



COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

Community-driven organization
advancing Earth science by developing
and disseminating software for
geophysics and related fields.

2023 CIG Annual Business Meeting

13 December 2023
5:30-7:30P
SPIN



SAN FRANCISCO
— CALIFORNIA —

Crafting Quality Research Software and Navigating Publication in Software Journals

December 13, 2023, AGU Fall Meeting

Rene Gassmoeller, Lorraine Hwang, Mohamed Gouiza

Computational Infrastructure for Geodynamics

Find the slides and other materials on: <http://bit.ly/2023-cig-joss>



National
Science
Foundation



dynamo



seismology



computational
science



short-term
crustal dynamics



COMPUTATIONAL
INFRASTRUCTURE
for GEODYNAMICS



mantle
convection



melt and volatiles



long-term
tectonics



education

Starting remarks

- Crafting good enough research software is no magic, there are published guidelines and checklists
- Many steps to create good research software are purely mechanical - this will not solve the functional design, but improve a lot of the "scaffolding"
- These steps take some effort, but:
 - Save your time
 - Grow your community
 - Improve your software sustainability
- A publication of your software rewards these efforts

Best Practices

- Useful Materials:
 - Computational Infrastructure for Geodynamics maintains best practices: <https://geodynamics.org/software/software-bp>
And a contributing checklist for CIG software: [https://github.com/geodynamics/best_practices/blob/main/Contributing Checklist.md](https://github.com/geodynamics/best_practices/blob/main/ContributingChecklist.md)
 - CIG maintains a template repository to create new projects: https://github.com/geodynamics/software_template
 - Journal of Open Source Software submission instruction: <https://joss.readthedocs.io/en/latest/submitting.html>
And a reviewer checklist https://joss.readthedocs.io/en/latest/review_checklist.html
- If your software fulfills CIG or JOSS criteria, it fulfills most criteria for both

Best Practices – Summary

(<https://geodynamics.org/software/software-bp>)



Software Development Best Practices for the CIG Community

	Minimum	Standard	Target
Licensing	Open source	Same as Minimum.	Same as Minimum.
Version Control	All source in version control.	Differentiation between maintenance and new development.	(a) New features added in separate branches. (b) Stable development branches for rapid release of new features.
Coding		(a) User-friendly specification of parameters at run time. (b) Development plan, updated annually. (c) Comments in code with purpose of each function. (d) Users can add features or alternative implementations without modifying main branch. (e) User errors generate message that helps user correct the problem.	Standard + (a) Functionality implemented as a library rather than an application. (b) Output of provenance information. (c) Parallel access to inputs/outputs. (d) Checkpointing.
Portability, configuration and building	(a) Codes builds on Unix-like machines with free tools. (b) Portable build system.	Minimum + (a) Dependency checking. (b) Automation and portability of configuration and building. (c) Each simulation outputs all configuration and build options for reproducibility.	Standard + (a) Selection of compilers, optimization, build flags during configuration without modifying files under version control. (b) Multiple builds using same source. (c) Allows installation to a central location.
Testing	(a) Code includes tests that verify it runs properly. (b) Results of accuracy and/or performance benchmarks (if established by the community).	Code includes pass/fail tests that verify it runs properly.	(a) Pass/fail unit testing for verification at a fine grain level. (b) Method of Manufactured Solutions for verification at a coarse grain level.
Documentation	(a) Instructions for installation. (b) Description of all parameters. (c) Explanation of physics the code simulates. (d) Cookbook examples with input files. (e) Citable publication.	(a) Description of workflow for research use. (b) Description of how to extend code in anticipated ways.	Standard + (a) Guidelines on parameter scales/combinations for which code is designed/tested. (b) FAQs or knowledge base.
User workflow		(a) Changing simulation parameters does not require rebuilding. (b) User-specified directories and filenames for input/output. (c) Use of standard binary formats. (d) Citation for code version.	Standard + Reproducibility via archiving of workflow.

Minimum: Practices that codes must follow in order to be accepted by CIG.

Standard: Practices in addition to the Minimum that should be used by all codes developed within the CIG community. Codes not meeting these standards should be actively working to eliminate deficiencies.

Target: Desirable practices beyond the Standard that developers should consider in defining development priorities for codes developed within the CIG community.

Publishing Software - Why?

- Software is a research product, just like scientific results
- Traditional publications are not focussed on the software, the "implementation" is typically not reviewed
- Scientific results depend on software, there should be independent review
- A mechanism for credit
- A mechanism for reproducibility

Publishing Software - Which Journal?

- A number of existing and new journals accept software:
 - JOSS: The Journal of open-source software
 - GMD: Geoscientific Model Development
 - AGU journals (e.g. G3): Technical Report:Methods
 - SoftwareX
 - For more see this list by the Software Sustainability Institute: <https://www.software.ac.uk/which-journals-should-i-publish-my-software>
- We will mainly discuss a JOSS publication, but briefly review other journals

Software journals brief comparison

	JOSS	GMD	AGU	SoftwareX
Software best practices	yes	maybe	no	maybe
Open source / free to use	yes	maybe	maybe	yes
Domain relevance	no	yes	yes	no
Methods	maybe	yes	no information	no information
Open review	yes	yes	no	no
Open access (article)	yes	yes	yes	no
Free to publish	yes	no	no	no

yes
maybe
no
no information

**CIG is exploring a cooperation with JOSS
to support software publication**

JOSS Submission requirements

1. The software must be open source as per the [OSI definition](#).
2. The software must be hosted at a location where users can open issues and propose code changes freely.
3. The software must have an **obvious** research application.
4. You must be a major contributor to the software you are submitting.
5. Your paper must not focus on new research.
6. Your paper must be in a Git-based repository with your software.

JOSS paper contents

A JOSS paper is a short (2-3 page) article

1. A list of the authors of the software and their affiliations.
2. A summary describing the high-level functionality and purpose of the software for a diverse, *non-specialist audience*.
3. A *Statement of need* section that clearly illustrates the research purpose of the software and places it in context.
4. A list of key references, including to other software for related needs.
5. Mention (if applicable) a representative set of past or ongoing research projects about it and scholarly publications enabled by it.
6. Acknowledgement of any financial support.

Submission template

JOSS provides a template for a software paper:

<https://joss.readthedocs.io/en/latest/submitting.html#example-paper-and-bibliography>

Examples for JOSS papers about Geoscientific software are:

- [BurnMan \(CIG\): https://joss.theoj.org/papers/10.21105/joss.05389](https://joss.theoj.org/papers/10.21105/joss.05389)
- [Motorcycle: https://joss.theoj.org/papers/10.21105/joss.05097](https://joss.theoj.org/papers/10.21105/joss.05097)
- [Underworld2: https://joss.theoj.org/papers/10.21105/joss.01797](https://joss.theoj.org/papers/10.21105/joss.01797)
- [SeisModels.js: https://joss.theoj.org/papers/10.21105/joss.02043](https://joss.theoj.org/papers/10.21105/joss.02043)
- [GeoHexViz: https://joss.theoj.org/papers/10.21105/joss.05073](https://joss.theoj.org/papers/10.21105/joss.05073)

JOSS submission workflow

Submission is as simple as:

- Filling in the [short submission form](#)
- Waiting for the managing editor to start a pre-review issue over in the JOSS reviews repository: <https://github.com/openjournals/joss-reviews>
- Waiting for reviews and address the comments in an interactive review process

No submission fees

There are no fees for submitting or publishing in JOSS. You can read more about their [cost and sustainability model](#).

Preprint Policy

Authors are welcome to submit their papers to a preprint server ([arXiv](#), [bioRxiv](#), [SocArXiv](#), [PsyArXiv](#) etc.) at any point before, during, or after the submission and review process.

Review process

- The review process happens in a github issue in a repository: <https://github.com/openjournals/joss-reviews>.
- The review process is public.
- The review process is initiated by the editor, and performed by independent reviewers.
- An [EditorialBot](#) supports and automates common operations.

See an example review process:

<https://github.com/openjournals/joss-reviews/issues/5073>

Thank you for your attention

- Slides and recordings of the full workshop are available on Youtube and at <http://bit.ly/2023-cig-joss>
- Consider contributing your software to CIG or JOSS to receive more publicity and credit.
- The CIG Seismic Cycles Working Group is organizing a special issue in JOSS. Let us know if you are interested.

And thank you to the National Science Foundation for making these workshops possible.

