Geosci. Model Dev. Discuss., 7, C2725–C2730, 2014
www.geosci-model-dev-discuss.net/7/C2725/2014/

Geoscientific
Model Development
Discussions
Open Access

Interactive comment on "MetUM-GOML: a near-globally coupled atmosphere–ocean-mixed-layer model" *by* L. C. Hirons et al.

L. C. Hirons et al.

l.c.hirons@reading.ac.uk

Received and published: 22 December 2014

My main question concerns computational cost of the system, "comparable [...] to running in atmosphere-only mode". It is written that MC-KPP is "computationally inexpensive (<5% of the cost of the atmosphere ...)", and it is clear that this cost is smaller than for a full ocean model, but I would appreciate to know if the MPI communication library has been used to parallelize this mixed-layer model, to be able to evaluate its scalability at higher resolution. I also would like to know how much additional computing cores are necessary to run MC-KPP. To summarize, and have a clear idea of this computational cost (including coupler and coupling synchronization), may I suggest to the authors to provide two simple numbers for both atmosphere-only and MC-KPP coupled runs ? The number of simulated year per day and the number of core.hours necessary to simulate a year.

We do not use MPI communication within MC-KPP, but we have incorporated OpenMP threading within the model's one-dimensional physics routines to improve performance for large coupling domains or fine horizontal resolution.

The additional computational cost from coupling depends on the horizontal resolution and the construction of the supercomputing system. We used the ARCHER (Cray XC30) supercomputer to perform these simulations. On ARCHER, it is not possible to share a computing node between executables. There are 24 cores per node. For the MetUM-GOML simulations, we ran the MetUM atmosphere on four nodes, the OASIS coupler on one node (as a serial process) and the MC-KPP ocean model on one node. Timing diagnostics from our simulation indicate that MetUM CPU time was 98% of wallclock time, OASIS CPU time was 3% of wallclock time and MC-KPP CPU time was 7.5% of wallclock time. The difference between each of these numbers and 100% represents the amount of time that each component of the coupled model is waiting for data from the other components. We simulated 1.67 years per wallclock day and required 2073.6 core hours per model year.

Based on these figures, the MetUM atmosphere requires: 4 nodes x 24 cores x 0.98 (CPU time/wallclock time) / 1.67 (years/wallclock day) x 24 (hours/day) = 1352 core hours per model year (of actual CPU time)

The MC-KPP ocean requires 1 node x 24 cores x 0.075 (CPU time/wallclock time) x 1.67 (years/wallclock day) x 24 (hours/day) = 72 core hours per model year (of actual CPU time)

From these two numbers (72 divided by 1352), we conclude that MC-KPP requires approximately 5% of the computational cost of the atmosphere, measured by CPU time. We acknowledge that the design of the supercomputer does not allow this efficiency in *practice, however.*

The MetUM atmosphere-only simulations in this study used the same number of cores (144), but obviously all cores were devoted to the atmospheric model. Those simulations achieved 2.08 years per wallclock day and required 1661.5 core hours per model year. Based on these numbers, MetUM-GOML requires approximately 25% more wallclock time per model year.

We emphasize that these numbers depend strongly on the number of cores that the user decides to devote to the atmospheric model. The MetUM does not scale linearly, so varying the number of cores assigned to the atmosphere will alter the how "cheap" MC-KPP appears relative to the cost of the atmosphere. Even at a horizontal resolution 25 times finer (i.e., 25 times more gridpoints) that the one used here, it is possible to run MC-KPP on only one node without slowing the coupled model simulation. In such a simulation, the atmosphere is typically configured with at least 32 nodes (768 cores), but MC-KPP still requires only one 24-core node.

We have clarified the costs in the revised text, noting the difference between the cost in CPU time and the cost in wallclock time. We hope these details are useful to you. Please contact us if you would like further information.

My second question is related to flux correction technique (2.1.1 §). I wouldn't say that "constraining ocean temperature and salinity profiles in the coupled model produces small SST biases" "by construction" "in the resulting free-running simulation". In this free-running simulation, a correction is imposed "with no interactive relaxation": it means that the coupling could freely lead to substantially bigger anomalies than the imposed correction. It is not the case in your simulation, but it is not as obvious as "by construction" can suggest it. Could you describe with more details how this correction is calculated ? And could you precise if Figure 1b shows results of a free-running coupled simulation with correction made with a15-day relaxation run ?

The aim of section 2.1.1 is to describe the flux correction technique in general terms.

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

Discussion Paper

The specific details of the setup used in the experiments which are the focus of the remainder of the paper are later detailed in section 2.2. This is purposefully done separately; the modelling framework is presented first in general terms then one particular application of that framework is described and analysed. Therefore, the comment is correct in saying that constraining ocean temperature and salinity profiles does not (always) 'by construction' lead to small SST biases. The text about small SST biases at the end of section 2.1.1 was linked to a particular example of this modelling setup in which observations were used as the reference climatology in the relaxation run to calculate the temperature and salinity corrections.

It has been made more obvious in the text that the comment about small SST biases and Figure 1 relates to an example (with observations as the reference climatology and a 15-day relaxation timescale). The subtitles in Figure 1 have been updated to show exactly which observational dataset the biases are calculated against and in the caption it has been made clear that this is from a free-running MetUM-GOML simulation in which the flux corrections have been calculated as described in section 2.2 (observations as a reference climatology and a 15-day relaxation timescale).

Another correlative question that seems not addressed in this document is the possible drift in the free-running simulations ("K"). Mixed-layer models are particularly sensible to flux correction calculation: small differences in imposed correction values lead, sooner or later, to temperature and salinity drifts in free-running simulations. I didn't find any information about this drift, since Figure 1b only shows an average value (of 10 years ?). This value is important to determine the maximum length of the free-running simulations (before reaching biases values comparable to that of full ocean coupled simulations).

To clarify, Figure 1b shows an average of 60 years of free-running simulation (calculated from 3x20 years of MetUM-GOML simulation). The global RMSE of SST over those 3x 20 years of simulation for the three K-O ensemble members is shown in the graph below. All three members remain below a RMSE of 0.38 for the entire 20 years, *compared with an RMSE value of 0.82 for MetUM-NEMO (dark red dot). There is no clear drift over the first 20 years. Coupled MetUM-GOML simulations have been continued to 60 years and do not indicate signs of drift.*

The subtitles and captions of Figure 1 have been updated to clarify what the Figure shows. A comment about there being no signs of drift in these simulations has been added to the end of section 2.1.1.

I am interested by this powerful and necessary framework and would like to know how easy it is to adapt MC-KPP to another atmosphere grid. Maybe a few sentences about how this grid is defined and how parallelization (if any) is organized would help me to evaluate the amount of work necessary for such operation.

MC-KPP defines its grid by reading the land/sea mask of the atmospheric GCM to which it is coupled. Since the columns are independent, their spacing and organization are arbitrary. As stated above, there is no parallelization in MC-KPP, only OpenMP threading.

One of the authors, Nicholas Klingaman, has successfully coupled MC-KPP to two other GCMs, the NCAR CAM and the NCEP GFS. Both of these are spectral models, which demonstrates that MC-KPP is easily adaptable to irregular grids.

Interactive comment on Geosci. Model Dev. Discuss., 7, 6173, 2014.

Full Screen / Esc

Printer-friendly Version

Interactive Discussion

Discussion Paper

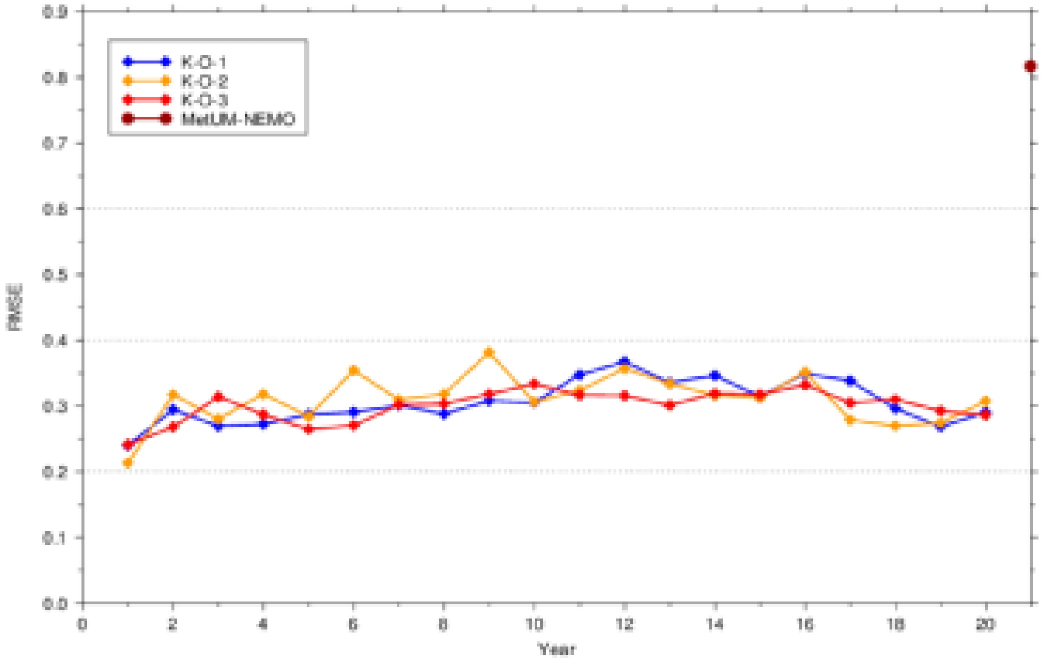

Fig. 1. Root Mean Square Error (RMSE) of global SST for 3x 20 years of free-running coupled MetUM-GOML simulation (blue,red and yellow lines) compared with the MetUM coupled to NEMO (dark red dot).