Testing robustness in a global ecosystem model: the effects of language translation

M.Bithell*[1], P. J. Underwood[2], M.B. Harfoot[2] and D. Tittensor[2]

[1]Department of Geography, University of Cambridge, Downing Place, Cambridge, CB2 3EN, United Kingdom
[2]UNEP World Conservation Monitoring Centre,219 Huntingdon Road, Cambridge, CB3 0DLUnited Kingdom
*Email: mike.bithell@geog.cam.ac.uk

Abstract

The Madingley model is a generalised global ecosystem model that represents plants and animals using cohorts to represent collections of individuals across sets of functional groups. We tested how robust this model is to the implementation platform by translating the existing C# code, running on Windows, into C++ running on Linux. Results suggest that the quantitative values returned from a single model run are not all robust to this change of platform. Whilst cell-based plant modelling and dispersal algorithms reproduce well, ecological processes that affect animal abundance and biomass show changes between the two models that differ by functional group, despite the algorithms in the two models being nominally identical. Analysis suggests that this results from rounding errors being treated differently by the two implementations where functional group cohort sizes become very large. However, the qualitative and statistical properties of the outputs seem to be better preserved.

Keywords: Ecosystems, Agent-based Models, Global Modelling, Computational representations

1. Introduction

The Madingley model is a global general ecosystem model that is able to represent not just vegetation but also animal components (Purves et al. 2013, Harfoot et al. 2014). The model is agent-based, but because of the vast numbers of animals that exist in certain categories (zooplankton for example) it uses a cohort representation in which a single agent may represent just one individual or many millions. In order to keep the model as general as possible, the cohorts may also represent multiple species that occupy the same functional group, so that our limited knowledge of the specifics does not prevent a meaningful representation of the dynamics. Grid-based climate driving fields and land-surface representations are used to link the individuals to environmental factors, with a variety of configurations that allow for the exploration of different scales. The original version of the model was written in C#, but as part of an exercise to port the model to a Linux-based high performance computing environment, a translation exercise was undertaken to create a C++ version. The ideas behind this were

a) Not having to use an extra library such as mono in order to run the model in Linux should lead to improved performance
b) Moving to explicit memory management rather than using C# garbage collection may help performance
c) Using C++ would lead to a track that would allow adoption of MPI-based frameworks such as RePastHPC in order to scale the model up.
d) Coupling to models of climate or vegetation written in FORTRAN should be possible through existing FORTRAN-C++ methodologies (e.g. Collins et al. 2005, Valcke et al. 2012, but see also caveats in Bruggeman, and Bolding, 2014.)
e) C# and C++ syntax and structure are sufficiently similar that the re-coding would require significantly lower effort than switching to a completely unrelated language such as FORTRAN.
f) An opportunity would arise to re-factor the code to improve maintainability without needing to make a complete re-write. In practice this allowed nearly half of the existing 20000 lines of code to be removed and to a very considerable reduction in the complexity of function calls.

However, in making such a translation careful testing is needed in order to check whether model algorithms are still functioning as expected. It is frequently the case that switching a complex model to a different platform or even just to a different compiler may lead to changes in output. The same can be true of random number generation if one relies on built-in libraries that may have unknown back-end algorithms. To mitigate this last problem the random number generators in the current case were explicitly coded and tested to be sure they were in agreement. In the following we look at the way that the output differs between the two versions, despite their being nominally algorithmically identical.

2. Testing the outputs

Both models were run with the same local set of input data files, single core and over the same latitude/longitude ranges, with the same initial conditions and same time and space resolution (monthly time steps, 10-degree spatial discretisation). Testing was broken down into subsections corresponding to terrestrial vegetation (a grid-based representation), dispersal modelling for moving animals and then dynamical modelling of other ecological processes. Both the terrestrial vegetation model and the dispersal model were found to agree exactly (to double precision floating point) for an indefinite length of model run. However, differences began to appear once other model components were added. Whilst the first few timesteps would agree, within a small number of steps the results from the two models diverged. Figure 1 shows an example for the global biomass and abundance of zooplankton, feeding on phytoplankton forced from a global time-series (i.e. the phytoplankton field itself was not dynamic). Figure 2 shows a snapshot of the spatial pattern of abundances that lead to Figure 1 and Figure3 the timeseries of abundance in a single cell. Close examination of the code shows that the single-cell differences arise as a result of rounding errors accumulating differently in the two versions as a result of highly non-linear processes (such as death by starvation) acting on cohorts with abundances of order 10^{18} . Since these numbers this far exceed the 15 or so digits that can be stored by double precision floating point, and since the abundance can fluctuate by this order of magnitude in a single timestep, the effects of rounding are strongly pronounced (note that both implementations use IEEE754 64-bit doubles, so at least nominally the representable numbers should be identical) . However, the general qualitative features of the two time series in figure three, in terms of the size, duration and overall frequency of fluctuations are in

general agreement: what seems to differ is the timing of plankton bloom events. While these time series are clearly chaotic, they also fall back into register occasionally (e.g. at month 174) presumably as a result of the regular forcing coming from the phytoplankton input file. One might expect that these differences would decrease with a larger number of cohorts per cell, shorter timestep and higher resolution, as all of these should reduce the rates of change in cohort abundances. Re-running the model with the same number of cohorts per cell but at 5 degree and 2 degree resolution does indeed reduce the difference in total biomass and total abundance, but the root-mean-square cell-by-cell differences remain almost unchanged (figure 4) implying that rounding-error induced noise levels are still high. However, averaging over multiple cells reduces this disagreement, suggesting that the overall statistical properties are preserved between versions.

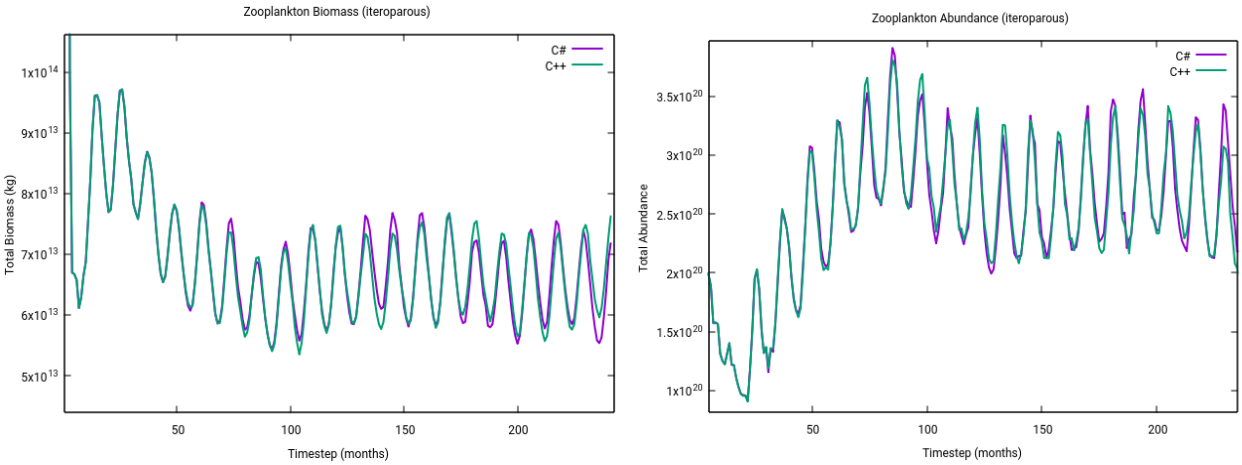

Figure 1: Comparison of zooplankton biomass and abundance. Although the values agree very closely for the first few timesteps, differences become evident near seasonal peaks and troughs as a result of rounding errors. 10degree resolution run with number of cohorts per cell capped at 100.

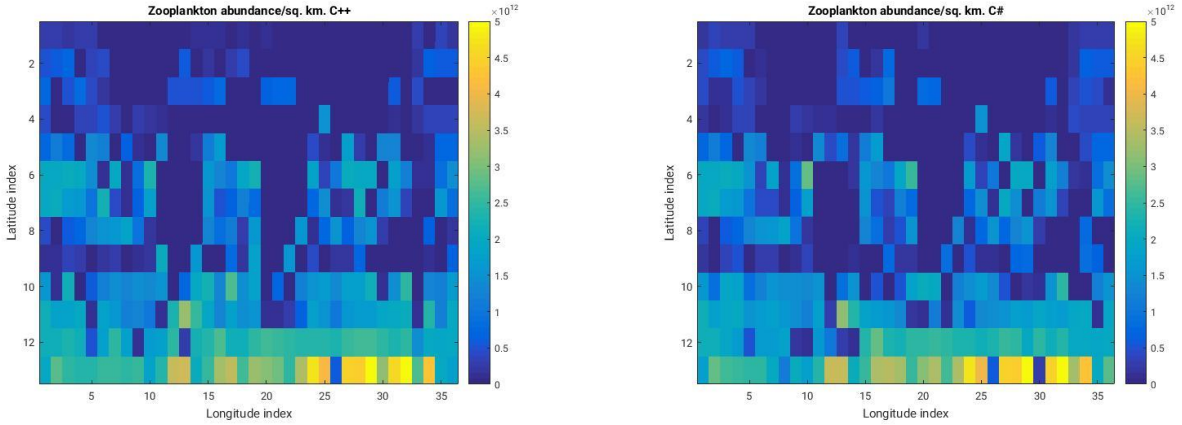

Figure 2: Spatial distribution of zooplankton abundance per sq. km. at 10 degree resolution, month 98. Left C++, right C#. Although the overall pattern is similar note the large cell-by-cell differences e.g. along the lowest line of latitude

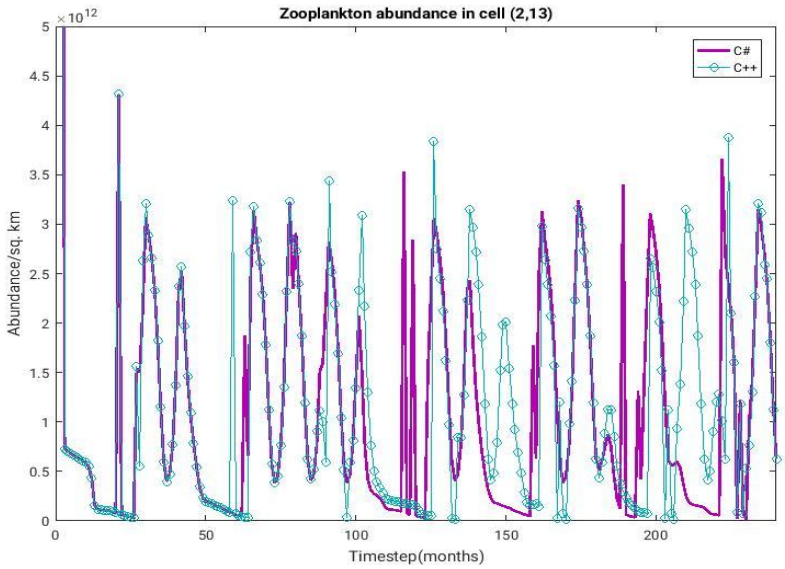

Figure 3: Time series of zooplankton abundance in a single cell, 10 degree resolution. Purple line C#, blue line with markers C++

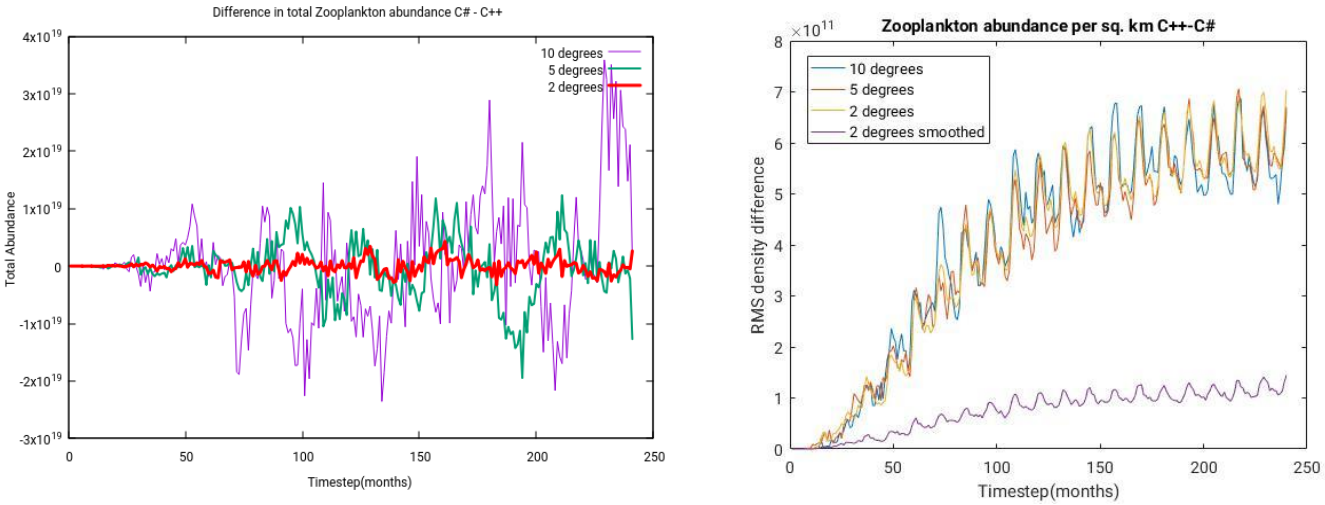

Figure 4: Time series of differences in total abundance and RMS cell-cell density differences for three model resolutions. While the total converges, local cell to cell noise levels remain unchanged. However, smoothing the 2 degree fields with a 5*5 top hat and then taking the RMS difference leads to a much reduced disagreement relative to the 10 degree runs

When broken down by other functional groups across the model, we find that the quantitative agreement between the two version varies from group to group , but that again qualitative features seem to be persistent. Figure 5 shows this for two land-based examples, this time showing total global biomass. On the left we can see considerable differences in overall totals, whereas the

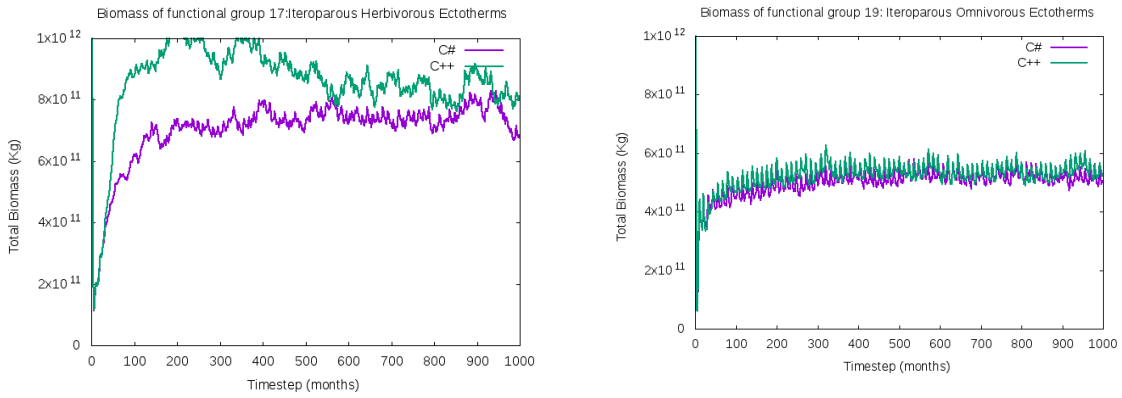

Figure 5: Comparison between C# and C++ for two land-based functional groups.

right hand plot show much closer agreement. However, the variability in both cases again seems to have the same character in terms of size, frequency and duration of fluctuations.

3. Conclusion

While we are confident that the implementation of algorithms in the two model versions is as close as possible, given the differences between the available data structures in the two languages, and that the models agree well for the first 50 or so timesteps, the differences that arise later indicate that the individual quantitative values in a ***single*** model are not robust to implementation platform. Averages over multiple cells are able to mitigate this problem to some extent, and this suggests that averages over multiple time series in runs with different random seeds (not yet tested) might similarly prove helpful. On the other hand, the qualitative features of time series do appear to have some degree of reliability. This suggests that for this kind of highly non-linear stochastic model, the representation of the kinds of things that can happen may prove to be more reliable than the size or timings of individual events.

4. References

Bruggeman, J. and Bolding, K., 2014. *A general framework for aquatic biogeochemical models.* Environmental Modelling & Software, 61, pp.249-265

Collins, N., Theurich, G., Deluca, C., Suarez, M., Trayanov, A., Balaji, V., Li, P., Yang, W., Hill, C. and Da Silva, A., 2005. *Design and implementation of components in the earth system modeling framework.* The International Journal of High Performance Computing Applications, 19(3), pp.341-350.

Harfoot, M.B., Newbold, T., Tittensor, D.P., Emmott, S., Hutton, J., Lyutsarev, V., Smith, M.J., Scharlemann, J.P. and Purves, D.W., 2014. *Emergent global patterns of ecosystem structure and function from a mechanistic general ecosystem model*. PLoS Biol, 12(4), p.e1001841.

Purves, D., Scharlemann, J.P., Harfoot, M., Newbold, T., Tittensor, D.P., Hutton, J. and Emmott, S., 2013. *Ecosystems: time to model all life on Earth*. Nature, 493(7432), pp.295-297.

Valcke, S., Balaji, V., Craig, A., DeLuca, C., Dunlap, R., Ford, R.W., Jacob, R., Larson, J., O'Kuinghttons, R., Riley, G.D. and Vertenstein, M., 2012. *Coupling technologies for earth system modelling*. Geoscientific Model Development, 5(6), p.1589.