Geosci. Model Dev. Discuss.,
doi:10.5194/gmd-2016-140-AC1, 2016
© Author(s) 2016. CC-BY 3.0 License.

Geoscientific Model Development Discussions
Open Access
EGU

Interactive comment on "A numbering algorithm for finite elements on extruded meshes which avoids the unstructured mesh penalty" *by* Gheorghe-Teodor Bercea et al.

Gheorghe-Teodor Bercea et al.

gb308@doc.ic.ac.uk

Received and published: 15 September 2016

We would like to thank the first reviewer for the suggestions to improving the manuscript. We have addressed all the questions and comments individually.

Comment 1: *Data structures for layered meshes have been considered and implemented before - certainly in single-purpose codes, but also in frameworks (DUNE's prism- grid module, e.g.); I am aware that providing a survey of respective approaches to grid numbering in such packages might be impossible to do, but I think a general discussion on what options actually exist (and what implications resp. choices might have) when designing the numbering scheme could make the paper stronger.*

Printer-friendly version

Discussion paper

CC BY

Answer: We have added some additional discussion in the introduction around alternative approaches to this same problem.

Comment 2: *This is a bit related to the choice of title: at first reading I found myself expecting such a discussion; however, the paper clearly focuses on the approach followed in Firedrake (which is fine in itself, but a bit in contrast to the generic title and the abstract).*

Answer: The paper describes a numbering algorithm which is completely generic to finite element approaches. We have flagged up in the abstract that the evaluation is in Firedrake. The title has also been changed to explicitly reflect that the performance evaluation of the algorithm is done in Firedrake.

Comment 3: *You chose to number the DOFs of an entity contiguously, such that all DOFs of an edge (or cell) would be contiguous in memory. However, for low order methods and when the key design goal is to allow vectorization, you might want to strictly keep a stride-1 access on corresponding DOFs in layers - effectively this would mean exchanging the l and d2 loops in Alg. 1. In any case, this choice depends on the type of operations we expect in simulations (whether we are strongly memory or compute bound, what the memory access patterns are, etc.), so a discussion on this would be interesting.*

Answer: We acknowledge that the ordering within each entity column is not unique. However, we do not see an obvious advantage to interchanging as suggested here. We have commented on this in section 3.2.

Comment 4: *As far as I got it, your concept of a stencil goes beyond the strict notion typically used for finite difference methods on structured grids: your stencils may also include element-local operations in finite-element- type methods (requiring a cell and its faces, edges, vertices) or also a face-based flux operation as in finite volume methods (which might require a face and its two adjacent cells). In any case, you might explain this in a bit more detail, and maybe state one or two examples.*

CC BY

Answer: The reviewer is entirely correct. We have made our definition of a stencil explicit in a new section 2.4.

Comment 5: *What kind of unstructured mesh did you actually use for your results? You discuss in the paper that having a structured mesh as base mesh is advantageous for performance. Hence you might even explicitly address this issue by comparing results for a structured mesh (stored in an unstructured way) and one (or more?) typical unstructured meshes from applications.*

Answer: In the problem setup, we have described how we generate the base mesh, using Gmsh. Although the domain of computation is regular, the mesh itself is unstructured.

We believe the comparison to a topologically structured mesh is outwith the scope of the paper. In particular, the comments relative to structured base meshes are in place to indicate that we are explicitly depriving ourselves of these advantages, since we are aiming for an iteration algorithm that gives good performance irrespective of the base domain. Our results demonstrate that, for layered meshes, a reasonable base numbering (obtained in our case via RCM) is sufficient to obtain performance close to hardware bounds at significantly fewer layers than are scientifically interesting.

Comment 6: *As my only major suggestion, I would like to encourage you to switch from GFlop/s to GB/s in all performance plots: as you are in a memory-bound regime and the numbering scheme primarily addresses achievable "valuable bandwidth", "GB/s" would be the natural metric.*

Answer: We have clarified in section 4.3 that the relevant bound is, in fact, operation count, and not bandwidth. We were ourselves surprised by this conclusion, however the performance results support this hypothesis. As such, GFlop/s is the correct metric.

Comment 7: *You might check whether having a log-scale for x-axes makes the results for few layers better visible*

CC BY

Answer: We tried this, but it did not create a more useful figure.

Comment 8: *It would be helpful to a add a sketch for illustration of the indexing scheme defined in Eq. (5)*

Answer: We agree and have taken the opportunity to do so as Figure 3.

Comment 9: *I was wondering what kind of stencil a DG0xDG0 discretization would produce for the residual; aren't all accesses element-local then? In general, would it make sense to add a table (or similar) that describes which entities are accessed for the various discretisations?*

Answer: This is correct, and Figure 5 shows the degrees of freedom and therefore entities which are accessed in each case.

Additionally, we have fixed the suggested typos.

Interactive comment on Geosci. Model Dev. Discuss., doi:10.5194/gmd-2016-140, 2016.

CC BY