

Research Data Management for Computational Science

Christian Jacobs

`c.jacobs10@imperial.ac.uk`

`ctjacobs.github.io`

and

Alexandros Avdis, Gerard Gorman, Matthew Piggott

Data requirements

- Data produced by scientific software should be **recomputable** and **reproducible**.
- This requires:
 - the **software** itself (with info about the specific version used)
 - **raw data** (input and output files)
 - **provenance metadata**
- We need a way of **publishing** this data and software at the push of a button...
- ...and a way of **referencing** it correctly in papers.

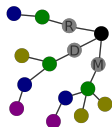
Online repositories

- Organisations such as **Figshare** (figshare.com) and **Zenodo** (zenodo.org) provide hosting for **code** and **datasets**.
- Each code/dataset is given its own **Digital Object Identifier (DOI)**.
- Programs developed by users can interface with Figshare and Zenodo via their APIs.

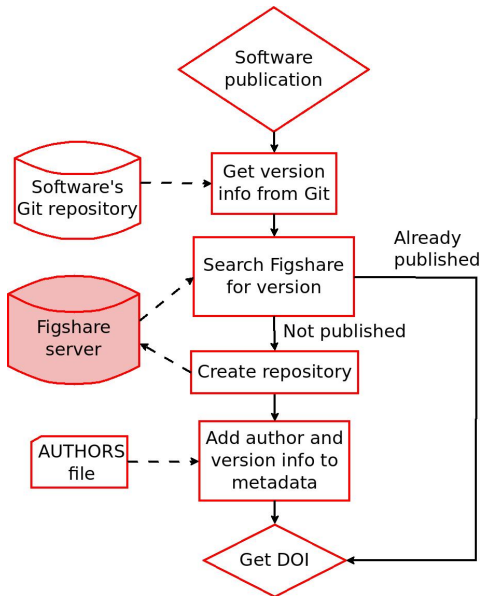
Aims

- Develop a software tool which facilitates the **automated publication** of both **software** and **data** to Figshare, Zenodo and DSpace repositories.
- Incorporate this tool into the workflow of **Fluidity** – an open-source CFD code for fluid flow simulations (<http://fluidity-project.org>).
- DOIs are ‘minted’ automatically, and added to the current **metadata** of simulation output.

- **PyRDM**: **R**esearch **D**ata **M**anagement with **P**ython
- Released under the GNU GPL version 3 license.
- **Source code** on GitHub: <http://github.com/pyrdm>



- **Paper**: C. T. Jacobs, A. Avdis, G. J. Gorman, and M. D. Piggott (2014). PyRDM: A Python-based library for automating the management and online publication of scientific software and data. *Journal of Open Research Software*, 2(1):e28, DOI: 10.5334/jors.bj



- User specifies data files to be uploaded (e.g. [`*.vtu`]).
- If a **new version** of the repository is created (e.g. during peer review), **MD5 checksums** used to selectively re-upload only those files that have been modified.

Application: Fluidity + PyRDM

- PyRDM has been integrated into the **workflow** of Fluidity.
- Users enable a **'publish' option** in their simulation's setup file, then run a Fluidity-specific publishing tool which uses the PyRDM library.
- The end-user just has to provide:
 - Their Figshare authentication details.
 - A list of any data files they want to publish (e.g. *.vtu).
 - Optionally: an existing Figshare publication ID and DOI.
- DOIs are **recorded** in the simulation setup file – if the simulation is run again, the same DOI is used to store the data.
- The DOIs for the software and input data are appended to the simulation output for **data provenance**.

Application: Fluidity + PyRDM

The screenshot shows the Fluidity GUI interface. The title bar reads "Diamond: top_hat_cg_supg.flml (/data/fluidity-rdm/tests/top_hat_cg_supg)". The menu bar includes "File", "Edit", "View", "Validate", "Tools", and "Help".

The left pane displays a tree view of the project structure under the "Node" header:

- fluidity_options
 - simulation_name
 - problem_type
 - ▶ geometry
 - ▶ io
 - ▶ timestepping
 - ▶ physical_parameters ✖
 - ▶ material_phase (Fluid)
 - material_phase
 - mesh_adaptivity
 - imported_solds
 - turbine_model
 - ocean_biology
 - ocean_forcing
 - reduced_model
 - porous_media
 - embedded_models
 - flrecomp
 - multiphase_interaction
 - ▼ publish ✖
 - service (highlighted)
 - ▶ software
 - ▼ input_data
 - files
 - article_id
 - doi
 - ▶ output_data

The right pane shows the "Option Properties" for the selected "service" node:

- Description:** The online publishing service that you would like to use.
- Data:** figshare
- Comment:** No comment

Application: Fluidity + PyRDM

Example: simulation of the top_hat_cg_supg test case

The screenshot displays a software interface with a top navigation bar containing three buttons: "My data" (with a person icon), "Projects" (with a folder icon), and "Activity" (with a list icon). Below the navigation bar is a storage status indicator showing "11% of private storage used" with a progress bar. The main content area features a table with the following elements:

- Row 1: A header bar with a checkbox, two buttons "Add to Fileset" and "Batch edit", and a "Type" dropdown menu showing "mouseover(1)".
- Row 2: A checkbox, the file name "top_hat_cg_supg-output-data", and a "DATASET" icon.
- Row 3: A checkbox, the file name "top_hat_cg_supg-input-data", and a "DATASET" icon.
- Row 4: A checkbox, the file name "Fluidity-version-b68c9225ef2c84e827af39541bf45197d2468165", and a "DATASET" icon.

Screenshot of software, input data and output data automatically pushed to Figshare.

Application: Fluidity + PyRDM

Example: simulation of the top_hat_cg_supg test case

```
<constant name="FluidityVersion" type="string"  
value="1baf80aac1e7e735b1cf182bc20761a0c6df7767"/>
```

```
<constant name="SoftwareDOI" type="string"  
value="http://dx.doi.org/10.6084/m9.figshare.1035081"/>
```

```
<constant name="InputDataDOI" type="string"  
value="http://dx.doi.org/10.6084/m9.figshare.1035083"/>
```

```
<constant name="CompileTime" type="string" value="May 23  
2014 15:22:23"/>
```

```
<constant name="StartTime" type="string" value="20140523  
154857.775+0100"/>
```

- Need a better way of affiliating authors - ORCID IDs?
- Lack of API support.
- Need more storage space for private data.