2004

NASA FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER

THE UNIVERSITY OF ALABAMA
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE
ALABAMA A&M UNIVERSITY

CONVERGENCE ACCELERATION AND DOCUMENTATION OF CFD CODES FOR TURBOMACHINERY APPLICATIONS

Prepared By:	Jed E. Marquart, Ph.D., P.E.
Academic Rank:	Professor
Institution and Department:	Ohio Northern University Mechanical Engineering Department
NASA/MSFC Directorate:	Space Transportation
MSFC Colleague:	Dr. Daniel Dorney

Introduction

The development and analysis of turbomachinery components for industrial and aerospace applications has been greatly enhanced in recent years through the advent of computational fluid dynamics (CFD) codes and techniques. Although the use of this technology has greatly reduced the time required to perform analysis and design, there still remains much room for improvement in the process. In particular, there is a steep learning curve associated with most turbomachinery CFD codes, and the computation times need to be reduced in order to facilitate their integration into standard work processes.

Two turbomachinery codes have recently been developed by Dr. Daniel Dorney (MSFC) and Dr. Douglas Sondak (Boston University) [1]. These codes are entitled Aardvark (for 2-D and quasi 3-D simulations) and Phantom (for 3-D simulations). The codes utilize the General Equation Set (GES), structured grid methodology, and overset O- and H-grids. The codes have been used with success by Drs. Dorney and Sondak, as well as others within the turbomachinery community, to analyze engine components and other geometries.

One of the primary objectives of this study was to establish a set of parametric input values which will enhance convergence rates for steady state simulations, as well as reduce the runtime required for unsteady cases. The goal is to reduce the turnaround time for CFD simulations, thus permitting more design parametrics to be run within a given time period. In addition, other code enhancements to reduce runtimes were investigated and implemented.

The other primary goal of the study was to develop enhanced users manuals for Aardvark and Phantom. These manuals are intended to answer most questions for new users, as well as provide valuable detailed information for the experienced user. The existence of detailed user's manuals will enable new users to become proficient with the codes, as well as reducing the dependency of new users on the code authors. In order to achieve the objectives listed, the following tasks were accomplished:

- Parametric Study Of Preconditioning Parameters And Other Code Inputs
- Code Modifications To Reduce Runtimes
- Investigation Of Compiler Options To Reduce Code Runtime
- Development/Enhancement of Users Manuals for Aardvark and Phantom

Parametric Study Of Preconditioning Parameters And Other Code Inputs

The Aardvark and Phantom codes both incorporate a form of 'preconditioning' to enhance the convergence of steady state solutions, as well as to speed the analysis of unsteady solutions, for low speed and incompressible flow problems [2]. This methodology, although very useful and effective, is not an 'exact science', in that no one set of parameter values will result in the lowest required CPU time for all flow cases. Therefore, it is essential that various preconditioning parameter values be examined to determine their effect on the solution time for certain typical flow/geometry cases. In addition, the values and effects of some parameters that are not directly associated with preconditioning were examined. In particular, the following parameters were examined:

- ivp_parm – used for selection of velocity for the preconditioning velocity term

- lpmax – number of Newton iterations to be run at each timestep
- upw – order of accuracy of inviscid fluxes
- ifreeze_prop – whether or not thermodynamic property values should be 'frozen' at inlet conditions

The values of these parameters were examined in terms of their effects on convergence time required and also accuracy of the solution. An example of one of the sample cases, illustrating pressure contours, is shown in Figure 1.

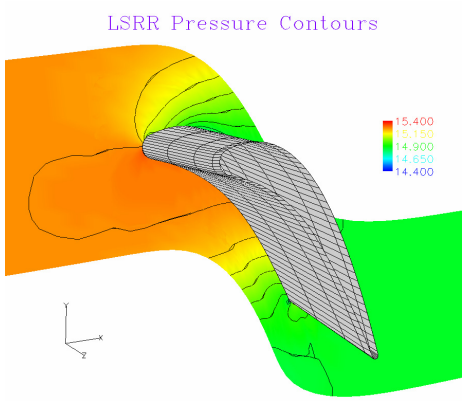

Figure 1: Pressure Contours for Sample Case

From the study, it was determined that the value of the preconditioning velocity parameter (ivp_parm) had very little effect on the number of iterations required for convergence. In a similar fashion, the number of inner iterations had negligible effect on the convergence rate, but fewer outer iterations could be run if more inner iterations were used. The order of the inviscid fluxes had very little effect, other than the fact that higher order should be used if vortex shedding studies are performed. Freezing the values of thermodynamic properties, however, did have a positive (reduction) effect on the runtime required, with negligible effect on the results for essentially isothermal cases.

Code Modifications to Reduce Runtimes

Since the overall goal of this effort was to reduce the runtimes required for flow solutions, it was also obvious that the codes should be examined to determine if there might be some methods to speed the solutions by recoding or 'streamlining' portions of the codes. It was determined that this was indeed possible. The following changes were made to the codes:

- matrix multiplications were 'unrolled', and calls to matrix multiplication routines eliminated
- the ability to 'turn off' the energy equation for isothermal cases was created
- 2nd order NIST thermodynamic property value determination was installed
- superfluous calls to thermodynamic property value routines were eliminated

Through the restructuring of the codes as listed above, improvements in runtime speeds of 6% to 21% over the baseline code were achieved for Aardvark, and 13% to over 50% for Phantom. The precise amount of improvement depended upon the configuration being run, as well as the

platform on which it was run. For both codes, the decrease in required runtime was dramatic, and a welcome occurrence.

Investigation Of Compiler Options To Reduce Code Runtime

In addition to code restructuring and streamlining, another method used to increase the speed of the codes was to investigate the effects of various compiler options when creating the executable code. Each platform has its own set of compiler options available, with a corresponding anticipated increase in code speed, and thus reduced required runtimes.

Specifically, the platforms on which the compiler options were evaluated included SGI and Sun workstations, with both 2-D and 3-D test cases. It was interesting, and pleasing, to find that, simply by altering the compilation options, a speed increase of over 2% to over 8% was achieved. It was determined that the "-ipa –Ofast" (or its equivalent) compiler option produced the best results in most cases.

Development/Enhancement of Users Manuals for Aardvark and Phantom

Prior to this effort, basic manuals did exist for both the Aardvark and Phantom codes. These documents consisted primarily of copies of variable descriptions, variable default values, some input/output descriptions, and a basic procedure for running the codes. The variable information was taken directly from the codes.

Although providing a start for the user, the documents left something to be desired as far as getting new users up and running quickly and with few or no questions. In particular, there were no sections addressing grid generation and post-processing within the document, although basic documentation did exist separately for these codes. Since the codes which accomplish grid generation and post-processing are not commonly used in industry, the need for their documentation to be incorporated into the Aardvark and Phantom users manuals became apparent. By including documentation for all packages necessary to create a grid, obtain a flow solution, and post-process the solution within one manual, the package becomes 'complete'. Users have one source of information necessary for the entire package, and the need to contact the code developers for questions is reduced.

In order to make the Aardvark and Phantom users manuals complete and self-sufficient, the following sections were added and/or enhanced:

- Licensing information
- Files required by the codes – including file types and descriptions/uses
- Grid generation – including input/output variables descriptions, hints for use, and a sample case
- Input parameter sets – including full variables descriptions, types, default values, etc.
- Parameter definitions – definition of basic code parameters
- Output description – including output file variables descriptions
- Post-processing – including output variables descriptions and basic use of the post-processing code

Upon completion of the first revision of the manuals, they were distributed to existing users for comments. Response was favorable, and returned comments were incorporated to develop more refined manuals. The manuals will continue to evolve and be updated as the codes change or as it becomes apparent that additional information would be beneficial to the users.

Future Applications

Code development/enhancement is, of course, an ongoing task. Methodology and coding improvements to increase the speed and accuracy of the CFD codes are anticipated. With the improvements made as a part of this study, the turbomachinery community can make tremendous use of these CFD codes in their present form for analysis purposes. As new methods and coding structures emerge, the time required to perform accurate and timely analyses will continue to be reduced, thus providing the community with even more cost effective and useful tools to develop the turbomachinery of the future. This effort has provided a step in the direction of this future work. In the future, CFD analysis will continue to grow in popularity as an analysis tool for not only turbomachinery applications, but all sorts of fluid mechanics applications. It will provide an extremely cost- and time-effective mechanism for analysis, thus reducing the investment in both time and dollars to achieve a successful and efficient design.

Conclusions

The CFD turbomachinery codes, Aardvark and Phantom, have been enhanced to decrease the time required for steady state or time-dependent flow solutions. This was accomplished by determining the values of input parameters which reduced the number of iterations required to achieve steady state, as well as lowering the CPU time per iteration. In addition, the users manuals were enhanced to provide additional information, thus making it possible for new users to get up to speed with the codes quickly, as well as providing current users with information not previously included.

Acknowledgements

The author would like to express his appreciation to the members of the Space Transportation Directorate, Fluid Dynamics Analysis Group (TD-64) for the opportunity to participate in this program. In particular, special thanks are in order to Dan Dorney, Lisa Griffin, and Robert Garcia. Their assistance made this effort a fantastic experience for the author, as well as a successful venture for NASA and the turbomachinery community.

References

[1] Sondak, D. L., and Dorney, D. J., "General Equation Set Solver for Compressible and Incompressible Turbomachinery Flows", AIAA Paper 2003-4420, 39th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, July 20-23, 2003, Huntsville, AL

[2] Venkateswaran, S., and Merkle, C.L., "Analysis of Preconditioning Methods for the Euler and Navier-Stokes Equations", Von Karman Institute Lecture Series, March 8-12, 1999.