Final Report
NAG3-2502

DEVELOPMENT OF COMPUTATIONAL AEROACOUSTICS CODE FOR JET NOISE AND FLOW PREDICTION

submitted to

National Aeronautics and Space Administration
Glenn Research Center at Lewis Field
Cleveland, Ohio

Dr. Theo G. Keith, Jr.
Distinguished University Professor

Dr. Duane R. Hixon
Senior Research Associate

Department of Mechanical Industrial and Manufacturing Engineering
The University of Toledo
Toledo, Ohio 43606

Final Report

For

NASA Grant NAG3-2502

Entitled

Development of Computational Aeroacoustics Code for Jet Noise and Flow Prediction

By

Duane R. Hixon
Senior Research Associate, University of Toledo
7/29/02

1) Introduction and Previous Work

Accurate prediction of jet fan and exhaust plume flow and noise generation and propagation is very important in developing advanced aircraft engines that will pass current and future noise regulations. In jet fan flows as well as exhaust plumes, two major sources of noise are present: large–scale, coherent instabilities and small–scale turbulent eddies. In previous work for the NASA Glenn Research Center, three strategies have been explored in an effort to computationally predict the noise radiation from supersonic jet exhaust plumes. In order from the least expensive computationally to the most expensive computationally, these are:

1) Linearized Euler equations (LEE).
2) Very Large Eddy Simulations (VLES)
3) Large Eddy Simulations (LES)

The first method solves the linearized Euler equations (LEE). These equations are obtained by linearizing about a given mean flow and the neglecting viscous effects. In this way, the noise from large–scale instabilities can be found for a given mean flow. The linearized Euler equations are computationally inexpensive, and have produced good noise results for supersonic jets where the large–scale instability noise dominates (e.g., Refs. 1–5), as well as for the tone noise from a jet engine blade row (e.g., Refs. 6–7). However, these linear equations do not predict the absolute magnitude of the noise; instead, only the relative magnitude is predicted. Also, the predicted disturbances do not modify the mean flow, removing a physical mechanism by which the amplitude of the disturbance may be controlled. Recent research for isolated airfoils[8] indicates that this may not affect the solution greatly at low frequencies.

The second method addresses some of the concerns raised by the LEE method. In this approach, called Very Large Eddy Simulation (VLES), the unsteady Reynolds averaged Navier–Stokes equations are solved directly using a high–accuracy computational aeroacoustics numerical scheme. With the addition of a two–equation turbulence model and the use of a relatively coarse grid, the numerical solution is effectively filtered into a directly calculated mean flow with the small–scale turbulence being modeled, and an unsteady large–scale component that is also being directly calculated. In this way, the unsteady disturbances are calculated in a nonlinear way, with a direct effect on the mean flow. This method is not as fast as the LEE approach, but does have many advantages to recommend it; however, like the LEE approach, only the effect of the largest unsteady structures will be captured. An initial calculation was performed on a supersonic jet exhaust plume, with promising results[9], but the calculation was hampered by the explicit time marching scheme that was employed. This explicit scheme required a very small time step to resolve the nozzle boundary layer, which caused a long run time. Current work is focused on testing a lower–order implicit time marching method to combat this problem.

The third method is the Large Eddy Simulation (LES). In the LES method, the full nonlinear Navier–Stokes equations are solved for all time and space scales except the small turbulent eddies, which are modeled (e.g., Ref. 10). The advantage of this method is that it directly calculates the mean flow, large scale eddies, and a large range of the smaller eddies. However, a large number of grid points and time steps are required in order to accurately capture the flow dynamics over such a wide range of scales, which translates to large amounts of memory and CPU time requirements.

All three of these methods require highly accurate computational schemes in order to reduce the amount of grid points and time steps required to resolve the unsteady flow dynamics. These computational schemes must be used on nonuniform grids wrapped about complex geometries. The boundary conditions for these schemes must be accurate and highly nonreflective in order to allow the smallest possible computational domain to be used. Also, these schemes must be highly parallelizable in order to take advantage of the low–cost parallel PC clusters that are replacing the high–cost supercomputers of past years. In past work, all four of these issues were investigated (Refs. 11–20), and a three–dimensional, generalized coordinate structured multiblock LES/VLES code was developed and tested (refs 21–22).

This code worked well, but had several flaws. First, it was not coded in a safe, object–oriented manner, so it was hard for several programmers to work on the code simultaneously. Second, the parallelization was not designed for low–cost clusters; the code was heavily synchronized and passed many short messages at every time step. Third, the two time stepping methods in the code were designed for flows with a single time scale; when calculating a problem with a wide range of time scales, the code would force all grid points to march at the time step required to resolve the smallest time scale regardless of whether that temporal resolution was needed locally.

The benchmark results from that code were very good[22], so it was proposed to rewrite the code to be more maintainable and portable, more efficient in its parallel performance on low–cost clusters, and to incorporate a wider range of time stepping methods including single–step explicit, single–step implicit, and multiple–step explicit. This paper details the work accomplished so far on this effort in FY 2001 and 2002.

2) Research Conducted in FY 2001 and FY2002

Work began on the code in October 2000. The most important choices in the code development process were made early on: the programming language to use, the parallelization method to use, and the data structure of the code itself. The algorithms used in the code were chosen before this work began, and no development of new algorithms or methods was done during this work.

The programming language used for the CAA code was Fortran 90, due to its computational efficiency, its dynamic memory allocation, its module safety, its data structure capabilities, and its portability between different computational platforms. For portability between different parallel architectures, the Message Passing Interface (MPI) was used for the code parallelization.

The code has two parts, each with its own data structure. The first part of the code is the input grid decomposition into time stepping levels for the multiple–time–stepping Adams–Bashforth method. In this section, the initial grid blocks are divided into groups of blocks, each stepping at a multiple of the smallest time step used in the calculation. Once this procedure is complete, each group of blocks are divided across the computational nodes in the parallel computed; thus, each node has a block or group of blocks for each time stepping level.

Since the code does not know beforehand how many time stepping levels or how many blocks will result from this decomposition, a series of linked list data structures are used to facilitate adding and subtracting blocks from each level or node.

Because a linked–list data structure is computationally inefficient for the actual flow computation section of the CAA code, another data structure is defined after the decomposition section is complete and the block data locations are known. This data structure uses fixed arrays for the block and flow data; after the data is transferred from the linked lists to the fixed arrays, the linked lists are deallocated until the next time a parallel decomposition is needed.

In the second part of the code, the actual flow and noise computation is performed. Since the previous code performed many of the same operations, a large percentage of the previous code was reused with only slight modifications. In fact, the flow computation section of the code was running and tested as soon as March 2001. However, the initial data decomposition structures took until November 2001 to be completed to a point that the code could be initially tested.

Work was also performed to improve the interface communication between interior grid blocks. As opposed to the previous code, the input required from the user to define interior block interfaces is much less, with a large amount of error–checking in the code to identify incorrect user inputs. Also, the code allows arbitrary translations and rotations between the receiving and sending blocks; this will allow reduced grids to be used for an interior cascade or fan simulation.

During the development process, a requirement for the code was to allow new boundary conditions to be implemented easily for testing. To accomplish this, the boundary condition routines are completely modular and contained within the block data structure. This allows new boundary conditions to be added in the block data module and automatically propagate throughout the code data structure. The boundary conditions that are currently implemented and being tested in the code are the acoustic radiation inflow condition and Tam and Webb outflow condition (both implemented by

Dr. S. Sawyer) as well as the Giles inflow and outflow conditions (both implemented by Dr. M. Nallasamy).

The boundary conditions that are implemented in the current code are much more general than the same boundary condition in the old code. For example, in the old code the acoustic radiation boundary condition required the mean flow to be aligned with the x–axis, at times requiring the grid to be rotated to correctly orient the flow. In the current code, the mean flows for the acoustic radiation boundary condition, Tam and Webb outflow condition, and the Giles inflow and outflow boundary conditions allow the mean flow to be in any direction.

The wall boundary condition has been investigated by Dr. R. Dyson in an attempt to increase the accuracy of the inviscid wall boundary condition.[23] Initially, the Hixon boundary condition used in the old code was implemented[18], with slight modification to make it independent of the time marching method used. When applied to the validation benchmark cases that the old code had previously been used to calculate, the performance and accuracy of the wall boundary condition, which used a correction to the normal derivative of the pressure, was good. However, when applied to the more heavily loaded 2–D cascade benchmark problem (Ref. 24), the wall boundary condition began to show some shortcomings. In this case, a vortical and entropy boundary layer developed, showing that the wall condition was not correct. To attempt to improve this condition, Dr. R. Dyson formulated a 1–D characteristic boundary condition that corrected the normal derivative of the velocity normal to the wall in addition to the normal derivative of the pressure. In this way, the normal derivative correction at the wall corresponds to an incoming acoustic wave. This condition improved the performance of the wall boundary condition, and work continues on deriving and implementing higher–order wall boundary conditions.

Once the code was running for complex 2D geometries, the code needed to be parallelized. The code was written to minimize the number of messages between processors, as well as to allow an automatic parallelization method to be implemented. The initial parallelization was completed, with good results on both multi–processor SGI's and a low–cost Intel Pentium 4 cluster.[25] Work is continuing to completely debug the parallel message passing in the code and to add the automatic parallelization routine.

The 2D benchmark problem of Ref. 24 has opened new areas of research needed for a CAA code to be used for more realistic problems of this type. The results of the code for the mean flow through the blade row are promising, with the wall boundary condition issues discussed above.

Another interesting result is the effect that the initial transient produced by the blade row has on the mean flow specification. The transient is caused by the effect of the solid blade on the initial flow condition; the blade turns the flow and causes an acoustic wave to exit the inflow boundary. This wave turns the incoming flow, which changes the problem specification. While the initial condition can be adjusted to recover the specified mean flow, this requires an iterative method to determine a usable initial condition. Current research is directed towards developing boundary conditions that will be applied only to the mean flow, using the nonreflective boundary conditions on the unsteady flow.

To sum up, the code, which is 75,000 lines at last count, is designed to be a portable, maintainable, generally applicable high–accuracy unsteady code for 2D and 3D compressible nonlinear flows with complex geometries. The first results are beginning to be obtained, and appear very promising. Most of the data structures and some coding already exists for automatic parallelization, multiple–time–stepping Adams–Bashforth time marching, grid motion and grid adaption, viscosity, and several turbulence models. This code is a solid foundation for future work.

3) Conclusions and Proposed Future Directions

In conclusion, several lessons have been learned in the work conducted in FY 2001–2002. First, and most important, the algorithm used in the code is capable of producing high–quality results for complex, nonlinear flows on curvilinear grids with grid singularities. Second, the object–oriented programming style used in this work allows multiple programmers to work on the code simultaneously with ease, as well as allowing new boundary conditions, spatial derivative methods, and time marching methods to be implemented in the existing framework of the code. Third, the use of standard Fortran 90 and MPI allows full portability across many computational platforms. The work so far has provided a strong foundation for a high–accuracy CAA code that can be maintained and expanded well into the future.

In the future, it is proposed to continue this work, focusing on these areas:

1) Continue work on the automatic parallelization and load balancing in the code.
2) Finish implementation of the single–time–stepping Adams–Bashforth method in preparation for the multiple–time–stepping Adams–Bashforth method to be used.
3) Implement mean–flow boundary conditions to improve the code's applicability to fan noise problems.
4) Extend the code to three dimensions and validate for a stator blade geometry.

The exceptional progress made so far and the general applicability built into this code illustrate the strength of the foundations of this work. In the near future, codes such as this one running on large distributed computers will be able to directly predict the flow and noise from jet fan and exhaust configurations.

Acknowledgements

The author would like to acknowledge Dennis Huff, Dr. Edmane Envia, Dr. James R. Scott, Dr. M. Nallasamy, Dr. Scott Sawyer, Dr. Rodger Dyson, Ms. Danielle Koch, Dr. Vladimir Golubev, Dr. Reda Mankbadi, Dr. James Bridges, Dr. Theo. G. Keith and Dr. Andrew T. Norris for their important contributions to this work.

4) References

1) Mankbadi, R. R., Hixon, R., Shih, S.–H., and Povinelli, L. A., 'Linearized Euler Equations as a Tool in Jet Noise Prediction', *AIAA Journal*, Vol. 36, No. 2, 1998, pp. 140–147.
2) Hixon, R., Shih, S.–H., and Mankbadi, R. R., 'Direct Prediction of the Three–Dimensional Acoustic Field of a Supersonic Jet Using Linearized Euler Equations', CEAS/AIAA Paper 95–116, 2nd AIAA/CEAS Aeroacoustics Conference and Exhibit, Munich, Germany, June 1995.
3) Hixon, R., Shih, S–.H., and Mankbadi, R. R., 'Numerical Simulation of the Effect of Heating on Jet Noise', *FED–Vol. 219, Computational Aeroacoustics*, eds. A. S. Lyrintzis, R. R. Mankbadi, O. Baysal, M. Ikegawa, ASME, June 1995, p. 89–94.
4) Hixon, R., Shih, S.–H., and Mankbadi, R. R., 'Effect of Input on Jet Noise Prediction', Proceedings of International Congress on Fluid Dynamics & Propulsion, Cairo, Egypt, Dec. 1996.
5) Hixon, R., Shih, S.–H., and Mankbadi, R. R., 'Effect of Coannular Flow on Linearized Euler Equation Predictions of Jet Noise', AIAA Paper 97–0284, 35th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 1997.
6) Verdon, J. M., Montgomery, M. D., and Chuang, H. A., 'Development of a Linearized Unsteady Euler Analysis with Application to Wake/Blade–Row Interactions', NASA CR–1999–208879, March 1999.
7) Verdon, J. M., 'Linearized Unsteady Aerodynamic Analysis of the Acoustic Response to Wake/Blade–Row Interaction', NASA CR–2001–210713, January 2001.
8) Crivellini, A., Golubev, V., Mankbadi, R., Scott, J., Hixon, R., and Povinelli, L., 'Nonlinear Analysis of Airfoil High–Intensity Gust Response Using a High–Order Prefactored Compact Code', AIAA Paper 2002–2535, 8th AIAA/CEAS Aeroacoustic Conference, Breckenridge, CO., June 2002.
9) Mankbadi, R. R., Hixon, R., and Povinelli, L. A., 'Very Large Eddy Simulations of Jet Noise', AIAA Paper 2000–2008, 6th AIAA/CEAS Aeroacoustics Conference, Lahaina, Hawaii, June 2000.
10) Mankbadi, R. R., Shih, S.–H, Hixon, R., and Povinelli, L. A., 'Large–Scale Simulations of Nonlinear Effects in Jet Noise', *Journal of Propulsion and Power*, Vol. 14, No. 5, 1999.
11) Hixon, R., 'Prefactored Small–Stencil Compact Schemes', *Journal of Computational Physics*, Vol. 165, 2000, p. 522–541.
12) Hixon, R., 'Nonlinear Comparison of High–Order and Optimized Finite–Difference Schemes', *International Journal of Computational Fluid Dynamics*, Vol. 13, No. 3, 2000, p. 259–277.
13) Hixon, R., Shih, S.–H., Dong, T., and Mankbadi, R. R., 'Evaluation of Generalized Curvilinear Coordinate Transformations Applied to High–Accuracy Finite–Difference Schemes', AIAA Paper 98–0370, 36th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 1998.
14) Hixon, R., 'Numerically Consistent Strong Conservation Grid Motion for Finite–Difference Schemes', *AIAA Journal*, Vol. 38, No. 9, 2000, p. 1586–1593.

15)Hixon, R.,Shih, S. H., and Mankbadi, R. R. 'Numerical Treatment of Cylindrical Coordinate Centerline Singularities', AIAA Paper 99–2391, 35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Los Angeles, CA, June 1999.

16)Hixon, R., Shih, S.–H., and Mankbadi, R. R., 'Evaluation of Boundary Conditions for Computational Aeroacoustics', *AIAA Journal*, Vol. 33, No. 11, 1995, pp. 2006–2012.

17)Hixon, R., Shih, S.–H., and Mankbadi, R. R., 'Evaluation of Boundary Conditions for the Gust– Cascade Problem', *Journal of Propulsion and Power,* Vol. 16, No. 1, 2000, pp. 72–78.

18)Hixon, R., 'Curvilinear Wall Boundary Conditions for Computational Aeroacoustics', AIAA Paper 99–2395, 35th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Los Angeles, CA, June 1999.

19)Hixon, R., '3–D Characteristic–Based Boundary Conditions for CAA', AIAA Paper 2000–2004, 6th AIAA/CEAS Aeroacoustics Conference, Lahaina, Hawaii, June 2000.

20)Hixon, R., Shih, S. H., Mankbadi, R. R., and Scott, J. R., 'Time Domain Solution of the Airfoil Gust Problem Using a High–Order Compact Scheme', AIAA Paper 98–3241, 34th AIAA/ ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Cleveland, OH, July 1998.

21)Hixon, R. and Wright, J., 'Validation of a High–Order Prefactored Compact Scheme on Nonlinear Flows with Complex Geometries', *Third Computational Aeroacoustics (CAA) Workshop on Benchmark Problems*, NASA CP–2000–209790, August 2000, p. 117–132.

22)Hixon, R., Mankbadi, R. R., and Scott, J. R., 'Validation of a High–Order Prefactored Compact Code on Nonlinear Flows with Complex Geometries', AIAA Paper 2001–1103, 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 2001.

23)Dyson, Rodger W. and Hixon, R., 'Towards Arbitrary Accuracy Inviscid Surface Boundary Conditions', AIAA Paper 2002–2438, 8th AIAA/CEAS Aeroacoustic Conference, Breckenridge, CO., June 2002.

24)Envia, E., '2D Benchmark Problem for Evaluating Non–Linear, Time–Accurate Computations of Gust Blade–Row Interaction', NASA Glenn Research Center, March 2002.

25)Hixon, R., Nallasamy, M., Sawyer, S. D., 'Parallelization Strategy for an Explicit Computational Aeroacoustics Code', AIAA Paper 2002–2583, 8th AIAA/CEAS Aeroacoustic Conference, Breckenridge, CO., June 2002.