

## Contributing Software

CIG welcomes contributions of modeling codes in geophysics that meet our software best practices (<https://geodynamics.org/cig/dev/best-practices/>). Software best practices aid your project in becoming discoverable and reusable. Best practices help you increase the impact of your work and get appropriate credit. By sharing your software, you make science more reproducible, and contribute to a growing software ecosystem. Additionally, following best practices allows you to direct more effort towards feature rich code while increasing your userbase. CIG provides support and resources for projects that have been accepted as CIG software.

The Checklist below, is meant as a guide to prepare your software for CIG review.

### CHECKLIST

**Bold:** mandatory – following minimum best practices

Plain: optional – following standard or target best practices

\*: CIG support available.

- ☐ **Include a 'README.md' file**

The file should have the following attributes:

- ☐ **Description**

Explain the research problem the software is designed to address. Discuss any limitations and vulnerabilities. Include instructions on the best way to install and run the software.

- ☐ **Feedback, support, and help**

Explain how a user should provide feedback, report bugs, and get help. Describe how to create new issues and the project's timeline for responding to bug reports and enhancement requests. You may choose to include this in the CONTRIBUTING.md file.

- ☐ **How to Cite**

Summarizes and reinforces the information in CITATION.

- ☐ **How to get more information.**

Provide links to a website, manual, contact information, and any other information helpful to the user.

- ☐ **License**

States the license type and links to the full LICENSE file.

- ☐ **Include an easy-to-find documentation directory, preferably named 'doc/'**

Folder containing user documentation. User manual describes the problem the code addresses and its theoretical basis. Should include a description of the layout of the code, installations instructions, how to get started, workflows, included benchmarks to verify and validate results where appropriate, and user examples (cookbooks).

☐ **Include a 'LICENSE' file**

Add a plain text copy of your OSI approved software license to your repository.

<https://opensource.org/licenses>

CIG codes are required to use an OSI approved license. We recommend the GPL-2.0, GPL-3.0, or MIT licenses.

See:

<https://geodynamics.org/cig/dev/software-ownership-and-licensing/gnu-gpl/>

☐ **Include a 'CONTRIBUTING.md' file**

Describe how to contribute to code development. Should include guidelines for coding standards, acceptable contributions, and requirements for additional tests.

☐ **Include an 'INSTALL' file, or refer to installation instructions in your 'README.md'**

Describes how to build the code including all dependencies in plain text. Codes are required to use a build system like cmake\*, or make\*

☐ **Optional: Include a folder docker/\***

Folder containing instructions to build a docker or similar container.

☐ **Include a 'VERSION' file**

State the current version number. Version numbers must be unique. We recommend following the semantic versioning scheme: <https://semver.org/>

☐ **Include a 'CHANGES' file or another form of release notes**

Provide a plain text summary of major changes in all existing releases (release notes). Changes should describe new features as well as incompatibilities to previous versions.

☐ **If available, use a continuous integration system\* and include necessary files**

Continuous integration testing is used to automatically build and test code. Automated tests should be provided to verify software functionality.

☐ **Include tests**

Provide at least one test, e.g. in a directory 'tests/' that ensures the correct execution of the compiled program. Include instructions for how to run and interpret the results in our 'README.md' or inside the documentation.

☐ **Include citation information at least by providing a 'CITATION' file**

Explain how you want your work cited in plain text. This should include a text and bibtex version of any publications and to the software package itself.

We recommend a machine-readable version using one of the following formats:

☐ **codemeta.json**

<https://codemeta.github.io/create/>

- ☐ citation.cff  
<https://citation-file-format.github.io/>
- ☐ Optional: Include an 'AUTHORS' file  
List the authors of your code. This may or may not be the same as those listed in the citation and/or the list of project committers.
- ☐ Optional: Include an 'ACKNOWLEDGE' file  
Acknowledge funding agencies or other sources of support to credit.
- ☐ Optional: Include a 'CODE\_OF\_CONDUCT.md' file  
All accepted projects are required to abide by the CIG Code of Conduct:  
<https://geodynamics.org/cig/about/code-conduct/>  
You may add your own code of conduct. If you do, we recommend using the Contributor Covenant CoC as a base file:  
<https://www.contributor-covenant.org/>

## Helpful Examples and Links

Useful CIG projects as examples:

<https://github.com/geodynamics/aspect>  
<https://github.com/geodynamics/Rayleigh>  
<https://github.com/geodynamics/pylith>

Some other useful checklists:

CII Best Practices Badge Program, Best Practices Criteria for Free/Libre and Open Source Software (FLOSS)

<https://github.com/coreinfrastructure/best-practices-badge/blob/master/doc/criteria.md>

Journal of Open Source Software – Review checklist

[https://joss.readthedocs.io/en/latest/review\\_checklist.html](https://joss.readthedocs.io/en/latest/review_checklist.html)

EURISE

<https://fair-software.nl/recommendations/checklist>