AIAA

AIAA 2001-0080
Transition Of A Three-Dimensional Unsteady Viscous Flow Analysis From A Research Environment To The Design Environment

Suzanne M. Dorney, Daniel J. Dorney
NASA Marshall Space Flight Center
MSFC, AL

Frank Huber
Riverbend Design Services
Palm Beach Gardens, FL

David A. Sheffler
The Agilis Group
Palm Beach Gardens, FL

39th Aerospace Sciences Meeting & Exhibit
8-11 January 2001 / Reno, NV

TRANSITION OF A THREE-DIMENSIONAL UNSTEADY VISCOUS FLOW ANALYSIS FROM A RESEARCH ENVIRONMENT TO THE DESIGN ENVIRONMENT

Suzanne M. Dorney[*] Daniel J. Dorney[†]
NASA Marshall Space Flight Center
MSFC, AL, USA

Frank Huber[‡]
Riverbend Design Services
Palm Beach Gardens, FL, USA

David A. Sheffler[§]
The Agilis Group
Palm Beach Gardens, FL, USA

ABSTRACT

The advent of advanced computer architectures and parallel computing have led to a revolutionary change in the design process for turbomachinery components. Two- and three-dimensional steady-state computational flow procedures are now routinely used in the early stages of design. Unsteady flow analyses, however, are just beginning to be incorporated into design systems. This paper outlines the transition of a three-dimensional unsteady viscous flow analysis from the research environment into the design environment. The test case used to demonstrate the analysis is the full turbine system (high-pressure turbine, inter-turbine duct and low-pressure turbine) from an advanced turboprop engine.

NOMENCLATURE

M	- Mach number
P	- Static pressure
$P.S.$	- Pressure surface
$S.S.$	- Suction surface
Ω	- Rotation speed

SUBSCRIPTS

t	- Stagnation quantity
1	- HPT vane inlet quantity
6	- LPT rotor exit quantity

INTRODUCTION

The rapid advances in computing resources has made the inclusion of steady and unsteady three-dimensional flow analyses as an integral part of the design process a feasible proposition. One example of the integration of a three-dimensional steady flow procedure into a design system is given in Ref. [1], where the redesign of a high-pressure compressor was accomplished using numerical simulations. Another example is the development of a steady three-dimensional Navier-Stokes analysis for analyzing the counter-rotating blade configurations used in high-speed turboprop systems [2].

The transition of unsteady flow analyses into the design environment, however, requires the codes to be robust, efficient and easy to use. The goal of the current research program is to modify a three-dimensional unsteady Navier-Stokes analysis for use in preliminary and advanced stages of the design process.

[*] Computer Scientist
[†] Aerospace Engineer, Senior Member AIAA.
[‡] President
[§] Project Manager

Three different aspects have been considered in making the flow analysis suitable for design work:

1. Improving the speed of the analysis through the implementation of parallel processing protocols
2. Implementation of a graphical user interface (GUI)
3. Determination of the minimal computational grid density needed to resolve the pertinent flow physics and evaluate different designs

The analysis system described in this paper has been used in the design and analysis of a full turbine system (high-pressure turbine, inter-turbine duct and low-pressure turbine) for an advanced turboprop engine.

NUMERICAL ANALYSIS

The governing equations considered in this study are the time dependent, three-dimensional Reynolds-averaged Navier-Stokes equations. Both the full and thin-layer Navier-Stokes equations can be solved in the flow analysis. To extend the equations of motion to turbulent flows, an eddy viscosity formulation is used. The turbulent viscosity is calculated using a highly-modified derivative of the two-layer Baldwin-Lomax algebraic turbulence model [3].

The numerical algorithm used in the three-dimensional computational procedure consists of a time-marching, implicit, finite-difference scheme. The procedure is third-order spatially accurate and second-order temporally accurate. The inviscid fluxes are discretized according to the scheme developed by Roe [4]. The viscous fluxes are calculated using standard central differences. An approximate-factorization technique is used to compute the time rate changes in the primary variables. In addition, Newton sub-iterations are used at each global time step to increase stability and reduce linearization errors. For the case investigated in this study, one Newton sub-iteration was performed at each time step. Further details on the numerical procedure can be found in Dorney *et al.* [5, 6, 7]

The Navier-Stokes analysis uses O- and H-type zonal grids to discretize the flow field and facilitate relative motion of the airfoils. The O-grids are body-fitted to the surfaces of the airfoils and generated using an elliptic equation solution procedure. They are used to properly resolve the viscous flow in the blade passages and to easily apply the algebraic turbulence model. Algebraically generated H-grids are used to discretize the remainder of the flow field.

The code has been parallelized using the Message Passing Interface (MPI) library. In the current implementation, the solution can be decomposed by blade row, blade passage, or individual grid depending on the number of processors available.

GRAPHICAL USER INTERFACE

One of the primary components in the successful transition of a research code to the design environment is the implementation of a graphical user interface (GUI). An easy-to-use GUI will enable a new user to readily take advantage of an unfamiliar tool. In this particular implementation the GUI was designed to coordinate the two aspects of the computational analysis: the grid generation and the flow solution. The actual implementation of the GUI, and its advantages, are discussed in the following two subsections.

Implementation of the GUI

The GUI was implemented in JAVA 1.2 for several reasons. The primary reason is portability; the resulting system could be used on several different computer platforms without having to modify or even re-compile the code. The second reason is that JAVA includes extensive interface classes that allow for such GUI components as buttons, slide bars, and text boxex to be implemented easily. JAVA also provides for the event handling of the interface to be readily coordinated with the handling of the grid generator and flow solver.

The operation of the GUI focuses on the coordination of the following functions:

1. Modifying the input files for the grid generator and flow solver
2. Starting the grid generator and flow solver
3. Examining the output of the grid generator
4. Specifying and generating post-processed output from the flow solver
5. Providing a help feature which includes detailed definitions of the input values

The main control panel of the GUI is shown in Fig. 1.

The primary purpose of installing a GUI on an existing research code is to enable a less experienced user to easily use the utility. The interface can guide a new user through the steps of the process, as well as help to eliminate errors in using a new tool. A help facility is available that gives a detailed description of the purpose of each input value. In addition to the detailed help facility, when the cursor is held over an input window label for more than three seconds a brief description of the input value appears.

The first step in using the grid generator or the flow solver is to generate or modify the input files. In most cases an existing file is modified rather than starting from scratch. Since the input files can be complex the values have been separated into a set of logical groups, with each group appearing in a separate panel of the interface. The values in the current file will be displayed in the panel and the user can modify each value. One such panel used to modify the input file for the flow solver is shown in Fig. 2.

Once the modifications to the input file have been completed it is written out and the appropriate code, grid generator or flow solver, can be executed. This can be done directly from the GUI because JAVA allows for system commands to be executed in order to start another utility. The graphical screen output from the grid generator, is still sent to the screen and the user can interact with it. An example of this screen is shown in Fig. 3. Once this is completed the user is then given the option to examine the output file written by the grid generator.

A similar process can then be followed for modifying the flow solver input file and running the flow solver.

Advantages of the GUI

There are many advantages to having a GUI on a grid generator and flow solver. The first is that it makes each of the codes easier to use. As mentioned above, the input files are usually modified rather than written from scratch. Therefore, it is easier to change the selected values from within a text box than it is to manually edit a text file. This is especially true when transitioning a research code into the design environment because new users may not be familiar with the layout of the text file.

When using the grid generator or flow solver there are many different components to the input files. Using a GUI to display the respective files, the logically different sections can be displayed separately. For example, Fig. 4 shows how the multiple blade rows in the flow solver input file are displayed. Each of the blade rows are displayed as a separate *tab* on the interface panel.

Another benefit of using a GUI is the reduction of erroneous data within the input file. Each item in the input files has an allowable range of values. The GUI can be written so that only acceptable values can be entered into each field. and thus written to the input file.

Since the interface is directly connected to the grid generator and the flow code, once the input files have been modified the codes can be started. This eliminates the user from having to issue the start up commands manually, which can greatly increase the ease of use of the systems and eliminate errors. The output from the two utilities can also be directed to the screen, specified output files, or post-processing software when appropriate.

Another common cause of errors when these codes are run in the traditional manner is a failure to notice when informational messages are produced and written to a file. Some of these messages concern the status of the simulation. while some are error messages. To distinguish error messages from status messages a summary panel appears upon the completion of the code. One such panel is shown in Fig. 5. This panel summarizes the results of running the grid generator and allow the user the view one of the output files when errors have been detected.

TEST GEOMETRY AND GRID

The geometry under consideration is the complete turbine section of an advanced turboprop engine, including the high-pressure turbine (HPT), the inter-turbine duct (ITD) and the low-pressure turbine (LPT). The design Mach number at the inlet to the HPT is approximately $M_1 = 0.15$, and the ratio of the LPT exit static pressure to the HPT inlet total pressure is approximately $P_6/P_{t1} = 0.10$. The HPT rotates at $\Omega = 44,814$ RPM and the LPT rotates at $\Omega = 32,500$ RPM.

The actual turbine contains 23 vanes and 40 rotors in the HPT, while the LPT contains 17 vanes and 28 rotors. In the modeling of the turbine it was assumed that the HPT contained 28 vanes and 42 rotors, while it was assumed that the LPT contained 14 vanes and 28 rotors. Thus, the simulation was computed using a 2-vane/3-rotor/1-duct/1-vane/2-rotor approximation. The airfoils were scaled appropriately to keep the pitch-to-chord ratio constant. The computational grid contained 31 spanwise planes (with tip clearance regions modeled with 5 spanwise planes) and approximately 1.24 million grid points. Figure 6 illustrates the midspan section of the grids, while Fig. 7 shows an oblique rendering of the turbine geometry. The grid density requirements were determined by performing simulations for each individual turbine component in isolation (*i.e.*, the HPT, the ITD and the LPT).

The computational domain was decomposed by blade passage and the simulation was performed on 10 processors of a Silicon Graphics Inc., Origin2000 computer containing R10000 195 MHz processors. The simulation required 5.0×10^{-6} secs/grid point/iteration computation time and was run for one characteristic through-flow period (which equates to approximately one half-revolution of HPT rotor). Note, if 10 more processors had been available the domain could have been decomposed by O- and H-grids (the clearance grids are lumped in with the O-grids).

NUMERICAL RESULTS

Figures 8 and 9 illustrate the midspan unsteady pressure envelopes for the HP and LP turbines, respectively. The unsteadiness on the HPT vanes is confined to the suction surface downstream of the throat (see Fig. 8). The unsteadiness on the vanes is generated by potential interactions with the downstream rotors. A considerable amount of unsteadiness exists in the leading edge region on the suction surface of the HPT rotor, and is caused by potential interactions with the upstream vanes and periodic interaction with the vane wakes. Moderate amounts of unsteadiness are evident on the remainder of the suction surface and the pressure surface. The LPT vane exhibits little unsteadiness and does not indicate significant potential interaction with the downstream rotors (see Fig. 9). A moderate amount of unsteadiness is observed on the suction surface of the LPT rotors, and is mainly generated by periodically passing through the wakes of the upstream vanes.

Contours of the time-averaged midspan nondimensional entropy for the complete turbine are shown in Fig. 10. This figure highlights the convection of the airfoil wakes, and indicates that the wakes from the HPT rotors are completely mixed out before reaching the LPT vanes. Figure 11 illustrates time-averaged relative Mach contours at midspan of the turbine. Mach contours are helpful in identifying shocks, airfoil boundary layers, airfoil wakes and the potential fields associated with the airfoils.

Figures 12 and 13 show instantaneous entropy and relative Mach contours, respectively, at midspan of the HPT. The entropy contours (see Fig. 12) elucidate the convection of the vane wakes, and their stretching into a V-pattern in the rotor due to differences in the velocity between the suction and pressure sides of the passage. The variations in the contours from passage to passage underscore the need for accurately modeling the airfoil count ratios. The relative Mach contours shown in Fig. 13 appear discontinuous in between the vane and rotor passages due to the rotation of the rotor airfoils.

Figures 14 and 15 contain instantaneous entropy and relative Mach contours, respectively, at midspan of the LPT. The entropy contours indicate that the flow separation on the suction surface of the rotor is a function of its location with respect to the vane wakes.

Although some redesign occurred based on the results of the simulation, the performance parameters predicted by the analysis exhibited excellent agreement with the design intent.

CONCLUSIONS

A comprehensive unsteady three-dimensional flow analysis system has been developed and applied to the design of the turbine system used in an advanced turboprop engine. All aspects of the grid generation and flow solution process are handled using coordinated user interface panels. The computational domain can be decomposed by blade row, blade passage or individual grid to enable rapid solutions. The analysis system was used to compute an unsteady solution for a turbine system which included a high-pressure turbine, an inter-turbine duct and a low-pressure turbine. The predicted results exhibited excellent agreement with the design intent, although it also highlighted high-loss regions which were subsequently redesigned.

References

[1] LeJambre, C. R., Zacharias, R. M., Biederman, B. P., Gleixner, A. J. and Yetka, C. J., "Development and Application of a Multistage Navier-Stokes Solver: Part II - Application to a High-Pressure Compressor Design," *ASME Journal of Turbomachinery*, Vol. 120, April, 1998, pp. 215-223.

[2] Hall, E. J., Topp, D. A., Heidegger, N. J., and Delaney, R. A., "Investigation of Advanced Counterrotation Blade Configuration Concepts for High Speed Turboprop Systems: Task 8 Cooling Flow/Heat Transfer Analysis User's Manual" NASA CR-195360, September, 1994.

[3] Baldwin, B. S., and Lomax, H.,"Thin Layer Approximation and Algebraic Model for Separated Turbulent Flow," AIAA Paper 78-257, Huntsville, AL, January, 1978.

[4] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357-372.

[5] Dorney, D. J., Davis, R. L., Edwards, D. E., and Madavan, N. K., "Unsteady Analysis of Hot Streak Migration in a Turbine Stage," *AIAA Journal of Propulsion and Power*, Vol. 8, No. 2, 1992, pp. 520-529.

[6] Dorney, D. J., and Davis, R. L., "Numerical Simulation of Turbine 'Hot Spot' Alleviation Using Film Cooling," *AIAA Journal of Propulsion and Power*, Vol. 9, No. 3, 1993, pp. 329-336.

[7] Dorney, D. J., and Schwab, J. R., "Unsteady Numerical Simulations of Radial Temperature Profile Redistribution in a Single-Stage Turbine," *ASME Journal of Turbomachinery*, Vol. 118, No. 4, October, 1996, pp. 783-791.

Figure 1: Main control panel.

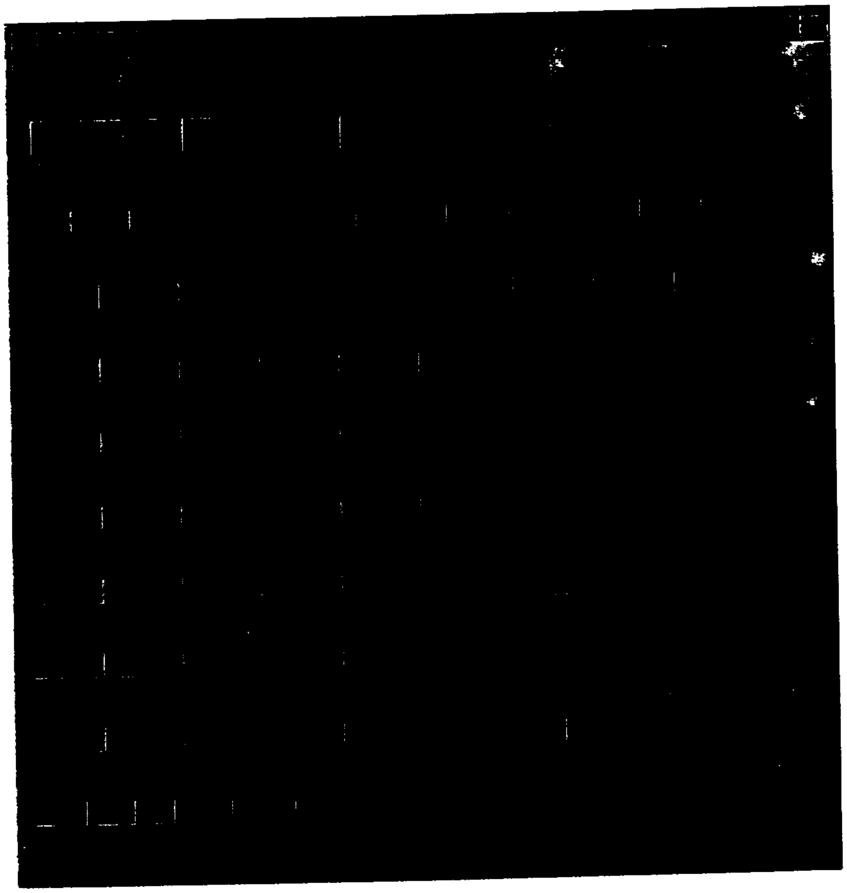

Figure 2: Flow solver input file modification panel.

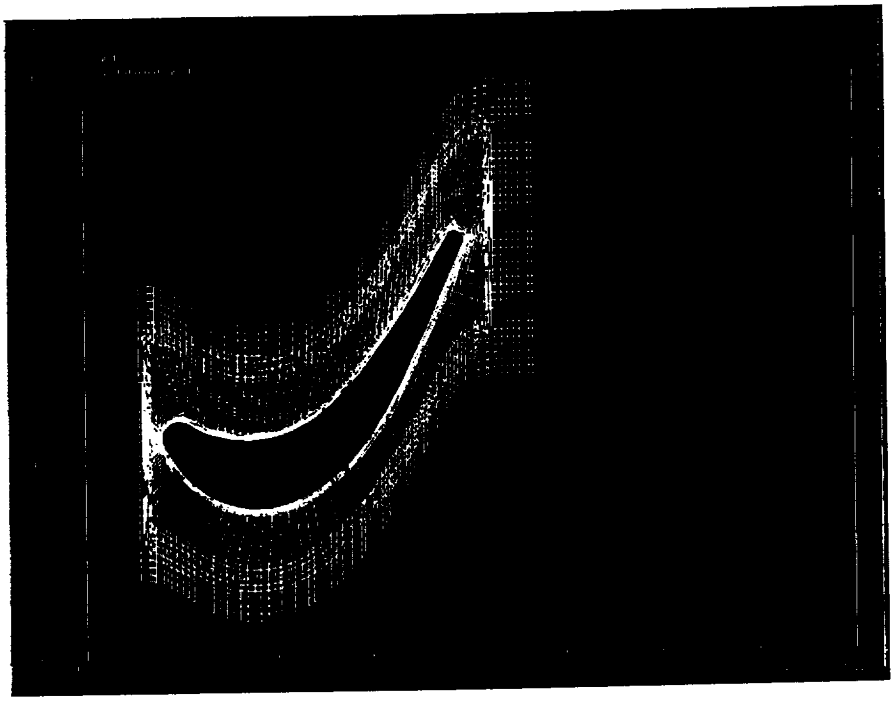

Figure 3: Interactive graphical output from the grid generator.

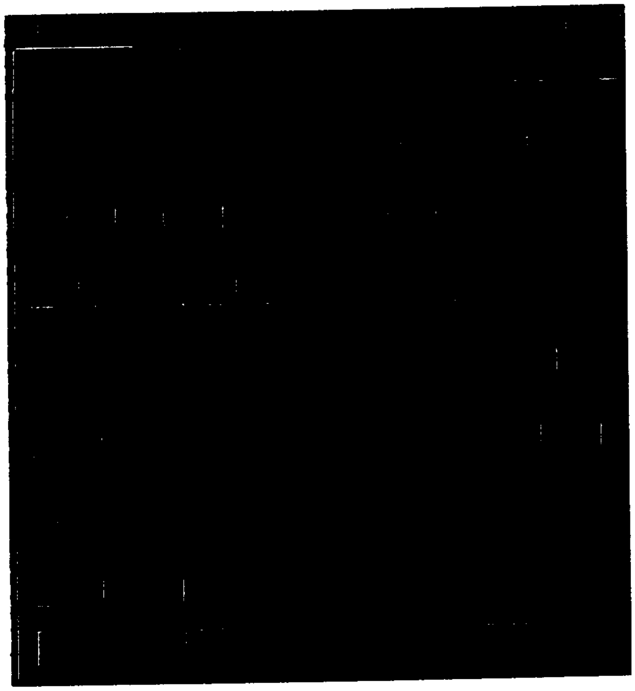

Figure 4: Information for multiple blade rows in the flow solver input file.

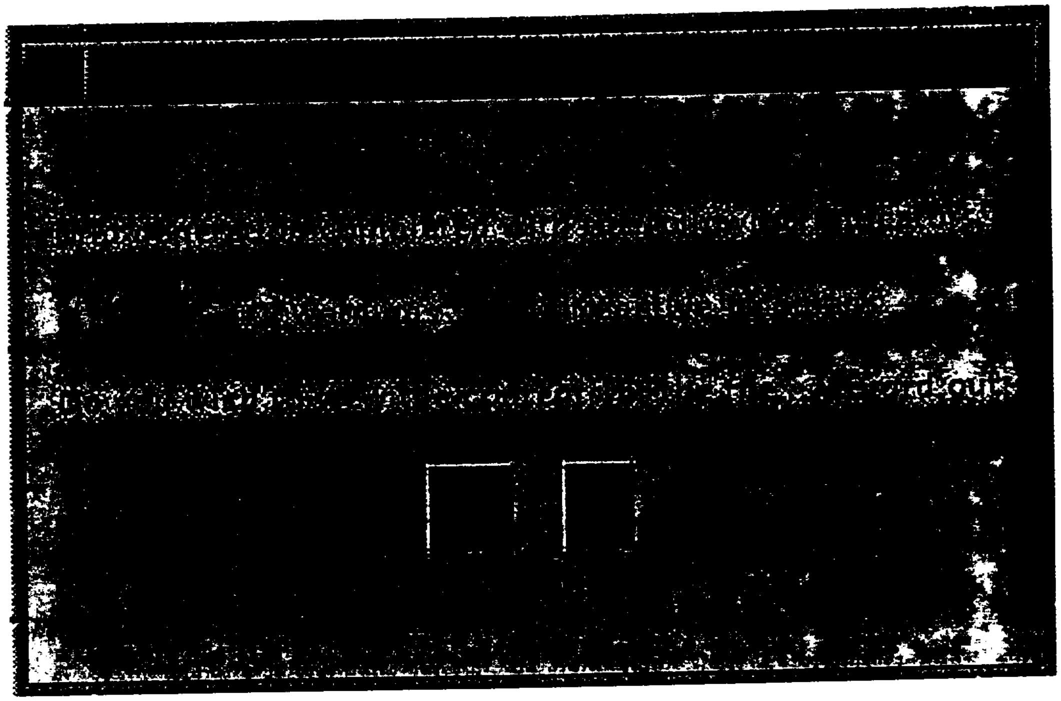

Figure 5: Information on the status of the grid generator.

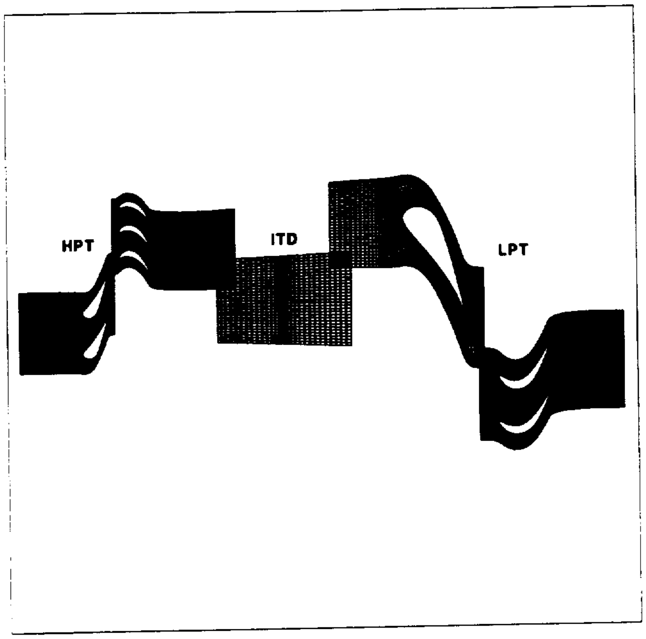

Figure 6: Computational grid for the full turbine system - view 1.

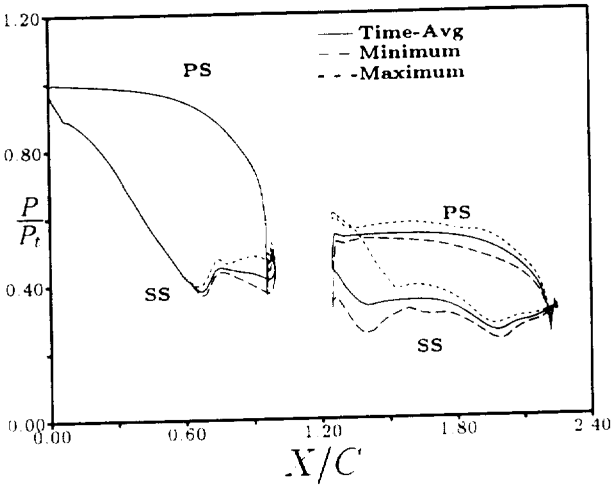

Figure 8: Unsteady pressure envelope - HPT - 50.0% span.

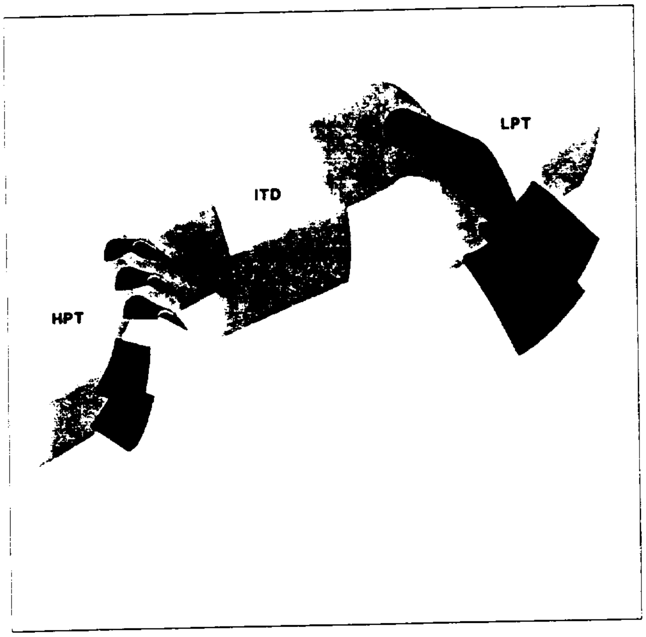

Figure 7: Computational grid for the full turbine system - view 2.

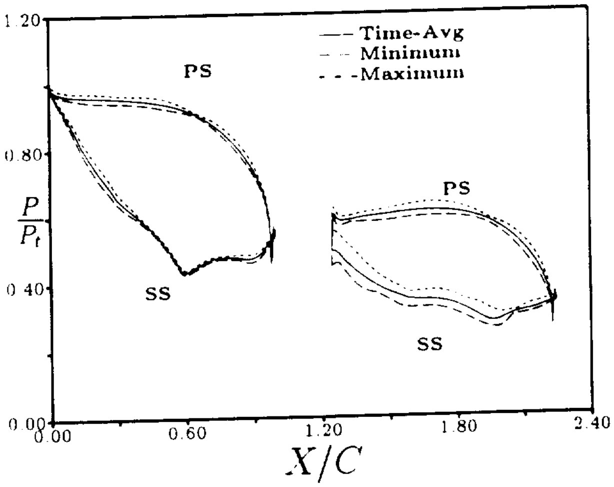

Figure 9: Unsteady pressure envelope - LPT - 50.0% span.

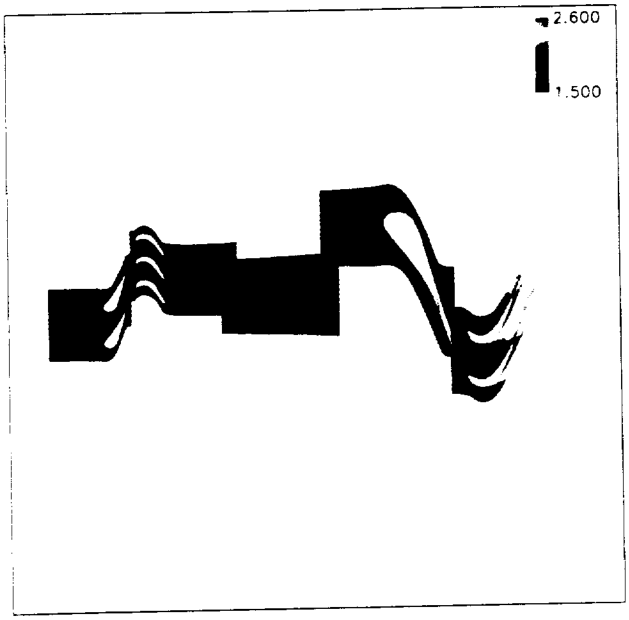

Figure 10: Time-averaged entropy - full turbine - 50.0% span.

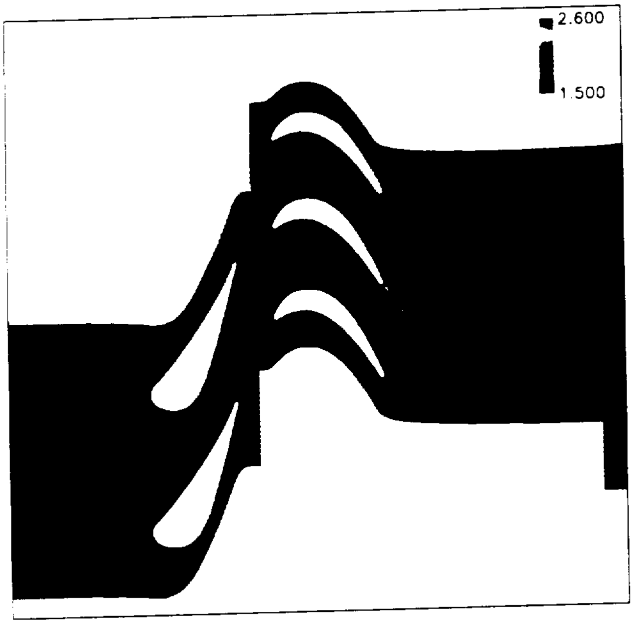

Figure 12: Instantaneous entropy - HPT - 50.0% span.

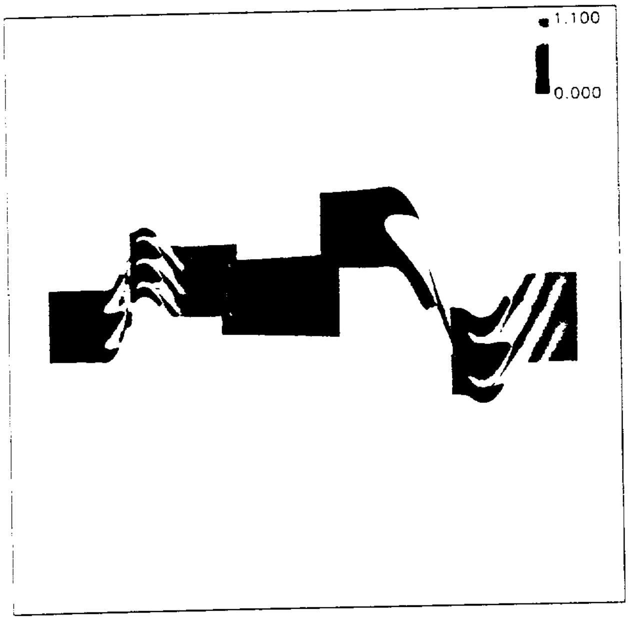

Figure 11: Time-averaged relative Mach number - full turbine - 50.0% span.

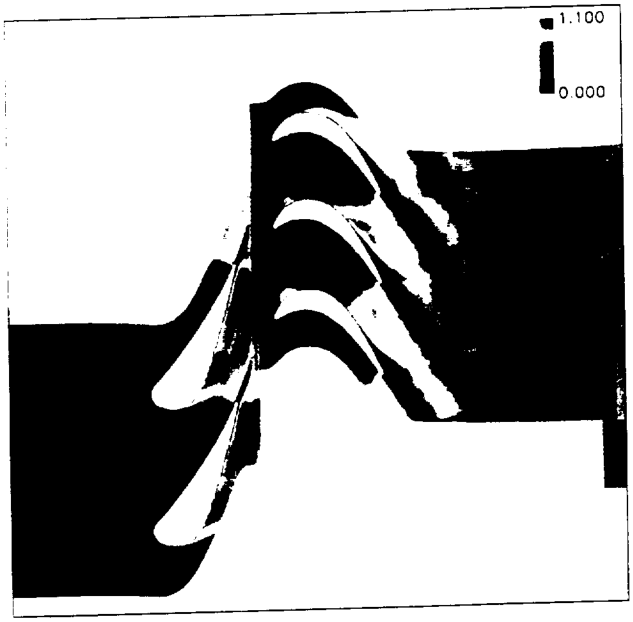

Figure 13: Instantaneous relative Mach number - HPT - 50.0% span.

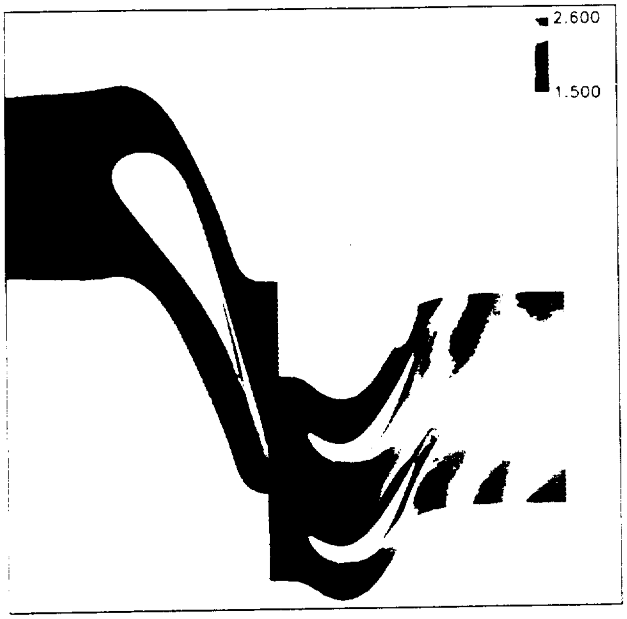

Figure 14: Instantaneous entropy - LPT - 50.0% span.

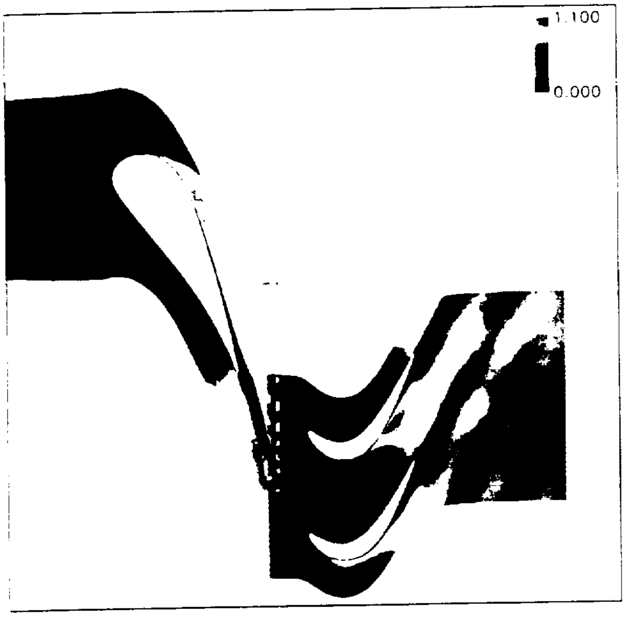

Figure 15: Instantaneous (relative frame) Mach number - LPT - 50.0% span.