Image Cover Sheet

CLASSIFICATION	SYSTEM NUMBER 513116
UNCLASSIFIED	

TITLE

A Multi-Method Solver for Flow Around Ships

System Number:

Patron Number:

Requester:

Notes:

DSIS Use only:

Deliver to:

A MULTI-METHOD SOLVER FOR FLOW AROUND SHIPS

David Hally, *Defence Research Establishment Atlantic*
E-mail: hally@drea.dnd.ca

INTRODUCTION

One of the primary goals of the Ship Noise Group at DREA is to reduce cavitation noise on Canadian Forces ships. Reduction in cavitation is achieved by improving propeller designs which in turn requires an accurate prediction of the flow into the propeller. This flow is incompressible, has very high Reynolds number (approximately 10^9 based on ship length), can have complex geometry if propeller shafts, brackets, rudders, etc. are included, and has a free surface. Moreover, it is unlikely that the flow near the stern can be modelled adequately if wall functions are used. Consequently, solvers for this application have high memory requirements due to the large number of nodes needed.

For the past few years DREA has been developing a multi-method solver for this application; it is called TRANSOM[1]. This paper is a progress report for the TRANSOM project.

TRANSOM's design allows for a multi-block hybrid grid having different flow solvers on each block: a finite-difference pseudo-compressibility method on structured blocks (called SPC for structured pseudo-compressibility); and a finite element (FE) method on unstructured blocks. However, the implementation of the coupling across block interfaces on hybrid grids is not yet complete, so that currently either a fully structured or unstructured grid must be used.

COUPLED AND DECOUPLED SOLVERS

TRANSOM allows the user a great deal of flexibility in the way in which the solvers on different blocks are executed. The solvers can be assembled into a single global linear system or, to reduce memory requirements, any solver or subset of solvers can be decoupled so that it uses a separate linear system. Because the memory requirements for our application are so high, the possibility of reducing them through decoupled solvers is a useful feature of the code.

Decoupled solvers can use overlapping grids (the possibility of overlapping grids for coupled solvers will be investigated in the future). An example is shown in Fig. 1. It shows the region of flow around a NACA 0012 airfoil split into four grids (the airfoil has been enlarged to make it more visible). TRANSOM was run twice: once using a grid in which all four blocks were coupled in a single solver; and once with two decoupled solvers, one using blocks 1, 2, and 3, and the other using blocks 1, 3, and 4. The convergence histories for the runs are shown in Fig. 2. As expected, the decoupled solvers converge more slowly for the first few iterations; however, eventually they outperform the coupled solver. This is because their linear systems are better conditioned than that of the coupled solver (as neither block has a re-entrant boundary) and because there are two updates of the nodes near the wake in every iteration, since they lie in Block 1 which is common to both decoupled solvers. The decoupled solvers required roughly 40% less memory than the coupled solver.

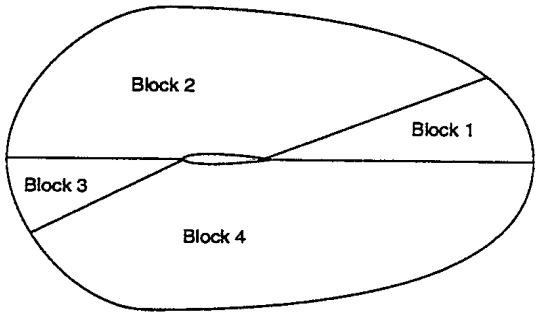

Fig. 1: *Blocks used to calculate the flow around a NACA 0012 airfoil.*

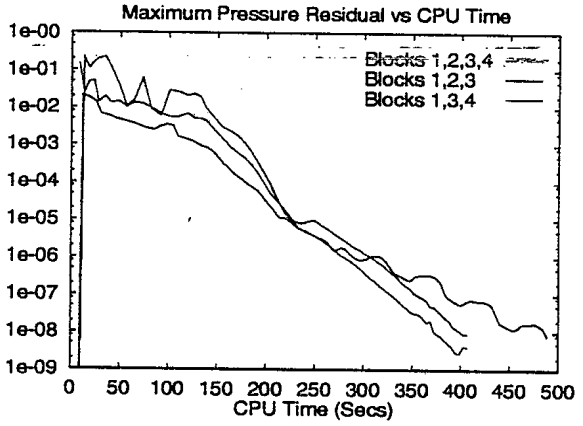

Fig. 2: *Convergence history of coupled and decoupled solvers for the flow around a NACA 0012 airfoil.*

SOLVERS ON STRUCTURED BLOCKS

The TRANSOM solvers on structured block use a pseudo-compressibility method based on the work of Rogers and Kwak[2,3]. The equations of motion with pseudo-compressibility are:

$$\frac{1}{\beta}\frac{\partial p}{\partial t} + \frac{\partial u_m}{\partial x_m} = 0 \tag{1}$$

$$\frac{\partial u_n}{\partial t} + \frac{\partial (u_m u_n)}{\partial x_m} = -\frac{\partial p}{\partial x_n} + \frac{\partial}{\partial x_m}\left[\nu_t\left(\frac{\partial u_n}{\partial x_m} + \frac{\partial u_m}{\partial x_n}\right)\right] \tag{2}$$

where β is the pseudo-compressibility constant, t is pseudo-time, ν_t is the sum of the kinematic viscosity and a turbulent eddy viscosity, and the constant density has been absorbed into p. The Einstein summation convection is used. The equations are integrated forward in time until a steady state is reached. The term in β then vanishes so that the solution also satisfies the steady Navier-Stokes equations.

In a coordinate system aligned along grid lines, (ξ_1, ξ_2, ξ_3), these equations can be rewritten as:

$$\sqrt{g}\frac{\partial p}{\partial t} + \beta\frac{\partial}{\partial \xi_r}\left(\sqrt{g}\frac{\partial \xi_r}{\partial x_m}u_m\right) = 0 \tag{3}$$

$$\sqrt{g}\frac{\partial u_n}{\partial t} + \frac{\partial}{\partial \xi_r}\left(\sqrt{g}U_r u_n\right) = -\frac{\partial}{\partial \xi_r}\left(\sqrt{g}\frac{\partial \xi_r}{\partial x_n}p\right) + \frac{\partial}{\partial \xi_r}\left(\sqrt{g}\nu_t\left[g^{rs}\frac{\partial u_n}{\partial \xi_s} + \frac{\partial \xi_r}{\partial x_m}\frac{\partial \xi_s}{\partial x_n}\frac{\partial u_m}{\partial \xi_s}\right]\right) \tag{4}$$

where

$$U_r = \frac{\partial \xi_r}{\partial x_m}u_m; \quad g^{rs} = \frac{\partial \xi_r}{\partial x_m}\frac{\partial \xi_s}{\partial x_m}; \quad g = \det(g^{rs}) \tag{5}$$

Rogers and Kwak reorganized the second term in square brackets in Eq. (4) using the continuity equation to yield more compact expressions:

$$\sqrt{g}\frac{\partial u_1}{\partial t} = -\frac{\partial}{\partial \xi_1}\left(\nu_t\left[\frac{\partial u_2}{\partial \xi_2} + \frac{\partial u_3}{\partial \xi_3}\right]\right) + \frac{\partial}{\partial \xi_2}\left(\nu_t\frac{\partial u_2}{\partial \xi_1}\right) + \frac{\partial}{\partial \xi_3}\left(\nu_t\frac{\partial u_3}{\partial \xi_1}\right) + \text{other terms} \tag{6}$$

with similar expressions for u_2 and u_3. When implemented with first order three-point finite difference approximations for the derivatives, we have found this reformulation to be unstable, whereas the expression in Eq. (4) causes no difficulties. Therefore, the TRANSOM SPC solver uses the form given in Eq. (4).

Integration in time is implemented by writing Eqs. (3) and (4) as:

$$\frac{\partial F}{\partial t} = -R \tag{7}$$

where $F = (p, u_1, u_2, u_3)$ and the residual, R, contains all terms which do not involve a time derivative. An implicit Euler discretization is used:

$$\frac{\Delta F}{\Delta t} = -R(t + \Delta t) \tag{8}$$

with $\Delta F = F(t+\Delta t) - F(t)$. Expanding the residual in a Taylor series and retaining only the first order term one obtains:

$$\frac{\Delta F}{\Delta t} = -R - \frac{\partial R}{\partial F}\Delta F \tag{9}$$

so that

$$\left[\frac{I}{\Delta t} + \frac{\partial R}{\partial F}\right]\Delta F = -R \tag{10}$$

Because the time discretization is implicit, large time steps can be used leading to quick convergence.

The convective terms are discretized using upwind differencing and flux-difference splitting yielding accuracies up to fifth order. Details of the discretization scheme may be found in Refs. 2, 3, 4, and 5.

Turbulence Modelling

On structured grids the Baldwin-Lomax[6] and Spalart-Allmaras[7] turbulence models have been implemented. The Chen-Patel[8] version of the k-ϵ model is also available on two-dimensional blocks but has not yet been extended to three dimensions.

The turbulence solvers are decoupled from the pressure-velocity solver.

Boundary Conditions at Block Interfaces

Different methods have been tried for solving at the nodes on structured block interfaces when the metrics are discontinuous.

The first method simply interpolates values from either side of the interface. Consider the block interface shown in Fig. 3. The value of a variable, p, at node 0 is determined from the values at nodes 1, 2, ... and nodes -1, -2, ... In the general scheme the

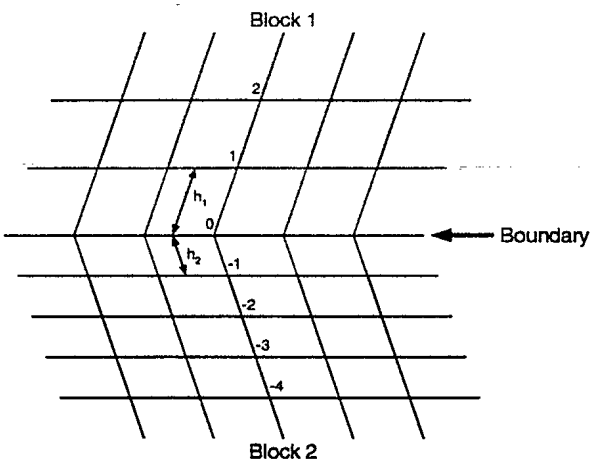

Fig. 3: *Nodes used for the interpolation boundary condition.*

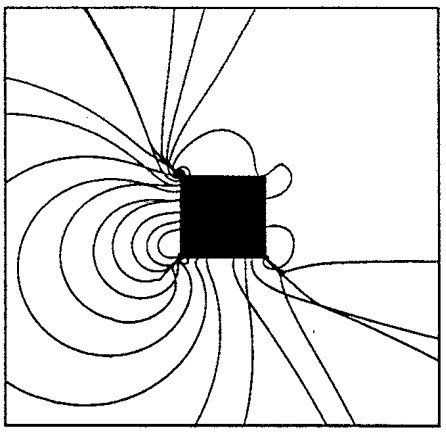

Fig. 4: *Pressure contours using continuous metrics (blue) and discontinuous metrics with interpolation (red).*

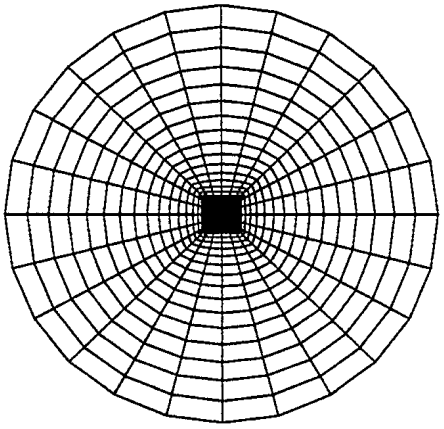

Fig. 5: *The grid used for the calculation of Fig. 4. Not all grid lines are shown.*

values at the nodes in Block 1 are extrapolated to yield a value, p_{b1}, at node zero. Similarly, the nodes in Block 2 are extrapolated to yield p_{b2}. The value of p at node zero is set to the average of p_{b1} and p_{b2} weighted by the grid spacing.

$$p_0 = \frac{h_2 p_{b1} + h_1 p_{b2}}{h_1 + h_2} \tag{11}$$

In the zeroth order scheme, p_{b1} and p_{b2} are simply set to the values at the closest nodes in their respective blocks:

$$p_{b1} = p_1; \qquad p_{b2} = p_{-1} \tag{12}$$

In the first order scheme, p_{b1} and p_{b2} are determined by linear extrapolation from the two closest nodes:

$$p_{b1} = 2p_1 - p_2; \qquad p_{b2} = 2p_{-1} - p_{-2} \tag{13}$$

In the second order scheme the three closest nodes are used:

$$p_{b1} = 3p_1 - 3p_2 + p_3; \qquad p_{b2} = 3p_{-1} - 3p_{-2} + p_{-3} \tag{14}$$

This method performs very poorly. An example is shown in Fig. 4. It shows pressure contours for a two-dimensional flow around a square with the flow from bottom left to top right at an angle of attack of 30°. The grid is shown in Fig. 5. It is a single block with O topology. Not all the grid lines are shown in the figure; there were 161 radial and 121 circumferential grid lines. The re-entrant interface is shown in blue along a line extending diagonally outward from the bottom right corner; it has continuous metrics. The blue contours in Fig. 4 are from a normal TRANSOM calculation on this grid. These contours are an accurate representation of the fully converged solution on a very fine grid.

The grid was also split into four blocks whose interfaces extend outward from each of the corners; they are shown in blue and red in Fig. 5. The red contours in Fig. 4 show the pressure as calculated using these blocks with the first order interpolation boundary condition on each interface (the second order boundary condition would not converge). On a grid in which the metrics really are discontinuous at the interfaces, the interpolation boundary condition performs even worse. Moreover, use of the interpo-

lation boundary condition seriously degrades the robustness of the solver making it much less likely to converge and increasing execution times in the cases in which it does converge.

Recently a first order finite volume formulation was developed for the interface nodes[9]. It performs much better than the interpolation scheme but there are still inaccuracies due to the reduction to first order at the interface.

A third method will be tried shortly: the cells neighbouring the interface will be treated as an unstructured block and a finite-element solver applied. For example, the cells outlined in red in Fig. 3 would be used in a finite element solver to assemble equations for the nodes along the boundary. All other nodes, including those marked 1 and -1 in the figure, would be assembled using the SPC solver.

SOLVERS ON UNSTRUCTURED BLOCKS

On unstructured blocks the solvers are based on the finite element method. In the original implementation of TRANSOM, a penalty function was used to incorporate the continuity equation within the momentum equations[10]. In hindsight, this was a poor choice for two reasons.

First, the linear system to be solved is nearly singular, precluding the use of iterative matrix solvers. Therefore large problems require large amounts of memory.

Second, the way in which the continuity equation is handled is fundamentally incompatible with the SPC solver. The penalty function method requires that the flow be incompressible to high accuracy while the SPC solver attains incompressibility only once a steady state is reached. If SPC solvers on adjoining blocks impose a net flux into an unstructured block, the penalty function solver cannot resolve the pressure correctly. This problem occurs primarily when the SPC and FE solvers are decoupled, but it is part of the TRANSOM design philosophy that the user should be able to couple or decouple solvers at will.

A pseudo-compressibility finite element solver is currently being developed as an alternative.

Turbulence models have not yet been implemented on unstructured blocks.

LINEAR SYSTEM SOLVERS

The linear system in the SPC solver is stored as a sparse matrix and solved using GMRES with preconditioning by incomplete LU factorization with arbitrary fill order (ILU(n))[11]. Because the FE solver cannot use an iterative method, it is stored as a skyline matrix and solved using Gaussian elimination. When the FE solver is updated to use pseudo-compressibility, it will use the same storage method and GMRES routines as the SPC solver.

Recently the ability to use a matrix-free GMRES solver has been added to the SPC solver. This method uses the fact that the linear system matrix is of the form

$$M = \frac{I}{\Delta t} + \frac{\partial R}{\partial F} \tag{15}$$

(see Eq. (10)). The result of a multiplication of vector v by M can be approximated by

$$M \cdot v = \frac{v}{\Delta t} + \frac{\partial R}{\partial F} \cdot v \approx \frac{v}{\Delta t} + \frac{R(F + \epsilon v) - R(F)}{\epsilon} \tag{16}$$

for any small ϵ. Since the GMRES algorithm uses M only in matrix-vector products, there is no need to calculate M explicitly.

This method has obvious attractions for memory limited solvers. However, for GMRES to converge, it is necessary to precondition the matrix. We have found that to obtain a similar level of robustness as for the standard GMRES solver, it is necessary to have a better preconditioner. With our methods this implies a higher level of fill in the ILU(n) preconditioner. The memory required for the preconditioner then negates any advantages obtained from the matrix-free formulation.

The second advantage of the matrix-free method is that the matrix more closely approximates the residual derivative, $\partial R/\partial F$, so that the convergence rates can be much faster. Fig. 6 shows the convergence rates of the maximum pressure residual with GMRES and matrix-free GMRES for the laminar flow past a square obstacle (four block grid with 24400 nodes). There is a dramatic increase in convergence rate when measured by number of iterations, but only a marginal one when measured in CPU time. In comparison with the standard GMRES solver, the matrix-free solver spends much more time calculating residuals and much less in the GMRES routines. Therefore the CPU measure is strongly dependent on the efficiency of calculating the residuals relative to the efficiency of the GMRES algorithm. In the current version of TRANSOM the GMRES algorithm is better optimized than the residual calculation; hence, we believe that CPU performance considerably better than that shown in Fig. 6 can be obtained with

the ship), and on four transverse planes. The colours range from red (lowest velocity) through green to blue (highest velocity). The vortex can be seen as a region of low velocity near the keel. The Reynolds number is 10^9 based on ship length, roughly equivalent to a full size ship. A structured grid of 495,232 nodes was used. The spacing of the nodes closest to the hull is about 10^{-7} ship lengths yielding y_+ values of about 3. The Spalart-Allmaras turbulence model was used.

On the lee side of the ship a vortex runs along the length of the keel, detaches at the stern, and is convected downstream in the wake. The purpose of this simulation was to test that TRANSOM would converge at such high Reynolds numbers rather than as a validation that the flow features were modelled correctly. For a proper validation the number of nodes would need to be increased to model the evolution of the vortex adequately. Moreover, there is no experimental data for a Wigley hull at full scale.

VALIDATION

Comparatively little validation of TRANSOM has been done so far. Validation on realistic hull forms is hampered by the lack of an adequate method for handling three-dimensional block interfaces with discontinuous metrics; therefore only simple grids, and hence only simple hull forms, can be handled. Moreover, experimental data at full ship scale that is good enough for validation purposes is very scarce. Validation against towing tank data (R_e typically in the range 10^6–10^7) and high speed cavitation tunnel data (R_e in the range 10^7–10^8) is in progress.

CONCLUDING REMARKS

TRANSOM is still very much a project in progress. Within the next few months a pseudo-compressibility version of the FE solver will be implemented. Coupling between the SPC and FE solver will then be possible as well as implementation of a general interface condition for block interfaces with discontinuous metrics. It will then be possible to validate the code against a wider range of ship flow data.

REFERENCES

1. D. Hally, "TRANSOM: A Multi-method Navier-Stokes Solver: Overall Design," DREA Technical Memorandum 97/231, 1997.
2. S. E. Rogers and D. Kwak, "An Upwind Differencing Scheme for the Incompressible Navier-Stokes Equations," NASA, Technical Report TN 101051, Nov. 1988.
3. D. Hally, "Implementation of a Pseudo-Compressibility Navier-Stokes Solver in TRANSOM," DREA Technical Memorandum 98/239, 1998.
4. D. J. Hall, D. W. Zingg, and C. R. Ethier, "A Laminar Incompressible Navier-Stokes Flow Solver," DREA Contractor Report CR/93/462, 1993.
5. M. Blanco, D. W. Zingg, and C. R. Ethier, "A Three-Dimensional Incompressible Navier-Stokes Solver," DREA Contractor Report in review, 1999.
6. B. Baldwin and H. Lomax, "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78–257, AIAA 16^{th} Aerospace Sciences Meeting, Reno, Nevada, Jan. 1978.
7. P. R. Spalart and S. R. Allmaras, "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, Technical Report, 1992.
8. H. C. Chen and V. C. Patel, "Near-Wall Turbulence Models for Complex Flows Including Separation," *AIAA Journal* **26**(6), 641–648 (1988).
9. T. Chisholm, D. W. Zingg, and C. R. Ethier, "Two-Dimensional Multiblock Enhancement of TRANSOM," DREA Contractor Report in review, 1999.
10. D. Hally, "Implementation of a Finite-Element Navier-Stokes Solver in TRANSOM," DREA Technical Memorandum 97/225, 1997.
11. D. Hally, "C++ Classes for Sparse Matrices and Iterative Linear System Solvers," DREA Technical Memorandum 98/240, 1998.
12. D. Hally, "Tests of the Sensitivity of Dawson's Panel Method to the Free Surface Panels," Bassin d'Essais des Carènes, Technical Report Etude 2522, No. 6, 1995.

#513116