8-29-2019

Collaborative Research: Development and Application of a Framework for Integrated Geodynamic Earth Models

Anthony Lowry
Utah State University, tony.lowry@usu.edu

Data Management Plan

The data products of this proposal are two-fold: The source code produced during the development of the features of our software as discussed in this proposal, along with extensive accompanying documentation; and the results of computer simulations. We will address these separately.

Source code and documentation. ASPECT has been distributed under the GNU General Public License (GPL, version 2 or later), an OSI-approved open source license since its beginning in 2011. The intention of using an open source license was to make the code available to a broad range of users in the sciences. This follows the spirit of nearly all of the projects the PIs of this proposal have been involved in, and which have been published under either the GPL or LGPL for the last nearly two decades. Five of the eight PIs of this proposal are Principal Developers of ASPECT and have consequently supported this model for several years; we will continue in this direction and have the support of the other three Principal Developers (see the corresponding letters of collaboration).

While we plan to continue with our once-or-twice-annual release schedule, we do not only make software available through periodic releases. Instead, we provide open software environments. For example, the deal.II library in which several of us have been involved and upon which ASPECT builds, has had a publicly available source code management system since before its first release in 2000 (first CVS and Subversion, now via github.com); ASPECT is also available via github.com. All mailing lists are public and archived. The ASPECT project also has a publicly available bug tracking and patch management system, as part of our github account; both are extensively used.

The amount of data generated this way – in source repositories, wiki pages, mailing lists, etc. – is on the order of a few gigabytes and can easily be stored and made accessible through the existing servers and services our projects are already operating. In summary, the ASPECT project has done most or all of the business in the public for a long time, and intends to continue doing so with all results of this project.

Numerical results. Running the software described in this proposal will produce significant amounts of data. For example, even running a simulation with 100 million unknowns produces approximately 1–3 GB of graphical output files per time step that can then be visualized or postprocessed in some other way. Compression will be able to shrink this to maybe one tenth. On the other hand, every simulation consists of thousands of time steps, i.e., the results of every simulation will be on the order of 100s of GB.

Storing all of this data for a significant number of computations is not practical. However, we will store representative subsets, for example every hundredth time step, for a number of years. We will make this data publicly available through an agreement with UNAVCO (see attached letter of collaboration), NSF's geodetic facility, that has agreed to host simulation results for us at a level of several TB.

The data made available in this way is static; it both allows for and requires that users who download further process it themselves for data extraction or visualization. The subproject at the University of Utah will also experiment with approaches where the data is stored on servers and users interact with it through web portals hosted at The Hive located at the University of Utah. For example, we already have experimental servers running on which one can remotely render the data. The exact form in which we will provide this access is not yet settled and, indeed, subject to research, but as with the software itself, it is our dedicated goal to make the products of this research available to the broader community; we will therefore continue to experiment with approaches to interact with the data we store.

In addition to storing the data, we will save the state of the source code (e.g., a git repository revision hash) along with input parameter files for all simulations from which we publish results. We have done this with a significant number of previous papers, and have had good experiences sharing this information publicly and/or with interested colleagues. This ensures reproducibility to the degree possible and provides proper provenance, at little to no additional cost. Given the progress of hardware, it may often be easier within only a few years to simply re-run a simulation than to transfer large amounts of data; tracking provenance of data will allow this kind of re-creation.

Data types and (meta)data standards. We will use the following (highly portable) file formats:

- *Source files:* Standard text files, formatted to either project's coding standards, and following the relevant C++ standards such as ISO/IEC 14882.
- *Documentation:* HTML files, plain text files, or files that follow the format required by documentation generation programs such as doxygen.
- *Input (control) files:* In the case of ASPECT, these are typically plain text files following the formats understood by the input file parsers of deal.II.
- *Data files:* ASPECT can ingest a variety of data files, for example for initial temperature fields, for plate motions/surface velocities, or for meshes. These can either be provided in simple text-based formats documented in the ASPECT manual, or are parsed from the formats documented in the software packages that produce them (e.g., in the case of mesh generators). Where we have a say in the definition of these file formats – for example for the format in which we will make the probabilistic initial conditions available –, we generally strive for either simple file formats that are both easy to document, generate, and parse if they need to be human-readable; or for XML-based formats with appropriately defined schemas. In contrast, we avoid file formats that are simply representations of how data is laid out in memory for a concrete implementation.
- *Output files:* ASPECT uses the deal.II facilities for graphical output, which includes a large number of formats understood by common visualization programs. Of particular importance among them and the most commonly used today are the legacy VTK and the XML-based VTU formats understood by the Visualization Toolkit (VTK) on which many of the more modern visualization programs are built (e.g. Visit or Paraview). Both the VTK and VTU formats are well-documented quasi-standards and are widely used in scientific computing. The VTU format supports compression, which makes it particularly attractive for long-term storage. Likewise, we can output data in HDF5, another standardized and widely supported file format.

Privacy issues and commercial interests. None of the data produced raise any ethical or privacy issues. While there may be possibilities for commercializing the source code of ASPECT (or at least of parts of it), we have consciously chosen not to do so and instead release our codes under open source licenses as we have done for many years. This is in accordance with our goal of providing widely usable software and has been approved by our employers.

Policies for access and sharing. As mentioned above, source code and documentation generated through this proposal will be made available through web pages and can be downloaded by everyone. This is true both for periodic releases and through public anonymous access of source repositories. All such data is and will be available centrally from the ASPECT project website, with links to other services that provide data for the project (such as the github account).

Data produced from simulations and the corresponding input files, will be made available without restriction through the UNAVCO services mentioned above, as well as possible other venues. We will also state such in publications. As much as possible, we will provide DOIs for both software and data sets to allow users to find them easily and universally. For software, we have been doing this via Zenodo for several years already. For data, UNAVCO will provide this service for us.

Long-term archival. ASPECT has used version control software (subversion, and now git) since the beginning of the project, nearly 10 years ago. Consequently, all generated source code and documentation is available in a versioned form. There is no question that the project will continue to use such systems indefinitely, thereby preserving the provenance of source code as long as the project is alive, and certainly throughout the duration of this project. We do not anticipate that the ASPECT project will cease to continue after the end of this proposed project in 2024. However, should this nonetheless happen, then we will ensure that the publicly available releases and as much of the ancillary documentation as possible will continue to be accessible even after the end of the project, for example by freezing the state of our web servers and long-term archival of this state. We will in that case also ensure that the version history is preserved, for example by keeping open the github account.

Both UNAVCO and we use servers for all data we store that are regularly backed up both on- and off-site. Services such as github.com have their own backup strategies that can be assumed to be safe.