**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**

# Extension of Navier-Stokes preconditioners for large-displacement fluid-structure interaction (FSI) problems

*** Jonathan W. Boyle[1], Richard Muddle[2], Andrew L. Hazel[3] and Matthias Heil[4]**

School of Mathematics, University of Manchester
Oxford Road, Manchester, M13 9PL, U.K.

[1] J.Boyle@manchester.ac.uk
[2] MuddleR@cs.man.ac.uk
[3] Andrew.Hazel@manchester.ac.uk
www.maths.manchester.ac.uk/∼ahazel/
[4] M.Heil@maths.man.ac.uk
www.maths.manchester.ac.uk/∼mheil/

**Key Words:** *Krylov Method, Multiphysics Problems, Incompressible Navier Stokes Equations, Preconditioning.*

## ABSTRACT

This talk describes a preconditioning strategy for large-displacement fluid-structure interaction (FSI) problems solved using the monolithic approach, and presents performance data for an implementation that is now included as part of OOMPH-LIB, the open-source object-oriented multi-physics finite element library [1] available at `http://www.oomph-lib.org/`.

In the FSI problems considered here the fluid flow is modelled using the discretised incompressible Navier–Stokes equations, and solid displacements are described using either a discretisation of Kirchhoff–Love shell or beam equations, or the general equations of large-displacement elasticity. In the monolithic approach adopted in OOMPH-LIB the large systems of non-linear algebraic equations that arise from the fully coupled discretisation are solved as a single system using Newton's method.

Within this framework, the most computationally expensive task is the repeated solution of the large linear systems during the Newton iteration. Direct solvers become too expensive with respect to storage as the number of degrees of freedom is increased and are sub-optimal in the sense that time and memory requirements do not scale with problem size. The alternative is to use Krylov-subspace iterative methods, such as GMRES, which require suitable preconditioners.

The FSI preconditioner methodology adopted in OOMPH-LIB is based upon a natural extension of the block preconditioning philosophy that has been shown to work very successfully for the discretised incompressible Navier–Stokes equations. The specific implementation in OOMPH-LIB employs a block-triangular approximation of the full Jacobian matrix as a preconditioner and uses Elman, Silvester

& Wathen's LSC preconditioner [2] on the Navier–Stokes block. The preconditioner is implemented within OOMPH-LIB's general (parallelised) block-preconditioning framework which allows the use of third-party solvers such as SUPERLU [3] or preconditioners from the HYPRE [4] or TRILINOS [5] libraries to exactly or approximately solve the subsidiary linear systems that arise during the application of the preconditioner.

We present GMRES iteration counts and solution times for a range of problems to demonstrate preconditioner performance as a function of uniform and adaptive mesh refinement for a variety of 2D and 3D finite Reynolds number flows discretised using $Q_2Q_1$ elements, with solids discretised using Hermite elements for Kirchhoff–Love beams and shells, and quadratic elements for finite-thickness solids. The numerical results demonstrate convergence rates that do not increase as the mesh is refined and solve times that scale well with respect to problem size and the number of processors.

## REFERENCES

[1] M. Heil and A.L. Hazel. "OOMPH-LIB - An Object-Oriented Multi-Physics Finite-Element Library" *Fluid-Structure Interactions*, Editors: M. Schafer and J. Bungartz, Springer (Lecture Notes on Computational Science and Engineering), pp. 19-49, 2006.

[2] H. Elman, D. Silvester and A. Wathen. *Finite Elements and Fast Iterative Solvers*, Oxford University Press, Oxford, 2005.

[3] J.W. Demmel, S.C. Eisenstat, J.R. Gilbert, X.S. Li, J.W.H. Liu (1999) A supernodal approach to sparse partial pivoting. SIAM J. Matrix Analysis and Applications 20:720-755. `http://crd.lbl.gov/~xiaoye/SuperLU/`

[4] HYPRE – High Performance Preconditioning Library. Center for Applied Scientific Computing at Lawrence Livermore National Laboratory. `http://www.llnl.gov/CASC/hypre/software.html`.

[5] TRILINOS. Sandia National Laboratories. `http://trilinos.sandia.gov/`.