Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2019-329-RC1, 2020

Geoscientific Model Development Discussions
Open Access
EGU

Interactive comment on "CLASSIC v1.0: the open-source community successor to the Canadian Land Surface Scheme (CLASS) and the Canadian Terrestrial Ecosystem Model (CTEM) – Part 1: Model framework and site-level performance" *by* Joe R. Melton et al.

Anonymous Referee #1

Received and published: 9 January 2020

This manuscript describes the CLASSIC v1.0 model, which is a combination of the CLASS and CTEM models. In particular CLASSIC represents a move to open-source development, with additional features to facilitate the use of the model by a diverse community in a range of settings. The paper also gives an overview of the performance of the model at a number of sites.

Overall the paper was well written and clear, and I consider it would be suitable for

Printer-friendly version

Discussion paper

publication after minor corrections.

This will be a relatively short review as I didn't find myself with many questions - essentially the paper appears to be a reasonable summary of essentially 'technical' developments to the modelling system. Many of my comments below are suggestions, which of course the authors might wish to argue against.

The only major thing that I think is missing is some discussion about governance - how will the community be run and governed so that the core model develops in a way that satisfies all or a majority of users? How will the trunk code be managed? How will the potentially conflicting needs of different user groups be managed? For example, I assume there are key stakeholders, such as the climate and earth system modellers who need a reliable, highly-optimised code, but equally there might be people who want to use CLASSIC to study a particular process that is perhaps only important in a small geographical area and/or to a few people but which requires a bigger change to the code base. That might be a rather extreme contrast, but my point is that I would like to know about the proposed process by which the main model trunk is managed - who is in control?

Slightly less important, but recurring and so I will mention it here: On P5 and elsewhere it wasn't clear to me whether CLASSIC supports the use of CLASS without CTEM, i.e. can one prescribe vegetation characteristics (such as height and LAI) rather than having these simulated by CTEM? I suspect this configuration is possible, but it is unclear. Such a configuration arguably makes some evaluations (such as against FLUXNET sites) more meaningful - e.g. looking at the turbulent fluxes is less interesting if CTEM has decided to produce vegetation that is very different from reality at a site (which also relates to the difficulties discussed in spinning up long-timescale processes with only a few years of meteorological data). And if this functionality does exist in CLASSIC, why not employ it in (at least some of) the FLUXNET-based evaluation described in Section 5? (I appreciate that prescribing vegetation characteristics is not a panacea for all of the difficulties outlines in Sec.5.)

Specific comments

P2 L15 Benefits of OSS: I wonder if the creation of a community of collaborating modellers might be considered another benefit of the OSS/community approach (though arguably it might come under your first point 'affordability').

Also on L10 you appear to equate OSS development with being a community model, but these could be considered separate - OSS means the code is accessible to all, but a community model *might* imply more than that in terms of there being a coordinated approach across groups to develop and use the model. I think it is this second sense that you intend, consistent with the aims of the other models you cite (JULES, CLM, etc.). Perhaps you could expand upon some of these aspects.

I'm not convinced that L16 'flexibility of configuration' is a benefit from OSS. Flexibility is a desirable characteristic of the model, but by itself OSS does not give this (but flexibility can become more affordable if OSS provides a bigger community of developers). I think you should remove this from the benefits section - or at least rephrase/recast the idea.

P1 L29 key features: You later describe how CLASSIC provides a means to check that results are reproducable (checksums etc.), which could just about fall under your 8th point here (benchmarking). To me this is an important aspect in its own right and might deserve explicit mention here.

P3 L30 'standard offline model setup' - what is this and where can a user find it (and other configurations)? Later you mention that various runs can be set up and performed using the information provided as part of the package/container, but in general I was unclear as to where (if at all) standard configurations can be obtained. Version-controlled code and the container environment are important, but obviously the model set up is also critical. At some point in the manuscript I would like to see some (or clearer) coverage of where standard configurations can be acquired. On P3 it might suffice to point to this information, e.g. "see Section X".

P9 L7-18: The level of detail and discussion about checksums is unnecessary. The basic principles are well-established and can (most likely) be cited. Most or all of these lines can be removed.

P9 Sec 3.2.3: I had a quick look at some of the material on Zenodo and it is possible that I didn't find the correct material, but I am wondering if there is any collated scientific documentation for CLASSIC. (In fact I'm thinking of science equations rather than the more code-specific documentation that is the focus of this section. I suspect the answer is 'no', as is the case for many models. Section 2 outlines the science and provides various references that can be followed up, but for many models it takes a keen student and days of work to assemble the definitive description of the science in the latest version of the model. As far as I can see the self-documenting code is very useful but does not address the more detailed scientific documentation. Are there any plans for further documentation? As I said, many models face similar challenges, so this is not a particular criticism of CLASSIC alone!)

P10 L25-end: The examples of how to select domains are unnecessary detail in such an overview paper; it is sufficient to know that the capability exists. I suggest removing this.

P11 L7-8: I don't understand this. It reads as if FORTRAN is used to convert namelist files into netCDF. Perhaps the ASCII files listed in the namelists are covered into netCDF? If so, rephrase.

P13 L22: I'm only moderately familiar with ILAMB but I thought it mainly (or solely) used colours to show rankings of results (rather than whether thresholds are passed). Also, even more pedantically, I think ILAMB and other tools can be very useful when assessing and presenting how different versions or configurations of a single model compare - in which case your objection to its use for a single model (L23) would not hold.

P6 L13: Being able to use the same code in standalone and coupled applications is,

of course, highly advantageous. I know other models also struggle to ensure that a single code base can be used for both (with the main difficulties often being at the higher-level interfaces to parent structures). To me this is an important consideration that could (possibly) be brought up earlier in the manuscript (perhaps as an important aspiration for CLASSIC, even if in fact it is still in the future). As it is, some readers might be slightly confused by the earlier claims that the models are used in coupled applications, yet at this late stage they are told the codes differ (even if in fact the differences are largely 'technical' rather than 'scientific').

Minor comments

P3 L10 'second generation' - I never like these terms as they are so vague, but it's just about OK as you quote it! I can't recall who first coined the term (Sellers, Shuttleworth?) but perhaps you could consider a citation (or explanation) to back it up.

P5 L14 "nine PFTs that map directly onto...": Is it that the 9 PFTs of CTEM are collapsed onto the 4 PFTs used by CLASS, so that two or more CTEM PFTs are represented by one CLASS PFT? This could be made clearer. I assume there is some motivating evidence that supports that this is a reasonable approach that saves CPU time.

P5 L28 fire: I realise Fig.1 is just a schematic, but possibly some representation of fire (and land use) could be included.

P8 I found the order of some sections in this area was slightly unexpected. Looking back, perhaps it was mainly that I thought the containerisation section might come later, after code design etc., as it is essentially a way to bundle everything that has been created/described previously. Coding standards and docuemenation I would also push up the list. The move to Git/Gitlab I would describe before other details such as how to add new PFTs. Similarly, P9 L20 mentions the move away from fixed-form FORTRAN...but mentally I have already checked outputs using checksums in the previous section. To me anything about the code repository and coding standards comes

Printer-friendly version

Discussion paper

before checksums. But really this is personal preference, and your order is fine!

P10 L20: As I read this, point runs can produce gridded (2-D) outputs. Is that really the case?

P11 L22 and others: I'd prefer "3 or 6" rather than "3/6" which I keep reading as a fraction. I assume CLASSIC is fairly flexible in terms of timestep lengths and data intervals, but this section (and maybe others) does read a bit as if only certain timesteps are allowed. e.g. L21 says inputs are typically 3 or 6 hourly (which is fine), but later in the paragraph we are talking about "3/6" hourly data and a 30 minute timestep - i.e. it becomes specific. Consider clarifying that the model is indeed more flexible than this.

P12 Sec4.2: Consider clarifying at this stage that AMBER is included in the container (as is later made clear).

Typos etc.

P3 L31: add a comma after 61m.

P10 L20: 'also' seems to be misplaced here. Remove?

P11 L4: remove 'which'

P12 L12: remove 'and'

P13 L21 (and at least one other location): change "don't" to "do not"

P14 L1 'as well': I'd prefer 'additionally'.

P15 L11: change 'Both' to 'both'

P18 L24: Misplaced comma (and I'd remove the first comma on L25 too).

P39 L38: I haven't pored over all the references, but by chance I noticed a slightly dubious entry here: '0, null'.

2019.