8th. World Congress on Computational Mechanics (WCCM8)
5th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2008)
June 30 –July 5, 2008
Venice, Italy

Efficient iterative solver to compute unsteady flows solved with higher order implicit time integration schemes

*** P. Lucas[1], H. Bijl[2]**

[1] Delft University of Technology
Kluyverweg 1, 2629 HS, Delft,
The Netherlands
P.Lucas@TUDelft.NL

[2] Delft University of Technology
Kluyverweg 1, 2629 HS, Delft,
The Netherlands
H.Bijl@TUDelft.NL

Key Words: *unsteady flow simulations, iterative solver,* JFNK*,* ILU*(k) preconditioning.*

ABSTRACT

Over the last decade(s) there has been an increasing demand to predict unsteady flows. Knowledge of unsteady loads on, for example, wind turbine blades have proved to be of vital importance for an efficient life cycle design. Due to the unsteadiness of these problems and the large stiffness associated with large Reynold number flows, these computations can take months on a series of processors. To speed up these computations we use higher order multistage implicit time integrations schemes, because they were shown to be more computationally efficient than the standard backward difference schemes, also for engineering orders of accuracies[1]. However, specially for larger time steps the performance of standard nonlinear multigrid deteriorates. The goal of this research is therefore to further speed up the computations.

Because of the potential speed up of a Jacobian-free Newton-Krylov (JFNK) method over standard nonlinear multigrid for these problems[2] we have implemented a JFNK method in our aerodynamic production code. Preconditioning of the linear systems that arise after Newton linearization is, however, of crucial importance for the computational efficiency of the JFNK method[3]. This paper therefore seeks for an optimal preconditioner to compute unsteady, large Reynolds' number flows solved with higher order implicit time integration schemes. Because higher order implicit time integrations schemes are not yet often used, not much is known on how to create an efficient preconditioner.

Possible preconditioners are: (non)linear multigrid, a recursive variant of GMRES and (matrix-free) approximate factorizations (AF) of the Jacobian. (Non)linear multigrid, matrix-free AF's and GMRESR have the advantage of a low memory consumption. However, they require more computational time per iteration and the low-frequency errors may be poorly damped. AF's of (an approximation of) the Jacobian can be very powerful because errors in the whole frequency domain are damped. Furthermore, once the factorization has been computed it can be reused for next linear solves, which can make them relatively cheap to apply. Finally, this type of preconditioner is relatively straightforward to implement in our aerodynamic production code. We have therefore chosen to precondition the linear systems with a Jacobian based Incomplete Lower Upper factorization based on the footprint of the Jacobian (ILU(k)). Preliminary results showed that these AF's greatly outperformed multilevel AF's and AF's based on dual thresholds.

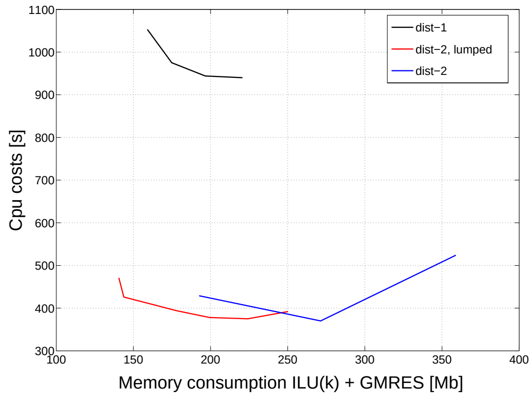

Figure 1: Comparison of different strategies to build the Jacobian from which the ILU(k) is made.

Disadvantages of ILU(k) preconditioners can be lack of robustness[4] and large memory consumption. The lack of robustness is easily circumvented by slightly increasing the diagonal dominance of the Jacobian. A common approach to reduce memory usage is to neglect contributions further away than nearest neighbors in the Jacobian[5]. However, for unsteady two-dimensional flows around wind turbine profiles on unstructured grids, we find a much better linear convergence with an ILU(k) preconditioner that is based on a lumped Jacobian. In Fig. 1 the cpu costs are given as function of the memory consumption of the ILU(k) and Krylov subspace. The total cpu costs are required to solve one physical time step with multiple stages with the JFNK method. The different points in Fig. 1 are found for different levels of fill in of the ILU(k). The ILU(k)'s are computed for different Jacobians: for 'dist-1' all contributions from points further away than nearest neighbors are neglected, while for 'dist-2' all points further away than the neighbors of neighbors are neglected. 'dist-2, lumped' has the same footprint as 'dist-1', however contributions of neighbors of neighbors are lumped to nearest neighbors.

To further enhance the efficiency of the the iterative solver we have investigated the successive combination of nonlinear multigrid and the JFNK method. Furthermore, we are investigating recycling of Krylov subspace to speed up the linear solves. In our paper results for a two and three dimensional unsteady test case are discussed.

REFERENCES

[1] A. H. van Zuijlen, and H. Bijl. *Implicit and explicit higher order time integration schemes for structural dynamics and fluid-structure interaction computations*, Computers & Structures, Vol. 83, 93-105, 2005.

[2] G. Jothiprasad, D.J. Mavriplis, D.J. and D. A. Caughey. *Higher-order time integration schemes for the unsteady Navier-Stokes equations on unstructured meshes*, Journal of Computational Physics, Vol. 191, 542-566, 2003.

[3] D.A. Knoll and D.E. Keyes. *Jacobian-free Newton-Krylov methods: a survey of approaches and applications*, Journal of Computational Physics, Vol. 193, 357-397 2004.

[4] E. Chow and Y. Saad. *Experimental study of ilu preconditioners for indefinite matrices*, Journal of computational and applied mathematics, Vol. 86, 387-414 1997.

[5] P. Wong and D.W. Zingg. "Three-Dimensional Aerodynamic Computations on Unstructured Grids Using a Newton-Krylov Approach". *17^{th} AIAA Computational Fluid Dynamics Conference*, Toronto, Canada, June 2005, AIAA 2005-5231.