Geosci. Model Dev., 12, 2215–2225, 2019
https://doi.org/10.5194/gmd-12-2215-2019
© Author(s) 2019. This work is distributed under
the Creative Commons Attribution 4.0 License.

Geoscientific Model Development
Open Access
EGU

Editorial: The publication of geoscientific model developments v1.2

GMD executive editors

David A. Ham: Department of Mathematics, Imperial College London, London SW7 2AZ, UK;
Julia C. Hargreaves (chief-executive editor): Blue Skies Research Ltd, The Old Chapel, Albert Hill, Settle, BD24 9HE, UK;
Astrid Kerkweg: Institute for Geoscience and Meteorology, University of Bonn, Bonn, Germany;
Didier M. Roche: Laboratoire des Sciences du Climat et de l'Environnement, LSCE/IPSL, CEA-CNRS-UVSQ, Université Paris-Saclay, Gif-sur-Yvette, France; Earth and Climate Cluster, Faculty of Science, Vrije Universiteit Amsterdam, Amsterdam, the Netherlands;
Rolf Sander: Air Chemistry Department, Max Planck Institute of Chemistry, P.O. Box 3060, 55020 Mainz, Germany

Correspondence: GMD executive editors (gmd-executive-editors@mailinglists.copernicus.org)

Published: 6 June 2019

Abstract. Version 1.1 of the editorial of *Geoscientific Model Development* (GMD), published in 2015 (GMD Executive Editors, 2015), introduced clarifications to the policy on publication of source code and input data for papers published in the journal. Three years of working with this policy has revealed that it is necessary to be more precise in the requirements of the policy and in the narrowness of its exceptions. Furthermore, the previous policy was not specific in the requirements for suitable archival locations. Best practice in code and data archiving continues to develop and is far from universal among scientists. This has resulted in many manuscripts requiring improvement in code and data availability practice during the peer-review process. New researchers continually start their professional lives, and it remains the case that not all authors fully appreciate why code and data publication is necessary. This editorial provides an opportunity to explain this in the context of GMD.

The changes in the code and data policy are summarised as follows:

– The requirement for authors to publish source code, unless this is impossible for reasons beyond their control, is clarified. The minimum requirements are strengthened such that all model code must be made accessible during the review process to the editor and to potentially anonymous reviewers. Source code that can be made public must be made public, and embargoes are not permitted. Identical requirements exist for input data and model evaluation data sets in the model experiment descriptions.

– The scope of the code and data required to be published is described. In accordance with Copernicus' own data policy, we now specifically strongly encourage all code and data used in any analyses be made available. This will have particular relevance for some model evaluation papers where editors may now strongly request this material be made available.

– The requirements of suitable archival locations are specified, along with the recommendation that Zenodo is often a good choice.

In addition, since the last editorial, an "Author contributions" section must now be included in all manuscripts.

1 Introduction

Geoscientific Model Development has policies which attempt to ensure that the source code for the model developments that are published is publicly available. Why have these policies? The answer to this question is important not only to justify the responsibilities that it places on authors but also to inform authors, reviewers and the wider scientific community of current best practice in the publication of source code and data.

The importance of open science and open data has become increasingly recognised in the scientific community. Since the last editorial in GMD (GMD Executive Editors, 2015), many other journals have made positive steps to encourage increased openness (e.g. Brewer, 2017; Editor, 2016; Baker,

2016). Here we focus on the particular issues that arise when the data are predominantly model code.

The short version of the argument is given by the motto of the Royal Society: "nullius in verba" ("take nobody's word for it"). Scientists who publish a result without publishing the calculations undertaken to achieve that result are requiring the world to take them at their word, which is inimical to the scientific method, and the only really effective way to publish the calculations behind a computer model is to publish the source code (Añel, 2011). Thus, for those GMD papers focussed on the development of models or on development of model analysis methods, the descriptive text part of the manuscript is incomplete if not partnered by source code.

Throughout this editorial, "must" means that the stated practice is required and that manuscripts which fail to comply will be rejected; "should" means that the practice is strongly encouraged, and authors will need to provide defensible reasons in cases where manuscripts do not comply.

2 Why the publication of geoscientific model description demands the publication of code

Geoscientific models exist to enable computational experiments to be conducted. Those experiments take mathematical systems which are believed to represent important features of some geoscientific system and calculate the predicted behaviour of that mathematical system, in order to gain insight into the real system, in order to either better understand the system or to predict its response to forcing. The hypothesis is as follows:

- This mathematical system captures sufficient features of the system such that useful predictions or insights about the geoscientific system can be drawn from it.

Where this hypothesis has survived extensive experimental testing, one might describe the mathematical system as, in some sense, a good model for the physical system. Establishing this places strong demands on the modelling process. In particular there must be sufficient evidence to demonstrate the following:

1. The model code faithfully represents the mathematical and theoretical specification of the model.
2. The computation that was conducted used the code, input data and configuration as intended.
3. The analysis of the model output was appropriate and performed correctly.

Publication of source code most directly addresses the first of these, whilst the second and third can be greatly aided by properly specifying precise versions of code and input data in GMD manuscripts.

For example, suppose a scientist develops a new advection scheme and implements it in an atmospheric dynamics model. They publish a paper describing the new scheme and present the results of both idealised verification experiments and realistic simulations showing an improvement in forecast skill, but the code is not released. While this may appear to be a valuable contribution to modelling science, the complete algorithm including all approximations is needed in order to appreciate the sufficiency of the solution to the mathematical model. Even this would not be fully sufficient because all code contains unintentional errors (bugs), and even in the best code these will somewhat influence the results obtained. These issues are expanded on in the following paragraphs.

Space in journal papers is limited, as is the ability of readers to absorb information and the time authors can spend preparing publications. This means that it is normal for even simple models to be significantly underspecified in the paper itself. At the more complex end, it is not uncommon for an entire general circulation model of perhaps a couple of a million lines of code to be described in a 20–30-page paper (including verification and some evaluation). This obviously results in many details being omitted from the paper. Papers may omit discussion of boundary conditions or only deal with simplified cases. More seriously, model code frequently contains important details that are not mentioned in the paper, such as error traps that prevent the model from crashing. Recording every implementation detail in the paper would both be impractical and would undermine the paper as a mechanism for communicating the core ideas behind a model in a manner intelligible to readers.

Bugs occur in essentially all non-trivial code. Good software engineering practices such as code review (Rigby and Bird, 2013) and verification experiments (Farrell et al., 2011) can help reduce their incidence but can never provably eliminate them. For example, undesirable behaviour may only occur in regimes that were not exercised by the test suite. The code may therefore unintentionally fail to implement the mathematics described, and this may not be apparent from the tests run so far.

Both underspecification and bugs result in model code whose behaviour can differ from the mathematical system that the author and/or reader believes is being run. Suppose a reader finds a surprising result in a published simulation using this model. What should they do? Assuming enough resources the reader could attempt to re-write the model: except that underspecification prevents this. Even if they achieve this or employ one or more different models of the same physical system, the most they will be able to achieve is to conclude that another model fails to reproduce the original result. Even given two codes attempting to implement the same algorithm, there is no guarantee that the results will be the same, and without source code it is impossible to establish the underlying cause of the differences. With the source code, however, readers can gain much deeper knowledge of the model than can ever be described in the paper.

The hypothesis highlighted above is easily adapted to all the other forms of coded products described in GMD papers, such as data assimilation systems, frameworks, databases and model evaluation tools. Although not always emulating a physical system, in these products there is an underlying mathematical structure that is realised in code, and it must be demonstrated to work as intended in just the same way as a model. For model experiment descriptions, the situation is very simple. The primary purpose of these papers is to enable modelling communities to perform the same experiments. Therefore, everything required to run the experiment must be provided, apart from the model itself.

3 Further steps towards best practice

3.1 Source code is necessary but not sufficient

In order for the reader to have some confidence that the hypotheses above hold, it is not sufficient that the source code is provided. It is also necessary to have access to all of the input data and to know all of the steps which were taken from raw data to points on graphs or numbers in tables. This also implies that all model configuration files are provided.

3.2 Manual processing considered harmful

A particular challenge to understanding and reproducing results occurs where model inputs or outputs have been manually processed by an author. This frequently occurs when figures are produced interactively. Unfortunately, this breaks the provenance chain between the model and the paper: nobody, not even the author, can definitively know what processing was done to the data and therefore how the results came about. The only remedy for this is for authors to consistently ensure that there is no manual processing of the data: models are run by a script, and all pre- and post-processing is scripted. These scripts themselves should then be archived and cited from the paper. All figures and tables must be scientifically reproducible from the scripts.

3.3 Source code or data may be unavailable

The preceding sections make the case that publication of source code and associated data is a necessary part of the scientific method. Nonetheless, it is the case that the authors of some papers may not be able to publish their code or data. For example, this can occur when their institution owns the copyright and refuses to allow the code or data to be licensed in a manner which enables open archiving, or it may be the case that the paper authors are dependent on code or data whose copyright is owned elsewhere and for which they do not have a licence to redistribute. In particular, some of the current Earth system models have restricted licences controlled by large institutions. In other cases, model input or output may simply be too large to be uploaded to any open archival system that is available to the authors.

This presents a challenge: should GMD insist on publication of source code and data and therefore not publish papers about some of the most important models in the geosciences, or should it accept such papers even though the rigour of those papers is compromised by the lack of code? At the time of writing, the GMD editors considered that the balance falls on the side of allowing publication. However this does compromise the scientific standards of the journal, so the circumstances in which source code publication will not be required must be drawn as narrowly as is feasible. All manuscripts must at a minimum provide confidential access to the code and data developed in the manuscript for the editor and reviewers in order to enable peer review (see Appendix A1).

This position is broadly consistent with the American Geophysical Union publications data policy[1] though less strict than, for example, *Computers and Geosciences*, who now have an open-source-only policy[2].

In the case where the new code and data described in the paper are not restricted but are part of a larger code and data structure that has other restricted elements, it is still possible to satisfy the GMD requirements by making the new parts of the code and data available. As the result is usually not a coherent model which can be expected to compile, authors sometimes prefer to upload these code fragments to the supplement rather than to a repository. Authors may have to remove restricted elements from their model code base, but in the meantime this remains an acceptable if somewhat unsatisfactory solution.

3.4 Embargoes

Recently, some authors have offered to provide public code access only on acceptance of the final manuscript or after a defined period, such as 12 months. In these circumstances, it is clear that the authors are able to provide code access. By preventing public access to the code during open review and/or during the immediate period after publication, they are impeding the scrutiny of their work at the most critical points in the publication process. Having determined that source code publication is a necessary part of the publication process, GMD does not permit embargoes on code release. It is the opinion of the GMD editors that if the code is not ready, then neither is the manuscript. Therefore, if the code is not subject to licensing or other restrictions preventing its ultimate release, it must be made available to the editor and reviewers upon submission.

[1] https://publications.agu.org/author-resource-center/publication-policies/data-policy/ (last access: May 2019)

[2] https://www.journals.elsevier.com/computers-and-geosciences (last access: May 2019)

3.5 Archive on submission

A related issue to embargoing is raised by manuscripts which point to code located on a website, accompanied by a promise in the cover letter that the code will be properly archived when the final paper is accepted. This approach compromises the review process for two reasons. First, at the core of the open review process of European Geosciences Union journals is the idea that the submitted manuscript and the review process are openly and persistently available to scrutiny. The version of the code which matches the submitted manuscript directly influences that process, so it must remain available to anyone who wishes to examine the review of the manuscript.

Second, the whole review process, including initial editor review, external referees and executive editor intervention when needed, is in essence a protracted quality control process for the scientific content of the manuscript but also for the technical requirements such as code and data archiving. If authors are permitted to delay compliance with technical requirements, then this effectively disables most of the layers of this quality control process. For this reason, the full archival requirements must be met by manuscripts on submission, and handling topical editors should require revised submissions by authors until these requirements are met. This provides reviewers and executive editors with the opportunity to act as a backstop check during the review process, thereby minimising the chance of a final paper being published which fails to meet the code and archiving requirements.

The key objection that some authors hold to archiving on submission is that it can result in old, uncorrected, versions of the code being persistently available online. There are mechanisms available to assuage this concern. First, the code and data availability section in the manuscript can and should identify the preferred download location for the latest model version, in addition to citing the archived version corresponding to the given publication. Second, archives such as Zenodo support the archival of successive versions of data sets and flag to the reader that a newer version is available. Further, the archive metadata can also direct readers to the preferred download location.

4 Journals and archives

A possible response to the issues caused by lack of access to the source code is to go back to the authors and ask them to provide code or otherwise assist in investigating a surprising result. The success of this approach is dependent on the authors being willing and able to assist. The latter is a particularly difficult problem: if code has not been curated sufficiently well, it may not be possible to recover the exact version used in a paper; the scientist who did the work may have moved on, or the resources (human or computational) required for assistance may not be available.

This undermines the role of a journal as a persistent, public and definitive archive of scientific results. In order for journals to fulfil this function, it must be possible for readers to trace the provenance of the results presented without further assistance by the authors. For source code (and other data) to fulfil this role, the data archive must be as persistent, public and definitive as the journal itself. Relatively small models, data sets or documents can often be uploaded as supplements to the model and stored alongside the paper itself. However this approach is impractical for larger data sets. It is also not always clear that the very high standards of long-term preservation that journal publishers provide for articles are also applied to supplements.

4.1 Archival requirements for code and other data

There are three highly desirable features of an archival system which is to be used for the code and other data on which a journal publication depends:

1. *Institutional persistence.* The archive's institutional arrangements and financial support or business model must be such that one can be reasonably confident that the archive will remain and be publicly accessible for many years/decades into the future.

2. *Irrevocability.* It must not be possible for the authors of the archived data to unilaterally remove it. Copyright infringement or other important considerations may sometimes require material to be removed from an archive, but this must involve an independent editorial decision.

3. *Persistent identifiers.* It must be possible to unambiguously refer to the data in a manner that does not depend on impermanent implementation details such as the arrangement of the archive's web interface. The usual standard mechanism for archives to meet this requirement is by issuing DOIs (digital object identifers) for archived items, although alternative unique identifiers with expected persistence on the timescale of decades may also be acceptable.

4.2 Examples of suitable archives

4.2.1 Zenodo

Zenodo is a scientific data archive funded by the European Union and hosted by CERN. It has an institutional guarantee of at least 20 years of future support and has policies restricting withdrawal of deposited data[3]. Zenodo issues DOIs for all deposits and supports DOI versioning to connect together

[3]http://about.zenodo.org/policies/ (last access: May 2019) V1.0 was current at time of writing.

successive versions of a data set[4]. A feature of Zenodo which makes it particularly easy to use for many GMD authors is its integration with GitHub[5]. It is straightforward for the owner of a GitHub repository to link their GitHub and Zenodo accounts and have a Zenodo archive created automatically for every GitHub release[6].

4.2.2 arXiv

On occasion, GMD requires authors to archive grey literature, for example, technical reports about a previous model version. Where these are simply a single document, Zenodo may not be the most convenient archival system. In these cases the arXiv preprint server is a good choice[7]. The arXiv has a long track record dating back to 1991, institutional support from Cornell University and financial backing from a large worldwide consortium of research institutions. The arXiv has very strict policies against removal of articles[8]. DOIs are not issued, but there is a system of arXiv identifiers which fulfil a technically equivalent role of providing a persistent, location-independent reference for an arXiv document[9].

4.2.3 National or discipline-specific data archives

At the time of writing, Zenodo had an upload limit of 50 GB. This is more than enough for source code and indeed for the input data for many GMD papers. However some papers, especially evaluation papers for complex models, may require much more than this. Because the archival of large data sets is expensive, there is no generic free solution to this case: the appropriate repository may depend on the application area of the model in question, the source of funding for the research or the location of the authors' institutions. The requirements of Sect. 4.1 still apply and must be assessed in each case. Useful guidance may be drawn from the lists of recommended repositories provided by Springer Nature[10], PLOS[11] and *ESSD*[12].

4.3 Approaches which do not meet archival requirements

4.3.1 Institutional websites

Hosting source code and other data on the authors' institutions' websites fails to satisfy the requirements for irrevocability and persistence of reference. Even large institutions periodically refresh their web presence with the result that URLs change, and data from long-finished projects may not be preserved. Of course data archives also have web interfaces, and some authors work for institutions that host suitable data archives. Authors are not debarred from using a data archive which complies with GMD policy simply because they are affiliated with the archive's host organisation. However by the same token, the fact that an organisation hosts a suitable archive does not mean that organisation's entire web presence can be considered an archival location.

4.3.2 GitHub and other online revision control systems

In the last few years, many authors have submitted manuscripts whose code availability sections consist of direct links to online revision control systems, either institutionally hosted or online services such as GitHub[13], GitLab[14] or Bitbucket[15]. These services are excellent platforms for developing models. They provide revision control and support code review, distribution and integration with continuous testing systems. Authors are encouraged to use such platforms for code development. They are, however, insufficient for publication as they lack the required persistence and irrevocability: if the project ends or decides to move to a different hosting platform, the data will disappear from the location given in the paper. In addition, if the project moves to a different revision control system, the version indicators provided may well cease to be valid. A more persistent solution is required. As already noted in Sect. 4.2.1 it is straightforward for the owner of a GitHub repository to link their GitHub and Zenodo accounts and have a Zenodo archive created automatically for every GitHub release.

5 Conclusions

The main purpose of this editorial is to provide greater clarity on the code and data policy at GMD. The most significant change is that all source code (including input data for model description papers and data sets for evaluation data set description papers) must be made available to both the editor and reviewers.

In Appendix A to this editorial we include the revised code and data policy information, which replaces the previ-

[4] http://help.zenodo.org/\T1\textbackslash#versioning (last access: May 2019)

[5] https://github.com (last access: May 2019)

[6] https://guides.github.com/activities/citable-code/ (last access: May 2019)

[7] https://arxiv.org (last access: May 2019)

[8] https://arxiv.org/help/withdraw (last access: May 2019)

[9] https://arxiv.org/help/arxiv_identifier (last access: May 2019)

[10] https://doi.org/10.6084/m9.figshare.1434640.v11 (last access: May 2019)

[11] https://fairsharing.org/recommendation/PLOS (last access: May 2019)

[12] https://www.earth-system-science-data.net/for_authors/repository_criteria.html (last access: May 2019)

[13] https://github.com (last access: May 2019)

[14] https://gitlab.com (last access: May 2019)

[15] https://bitbucket.org (last access: May 2019)

ous GMD-specific text on the code and data policy web page. There are also some small changes made to the paper types page, the revised version of which is included in Appendix B.

Appendix A: GMD code and data availability policy

A1 Core principles

Every paper must include a section at the end of the paper before the "Acknowledgements" entitled "Code availability" or "Code and data availability" as appropriate.

1. This section must include citations for the persistent public archives of the precise versions of all of the code and data associated with the paper. The generic means to access other versions of the code and data as well as the licence of the code should also be explained. The licence should conform to the Open Source Definition[16]. Suitable licences[17] are for example GPL[18] or MIT[19].

2. Where the authors cannot, for reasons beyond their control, publicly archive part or all of the code and data associated with a paper, they must clearly state the restrictions. They must also provide confidential access to the code and data for the editor and reviewers in order to enable peer review. The arrangements for this access must not compromise the anonymity of the reviewers. All manuscripts which do not make code and data available at this level are to be rejected. Where only part of the code or data is subject to these restrictions, the remaining code and/or data must still be publicly archived. In particular, authors must make every endeavour to publish any code whose development is described in the manuscript.

Code and data access must be provided at the time that the discussion paper is submitted. Embargoes, whether pending acceptance or for a defined period, are not acceptable.

A2 Scope

The code and data associated with a paper which are subject to the above requirements include, depending on the paper type, the following:

1. the source code for the complete model or module or other coded product described in the paper (must be provided for model description, development and technical, and methods for assessment paper types);

2. the manual and any other model documentation (applies to model description, development and technical, and methods for assessment, to the extent the editor considers applicable);

3. all configuration files, boundary conditions, and input data (must be provided for experiment description papers and any other papers in which results from model runs are reported);

4. data sets for forcing of models or comparison with model output (must be provided for papers describing such data sets or for papers in which model output are compared with such data);

5. preprocessing, run control and postprocessing scripts covering every data processing action for all the results reported in the paper (applies for all papers, to the extent the editor considers applicable).

In every case, the citation from the paper must identify the exact version of the code and/or data used.

Although the code and data will not be reviewed formally, the editor and reviewers are free to make general comments on any code or data, if they so wish. During the review process, the ease of model download, compilation, and running of test cases may be assessed.

A3 Archive standards

A frozen version of the code and data as developed in the paper must be archived. Usually, a third-party archive is preferable. In some cases, such as when the code is a fragment from a larger model, authors may include the code in the supplement to the paper. Third-party archives must have the following:

1. institutional support providing reasonable confidence that the material will remain available for many years/decades

2. mechanisms preventing the depositor of the material from unilaterally removing it from the archive

3. mechanisms for identifying the precise version of the material referred to in a persistent way. This will usually be a DOI.

Where code and data change during the revision process of the manuscript, the updated versions must also be archived. Authors must take care that the results in revised manuscripts are correctly associated with the corresponding archived data (with different DOIs referenced in the submitted and final manuscripts in cases where data have changed).

Many GMD authors find Zenodo[20] a suitable archival location. Zenodo's GitHub integration[21] makes archiving particularly easy for the large proportion of authors who manage their code using Git. Authors who need to archive a single documentation file, such as a technical report, may

[16] http://www.opensource.org/docs/osd (last access: May 2019)

[17] http://www.opensource.org/licenses/alphabetical (last access: May 2019)

[18] https://opensource.org/licenses/GPL-3.0 (last access: May 2019)

[19] https://opensource.org/licenses/MIT (last access: May 2019)

[20] https://zenodo.org (last access: May 2019)

[21] https://guides.github.com/activities/citable-code/ (last access: May 2019)

find the arXiv suitable[22]. Authors whose data are too large to be archived at Zenodo will need to identify a suitable alternative. Appropriate choices may depend on the topic of the paper, the funder of the research, and the country where the research was conducted. One of the repositories listed by Springer Nature [23], PLOS[24] or *ESSD*[25] may be suitable. In any case, the requirements above must be satisfied.

Project or institution websites and online revision control sites such as GitHub[26], GitLab[27] or Bitbucket[28] are made for code development but not suitable for archiving frozen code versions. Authors are encouraged to provide links to a website or revision control system as a preferred download location, so long as this is in addition to, and not instead of, the citation of an archive.

A4 Template for code and data availability section

The following code and data availability section meets the requirements of this policy for papers focussed on development of models or development of methods for assessment of models. Other wordings are, of course, possible so long as the required information is all present. For larger models it is very helpful if authors can identify the location of the main parts of the code that are discussed in the manuscript. For experiment description papers, evaluation papers, and some technical and development papers where details for a variety of different data sets or models are required, the section will be considerably longer.

> The current version of *model* is available from the project website: *url* under the *licence name* licence. The exact version of the model used to produce the results used in this paper is archived on Zenodo *(citation)*, as are input data and scripts to run the model and produce the plots for all the simulations presented in this paper *(citation)*.

In line with the FORCE11 Joint Declaration of Data Citation Principles, the data citations should appear in the bibliography and be referenced in the text in the same way as other publications (Martone, 2014).

Appendix B: Manuscript types

Below is the manuscript types web page content. Changes from Editorial 1.1 are in bold font.

In the following, "must" means that the stated practice is required and that manuscripts which fail to comply will be rejected; "should" means that the practice is strongly encouraged, and authors will need to provide defensible reasons in cases where manuscripts do not comply.

Code and/or data availability sections must be included in all papers and should be located at the end of the article, after the conclusions, and before any appendices or acknowledgements. **Source code must be published on a persistent public archive with a unique identifier or be uploaded to the supplement, unless this is impossible for reasons beyond the control of the authors.** For more details refer to the code and data policy.

There are seven different manuscript types accepted at GMD. During the submission process, authors will need to select the type which most closely matches the aims of their manuscript. The types are as follows:

- model description papers
- development and technical papers
- methods for assessment of models
- model experiment description papers
- model evaluation papers
- review and perspective papers
- corrigenda.

Updates: Minor version updates or correction of actual errors in a model, model development or experiment protocol should be submitted as a regular submission within one of the standard manuscript types. Authors may request that these form part of a model special issue including the previously published papers.

B1 Model description papers

Model description papers are comprehensive descriptions of numerical models which fall within the scope of GMD. The papers should be detailed, complete, rigorous, and accessible to a wide community of geoscientists. In addition to complete models, this type of paper may also describe model components and modules, as well as frameworks and utility tools used to build practical modelling systems, such as coupling frameworks or other software toolboxes with a geoscientific application. The GMD definition of a numerical model is generous, including statistical models, models derived from data (whether model output or observational data), spreadsheet-based models, box models, 1-dimensional models, through to multi-dimension mechanistic models.

- The main paper must give the model name and version number (or other unique identifier) in the title.

[22] https://arxiv.org (last access: May 2019)

[23] https://doi.org/10.6084/m9.figshare.1434640.v11 (last access: May 2019)

[24] https://fairsharing.org/recommendation/PLOS (last access: May 2019)

[25] https://www.earth-system-science-data.net/for_authors/repository_criteria.html (last access: May 2019)

[26] https://github.com (last access: May 2019)

[27] https://gitlab.com (last access: May 2019)

[28] https://bitbucket.org (last access: May 2019)

- The publication should consist of three parts: the main paper, a user manual, and the source code, ideally supported by some summary outputs from test case simulations.
- The main paper should describe both the underlying scientific basis and purpose of the model and overview the numerical solutions employed. The scientific goal is reproducibility: ideally, the description should be sufficiently detailed to in principle allow for the re-implementation of the model by others, so all technical details which could substantially affect the numerical output should be described. Any non-peer-reviewed literature on which the publication rests should be either **made available on a persistent public archive, with a unique identifier, or** uploaded as supplementary information.
- The model web page URL, the hardware and software requirements and the licence information should be given in the text. If papers are describing subsequent development to a paper already published in GMD, **authors should request them to** be electronically linked to the previous version(s) in a special issue, and an overview web page will be created.
- The model description should be contextualised appropriately. For example, the inclusion of discussion of the scope of applicability and limitations of the approach adopted is expected.
- Examples of model output should be provided, with evaluation against standard benchmarks, observations, and/or other model output included as appropriate. In this respect, authors are expected to distinguish between verification (checking that the chosen equations are solved correctly) and evaluation (assessing whether the model is a good representation of the real system). **Sufficient verification and evaluation must be included to show that the model is fit for purpose and works as expected**. Where evaluation is very extensive, a separate paper focussed solely on this aspect may be submitted.
- **Code must be published on a persistent public archive with a unique identifier for the exact model version described in the paper or uploaded to the supplement, unless this is impossible for reasons beyond the control of authors.** All papers must include a section, at the end of the paper, entitled "Code availability". Here, either instructions for obtaining the code, or the reasons why the code is not available should be clearly stated. For established models, there may be an existing means of accessing the code through a particular system. In this case, there must exist a means of permanently accessing the precise model version described in the paper. **Making code available through personal websites or via email contact to the authors is not sufficient**. After the paper is accepted the model archive should be updated to include a link to the GMD paper.
- When **code cannot be made public,** topical editors **and reviewers** must still be given access to the model code.
- Although the source code and user manual will not be reviewed formally, the editors and reviewers are free to make general comments on the code if they so wish. During the review process, the ease of model download, compilation and running of test cases may be assessed.

B2 Development and technical papers

These papers describe technical developments relating to model improvements such as the speed or accuracy of numerical integration schemes as well as new parameterisations for processes represented in modules. Also included are papers relating to technical aspects of running models and the reproducibility of results, e.g. assessments of their performance with different compilers, or under different computer architectures. In addition, papers focussing on data assimilation are welcome. Development and technical papers usually include a significant amount of evaluation against standard benchmarks, observations, and/or other model output as appropriate.

In the case where new code is described in the paper, this is subject to the same availability requirements as for complete model descriptions. The code should be made available, and a model availability paragraph must be included.

If the model development relates to a single model, then the model name and the version number must be included in the title of the paper. If the main intention of an article is to make a general (i.e. model independent) statement about the usefulness of a new development, but the usefulness is shown with the help of one specific model, the model name and version number must be stated in the title. The title could have a form such as, "Title outlining amazing generic advance: a case study with Model XXX (version Y)".

B3 Methods for assessment of models

Methods for assessment of models include work on developing new metrics for assessing model performance and novel ways of comparing model results with observational data. Also included are discussions of novel methods for data analysis, visualisation with relevance to geoscientific modelling, or the application of existing techniques to this field. These papers may also be theoretical, in which case an example implementation should be provided as supplementary information. They may also be based on the description of a fully fledged software tool.

The process of analysing model output for comparison with data may involve algorithms similar to those implemented in complex numerical models. In these cases, model

output is input to another model in order to produce output comparable to observed quantities. Papers describing these algorithms may be submitted as either methods for model assessment or model description papers.

Descriptions of software tools are subject to the same criteria as model descriptions (name and version must be identified in the title, code must be supplied for the peer-review process, etc.), and a code availability paragraph must be included in the manuscript.

B4 Model experiment description papers

Model experiment description papers contain descriptions of standard experiments for a particular type of model, such as might be used in a MIP **(model inter-comparison project)**. Configurations and overview results of individual models can also be included as well as descriptions of the methodology of experimental procedures such as ensemble generation. Such papers should include the discussion of why particular choices were made in the experiment design and sample model output. In the case of papers describing MIPs, they should explain any specific project protocols, should highlight differences in the application of the protocol by the different groups, and should include sufficient descriptions/figures of model results to give an overview of the project. For model experiment description papers, similar version control criteria apply as to model description papers: the experiment protocol should be given a version number; boundary conditions should be given a version number; **a data availability paragraph must be included in the manuscript;** and links to the GMD paper should be included on the experiment website. **Since the primary purpose of these papers is to make experiments accessible to the community, all input data required to perform the experiments must be made publicly available.**

Papers describing data sets designed for the support and evaluation of model simulations are within scope and included in this paper type. These data sets may be syntheses of data which have been published elsewhere. The data sets must also be made available, and any code used to create the syntheses should also be made available.

B5 Model evaluation papers

Model evaluation is an important component of most GMD papers. Model development papers in particular often include a large proportion of evaluation. Typically, this comprises a comparison of the performance of different model configurations or parameterisations. In some cases, the evaluation is sufficiently substantial that a stand-alone paper is required. In this case it is required that the model, model development, or model experiment has already been described in another paper (or that the description is also under review). **The model name and version number should be identified in the title.** The authors **must** provide the citation of the description paper in the evaluation manuscript itself and also in the letter to the editor when submitting an evaluation manuscript. If the description is in GMD, then there is the possibility of linking the papers, either in the form of a companion paper (e.g. Part 1 and Part 2) or as part of a special issue devoted to a particular model or experiment. **Preprocessing, run control and postprocessing scripts covering every data processing action for all the results reported in the paper should be provided for evaluation papers.**

It is, however, common for pure evaluation papers to contain substantial conclusions about geoscience rather than about models, and such papers are not suitable for submission to GMD. These are more likely to reach the appropriate audience in those EGU journals which publish scientific results related to the GMD subject areas.

B6 Review and perspective papers

Review and perspective papers summarise the status of knowledge and outline future directions of research within the scope of the journal.

Before preparing and submitting a review article, please contact the executive editors. A code and/or data availability section must be included. By default, the code and data availability requirements for models, experiments, code and data discussed in review papers are the same as for the other paper types, but in some cases deviations from this standard may be appropriate (for example, authors may need to discuss some code or data from external sources, for which they have no means of gaining or granting access). This should be discussed with the executive editors prior to submission of the paper.

B7 Corrigenda

Corrigenda correct errors in preceding papers. The manuscript title is as follows: Corrigendum to "TITLE" published in JOURNAL, VOLUME, PAGES, YEAR. Please note that corrigenda are only possible for final revised journal papers and not for the corresponding discussion paper. Corrigenda should only be used for correcting errors in the papers and not for those occurring in the model development being described.

Author contributions. The guidelines presented in this paper have been approved by all GMD executive editors. Most of text was written by DAH and JCH, with all other editors (AK, DMR, and RS) contributing to the discussion of the issues presented here at some length via frequent email communication. This editorial presents what we believe to be an achievable best practice for the journal over the next few years and as such does not represent the ideal for any of the executive editors.

Acknowledgements. The GMD executive editors gratefully acknowledge the GMD topical editors for their hard work, support and helpful comments on the manuscript. David A. Ham was supported by a United Kingdom Natural Environment Research Council Independent Research Fellowship [grant no. NE/K008951/1]. Didier Roche was supported by the Centre national de la recherche scientifique (CNRS) and by the Vrije Universiteit Amsterdam.

References

Añel, J. A.: The Importance of Reviewing the Code, Communications of the ACM, 54, 40–41, https://doi.org/10.1145/1941487.1941502, 2011.

Baker, M.: Why scientists must share their research code, Nature, News, https://doi.org/10.1038/nature.2016.20504, 2016.

Brewer, P.: Do you expect me to just give away my data, Eos, 98, https://doi.org/10.1029/2018EO081175, 2017.

Editor: Announcement: Where are the data?, Nature, Editorial, 537, https://doi.org/10.1038/537138a, 2016.

Farrell, P. E., Piggott, M. D., Gorman, G. J., Ham, D. A., Wilson, C. R., and Bond, T. M.: Automated continuous verification for numerical simulation, Geosci. Model Dev., 4, 435–449, https://doi.org/10.5194/gmd-4-435-2011, 2011.

GMD Executive Editors: Editorial: The publication of geoscientific model developments v1.1, Geosci. Model Dev., 8, 3487–3495, https://doi.org/10.5194/gmd-8-3487-2015, 2015.

Martone, M. (Ed.): Data citation synthesis group: Joint declaration of data citation principles, FORCE11, https://doi.org/10.25490/a97f-egyk, 2014.

Rigby, P. C. and Bird, C.: Convergent Contemporary Software Peer Review Practices, in: Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2013, 202–212, ACM, New York, NY, USA, https://doi.org/10.1145/2491411.2491444, 2013.