MAKING REPRODUCIBLE COMPUTATIONAL RESEARCH A REASONABLE CHOICE FOR YOUNG FACULTY ON TENURE TRACK

BENJAMIN SEIBOLD

Abstract. Conducting computational research in a reproducible way can be very rewarding in the long run. However, given the current situation of merit, journals, grant funding, etc., it is likely to have a negative impact on the research record in the short term. This leads to the crucial question: "How can a young researcher on tenure track (or on a temporary postdoctoral position) conduct reproducible computational research without having to take a hit on his research record?"

1. Background

I am a fourth year tenure track Assistant Professor in the Department of Mathematics at Temple University. I am involved in a variety of computational research projects, some of which are fully reproducible, others (still the majority) are more in the classical JCP-style of non- or semi-reproducible research. Playing in both of these camps, and being at a career stage where my current research record will be of crucial importance for my further career (→ tenure), I am experiencing the drawbacks that an investment into reproducibility incurs. At the same time: the older the dog, the harder it is to learn new tricks. It should therefore be of crucial importance for senior members of our profession to shape our research community (journals, merit, prizes, funding agencies, etc.) so that the incentives (and rewards) associated with reproducible computational research are increased.

2. Models of Computational Research for a Young Researcher

Young researchers, such as tenure track assistant professors, in computational research are likely to be deeply involved in code development aspects, and are "getting their hands dirty" in using computational resources. They are therefore most frequently in direct control of how to organize a software code and whether to make it publicly accessible. In my experience, there are three models (and gray zones in between):

(1) Write a single-purpose code, to be used for one particular task or publication; produce computational results for the paper; never use that code again.
(2) Write a code (or code segment) with the goal to re-use it in the future. Compared to (1), extra time must be invested to create a clean and sufficiently documented code so that one (or an informed collaborator) understands it a year later.
(3) Create a code (or code segment) to be shared publicly, so that everybody can use it and reproduce computational results published in a paper. Compared to (2), *a lot* of extra time must be invested to make the code usable and understandable by others, in particular if the code was not started with the goal to be shared, but instead grew from bits and pieces.

Date: November 30, 2012.

3. Recognition for a Young Researcher

Realistically, the main aspects that count in the record of a tenure track assistant professor are journal publications and grant funding. In contrast, other aspects are less crucial, such as: as long as one does a good job at teaching, things are okay here; also, as long as one attends conferences regularly, and shows some involvement in the department's committee work, the record looks fine here. Prizes and special awards are great, but it is probably not a good idea to plan a career based on which prizes one intends to win, particularly not before having tenure. And other aspects, such as industry projects or the creation of software products, do not hurt either, but they rarely make a big difference to the record.

Hence, unless the landscape of academic recognition changes beyond the scope of mathematics and/or computational science, the focus must be on journals and funding opportunities. Below, I address some problems and ideas for these two items. Moreover, there is the point of how to communicate the issue of reproducible computational research to university officials (department chairs, deans, provosts, etc.)

4. Journal Publications

The current situation presents the following problems for a young researcher in computational mathematics/physics/engineering:

(a) The vast majority of good journals in the field focus on pure research results. In particular, if the results are judged to not quite meet the bar for a publication, the fact that the results are fully reproducible rarely raises a manuscript back over the bar.
(b) The journals with the biggest names (or impact factors, in case this matters) tend to be journals of the type described in (a), such as JCP, CMAME, etc. In contrast, journals that give merit to code development and/or reproducibility, such as TOMS, do not quite have the same level of reputation (at least not in my research communities).
(c) Since in most journals, no requirements in terms of reproducibility are imposed, people can use parts of codes that other people have make publicly available, without giving proper (or any) credit to their sources, and there will be no way to recognize such a violation of good academic conduct from their publication.

For the young researcher, the most straightforward messages to be taken from these points are:

(a) Spend your time on producing more and better results, rather than cleaning up or documenting your codes.
(b) Do not bother sending manuscripts to journals that give merit to reproducibility. Instead, focus on journals that have the biggest names and/or impact factors.
(c) Do not make any codes (or pieces thereof) publicly available. Because if you do, other researchers may use them to publish results that otherwise you could have published yourself.

5. Grant Funding

For young researchers in mathematics/physics, NSF is the number one funding agency. In fact, often times, DMS Computational Mathematics or DMS Applied Mathematics are the only programs in which a young researcher without a significant funding track record has good chances of receiving funding.

Clearly, NSF proposals in DMS get reviewed on the basis of a variety of criteria besides the intellectual merit of the proposed research, such as education, training, collaborative/interdisciplinary nature, etc. However, the most important criterion is that the reviewers and panel members like the actual research itself; and most commonly, the proposed research is judged in the flavor of traditional journal publishing. A promise to make the computational research reproducible is unlikely to push a proposal from "declined" to "awarded".

With other funding agencies the situation is worse, in the sense that applicants tend to be judged based on their past record more than with NSF. And it is hard to establish a track record in reproducible research within a short time frame.

6. Awareness of University Officials about Reproducibility

Assume that, despite the difficulties with journals and grant funding, a young researcher on tenure track has a record in reproducible research and/or software development, but in turn fewer publications than he/she would have had when ignoring the reproducibility component. The question is now whether the record in reproducible research will be appropriately recognized by department chairs, deans, provosts, and tenure-and-promotion committee members from other departments.

Clearly, many differences between areas of research are present independent of the question of reproducible research. For instance, in some fields it is common to publish a lot of proceedings; in other fields it is common to publish papers with many authors, and the ordering of authors is of great importance; yet is other fields industry projects are common in addition to publications. It is therefore common practice that members of tenure-and-promotion committees inform each other about the difference in their respective fields of research.

The problem with reproducible computational research is that the push towards it is a relatively recent development. It is therefore very possible that nobody in a tenure-and-promotion committee be aware of this aspect.

There is an additional problem in mathematics: department chairs and senior faculty who work in pure mathematics may be unaware—and possible even unreceptive—to the merits in code development and the aspect of reproducibility in computational research. In fact, pure mathematicians might have a lower level of understanding for this issue than, for example, an experimental chemist or biologist, who clearly will understand the importance of reproducibility.

7. What Could be Done?

Some ideas of steps that could be taken to encourage more young researchers to conduct reproducible computational research, and also to make sure that the ones who do, receive appropriate merit for it:

(a) Faculty member in editorial boards of "traditional" computational journals (JCP, CMAME, etc.) should push for the explicit mention of reproducibility in the review criteria of manuscripts in this journal. Moreover, reviewers should be asked specifically to address the aspect of reproducibility in their review. EICs should incorporate these aspects in their decisions.
(b) More journals must be founded (or existing journals be modified) that have specific components in reproducibility. SISC is one example of such a journal. Journals could go as far as to require a section in each article that addresses the reproducibility of the presented results, similar to the "Materials and Supplies" section in lab-based research papers.
(c) The writing and reading of good code must receive a stronger attention in the mathematics curriculum. Moreover, an understanding must develop that modular and short codes in Matlab or Python are not intrinsically inferior to a long code project in C^{++}.
(d) NSF must create new programs specifically focused on reproducible research. In particular, these programs must be designed so that individual researchers at universities, who do not work within an established large code (as they are commonly developed at national labs), have a shot at receiving funding.
(e) Organizations in applied mathematics and computational science (SIAM, GAMM, ICIAM, etc.) should create various prizes associated to reproducible research and/or software development, particularly aimed at a smaller level rather than big many-person projects.
(f) The use of the computer via short scripts and elegant programs must penetrate deeper into the mathematics education even in theoretical courses, such as the use of Matlab's plotting in undergraduate calculus as well as graduate level geometry, or computer algebra systems in undergraduate and graduate level algebra and number theory courses. This way, an atmosphere will be created that the use of publicly available software (even if just a 20-liner in Matlab) is a standard component of mathematical research and education.
(g) Concept papers that explain the issue of reproducibility in computational research must be worked out, and must be distributed to mathematicians/physicists/engineers who do not conduct computational work. Specifically, such concept papers should be distributed via the AMS. Moreover, all mathematics department chairs should be specifically primed on this issue.

(Benjamin Seibold) Department of Mathematics, Temple University,
1805 North Broad Street, Philadelphia, PA 19122

E-mail address: `seibold@temple.edu`

URL: `http://www.math.temple.edu/~seibold`