Research Data Management for Computational Science

Christian Jacobs
c.jacobs10@imperial.ac.uk
c tjacobs.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.
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: Research Data Management with Python

Released under the GNU GPL version 3 license.

Source code on GitHub: http://github.com/pyrmdm

Publishing software source code

1. Software publication
2. Get version info from Git
3. Search Figshare for version
   - If published, go to End
   - If not published, go to Create repository
4. Create repository
5. Add author and version info to metadata
6. Get DOI
PyRDM
Publishing data

- 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.
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

Screenshot of the 'publish' option in the Fluidity simulation setup file.
Application: Fluidity + PyRDM
Example: simulation of the top_hat_cg_supg test case

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"/>
Issues

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