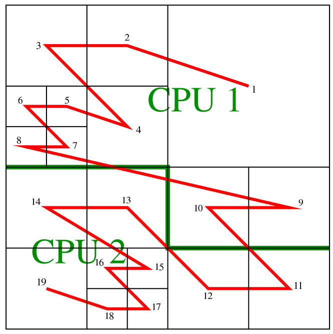
New tree-based gravity solver for octal tree AMR codes

Richard Wünsch,[1,2⋆] Anthony P. Whitworth,[2]

Finite difference hydrodynamic codes traditionally use either spectral methods (FFT, multipole expansion) or multi-grid methods to solve for the gravitational potential. On the other hand, particle based codes (N-body, SPH) typically compute the gravitational potential either by direct integration or by tree based algorithms. We developed an octal tree-based gravity solver for AMR grid codes and implemented it for the publicly available hydrodynamic code FLASH. We provide accuracy and scaling tests and show that the new code is competive with the traditional multi-grid method. Due to its simplicity, this algorithm can be relatively easily implemented for Graphical Processor Units (GPUs) with approximately 100 times more computational power than standard CPUs. As an example, we present simulations of the gravitational fragmentation of an expanding shell - an idealised model of the collect and collapse star formation scenario.

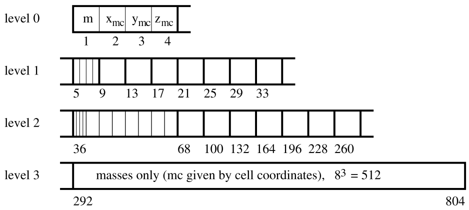


An example of the AMR block structure organized as an octal tree. The red *Morton curve* is used for load balancing (distributing blocks among CPUs).

Algorithm

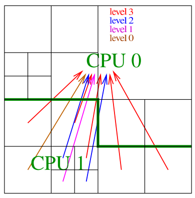
Building tree

- Global tree: follows AMR tree of blocks, stores mass and mass centre positions
- Local trees: additional octal tree for each block; stored by levels in a very compact form in memory



Communication

- Global gravity tree is distributed among all processors
- Local trees are cut to a level necessary on a given CPU; all of them are packed into a sigle message for a given CPU and sent



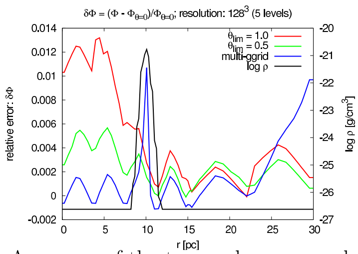
Tree walk

Three types of interaction. A cell c from a block B may interact with:

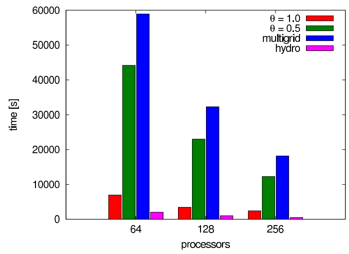
- another cell in the block B or in a neighbouring block
- a tree node (group of cells)
- another block

Due to the regular structure of the grid, interaction lists can be used for the first and often also for the second type of interaction. In the case of the third type, the opening criterion is tested only once for all cells in the block B.

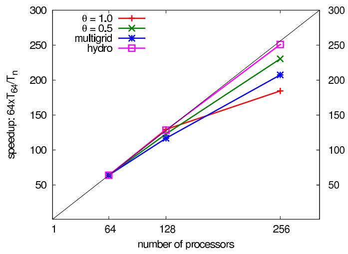
Scaling and accuracy



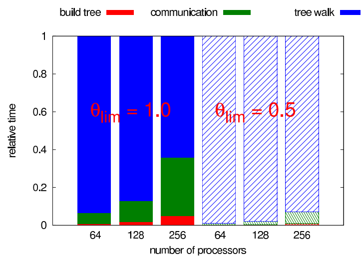
Accuracy of the tree code compared with the mutli-grid method by computing the gravitational potential of the shell on 128^3 grid. The tree code has the maximum error in the shell cavity, the multi-grid method at the peak of the density and at the borders. $\theta_{\rm lim}$ is the limit angle.



Time required for the potential computation by the tree code with $\theta_{\rm lim} = 1.0$ and 0.5 compared with the time required by the multi-grid method. The times are shown for computation on 64, 128 and 256 processors.



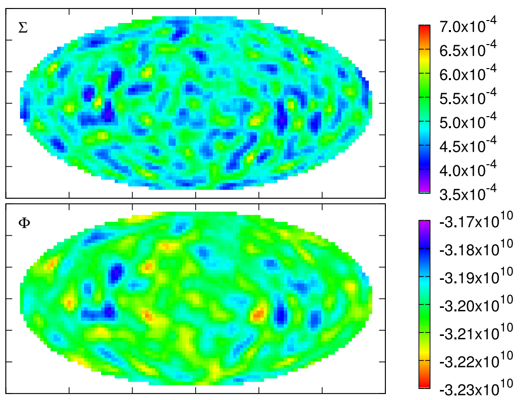
Speed-up (defined as $s = 64T_n/T_{64}$ where T_n is time on n processors) of the tree code with different limit angles, compared with the multi-grid method and with the hydrodynamic computation (the standard Flash PPM Rieman solver).



Relative time required by different parts of the tree code algorithm for two limit angles $\theta_{\rm lim} = 1.0$ and 0.5. The most time-consuming is the tree walk, which is 100% parallel, and therefore a good scaling is expected for even higher numbers of CPUs.

Potential of the expanding shell

We implemented the new tree gravity solver for the hydrodynamic code Flash v2.5 (Fryxel et al. 2000) and used it in 3D simulations of expanding shells. We studied their gravitational fragmentation and compared results obtain by Flash AMR with SPH simulations and with analytical predictions (Dale et al. 2009). The Figure on the right shows distribution of the surface density of the shell in Hammer projection (top panel) and the gravitational potential (bottom). The simulation was run on 256 processors of the ARCCA HPC facility at the Cardiff University.



Future

- we are working on porting the tree code to GPUs based on the nVidia CUDA interface
- quadrupole moments may be easily added for higher accuracy
- optimization, optimization, optimization...

⋆ E-mail: richard@wunsch.cz
[1]Astronomical Institute, Academy of Sciences of the Czech Republic, Boční II 1401, 141 31 Prague, Czech Republic, v.v.i.
[2]Cardiff University

References:
Dale, J. E., Wünsch, R., Whitworth, A. P., Palouš, J., 2009, MNRAS, in press, arXiv:0906.1670
Fryxell, B.; Olson, K.; Ricker, P.; Timmes, F. X.; Zingale, M.; Lamb, D. Q.; MacNeice, P.; Rosner, R.; Truran, J. W.; Tufo, H., 2000, ApJS, 131, 273; (Flash paper)