Geosci. Model Dev. Discuss., 6, C1612–C1618, 2013
www.geosci-model-dev-discuss.net/6/C1612/2013/

Geoscientific
Model Development
Discussions
Open Access

Interactive comment on "CUDA-C implementation of the ADER-DG method for linear hyperbolic PDEs" *by* C. E. Castro et al.

C. E. Castro et al.

ccastro@uta.cl

Received and published: 6 October 2013

The authors are grateful for the critique of the reviewer, asking for more details on the performance and details of the GPU optimized code. Indeed, we did not place the speed up factor upfront in the abstract as it is not the central point of our manuscript. We acknowledge that we did not describe with enough rigor, what the main point of our work was; namely to investigate the potential speedup of a GPU enabled code under the constraints of practical applications, where the same accuracy is necessary and the effort to change the code is to be minimized (see figure 1). We think, it is extremely dangerous to reduce all information into a speed up factor as it will depend on the specific CPU/GPU model, architecture, the implementation details, compiler version and the optimization. We will make this point clearer in the revised manuscript.

At the same time, we will also improve the description of what the reviewer asks for. So, we will add speed-up information more prominently in the revised version. The comment in the abstract regarding the particular benchmark test is true in the sense that we use numerical test problems not specifically designed to demonstrate the maximum number of floating point operations per second, but benchmarks that allow us to check implementation issues and convergence. We consider these kinds of test problems more relevant for building a numerical tool useful for computer supported simulations in geoscience as these metrics are fundamental to check correctnesses of our numerical implementation. We would suggest to present our work in a more qualitative perspective, that is:

- It is possible to implement a GPU version of the ADER-DG numerical method with the expected accuracy.
- The GPU version of the algorithm presented in this manuscript is faster for the test problems presented.

We provide two speed up factor figures. Figure 2 for test problems 1 and 2 and a Fig. 3 for test problem 4 (the elastic wave equation).

In the introduction we mention two important geoscience topics like tracer advection and linear elasticity for seismic wave. To our knowledge these are two extremely important and relevant topics for geoscientific research. We are more than open to explain their relevance for oil spill modeling, atmospheric tracer dispersion modeling, pollutant transport, seismic survey support for gas and oil exploration, earthquake research and hazard assessment, etc. However, we were confident that the readership of GMD is familiar with these topics and the relationship of the equation sets in both examples to the mentioned application fields. Some guidance from the editor would be helpful for this. We are grateful to the reviewer in pointing us to the article of Klöckner et al. and will include it in the references.

In the description of our actual implementation we focus on the high-order properties of this numerical method and state the research question of whether it can actually be reproduced in the GPU implementation. We agree on that we did not emphasize the relevance of such high-order numerical method and the benefits gained by providing a numerical implementation which can generate accurate numerical solution in lower computational times therefore we will add this information to the introduction.

We acknowledge that the description of our GPU optimization strategy has not been highlighted in enough detail. This is partly due to the fact that our code was not included in the submission, as promised, and we apologize for that. Of course we will provide this in a revised version. To a greater part it is due to our decision to not give the technical details in the text in order to keep it readable. We propose therefore, to include pseudo-code examples (see Author Comment) for the important parts of our CUDA implementation. Moreover our approach to porting a successful numerical method in geoscience (ADER-DG) to GPU architecture is from a practitioner point of view and not as computer scientist. In our work we show that the mentioned numerical method is portable to GPU, we propose one successful strategy that will help other scientist to reproduce our work and to port their own code and, as observed in this study, has potential to produce even better speed up factor. Moreover, in this work we empirically prove the expected order of convergence for the GPU implementation and show evidence that in the single precision mode this convergence could be very limited up to a point where results are unacceptable. This is relevant when we see recent publications like Mu et al. "Accelerating the discontinuous Galerkin method for seismic wave propagation simulations using the graphic processing unit (GPU)âĂŤsingle-GPU implementation" Computers & Geosciences 51 (2013) 282–292, where the only metric is speed up factor and the accuracy is just commented by the solutions between new GPU and classical implementation "have no distinguishable differences".

The speed up factor is visible in Fig. 9 where we observe 10x for the third order method presented in light blue (continuous v/s dashed line). Also in Fig. 11 we observe 10x

speed up factor for the third order method. In Fig. 14 we see 10x speed up factor but it decreases to 2x for the higher order runs. The visible maximal factor on certain order is in agreement with the observation of Klöckner et al. in Fig. 8a. We will improve the discussion on each test and conclusions providing further information regarding the output observed and comparing this to the information provided by the nvcc compiler and occupancy calculator, in particular the limiting resources as shared memory and register and the number of thread per block used

Interactive comment on Geosci. Model Dev. Discuss., 6, 3743, 2013.

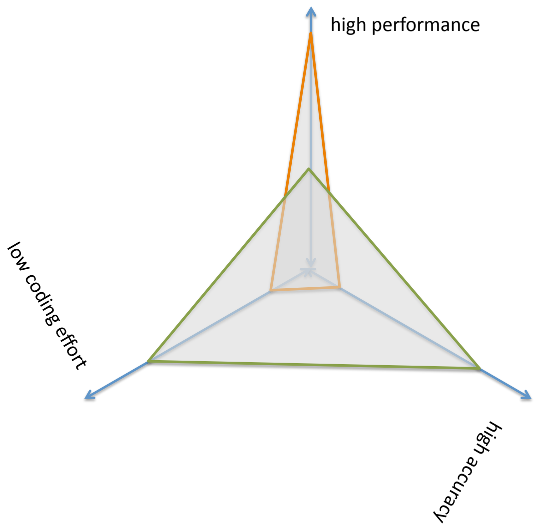

Fig. 1. Optimization problem for transforming original code into GPU enabled code. Our approach tries to maximize the area of the triangle (green) while many authors so far have focused on the speed-up factor

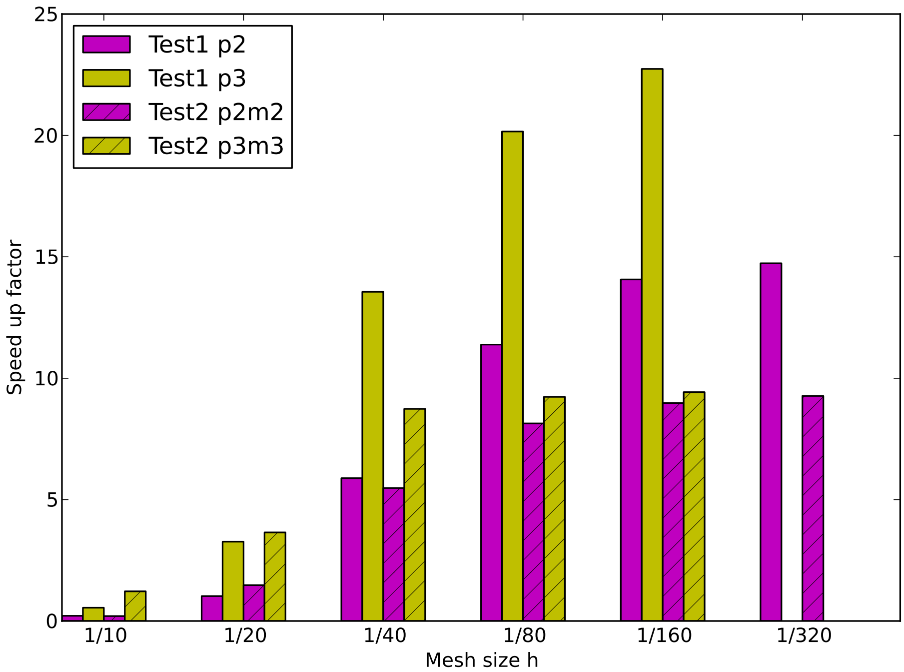

Fig. 2. Speed up for Test 1 and 2

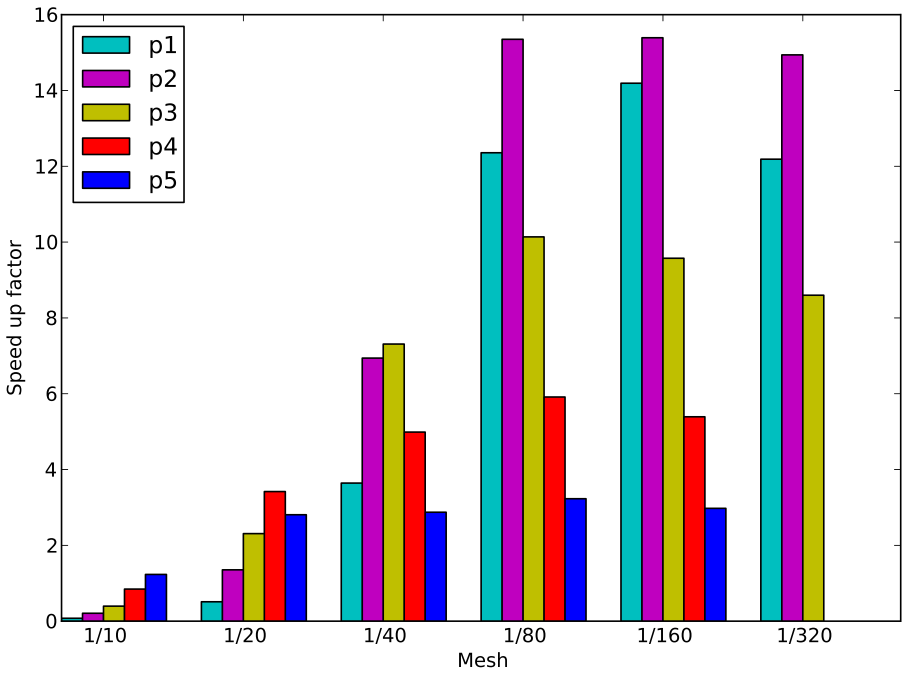

Fig. 3. Speed up for elastic wave equation