PERFORMANCE OF HIERARQUICAL PARALLEL PRECONDITIONERS FOR THE FINITE ELEMENT SOLUTION OF LEVEL SET PROBLEMS

Deborah R. de Bruycker*, Jose J. Camata*, Alvaro L.G.A. Coutinho*

* High Performance Computing Center and Dept. of Civil Engineering
COPPE/Federal University of Rio de Janeiro
PO Box 68506, Rio de Janeiro, RJ 21941-598, Brazil
D.R.deBruycker@student.tudelft.nl, camata@nacad.ufrj.br, alvaro@nacad.ufrj.br

ABSTRACT

The level-set method is one of the most successful methods to capture complex moving boundaries [3]. For simplicity, the numerical implementation of the LS method usually employs the same discretization and time marching scheme used for solving the flow equations. Thus, on implicit finite element discretizations, which generally involves solving large sparse systems they often dominate the overall computational time. Due to the increase in computational capability of modern computers, it becomes possible to simulate more complex problems, resulting in larger systems, which need to be solved in parallel. This paper investigates the performance of some new variants of Krylov methods [2], which have originally been developed for the use on parallel machines with over 10,000 cores, when applying them to more moderate core counts which are more commonly available. This performance study has been done using the libMesh library [1], which interfaces with PETSc. The methods under consideration were the improved BiCGStab method, or IBiCGStab, and the hierarchical FGMRES method, or h-FGMRES. These methods are derived from BiCGStab and FGMRES, respectively, by reducing overall communication and synchronization such as to continuously improve computational times when increasing the number of cores. The performance of the new methods has been evaluated against the original algorithms by comparing CPU times and scaling properties. This evaluation was done using two standard level set test cases and by varying several preconditioning parameters. Comparison between BiCGStab and IBiCGStab showed that BiCGStab results in lower CPU times, especially for the lowest core counts. As core counts increase, IBiCGStab seems to become the method of preference as it proofs to posses significantly better scaling properties. A similar behavior could be observed when FGMRES and h-FGMRES were compared towards each other. FGMRES resulted in significantly lower CPU times, however, for the highest core count tested, h-FGMRES started to approach FGMRES nicely. Furthermore it was found that h-FGMRES with the use of more Jacobi-blocks has the best scaling properties. However, for each core count there is a threshold. Only above this point the use of more Jacobi-blocks becomes beneficial. It can be concluded that when core counts will be increased even further, it can be expected that h-FGMRES with a the use of a large number of Jacobi-blocks will be the method with better performance. IBiCGStab, however, also indicated some very promising results.

REFERENCES

[1] Kirk, B. S., Peterson, J. W., Stogner, R. H., & Carey, G. F., "libMesh: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations", *Engineering with Computers*, 22(3–4), 237–254 (2006).

[2] McInnes, L. C., Smith, B., Zhang, H., & Mills, R. T., "Hierarchical Krylov and nested Krylov methods for extreme-scale computing", *Parallel Computing*, 40(1), 17–31 (2014).

[3] Osher, S., & Fedkiw, R. P., "Level set methods and dynamic implicit surfaces", Springer (2003).