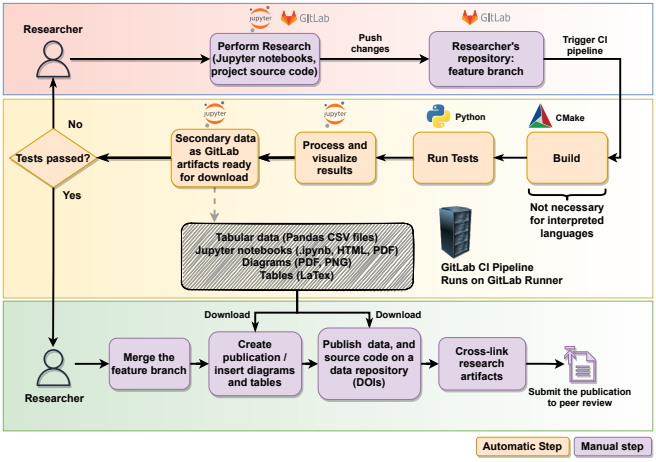
Research Data Management: A Perspective from the CRC 1194

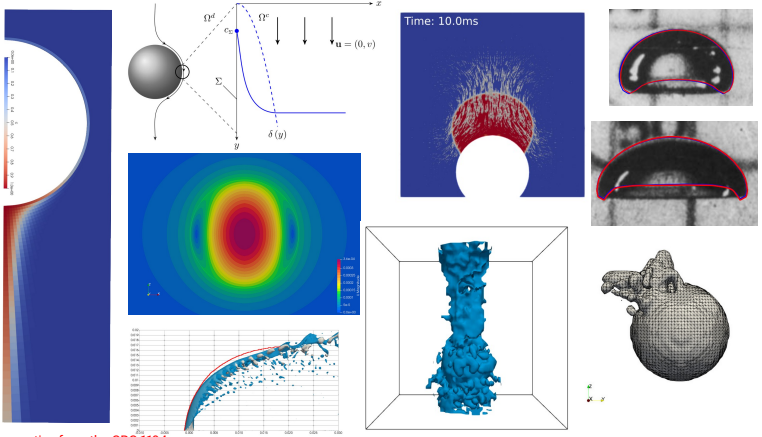
TECHNISCHE UNIVERSITÄT DARMSTADT



Two-phase flow simulation methods (B01, B02, Z-INF)

Validation, verification, and cooperations within the CRC1194

TECHNISCHE UNIVERSITÄT DARMSTADT



Collaborations between simulative and experimental projects

TECHNISCHE UNIVERSITÄT DARMSTADT

Numerical simulation methods require thorough testing:

- Verification against exact solutions.
- Validation with respect to experiments.
- Serial and parallel computational efficiency: statistical performance experiments.

Experiments are adjusted using feedback based on numerical methods' results.

Collaborations within a CRC require data exchange

- Standardized, simple, interoperable secondary data/metadata format.
- Findable Accessible Interoperable and Re-usable (**FAIR**) Research.

FAIR research (data manegement) in a CRC

Boundary and initial conditions

TECHNISCHE UNIVERSITÄT DARMSTADT

- Dedicated resources for handling research data are not available on project-level.
- Research data management concepts live or die with PhD students.
  - Cooperative publications are paramount in a CRC: cooperation overhead.
  - Research data management is checked increasingly critically.
  - Ph.D. candidates just want to graduate: anonymous poll showed 90% target industry positions.
  - Research data management causes some workload overhead.
  - **KISS: practical effective solutions, with very low overhead.**
  - **Research data publication as a Ph.D. defense requirement?**
- Ph.D. students rotate every funding period
  - Little or no overlap between successors and predecessors.
  - **Data processing pipeline automation and documentation.**

Research Data Management Concept of the CRC1194

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Use version control ("git").
   - Pays off when investigating / switching between different research ideas.
2. Automate and document (script) data-processing steps.
   - Pays off after several executions.
3. Implement case studies using literal programming (Jupyter Notebooks).
   - Pays off in collaborations and when writing results sections in publications.
4. Organize case studies to easily retrieve parameter-specific information.
   - Pays off when investigating errors: nothing ever works from the first try!
5. Focus first on secondary data, and use a simple data/metadata format.
   - Low hanging fruit, small data sizes.
6. Use milestones to cross-link digital research artifacts on data/publication repositories.
   - Satisfies the DFG requirement, takes half an hour.
7. Use Continuous Integration for automating research workflows.
   - Pays off during peer-review.

Version control

TECHNISCHE UNIVERSITÄT DARMSTADT

- Management of versions of (usually) text data, like publications and scientific codes.
- Nowadays version control is **essential** for scientific codes of all shapes and sizes.
- Version control can be adapted to needs and constraints of the CRC[1].

[1] Maric, Tomislav, Lehr, Jan-Patrick, Papagiannidis, Ioannis, Lambie, Benjamin, Bothe, Dieter, & Bischof, Christian. (2021, April). A Workflow for Increasing the Quality of Scientific Software (Version 1.0). Zenodo. http://doi.org/10.5281/zenodo.4668439

Git version control system (VCS)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Use Overleaf for papers, Git for everything else.

Version control "enforces" modularity

TECHNISCHE UNIVERSITÄT DARMSTADT

Git conflicts

- A file is changed differently on two branches and a merge is needed.
- Two team members edit the same file at once.

Modularity reduces conflicts and speeds up teamwork

- Separate implementation into separate version-controlled files.

Version control and Clean Code

TECHNISCHE UNIVERSITÄT DARMSTADT

- University research teams are small (1 - few members).
- ▸ Separation of Concerns (SC) and ▸ Single Responsibility Principle (SRP) significantly simplify the branching model.
- **Separation of Concerns**: code is organized in non-overlapping layers and sections.
- **Single Responsibility**: functions or classes perform single clear tasks.
- SC and SRP can be applied to any software.
- Dogmatism should be avoided: single responsibility vs less responsibilities.

Integrating different versions (features, functionalities)

TECHNISCHE UNIVERSITÄT DARMSTADT

Maintainers coordinate the integration.

- Keep the branching model (version structure) as simple as possible.
- Main and development branches are protected and managed by Maintainers.
- Maintainers are responsible for git tags and cleanup:
  - **Main**: integrations from accepted publications and development branch.
  - **Development**: integration of automatically-tested improvements.
  - **Feature**: reduce git-conflicts by separating responsibilities and concerns.

Research Data Management Concept of the CRC1194

TECHNISCHE UNIVERSITÄT DARMSTADT

1. ~~Use version control ("git").~~
   - ~~Pays off when investigating / switching between different research ideas.~~
2. **Automate and document (script) data-processing steps.**
   - Pays off after several executions.
3. **Implement case studies using literal programming (Jupyter Notebooks).**
   - Pays off in collaborations and when writing results sections in publications.
4. **Organize case studies to easily retrieve parameter-specific information.**
   - Pays off when investigating errors: nothing ever works from the first try!
5. Focus first on secondary data, and use a simple data/metadata format.
   - Low hanging fruit, small data sizes.
6. Use milestones to cross-link digital research artifacts on data/publication repositories.
   - Satisfies the DFG requirement, takes half an hour.
7. Use Continuous Integration for automating research workflows.
   - Pays off during peer-review.

Test Driven Development: Define Research Test Results First

TECHNISCHE UNIVERSITÄT DARMSTADT

Test-Driven Development[2] for CSE

1. Do:
2. Define final test results you want to publish.
3. Define the evaluation of the test results.
4. Program top-down, i.e. write calls to non-existing functions.
5. Don't unit-test, yet! Only if a top-level test fails, dig deeper.
6. Implement a fix to a failing test.
7. While (tests fail)

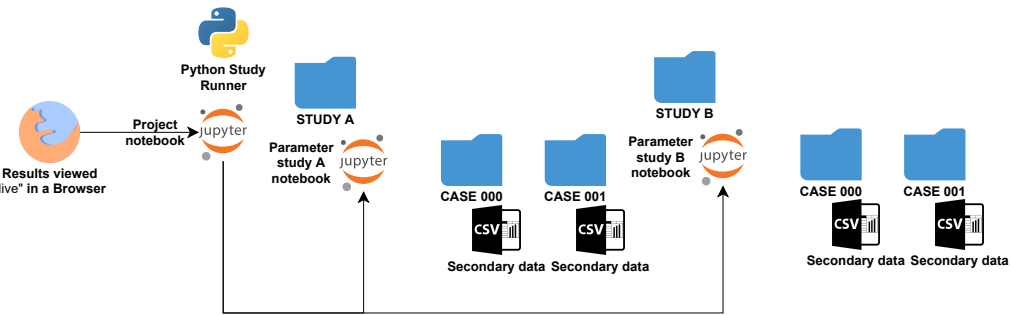
[2] Freeman, Steve, and Nat Pryce. Growing object-oriented software, guided by tests. Pearson Education, 2009.

Jupyter notebooks[3]

- **Documentation**: geometry, initial and boundary conditions, error norms, comparison data.
- **Processing**: verification errors (conservation, convergence, stability), validation errors.
- **Result analysis**: very straightforward, interactive, remote, directly-usable in publications!

[3] https://jupyter.org/

Case Parameterization



- Don't store metadata in folder names.
- Generate locally-unique case ID's, each Case ID mapping to a Parameter Vector.
- Store the mapping (Case ID $\rightarrow$ Parameter Vector) in a simple format (CSV).

Research Data Management Concept of the CRC1194

TECHNISCHE UNIVERSITÄT DARMSTADT

1. ~~Use version control ("git").~~
   - ~~Pays off when investigating / switching between different research ideas.~~
2. ~~Automate and document (script) data-processing steps.~~
   - ~~Pays off after several executions.~~
3. ~~Implement case studies using literal programming (Jupyter Notebooks).~~
   - ~~Pays off in collaborations and when writing results sections in publications.~~
4. ~~Organize case studies to easily retrieve parameter-specific information.~~
   - ~~Pays off when investigating errors: nothing ever works from the first try!~~
5. **Focus first on secondary data, and use a simple data/metadata format.**
   - Low hanging fruit, small data sizes.
6. Use milestones to cross-link digital research artifacts on data/publication repositories.
   - Satisfies the DFG requirement, takes half an hour.
7. Use Continuous Integration for automating research workflows.
   - Pays off during peer-review.

Importance of Secondary Data

TECHNISCHE UNIVERSITÄT DARMSTADT

Secondary Data (tables, diagrams)

- Decide if a paper will be accepted or rejected.
- Are the basis for communication in CRC's cooperations.
- Are the basis for comparison of research results from literature.
- Are very small and easily manageable.

Standardizing Secondary Data

TECHNISCHE UNIVERSITÄT DARMSTADT

- We compare data from studies involving parameters.
  - Physical parameters.
  - Method parameters.
- Study parameters are metadata, results are data.
- We need to easily compare results between different parameter sets.
- We need to easily investigate correlations in the parameter space.
- Parameters are often varied in a structured way, naturally leading to Hierarchical Data Format.

TECHNISCHE UNIVERSITÄT DARMSTADT

- The Hierarchical Data Format[4] is a data format specification.
- HDF5 encodes a hierarchy of data items with their metadata and supports grouping data.
- HDF5 is an interoperable and open standard.
- HDF5 is implemented for scalable parallel Input / Output.
- HDF5 implementations exist in many programming languages.

[4] Collette, Andrew. Python and HDF5: unlocking scientific data. " O'Reilly Media, Inc.", 2013.

Standardizing Secondary Data

Hierarchical Data Format: HDF5 II

TECHNISCHE UNIVERSITÄT DARMSTADT

```
File: example.h5
+-- Attributes
|   +-- experiment_name: 'Sample Experiment'
|   +-- experiment_date: '2024-07-01'
|   \-- description: 'Sample description.'
+-- group1
|   +-- Attributes
|   |   \-- description: 'This group contains data from sensor 1'
|   \-- data1
|       +-- Attributes
|       |   +-- units: 'meters'
|       |   +-- description: 'Distance measurements from sensor 1'
|       |   +-- physical_quantity: 'distance'
|       |   +-- calibration_date: '2024-06-30'
|       |   \-- calibration_coefficient: 1.01
|       \-- Data: [0.123, 0.456, 0.789, ...]
```

Standardizing Secondary Data

Hierarchical Data Format: HDF5 III

TECHNISCHE UNIVERSITÄT DARMSTADT

```
\-- group2
    +-- Attributes
    |   \-- description: 'This group contains data from sensor 2'
    \-- data2
        +-- Attributes
        |   +-- units: 'meters'
        |   +-- description: 'Distance measurements from sensor 2'
        |   +-- physical_quantity: 'distance'
        |   +-- calibration_date: '2024-06-29'
        |   \-- calibration_coefficient: 0.98
        \-- Data: [0.234, 0.567, 0.891, ...]
```

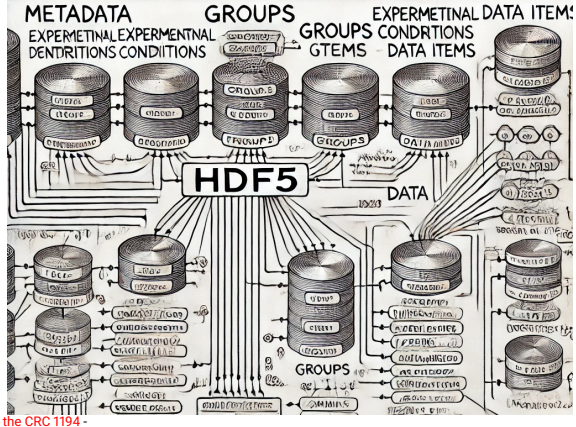
TECHNISCHE UNIVERSITÄT DARMSTADT

Now let's try to get data from sensor with calibration coefficient $\in [0.9, 1]$

```
with h5py.File('example.h5', 'r') as f:
    def check_group(group):
        for key, item in group.items():
            if isinstance(item, h5py.Group):
                # Recursively check sub-groups
                check_group(item)
            elif isinstance(item, h5py.Dataset):
                # Check if the dataset has the attribute
                if 'calibration_coefficient' in item.attrs:
                    coeff = item.attrs['calibration_coefficient']
                    if min_coeff <= coeff <= max_coeff:
                        print(f"Dataset: {item.name},
                        Calibration Coefficient: {coeff}")
```

TECHNISCHE UNIVERSITÄT DARMSTADT

This is how ChatGPT and I together see HDF5 for secondary data :)



A Simple Format for Secondary Data and Metadata I

It's just a simple table.

TECHNISCHE UNIVERSITÄT DARMSTADT

- **We leverage the small size of secondary data**[5]
- We store metadata and data as columns in a table.
- We store the table as CSV and use Python Pandas to work with it.
  - Yes, you can also use Excell.
- **We duplicate metadata**, we don't care, because
  - We get a 100% interoperable dataset.
  - This trivializes data analysis, exchange, and comparison.

[5] Marić, Tomislav, et al. "A pragmatic workflow for research software engineering in computational science." arXiv preprint arXiv:2310.00960 (2023).

A Simple Format for Secondary Data and Metadata II

It's just a simple table.

TECHNISCHE UNIVERSITÄT DARMSTADT

Blue are metadata, red are (secondary) data.

PARAM_HIDDEN_LAYERS	PARAM_OPTIMIZER_STEP	PARAM_MAX_ITERATIONS	PARAM_EPOCH	TRAINING_MSE	
10,10,10,10	0.0001	3000	1	1.091560	
10,10,10,10	0.0001	3000	2	1.082970	
10,10,10,10	0.0001	3000	3	1.077200	
10,10,10,10	0.0001	3000	4	1.072650	
...	...	...	...	...	...
10,10,10,10	0.001	3000	1	0.992354	
10,10,10,10	0.001	3000	2	0.991959	
10,10,10,10	0.001	3000	3	0.995102	
10,10,10,10	0.001	3000	4	0.996143	
...	...	...	...	...	...

A Simple Format for Secondary Data and Metadata III

TECHNISCHE UNIVERSITÄT DARMSTADT

Let's do some basic research using this secondary data format.

- Sub-sample using metadata intervals

```
data = study[(study['calibration_parameter'] >= 0.9) and
             (study['calibration_parameter'] <= 1.1)]
```

- Analyzing correlations of data w.r.t. metadata is equally easy.
- Comparing results from two studies

```
value_columns = [col for col in df1.columns
                 if not col.startswith('PARAM_')]
difference_df = study1[value_columns].copy() -
                study2[value_columns].copy()
```

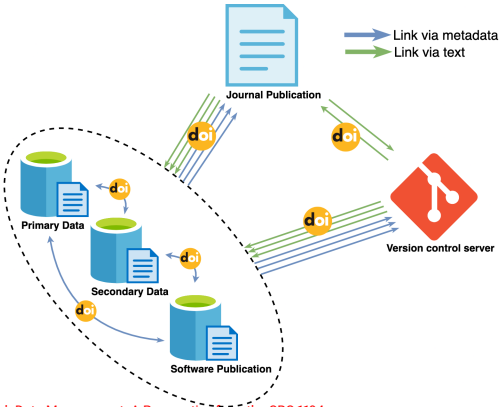
- Pandas is one line of code away from Numpy, i.e. from Machine Learning.
- We haven't used it on primary data, but first examples seem to work.
  - We only duplicate metadata!

Research Data Management Concept of the CRC1194

TECHNISCHE UNIVERSITÄT DARMSTADT

1. ~~Use version control ("git").~~
   - ~~Pays off when investigating / switching between different research ideas.~~
2. ~~Automate and document (script) data-processing steps.~~
   - ~~Pays off after several executions.~~
3. ~~Implement case studies using literal programming (Jupyter Notebooks).~~
   - ~~Pays off in collaborations and when writing results sections in publications.~~
4. ~~Organize case studies to easily retrieve parameter-specific information.~~
   - ~~Pays off when investigating errors: nothing ever works from the first try!~~
5. ~~Focus first on secondary data, and use a simple data/metadata format.~~
   - ~~Low hanging fruit, small data sizes.~~
6. **Use milestones to cross-link digital research artifacts on data/publication repositories.**
   - Satisfies the DFG requirement, takes half an hour.
7. Use Continuous Integration for automating research workflows.
   - Pays off during peer-review.

Cross-linking data, source code and reports/publications I

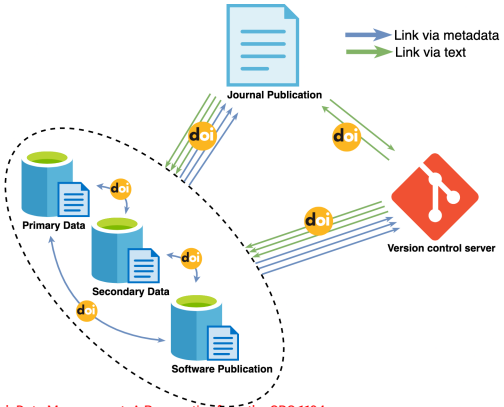


Cross-linking milestones

- A pre-print is ready for publication.
- A publication is revised and ready for re-submission.
- A publication has been accepted for publication.
- A PhD thesis is ready for submission.
- A method has found not to work.

Research Data Management: A Perspective from the CRC 1194 -
M. Schwarzmeier, T. Tolle, JP. Lehr, I. Pappagianidis, B. Lambie, D. Bothe, C. Bischof, **T. Marić**

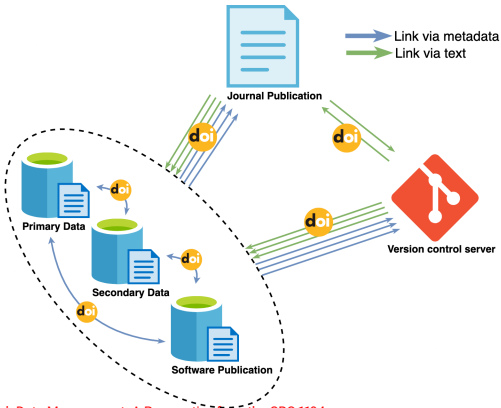
Cross-linking data, source code and reports/publications II



- Save Overleaf label and / or "tag" the manuscript version.
- Upload data and software to a data repository.
  - Zenodo.org is not indexed by Google (Scholar).
  - While data citations don't help you, indexed data links to papers!
- Cite data and software in the preprint.
- Upload pre-print, obtain it's DOI.
- Add pre-print DOI to data and software metadata on the data repository.

Research Data Management: A Perspective from the CRC 1194 -
M. Schwarzmeier, T. Tolle, JP. Lehr, I. Pappagianidis, B. Lambie, D. Bothe, C. Bischof, **T. Marić**

Cross-linking data, source code and reports/publications III



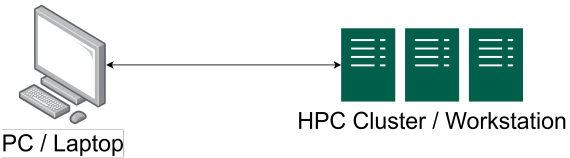
- First time it takes an afternoon, second time it takes 10-15 minutes.
- Data and software increases citation count and visibility (subjective observation).
- How long does it take to get results, generate software, data, and write the preprint?
- How much tax funds is behind a single DFG-funded publication?
- We should strive to ensure FAIRness of our research as scientists.

Research Data Management: A Perspective from the CRC 1194 -
M. Schwarzmeier, T. Tolle, JP. Lehr, I. Pappagianidis, B. Lambie, D. Bothe, C. Bischof, **T. Marić**

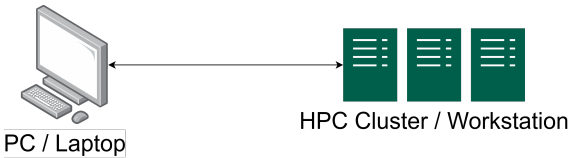
Research Data Management Concept of the CRC1194

TECHNISCHE UNIVERSITÄT DARMSTADT

1. ~~Use version control ("git").~~
   - ~~Pays off when investigating / switching between different research ideas.~~
2. ~~Automate and document (script) data-processing steps.~~
   - ~~Pays off after several executions.~~
3. ~~Implement case studies using literal programming (Jupyter Notebooks).~~
   - ~~Pays off in collaborations and when writing results sections in publications.~~
4. ~~Organize case studies to easily retrieve parameter-specific information.~~
   - ~~Pays off when investigating errors: nothing ever works from the first try!~~
5. ~~Focus first on secondary data, and use a simple data/metadata format.~~
   - ~~Low hanging fruit, small data sizes.~~
6. ~~Use milestones to cross-link digital research artifacts on data/publication repositories.~~
   - ~~Satisfies the DFG requirement, takes half an hour.~~
7. **Use Continuous Integration for automating research workflows.**
   - Pays off during peer-review.



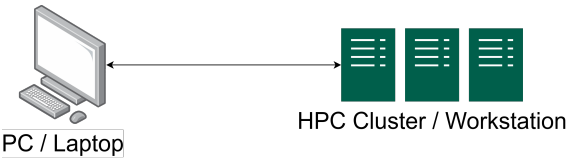
while Results are unsatisfactory **do**
    Implement some data-processing code.
    (Compile the code.)
    for All studies **do**
        Run the study.
        Analyze results.
    end for
    Compare old and new results.
end while



Issues...

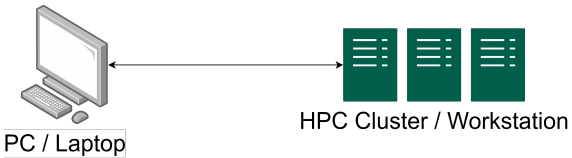
- Starting studies takes time.
- Analyzing results takes time.
- Often the results are not checked "live" as the study runs - **waste of research time and CPUh**.
- **Only the researcher knows the details** behind the initialization, running and post-processing scripts - **when this person leaves, the reproducibility is gone.**
- A researcher may forget to run a study and believe all tests have passed.

Continuous Integration of Scientific Software III



while Results are unsatisfactory **do**
    Implement some data-processing code.
    (Compile the code.)
    for All studies **do**
        Run the study. ▷ Manual.
        Analyze data. ▷ Manual.
    end for
    Examine results. ▷ Manual.
end while

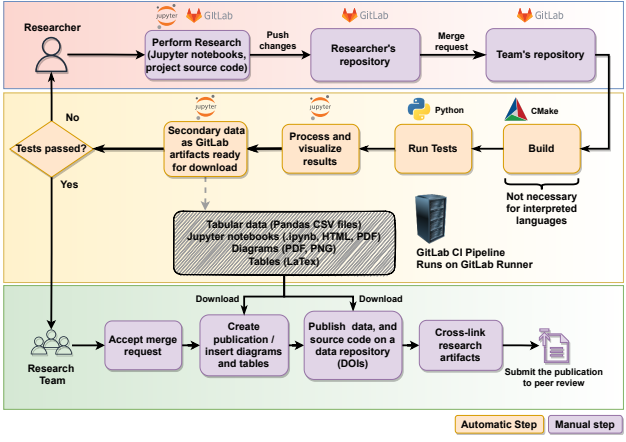
Continuous Integration of Scientific Software III



while Results are unsatisfactory **do**
    Work on algorithms.
    (Compile the code.)
    Run the studies (jobs). ▷ Automatic.
    Run data analysis (jobs). ▷ Automatic.
    Examine results. ▷ Manual.
end while

(Continuous) Integration of scientific software IV

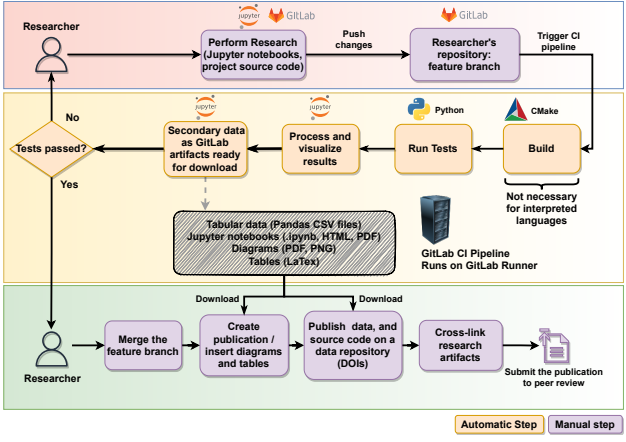
Schematic diagram for the team workflow



Working in a team.

(Continuous) Integration of scientific software V

Schematic diagram for the individual workflow

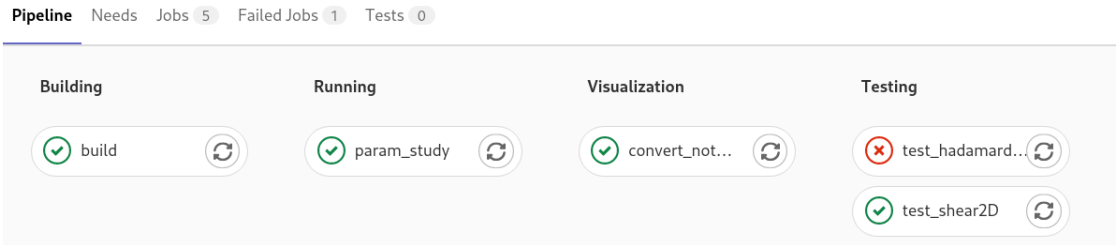


Working alone.

(Continuous) Integration of scientific software VI

TECHNISCHE UNIVERSITÄT DARMSTADT

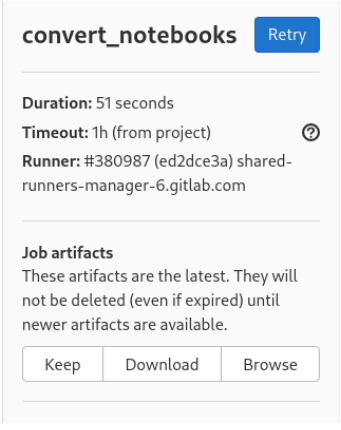
An example CI pipeline you can view in your favorite browser



(Continuous) Integration of scientific software VII

CI in a nutshell II

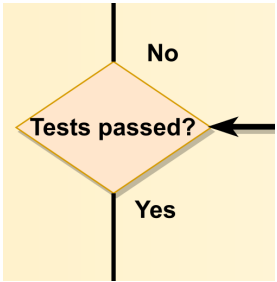
TECHNISCHE UNIVERSITÄT DARMSTADT



- Files created within a CI job are gone when the job ends.
- GitLab uses **job artifacts** to pass on data from one job to the next.
- **Job artifacts can only be files stored in project's sub-folders.**
- Libraries and applications are passed to other jobs as artifacts.
- **Artifacts can be downloaded on the GitLab project website.**

(Continuous Integration with result visualization) I

Test evaluation I

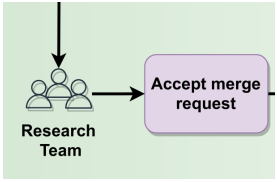


Straightforward for easily quantifiable errors

- **Examples:** volume conservation, order of convergence, total wall clock time, weak scaling, ...
- **Python scripts test secondary data** agglomerated by Jupyter notebooks from simulation results.

(Continuous Integration with result visualization) II

Test evaluation II



Difficult for errors that cannot be quantified easily

- **Examples:**
  - Is is the difference between simulation and experiment data $\leq 4\%$?
  - How to quantify the difference for complex signals?
- **Option 1: Researchers evaluate the test results even if all CI jobs pass.**
  - A simple and efficient solution 🎓.
- **Option 2: Use statistics to quantify the difference.**

Research Data Management Concept of the CRC1194

TECHNISCHE UNIVERSITÄT DARMSTADT

1. ~~Use version control ("git").~~
   - ~~Pays off when investigating / switching between different research ideas.~~
2. ~~Automate and document (script) data-processing steps.~~
   - ~~Pays off after several executions.~~
3. ~~Implement case studies using literal programming (Jupyter Notebooks).~~
   - ~~Pays off in collaborations and when writing results sections in publications.~~
4. ~~Organize case studies to easily retrieve parameter-specific information.~~
   - ~~Pays off when investigating errors: nothing ever works from the first try!~~
5. ~~Focus first on secondary data, and use a simple data/metadata format.~~
   - ~~Low hanging fruit, small data sizes.~~
6. ~~Use milestones to cross-link digital research artifacts on data/publication repositories.~~
   - ~~Satisfies the DFG requirement, takes half an hour.~~
7. ~~Use Continuous Integration for automating research workflows.~~
   - ~~Pays off during peer-review.~~

Acknowledgements

TECHNISCHE UNIVERSITÄT DARMSTADT

Interaction between Transport and Wetting Processes

Funded by the German Research Foundation (DFG) – Project-ID 265191195 – CRC 1194 : Z-INF