Comments on "**The generic simulation cell method for developing extensible, efficient and readable parallel computational models**"

by *I. Honkonen*

Overall impression:

This paper presents a method to improve and facilitate readability, validation and verification of existing and prospective simulation codes. This can be achieved by making programs reusable and extendable through coupling, without modifying existing code. The reuse of already verified and validated programs is thereby evidently less error-prone than programs written from scratch or after extensive modifications.

The introduced "generic simulation cell method" library enables this coupling technically by providing a flexible data structure that abstracts cells of a computational grid. These cells store and encapsulate all variables that are relevant for a certain cell and also provide an API for message passing control. With this API, variables can be activated and deactivated for the coupling in a shared memory environment on a cell-by-cell basis. An MPI_Datatype based on that scheme can be queried afterwards. The "gensimcell" library is realized by variadic templates in C++, which are part of the C++11 standard. It also depends on the Boost C++ library and MPI.

The usage of "gensimcell" is explained in this paper with the help of geoscientifically relevant examples (Conway's Game of Life, scalar advection and Lagrangian particle transport). The examples also depend on a library from an earlier publication of the author (and colleagues) "dccrg" ("Honkonen et al. 2013, Comp. Phys. Comm. 184"). A short benchmark shows that the serial as well as the parallel performance is barely influenced by the variadic template approach. It is also notable that the source code and the examples are provided at public access.

The paper is well written. The "gensimcell" is a remarkably clean, structured and well documented library. In my opinion the paper is ready for publication after considering the two following points of criticism and some technical corrections.

Points of criticism:

1) In my opinion, it would be appropriate to give a few more details on the presented approach; this is, to show some implications of this method. Most obvious (at least for me) is the memory-layout (each variable value is interleaved in the cache/RAM by the total number of variables for this cell). Even though the text points out that readability and maintainability of the code is preferable to performance, I think it is desirable to

discuss the effects that come along with properties stored per cell vs. properties stored as vector or array. For example: While the cell approach might be beneficial for functions of the kind:

```
grid[x][f()]=grid[x][a()]+grid[x][b()]+grid[x][c()]
```

it is probably not for:

```
grid[x][a()]=grid[x-1][a()]+grid[x][a()]+grid[x+1][a()]
```

A performance comparison would be interesting, but is certainly not in the scope of this paper.

2) In my opinion the examples provided are not fully comprehensible solely with the listings. Even though Fig. 1-2 are sufficient to get a good impression of the serial usage, Fig. 3-4 provide no additional information. Fig. 3 has not enough context and shows the same calls as in Fig. 2. In relation to this, Fig 4 is partially restructured, which is confusing and it only shows the flow, while the interesting parts are outsourced to other files and the dccrg library. Fig. 5-7 are OK even with little context (Fig. 6 could have a little more context, for example, variable declaration), since they only illustrate a special problem.
   I would suggest replacing Fig.2-4 by one simple (e.g. GoL) complete, parallel example, which illustrates all API/library calls. This would be preferably without gccrg. If this is not possible, relevant snippets from gccrg would be helpful (e.g., message passing).
   If this suggestion is not feasible, it might also be possible to show parallel examples right from the beginning, hence having consistent transitions between the code snippets.

Technical comments:

P 4581, l 26: missing reference to “Conway’s Game of Life”.

P 4582, ll 3-5: This is very specific. Instead of suspending “copy+paste” it might be better to loosen the term: “without any modification”. It is misleading on several locations.

P 4583, l 1: “from from” typo?

P 4585, l 1: “boost::indetermined” would be clearer.

P 4591, l 1: “requerd” typo