Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2019-273-RC2, 2019

Geoscientific Model Development Discussions

GMDD

Interactive comment

Interactive comment on "Description and validation of the ice-sheet model Yelmo (version 1.0)" *by* Alexander Robinson et al.

Fuyuki SAITO (Referee)

saitofuyuki@jamstec.go.jp

Received and published: 29 November 2019

This paper describes the numerical ice-sheet model Yelmo version1.0. The Yelmo model is available on a git repository with sufficient information to run, and this manuscript contains mostly enough description of the model physics and example application. I think this paper is fairly well written with some exception below, and can be accepted with minor revision.

(1) One point is about symmetry of the model (P14, L21 and after). Figure 3 is the simulated basal temperature of the experiments A and F of EISMINT2 configuration, and the paper states 'Yelmo produces symmetrical temperature patterns in both experiments,' First, minor one, I suggest to describe white kind of symmetry is the topic in

Printer-friendly version

Discussion paper

this section. The configuration of EISMINT2 is 'radially' symmetric, but we all failed to simulate true radially symmetric pattern in particular for experiment F.

Second, major one: what is the degree of symmetry in the argument of this section? Actually, taking a closer look, some breaks of symmetries are already visible in the figure 3. The result of experiment A looks symmetric both along X and Y axis, but not along x=y diagonal. The result of F, even worse, shows breaks of symmetry both along X and Y axis (I attached a copy of figure 3 with marks to show the breaks of symmetry). So, even under the figure resolution, Yelmo already failed to produce symmetrical temperature patterns.

In my opinion, preservation of model symmetries requires full control on the source and compilation, because even single change of arithmetic orders (e.g., (A+B)+C vs A+(B+C)) in a model can trigger and amplify breaks of symmetries. Yelmo depends on an external library in order to solve SSA equations, which is hardly controlled from outside, therefore Yelmo may find such symmetry breaking under an idealized configuration with ice shelves, even the SIA part is perfect.

On the other hand, although preservation of the symmetries in the model is desired, it is not a top priority of a model, especially for one to simulate realistic worlds. We believe that such minor points have little influence on the simulation under realistic, highly asymmetric configuration for most of the application.

So, I suggest the authors to keep the argument of Yelmo symmetries, and also state clearly the standpoint and/or main targets of Yelmo.

It may not be a reviewer's work, I check the Yelmo source code to find the source of symmetry breaking. I attached some suggestion for Yelmo to preserve the numerical symmetries as a series of patch files since revision ed94c608516e2c46c7985ea98eea94fce47b37d8 (you can run git-am to apply them). It may be not complete and, honestly speaking, it may have bugs because I did not check in detail. The author can import if they like them, but hope them to check the revision and results in detail before inclusion. If fortunate, SIA results will be more symmetric than before.

(2) Another point is about precision (P4, L2). (See also minor points below for terminology of floating-point types).

The paper states that single and double precision give equivalent results, because the units of all time variables in Yelmo are cast in years instead of seconds and thus very small numbers are avoided. I do not understand this statement. I do not claim for the result but for the reasoning.

As far as all the quantities are larger than the smallest limit of floating-point number representation, same precision (significant digits) is kept either for the case with unit seconds and with unit years, because it differs not in precision but in the order of magnitude. The smallest number of a typical 'single' precision is around 1.18e−38. What variables do have possibility to show smaller value than this? The rate factor can be small, but even ice temperature is -100 degree Celsius, its magnitude is 1e-31 /s/Paˆ3, which is large enough to be represented by single precision.

I agree that, a typical number of significant decimal digits of single precision is 7, which is actually smaller than the digits of the factor from year to second (31556926, 8 digits). If all the quantities in the model is originally defined with unit year, then it is possible to meet such situation, where unit-second version shows different results in the final digit. However, many parameters are originally defined in unit second and converted into unit year in the model, thus round-off happens in some parameters themselves instead, which are almost the same situation as the unit-second case. (By the way, fortunately 31556926 can be fully represented by single floating-point number while 3155692[57] are not).

If there is a variable to be smaller than the limit in the case of unit-second, my question is solved. So, please give me an example.

Again but from a different point of view: this argument should depend on the model spatial resolution. As I mentioned, difference between single and double precision is merely the number of significant digits, if order of magnitude of all the quantities can be represented enough by the single. For a coarse resolution, difference in values at two adjacent grids of a field (e.g., surface elevation) is large enough to keep precision in their differences, However, for a higher resolution where the values at two become closer, so-called cancellation effects become large enough to reduce the precision of their difference. Relative error of the difference can be large enough for single precision to deviate from that by double precision. Generally speaking, a higher resolution experiment require high-precision computation to avoid such cancellation effects.

Minor points:

About precision (P4, L2) 'Single' and 'double' precision are in principle machine dependent characteristics although there are few exceptions. There is some definition of typical floating-point representation in IEEE754.

2.3, around Eq.(14). ub is defined as basal sliding above Eq.(14) while a depth-averaged velocity below (14). I am confused. Possibly typo?

2.4, below Eq. (27). Better to write as 'Horizontal diffusion is assumed negligible.'

Table 1 last column. No degree mark.

SAITO Fuyuki.

Please also note the supplement to this comment:
https://www.geosci-model-dev-discuss.net/gmd-2019-273/gmd-2019-273-RC2-supplement.zip

---

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2019-273, 2019.

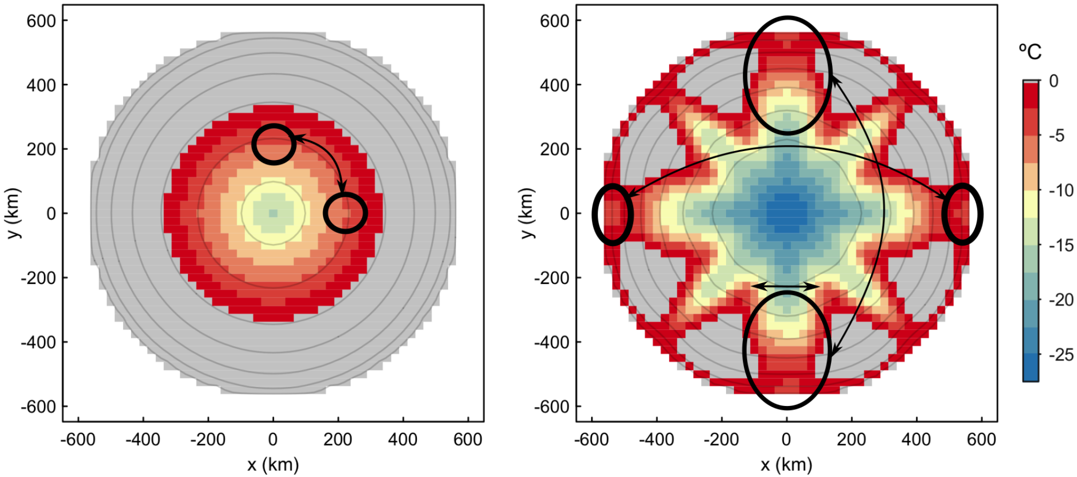

Fig. 1. Annotation on Figure 3 in the paper.