are too young and sexy for an average student at FernUniversität in Hagen. Research has shown that students who feel integrated perform better in tests [78],[93]. Older students might feel better integrated into the university culture if the speech-bubble images also display people of their age.
- Using the Jupyter widget 'interact' to display the interactive graphics about Amdahl's and Gustafson's law. This will allow the reader to dynamically adjust the input values of the graphics on a slide bar. Figure 24 displays an example usage of this widget.



```
In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
                σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0))
```

```
   angle  308.2
max_time  12
       σ  10
       β  2.6
       ρ  28
```

FIGURE 24: DEMONSTRATION OF THE JUPYTER WIDGET INTERACT [1]

- Creating a printable summary of the introduced theory and used mathematical formulas. One drawback of using Jupyter Notebook for educational instructions is the missing navigation menu within notebooks. Further, the interactive components, such as the code sections, make the presentation of theory less compact. Providing a handout which summarizes theory and formulas will make it easier for the learners to keep an overview.
- Creating exercises which give the students specific tasks related to the explorative content of the notebooks. This will ensure that all learners spend some time testing different benchmarks for performance.

In addition, there are also some suggestions which we feel should be considered depending on the circumstances:

- Using less or no speech-bubble pictures. It will be up to the discretion of whomever will use the notebooks to decide about the suitability of these images. In the previous section we proposed a trade-off between professional appearance and educational advantage. Whether speech-bubble pictures should be used depends on how much the presenter values a professional appearance, and on the opinion about the pedagogical advantage of these pictures.
- Using a more formal language style. Here too we proposed a trade-off between professional appearance and educational advantage. Depending on how an institution wants to present itself it might be inadequate to use informal language. Research results, however, suggest that a conversational and personalized language style is more suitable for learning [5, p. 179].

- Using different or additional benchmarks. One participant has suggested to use more complex data structures for the performance tests. Which benchmarks are most suitable to demonstrate the effect of parallelization on performance will depend on the targeted audience. In a course for graduate students more complex programs might be suitable, whereas undergraduate students might feel overwhelmed if they are confronted with complex algorithms they have never seen before.
- Making the notebook runnable locally. The notebooks were designed to be used through the JupyterHub platform, which comes with a few advantages that are lost if the notebooks are run locally. Using JupyterHub ensures that all participants access the same computational resources, which means that all participants get comparable performance results when running the benchmarks. This is important if students want to compare and discuss the achieved performance of certain programs with each other. Also, the textual instructions of the notebooks were written referring to the performance achievable on the JupyterHub server. Another point to consider is that most standard local machines offer 8 threads, whereas the students can get 64 threads if they run code on the server located at FernUniversität in Hagen. Thus, we suggest that it makes more sense for the students to access the notebooks through JupyterHub. In theory, the notebooks can be adjusted so that they are runnable on a local system. This could be advantageous for students who do not have continuous access to the internet.
- Investigating about alternative possibilities to integrate self-test questions into the notebooks. Two participants mentioned that they did not like that they had to first run the code cell before being able to submit an answer. There might be a better solution that we have not thought about yet. Using the extension 'nbgrader' could be an alternative, especially if the instructor wishes to collect the provided answers for grading. Nbgrader also allows for auto-grading, making the achieved grades of the students visible to the instructor. We currently do not know if nbgrader simply returns a 'pass' or 'fail' to the student, or if nbgrader also allows to return dynamic feedback.

7 Conclusion

The aim of this thesis was to test and evaluate the suitability of Jupyter Notebook as medium for teaching. The literature review has revealed that Jupyter Notebook is becoming increasingly popular as an ICT tool in education. This holds especially true with the Covid-19 crisis, where many educational institutions had to find alternative solutions to teaching [3]. Most universities, however, use Jupyter Notebook as a programming environment in lab-sessions and not as the main medium for teaching. Analyzing the features of Jupyter Notebook with respect to its suitability in education has shown that there is no reason why Jupyter Notebook could not also be used as a medium for delivering instructions. Jupyter Notebook's inherent features allow to implement interactive components for learning. These are, for example, interactive graphics, integrated self-test questions with dynamic feedback and code sections which allow to demonstrate, explore, or discover theoretical concepts.

The Jupyter notebooks that we have developed implement all these components. In addition, we also applied some pedagogical methods which are traditionally not used in university courses. These are a personalized and conversational language style as well as speech-bubble images. The evaluation of the project has shown that the participants had mixed opinions with respect to the used language style and the speech-bubble pictures. We proposed a trade-off between professional appearance and the learning benefit that these methods enable. The pedagogical advantages of a conversational and personalized language style have been confirmed by research but with respect to the speech-bubble pictures no such research could be identified [5, p. 186], [62]–[64].

The results of the evaluation indicate that the students perceived the code sections and the self-test questions as beneficial for their learning process. Less influence was attributed to the explorative content in the additional notebooks, and very few participants felt that the speech-bubble pictures had a positive influence on learning outcome. These opinions give an indication about the students' attitude towards these methods, but they do not allow to draw conclusions about the influence of these methods on learning. For this, it would be necessary to measure the learning outcome of the participants, and a control group would be needed to compare the results. Ideally, the learning outcome is also measured at different time intervals, which allows to make conclusions about the influence on retainability.

Overall, the participants gave positive feedback about the developed notebooks and Jupyter Notebook in general. Most criticism was either content related, or about the speech-bubble pictures or language style. The keyword most mentioned in the answers about what was liked was 'interactivity'. Jupyter Notebook allows to create a learning experience where students are not just passively exposed to information, but actively involved instead. The results also revealed that almost half of all participants prefer Jupyter Notebook over other more traditional learning mediums such as books or lectures, if they had to choose one medium for learning about performance in parallel programming. The evaluation has made it evident, that different students have different learning preferences. Jupyter Notebook allows to address these varying preferences to some extent, for example by exporting notebooks to pdf format or by embedding videos of lectures into the notebooks.

The results showed that working in the Jupyter Notebook environment was a feasible task for the participants, even for the ones that used Jupyter Notebook for the first time. This is not a surprise, since almost all participants are studying computer science. More research is needed to evaluate Jupyter Notebook's ease of use for students who do not have a background in computer science.

In theory, Jupyter Notebook can be used to teach any subject. That which sets Jupyter Notebook apart from other teaching media, however, is its potential for exploration. Not every university course will benefit equally from the interactivity that Jupyter Notebook enables. The type of courses which can benefit most from the usage of Jupyter Notebook are the ones where it makes sense to implement interactive graphics, simulations, self-test questions with dynamic feedback or integrated coding exercises.

8 Future Works

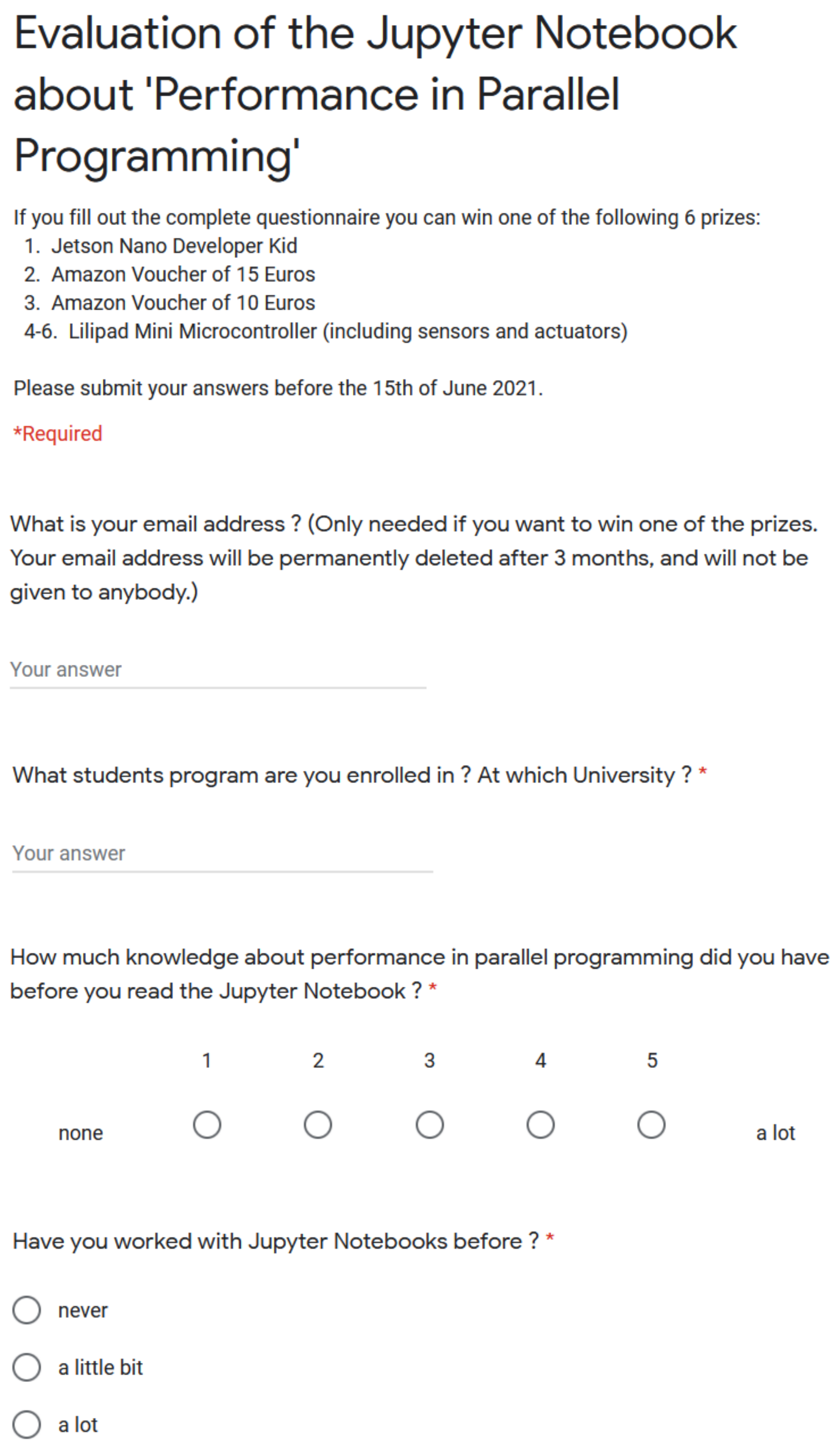
The literature review has revealed that at Lewis & Clark College Portland a set Jupyter notebooks to teach students about high performance computing (HPC) has been developed [40]. We suggest requesting access to these notebooks from the authors, which would allow to compare the notebooks developed at FernUniversität Hagen with the ones developed at Lewis & Clark College Portland. Analyzing how other instructors approached the task of teaching students about HPC with Jupyter Notebook might give valuable insights about how our own notebooks could be further improved.

Currently the developed notebooks only focus on the topic of performance, assuming the students have already learned the basics about parallel programming. It could be an interesting approach to teach students about parallel programming and performance simultaneously. Instead of presenting the learner with prewritten benchmarks, the students can be challenged to write the programs themselves. This would require interleaving the current contents of the notebook with instructions about parallel programming and programming exercises. The advantage of this approach could be that students would feel more motivated to test the performance of programs, since they have written the programs themselves. Further, it would benefit their learning experience in parallel programming because they would be given the knowledge and environment needed to analyze the performance of their parallelized code. Instead of alternating instructions about performance and parallel programming, it could also be an option to have two separate sets of Jupyter notebooks, which link to each other at appropriate sections.

As far as research is concerned, we suggest collaborating with the institute of educational science at FernUniversität in Hagen to conduct further research about the pedagogical advantages of using Jupyter Notebook in education.

A. Appendix

A.1 Questionnaire



Evaluation of the Jupyter Notebook about 'Performance in Parallel Programming'

If you fill out the complete questionnaire you can win one of the following 6 prizes:

1. Jetson Nano Developer Kid
2. Amazon Voucher of 15 Euros
3. Amazon Voucher of 10 Euros

4-6. Lilipad Mini Microcontroller (including sensors and actuators)

Please submit your answers before the 15th of June 2021.

*Required

What is your email address ? (Only needed if you want to win one of the prizes. Your email address will be permanently deleted after 3 months, and will not be given to anybody.)

Your answer

What students program are you enrolled in ? At which University ? *

Your answer

How much knowledge about performance in parallel programming did you have before you read the Jupyter Notebook ? *

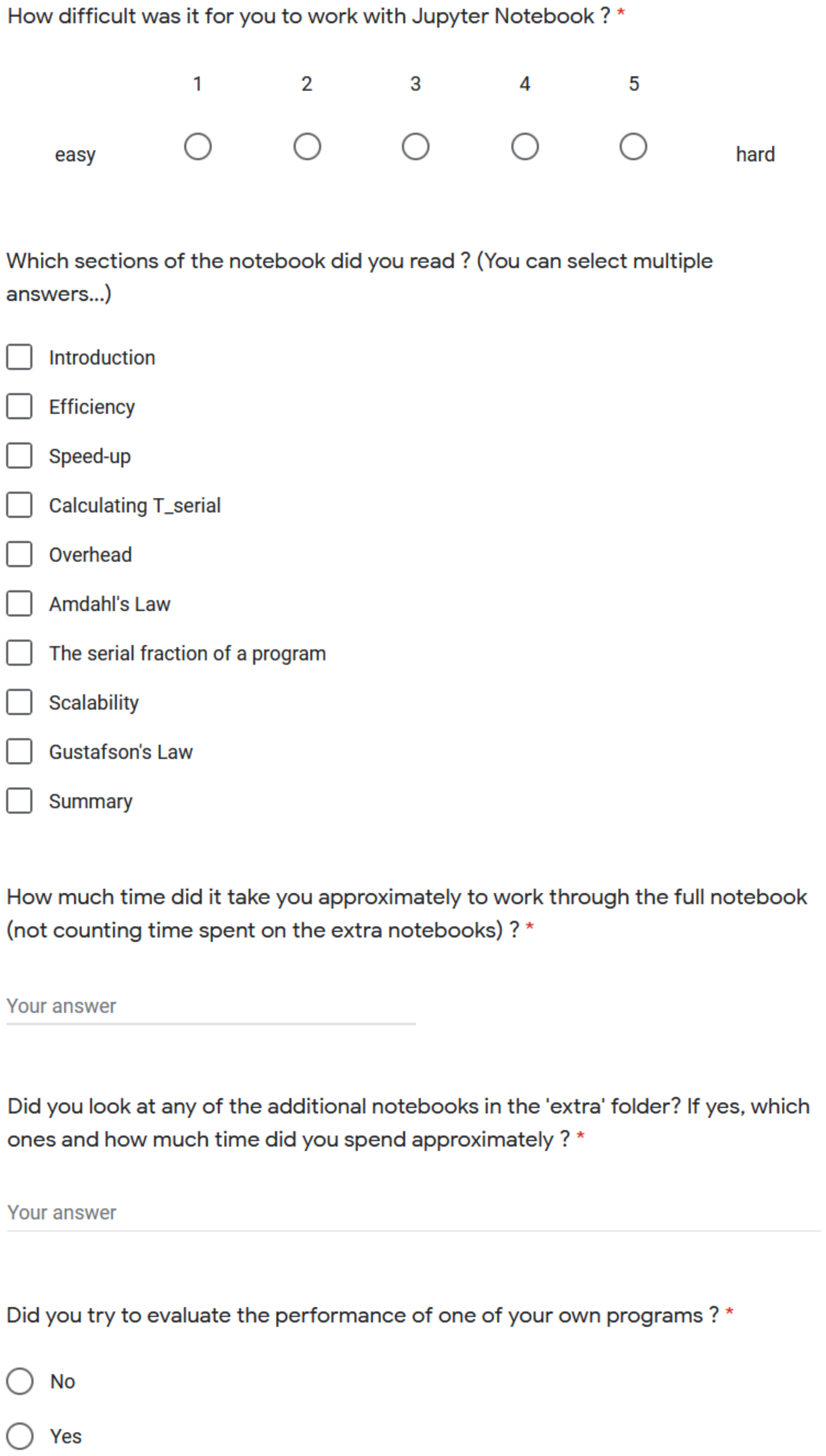
	1	2	3	4	5	
none	○	○	○	○	○	a lot

Have you worked with Jupyter Notebooks before ? *

○ never

○ a little bit

○ a lot



How difficult was it for you to work with Jupyter Notebook ? *

	1	2	3	4	5	
easy	○	○	○	○	○	hard

Which sections of the notebook did you read ? (You can select multiple answers...)

□ Introduction

□ Efficiency

□ Speed-up

□ Calculating T_serial

□ Overhead

□ Amdahl's Law

□ The serial fraction of a program

□ Scalability

□ Gustafson's Law

□ Summary

How much time did it take you approximately to work through the full notebook (not counting time spent on the extra notebooks) ? *

Your answer

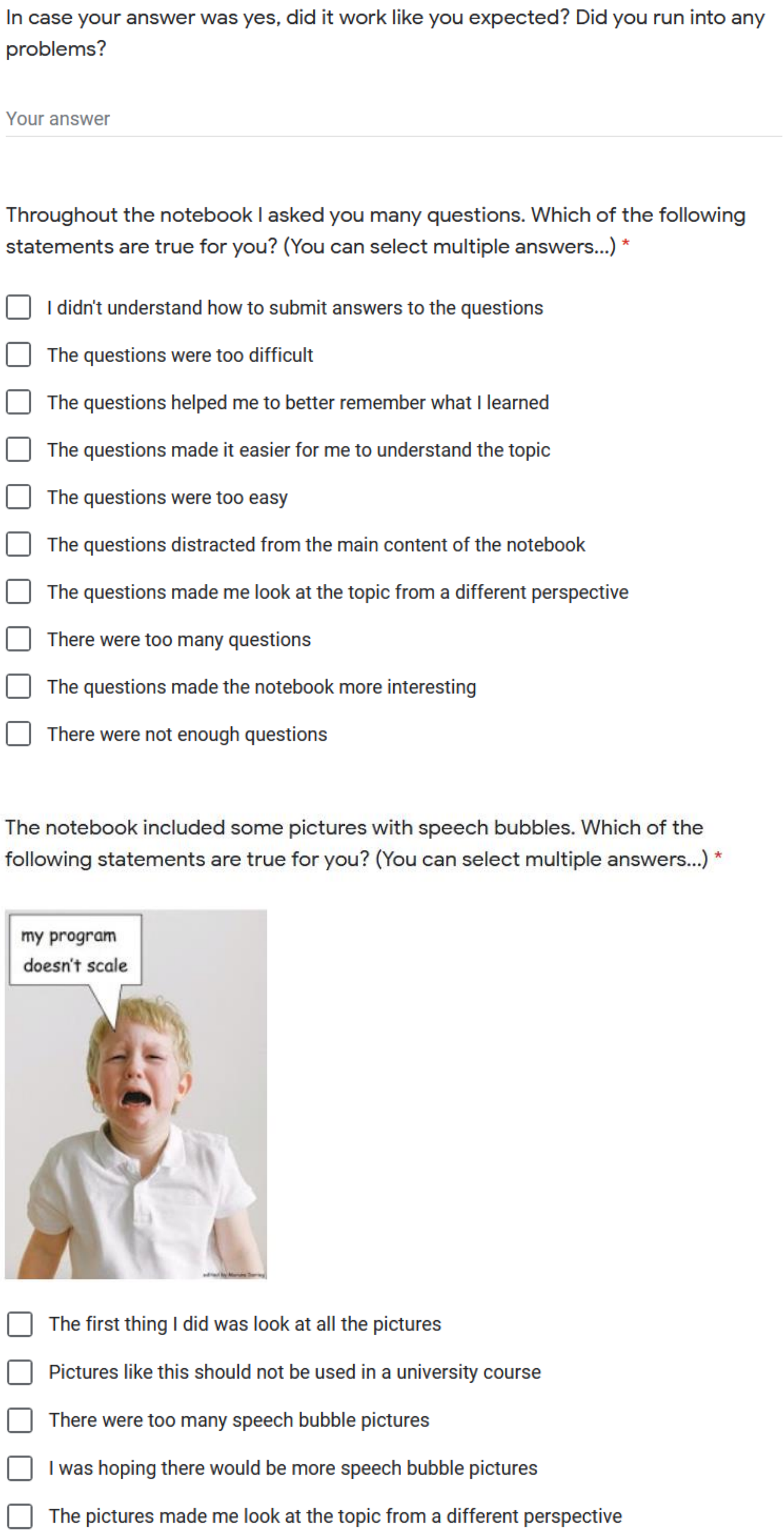
Did you look at any of the additional notebooks in the 'extra' folder? If yes, which ones and how much time did you spend approximately ? *

Your answer

Did you try to evaluate the performance of one of your own programs ? *

○ No

○ Yes



In case your answer was yes, did it work like you expected? Did you run into any problems?

Your answer

Throughout the notebook I asked you many questions. Which of the following statements are true for you? (You can select multiple answers...) *

□ I didn't understand how to submit answers to the questions

□ The questions were too difficult

□ The questions helped me to better remember what I learned

□ The questions made it easier for me to understand the topic

□ The questions were too easy

□ The questions distracted from the main content of the notebook

□ The questions made me look at the topic from a different perspective

□ There were too many questions

□ The questions made the notebook more interesting

□ There were not enough questions

The notebook included some pictures with speech bubbles. Which of the following statements are true for you? (You can select multiple answers...) *

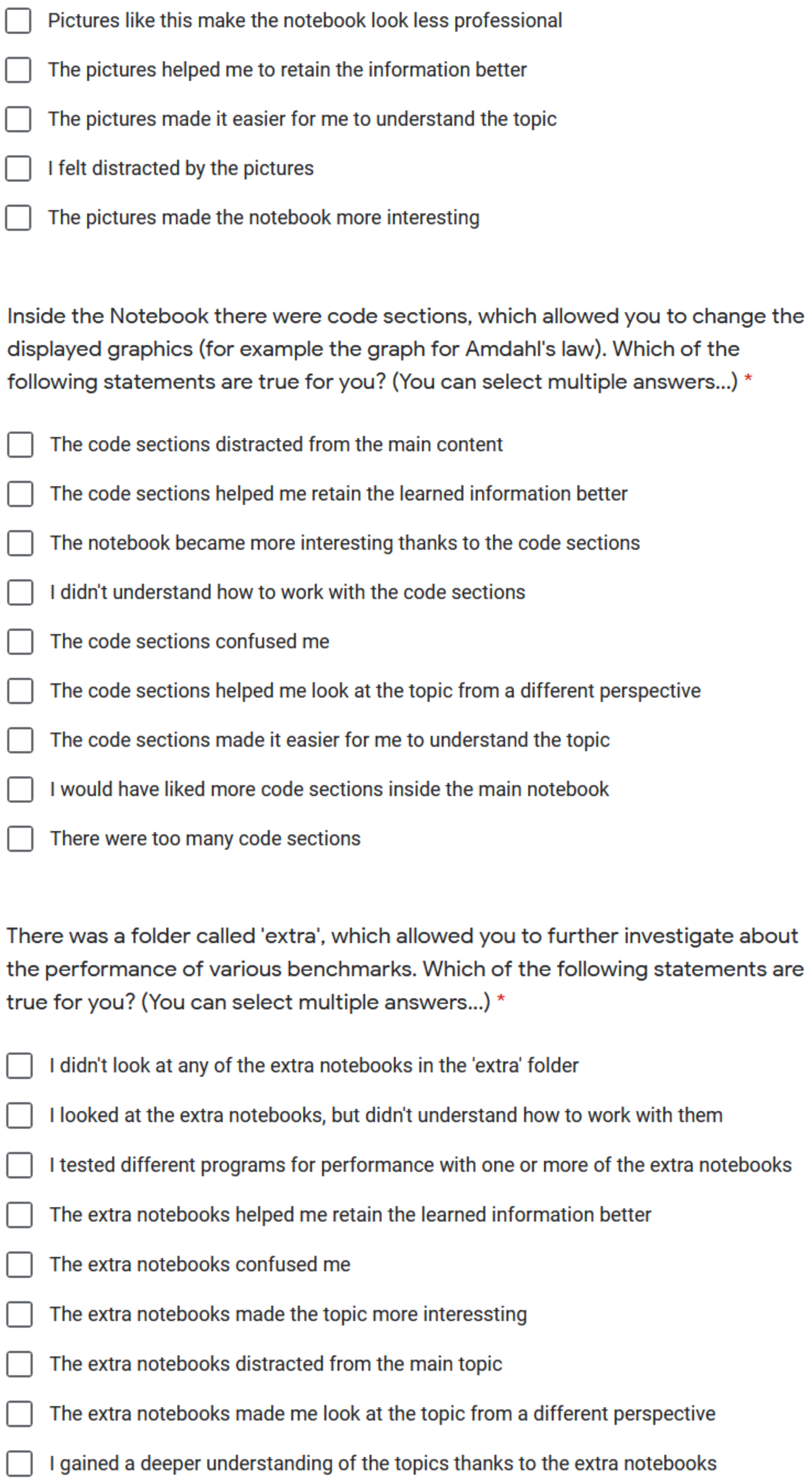
□ The first thing I did was look at all the pictures

□ Pictures like this should not be used in a university course

□ There were too many speech bubble pictures

□ I was hoping there would be more speech bubble pictures

□ The pictures made me look at the topic from a different perspective



□ Pictures like this make the notebook look less professional

□ The pictures helped me to retain the information better

□ The pictures made it easier for me to understand the topic

□ I felt distracted by the pictures

□ The pictures made the notebook more interesting

Inside the Notebook there were code sections, which allowed you to change the displayed graphics (for example the graph for Amdahl's law). Which of the following statements are true for you? (You can select multiple answers...) *

□ The code sections distracted from the main content

□ The code sections helped me retain the learned information better

□ The notebook became more interesting thanks to the code sections

□ I didn't understand how to work with the code sections

□ The code sections confused me

□ The code sections helped me look at the topic from a different perspective

□ The code sections made it easier for me to understand the topic

□ I would have liked more code sections inside the main notebook

□ There were too many code sections

There was a folder called 'extra', which allowed you to further investigate about the performance of various benchmarks. Which of the following statements are true for you? (You can select multiple answers...) *

□ I didn't look at any of the extra notebooks in the 'extra' folder

□ I looked at the extra notebooks, but didn't understand how to work with them

□ I tested different programs for performance with one or more of the extra notebooks

□ The extra notebooks helped me retain the learned information better

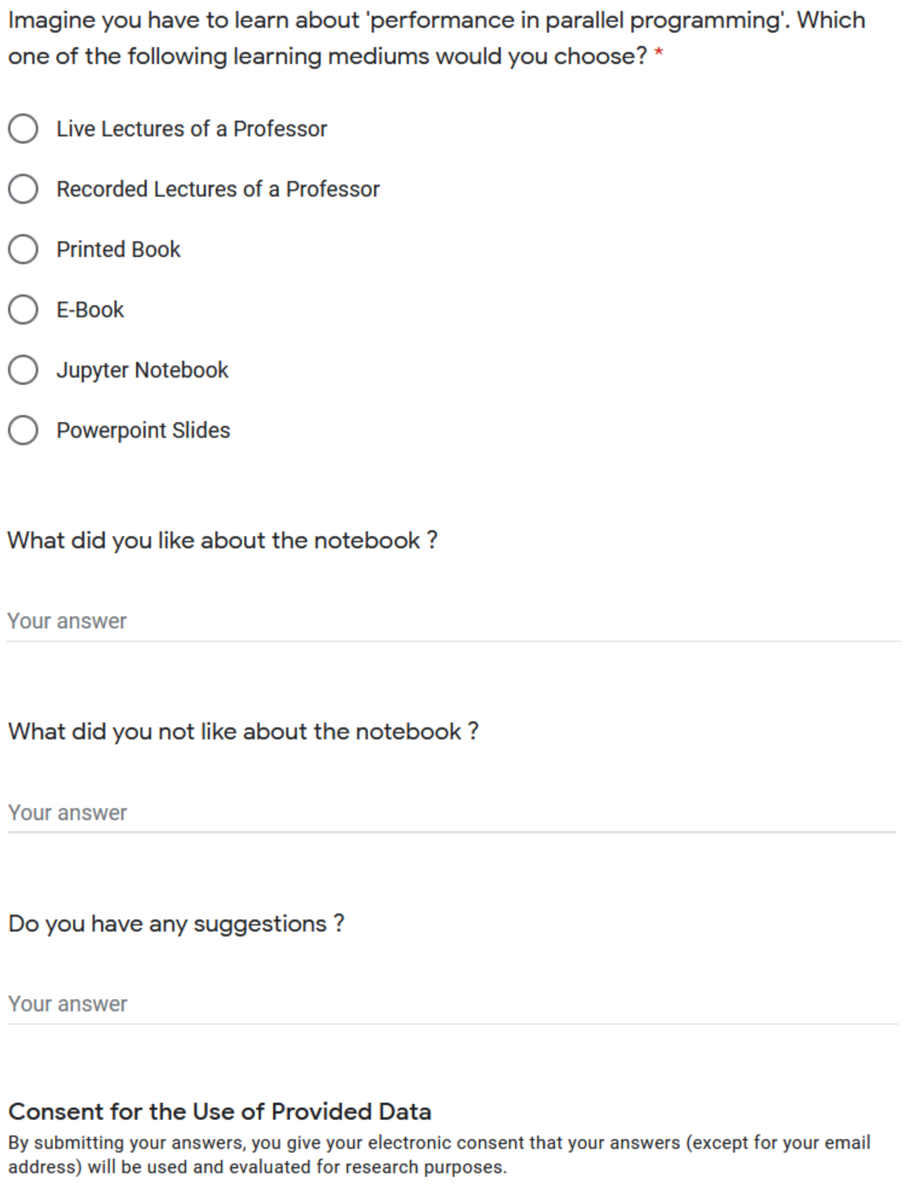
□ The extra notebooks confused me

□ The extra notebooks made the topic more interessting

□ The extra notebooks distracted from the main topic

□ The extra notebooks made me look at the topic from a different perspective

□ I gained a deeper understanding of the topics thanks to the extra notebooks



Imagine you have to learn about 'performance in parallel programming'. Which one of the following learning mediums would you choose? *

○ Live Lectures of a Professor

○ Recorded Lectures of a Professor

○ Printed Book

○ E-Book

○ Jupyter Notebook

○ Powerpoint Slides

What did you like about the notebook ?

Your answer

What did you not like about the notebook ?

Your answer

Do you have any suggestions ?

Your answer

Consent for the Use of Provided Data

By submitting your answers, you give your electronic consent that your answers (except for your email address) will be used and evaluated for research purposes.

A.2 Dimensions that the Statements about the Pedagogical Methods Measure

Dimension	Sub-dimension	Statement	Pedagogical Method	Phrasing
Usage	Usage of Additional Notebooks	I didn't look at any of the extra notebooks in the 'extra' folder	Extra Notebook	Negative
		I tested different programs for performance with one or more of the extra notebooks	Extra Notebook	Positive
Ease of Use	Ability to Work with Jupyter Notebook	I didn't understand how to submit answers to the questions	Questions	Negative
		I didn't understand how to work with the code sections	Code Sections	Negative
		I looked at the extra notebooks, but didn't understand how to work with them	Extra Notebooks	Negative
	Transparency	The code sections confused me	Code Sections	Negative
		The extra notebooks confused me	Extra Notebooks	Negative
Opinion about Quality	Frequency	There were too many questions	Questions	Negative
		There were not enough questions	Questions	Negative
		There were too many speech-bubble pictures	Speech-Bubble Pictures	Negative
		I was hoping there would be more speech-bubble pictures	Speech-Bubble Pictures	Negative
		There were too many code sections	Code Sections	Negative
		I would have liked more code sections inside the main notebook	Code Sections	Negative
	Level of Difficulty	The questions were too difficult	Questions	Negative
		The questions were too easy	Questions	Positive
	Professional Appearance	Pictures like this make the notebook look less professional	Speech-Bubble Pictures	Negative
	Adequacy	Pictures like this should not be used in a university course	Speech-Bubble Pictures	Negative
Opinion about the Ability to Foster Learning	Retainability	The questions helped me to better remember what I learned	Questions	Positive
		The pictures helped me to retain the information better	Speech-Bubble Pictures	Positive
		The code sections helped me retain the learned information better	Code Sections	Positive
		The extra notebooks helped me retain the learned information better	Extra Notebooks	Positive
	Understand-ability	The questions made it easier for me to understand the topic	Questions	Positive
		The pictures made it easier for me to understand the topic	Speech-Bubble Pictures	Positive
		The code sections made it easier for me to understand the topic	Code Sections	Positive
		I gained a deeper understanding of the topics thanks to the extra notebooks	Extra Notebooks	Positive
	Distraction	The questions distracted from the main content of the notebook	Questions	Negative
		I felt distracted by the pictures	Speech-Bubble Pictures	Negative

		The code sections distracted from the main content	Code Sections	Negative
		The extra notebooks distracted from the main topic	Extra Notebooks	Negative
	Different Perspective	The questions made me look at the topic from a different perspective	Questions	Positive
		The pictures made me look at the topic from a different perspective	Speech-Bubble Pictures	Positive
		The code sections helped me look at the topic from a different perspective	Code Sections	Positive
		The extra notebooks made me look at the topic from a different perspective	Extra notebooks	Positive
	Engagement	The questions made the notebook more interesting	Questions	Positive
		The pictures made the notebook more interesting	Speech-Bubble Pictures	Positive
		The first thing I did was look at all the pictures	Speech-Bubble Pictures	Positive
		The notebook became more interesting thanks to the code sections	Code Sections	Positive
		The extra notebooks made the topic more interesting	Extra Notebooks	Positive

Table 9: Dimensions that the Statements about the Pedagogical Methods Measure

A.3 The Participants' Answers to the Questionnaire

Question 1: What student's program are you enrolled in? At which University?

Participant	Answer
1	Master Computer Science, Fernuniversitaet Hagen
2	M. Sc. Informatik, FernUniversität in Hagen
3	B.S. Informatik
4	Informatik Bachelor, Fernuniversität in Hagen
5	Mathmatics B.Sc., Fernuniversität Hagen
6	Bachelor Informatik, Fernuni Hagen
7	Informatik, Fernuni Hagen
8	Bachelor (Informatik)
9	Angewandte Mathematik und Informatik, FH Aachen
10	B.Sc. Informatik, FernUniversität Hagen
11	Mathematics Bsc. at FernUniversität in Hagen
12	Bachelor Informatik, Fernuni Hagen
13	Computer Science, FUni Hagen
14	BSc M/CS/MATSE Hagen
15	Fernuni Hagen
16	B.Sc.Informatik Fernuni Hagen
17	Fernuniversität Hagen, Akademiestudium (plus TU Dresden, Maschinenbaudiplom)
18	B. Sc. Informatics Fernuniversität in Hagen Paralell Programming
19	Ph.D. student, FernUniversität in Hagen
20	Computer Science (Master) FernUni Hagen
21	Computer science bachelor, Fernuniversität Hagen
22	Fernuni Hagen, Bachelor Informatik
23	B.Sc. Informatik, Fernuni Hagen
24	Bachelor Informatik, FernUniversität in Hagen
25	Akademiestudent Informatik - Feruniversität in Hagen
26	None, studied CS, RWTH aachen
27	M.Sc. Computer Science, FU Hagen
28	PhD program, FernUniversität in Hagen

Question 2: How much knowledge about performance in parallel programming did you have before you read the Jupyter Notebook?

Coding (Likert scaling): 1=none; 5=a lot

Participant	Answer
1	4
2	5
3	4
4	4
5	2
6	2
7	1

Participant	Answer
8	1
9	2
10	1
11	1
12	3
13	1
14	4

Participant	Answer
15	1
16	1
17	2
18	4
19	4
20	3
21	2

Participant	Answer
22	2
23	1
24	1
25	1
26	2
27	1
28	3

Question 3: Have you worked with Jupyter Notebooks before?

Coding: 1=never; 2=a little bit; 3=a lot

Participant	Answer	Participant	Answer	Participant	Answer	Participant	Answer
1	2	8	1	15	1	22	2
2	2	9	3	16	1	23	2
3	2	10	2	17	3	24	1
4	3	11	1	18	2	25	1
5	1	12	1	19	2	26	1
6	2	13	2	20	2	27	1
7	1	14	2	21	2	28	2

Question 4: How difficult was it for you to work with Jupyter Notebook?

Coding (Likert scaling): 1=easy; 5=hard

Participant	Answer	Participant	Answer	Participant	Answer	Participant	Answer
1	1	8	2	15	2	22	2
2	2	9	1	16	3	23	2
3	1	10	1	17	1	24	2
4	1	11	3	18	3	25	3
5	3	12	1	19	1	26	2
6	1	13	2	20	1	27	1
7	5	14	1	21	2	28	1

Question 5: Which sections of the notebook did you read? (You can select multiple answers...)

Coding: 1=Introduction; 2=Efficiency; 3=Speed-up; 4=Calculating T_serial; 5=Overhead; 6=Amdahl's law; 7=The serial fraction of a program; 8=Scalability; 9=Gustafson's law; 10=Summary

Participant	Answer	Participant	Answer
1	1;2;3;4;5;6;7;8;9;10	15	1;2;3;4;10
2	1;2;3;4;5;6;7;8;9;10	16	1;2;3;4;5;6;7;8;10
3	1;2;3;4;5;6;7;8;9;10	17	1;2;3;4;5;6;7;8;9;10
4	1;2;3;4;5;6;7;8;9;10	18	1;2;3;4;5;6;7;8;9;10
5	1;3;4;5;7;8;10	19	1;2;3;4;5;6;7;8;9;10
6	1;2;3;4;5;6;7;8;9;10	20	1;2;3;4;5;6;7;8;9;10
7	1;2;3;4;5;6;10	21	1;2;3;6;7;8;10
8	1;2;3;4;5	22	1;2;3;4;5;6;7;8;9;10
9	1;2;3;4;5;6;7;8;9;10	23	1;2;3;4;5;6;7;8;10
10	1;2;3;4;5;6;7;8	24	1;2;3;4;5;6;7;8;9;10
11	1;2;3;4;5;6;7;8;9;10	25	1;2;3;4;5;6;7;8
12	1;2;3;4;5;6;7;10	26	1;2;3;4;5;6;7;8;9;10
13	1;2;3;4;5;6;7;8;9;10	27	1;2;3;4;5;6;7;8;9;10
14	1;2;3;4;5;6;7;8;9;10	28	1;2;3;4;5;6;7;8;9;10

Question 6: How much time did it take you approximately to work through the full notebook (not counting time spent on the extra notebooks)?

Participant	Answer
1	30 minutes
2	1.5h
3	30 min
4	~2 stunden
5	3 hours
6	1h
7	zwei Stunden
8	3
9	ca. 1,5 h
10	2 hours
11	3 hours
12	45 Minuten
13	2hrs
14	2h

Participant	Answer
15	1,5 h
16	1-2 hours
17	3 hours? not easy to say, I did not read it in one go and did other things in between
18	6h
19	maybe 2h (without proofreading)
20	3 days
21	120 min
22	30 min
23	2h
24	30-35 minutes
25	50 minutes
26	60min
27	80min
28	30 minutes

Question 7: Did you look at any of the additional notebooks in the 'extra' folder? If yes, which ones and how much time did you spend approximately?

Partici-pant	Answer
1	10 Minuten
2	Measuring execution time: 0.5h, rest: superficial
3	all of them, approx. 5-10 min. each
4	Momentan noch geöffnet (evtl h: Measuring_Execution_Time, measuring_serial_time, overhead) Insgesamt 10-20min
5	0,5 hours
6	10 Minutes
7	Nein
8	Measuring Execution time, 15 min
9	'Measuring_Execution_Time' -ca. 20min, 'Overhead'-ca. 10 min, 'compare_timemeasuring_Mathodes'-ca. 10min
10	no
11	No
12	Measuring Execution Time, 20 Minuten
13	no
14	no

Partici-pant	Answer
5	Yes, speed-up and measusring execution time. 20 Minutes for bothes
16	no
17	Measuring Execution time: ca. 20 min Weak Scaling: ca. 10 min
18	only peak looks in overhead and weak scaling
19	all, maybe 1h
20	Measuring Execution Time (30 min), Measuring tserial(15 min), Overhead(15 min), Weak Scaling(60 min), Compare Time Measuring Methods(120 min)
21	yes, 20 min
22	Yes, comparision of time measuring methods, approx. 30 mins
23	Measuring Execution Time, ca. 0.5h
24	Not yet, maybe later on.
25	Measuring Execution Time (10 min)
26	no
27	just a few minutes, getting an overview
28	none

Question 8: Did you try to evaluate the performance of one of your own programs?

Coding: 0=no; 1=yes

Participant	Answer
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	1
19	0
20	1
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0

Question 9: In case your answer was yes, did it work like you expected? Did you run into any problems?

Participant	Answer
18	The most problems are to small to really see the Speedups... On a Raspberry Pi its much better ;)
20	sometimes it is not so easy to explain the results

Question 10: Throughout the notebook I asked you many questions. Which of the following statements are true for you? (You can select multiple answers...)

Coding:
1 = I didn't understand how to submit answers to the questions
2 = The questions were too difficult
3 = The questions helped me to better remember what I learned
4 = The questions made it easier for me to understand the topic
5 = The questions were too easy
6 = The questions distracted from the main content of the notebook
7 = The questions made me look at the topic from a different perspective
8 = There were too many questions
9 = The questions made the notebook more interesting
10 = There were not enough questions

Participant	Answer
1	3;4;5;10
2	5;7
3	3;9
4	5;9
5	3;9
6	3;4;7;9
7	3;4;7;9
8	4;5;9;10
9	4;7
10	3;4;9
11	3;9
12	3;7
13	3;4
14	3;4;9
15	4;9
16	3;4;9
17	4;9
18	5;9
19	3;4;9
20	3
21	3;4;9
22	4;7;9
23	7;9
24	3;4;7;9;10
25	4;9
26	3;4;7;9
27	3;4;9
28	9

Question 11: The notebook included some pictures with speech-bubbles. Which of the following statements are true for you? (You can select multiple answers...)

1 = The first thing I did was look at all the pictures
2 = Pictures like this should not be used in a university course
3 = There were too many speech-bubble pictures
4 = I was hoping there would be more speech-bubble pictures
5 = The pictures made me look at the topic from a different perspective
6 = Pictures like this make the notebook look less professional
7 = The pictures helped me to retain the information better
8 = The pictures made it easier for me to understand the topic
9 = I felt distracted by the pictures
10 = The pictures made the notebook more interesting

Participant	Answer
1	2;3;9
2	1;2;3;6;9
3	3;6
4	6;10
5	1;3
6	7;10
7	3;5;7;8;10
8	10
9	1;5;10
10	3;9
11	9
12	1;4;7;10
13	7
14	10
15	5;7;8;10
16	6
17	10
18	10
19	1;2;6;10
20	1
21	3;6;9
22	10
23	6
24	1;10
25	4;10
26	10
27	10
28	3;6

Question 12: Inside the Notebook there were code sections, which allowed you to change the displayed graphics (for example the graph for Amdahl's law). Which of the following statements are true for you? (You can select multiple answers...)

Coding:
1 = The code sections distracted from the main content
2 = The code sections helped me retain the learned information better
3 = The notebook became more interesting thanks to the code sections
4 = I didn't understand how to work with the code sections
5 = The code sections confused me
6 = The code sections helped me look at the topic from a different perspective
7 = The code sections made it easier for me to understand the topic
8 = I would have liked more code sections inside the main notebook
9 = There were too many code sections

Participant	Answer
1	3
2	3;7
3	3
4	3
5	2;4
6	4
7	2;3;6;7
8	2;3;7
9	2;3;6;7;8
10	3
11	2;3;6;7
12	2;3;7
13	2;3
14	3
15	2;3
16	2;3;7
17	3;7
18	2;3;7
19	2;3;7
20	7;8
21	2;3;7;8
22	2
23	3;8
24	2;3
25	5
26	2;3;6;7
27	3
28	3

Question 13: There was a folder called 'extra', which allowed you to further investigate about the performance of various benchmarks. Which of the following statements are true for you? (You can select multiple answers...)

Coding:
1 = I didn't look at any of the extra notebooks in the 'extra' folder
2 = I looked at the extra notebooks, but didn't understand how to work with them
3 = I tested different programs for performance with one or more of the extra notebooks
4 = The extra notebooks helped me retain the learned information better
5 = The extra notebooks confused me
6 = The extra notebooks made the topic more interesting
7 = The extra notebooks distracted from the main topic
8 = The extra notebooks made me look at the topic from a different perspective
9 = I gained a deeper understanding of the topics thanks to the extra notebooks

Participant	Answer
1	6;9
2	6;9
3	4;6
4	7;9
5	2;6
6	3
7	1

Participant	Answer
8	6;9
9	3;7;8
10	1
11	4
12	4;6;9
13	1
14	1

Participant	Answer
15	6
16	1
17	4;8;9
18	1
19	3;4;9
20	3;9
21	8;9

Participant	Answer
22	3
23	8
24	1
25	5
26	1
27	3
28	1

Question 14: Imagine you have to learn about 'performance in parallel programming'. Which one of the following learning mediums would you choose?

Coding:
1 = Live Lectures of a Professor
2 = Recorded Lectures of a Professor
3 = Printed Book
4 = E-Book
5 = Jupyter Notebook
6 = PowerPoint Slides
7 = Unknown

Participant	Answer
1	4
2	4
3	3
4	5
5	2
6	2
7	2

Participant	Answer
8	1
9	5
10	5
11	5
12	2
13	5
14	3

Participant	Answer
15	5
16	5
17	5
18	5
19	1
20	5
21	2

Participant	Answer
22	2
23	5
24	5
25	2
26	5
27	7
28	3

Question 15: What did you like about the notebook?

Participant	Answer
1	The notebook is a good method to learn interactively and run test programs and visualizations directly in the text sections.
2	Generell halte ich das für eine sehr gute interaktive Einführung zum Thema. Reihenfolge der Themen ist gut gewählt, Strukturierung in einem adäquaten Maß.
3	Nice introduction to some central terms. The dynamically adjustable graphics help with strengthening intuitive understanding. The "test question" schematic is a good starting point for developing a deeper user interaction.
4	* die Bilder und Graphen * Die Möglichkeit den Code/Graphen zu ändern, und die Änderungen sofort sehen zu können * Beispiele
5	The clarity and structure is very well done. The text sections with the answers helped me a lot - I think that this very important for beginners. Furthermore the questions put me in an active role and from my point of view that is crucial when you want to learn programming.
6	I liked the pictures. They made me smile. Paralleization was well explained, I especially liked the sand example
7	der Text war interessant geschrieben, z. Bsp. der Vergleich mit dem Strand bauen mit Freunden
8	Es ist interaktiver und man hat keine nervigen Medienbrüche, zum Beispiel zwischen Skript und Ausführung/Schreiben von Programmen
9	Der rote Faden, die leicht verständliche Einarbeitung in das Thema und generell die freundliche und motivierende Art. Zudem finde ich die Idee mit den Fragefeldern im Notebook sehr gelungen!
12	Viel input auf sehr schöne, leicht verständliche und anschauliche Weise erklärt. Die Bilder, Grafiken und Code Sections lockern den Text etwas auf und machen das Bearbeiten interessanter. Auch die optionalen extra Seiten regen an sein Wissen zu vertiefen, ohne direkt von "zu viel Text" erschlagen zu werden.
14	Interactivity
17	The interactivity and the tone
18	The possibility to play with the code.
19	Understandable and in-depth explanation of the topics
20	A profound practical explanation, many examples, well visualized results, good structured and interactive representation
21	I liked the compact forwarding of information and the interactivity.
22	It is another approach towards the learning topic. Also the possibility to try out code is very nice.
23	The possibility to try out stuff directly.
24	Unusual way with speech-bubbles. Never used Jupyter before, love the way to interact with the learning material.
25	I liked that it has many questions and graphics. It helped me to understand the topic.
26	Good explanations and examples, interactive components to explore the pheonomenons on real system, questions to test the learned concepts.
27	the casual tone, the right mix of detail and overview
28	detailed and thorough approach

Question 16: What did you not like about the notebook?

Participant	Answer
1	The wording is sometimes not appropiate and too colloquial for a scientific university course.
2	Aufmachung zum Teil unprofessionell, leider sehr viele Schreibfehler, die den Lesefluss behindern.
3	To me, the unrelated pictures and movies ("WHAT") felt distracting, a bit too ADHDy (personal opinion, not to be taken personally though).

4	* Von Quellen aus nicht wieder zurück navigierbar (Workaround: Notebook 2mal öffnen)
5	I am a beginner in programming so I could not test own programs but I am pleased that I could participate and get to know the notebook.
6	In the end it was a bit hard to understand
7	Es sollte besser erklärt werden, wie man ihn benutzt
8	Dass man die Antwortfelder erst anstupsen musste und sie nicht von vornherein vorhanden waren.
9	Die Abhängigkeit der Betriebssystems, ich habe einige Zeit benötigt es unter Windows verwenden zu können, da ich kein Linux System habe.
12	Viele grammatikalische Fehler im englischen Text, die das Lesen erschweren.
14	I assumed I would need 45 min for the time to work through the notebook, alas that was the time estimation for the questionaire.. so it took longer than I planned originally.
17	at one point, LaTeX formulas stopped working, I don't know why
18	A Introduction is missing... how to use the notebook. Sometimes you need to jump through the notebook, maybe splitting the Notebook into chapters would be beneficial.
19	Phrases like: "If your eyes don't start shining when you hear the word 'mathematics'" or if the author writes in first person.
21	The bubble pictures, sorry...
23	Too much theory, not enough practical info. Silly images. The somewhat flippant phrases.
24	I had to do some fixes to run the notebook on my own computer. It's more a problem of Jupyter.
25	I didn't like that I can not highlight any for me important parts of the text. So i felt a little bit lost while reading.
26	the questions should be directly embedded in the notebook, not as a codeblock, but as a feature within the text. but i doubt that this is easily doable in Jupyter.
27	my personal antipathy of mathematics made it hard to understand the amdahl and expecially the gustafson part
28	casual style

Question 17: Do you have any suggestions?

Participant	Answer
1	The idea of using jupyter notebook for an university course is good. A professional look without the pictures with speech-bubbles and a more formal wording would increase the acceptance as an academic teaching material.
2	Die Aufmachung ist zum Teil nicht so gut und strotzt leider vor Schreibfehlern. Auch in den Charts. Das würde ich komplett überarbeiten. Die den Lesefluss "begleitenden" Bilder würde ich weglassen, denn sie sind zwar nett anzusehen, aber oft wirken sie eher kindlich. Erklärende Bilder könnte man noch mal überarbeiten, bspw. das Strand-Beispiel. Die Idee ist gut, aber die Aufmachung kindlich. Als Kursteilnehmer würde ich Bilder erwarten, bei denen ich direkt verstehe, was der Autor mir im Fließtext versucht zu sagen. Das ist hier eher selten der Fall. Man könnte die Reihenfolge einiger Inhalte vertauschen, sodass diese ggf. mehr Sinn machen. Deswegen einige Anmerkungen zu den Abschnitten in der Performance-Darstellung und weiteren Unterdarstellungen. Allgemein / Performance: * Die Bilder mit den Sprechblasen sind zwar nett anzusehen, aber zum Verständnis meist nicht nötig. * Teilweise hindern sie sogar den Lesefluss: "it looks so easy... people don't understand

how complicated it truly is".
* Einheitliche Schreibweise bspw. "Speed Up" vs. "Speed-Up" vs. "Speed up".
* Zu den Sortieralgorithmen in der Einleitung wäre bspw. eine Info interessant, dass sich die untere Schranke für die Sortierung dennoch nicht ändert. Ein Bezug zur Laufzeitkomplexität also.
   * Warum wird nur mit 2500 Elementen sortiert? Interessant wären statt Zahlen bspw. auch komplexere Datenstrukturen, um den Speedup zu sehen.
   * t_serial berechnen könnte auch vor Effizienz kommen, denn es geht hier um die Messung des Speedups.
* Formulierungen sind zum Teil recht platt: "The conclusion basically is that it is ok to just set thread number to 1, if you don't write a scientific paper and just want to explore things."
* Nicht klar, wo jetzt das Vorwissen enden soll. Es wird plötzlich von der Zeit für das Laden der OpenMP Run-time gesprochen. Was ist im Falle von normalen Systemaufrufen für Threading und wenn man OpenMP nicht kennt?
* Typo: "pyhton module timeit"
* Tipps wie "Schwellwert, wann man zwischen dem parallelen und nicht-parallelen Algorithmus umschaltet", sind wertvoll und könnten einfach überlesen werden.
* Allgemein könnte die TeX-Nutzung besser sein. Im Originalkurs war die Darstellung von Formeln auf Powerpoint im Fließtext ungenügend (unleserlich), hier ist es zwar besser, aber ein Bruchstrich oder Operatoren könnten schon mit der entsprechenden Notation gekennzeichnet werden.
   * Im Fließtext könnten Bezeichner wie f auch in TeX-Schreibweise eingebracht werden.

Amdahls Law:
* Bild könnte aussagekräftiger sein, indem man dieses auf Basis eines Zeitstrahls zeichnet. So hilft es eigentlich kaum.
* t_parallel >= Operator für Beispiel: könnte schöner mit \geq geschrieben werden
* Typo im Schaubild für amdahl.ipynb: "Efficinecy"
* "Maximum speed-up is diverging towards 5.5" - sollte nicht "converging" gemeint sein?

Scalabilty:
* Scalabilty != Scaling
* Weak Scalabilty - Weak Scaling: Zahl der Threads wird vermutlich in einer "constant rate" erhöht
* When do we call a program strongly scalable or weakly scalable?: Satz beginnt mit "If when" im Block Quote

Measuring Execution Time:
* Typo: was busy executing our code, including time spent in library fuCPU time
* Eingabe könnte case insensitive sein und in einer Schleife abgefragt werden (Fehler = Cell neu starten).
* Beispiel "Graphical Effect": Im Python-Code könnten die Kommentare in der For-Schleife eingerückt werden, denn die aktuelle Darstellung verwirrt Leser, die ggf. nicht mit Python vertraut sind.
* Histogramm: Y-Achse sollte mit Label beschriftet sein.
* Cache-Beispiel: es könnte erwähnt werden, dass auf einem realen System auch Ausreißer auftauchen können, wenn die Daten nicht in den Cache passen und ein erneuter Cache Load erfolgt.
* Beispiel "largest, average, smallest":
   * Typo "exectution", "exectuion"
   * Sichtweise ist relevant. Würde mich nicht interessieren, wie sich meine Implementierung im schlechtesten Fall verhält, um Optimierungen zu treffen?
* Im darauf folgenden Chart ist auch "execution" falsch geschrieben.
* Im großen und ganzen wird der Unterschied bei den Timern nicht klar. Ist der Timer in Python pro Thread und der Timer in Jupyter über die gesamte Ausführungszeit?
* timeit-Beispiel: "first argument" passt nicht unbedingt. Hier sollte der Parameter explizit

	benamt werden. * Statt "code" für "time.best" würde ich "Kommando" schreiben, denn ersteres lässt denken, dass man etwas programmieren muss. * Typo: "Exection time of algorithm only", "We will talk more about his" -> this * "Compare Elapsed Time" als Titel für den Chart. Titel nicht klar. Elapsed time seit wann?
3	In the 3 friends at the beach and at the house section, you should clarify that you are not helping them but only driving the truck. I myself assumed helping in the first place, which would result in a total time of 4.25 hrs.
4	* speed-up besser speed_up p nicht defninert (Anzahl von parallel processing units?) * Ich würde Effizienz (auch) wie folgt erklären: Wenn effiency nahe 1, dann wird jeder thread "gut ausgelastet" Ich würde effiency als "durchschnittliche Auslastung" beschreiben * Es wird nicht erklärt, warum für scientific Arbeiten thread_cnt=1 nicht genügend sein kann. Man könnte sogar argumentieren, dass thread_cnt=1 besser ist. Denn das das initialisieren der thread-lib (Was wohl für den konstanten Geschwindigkeits-Unterschied verantwortlich ist) ist ja (auf theoretischer Basis) nicht Teil des paralllel Algos. Es ist ein implementations-Detail, und andere threading-Implementatinoen haben diesen Overhead evtl nicht. * Quellen-Angaben könnten wahrscheinlich genauer sein als nur [2] (z.B. [2, S. 44], oder [2, Kapitel XY]) * Limitations of Amdahls law: erster und letzer Punkt sind sehr ähnlich * Das Wort "Gustafson's Law" wird vor Einführung genutzt * test18: 0.2 könnte (zumindest auf den ersten Blick) gerade noch so passen, ist aber bereits ungültig * matrix size 16 = 16x16, oder 4x4? Vielleicht besser "16 Einträge/elements/entries" schreiben * Müssen bei strong scaleability die Anzahl an prozessoren erhöt/verändert werden, oder einfach die Anzahl an workern (-> also auch threads) * Bzgl strong/weak scaleabilty: Es ist mMn intuitiver, wenn man sich effieiency wie "Auslastung" vorstellt, welche (optimaler weise) 1 sein sollte. Wenn man sowohl thread-Anzahl als auch Problemgröße erhöt, ist das overhead Problem weniger gegeben (Es gibt jetzt ja auch mehr zu tun), daher nur "weak" * Verwirrend: * Erst heißt es, dass bei weak-scalability sowohl Problemgrö0e als auch Thread-Anzahl via "constant rate" erhöt werden. Später heißt es, dass "weak & poor scaling" gegeben ist, wenn man problemgröße expoentinell (bzgl Anzahl an Threads) wachsen lassen muss. Aber dann ist es keine "constant rate" mehr, also nicht mehr weak scaling? * Frage: Wenn es weder weak noch strong scaleability gegeben ist, was ist der Algo dann? Gibt es vielleicht noch eine 3te Kategorie? Woher kommen die Begriffe ursprünglich? Und wofür gibt es den Begriff "strong scaling", wenn es in der Praxis sowieso unmöglich ist? Sagt man z.B. "scales strongly with thread count < 128"? (Falls ja, evtl erwähnen) (Ok, wurde für weakly-scaling später gemacht. Evtl sollte man dies jedoch für strong-scaling direkt dort erwähnen, wo gesagt wurde, dass es in der Praxis im Grunde unmöglich ist)

	* "we call 1-f" -> "is 1-f" * "When we increase the problem size we can usually observe that the execution time of the parllelizable part of the program increases much more than the execution time of the inherently serial part of the program" Aber genau das Gegenteil wurde doch im Experiment oben gezeigt. Evtl kurz auf diese Diskrepanz eingehen? (Denn sonst hätte die Effizienz bei MM-Multipilikation doch bei größeren Matrizen steigen müssen?) * Erst werden Î± und Î² als Funktionen eingeführt, dann jedoch als Skalare genutzt. Die Problemgröße wird daher nie explizit als Variable benutzt(?) * vielleicht statt "Gustafson's law does not account for parallel overhead" besser "Gustafson's law assumes parallel overhead is constant" ? * Bitte nicht falsch verstehen: Ich fand das Notebook insgesamt gut, auch wenn hier jetzt deutlich mehr steht als bei "What did you like about the notebook ?". Liegt einfach daran, dass ich mir unklare Dinge (= obiges, meist Kleinigkeiten) sofort notiert habe, aber "positives" einfach "akzeptiert"/hingenommen habe, und keine extra Notiz erstellt habe.
5	Maybe different files and texts for beginners and advanced students could be helpful. Beginners could get to know the basic functions and test their first own programs while advanced students who know these things can proceed further.
6	no
14	Everything was fine. I only noticed few typos.I will redo the notebook (and do the extras) on a weekend when I have more time and am not tired from work.
16	For absolute beginners like me, there should be more information about the jupyter notebook itself. I have still no idea, what the principle of the notebook actually is.
18	1. The notebook is not really portable. %run ~/performance/.topsecret/test1.ipynb is calling on the one system the right notebook on the other one it is not there... the tilde is representing the real home directory. So the Notebook needs to be exactly in the homefolder. something like -------- Pfad=%pwd %run $Pfad/.topsecret/test1.ipynb --------- worked for me nice, but when the cwd changes this is of course failing. suggestion: save the cwd on beginning and restore it at the end of the scripts or maybe there is a possibility to get the path on loadtime and save it in some variable... so, these cd's needs to be removed... ----- import numpy as np %cd ~/performance/programs/bubblesort ! ./timing.o 1 100 2500 >/dev/null time_serial_bubblesort = np.genfromtxt('/tmp/time.txt') ----- 2. Chapter: "Calculating tserial" t_parallel we simply measure the time spent in the parallelized mergesort function "mergesort" should be bubblesort? 3. Bad Question: Question: With 3 friends at the beach and 3 friends at home, what is the minimal amount of hours we need for our project? 3 Friends and what is with one self? he drove and he is now tired ;)

	4. maybe check for "," in the numeric answers and give a warning instead of it's wrong... i would appreciate it 5. The Chapter (What is the serial fraction of a program ?) rised more questions... the formula (1/f)/1 = f is wrong... when you divide something with 1 nothing changes... so 1/f = f is left.. and that is only true for 1... i think you meant 1/(1/f) And the result of the question is in my opinion a little bit to strict... 0.2 should also be accepted... That is the result what we get with the speedup of 5... 6. Above Scalability: Just curios... how did you come from s=1/(f+((1-4)/p)) to f = (p-s)/(s(p-1))?? it seems i am missing a math trick...
19	Less speech-bubble pictures (if any). Do not wirte in first person. Proofreading. Delete the video of the Minions.
20	I would appreciate some highlights in text (ex. with color, bold or italic font)
21	Less bubble pictures :) Keep working on these fine notebooks. Make them runnable on Windows as well! Personally I use Jupyter on Windows and LINUX on different machines and believe more people will use it soon when they get in touch with data science, big data or similar.
22	Maybe find a way to provide the actual notebook directory. If you didnt copy it inside your home folder you're screwed with the questions and code-elements.
23	Some typos: It sound super easy. -> It sounds super easy., In Measuring Execution Time .ipynb: ". nctions. The wall time on the other hand"... -> unclear what nctions means. Diagram: Amdahl's Law - Efficinecy -> Amdahl's Law -Efficency Once, I stumbled over a newly introduced abbreviation where I only found its meaning after thorough reading of the text before and after. This was the case with p in the efficiency formula. p standing for parallel. While I do get the point of making a participants more interesting with images, it seems to me that the images used here were a desperate attempt at making the participants funny. It doesn't seem that they help with the participants or fit in naturally. And while I do like that there is a high number of women represented, especially within the male-dominated IT world, and I suppose that this was intentional, I think the selection of stereotypical young, sexy women in the pictures doesn't really help with representing a diverse crowd or the average (female) student at Fernuniversität Hagen.
24	A small handout with the most important information about the notebook. Like vocabulary, mathematical formulas. Maybe a test with a few questions about all sections.
26	Instead of one long text better use multiple independent chapters / multiple notebooks?
28	proofreading

A.4 Data Cleansing

Question about the Current Study Program

Question 1 asked: *"What student's program are you enrolled in? At which university?".* In general, it makes more sense to ask this type of questions as a closed question, since this would guarantee no missing values. In our case we expected the group of participants to be rather diverse, which is why we opted for an open question instead. This also allows students not to fully disclose this information in case they have anonymity concerns. The downside is, that there are a few missing values for variable D1 and D2, which were all encoded as 'unknown'. Further, there was one student who stated to be enrolled in 3 programs: 'Computer Science', 'Mathematics' and 'Mathematical and Technical Software Development'. At FernUniversität in Hagen the program 'Mathematical and Technical Software Development' consists exclusively of courses offered in the programs 'Computer Science' and 'Mathematics'. Therefore, we have encoded that entry as 'Mathematics and Computer Science'. One participant stated to be studying 'Applied Mathematics and Computer Science' at FH Aachen. This entry too was encoded as 'Mathematics and Computer Science'.

Questions about Time

Question six and seven asked the amount of time spent working on the main notebook, and if applicable on the extra notebooks. Most participants answered these questions by giving a rough estimate in hours or minutes. The given answers have been numerically encoded in minutes. Participant 20 has stated that it took three days to work through the main notebook. Since this information is too vague, this entry had to be eliminated in the quantitative calculations. Some participants answered with a time interval instead of a fixed number, such as '10-20 minutes'. In these cases, the middle value of the given interval was used. For question seven some participants did not give a numerical value but instead said that they 'briefly' looked at the notebooks. In these cases, we encoded the entries with five minutes.

Question About Preferred Medium for Studying the Topic

Question 14 was not included in the first draft of the questionnaire. At the time, when Question 14 was added to the questionnaire, one participant had already submitted its answers. Since that participant has not left an email-address, it was not possible to contact the person about the missing value. Thus, there is one missing value in the data of this question.

Literature

[1] "Official Website of Project Jupyter." https://jupyter.org/ (accessed Jun. 05, 2020).

[2] S. Kumar Basak, M. Wotto, and P. Bélanger, "E-learning, M-learning and D-learning: Conceptual definition and comparative analysis," *E-Learn. Digit. Media*, vol. 15, no. 4, pp. 191–216, Jul. 2018, doi: 10.1177/2042753018785180.

[3] D. Burgos, A. Tlili, and A. Tabacco, *Radical solutions for education in a crisis context: COVID-19 as an opportunity for global learning*. 2021.

[4] E. S. Lima, "Emergency Curriculum For Education During and After the Pandemic," *Cad. CEPAOS*, 2020.

[5] R. C. Clark and R. E. Mayer, *E-learning and the science of instruction: proven guidelines for consumers and designers of multimedia learning*, 4th ed. Hoboken: Wiley, 2016.

[6] R. E. Mayer, *Multimedia Learning*, 2nd ed. Cambridge University Press, 2009.

[7] National Science Foundation, "Survey of Federal Funds for Research and Development Fiscal Years 2018-19. Table 1 and 15." https://ncsesdata.nsf.gov/fedfunds/2018/ (accessed Apr. 25, 2021).

[8] Bundesministerium für Bildung und Forschung, "Ausgaben des Bundes für Wissenschaft, Forschung und Entwicklung nach Förderbereichen und Förderschwerpunkten." https://www.bundesbericht-forschung-innovation.de/_daten_abbild/tab_6/tab_6.pdf (accessed Apr. 25, 2021).

[9] K. Beaver and J. Wright, "The association between county-level IQ and county-level crime rates," *Intelligence*, vol. 39, pp. 22–26, 2011, doi: 10.1016/j.intell.2010.12.002.

[10] H. Rindermann, *Cognitive capitalism: human capital and the wellbeing of nations*. Cambridge, United Kingdom; New York, NY: University Printing House, 2018.

[11] B. Caplan and S. Miller, "Intelligence Makes People Think Like Economists: Evidence from the General Social Survey," *Intelligence*, vol. 38, 2010, doi: 10.1016/j.intell.2010.09.005.

[12] A. Maia, J. Borges, A. Reis, P. Martins, and J. Barroso, "Integration of Technologies in Higher Education: Teachers' Needs and Expectations at UTAD," in *Research on e-Learning and ICT in Education*, T. A. Mikropoulos, Ed. Cham: Springer International Publishing, 2018, pp. 153–166. doi: 10.1007/978-3-319-95059-4_9.

[13] S. Laschou, V. Kollias, and I. Karasavvidis, "How Do Transformational Principals View ICT as a Means for Promoting Educational Innovations? A Descriptive Case Study Focusing on Twenty-First Century Skills," in *Research on e-Learning and ICT in Education*, T. A. Mikropoulos, Ed. Cham: Springer International Publishing, 2018, pp. 43–67. doi: 10.1007/978-3-319-95059-4_3.

[14] M. Zastre, "Jupyter Notebook in CS1: An Experience Report," in *Proceedings of the Western Canadian Conference on Computing Education*, Calgary AB Canada, May 2019, pp. 1–6. doi: 10.1145/3314994.3325072.

[15] J. L. Moore, C. Dickson-Deane, and K. Galyen, "e-Learning, online learning, and distance learning environments: Are they the same?," *Internet High. Educ.*, vol. 14, no. 2, pp. 129–135, Mar. 2011, doi: 10.1016/j.iheduc.2010.10.001.

[16] "Every Student Succeeds Act - Public Law 114-Dec.10,2015." https://www.congress.gov/114/plaws/publ95/PLAW-114publ95.pdf (accessed Jan. 05, 2021).

[17] D. J. Cothran et al., "A Cross-Cultural Investigation of the Use of Teaching Styles," *Res. Q. Exerc. Sport*, vol. 76, no. 2, pp. 193–201, Jun. 2005, doi: 10.1080/02701367.2005.10599280.

[18] U. D. Ehlers, *Qualität im E-Learning aus Lernersicht*. Wiesbaden: VS Verlag für Sozialwissenschaften, 2011. doi: 10.1007/978-3-531-93070-1.

[19] K. J. O'Hara, D. Blank, and J. Marshall, "Computational Notebooks for AI Education," 2015, doi: 10.13140/2.1.2434.5928.

[20] B. E. Granger and F. Perez, "Jupyter: Thinking and Storytelling With Code and Data," *Comput. Sci. Eng.*, vol. 23, no. 2, pp. 7–14, Mar. 2021, doi: 10.1109/MCSE.2021.3059263.

[21] N. Silaparasetty, *Machine learning concepts with Python and the Jupyter Notebook environment: using Tensorflow 2.0*. 2020.

[22] S. Lau, I. Drosos, J. M. Markel, and P. J. Guo, "The Design Space of Computational Notebooks: An Analysis of 60 Systems in Academia and Industry," in *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, Dunedin, New Zealand, Aug. 2020, pp. 1–11. doi: 10.1109/VL/HCC50065.2020.9127201.

[23] J. M. Perkel, "Why Jupyter is data scientists' computational notebook of choice," *Nature*, vol. 563, no. 7729, pp. 145–146, Nov. 2018, doi: 10.1038/d41586-018-07196-1.

[24] C. S. Pillay, "Analyzing biological models and data sets using Jupyter notebooks as an alternate to laboratory-based exercises during COVID-19," *Biochem. Mol. Biol. Educ.*, vol. 48, no. 5, pp. 532–534, Sep. 2020, doi: 10.1002/bmb.21443.

[25] R. Thomas, S. Cholia, K. Mohror, and J. M. Shalf, "Interactive Supercomputing With Jupyter," *Comput. Sci. Eng.*, vol. 23, no. 2, pp. 93–98, Mar. 2021, doi: 10.1109/MCSE.2021.3059037.

[26] C. Hill, *Learning Scientific Programming with Python*, 2nd ed. Cambridge University Press, 2020. doi: 10.1017/9781108778039.

[27] K. Le et al., "PyRosetta Jupyter Notebooks Teach Biomolecular Structure Prediction and Design," ENGINEERING, preprint, Feb. 2020. doi: 10.20944/preprints202002.0097.v1.

[28] "Jupyter Kernels." https://github.com/jupyter/jupyter/wiki/Jupyter-kernels (accessed May 28, 2021).

[29] J. F. Pimentel, L. Murta, V. Braganholo, and J. Freire, "A Large-Scale Study About Quality and Reproducibility of Jupyter Notebooks," in *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, Montreal, QC, Canada, May 2019, pp. 507–517. doi: 10.1109/MSR.2019.00077.

[30] A. Rule et al., "Ten simple rules for writing and sharing computational analyses in Jupyter Notebooks," *PLOS Comput. Biol.*, vol. 15, Jul. 2019, doi: 10.1371/journal.pcbi.1007007.

[31] "Project Jupyter Documentation - Architecture." https://jupyter.readthedocs.io/en/latest/projects/architecture/content-architecture.html (accessed May 29, 2021).

[32] "Project Jupyter Documentation - JupyterHub." https://jupyterhub.readthedocs.io/en/stable/ (accessed Jun. 01, 2021).

[33] "Project Jupyter Documentation - nbgrader." https://nbgrader.readthedocs.io/en/latest/ (accessed May 31, 2021).

[34] "Project Jupyter Documentation - nbgitpuller." https://jupyterhub.github.io/nbgitpuller/ (accessed May 31, 2021).

[35] S. Kilcommins, "Voilà— Everything You Need To Know," 2021.

[36] P. Jupyter et al., "nbgrader: A Tool for Creating and Grading Assignments in the Jupyter Notebook," *J. Open Source Educ.*, vol. 2, no. 11, p. 32, Jan. 2019, doi: 10.21105/jose.00032.

[37] N. I. Nwulu, U. Damisa, and S. L. Gbadamosi, "Students Perception about the Use of Jupyter Notebook in Power Systems Education," *Int. J. Eng. Pedagogy IJEP*, vol. 11, no. 1, p. 78, Jan. 2021, doi: 10.3991/ijep.v11i1.14769.

[38] M. J. Perri, "Online Data Generation in Quantitative Analysis: Excel Spreadsheets and an Online HPLC Simulator Using a Jupyter Notebook on the Chem Compute Web site," *J. Chem. Educ.*, vol. 97, no. 9, pp. 2950–2954, Sep. 2020, doi: 10.1021/acs.jchemed.0c00565.

[39] "ChemCompute." https://chemcompute.org/jupyterhub (accessed May 18, 2021).

[40] B. Glick and J. Mache, "Jupyter Notebooks and User-Friendly HPC Access," in *2018 IEEE/ACM Workshop on Education for High-Performance Computing (EduHPC)*, Dallas, TX, USA, Nov. 2018, pp. 11–20. doi: 10.1109/EduHPC.2018.00005.

[41] B. Golman, "A set of Jupyter notebooks for the analysis of transport phenomena and reaction in porous catalyst pellet," *Comput. Appl. Eng. Educ.*, vol. 27, no. 3, pp. 531–542, May 2019, doi: 10.1002/cae.22095.

[42] A. Suárez, M. A. Alvarez-Feijoo, R. Fernández González, and E. Arce, "Teaching optimization of manufacturing problems via code components of a Jupyter Notebook," *Comput. Appl. Eng. Educ.*, vol. 26, no. 5, pp. 1102–1110, Sep. 2018, doi: 10.1002/cae.21941.

[43] E. J. Menke, "Series of Jupyter Notebooks Using Python for an Analytical Chemistry Course," *J. Chem. Educ.*, vol. 97, no. 10, pp. 3899–3903, Oct. 2020, doi: 10.1021/acs.jchemed.9b01131.

[44] A. Cardoso, J. Leitão, and C. Teixeira, "Using the Jupyter Notebook as a Tool to Support the Teaching and Learning Processes in Engineering Courses," in *The Challenges of the Digital Transformation in Education*, vol. 917, M. E. Auer and T. Tsiatsos, Eds. Cham: Springer International Publishing, 2019, pp. 227–236. doi: 10.1007/978-3-030-11935-5_22.

[45] H. Guerra, L. M. Gomes, and A. Cardoso, "Agile approach to a CS2-based course using the Jupyter notebook in lab classes," in *2019 5th Experiment International Conference (exp.at'19)*, Funchal (Madeira Island), Portugal, Jun. 2019, pp. 177–182. doi: 10.1109/EXPAT.2019.8876536.

[46] M. Kastner and N. Podleschny, "Mit Jupyter Notebooks prüfen," TUHH Universitätsbibliothek, Sep. 2019. doi: 10.15480/882.2435.

[47] M. Kastner, "Prüfungen mit JupyterHub," *INSIGHTS - Lehre und Forschung im digitalen Experimentierfeld der TU Hamburg*, 2019. https://www2.tuhh.de/zll/jupyter-notebooks/ (accessed May 18, 2021).

[48] B. T. M. Wong, *Innovations in open and flexible education*. New York, NY: Springer Berlin Heidelberg, 2018.

[49] "GitHub Webpage of PyRosetta Jupyter Notebooks." https://github.com/RosettaCommons/PyRosetta.notebooks (accessed May 18, 2021).

[50] L. Krischer et al., "seismo-live: An Educational Online Library of Jupyter Notebooks for Seismology," *Seismol. Res. Lett.*, vol. 89, no. 6, pp. 2413–2419, Nov. 2018, doi: 10.1785/0220180167.

[51] "Github Webpage of Seismo-live." https://github.com/krischer/seismo_live (accessed May 18, 2021).

[52] M. Wintjen, *Practical Data Analysis Using Jupyter Notebook*. 2020.

[53] P. S. Pacheco, *An introduction to parallel programming*. Amsterdam : Boston: Morgan Kaufmann, 2011.

[54] *Introduction to parallel computing*. New York, NY: Springer Berlin Heidelberg, 2018.

[55] G. Bengel, C. Baun, M. Kunze, and K.-U. Stucky, *Masterkurs parallele und verteilte Systeme: Grundlagen und Programmierung von Multicore-Prozessoren, Multiprozessoren, Cluster, Grid und Cloud*, 2nd ed. Wiesbaden: Springer Vieweg, 2015.

[56] X. Li, "Scalability: Strong and Weak Scaling," *PDC Blog by PDC Center for High Performance Computing*, 2018. https://www.kth.se/blogs/pdc/2018/11/scalability-strong-and-weak-scaling/ (accessed Jun. 06, 2021).

[57] B. Schmidt, J. Gonzalez-Dominguez, C. Hundt, and M. Schlarb, *Parallel programming: concepts and practice*. 2017.

[58] R. Trobec, B. Slivnik, P. Bulić, and B. Robič, *Introduction to Parallel Computing: From Algorithms to Programming on State-of-the-Art Platforms*. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-98833-7.

[59] J. L. Gustafson, "Reevaluating Amdahl's law," *Commun. ACM*, vol. 31, no. 5, pp. 532–533, May 1988, doi: 10.1145/42411.42415.

[60] M. D. McCool, A. D. Robison, and J. Reinders, *Structured parallel programing: patterns for efficient computation*. Amsterdam: Elsevier, Morgan Kaufmann, 2012.

[61] L. Massaron and A. Boschetti, *Regression analysis with Python: learn the art of regression analysis with Python*. Birmingham Mumbai: Packt Publishing, 2016.

[62] G. Kartal, "Does language matter in multimedia learning? Personalization principle revisited," *J. Educ. Psychol.*, vol. 102, no. 3, pp. 615–624, 2010, doi: 10.1037/a0019345.

[63] R. E. Mayer, S. Fennell, L. Farmer, and J. Campbell, "A Personalization Effect in Multimedia Learning: Students Learn Better When Words Are in Conversational Style Rather Than Formal Style.," *J. Educ. Psychol.*, vol. 96, no. 2, pp. 389–395, 2004, doi: 10.1037/0022-0663.96.2.389.

[64] R. Moreno and R. E. Mayer, "Engaging students in active learning: The case for personalized multimedia messages.," *J. Educ. Psychol.*, vol. 92, no. 4, pp. 724–733, 2000, doi: 10.1037/0022-0663.92.4.724.

[65] "An Interview with Kathy Sierra Co-Author of Head First Design Patterns." https://objectsbydesign.com/books/KathySierra.html (accessed Jun. 20, 2021).

[66] K. Sierra and B. Bates, *Head first Java*, 2nd ed. Sebastopol, CA: O'Reilly, 2005.

[67] E. R. Um, J. L. Plass, E. O. Hayward, and B. D. Homer, "Emotional design in multimedia learning.," *J. Educ. Psychol.*, vol. 104, no. 2, pp. 485–498, May 2012, doi: 10.1037/a0026609.

[68] H. Nittono, M. Fukushima, A. Yano, and H. Moriya, "The Power of Kawaii: Viewing Cute Images Promotes a Careful Behavior and Narrows Attentional Focus," *PLoS ONE*, vol. 7, no. 9, p. e46362, Sep. 2012, doi: 10.1371/journal.pone.0046362.

[69] S. F. Harp and R. E. Mayer, "How seductive details do their damage: A theory of cognitive interest in science learning.," *J. Educ. Psychol.*, vol. 90, no. 3, pp. 414–434, 1998, doi: 10.1037/0022-0663.90.3.414.

[70] L. Bender, A. Renkl, and A. Eitel, "Seductive details do their damage also in longer learning sessions – When the details are perceived as relevant," *J. Comput. Assist. Learn.*, vol. 37, no. 5, pp. 1248–1262, Oct. 2021, doi: 10.1111/jcal.12560.

[71] G. A. Taylor, "Emotional engagement in learning through humor and storytelling," *Pediatr. Radiol.*, vol. 50, no. 10, pp. 1352–1353, Sep. 2020, doi: 10.1007/s00247-020-04729-7.

[72] D. Tamblyn, *Laugh and learn: 95 ways to use humor for more effective teaching and training*. New York: AMACOM, 2003.

[73] A. Masek, S. Hashim, and A. Ismail, "Integration of the humour approach with student's engagement in teaching and learning sessions," *J. Educ. Teach.*, vol. 45, no. 2, pp. 228–233, Mar. 2019, doi: 10.1080/02607476.2018.1548169.

[74] F. Bakar and V. Kumar, "The use of humour in teaching and learning in higher education classrooms: Lecturers' perspectives," *J. Engl. Acad. Purp.*, vol. 40, pp. 15–25, Jul. 2019, doi: 10.1016/j.jeap.2019.04.006.

[75] M. L. Gick and K. J. Holyoak, "Schema induction and analogical transfer," *Cognit. Psychol.*, vol. 15, no. 1, pp. 1–38, Jan. 1983, doi: 10.1016/0010-0285(83)90002-6.

[76] C. C. Bonwell and J. A. Eison, *Active learning: creating excitement in the classroom*. Washington, DC: School of Education and Human Development, George Washington University, 1991.

[77] D. L. Schwartz and J. D. Bransford, "A Time For Telling," *Cogn. Instr.*, vol. 16, no. 4, pp. 475–5223, Dec. 1998, doi: 10.1207/s1532690xci1604_4.

[78] L. Averell and A. Heathcote, "The form of the forgetting curve and the fate of memories," *J. Math. Psychol.*, vol. 55, no. 1, pp. 25–35, Feb. 2011, doi: 10.1016/j.jmp.2010.08.009.

[79] P. Leavy, *Research design: quantitative, qualitative, mixed methods, arts-based, and community-based participatory research approaches*. New York; London: Guilford Press, 2017.

[80] S. L. Jackson, *Research methods and statistics: a critical thinking approach*, 3rd ed. Australia: Heinle Cengage Learning, 2008.

[81] L. Cohen, L. Manion, and K. Morrison, *Research methods in education*, 8th ed. London; New York: Routledge, 2018.

[82] Q. Ma and L. Liu, "The Technology Acceptance Model," *Adv. Top. End User Comput.*, vol. 4, 2017.

[83] J. W. Creswell, *Educational research: planning, conducting, and evaluating quantitative and qualitative research*, 4th ed. Boston: Pearson, 2012.

[84] D. J. Treiman, *Quantitative data analysis: doing social research to test ideas*, 1st ed. San Francisco: Jossey-Bass, 2009.

[85] M. G. Lodico, D. T. Spaulding, and K. H. Voegtle, *Methods in educational research: from theory to practice*, 2nd ed. San Francisco, CA: Jossey-Bass, 2010.

[86] C. Secolsky and D. B. Denison, *Handbook on measurement, assessment, and evaluation in higher education*. New York, NY: Routledge, 2012.

[87] U. Flick, Ed., *The Sage handbook of qualitative data collection*. Los Angeles: Sage Reference, 2018.

[88] D. Kumar, *Methodology of Educational Research And Statistics*. New Delhi: Lovely Professional University, 2003.

[89] S. W. VanderStoep and D. D. Johnston, *Research methods for everyday life: blending qualitative and quantitative approaches*, 1st ed. San Francisco, CA: Jossey-Bass, 2009.

[90] J. Dunlosky, K. A. Rawson, E. J. Marsh, M. J. Nathan, and D. T. Willingham, "Improving Students' Learning With Effective Learning Techniques: Promising Directions From Cognitive and Educational Psychology," *Psychol. Sci. Public Interest*, vol. 14, no. 1, pp. 4–58, Jan. 2013, doi: 10.1177/1529100612453266.

[91] D. L. Schwartz, J. M. Tsang, and K. P. Blair, *The ABCs of how we learn: 26 scientifically proven approaches, how they work, and when to use them*, 1st ed. New York, NY: W.W. Norton & Company, Inc, 2016.

[92] G. M. Walton, G. L. Cohen, D. Cwir, and S. J. Spencer, "Mere belonging: The power of social connections.," *J. Pers. Soc. Psychol.*, vol. 102, no. 3, pp. 513–532, 2012, doi: 10.1037/a0025731.

[93] C. M. Steele and J. Aronson, "Stereotype threat and the intellectual test performance of African Americans.," *J. Pers. Soc. Psychol.*, vol. 69, no. 5, pp. 797–811, 1995, doi: 10.1037/0022-3514.69.5.797.

List of Figures

List of Tables