Final Report

on

ADVANCED ADAPTIVE COMPUTATIONAL METHODS FOR NAVIER-STOKES SIMULATIONS IN ROTORCRAFT AERODYNAMICS

Contract #NAS2-13285

National Aeronautics and Space Administration
Ames Research Center

TR-93-02
March 1993

COMCO

Computational Mechanics Company, Inc.
7701 North Lamar, Suite 200
Austin, Texas 78752

Final Report

on

ADVANCED ADAPTIVE COMPUTATIONAL METHODS FOR NAVIER-STOKES SIMULATIONS IN ROTORCRAFT AERODYNAMICS

Contract #NAS2-13285

National Aeronautics and Space Administration
Ames Research Center

TR-93-02
March 1993

COMCO

Computational Mechanics Company, Inc.
7701 North Lamar, Suite 200
Austin, Texas 78752

Prologue

This document is the final report on the SBIR Phase II project conducted by the Computational Mechanics Company, Inc. under Contract A88-41 with NASA Ames. The objective of this Phase II effort was to develop a three-dimensional adaptive computer code for the numerical simulation of transonic flow around multi-bladed helicopter rotors in hover or forward flight. The major issues of concern included:

- Mathematical formulation of finite element based Euler equation in an arbitrary Lagrangian-Eulerian reference frame.
- development of an h-adaptive package along with error estimation capabilities to systematically reduce the error in the solution and captured fine scale flow features with a minimum number of added degrees of freedom.
- development of a grid generation code capable of modeling a variety of blade geometries and fuselage configurations.
- a sliding interface algorithm for modeling the Rotor-Fuselage interaction problem
- development of Explicit/Implicit solution algorithms.

Significant achievement has been made in each of the above areas. Documentation of results, theory and programmers notes are given in their respective manuals. A brief summary of the project including code performance issues and future work are discussed in this final report. A section of sample problems are also enclosed that demonstrate various capabilities.

Contents

1 Introduction

The unsteady flow field surrounding a rotorcraft vehicle in hovering and forward flight encompasses a wide range of complex flow phenomena which includes blade-vortex interaction, spiral vortex sheets, tip vortices, and unsteady effects, to name a few. Accurate modeling of these flow phenomenon is essential for efficient, high performance rotorcraft designs. In particular, a detailed analysis of the wake structure is needed to accurately predict acoustic and vibrational characteristics, as well as the airloads. One standard solution practice is to incorporate models for the tip vortex structure rather than capturing the structure numerically. These methods, however, tend only to be as good as the assumptions employed in the models.

More recent efforts for simulating rotorcraft aerodynamics include finite-difference and finite-volume methods with structured computational grids encompassing the entire rotor blade. However, due to difficulties in capturing the tip vortex structure (insufficient grid resolution) and numerical dissipation, alternative unstructured methods are being pursued.

One of the challenges in modeling fluid dynamics problems is creating a proper mesh for the simulation. In CFD modeling, a proper computational mesh is essential to ensure numerical convergence and solution accuracy. This is, in general, due to the fact that fluid flow problems have extremely large, non-uniform gradients, particularly in boundary layers and near shocks. Often, the location of these flowfield structures are difficult to predict, even for experienced CFD users. Adaptive methods provides a vehicle for automatically identifying and capturing the these types of phenomena associated with rotorcraft aerodynamics. In particular, h-adaptivity attempts to reduce the error in the solution by subdividing selected elements into smaller elements. h-adaptation is particularly useful for capturing flow structures that have sharp discontinuities, such as shocks and flow separation. The current project focus on the ability to anisotropic adaptation: that is, refinement of the mesh independently in the three local element directions, so that no degrees-of-freedom are "wasted".

With rather general set of goals in mind, the development of an unstructured flow solver to model rotorocraft aerodynamics, a Phase II research and development effort was conducted which focused on the following issues:

1. the developing of a finite element based computational fluid dynamics (CFD) analysis tool that combines h-adaptive technology and error estimation.
2. the development of general grid generation package for generating grids for rotorcraft configuration.
3. development of a sliding interface for modeling Rotor-Fuselage type problems (See Figure 1).

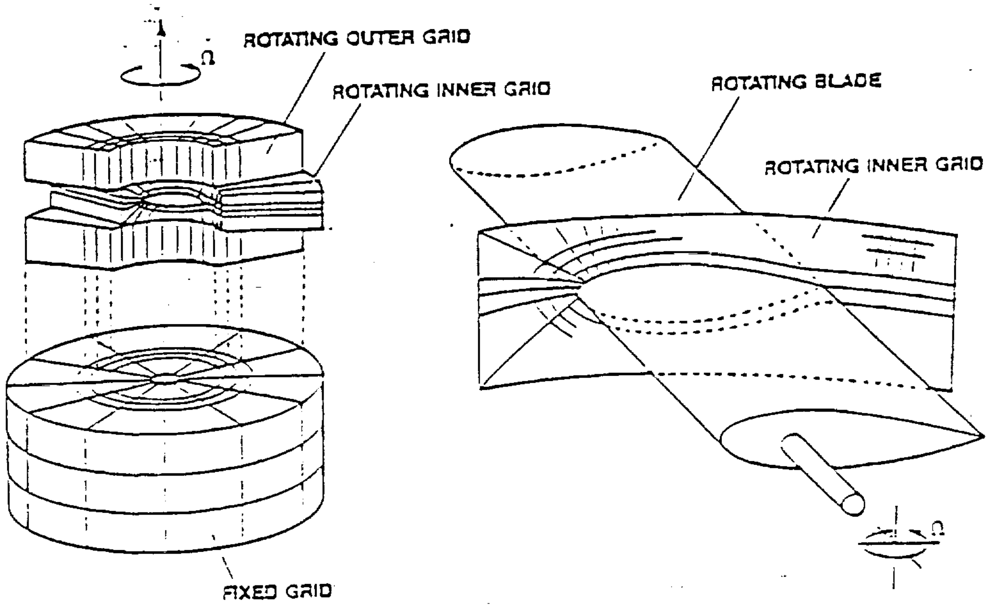


Figure 1: Schematic illustrating the idea of a sliding interface for Rotor-Fuselage Problem.

Presented in the following sections is a summary of the Phase II effort, a discussion of Performance Issues followed by a collection of results and finally an outline of Future Work.

2 Summary of the Phase II Effort

As outlined in the previous section, the Phase II effort has focused on the research and development of a number of ideas and methodologies which may be loosely grouped to include: 1) Adaptive Methods, 2) grid generation and data structure issues, 3) Flow solvers, and 4) the assimilation of parts 1-3 into a working code along with validation. Some general comments with regard to each of these areas follows.

The final code incorporates some features that are common to other software packages under development at COMCO. These include the data structure, the h-adaptive module, the graphics and postprocessing capabilities, and the GUI developed concurrently for our phase II operator splitting research project. Initial support for a phase III effort to further enrich and develop this technology has also been obtained from a partner company and should lead to the commercialization of a code containing some of these features within 12-18 months.

Adaptive methods for the analysis of transonic rotorcraft aerodynamics have been studied extensively during the Phase II effort resulting in the completion of an h-adaptive refinement/unrefinement package along with an error estimation capability to systematically reduce the error in the solution while capturing fine scale flow features with a minimum number of added degrees of freedom. The h-adaptive package employs both isotropic and anisotropic refinement capability. The algorithm has been throughly tested on a variety of problems (See Section 4 for a sample of results) and has demonstrated to be quite robust.

A considerable amount of effort during the Phase II also focused on developing a grid generation package (GAMMA3D) capable of modeling a variety of blade geometries and fuselage configurations. As mentioned earlier, the ability to generate a good quality mesh is of the upmost importance in order to ensure solution accuracy and flow solver stability. GAMMA3D employs both structured and unstructured grid techniques in an attempt to resolve the necessary details associated with multi-bladed configurations. A visualization package (MESHVUR) was also developed to work in conjunction with GAMMA3D as an aid in designing high quality meshes. Both of these tools. GAMMA3D and MESHVUR are complete and have been used in the generation of grids for all validation problems.

A variety of numerical algorithms related to the solution procedure were integrated together to produce a robust multi-blade ROTOR3D code. ROTOR3D has an explicit and implicit time accurate and steady state flow solvers as well as a modified lapidus aritificial dissipation model to aid in stabilizing the solution procedure. A sliding interface for modeling the combined Rotor-Fuselage problem was developed and implemented in ROTOR3D.

The flow solvers currently operational include an explicit lump mass solver and an explicit or implicit GMRES iterative procedure. Along the lines of an integrated Explicit/Implicit solver. several key developments have been incorporated including mesh coloring/grouping based on element type (Explicit or Implicit) and interior versus boundary element, and the generation of Explicit/Implicit lists.

At the current time. ROTOR3D is operational with a few exceptions. All of the grid motion terms have been included in the numerical algorithms. However. only the pure translation terms has been thoroughly tested. Due to performance issues which are discussed in the next section, it became difficult if not impossible to validate ROTOR3D on the three rotorcraft benchmark problems (i.e. Rotor Hover, non-lifting forward flight, and Rotor-Fuselage interaction).

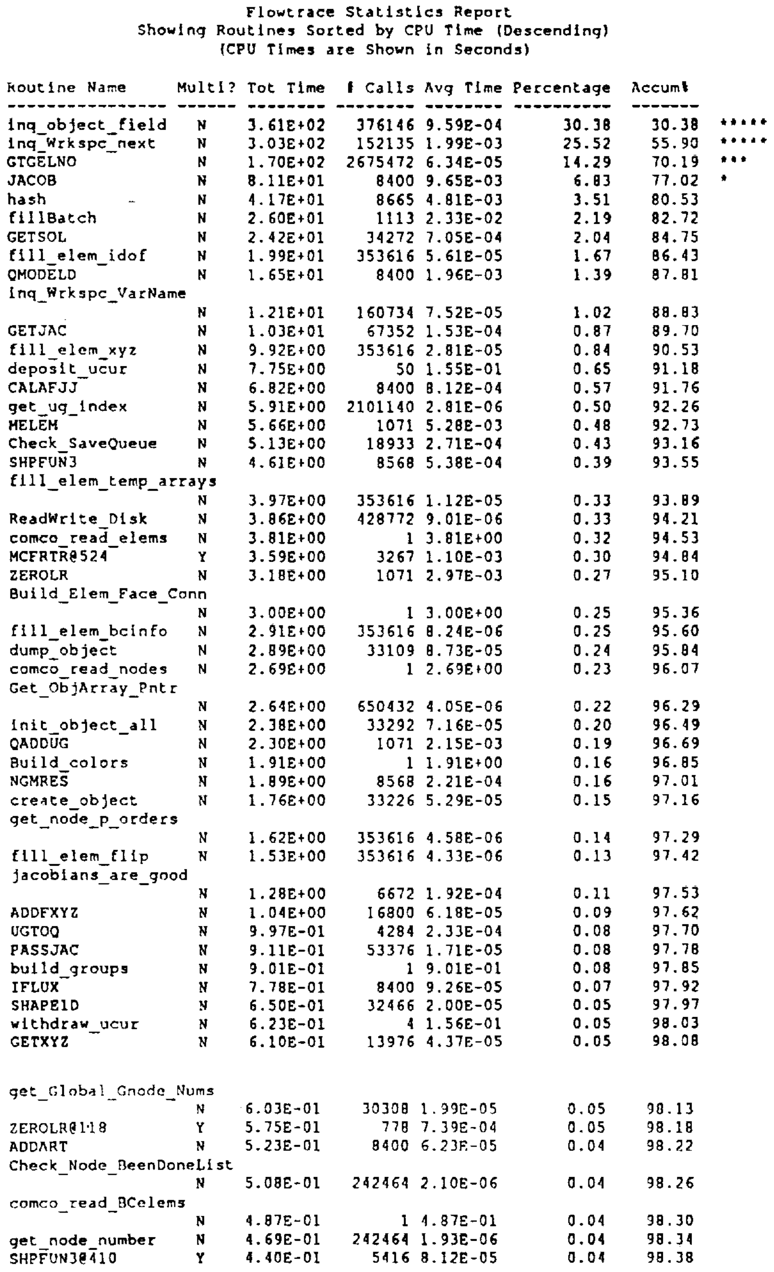
3 Performance Issues

In order to solve the large scale problems associated with rotorcraft aerodynamics where grid sizes normally exceed 20,000 degrees of freedom, code performance is a primary concern. In particular, the average operational vector length (VL) and the floating point operations per second (FLOPS or MFLOPS for millions of FLOPS) are two key performance indicators.

In this regard, we have profiled the ROTOR3D code on the CRAY YMP. This effort has revealed several bottlenecks associated with the performance which were not identified during the design and initial testing phases. The first of these bottlenecks was the general overall poor performance of the code, which had an average VL of only 11.1 with a corresponding megaflop rating of 1.5. This was a surprising result, primarily due to the fact that great care and a significant effort were placed on the design of the data base and the handling of the element computations in terms of groups and batches to maximize the vector length. In an effort to more closely identify which parts of the code were causing the poor performance, the computations associated with the boundary conditions were disabled. The resulting performance indicators showed that the VL increased to 54.5 and the megaflop rating increased to 21.3 (this VL of 54.5 is out of a maximum of 64 on the YMP, which is quite good, however, the MFLOPS are still quite low). Based on this simple experiment, it is apparent that the boundary conditions are completely scalar bound at the present time and that, even with the current grouping and coloring strategy, any gain in vectorization through the volume integrals is damped by the boundary integrals. We believe that a possible remedy to this BC related problem is to modify the grouping and coloring algorithm to collect elements with common boundary types and faces together.

A second problem that appeared on the Flowtrace Statistics report (see Fig. 2) is that when the boundary contributions are ignored, the three routines which appear at the top of the cpu timings are routines which retrieve data from the data base. This problem was not anticipated as the data base and data base access was initially prototyped on an SGI workstation and profiled with the PIXIE option, which did not flag any of the data base access routines. Referencing the YMP profile data, as shown in Fig. 2, it is apparent that the large use of CPU time associated with the data base retrieval does not necessarily come from slow data base access but more likely from the large number of calls to the data base. A possible remedy for this data base access issue would be to include much of the constant scalar data that is being loaded from the data base in either a temporary workspace or a common block. To incorporate such a remedy would require moderate restructuring of routines to hold the scalar parameters in common and would reduce the number of calls to the data base. One positive comment about the flowtrace report is that the calls to the data base to load up the groups and batches of data (ELDBRAT) does not appear to be significant.

As mentioned above, the ROTOR3D code was profiled in an SGI workstation using the PIXIE option (see Fig. 3). The results shown there flag the large CPU usage associated with the routines performing the boundary calculations and, in general, do not show the data base access to be an issue. Based on these performance numbers and the size of the average computational domain associated with rotorcraft simulations, significant performance enhancements in the code are required before any of the large scale computations associated with the benchmark problems can be reasonably performed and fully converged. In general, based on the MFLOPS obtained from the Flowtrace Statistics, a performance increase on the order of 15 to 25 (in terms of MFLOPS) is required to provide competitive performance numbers and run times.

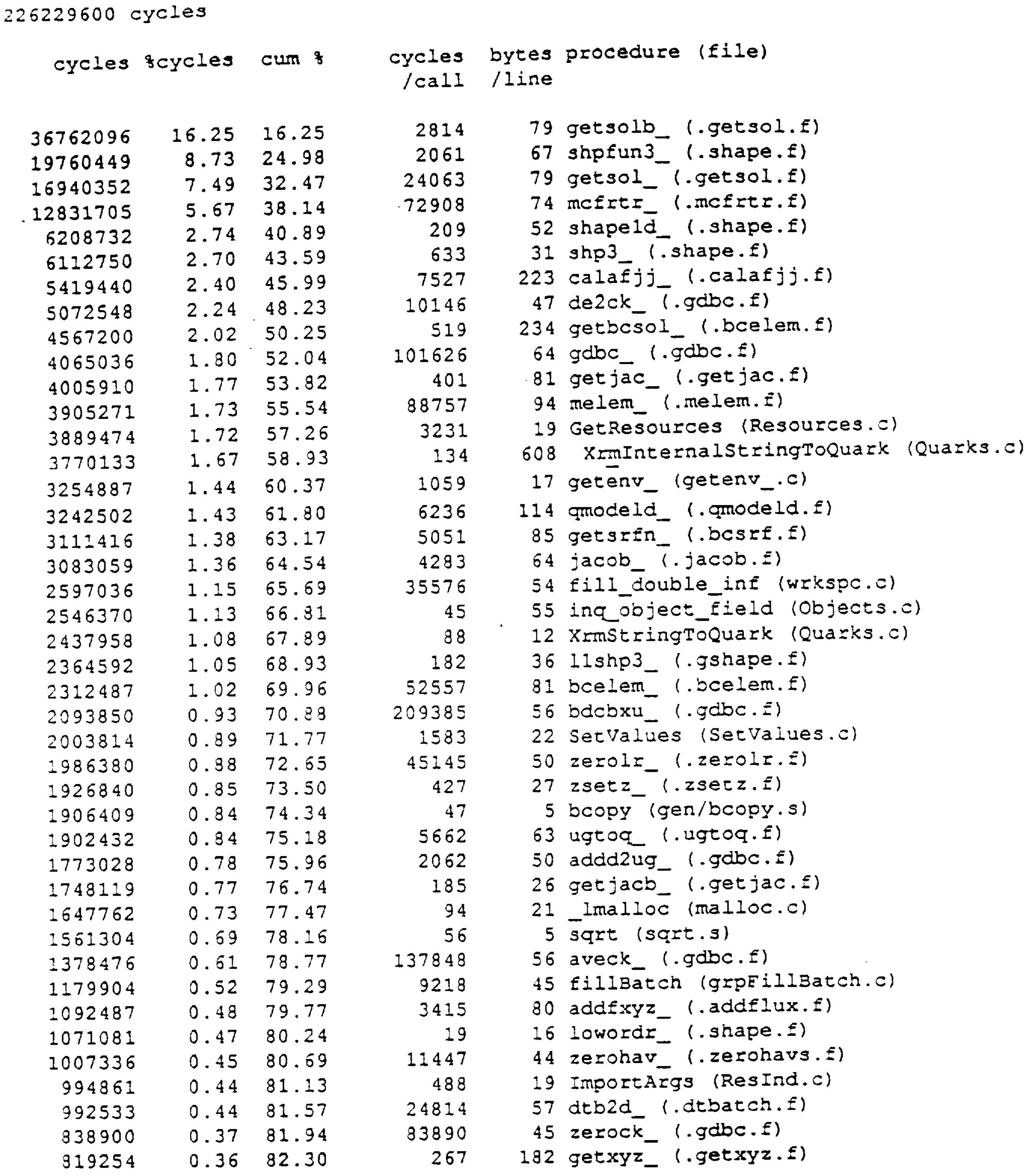


```
                        Flowtrace Statistics Report
               Showing Routines Sorted by CPU Time (Descending)
                       (CPU Times are Shown in Seconds)

Routine Name      Multi? Tot Time  # Calls Avg Time Percentage  Accum%
---------------- ------ -------- -------- -------- ----------  -------
inq_object_field   N    3.61E+02   376146 9.59E-04      30.38    30.38  *****
inq_Wrkspc_next    N    3.03E+02   152135 1.99E-03      25.52    55.90  *****
GTGELNO            N    1.70E+02  2675472 6.34E-05      14.29    70.19  ***
JACOB              N    8.11E+01     8400 9.65E-03       6.83    77.02  *
hash          _    N    4.17E+01     8665 4.81E-03       3.51    80.53
fillBatch          N    2.60E+01     1113 2.33E-02       2.19    82.72
GETSOL             N    2.42E+01    34272 7.05E-04       2.04    84.75
fill_elem_idof     N    1.99E+01   353616 5.61E-05       1.67    86.43
QMODELD            N    1.65E+01     8400 1.96E-03       1.39    87.81
inq_Wrkspc_VarName
                   N    1.21E+01   160734 7.52E-05       1.02    88.83
GETJAC             N    1.03E+01    67352 1.53E-04       0.87    89.70
fill_elem_xyz      N    9.92E+00   353616 2.81E-05       0.84    90.53
deposit_ucur       N    7.75E+00       50 1.55E-01       0.65    91.18
CALAFJJ            N    6.82E+00     8400 8.12E-04       0.57    91.76
get_ug_index       N    5.91E+00  2101140 2.81E-06       0.50    92.26
MELEM              N    5.66E+00     1071 5.28E-03       0.48    92.73
Check_SaveQueue    N    5.13E+00    18933 2.71E-04       0.43    93.16
SHPFUN3            N    4.61E+00     8568 5.38E-04       0.39    93.55
fill_elem_temp_arrays
                   N    3.97E+00   353616 1.12E-05       0.33    93.89
ReadWrite_Disk     N    3.86E+00   428772 9.01E-06       0.33    94.21
comco_read_elems   N    3.81E+00        1 3.81E+00       0.32    94.53
MCFRTR@524         Y    3.59E+00     3267 1.10E-03       0.30    94.84
ZEROLR             N    3.18E+00     1071 2.97E-03       0.27    95.10
Build_Elem_Face_Conn
                   N    3.00E+00        1 3.00E+00       0.25    95.36
fill_elem_bcinfo   N    2.91E+00   353616 8.24E-06       0.25    95.60
dump_object        N    2.89E+00    33109 8.73E-05       0.24    95.84
comco_read_nodes   N    2.69E+00        1 2.69E+00       0.23    96.07
Get_ObjArray_Pntr
                   N    2.64E+00   650432 4.05E-06       0.22    96.29
init_object_all    N    2.38E+00    33292 7.16E-05       0.20    96.49
QADDUG             N    2.30E+00     1071 2.15E-03       0.19    96.69
Build_colors       N    1.91E+00        1 1.91E+00       0.16    96.85
NGMRES             N    1.89E+00     8568 2.21E-04       0.16    97.01
create_object      N    1.76E+00    33226 5.29E-05       0.15    97.16
get_node_p_orders
                   N    1.62E+00   353616 4.58E-06       0.14    97.29
fill_elem_flip     N    1.53E+00   353616 4.33E-06       0.13    97.42
jacobians_are_good
                   N    1.28E+00     6672 1.92E-04       0.11    97.53
ADDFXYZ            N    1.04E+00    16800 6.18E-05       0.09    97.62
UGTOQ              N    9.97E-01     4284 2.33E-04       0.08    97.70
PASSJAC            N    9.11E-01    53376 1.71E-05       0.08    97.78
build_groups       N    9.01E-01        1 9.01E-01       0.08    97.85
IFLUX              N    7.78E-01     8400 9.26E-05       0.07    97.92
SHAPE1D            N    6.50E-01    32466 2.00E-05       0.05    97.97
withdraw_ucur      N    6.23E-01        4 1.56E-01       0.05    98.03
GETXYZ             N    6.10E-01    13976 4.37E-05       0.05    98.08


get_Global_Gnode_Nums
                   N    6.03E-01    30308 1.99E-05       0.05    98.13
ZEROLR@118         Y    5.75E-01      778 7.39E-04       0.05    98.18
ADDART             N    5.23E-01     8400 6.23E-05       0.04    98.22
Check_Node_BeenDoneList
                   N    5.08E-01   242464 2.10E-06       0.04    98.26
comco_read_BCelems
                   N    4.87E-01        1 4.87E-01       0.04    98.30
get_node_number    N    4.69E-01   242464 1.93E-06       0.04    98.34
SHPFUN3@410        Y    4.40E-01     5416 8.12E-05       0.04    98.38
```

Figure 2: Flowtrace of ROTOR3D neglecting boundary contributions – 50 steps.



```
226229600 cycles

   cycles %cycles  cum %      cycles  bytes procedure (file)
                               /call  /line

 36762096   16.25  16.25        2814     79 getsolb_ (.getsol.f)
 19760449    8.73  24.98        2061     67 shpfun3_ (.shape.f)
 16940352    7.49  32.47       24063     79 getsol_ (.getsol.f)
 12831705    5.67  38.14       72908     74 mcfrtr_ (.mcfrtr.f)
  6208732    2.74  40.89         209     52 shape1d_ (.shape.f)
  6112750    2.70  43.59         633     31 shp3_ (.shape.f)
  5419440    2.40  45.99        7527    223 calafjj_ (.calafjj.f)
  5072548    2.24  48.23       10146     47 de2ck_ (.gdbc.f)
  4567200    2.02  50.25         519    234 getbcsol_ (.bcelem.f)
  4065036    1.80  52.04      101626     64 gdbc_ (.gdbc.f)
  4005910    1.77  53.82         401     81 getjac_ (.getjac.f)
  3905271    1.73  55.54       88757     94 melem_ (.melem.f)
  3889474    1.72  57.26        3231     19 GetResources (Resources.c)
  3770133    1.67  58.93         134    608  XrmInternalStringToQuark (Quarks.c)
  3254887    1.44  60.37        1059     17 getenv_ (getenv_.c)
  3242502    1.43  61.80        6236    114 qmodeld_ (.qmodeld.f)
  3111416    1.38  63.17        5051     85 getsrfn_ (.bcsrf.f)
  3083059    1.36  64.54        4283     64 jacob_ (.jacob.f)
  2597036    1.15  65.69       35576     54 fill_double_inf (wrkspc.c)
  2546370    1.13  66.81          45     55 inq_object_field (Objects.c)
  2437958    1.08  67.89          88     12 XrmStringToQuark (Quarks.c)
  2364592    1.05  68.93         182     36 llshp3_ (.gshape.f)
  2312487    1.02  69.96       52557     81 bcelem_ (.bcelem.f)
  2093850    0.93  70.88      209385     56 bdcbxu_ (.gdbc.f)
  2003814    0.89  71.77        1583     22 SetValues (SetValues.c)
  1986380    0.88  72.65       45145     50 zerolr_ (.zerolr.f)
  1926840    0.85  73.50         427     27 zsetz_ (.zsetz.f)
  1906409    0.84  74.34          47      5 bcopy (gen/bcopy.s)
  1902432    0.84  75.18        5662     63 ugtoq_ (.ugtoq.f)
  1773028    0.78  75.96        2062     50 addd2ug_ (.gdbc.f)
  1748119    0.77  76.74         185     26 getjacb_ (.getjac.f)
  1647762    0.73  77.47          94     21 _lmalloc (malloc.c)
  1561304    0.69  78.16          56      5 sqrt (sqrt.s)
  1378476    0.61  78.77      137848     56 aveck_ (.gdbc.f)
  1179904    0.52  79.29        9218     45 fillBatch (grpFillBatch.c)
  1092487    0.48  79.77        3415     80 addfxyz_ (.addflux.f)
  1071081    0.47  80.24          19     16 lowordr_ (.shape.f)
  1007336    0.45  80.69       11447     44 zerohav_ (.zerohavs.f)
   994861    0.44  81.13         488     19 ImportArgs (ResInd.c)
   992533    0.44  81.57       24814     57 dtb2d_ (.dtbatch.f)
   838900    0.37  81.94       83890     45 zerock_ (.gdbc.f)
   819254    0.36  82.30         267    182 getxyz_ (.getxyz.f)
```

Figure 3: SGI profile of ROTOR3D using pixie.

4 Sample Results

In this section, we present the results of five sample problems that have been solved up to various stages during the testing and validation phases of the project. These problems include;

- NACA 0012 airfoil
- Ni Bump - 10% arc
- Fixed 3D wing
- Rotor hover
- Rotor-fuselage interaction

The particular details regarding the selection of the various flow conditions, adaptive parameters, initial conditions, and numerical results are discussed in the subsections below.

4.1 NACA 0012 Airfoil

A NACA 0012 airfoil with an approach Mach number 0.85 at 1° angle attack was selected as the first example problem. The initial mesh consisting of 149 quadratic elements (geometry only) is shown in figure 1. Openflow boundary conditions were imposed on all sides.

The initial mesh was adapted 3 different times during the calculations. Each time the mesh was adapted the refinement/unrefinement parameters were set to 0.95 and 0.45 respectively. The convergence tolerance was specified as 1×10^{-4} during each of the solution passes.

Numerical results for this case are shown in figures 5 - 7. Highlighted in figure 5 is the final adapted mesh with approximately 2509 quadratic elements. Pictured in figure 6 and 7 are contours of Mach number and pressure. Qualitively the results are good, however, there appears to be an overshoot near the traveling shocks.

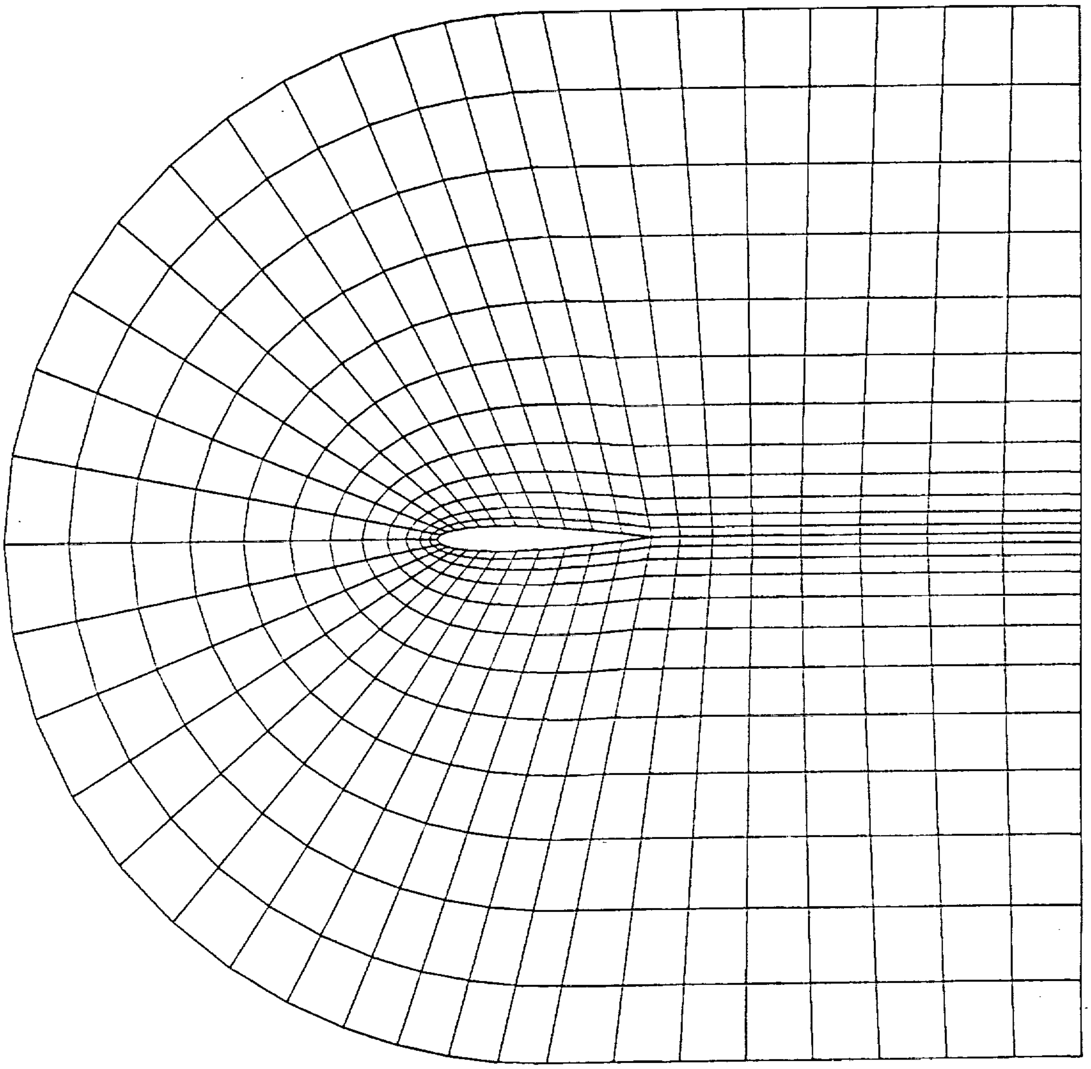


Figure 1: Initial grid for the NACA 0012 airfoil.

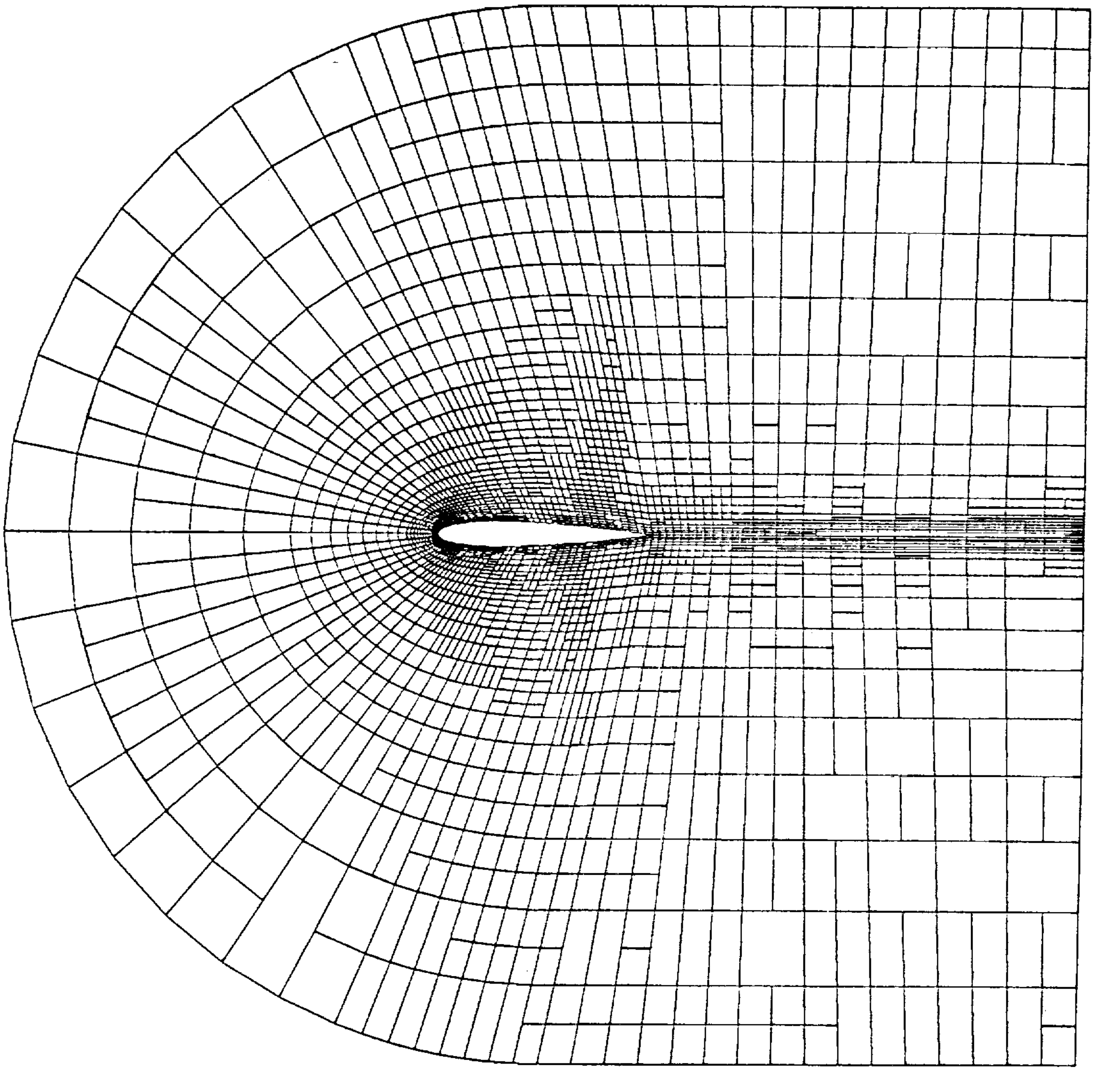


Figure 5: Final adapted mesh for the NACA 0012 airfoil with approach Mach number 0.85 at 1° angle of attack.

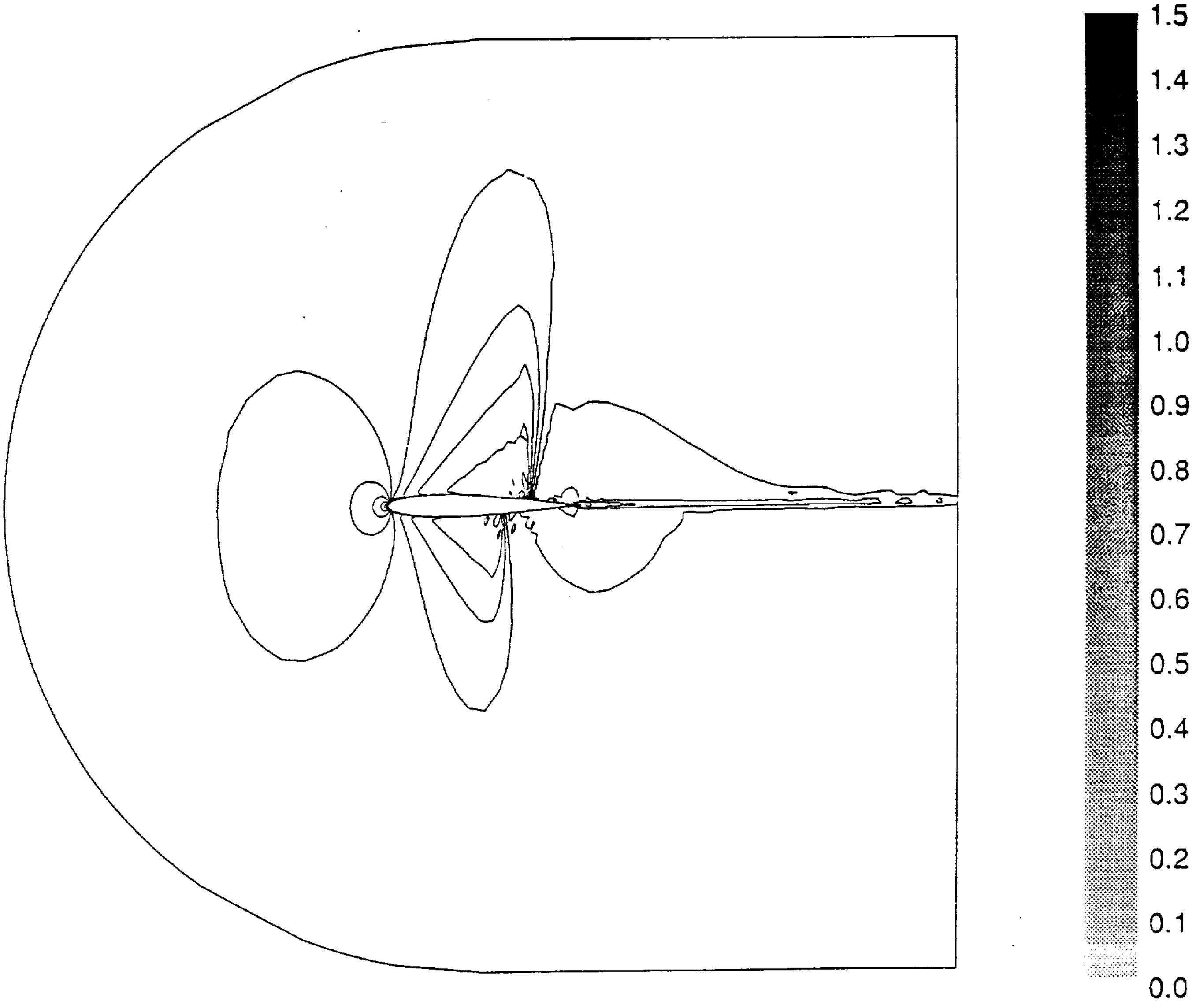


Figure 6: Mach number contours for the NACA 0012 airfoil - Mach number 0.85 and $\alpha = 1.0°$.

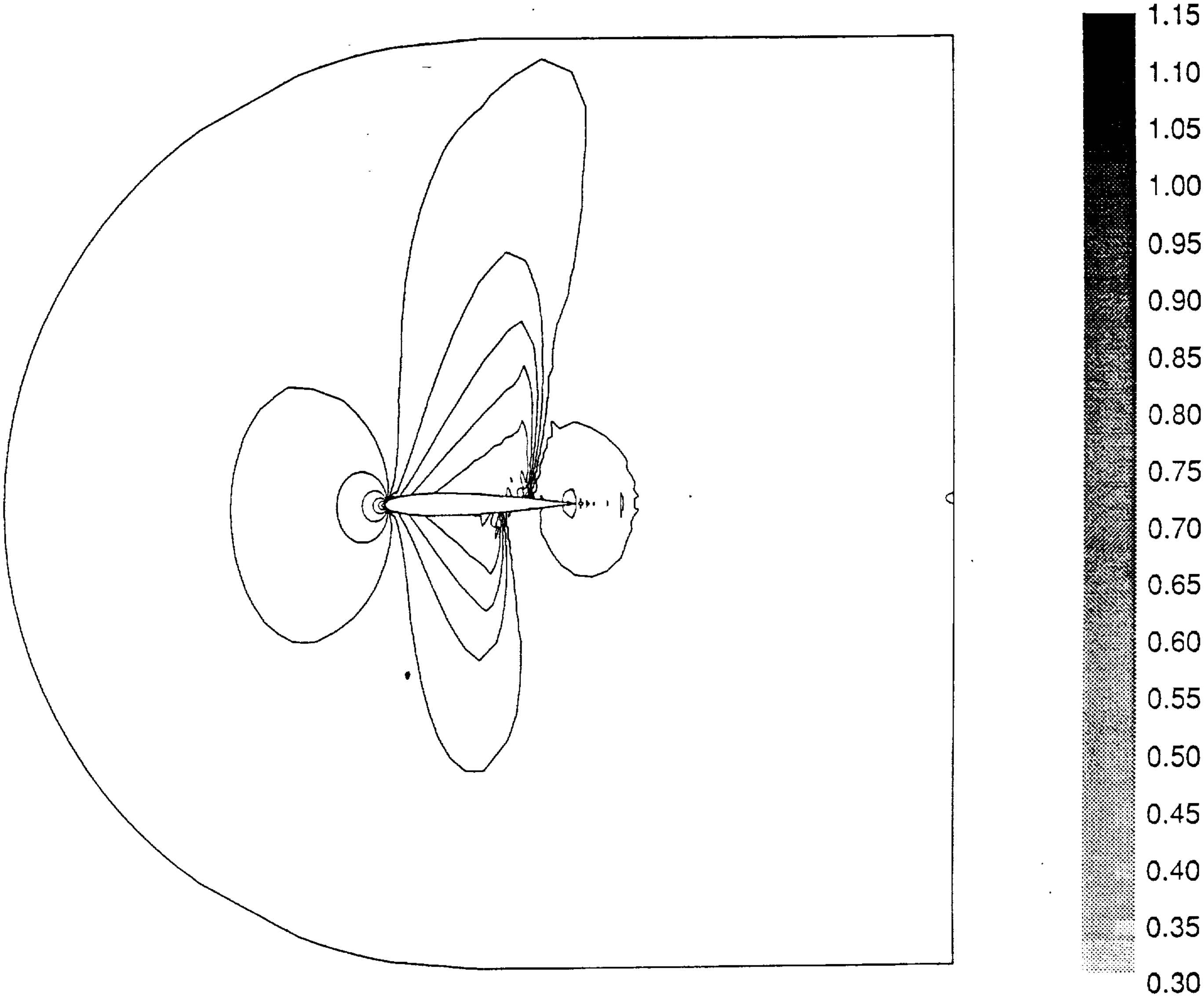


Figure 7: Pressure contours for NACA 0012 airfoil - Mach number 0.85 and $\alpha = 1.0°$.

4.2 Ni Bump - 10% arc

In this second example, compressible inviscid flow past a Ni Bump - 10% arc (See figure 8) was analyzed. In this example we have a single open flow boundary on the left, and right faces, and a solid wall boundary on the Ni bump and upper surface. As this is a two-dimensional problem, we have generated a simple mesh consisting of 40 quadratic hexahedral elements with no clustering. Inflow conditions for this example problem correspond to the choked flow case with Mach number = 0.73.

Some numerical results taken after 669 time steps are given in Figs. 9-11. Shown in Fig. 10 are contours of the Mach number. The location and the strength of shock compare very favorably with results by others. A carpet plot of Mach number along both the lower and upper surfaces is shown in Fig. 4, Again, the maximum value compares qualitatively with published reports.

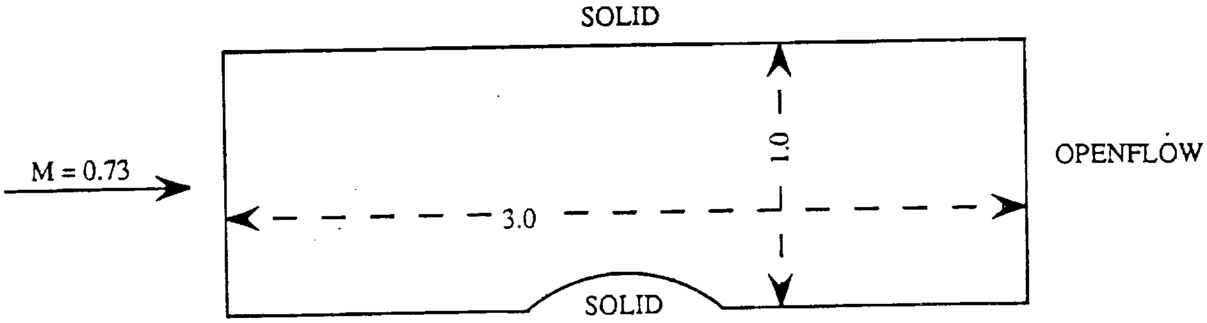


Figure 8: Schematic of supersonic flow over a Ni bump (10% arc).

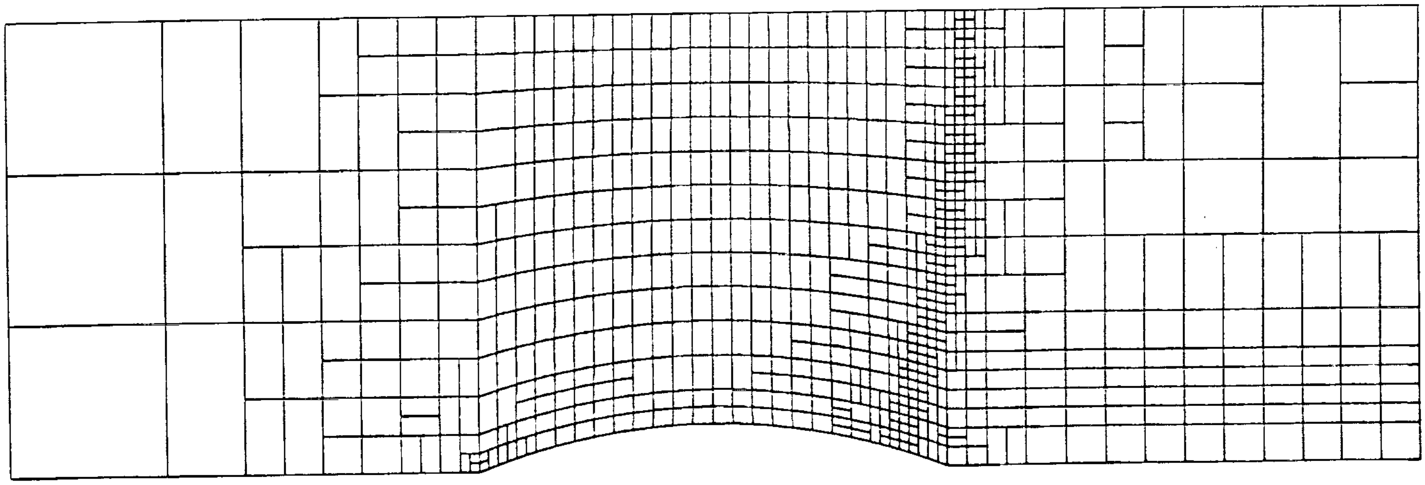


Figure 9: Final mesh for Ni bump (10% arc) – choked flow conditions.

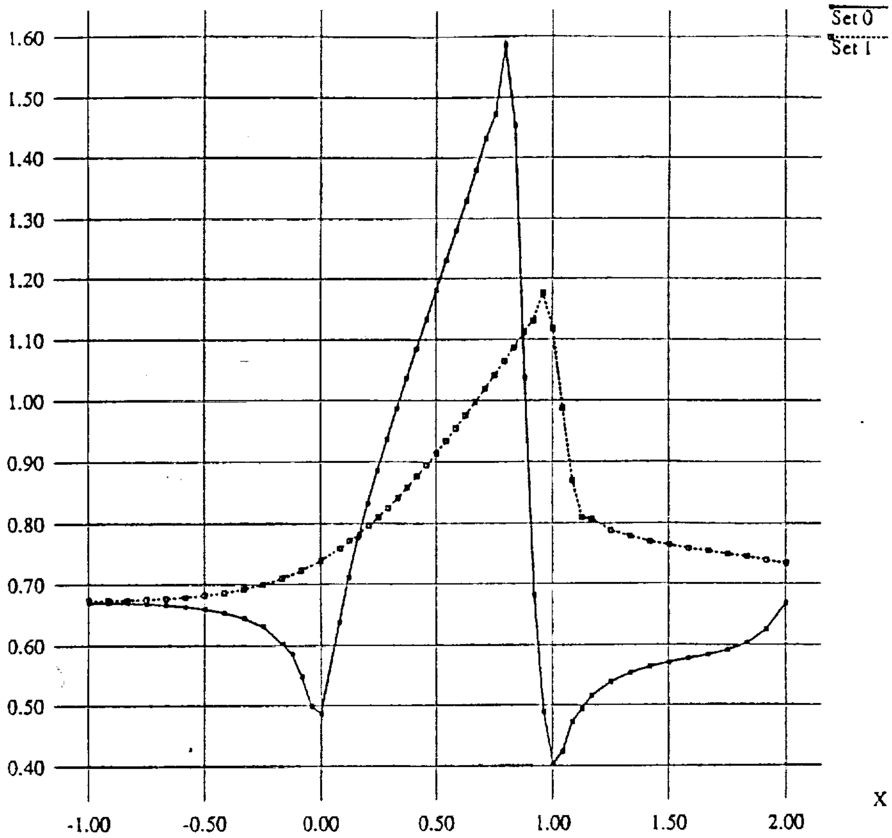
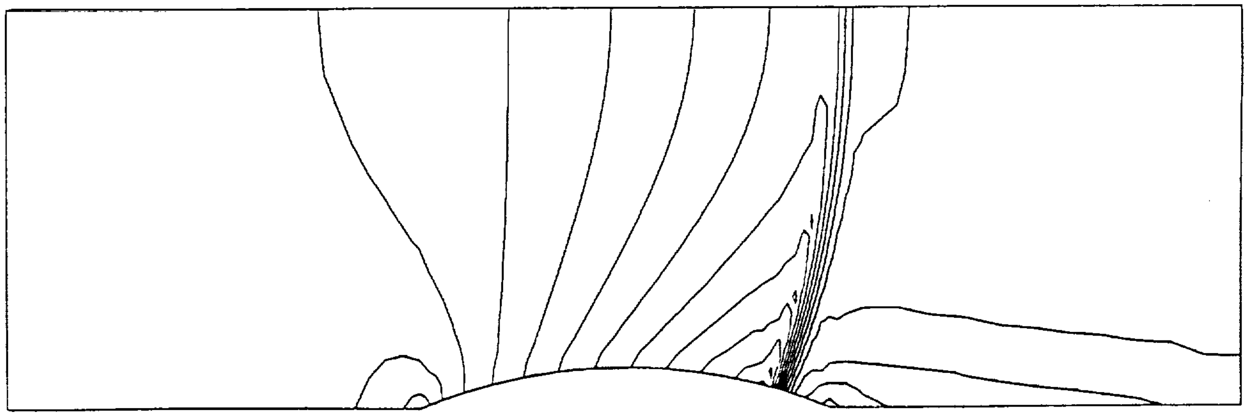


Figure 10: Mach distribution along lower and upper surface.



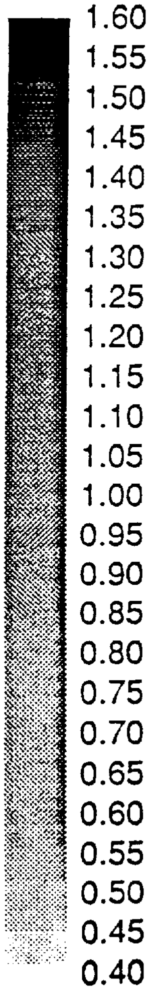


Figure 11: Mach number contours predicted by ROTOR3D.

4.3 Fixed 3D Wing

A fixed 3D wing with wing span 8 similar to the configuration employed in McAllister and Takahashi experiment [1] was chosen as the third case. Subsonic flow conditions with Mach number 0.85 were imposed with a zero angle of attack.

The initial mesh was constructed using both unstructured and structured techniques. The initial mesh shown in Fig. 12 consisted of 8035 nodal points and 7000 linear elements. Approximately 20 nodes were initially located on the wing surface at each span wise plane. The outer boundaries of the computational grid were located approximately 5 to 6 chord lengths above and below the wing. The gap between the wing tip and the side boundary was approximately 2 chord lengths. Open flow boundary conditions were imposed on all boundaries except the wing surface and the wing root plane. For those boundaries no flow and symmetry conditions were enforced respectively.

The initial mesh was adapted on 3 occasions during the solution process. During each adaptive pass the refinement/unrefinement parameters were prescribed as 0.90 and 0.30. The primary reason for the high refinement parameter was to control/minimize the number of elements refined. Two additional parameters – the maximum h level and the number of passes per adaption were also set to 3 and 2 respectively.

A closeup view of the wings upper surface after the final adaptation is presented in Fig. 13. Here, the shock pattern is evident by the refined elements. Notice the turning of the shock near the tip section. This is believed due to 3D effects and thus weakening of shock near the tip section. The final mesh consisted of 19997 elements and 26923 degrees of freedom.

Results for the pressure coefficient and Mach number for the final mesh are presented in Figs. 14-17. Although there is an apparent overshoot near the shock. qualitively the results look quite good. The results also show a slight asymmetry in the solution near the traveling edge. This asymmetry did not occur until after the third adaptive pass, and is believed to be a result of our anisotropic refinement procedure.

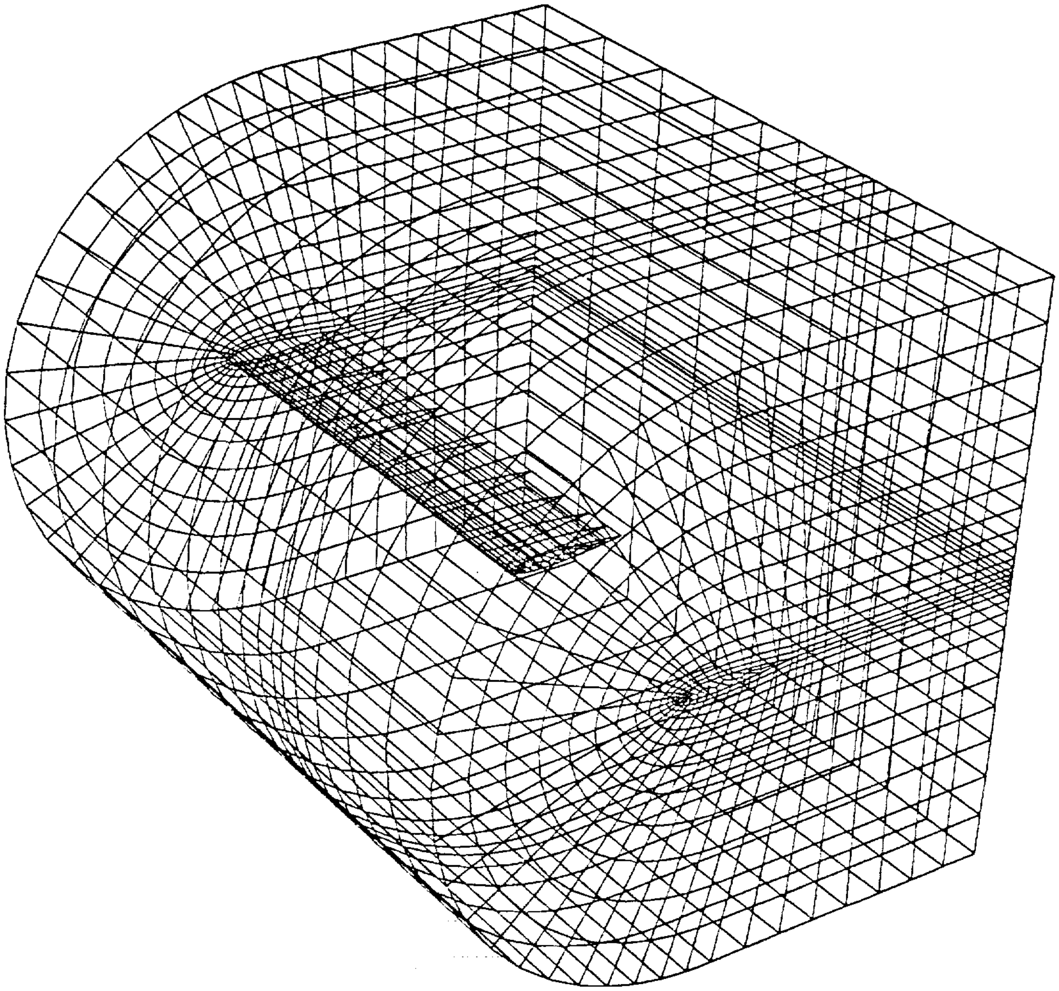


Figure 12: Initial mesh for the fixed 3D wing.

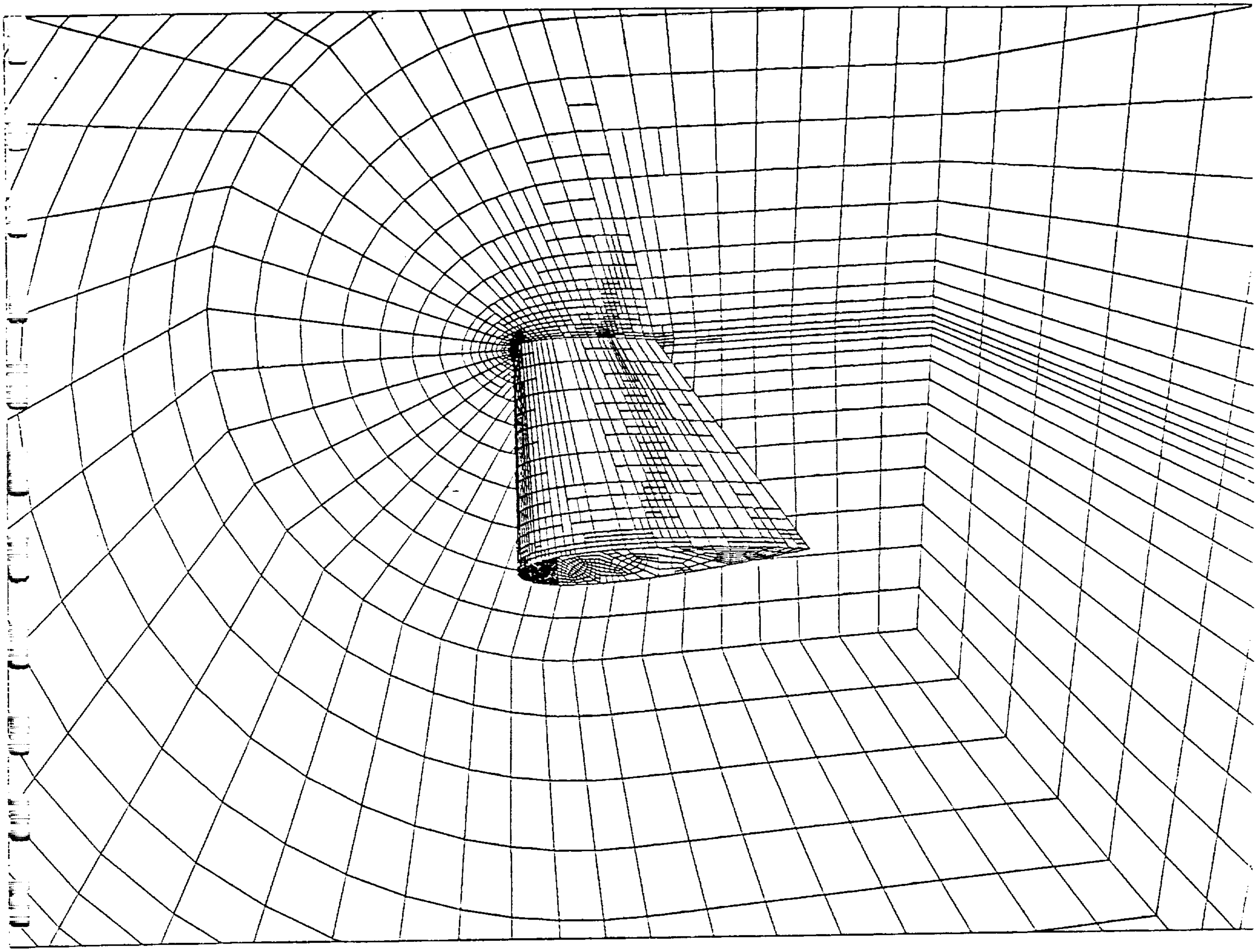


Figure 13: Closeup view of the wing upper surface highlighting the shock pattern as depicted by the adapted elements.

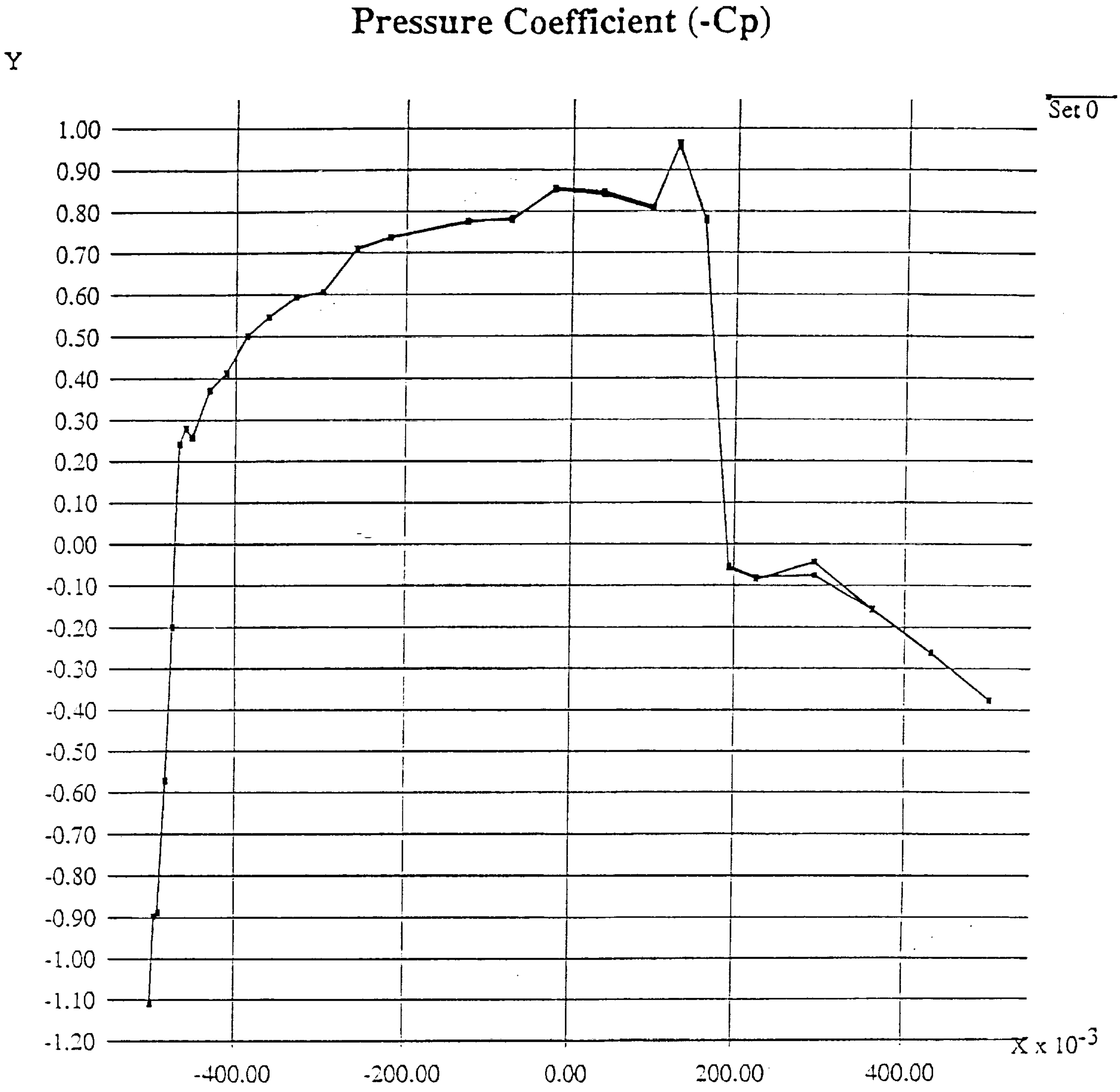


Figure 14: Predicted pressure coefficient (-Cp) versus streamwise distance—wing root section.

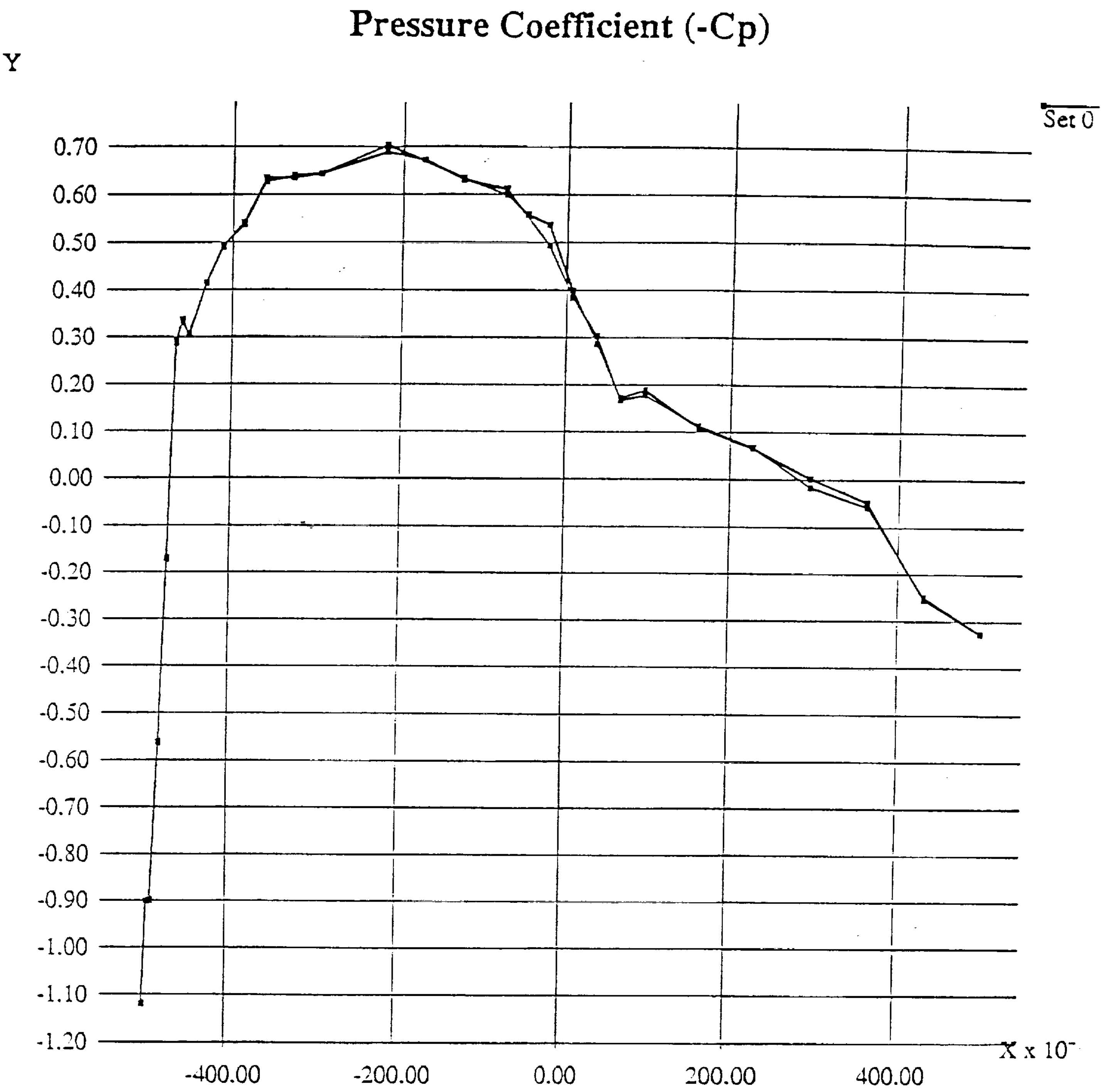


Figure 15: Predicted pressure coefficient (-Cp) versus streamwise distance--wing tip section.

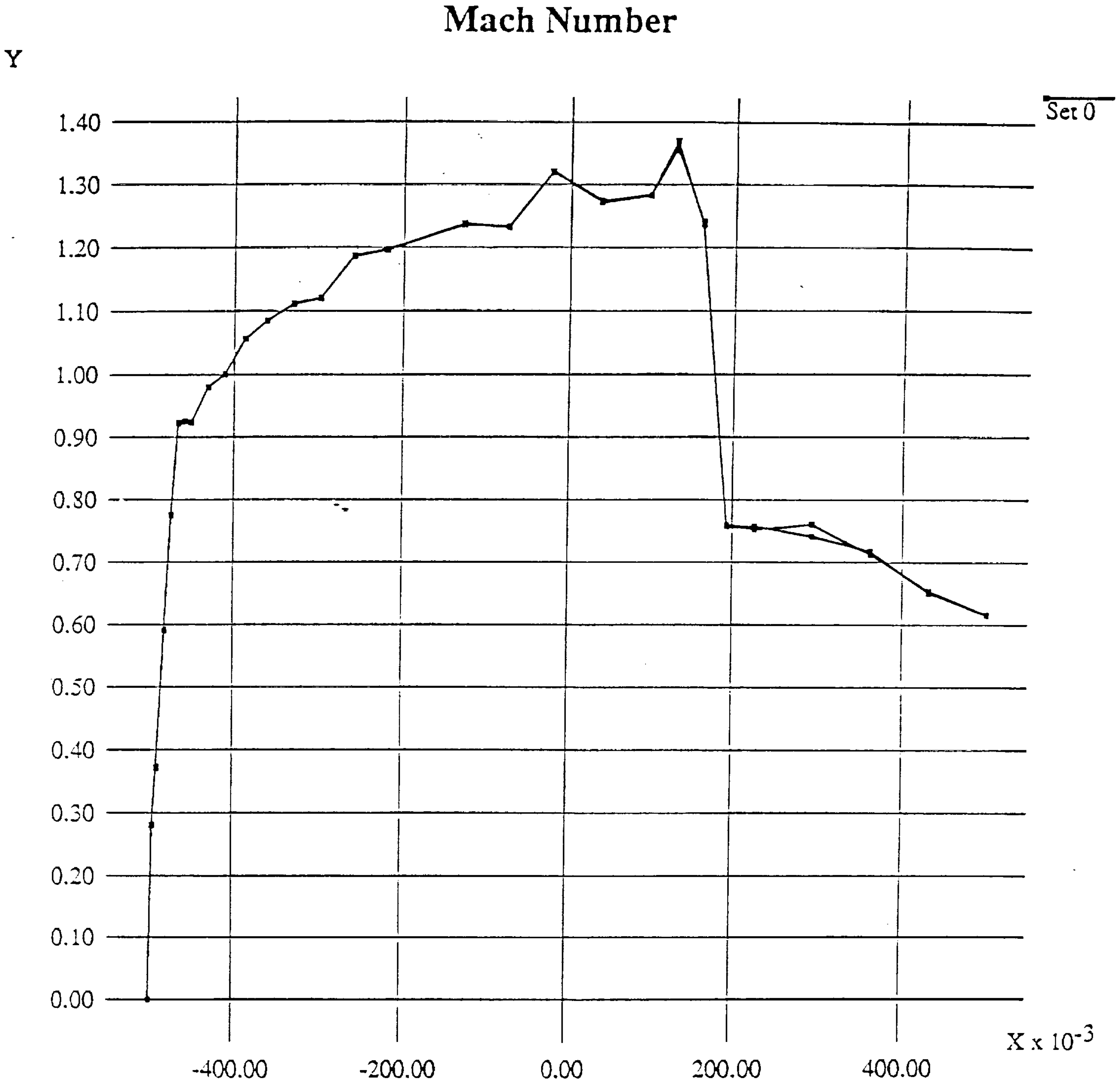


Figure 16: Predicted Mach number distribution—wing root section.

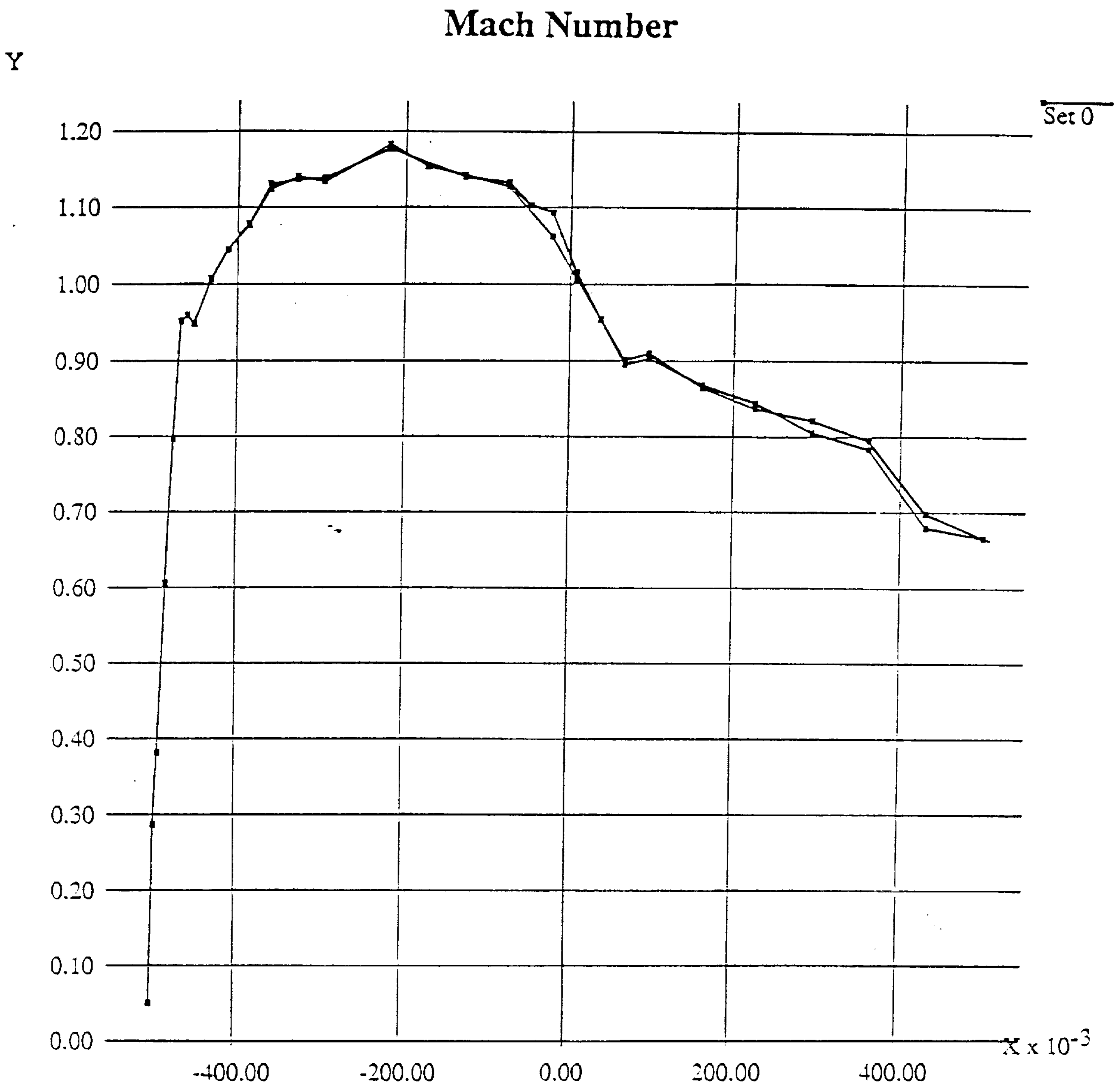


Figure 17: Predicted Mach number distribution—wing tip section.

4.4 Rotor Hover Simulation: Caradonna, Laub, and Tung Experiment [2]

The isolated model rotor experiment of Caradonna, Laub, and Tung [2] for a nonlifting test condition with tip Mach number = 0.439 and advance ratio ($\mu = 0.0$) was modeled. The rotor blade simulated is a rigid model rotor with aspect ratio of six (based on the blade radius) and a rectangular NACA 0012 airfoil section. The initial grid pictured in Fig. 18 consisted of 19698 nodes and 17184 linear elements.

A closeup view of the tip section is shown in Fig. 19. Note, in order to generate the mesh around the tip region and still maintain a given grid quality an unstructured was employed in the tip region.

Shown in Fig. 20 are isosurfaces of pressure for the rotor hover simulation after a total of 535 time steps.

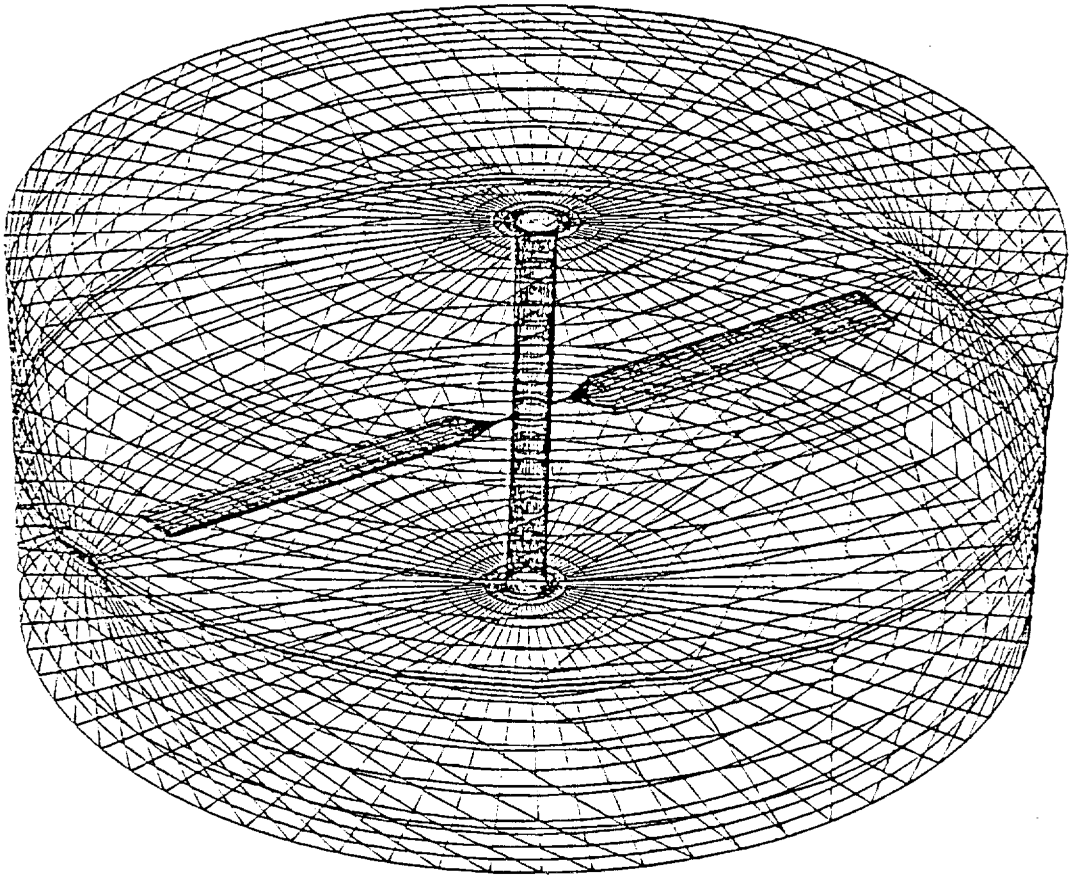


Figure 18: Computational grid for the rotor hover simulation. Iniital mesh consists of 17,184 elements and 19,698 nodes.

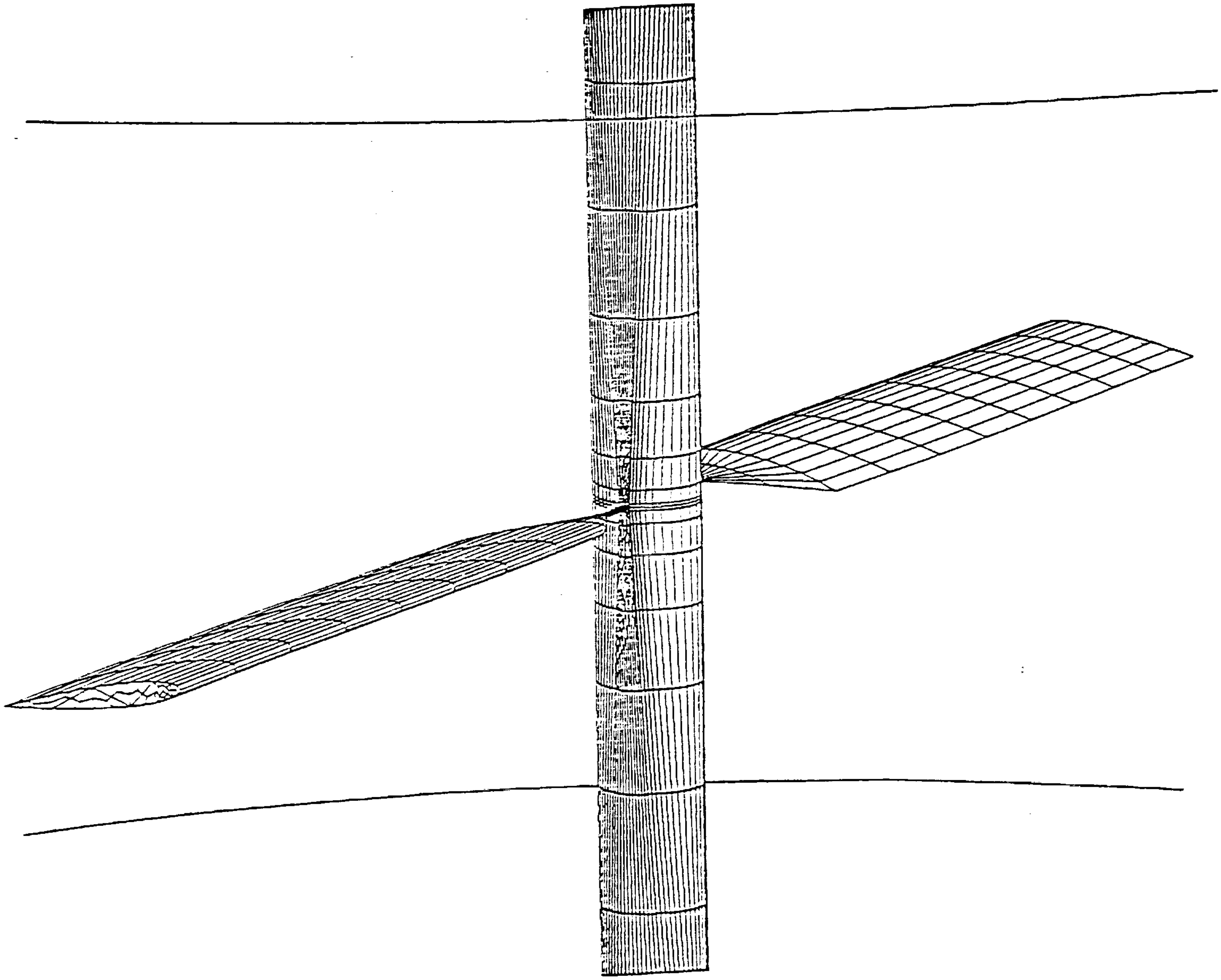


Figure 19: Close-up view of the rotor blades. Note, the blade tip employs an unstructured grid.

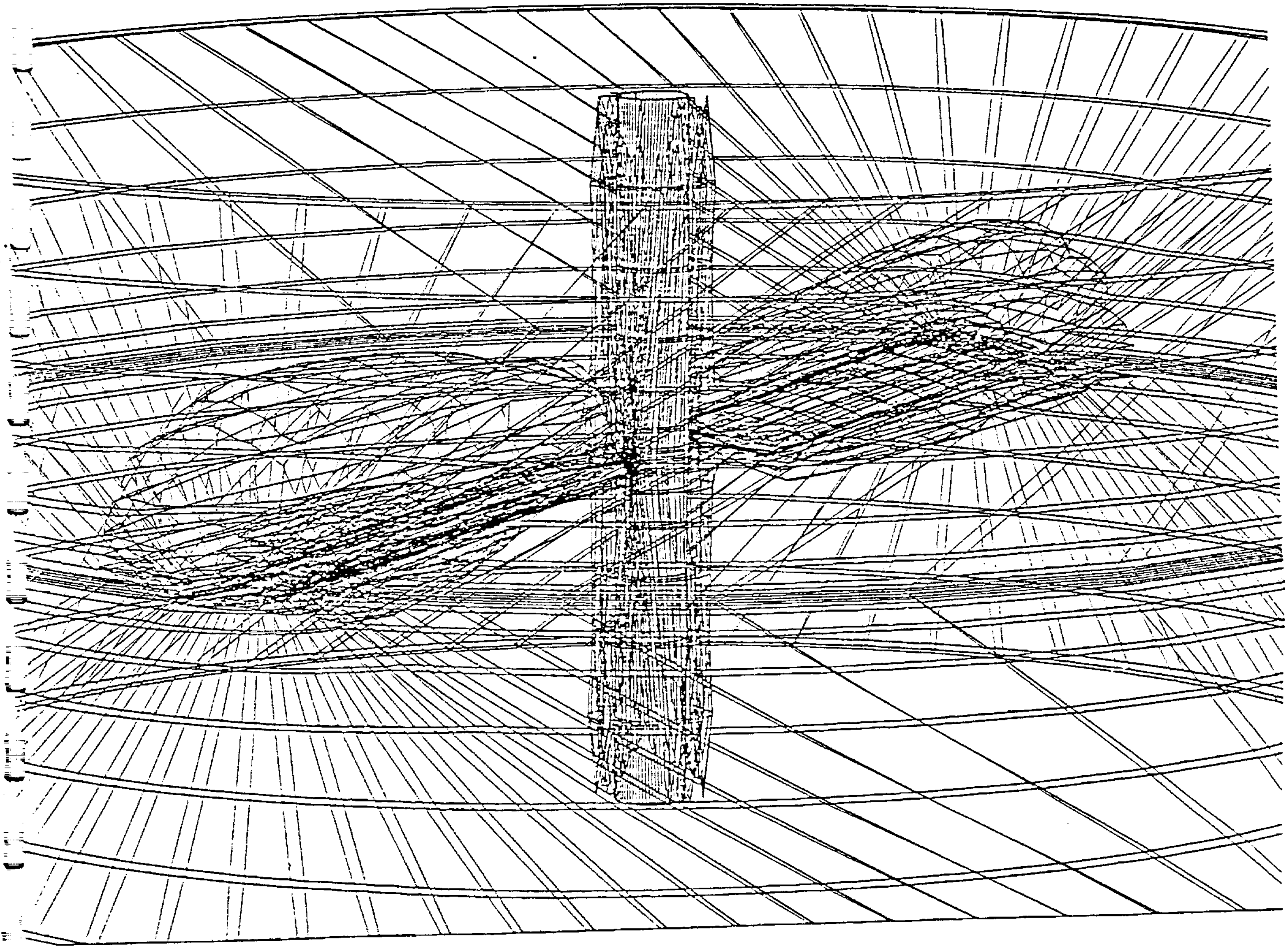


Figure 20: Isosurfaces plot of pressure for the rotor hover simulation. Total number of steps = 535.

4.5 Rotor-Fuselage Simulation: Smith and Betzina Experiment [3]

In this last problem, ROTOR3D is used to model the rotor-fuselage experiment investigated by Smith and Betzina [3]. The computational model for the rotor-fuselage simulation employs both unstructured/structured grid technology. The tip Mach number and advance ratio were specified as 0.6 and 0.15, respectively. The resulting angular velocity was 0.16 rad/sec.

The first step towards solving the complete Rotor-Fuselage problem targeted the sliding interface. To demonstrate the moving grid capability as well as the data structure, the rotor blades were allowed to rotate without actually calling the flow solver. Shown in Fig. 21 is the mesh with the blades in their initial location ($\Theta = 0$). The computational grid for this combined problem consists of approximately 30,000 elements. Fig. 22 shows the rotor blades after they rotated approximately 75° (1000 time steps).

The second step in the solution sequence required activating the flow solver. As discussed previously in Section 3, performance related issues prohibited any real computations even on this initial mesh.

4.6 References

1. McAlister, K.W., and Takahashi, R.K., "NACA 0015 Wing Pressure and Trailing Vortex Measurements," NASA TP 3151, 1991.

2. Caradonna, F.X., and Tung, C., "Experimental and Analytical Studies of a Model Helicopter Rotor in Hover," NASA TM 81232, Sept. 1981.

3. Strawn, R.C. and Barth, T.J., "A Finite-Volume Euler Solver for Computing Rotary Wing Aerodynamics on Unstructured Meshes," presented at the 48th Annual Forum of the American Helicopter Society, Washington, DC, June 3-5, 1992.

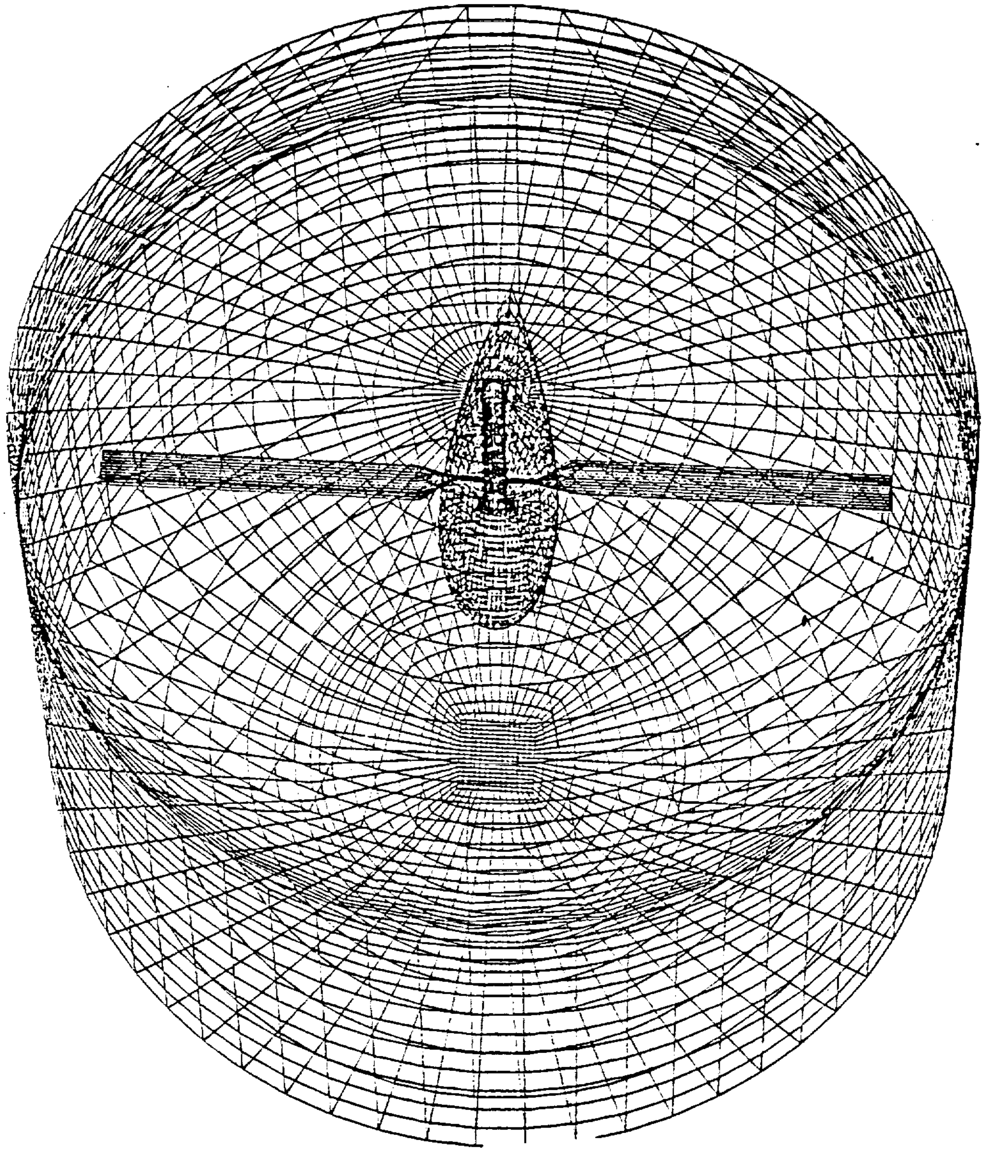


Figure 21: Computational grid for the rotor-fuselage simulation. Initial position, $\Theta = 0°$.

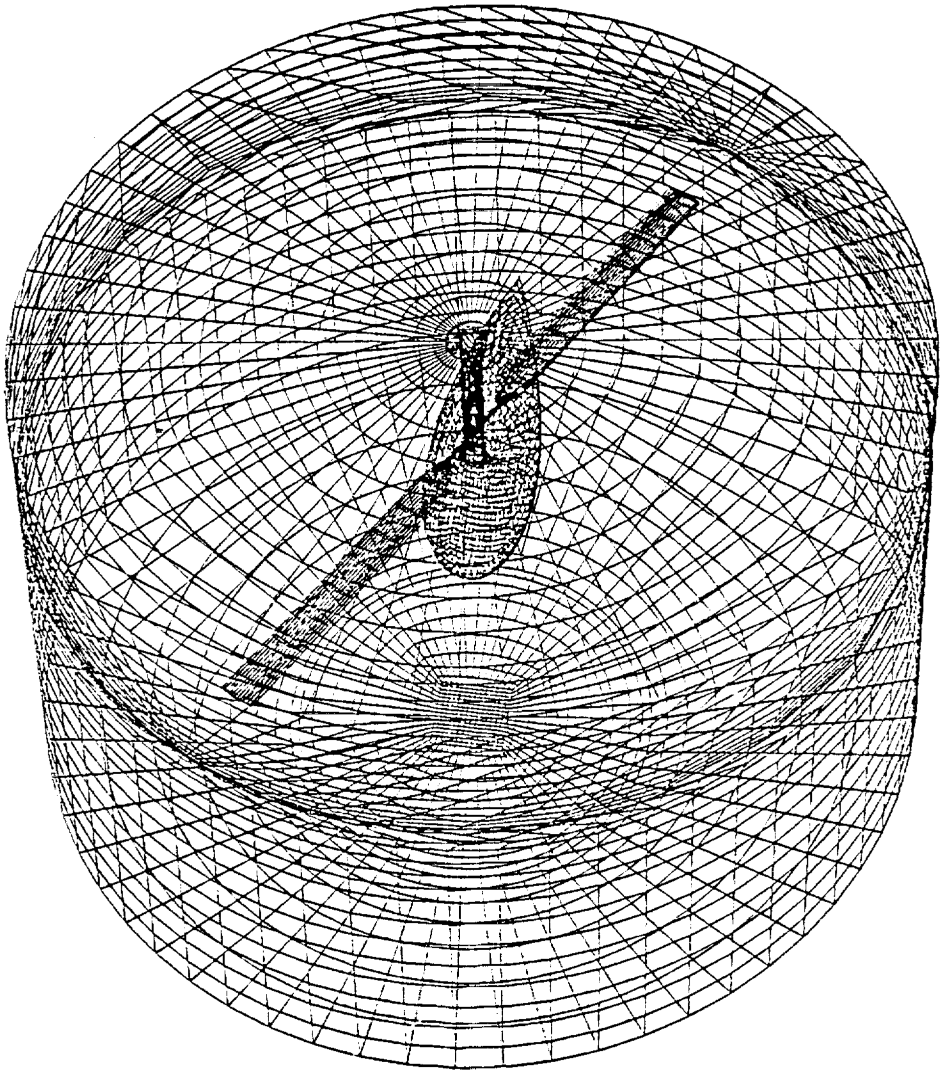


Figure 22: Rotor-fuselage grid with blades rotated approximately 75° (1000 steps).

5 Future Work

The main objectives of the Phase II project were to research and develop a set of computational methodologies which could be combined to effectively model transonic flow around multi-bladed helicopter rotors in hover and forward flight. The scope of this work included the development of data structures, adaptive methods, error estimator/indicator techniques, implicit-explicit and implicit/explicit time accurate and steady state flow solvers, iterative linear equation solvers, graphics, a user interface, and grid generation capabilities. All of these components were to be synthesized into an operational three dimensional code and used to solve predefined large scale benchmark problems.

At the conclusion of the project, many of these objectives were successfully met as outlined in Section 2 of this report. We were successful in solving several two-dimensional and three-dimensional test problems which included both static and moving grids. The structured/unstructured grid generation package, GAMMA3D, was completed early last summer and was used to generate several single and multi-blade meshes. In addition, this package was used to generate a rotor-fuselage mesh which we believe is the one of the first complete models of this type. We also completed the first version of the MESHVUR package which is a fully interactive package for viewing and conditioning computational meshes. Finally, the pieces of the effort which fell short in the final analysis we believe are all basically related to performance issues which were discussed in detail in Section 3.

Translating the performance issue into completion of the various tasks, obviously the benchmark problems identified in the original proposal were large scale computations and with the current performance of ROTOR3D could not be completed in a timely manner.

The current status of ROTOR3D is basically operational. It has been tested and validated on a number of rather small two and three-dimensional test problems. The immediate focus of any additional work on ROTOR3D should be targeted at improving its performance. The basic areas in which have been currently identified as requiring immediate attention include:

- Optimization of the boundary conditions part of the code (i.e. coloring and grouping bc to increase the vector length)
- Restructuring the access to the data base so the common blocks and global workspace is used to store scaler data used in many of the frequently accessed routines.

- Once these two pieces of optimization are in place, continue with profiling studies and in-lining as indicated by the YMP Flowtrace Statistics and other profiler options to improve overall vector length and performance.

NASA
National Aeronautics and
Space Administration

Report Documentation Page

1. Report No.
TR-93-02

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Advanced Adaptive Computational Methods for Navier-Stokes Simulations in Rotorcraft Aerodynamics

5. Report Date
March 4, 1993

6. Performing Organization Code

7. Author(s)

S. T. Stowers, J. M. Bass, J. T. Oden

8. Performing Organization Report No.

10. Work Unit No.

9. Performing Organization Name and Address

Computational Mechanics
7701 North Lamar, Suite 200
Austin, Texas 78752

11. Contract or Grant No.
NAS2-13285

13. Type of Report and Period Covered
Final Report

12. Sponsoring Agency Name and Address
NASA Ames Research Center
Moffett Field, CA 94035-1000

14. Sponsoring Agency Code

15. Supplementary Notes

None

16. Abstract

A Phase II research and development effort was conducted in area transonic, compressible, inviscid flows with an ultimate goal of numerically modeling complex flows inherent in advanced helicopter blade designs. The algorithms and methodologies therefore are classified as adaptive methods, which are error estimation techniques for approximating the local numerical error, and automatically refine or unrefine the mesh so as to deliver a given level of accuracy. The result is a scheme which attempts to produce the best possible results with the least number of grid points, degrees of freedom, and operations. These types of schemes automatically locate and resolve shocks, shear layers, and other flow details to an accuracy level specified by the user of the code. The Phase I work involved a feasibility study of *h*-adaptive methods for steady viscous flows, with emphasis on accurate simulation of vortex initiation, migration, and interaction. Phase II effort focused on extending these algorithms and methodologies to a three-dimensional topology.

17. Key Words (Suggested by Author(s))
Helicopter aerodynamics, computaional fluid dynamics, adaptive methods, unstructured meshes, finite elements

18. Distribution Statement
Unclassified, Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
None	None	33	n/a