

COMPJ-2022-05-0325: Response to referees

Harold Thimbleby

January 30, 2023

Thank you for this opportunity to respond to reviewers' comments. Almost all of my responses, detailed below, have resulted in helpful improvements to the paper.

I have written my responses to the reviewers' comments in red. If it helps in referring to them in future correspondence, I have numbered my responses.

In addition to responding to reviewers' comments, I have also edited the paper to improve its clarity, structure, and where possible to reduce its length. The writing is now tighter, and in particular the reviewers' repeated concern about "opinion" has been addressed by appropriate rewriting — mostly replacing "shoulds" with "coulds" and other ways of making clear the what was an unqualified opinion is a suggestion for further work.

Dear Professor Thimbleby,

COMPJ-2022-05-0325 – Maturing the computational sciences

I am writing to you with regard to your paper COMPJ-2022-05-0325, which you submitted to The Computer Journal. It has now been examined by referees and by a member of the editorial board.

In view of their recommendations, I would appreciate it if you would revise your paper, carefully taking into account all the comments, and resubmit it by 03-Jan-2023 via the Manuscript Central website at <http://mc.manuscriptcentral.com/compj>

Please also make sure a DATA AVAILABILITY STATEMENT is included in the manuscript. Please check

https://academic.oup.com/comjnl/pages/Manuscript_Preparation_Submission and https://academic.oup.com/journals/pages/authors/preparing_your_manuscript/research-data-policy#data2

To submit your revision, please go to *** PLEASE NOTE: This is a two-step process. After clicking on the link, you will be directed to a webpage to confirm. ***

https://mc.manuscriptcentral.com/compj?URL_MASK=ae7f1f867b624f72b0d553807c8d9146

Please also submit a list of the changes you have made in response to each of the referees' comments. Your paper will then be re-examined and a decision will be made about whether it may be published.

Thank you very much.
Best regards,
Deputy Editor
The Computer Journal

1 REVIEWERS' COMMENTS:

1.1 Reviewer: 1

The paper has a misleading title – I think it would be better if it focussed on software and people – or maybe on improving reliability.

1 | Thanks to this concern, I have improved and changed the title to “Improving science that uses code.”

Having a vision for the future is good, but it needs to be matched with a set of practical changes to get there. I don’t see that here, just a vision of an alternative world.

I think this paper is too long and too complex.

2 | As well as structural improvements, I have also added a short table of contents, which makes the structure of the much paper much clearer than before, also helping address this reviewer’s concern. In fact, I used the new table of contents to guide some streamlining throughout the paper!

Some of it is opinion, some of it is observation, and some of it is advertising. And these parts are interspersed in the paper.

In table 2, GitHub is listed as both a code repository and a data repository, which is confusing. It might be better to talk about the different meanings of repository, such as in the git sense vs in the library/archive sense.

3 | GitHub can be used as a repository for anything (such as *The Turing Way: A Handbook for Reproducible Data Science*, which is a massive project with 419 authors collaborating. I have therefore not changed table 2.

There is likely a correlation between good software engineering practices and good code quality, but I am less sure that there is a causation, or that good quality code cannot be written without good software engineering practices.

Re “A common oversight in scientific papers is to present a model, such as a set of differential equations, but omit how that model is transformed into code that generates the results the paper summarizes; if so, the code may have problems that cannot be identified as there is no specification to reference it to. – this reminds me of <https://doi.org/10.12688/f1000research.25561.1>

4 | I had thought of citing this f1000 paper, but it reports on a meeting duplicating the earlier work of TOP, which I do cite; furthermore this f1000 paper just makes a list of questions — unlike the TOP papers which have concrete proposals.

There is likely a difference between how statistical practices are described concisely in a paper (page 3 line 54) and how software can be described, which is probably better done through the software itself than through text about the software.

- 5 | Why? Just as a statistical transformation might be described by name in wording like “Bonferroni adjusted estimated mean difference ...” as my paper says, an algorithm can be described by precise wording like “we use Knuth’s (1977) generalization of Dijkstra’s (1959) algorithm for directed hypergraphs ...”

Re the comments on Ferguson in the intro to Section 3, this is correct, but it’s not the end of the story. An effort was made to improve the code, which I believe was successful.

- 6 | This effort was (and is) cited and discussed in the paper.

I don’t think this changes the argument of this paper, but the paper should acknowledge that we do know how to fix these things, we just don’t do it until we have to (due to limited resources, no culture that requires it, incentives, etc.)

- 7 | I disagree. The Ferguson case involved fixing a particular case of very bad code after the event of its unravelling and Ferguson’s admissions (as cited); it is not fixing the problem the paper addresses, which is fixing the persistent widespread use of bad code in science.

- 8 | The reviewer says “know how to fix these things, we just don’t do it until we have to” so we agree the issue is to create a route from the problem to a solution; I think my paper provides evidence that the problem is real and widespread, and lays out many ways to start doing this.

I don’t agree that C is not a dependable language, nor that it is not portable. The issues raised here are not unique to C, but are common to almost all languages in usage in computational science.

- 9 | The whole industry-standard MISRA C approach is based on the fact that the community recognizes that C is not dependable, and MISRA and it’s work is already cited. The reason for mentioning C in particular is that it was the language chosen by Ferguson, as explained in the case study; however I agree the unreliability issues are common across many languages, and I now make this point clearer.

These statement are philosophical in this context.

- 10 | Hmm. Only if you believe that C is dependable, a view which I argue against!

The RAP* discussion seems too detailed for this paper, and the supplementary material is too long – this could be published separately (or as a pre-print) and referenced here.

- 11 | The length of the RAP discussion in the paper main body seems right to me, and is supported by Reviewer 2’s comments. (Please note that I changed my term RAP* to RAP+ as I feel this much better indicates that RAP+ is an extension of the original RAP than RAP*.)

- 12 | I agree that the supplemental material is long, but it will be online so I am not sure that this is a problem. The supplemental material is divided into several topics, and includes a table of contents, so readers can navigate it easily.

Re Software Engineering Boards, what about other boards for other types of work:

perhaps we should have data collection boards, statistical review boards, reproducibility boards, etc. But I don't think creating new collections of gatekeepers is going to be either practical or useful.

13 | I hadn't thought of it as gatekeeping, thanks for raising the concern. I've now added a paragraph and a citation to an article in *Nature* to explain that the idea is *not* gatekeeping.

Better standards and better education make sense to me, but more enforcers do not.

14 | There is no sense, and no sense expressed in the paper, in which Software Engineering Boards are gatekeepers or enforcers. The paper makes an explicit analogy with Ethics Boards, which are widely accepted and work well. I don't see any problems, then, nor any "thin end of a wedge" problem the reviewer characterizes here.

I don't think the comment about methodological statements at the start of 5.b is correct. Most venues in which I publish do not require all or even many of these statements from authors.

15 | I changed "most journals" to "Many science journals" (although I know of none that do not require some such statements); in addition, I have cross referenced to the paper's discussion of PRISMA, which itself is a methodology that, if used, authors are expected to declare and comply with. PLOS ONE is an example of a major journal that makes a list of similar methodologies and when the journal expects compliance; I have now made a note of this point in the paper.

Much of this section and 5.c also seem to follow the advertising of RAP* with statements that not only is it good, but it should be required. This is again opinion, and not justified. Would this help? maybe. Is it the best solution? I have no idea.

16 | I disagree that the benefits of RAP and RAP* are opinion; I provide reasoned arguments, and examples of the problems RAP and RAP* avoid.

17 | I note that RAP and RAP+ (its new name) are very good solutions, and I am of course explicitly drawing on the existing very favorable literature on RAP. There may be better solutions available in the future, but until then they are the best known solutions. Furthermore, "the best is the enemy of the good" — it is routine to publish a good idea, as this paper proposes, and the scientific community can then seek ways to refine the idea. The point is, a good idea has to be published first.

The second paragraph in 5.e is not at all correct in my opinion. I think this is an unrealistic view of science.

18 | I briefly considered citing Einstein, who commented on this point, and others in my support, but I've deleted the comments in the paper, as the paper's core message is not in fact affected either way.

The acknowledgements statement isn't grammatically correct.

19 | Thanks for spotting this! Problem fixed.

After reading the paper, the only thing I feel that I have learned is that the author has strong opinions about how bad things are and how the world should work that would

be better. But I'm not convinced that this is the correct analysis and solution, or even if it is, how we would get to this future world from today's world in practice.

20 | The new Background section I have added in my revision makes it clear that my opinions are closely aligned to, and often summarize, substantial work by others.

1.2 Reviewer: 2

This manuscript on the state of computational science today is unusual in several respects. It is mostly an opinion paper, advocating for more rigor in the use of computational methods in scientific research.

21 | As mentioned in my comments to Reviewer 1, my opinions are rigorously justified, and traceable back to peer reviewed literature. It is of course an opinion that scientific research should be more rigorous, but I am unaware of significant dissent to this opinion. (I admire Paul Feyerabend, e.g., his *Against Method*, but the paper does not need to get into the philosophy of science.)

In doing so, it provides a rather exhaustive list of arguments, to the point of almost being a PhD-level course on the subject.

22 | I am aware — and I hope — that scientists, not just computer scientists and software engineers, who can benefit from my analysis will read the paper. I therefore felt it important to write clearly, and to provide as much background and direct information as possible. In particular, the Supplement provides a lot of useful material and pointers to further reading.

Finally, it contributes an analysis of code-and-data sharing in the literature to the arguments, making it in part a research paper. The writing style is much more colloquial than a typical scientific article, making the manuscript lengthy but also very readable. Whether or not these particularities are acceptable in The Computer Journal is a matter of editorial policy, which is not for me to judge.

23 | Thank you for recognizing the readability of my paper. I tried hard to make the paper accessible, as I hope many scientists, not just computer scientists/non-programmers, will read the paper.

My following review mostly treats the manuscript as an opinion paper.

Opinion papers being a matter of opinion, I'll do my best to judge the pertinence of the argument independently of whether I agree with them. But I'll state my prejudices right away: I do agree with most of the observations and conclusions in this paper.

24 | Thanks!

The most important omission in my opinion is the lack of context. Many past and present efforts are aligned with the author's objectives and have made and sometimes implemented similar propositions, but the manuscript does not mention them.

25 | I did, but the mentions were scattered around in the paper. I have now introduced a focused Background section (with additional references to substantiate a more powerful argument) early in the paper collecting everything together to address this helpful point.

The publication of scientific software in source code form has been advocated as early as 1969 by K.V. Roberts ([https://doi.org/10.1016/0010-4655\(69\)90011-3](https://doi.org/10.1016/0010-4655(69)90011-3)), using some of the same arguments that are made in the present manuscript. The next call for publishing software and data I am aware of is by Jon Claerbout in 1992 (<https://doi.org/10.1190/1.1822162>).

26 | The reviewer makes some pertinent points here, and I have therefore added material to the Background section to provide an overview of the relevant literature. In particular, I note that concern for publishing code goes back even further than the reviewer mentions; I cite work as far back as 1958, in fact to *CACM*'s first issue of its very first volume.

Since then, the pace at which similar calls for more rigorous treatment of software and computation have been made has accelerated steadily. More importantly, there are many initiatives that implement parts of the author's recommendations, and it would be more helpful to discuss how their impact and utility can be improved than to pretend they don't exist and propose new initiatives to be started from scratch.

27 | Yes, it was an oversight that there was no stand-alone, clear section in the paper discussing prior work: it was scattered around in various citations. I have therefore now added a new section, Background.

Some examples:

- The Software Sustainability [sic] Institute (briefly mentioned in the supplementary material) and CODECHECK in the UK, and CASCaD in France, resemble the author's SEBs in many ways. More generally, there is an ongoing movement in many countries for establishing the "research software engineer" as a recognized role in scientific research, which shows that the need for better software engineering is increasingly recognized.

28 | Rather than cite such an almost unlimited list of initiatives, I discuss some large studies and the key TOP initiative.

- New on-line journals such as The Journal of Open Source Software or ReScience practice reviewing of scientific code at various levels (and find that this is not as easy to do as it may seem).

29 | I hadn't heard of *ReScience C* (and *ReScience X*) — thanks. I now cite them (in the new Related work section) and what they are achieving.

- Various courses, e.g. by Software Carpentry and Data Carpentry, and also several on-line courses, teach what the author calls RAP (a name I have never seen before) to an audience of mostly PhD students.

30 | The training courses I am aware of (including all those I searched for and found on the internet) are very low level introductions to coding, and really inadequate for coding to support reliable peer reviewed research.

I find rather few completely new arguments and proposals in the present manuscript, but it has the merit of being exhaustive, detailed, and combining advocacy with practical advice.

In the following, I will concentrate on the new aspects.

An important original contribution is the analogy between the introduction of computation and statistics into mainstream research. It deserves publication on its own because of its pertinence. The analogy is very fitting, and helps both in understanding the inadequacy of today's scientific code management and in improving the situation. Moreover, the rigorification of statistics has well advanced to the point that most scientists now realize its benefits. That's a good motivation to launch a similar campaign for more rigor in computation.

28 | Thanks for these encouraging comments. The important point that scientists now recognize the value of rigorous statistics motivates considering rigor in computation is now mentioned in the introduction (section 1).

Another original contribution of the proposal of Software Engineering Boards, but I have to say that I don't find the arguments very convincing. I do agree about the necessity of introducing software engineering practices into computational research, and to check publications for respecting best practices in the field. But the implication of software engineers can take many forms (as part of research teams, as experts in peer review, ...) and I don't find the arguments for the specific institution "SEB" very compelling. The analogy with ethics boards is weak, as the numerous differences listed by the author suggest.

29 | The point of the analogy is that ethics boards *do* work and are recognized as available — if not required — in almost all research institutions. Ethics Boards are routine, and are not onerous as this reviewer suggests. I added a few extra references here, including an analogous proposal for methodology boards — in a recent *Nature* paper, which also explores their pros and cons.

A more general comment about the author's discussions of software engineering in science is the lack of any reference to the particularities of scientific software, compared to software in business or industrial contexts, which is the focus of most software engineers today. And even within the category of scientific software, there are vast differences. An epidemiology simulation (software as a reliable tool) meant to inform public policies must be treated differently from a prototype implementation of a new data analysis technique (software as an executable scientific method). Both need more rigor, but not in the same place.

30 | I do think this is an obvious point. For example, scientific coding is obviously different to game coding, e-commerce, and so on. Discussing the varieties of code seems to me to be beyond the scope of the paper.

As an example, the author heavily insists (in the supplementary material) on the use of formal methods. They are indeed the appropriate tools in situations where the conformance of an efficient implementation with a known-to-be-good specification is the bottleneck in establishing correctness. But often it is the specification itself that is in-

complete, informal, or incorrect. A common situation in computational research is a rapid development cycle in which scientific questions, specifications, and implementations are frequently updated. It is in my opinion an open question if formal methods are helpful in this setting. They add rigor to one step of the cycle, but at such a high cost (with the current state of the art at least) that other steps risk being neglected. Illustration: a piece of code formally proven correct to a rather deep level: <https://doi.org/10.1016/j.camwa.2014.06.004>. Impressive work, but also a Herculean effort, for a single relatively small algorithm.

31 | Indeed! I note that the paper this reviewer cites is 25 tight technical pages, and that its authors admit that their approach is beyond many coders. I have therefore now clarified (throughout) that a formal methods *mindset* is required, not a total formal proof of code as in this paper. For example, as now explained particularly in the Supplement, you could require explicit invariants, assertions and other formal ideas in code: this is a formal mindset in action, but avoids “the best is the enemy of the good” the reviewer seems to fear.

This illustrates a more general problem with the author’s recommendations (in the supplementary material): there is no clear path for getting from today’s research processes to more rigorous processes that include all of the author’s ideas. Implementing all of them is such a huge effort that at best it will suffer the same fate as climate change policies: everybody agrees it should happen, but the amount of work requires is so paralyzing that nothing happens at all. A grand view, such as outlined in this manuscript, needs to be complemented by a list of small initial steps that most scientists recognize as achievable.

32 | I think here the reviewer has fallen for the perfect-is-the-enemy-of-the-good interpretation! I hope my rewording throughout dispels this problem.

Another interesting original contribution is the RAP* method.

33 | I’ve renamed it RAP+, as this is more suggestive of an extension to RAP itself, which the name RAP* wasn’t.

It is not completely original in the sense that many people have practiced variants of it in the past, but I haven’t seen it named before, nor discussed to the extent presented in the manuscript. The author’s analysis of the literature also serves as a case study for RAP*. The discussion of RAP* is a welcome transition from “how to save research from computers” (the main topic of the manuscript) to “how to improve research using computers”. I hope to see more work in this direction in the future.

34 | Thanks for these comments. Of course, RAP* is on the route towards what I envisage, and therefore these welcome comments contradict the reviewer’s worries expressed in their previous paragraph!

Finally, in case the editors of The Computer Journal decide that this lengthy opinion-paper-plus-course does not fit their editorial policies, a few suggestions for shorter articles extracted from this material that I think would be received favorably by the scientific community:

35 | I put all the helpful “course” material as reference in the Supplement,
which will be online rather than part of the paper itself.

36 | I am not really clear why the reviewer focuses apparently negatively
on the paper being opinion. Surely, Tony Hoare’s paper on quicksort
was just opinion, and wasn’t even the best way of sorting under many
assumptions — but that doesn’t stop it being a brilliant and now classic
paper. So far as I can see, all my opinions are argued, justified or
readily traceable back to the literature. I’d be happy to fix any specific
unsupported opinion of concern but the review is not specific.

1. A much shorter and focused opinion paper on the topic: “We made statistics in research more rigorous, now let’s do the same for the use of software”.

37 | This is a good idea, but of course computation can do a lot more than
statistics, so I don’t think such a paper, properly done, would be much
different from my current offering — especially as there seems to be very
little literature making the same points one could thus build on much
more concisely.

2. A review of software engineering techniques, explaining their expected benefits for computational science. The main audience would be research software engineers.

38 | I put all of this material in the Supplement, which I imagine will be
online so its length is not material.

3. A tutorial on the RAP* method, using the data analysis from the present manuscript as a case study.

39 | This is already provided in the supplement.

Konrad Hinsin, CNRS, France (Note to editor: I have decided to sign all my reviews. Please don’t remove my signature.)

37 | I very much appreciate your disclosure. Thanks for all your work. Are
you happy being named in the paper’s Acknowledgements?

1.3 Reviewer: 3

David Spiegelhalter’s name is misspelt “Speigelhalter” throughout. The first time it is misspelt “Speigelhater”.

38 | Oh no! Thanks for this correction. Spelling now fixed throughout.

2 EDITORIAL BOARD RECOMMENDATION:

Associate Editor

Comments to the Author:

Please address the comments raised by the reviewers, and list what changes you have or haven’t made with some justifications.

3 REFERENCES:

General. Please construct references in the following style:

1. The references should appear in the text in numerical order, as should the reference list.

39 | Now done.

2. Where there are more than 10 contributing authors please list only the first author and use et al.

40 | Now done.

3. The following format for references should be followed:

41 | Now done.

Journal article: Galton, A. (1992) Logic as a formal method. *Comp. J.*, 35, 431-440. (author name, year, title of paper, title of journal, volume number, page numbers)

Book: Hogger, C. (1990) *Essentials of Logic Programming*. Clarendon Press, Oxford. (author name, year, title of book, publisher, publisher location)

Chapter: Harel, D. (1984) Dynamic logic. In Gabbay, D. and Guenther, F. (eds), *Handbook of Philosophical Logic*. D. Reidel, Dordrecht. (author name, year, title of chapter, editors names, title of book, publisher, publisher location)

Conference: Crochemore, M. and Verin, R. (1997) Direct construction of compact directed acyclic word graphs. *Proceedings of CPM 97*, Cambridge, MA, 12-14 August, pp. 192-211. Springer-Verlag, Berlin. (author name, year, title of conference, place of conference, date of conference, page numbers in proceedings, publisher, publisher location)

Technical documents: ITU-T. Z.500 (1997) *Framework on formal methods in conformance testing*. International Telecommunications Union, Geneva, Switzerland. (document number, year, document title, publisher, publisher location)

Could you please check your references carefully (in particular, proceedings: location of conference, date of conference, publisher name and location...) and make sure that they comply with the above format?

4 SUMMARY OF CHANGES

In addition to the changes in response to the reviewers' specific comments as listed above, I have uploaded a **changes.pdf** file which is an automatically generated difference listing of the changes I have made. Text in blue is new; text in red is deleted.

The diff process I used (using `latexdiff`) doesn't read included files and has other irritating bugs, so, for instance, changes to data (including changes in tables) are not

shown, and there are various typesetting errors. Regardless of the details, it shows how almost all of the paper's text has been substantially improved, usefully including showing deleted text that I or the referees considered unnecessary.

The added table of contents I hope shows the reviewers the revised paper is now much clearer and more clearly structured; I think the table of contents will be helpful for journal readers too, but it does not need to be included in the final paper if it is not considered to fall within the journal's accepted style.

The supplementary material has minor corrections and simplifications throughout, but no substantial changes comparable to those made in the main paper.