AIAA 2003–0595

Aerodynamic and Aeroelastic Applications of a Parallel, Multigrid, Unstructured Flow Solver

Sriram and Antony Jameson
Department of Aeronautics and Astronautics
Stanford University, Stanford, CA 94305-3030

41st AIAA Aerospace Science Meeting
January 6–9, 2003/Reno, NV

AERODYNAMIC AND AEROELASTIC APPLICATIONS OF A PARALLEL MULTIGRID UNSTRUCTURED FLOW SOLVER

Sriram and Antony Jameson *
Department of Aeronautics and Astronautics
Stanford University, Stanford, CA 94305-3030

The objective of this study is to develop a flow solver that can be used by aerodynamic design environments to analyze a variety of candidate designs in a relatively short period of time. To meet this goal, a combination of flow solvers developed by the second author, AIRPLANE and FLO77, which are well known for their robustness and accuracy, was parallelized. These flow solvers use unstructured grids to implement the numerical discretization of the governing equations and hence offer the flexibility of handling a variety of geometries through an automated grid generation process. The parallel flow solver was used to analyze aircraft configurations in transonic flight, and also extended to incompressible flows to perform aerodynamic simulations of sail geometries used in the Americas Cup. The turn-around time of the flow solver is typically under 5 minutes and the lift and drag converge in about 50-75 multigrid cycles. To obtain more realistic simulations for the sail configurations, aeroleastic simulations were performed by coupling the flow solver to the commerical finite element package, NASTRAN.

Introduction

Numerical experiments to analyze fluid flow problems are finding increasing use in engineering environments. The development of robust and accurate numerical methods to obtain steady and unsteady flow solutions[12–14] has enabled design environments to incorporate computational methodologies into the design cycle. Growth in computing power and parallel computing environments have added fuel to the development of new and more accurate numerical algorithms and coupled with their ability to provide detailed descriptions of the quantities of interest, computational methods have become an important component in the armor of modern day designers. RANS based simulations are gaining increasing acceptance within the aeronautical community as an alternative to experimental methods, but the lack of completely automated grid generation tools and an incomplete understanding of the phenomenon of turbulence inhibits their wide-spread use. Inviscid assumptions on the governing flow equations have helped aerodynamic design environments to take advantage of the developments in grid generation and flow solution methods. The accuracy and convergence properties of Finite Volume[1], Finite Element[2,3] and Finite Difference[4] approximations to the Euler equations have received much attention over the last two decades and hence robust and efficient numerical algorithms are available for inviscid calculations. Generation of multi-block structured grids around complete aircraft configurations, while seemingly tedious, is still a tractable problem[5,6]. Unstructured grid generation methods have become more robust in the last decade and hence, have made good on the promise to handle complex geometries[7,10]. Numerical discretizations for unstructured grids have also reached a fair degree of maturity[8,11] and it is quite common to see most industrial engineering environments use an unstructured computational tool for aerodynamic analysis and design. An important pre-cursor to most unstructured flow solvers was the AIRPLANE[15] code, which was developed during the mid-eighties. This flow solver was used to obtain some of the first three dimensional simulations over complete aircraft configurations, and the data structures and numerical algorithms used in this code is widely used by the aerodynamic community for transonic and supersonic flow simulations. Later, another computational package, FLO77, was developed by the second author that used multigrid techniques (along with the numerical algorithms used in the AIRPLANE solver) to reduce the computational cost of the flow simulations. With the wide-spread use of parallel computing environments for aerodynamic flow simulations, the turn-around times of the AIRPLANE and FLO77 flow solvers had to be improved for them to be useful in industrial design environments.

The aim of this paper is to outline some of the improvements that have been made to the AIRPLANE code and through applications to aerodynamic and aeroelastic systems, illustrate that this new computational package is an attractive tool that can be used in a design environment either to repeatedly analyze a variety of candidate designs or to make improvements to an existing design through an automated design procedure. Towards meeting this objective, a parallel unstructured multigrid flow solver was first developed. This computational package was then used to simulate the aerodynamic characteristics of two engineering systems, namely aircraft configurations in

*Professor, Stanford University

transonic flight and sail configurations used in the Americas Cup. Aeroelastic simulations to predict the steady flying shape of the sail geometries was also performed by coupling this flow solver to a commercial Finite Element program, NASTRAN. In the following sections, a brief overview of the data structures and numerical algorithms used in the baseline flow solvers is presented first. The methodology and choice of data structures used to parallelize the flow solver is discussed next. Aerodynamic flow simulations of complete aircraft configurations in transonic flight and aeroelastic simulations of sail configurations are discussed in the subsequent sections. Conclusions from the present study and possible directions for future research are outlined in the final section.

Overview of the baseline flow solvers

The AIRPLANE code is based on a cell-vertex discretization on unstructured grids and computes steady state solutions for the Euler equations. A finite-volume method is implemented on the underlying tetrahedral grid and control volumes are formed from perpendicular bisectors of the edges in the mesh. Loops over the edges or faces of the tetrahedra are used to accumulate the fluxes at each vertex. The convective fluxes are constructed from arithmetic averages of the flow variables on each edge. The numerical diffusive fluxes are typically constructed from a JST formulation using a blend of the first and third order differences along the edges. Alternate choices of the diffusive fluxes that are based on a SLIP construction are also provided. To accelerate convergence to steady state, residual averaging techniques are used along with modified Runge-Kutta time stepping schemes and local time-stepping and typically, steady state solutions for complete aircraft configurations are obtained in 300-400 cycles. The robust nature of the numerical implementation within the AIRPLANE code is well known to the aeronautical community.

FLO77, inherits the data structures and the numerical discretization used in the AIRPLANE code and uses multigrid techniques to further accelerate convergence to steady state. The coarser levels used in the multigrid cycle are independently generated from an automated grid generator and transfer coefficients between the meshes are computed in a pre-processing step. The use of multigrid methods allows for transonic flow simulations to be performed in under 50 multigrid cycles.

Parallel implementation of the unstructured multigrid flow solver

To exploit the availability of modern parallel computing platforms, a combination of the AIRPLANE and FLO77 computational programs, was parallelized. Due to the unstructured nature of the computational grid, a wide variety of possible data structures to implement the underlying numerical algorithms exist.

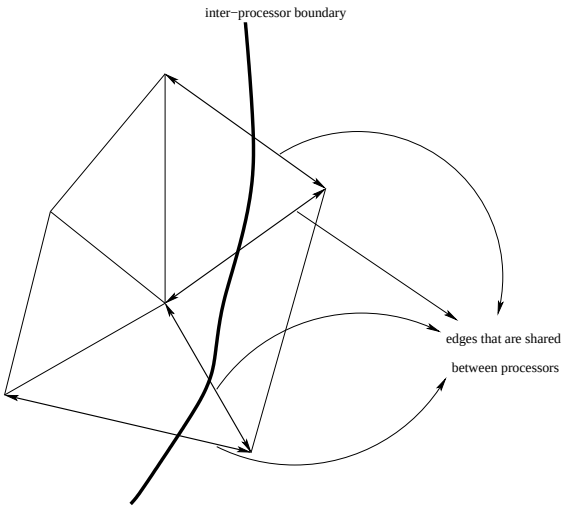


Fig. 1 Halo nodes and the distribution of edges along processor boundaries

The following sections outline the choice of data structures and algorithms that were made to parallelize the flow solver.

Domain Decomposition, Load Balancing

A coordinate bisection method was used to recursively divide a given computational mesh into sub-domains. The sub-domains were created such that they contained approximately the same number of computational nodes. No effort was made to optimize the domain decomposition process so as to minimize the number of edges that are shared between the sub-domains. The sub-domains are distributed among the available processors in a manner that minimizes a combination of the computational cost associated with the domains in each processor and the cost of communication among the processors for a given distribution. This methodology was found to result in well balanced load distributions.

As the flow solver uses an edge-based data structure to accumulate the fluxes at each vertex, the edges surrounding the nodes that lie within a partition are accumulated. If an edge connects a pair of nodes that lie across processor boundaries, this edge is duplicated in the two processors and 'halo' nodes are constructed for both processors. This idea is illustrated further in figure 1.

Parallel implementation of the multigrid algorithm

The use of multigrid techniques for flow analysis necessitates the need to exchange information between fine and coarse grid points and vice-versa. While the residuals from the fine grid points need to be accumulated at the coarse grid points, the corrections from the coarse grid points need to be transferred back to the fine grid points. In this study a non-nested multigrid method has been used, while the the sequence

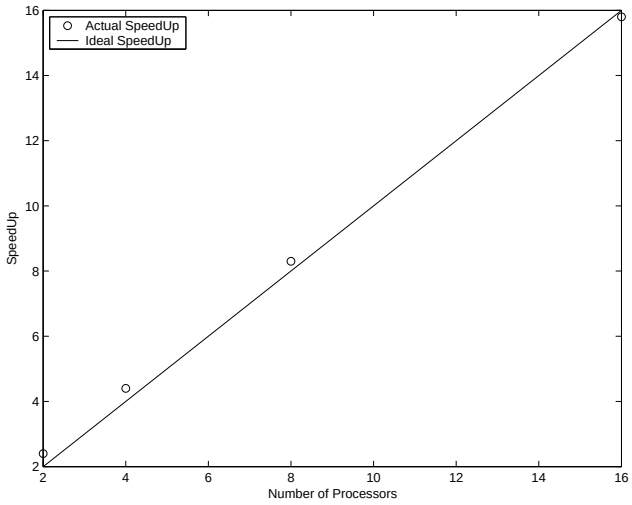


Fig. 2 Speedup from the parallel implementation

of meshes are independently generated from a mesh generator. Hence, efficient point location methods are used to determine the interpolation coefficients associated with each computational node in the multigrid cycle. This search methodology has been implemented using an octree-based search routine. Each grid in the multigrid cycle is recursively divided into octants that contain a certain number of points. Using this data structure, a given point is identified within an octant and the closest node closest in this octant is determined. The tetrahedra connected to this node are checked to see if they contain the search point. Once such a tetrahedron is identified, interpolation coefficients to aggregate the residuals and to interpolate the corrections are constructed. Further, to reduce the communication cost among processors during the transfer of information between the fine and coarse grids, sub-domains on the coarser grids are constructed and distributed so as to conform to the division and distribution of the fine mesh. Typical computational times to build the octree and to build the interpolation tables are in the range of a few minutes for a sequence of meshes containing a million nodes. The efficiency of the parallel implementation is shown in figure 2.

Aerodynamic computations of complete aircraft configurations

The AIRPLANE code has been previously used to analyze a variety of aircraft configurations in transonic flight. These geometries were analyzed again with the parallel multigrid solver. Typically three levels in the multigrid cycle were used in a *W* formation and the finest mesh contained around half a million nodes. The number of points in the coarser levels were typically reduced by a factor of 3-4 by controlling the distribution of points within the domain and on the surface of the geometry. Most modern day grid generators allow considerable flexibility in controlling the distribution of computational points. Hence generating meshes (with the desired coarsening ratios) repeatedly from a mesh generator does not pose any major hurdles. However, it does place considerable burden on the designer and it would be greatly helpful to automate this process through the use of alternate multigrid techniques, like edge-collapsing[16] and agglomeration[17,18]. These alternate multigrid techniques are currently being evaluated by the authors along with other researchers and the results will be reported in a future paper.

Using an unstructured grid generator (MESHPLANE), a series of non-nested meshes to be used in the multigrid cycle is generated. Later, in a pre-processing step, the nodes and edges of each mesh are distributed among the available processors while balancing the load among the processors. The weighting factors used to transfer the solution, aggregate the residuals and interpolate the corrections for the multigrid cycle are also computed in the pre-processing step. The pre-processing step, including the computation of the transfer co-efficients, of a fine mesh with 445,000 nodes and three levels in the multigrid cycle typically requires about 4-5 minutes on a 1.7 Ghz Intel processor with 1Gb RAM. Parallel implementation of this step along with alternate implementation of the load balancing algorithm is being studied, and will become a necessity for viscous applications.

Results from the transonic flow simulation of complete aircraft configurations are shown in figures 3, 4, 5, 6. The convergence histories obtained for these geometries are typical for transonic flow simulations. Typically, the lift and drag of these configurations converge in about 50-75 multigrid cycles. The flow simulations take under 5 minutes when 16 processors of a linux cluster with 1.7 Ghz Athlon processors are used. Hence, this computational tool can be effectively used in a design environment both for the analysis of aerodynamic configurations and to make design improvements.

Aeroelastic Computations to predict the Flying Shape of Upwind Sails

The parallel multigrid solver discussed in the previous section was extended to incompressible flows using the idea of artificial compressibility[20,21]. This incompressible flow solver was then used to obtain the characteristics of sail configurations used in high performance yachts for races like the Americas Cup. Modifications to the inlet boundary condition so as to simulate the boundary layer profile over the sea were incorporated, and a head and main sail combination used in the Americas Cup was analyzed (figure 7). Due to the fast turn-around times of the flow solver it was possible to use this computational package to obtain aerodynamic characteristics of the sail configurations for a variety of wind conditions, and also study the effect of varying twist, camber and sail trim. Further, to predict the flying shape of the sail geometries, this

flow solver was coupled to NASTRAN. The aeroelastic package uses an iterative algorithm that transfers the pressure loading obtained from the flow solver to the structural model, and uses the deflections from the structural analysis to modify the computational mesh for the fluid. This process is iteratively carried out until the deflections are below a particular threshold.

Aerodynamic simulations

A head and main sail combination representative of the sails used in the Americas Cup was analyzed using the aeroelastic package. Leeward and windward side pressure distributions are shown in figure 9 and figure 10 respectively. Three computational grids were used in a *W*-cycle with the finest mesh containing 414,200 nodes. Figure 8 shows the improvement in the convergence histories when multigrid techniques were used. Typically, the lift and the drag converge in under 100 multigrid cycles requiring about 5 minutes when 8 processors of an SGI Origin 300 are used.

The spanwise loading of the head and the main sail are shown in figure 11 and 12. The head sail displays a desirable pressure loading such that the lift gradually tapers towards the tip and has an elliptic distribution. This results in a weaker trailing vortex at the tip and hence reduced drag. Both the lift and the drag exhibit oscillations near the foot, possibly due to the gap between the foot and the symmetry plane. The loading of the main sail displays a gradual increase towards the tip, which is an undesirable feature. This feature becomes more prominent above the head sail, showing that favorable pressure gradients induced by the head sail have a significant influence on the performance of the main sail sections. The presence of large suction peaks on the main sail could be detrimental to the development of the boundary layer and might possibly induce separation. Due to the absence of a viscous model in the present analysis it is difficult to make qualitative estimates of the resulting loss in lift by the main sail. Plots of the pressure profiles at various sections of the head and the main sail suggest that the twist and the camber of the sections are not aligned to allow smooth entry for the incoming airstream. However, the sails were assumed to be rigid for these simulations. Subsequent aeroelastic simulations showed that the aerodynamic loading changes the twist and camber distributions so as to allow the flow to enter the head and the main sail in a smooth manner.

Structural Model

A structural model of the sail was introduced in order to allow for aeroleastic effects. The sail cloth was discretized into quadrilateral membrane finite elements with four nodes (after neglecting the presence of batten pockets). These elements withstand all external forces through tension and are incapable of resisting bending moments. The translational and rotational degrees of freedom along the foot of the main sail were suppressed. Along the mast, the translational degrees of freedom were inhibited while allowing for rotational motion. For the head sail, the point of attachment of the foot to the rig was constrained. The leech of the main and head sail were allowed to move freely to induce a geometric twist due to the aerodynamic loading. The mast was assumed to be rigid during the structural and aeroelastic calculations. The presence of battens and tension cables and other structural elements of the sail rig was neglected from this analysis. The linear system of equations relating the displacements to the force field was advanced to a steady state by an iterative process that incrementally added the load while obtaining a converged displacement field for each step. This computational methodology was included in order to allow for large deflections of the sail geometry. Wrinkling of the structure, which is an important consideration especially around the leading edge (luff) and at the sail tip, is not anticipated by this model, but the use of a numerical algorithm to track large deformations allows for wrinkling models to be included at a later stage. Figures 13 and 14 pictorially depict the boundary conditions used for the structural model.

Aeroelastic coupling

The pressure loading from the flow solver is fed to the structural analysis to estimate the deflected shape of the sail. To enable the transfer of loads and displacements to be conservative, the fluid mesh on the surface and the structural mesh are identical, eliminating the need for interpolation. The deflected shape of the sail is used to deform the computational mesh. The popular 'spring-analogy' method has been used to track the mesh deformations. While this method restricts the allowable deflections and may impair the quality of the deformed mesh, it provides a simple tool to track mesh deformations. The deformed mesh is then used to compute a new pressure loading for the sail. This iterative process offers no formal guarantee of convergence, but it usually predicts the deflected shape to reasonable accuracy in a few steps (typically 5 for sail geometries).

Aeroelastic simulations

The deflected shape of the head and the main sail are shown in figures 15 and 16. It can be seen from these plots that the lower sections of the head and the main sail do not undergo appreciable deformation. The largest deflections occur in the mid-sections of the main sail. As the point of attachment of the main sail to the mast and the leading edge of the head sail were not allowed to move, the aerodynamic loading changes the twist of the sail geometry. This has a favorable influence on the pressure distribution, especially on

the head sail (figures 17 and 18). The pressure distribution over the head and sail after the aeroelastic simulation highlights the need to perform aeroelastic analysis to obtain the flying shape of these sail geometries. While the lift and the drag distribution of the deformed shape is not significantly different from the undeformed shape, the pressure distribution over the sail sections show that the twist and the camber distribution of the head and the main sail have been altered to provide a smooth entry for the flow over the leading edge of both components.

Conclusions and Future directions of Research

A parallel unstructured multigrid flow solver has been developed and shown to be useful for industrial environments. The rapid turn-around times of the flow solver coupled with its robust nature allows designers to use this tool to evaluate a variety of candidate designs. Due to the availability of PC based parallel computing environments at affordable cost, this computational tool can be used by individual researchers and research labs alike. On another front, the development of robust and accurate flow solution methodologies based on unstructured grids enables automated design procedures to be incorporated in the design process. Gradient based methods can be exploited to identify optimal designs. Preliminary efforts on this front have already been completed by the authors and the results will be presented at a future conference. Future directions of research will include viscous models to provide more accurate estimates for the engineering community. Alternate discretizations that provide more accurate representations of the flow-field are also being studied. These discretizations assume no knowledge of the underlying computational grid, and hence can be used for a variety of flow simulations (race cars to sail boats to aircraft configurations in subsonic, transonic and supersonic flight) that use structured, unstructured or cartesian meshes.

Acknowledgments

This work has benifited greatly from the support of the Air Force Office of Scientific Research under grant No. AF F49620-98-1-2002. We would like to thank Prof. Margot Gerritsen for helping us acquire comparative data for the aerodynamic, structural and aeroelastic calculations of the sail geometries. The authors would also like to thank Timothy Baker for providing the mesh generation software.

References

[1]A. Jameson, W. Schmidt and E. Turkel, Numerical Solution of the Euler equations by finite volume methods using Runger-Kutta time stepping schemes. *AIAA Paper 81-1259*, June, 1981.

[2]R. Lohner, K. Morgan, J. Peraire, O.C. Zienkiewicz, Finite Element methods for high speed flows. *AIAA-Paper 85-1531* AIAA 7^{th} Computational Fluid Dynamics Conference, Cincinnati, OH, 1985.

[3]T.J.R. Hughes, L.P. Franca, M. Mallet, A new finite element formulation for Computational Fluid Dynamics, I, Symmetric Forms of the compressible and Navier Stokes and the second law of thermodynamics. *Computational Methods in Applied Mechanical Engineering*, Vol. 59, pp. 223-231, 1986.

[4]S. Osher, Reimann Solvers, the entropy condition, and difference approximations. *SIAM Journal of Numerical Analysis* , Vol. 121, pp.217-235, 1984

[5]N.P. Weatherill, C.A. Forsey, Grid Generation and flow Calculations for aircraft geometries. *Journal of Aircraft*, 22, pp.855-860, 1985

[6]K. Sawada and S. Takanishi, A numerical investigation on wing/nacelle interferences of USB configuration. *AIAA Paper 87-0455*, 25^{th} AIAA Aerospace Sciences Meeting, Reno, January, 1987.

[7]D.J. Mavriplis and S. Pirzadeh, Large-Scale Parallel Unstructured Mesh Computations for 3D High-Lift Analysis. *AIAA Paper 99-0537*, 37^{th} AIAA Aerospace Sciences Meeting, Reno, January, 1999.

[8]T.J. Barth, Apects of unstructured grids and finite volume solvers for the Euler and Navier-Stokes equations. *AIAA Paper 91-0237*, 29^{th} AIAA Aerospace Sciences Meeting, Reno, January, 1994.

[9]T. J. Barth and D. C. Jesperson, The design and application of upwind schemes on unstructured meshes. *AIAA Paper 89-0366*, January 1989

[10]S.E. Cliff and S.D. Thomas, Euler/experimental correlations of sonic boom pressure signatures. *AIAA Paper 91-3276*, AIAA 9^{th} Applied Aerodynamics Conference, Baltimore, Spetember, 1991.

[11]V. Venkatakrishnan, A perspective on unstructured grid flow solvers. *AIAA Journal*, Vol. 34, pp. 533-547, 1996.

[12]A. Jameson, Multigrid algorithms for compressible flow calculations. In W. Hackbusch and U. Trottenberg, editors, *Lecture Notes in Mathematics, Vol. 1228*, pages 166–201. Proceedings of the 2nd European Conference on Multigrid Methods, Cologne, 1985, Springer-Verlag, 1986.

[13]A. Jameson, Steady state solution of the Euler equations for transonic flow. In *Proceedings of Symposium on Transonic, Shock, and Multidimensional Flows*, pages 30–37, Academic Press, New York, 1982.

[14]J. J. Alonso and A. Jameson, Fully-Implicit Time-Marching Aeroelastic Solutions, *AIAA 94-0056*, AIAA 32nd Aerospace Sciences Meeting and Exhibit,Reno, NV, January 1994

[15]A. Jameson and T.J. Baker, Improvements to the Aircraft Euler Method. *AIAA Paper 87-0353*, 25^{th} AIAA Aerospace Sciences Meeting, Reno, January, 1987.

[16]P.I. Crumpton, Implicit time accurate solutions on unstructured dynamic grids. *AIAA paper 95-1671*, 1995

[17]D.J. Mavriplis and V. Venkatakrishnan, A 3D agglomeration multigrid solver for the Reynolds-averaged Navier-Stokes equations on unstructured meshes. *AIAA Paper 95-0345*, 1995

[18]M-H. Lallemand, H. Steve and A. Dervieux, Unstructured Multigridding by Volume Agglomeration: Current Status. *Computers and Fluids* Vol.21, 1992, pp. 397-433

[19]B. Koobus, M-H. Lallemand and A. Dervieux, Unstructured Volume-Agglomeration MG: Solution of the Poisson Equation. *INRIA Research Report* No. 1946, 1993

[20]A. Chorin, A Numerical Method for Solving the Incompressible Viscous Flow problem. *Journal of Computational Physics* Vol. 2, pp 12-26, 1967.

[21]J. Farmer, L. Martinelli and A. Jameson, A Fast Multigrid Method for Solving Incompressible Hydrodynamic Problems with Free Surfaces. *AIAA Paper 93-0767*, 31^{st} AIAA Aerospace Sciences Meeting, Reno, January, 1993.

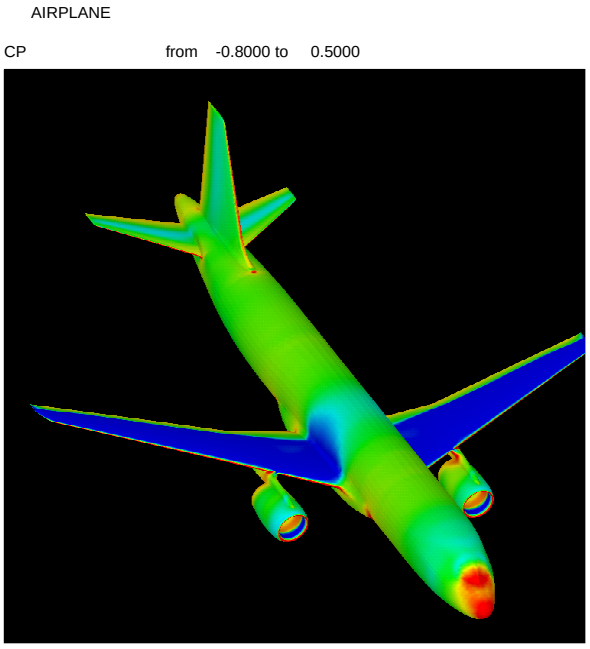


Fig. 3 Pressure contours around the A320 at Mach 0.82 and 2 degrees angle of attack

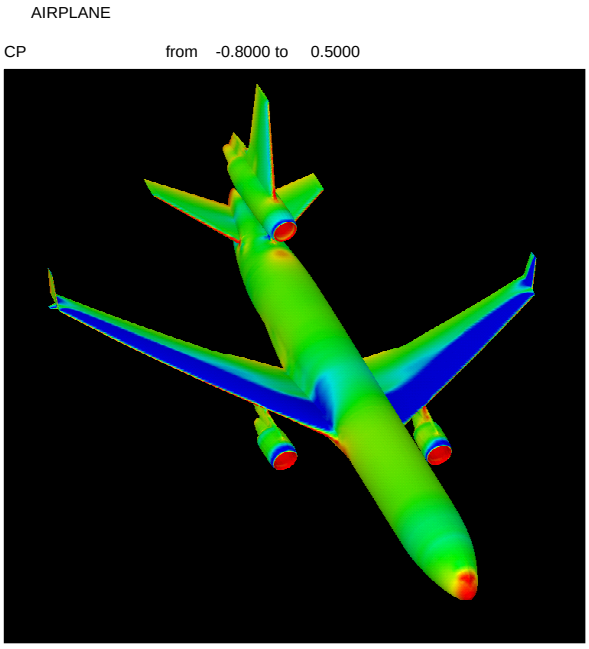


Fig. 5 Pressure contours for the MD-11 at Mach 0.84 and angle of attack of 2.4 degrees

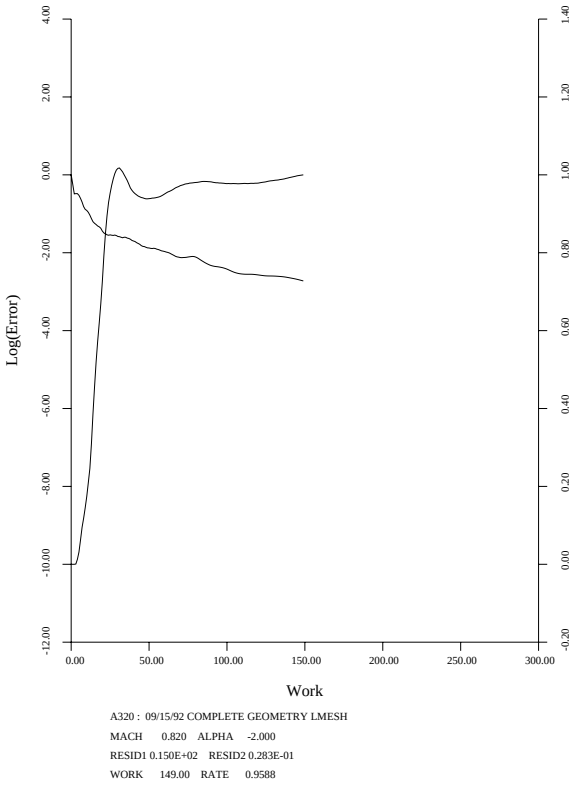


Fig. 4 Convergence history for the A320 configuration

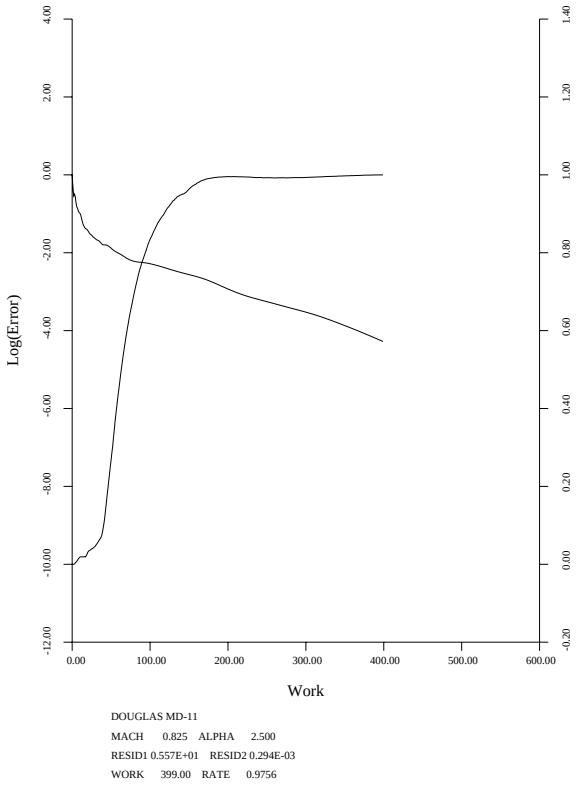


Fig. 6 Convergence history for the MD-11 configuration

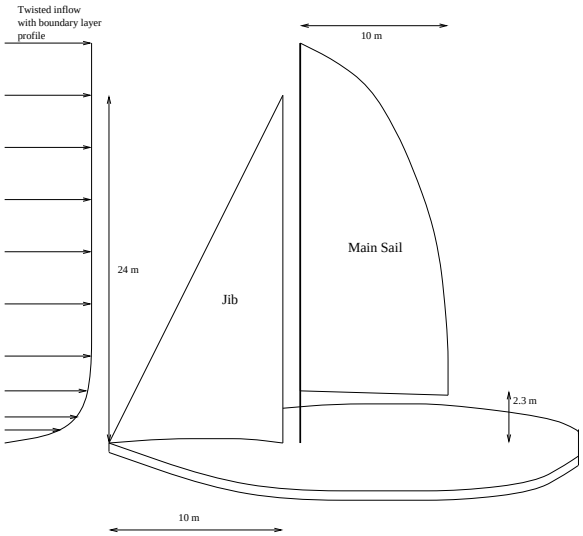


Fig. 7 Sail geometry

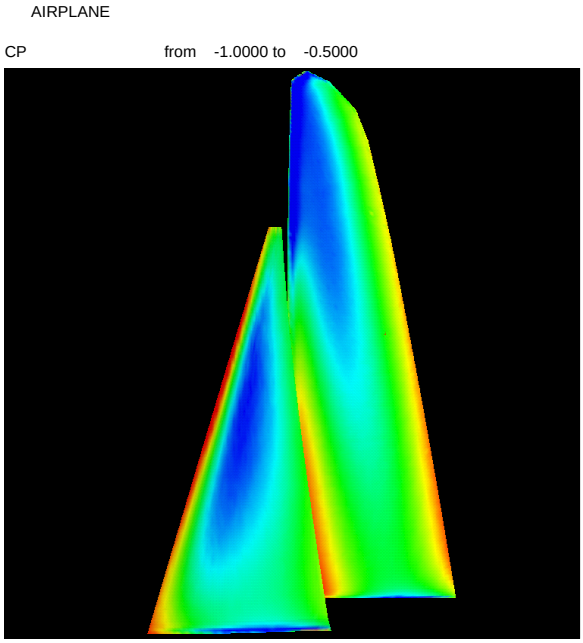


Fig. 9 Pressure contours on the leeward side

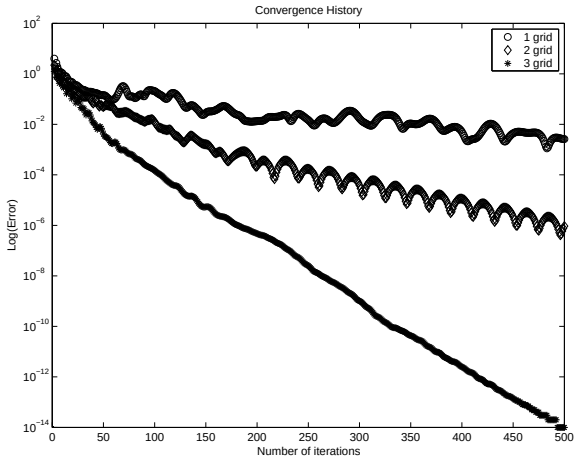


Fig. 8 Convergence history for sail geometries with artificial compressibility

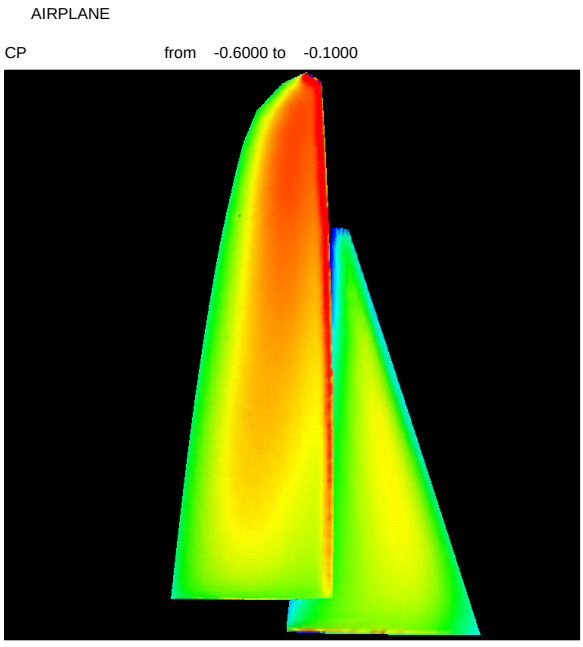


Fig. 10 Pressure contours on the windward side

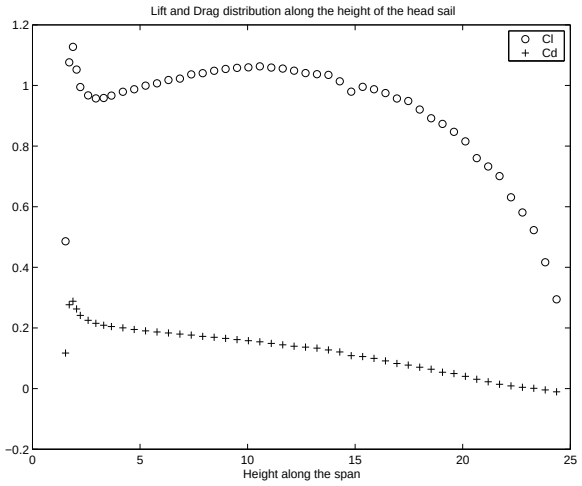


Fig. 11 Spanwise force distributions on the head sail

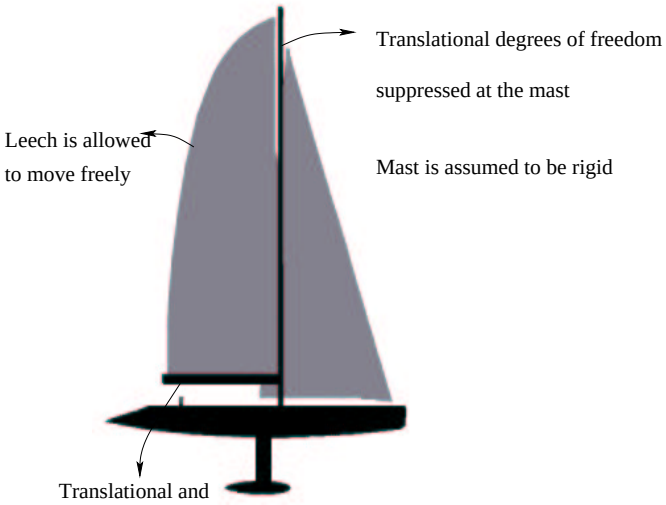


Fig. 13 Boundary conditions for the main sail

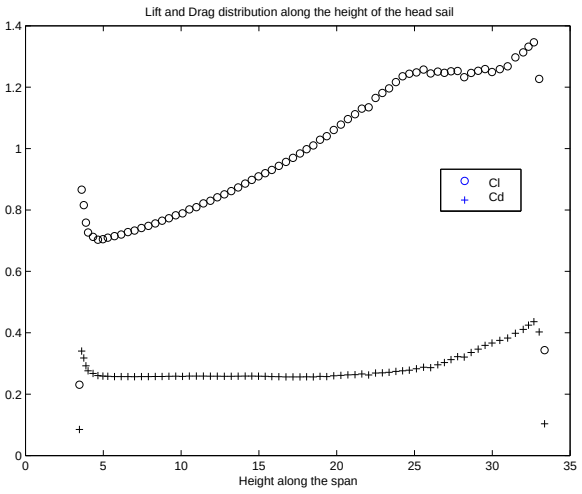


Fig. 12 Spanwise force distributions on the main sail

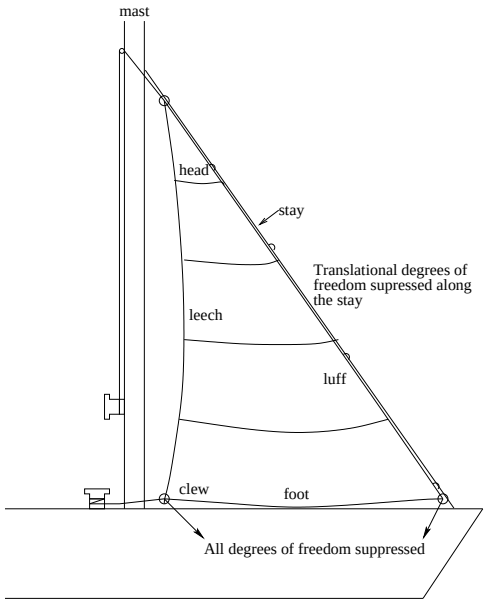


Fig. 14 Boundary conditions for the head sail

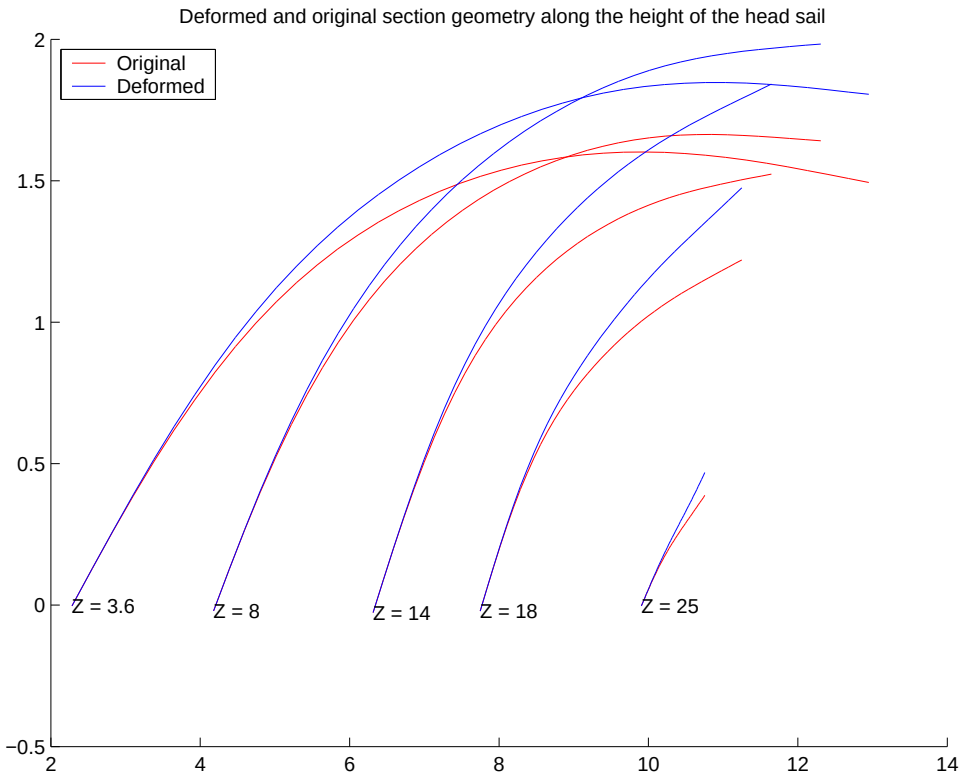


Fig. 15 Original and deformed sail sections for the head sail

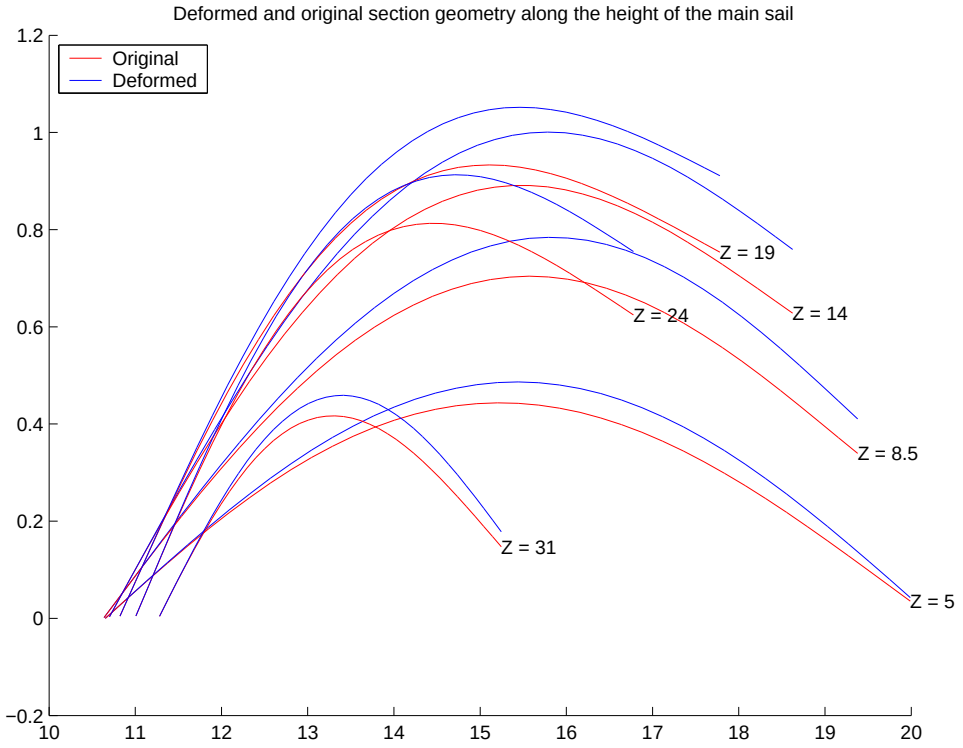
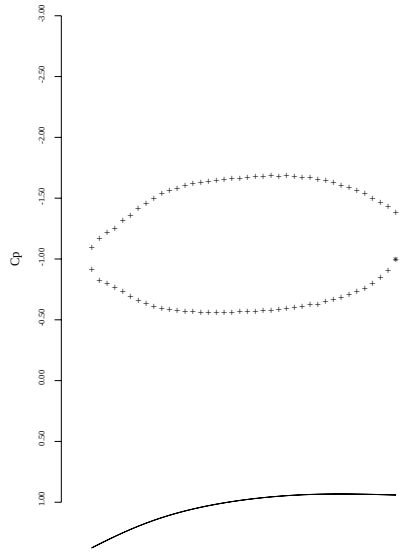
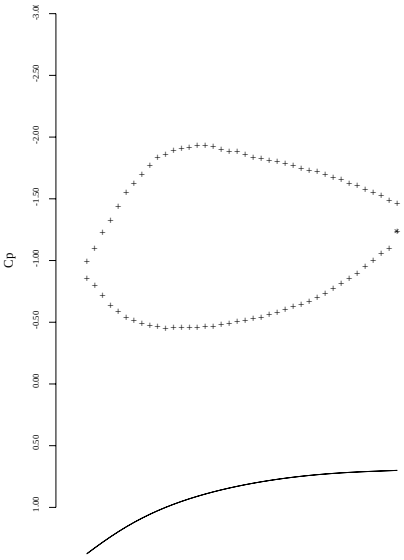


Fig. 16 Original and deformed sail sections for the main sail





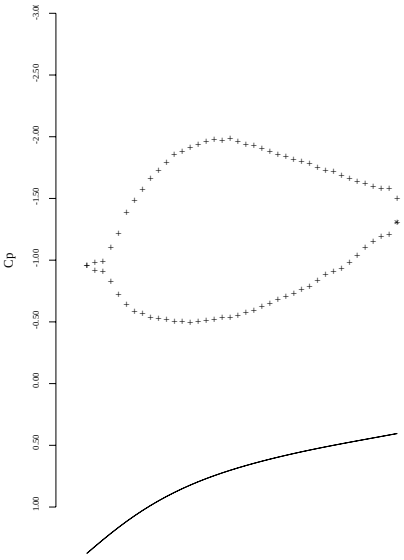
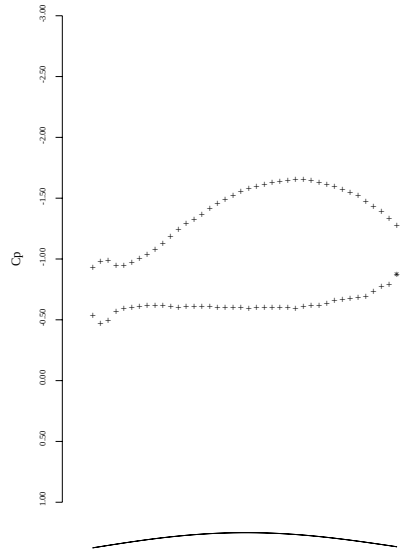
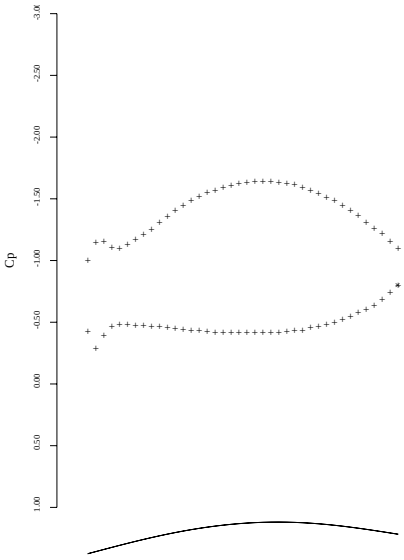


Fig. 17 Pressure distributions along sections at 1, 25 and 85 percent of the height of head sail after aeroelastic analysis





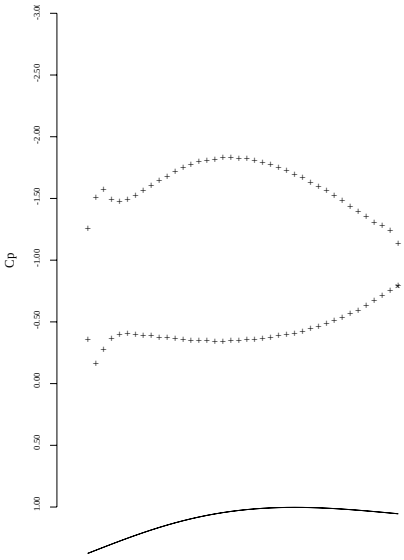


Fig. 18 Pressure distributions along sections at 1, 25 and 85 percent of the height of main sail after aeroelastic analysis