I. Advancing Earth Science through Best Practices in Open Source Software: CIG

II. Software Attribution for Geoscience Applications in CIG

Lorraine J. Hwang
Joe Dumit, Alison E. Fish, LOUISE H. KELLOGG,
MacKenzie Smith, and Laura Soito
About CIG

The Computational Infrastructure for Geodynamics is a community-driven organization that advances Earth science by developing and disseminating software for geophysics and related fields.

Primary Scientific domains:
- Geodynamo
- Mantle convection
- Seismology
- Short and long term deformation of crust & lithosphere
- Computational science
- Fluid migration/multiphysics

Funded primarily by the US National Science Foundation (NSF) Geoinformatics Program
COMMUNITY SOFTWARE

Domain relevant open source software contributed and developed by CIG and independent researchers.

- Domain scientists in collaboration with computational scientists

Maintained and developed by:

- Community - at-large
- Developer(s) – “hero” or small team
- Developers – large team

CIG repository hosts 32 codes in a GitHub repository github.com/geodynamics
Software Best Practices: Goals

- **Usability**
  Software is effective and is designed to promote ease-of-use.

- **Sustainability**
  Code can be improved and adapted to a changed environment; resilient.

- **Reproducibility**
  Scientific method built on reproducible and reliable results.
SBP: Supporting Community

Communications
• Mailing lists (8)  • Wikis

Training & Community building
We coordinate and collaborate with other organizations to provide:
• Workshops  • Hackathons
• Tutorials  • Webinars

We target early career researchers.

Facilitating access to high performance computing
• Allocation on XSEDE for testing
• Coordinate code performance & accuracy benchmarks

CIG has invested in the development of Rayleigh a massively parallel spherical harmonic code. Running on ALCF Mira 5th fastest.
## SBP: CIG Standards

<table>
<thead>
<tr>
<th>Licensing</th>
<th>Standard</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open source</td>
<td>Same as Minimum.</td>
<td>Same as Minimum.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Version Control</th>
<th>Differentiation between maintenance and new development.</th>
<th>(a) New features added in separate branches. (b) Stable development branches for rapid release of new features.</th>
</tr>
</thead>
<tbody>
<tr>
<td>All source in version control.</td>
<td></td>
<td>(a) New features added in separate branches. (b) Stable development branches for rapid release of new features.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coding</th>
<th>(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.</th>
<th>Standard + (a) Functionality implemented as a library rather than an application. (b) Output of provenance information. (c) Parallel access to inputs/outputs. (d) Checkpointing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) All source in version control. (b) Differentiation between maintenance and new development.</td>
<td></td>
<td>(a) New features added in separate branches. (b) Stable development branches for rapid release of new features.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Portability, configuration and building</th>
<th>Minimum + (a) Dependency checking. (b) Automation and portability of configuration and building. (c) Each simulation outputs all configuration and build options for reproducibility.</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Codes builds on Unix-like machines with free tools. (b) Portable build system.</td>
<td></td>
<td>(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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Testing</th>
<th>Code includes pass/fail tests that verify it runs properly.</th>
<th>(a) Pass/fail unit testing for verification at a fine grain level. (b) Method of Manufactured Solutions for verification at a coarse grain level.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Code includes tests that verify it runs properly. (b) Results of accuracy and/or performance benchmarks (if established by the community).</td>
<td></td>
<td>(a) Method of Manufactured Solutions for verification at a coarse grain level.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Documentation</th>
<th>(a) Description of workflow for research use. (b) Description of how to extend code in anticipated ways.</th>
<th>Standard + (a) Guidelines on parameter scales/combinations for which code is designed/tested. (b) FAQs or knowledge base.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(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.</td>
<td></td>
<td>(a) Guidelines on parameter scales/combinations for which code is designed/tested. (b) FAQs or knowledge base.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User workflow</th>
<th>(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.</th>
<th>Standard + Reproducibility via archiving of workflow.</th>
</tr>
</thead>
<tbody>
<tr>
<td>(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.</td>
<td></td>
<td>(a) Reproducibility via archiving of workflow.</td>
</tr>
</tbody>
</table>

---

**Write good code!**

**Can I build it?**

**Does it run?**

**Can I use it?**

**But is it easy?**

---

**SEE:** [https://geodynamics.org/cig/dev/best-practices/](https://geodynamics.org/cig/dev/best-practices/)
SBP: Citation

Why?
• Provide credit and recognition to developers
• Aid in discoverability, reuse, and reproducibility

Are we following our own Best Practices?
“citable paper”

Science Paper (15)
Paper on the Code (4)
User Manual (3)
Website (1)
Additional Attribution (9)

14 with no citation information:
Archived/legacy (7), variants (3), other (3), missing (1)
Sample: 5 years of self-reported or “searchable” publications (journals, conference proceedings, thesis): 308

Total Code Mentions: 500
  Version: 13% (65)
  URL: 21% (104)
  Licensing: <1% (4)

Includes non-CIG codes both commercial and open source.

CIG ONLY codes
Mentions Code Name: 83% (257)
Citation: 75% (206)
Acknowledge CIG: 19% (58)

 Citation practices show some consistency within research groups.
We use *PyLith 2.1.0 for linux* ([PyLith, 2015; Aagard and Williams, 2013; Aagaard et al., 2015](geodynamics.org)) published under the open source *MIT* license freely available through the *Computational Infrastructure for Geodynamics (geodynamics.org)*.

**ACKNOWLEDGEMENTS**

We thank the *Computational Infrastructure for Geodynamics (geodynamics.org)* which is funded by the *National Science Foundation under award NSF-094946*.

**IN REFERENCE**


**Questions:**
What is citable?
Who should be cited?
What is a persistent archive?
1. **Website**  
   Easy
   Standardized language accessible from landing page.

2. **Software**  
   Not too difficult
   Plug in to generate citations at runtime
   Future Questions:
   Transitive credit
   Promote reproducibility
   Others?

3. **Other**  
   Needs work
   Archiving, discoverability, workflows, etc.

---

**KEEP IT SIMPLE AND STRAIGHTFORWARD.**
Contact Us

Director
Prof. Louise H. Kellogg

Associate Director
Dr. Lorraine J. Hwang

contact@geodynamics.org

CIG
2119 Earth and Physical Sciences Building
One Shields Avenue
University of California, Davis, CA 95616

lhkellogg@ucdavis.edu
ljhwang@ucdavis.edu