NASA/CR—2003-212323

Turbofan Noise Propagation and Radiation at High Frequencies

Walter Eversman
University of Missouri at Rolla, Rolla, Missouri

May 2003

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at 301–621–0134

- Telephone the NASA Access Help Desk at 301–621–0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076

NASA/CR—2003-212323

Turbofan Noise Propagation and Radiation at High Frequencies

Walter Eversman
University of Missouri at Rolla, Rolla, Missouri

National Aeronautics and
Space Administration

Glenn Research Center

May 2003

Acknowledgments

This report contains preliminary
findings, subject to revision as
analysis proceeds.

TURBOFAN NOISE PROPAGATION AND RADIATION
AT HIGH FREQUENCIES

Walter Eversman
University of Missouri at Rolla
Rolla, Missouri 65401

INTRODUCTION

This report summarizes progress on NASA Glenn Research Center Grant NAG3–2718 to the University of Missouri-Rolla. This grant was awarded on February 22, 2002 and this report covers the performance period to September 30, 2002.

There is considerable overlap in research effort with previous NASA Glenn Grant NAG3–2340, as the current effort represents a continuation and extension of this previous grant, which with a no cost supplement terminated on January 31, 2002.

This report outlines progress on each task in the original proposal. In addition to progress on several of the specifically proposed tasks, considerable progress has been made in FEM algorithm development with the intent of introducing computational efficiencies required to model high frequency propagation and radiation and to open the possibility of expanding the scope of the modeling capability to three dimensional duct and nacelle geometries.

Appended to this document is a paper presented at the 8th AIAA/CEAS Aeroacoustics Conference in June 2002. This paper overlaps the present grant and the previous grant identified above, and it is noted that this paper has also been appended to the final report for NAG3–2304.

TASKS

Following is a brief description of progress on each task described in the proposal:

Task 1.

Inlet and aft fan duct radiation codes will be substantially modified to introduce cubic isoparametric serendipity elements in place of quadratic elements currently used.

There is increasing interest in the use of the use of optimization procedures for the design of acoustic treatment. The major emphasis in the current research effort is the improvement of FEM duct propagation models to achieve increased computational efficiency with no sacrifice in accuracy. Work on previous grant NAG3–2109 [1] revealed that the structure of the problem of acoustic propagation in non-uniform flow, cast in an acoustic potential formulation with acoustic pressure obtained by post-processing, requires mesh refinement (nodes per wave length) in excess of that generally used in the case of the extensively studied Helmholtz equation [2,3]. This added refinement was found to be due to post-processing and the generation of acoustic pressure from acoustic potential, an operation requiring the numerical derivative of acoustic potential. The proposed remedy was the use of higher order interpolation in the finite element

formulation. This was studied in grant NAG3–2304 and this work has been continued under the present grant and preliminary results reported in reference [4].

Work began with the study of the one dimensional convected wave equation for non-uniform ducts in terms of acoustic potential and was extended to axially symmetric ducts with reflection free terminations, and finally to axially symmetric propagation and radiation from inlet and aft fan ducts. Originally used quadratic isoparametric elements were replaced by cubic and quadratic elements and accuracy and efficiency was extensively studied. It was found that large improvements could be achieved by the use of cubic elements, with much smaller additional improvement attributable to fourth order elements. Little difference in accuracy was noted between serendipity shape (interpolation) functions and Lagrangian shape functions, so the more efficient serendipity shape functions were used in most of the study.

Cubic isoparametric serendipity elements have been introduced successfully into inlet and aft fan duct radiation codes leading to a significant increase in the frequency range over which these codes can be used for acoustic modeling. Reduction in CPU time for a given level of accuracy in high frequency cases varies from 30 to 40%. The following issues have been addressed:

    a. Cubic triangular elements have been used in the transition region near the duct exit lip in the aft radiation code, with good results.

    b. Transition elements in the jet shear layer for aft fan duct radiation have been modified to ensure compatibility with both cubic rectangular and triangular elements.

    c. Mapped infinite elements were modified to have interpolation functions of cubic order in the transverse direction in order to make them compatible with cubic elements in the near field. It was found, however, that transverse interpolation functions of quadratic order were adequate in the far field, so transition elements connecting the transverse quadratic far field the cubic order near field were introduced to limit the bandwidth induced by the infinite elements.

    d. The new impedance boundary elements, suitable for cubic interpolation, have not yet been implemented. This additional slight modification will be completed shortly.

    e. Convergence studies of error norms in cases with available baseline solutions and visual observations of the smoothness of iso-pressure contours, based on comparison with previous versions of the codes, show significant gains in efficiency using cubic elements without sacrificing accuracy. The reduction in CPU time for relatively high frequency cases can be as much as 40%.

Task 2.

    Restructuring of mapped infinite elements for reflection free boundary conditions to allow an arbitrary apparent source location will be investigated.

Mapped infinite wave envelope elements for non-reflecting boundary conditions were developed under grant NAG3–2109 [5]. These elements were studied in more detail under grant NAG3–2304 and shown to provide a means for extrapolation of a relatively near field solution to the far field, eliminating the need to use additional costly post-processing based on the Kirchoff Integral Equation [6]. Imbedded in the interpolation functions for the mapped infinite wave envelope elements is the capability to represent arbitrary apparent source locations, possibly improving the non-reflecting boundary condition without the expense of the use of high order

interpolation functions. This was precipitated by difficulties noted in achieving good solutions in cases of particularly difficult geometries.

The use of an arbitrary apparent source location for the mapped infinite elements was investigated after cubic elements were implemented in the radiation codes. It was found that the introduction of cubic elements in the near field affected the far field solution and resolved most of the problem, and it obviated the need to introduce the additional complication of arbitrary source apparent location. It is still necessary in difficult duct geometries (particularly aft fan duct cases) to adjust the origin of the mesh coordinate origin to obtain good solutions.

Task 3.

Mesh generation and mean flow codes will be restructured to conform to the introduction of cubic elements.

Mesh generation and mean flow codes which are required for data generation for the inlet and aft fan duct acoustic radiation codes have been restructured for cubic elements.

Task 4.

Post processing to produce acoustic pressure from acoustic potential and the generation of plotting files for Postscript and Tecplot plotting will be made compatible with the modified radiation codes.

Post processing, to produce acoustic pressure from acoustic potential, has been made compatible with the modified radiation codes. Output files are suitable for direct import into Tecplot for plotting purposes. Postscript files are also updated.

Task 5.

Duct interior propagation codes, including a code with a lined splitter model, will be modified to a level compatible with radiation codes.

Cylindrical/annular duct interior propagation codes and the corresponding mesh generation and mean flow codes have been modified to use cubic elements. The lined splitter code has yet to be modified.

Task 6.

If deemed appropriate by NASA, efficient in-duct propagation codes will be used as a test bed for new lining concepts, such as extended reaction linings suitable for broad-band attenuation.

In-duct propagation codes have not yet been used for new lining concepts in the current program. There is considerable interest in design of acoustic treatment for broad-band attenuation. In a program funded by NASA Langley Research Center through Goodrich Aerospace, the issue of lining design for multiple modes and broad-band cases has been

considered. In an attempt to minimize the effect of the identity of individual modes, an assumption of equal power in each propagating mode with phases has been tried as an option. Results indicate that maximum achievable attenuation and the corresponding optimum lining configuration are very sensitive to the relative phasing of incident acoustic modes. This suggests that a statistical description of acoustic modal content is required, and that lining parameters and attenuation can only be determined in a statistical sense unless precise source amplitude and phase data is available. This adds to the numerical intensity of design optimization and reinforces the need for efficient computation.

Task 7.

Experience with development of a duct interior propagation code with a splitter model can lead to a similar capability in the aft fan duct radiation code.

Aft fan duct radiation code with splitter capability will not be addressed until the interior propagation splitter code has been upgraded to cubic elements. The requirement here is to redefine the aft fan duct geometry to allow the inclusion of a splitter interior to the duct. This will be deferred to the second year of the program.

Task 8.

A third code in which aft fan duct and inlet radiation is combined has been developed as part of the continuing research program at UMR. Continuing work on this has been suspended while new elements have been investigated. After the new elements have been proven in the inlet and aft fan duct codes, they will be implemented in this code as well.

The combined aft fan/inlet radiation code has not been updated with cubic elements. Computational requirements for acoustic radiation predictions at meaningful frequencies will require improvements in solution algorithms, as discussed in the following section.

PROGRESS BEYOND SPECIFIC TASKS

In addition to specific tasks outlined in the proposal, it has been deemed appropriate to reconsider the finite element assembly and linear equation solving procedures used in the radiation codes and in the interior propagation codes. The purpose is to attempt to obtain further reductions in computational time and to allow modeling of higher frequency duct propagation. The current codes use the frontal solution procedure, while current progress in linear equation solvers has centered on LU decomposition. The in-duct propagation code has been restructured to incorporate the Super LU sparse matrix algorithm available in the public domain. Also under investigation is an algorithm for generation of optimal nodal numbering to further enhance computational efficiency. Results to date show 40 to 50% reductions in computer time for the Super LU as compared to the original frontal solver. So far cases as large as 61,000 degrees of freedom have been studied. At this point required RAM becomes a problem, and further work must be done on storage allocation. This is complicated be the fact that part of the Super LU uses

C language routines, and storage demands are difficult to determine. This extension of the investigation has proved to be very fruitful, and will be continued.

Given the potential gains in frequency range that can be modeled, and in computational efficiency that might be achieved by the combination of cubic elements and improved equation solving, it may be appropriate to consider reprioritizing some of the tasks in the original proposal.

APPENDICES

Attached to this document are three appendices. Appendix A references the manuscript for AIAA Paper 2003–2515, which formally reports results of studies of accuracy and convergence of computations for propagation and radiation obtained from the inlet and aft radiation codes with quadratic and cubic elements. The conclusion is that cubic elements are superior in terms of accuracy for specified number of nodes and in terms of the number of nodes to obtain a specified accuracy. Based on these results these codes have been modified to implement the higher order elements.

Appendix B discusses the preliminary results obtained in implementing a new banded, sparse LU linear equation solving routine in interior duct code. The interior duct code was chosen as the test bed for this routine because the relatively small bandwidth leads to short computation times and more efficient debugging. Implementation in the radiation codes will follow. The conclusion is that significant gains in computational efficiency are achievable. These solution routines include some C code which requires dynamic memory. The HP system used in this investigation is limited in this respect, requiring further study to get around this problem. The possibilities of switching to another platform or using networked PCs are being considered.

Appendix C is a working paper examining the possibility of the development of a sparse, banded, block LU decomposition solution algorithm to minimize the requirement for both static and dynamic memory.

REFERENCES

1. Listerud, E. and Eversman, W. 2000 AIAA Paper 2000–1953, AIAA/CEAS 6th Aeroacoustics Conference, Maui, Hawaii. Acoustic Propagation at High Frequencies.
2. Ihlenberg, F. and Babuska, I. 1995 International Journal for Numerical Methods in Engineering, 38, 3745–3774. Dispersion Analysis and Error Estimation of Galerkin Finite Element Methods for the Helmholtz Equation.
3. Ihlenberg, F. 1998 **Finite Element Analysis of Acoustic Scattering.** Springer Verlag, New York.
4. Listerud, E. and Eversman, W. 2002 AIAA Paper 2002–2515, AIAA/CEAS 8th Aeroacoustics Conference, Breckenridge, Colorado. Accuracy and Efficiency in FEM Modeling of Turbofan Acoustic Radiation.
5. Eversman, W. 1999 Journal of Sound and Vibration 224(4), 665–687. Mapped Infinite Wave Envelope Elements for Acoustic Radiation in a Uniformly Moving Medium.
6. Danda Roy, I. and Eversman, W. 2001 AIAA Journal 39(12), 2255–2261. Far Field Calculations for Turbofan Noise.

APPENDIX A

Listerud, E. and Eversman, W. 2002 AIAA Paper 2002–2515, Presented at the AIAA/CEAS 8th Aeroacoustics Conference, Breckenridge, Colorado. Accuracy and Efficiency in FEM Modeling of Turbofan Acoustic Radiation.

Please note that this paper bridges the time period of NASA Grants NAG3–2304 and NAG3–2718. It was included in the final report for the Grant NAG3–2718.

APPENDIX B

Preliminary results using a sparse, banded LU decomposition linear equation solution routine.

Comparison of frontal solver and SuperLU

The SuperLU solver for sparse matrices was implemented in the acoustic propagation code using quadratic (p=2) elements. This code was then run for various degrees of freedom to compare the results and CPU time on an HP 9000/899/K570 machine using the HP–UX 10 operating system. From Figure 1 it is clear that the SuperLU solver performs better with respect to time than the frontal solver, with the exception of relatively small cases (<10,000 DOF). This leads to a significant reduction in the CPU time required to run the entire set of codes. For instance, at about 46,000 DOF the propagation code with the SuperLU solver uses about 48% less time than the propagation code with the frontal solver, leading to an overall reduction of about 39% in CPU time. The problem with the SuperLU is the dynamic memory allocation required during run-time to execute successfully. On the HP–UX 10 operating system the largest case run was about 61,000 DOF. It is difficult up front to estimate the size of the domain that can run since most memory is sucked up dynamically in the C code. On the HP–UX 10 operating system the static and the dynamic memory are different entities and independent of each other. In this system the maximum static memory any code could use is above 1 GB, since the frontal solver has been able to run at a maximum of about 1.2 GB. Supposedly, the maximum ceiling for dynamic memory allocation is only about 370 MB, therefore limiting the SuperLU to cases of less than 61,500 DOF on this system.
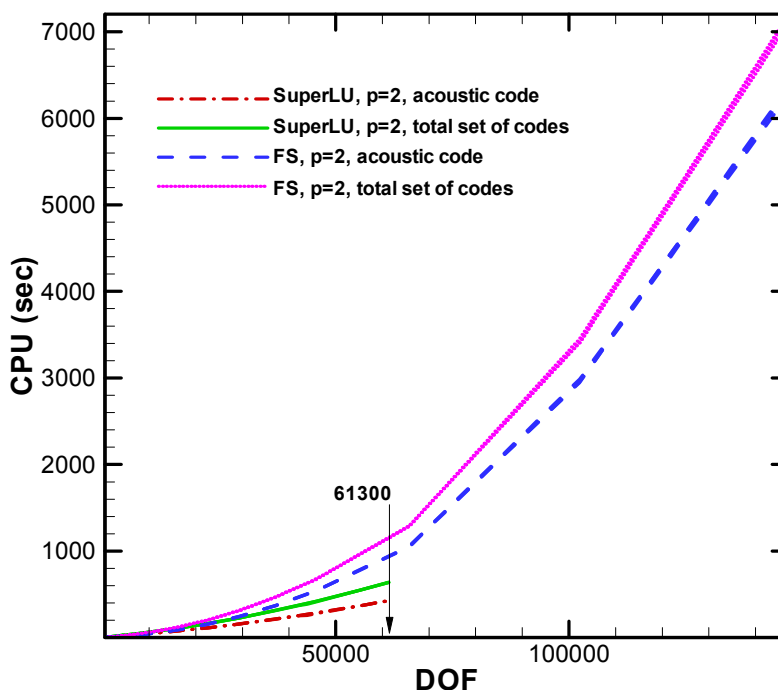

Figure 1 *Comparison of CPU time in seconds versus the total degrees of freedom in the computational domain for the acoustic propagation code and the total set of codes, using a frontal solver and SuperLU.*

Note that neither the frontal solver code nor the SuperLU code was optimized during compilation. Also, it is unclear what benefit, if any, there would be if SuperLU was used in the potential flow code since this one is not using the frontal solver, but rather a symmetric, banded LU solver that seems to be fairly efficient. It does, however, use significantly more static memory that the frontal solver. Also note that there is no significant difference in the solution of the two different solvers. A number by number comparison has been conducted for cases up to about 16,000 DOF.

Other recent solution approaches for aeroacoustic problems

In recent work Astley,[1] et al. consider propagation from turbofan inlets. Most of the results use a mesh of 43,228 nodes and the computations are performed on a single-processor 1GHz~PC with 1~GByte of RAM. The acoustic transmission code ACTRAN, developed by Far Field Technologies SA of Belgium, has been used. ACTRAN offers both a direct skyline solver and in-core and out-of-core sparse solvers, including SuperLU. It is unclear which solver Astley utilized.

Susan-Resiga and Atassi[2] consider an exterior aerodynamic-aeroacoustic problem and use the SuperLU solver. They reduce the memory requirements significantly by using a domain decomposition algorithm. The computational effort is also claimed to be reduced, even for a single processor computer. Their model is based on a 17,235 noded mesh.

It is interesting to take a look at a approach being used at the high end of the computational spectrum in aeroacoustics. Watson[3] considers 3-D models using up to 776,304 nodes. It must be noted that this paper only considers cases with symmetric stiffness matrices (no flow). Also, the computational platform in use is a SGI Origin 2000 with 13 GB of RAM and 8 processors. Watson uses a parallel sparse solver (ZPSLDLT) instead of a Vectored Space Solver (VSS) since the first is readily available to the public and the second is proprietary. VSS and ZPSLDLT have nearly identical time and memory requirements. The VSS is an LU sparse solver using LDL^T factorization rather than Choleski LL^T, but that is probably the case with most LU solvers these days (- the LL^T is slower). NASA's general purpose solver,[4] of which the VSS is a commercial version, has been run for cases up to 550,000 DOF.

Although these codes are predominantly meant for parallel solvers, Watson compared the performance of the ZPSLDLT with a traditional banded solver that also uses LDL^T factorization (Gauss-Doolittle factorization) for a single processor. His results states that the sparse solver consumes less CPU time and less RAM at higher frequencies.

It is clear that sparse solvers save both time and RAM compared to traditional banded LU-solvers. However based on Figure 1, the memory requirements for a frontal solver are, as expected, much less than for a sparse solver. In order for sparse solvers to be usable for a wider range of problems (higher frequencies) without requiring memory capacity only reserved for high end platforms, this memory issue must be dealt with.

References

1. Astley, R.J., Hamilton, J.A., Baker, N., and Kitchen, E.H., "Modeling tone propagation from turbofan inlets—The effect of extended lip liners," AIAA Paper 2002–2449, June 2002.
2. Susan-Resiga, R.F. and Atassi, H.M., "Domain-Decomposition Method for Time-Harmonic Aeroacoustic Problems," *AIAA Journal*, **39**, No. 5, 2001, pp. 802–809.
3. Watson, W.R., "Three-Dimensional Rectangular Duct Code with Application to Impedance Education," *AIAA Journal*, **40**, No. 2, 2002, pp. 217–226.
4. Storaasli, O.O., "Performance of NASA Equation Solvers on Computational Mechanics Applications," AIAA Paper 1996–1505, April 1996.

APPENDIX C

Working paper on the development of a sparse, banded, block LU decomposition.

Sparse, Banded, Block LU Linear Equation Solvers

Among other goals, this project aimed to do the following:

(i)   To minimize memory usage through the use of compact storage schemes
(ii)   To be able to solve large systems with hundreds of thousands of DOF (Degrees of Freedom) by keeping the elements of a large stiffness matrix in secondary storage and loading only the needed portions of the matrix in the primary memory
(iii)   To speedup the solution process by performing repeated solutions for the same stiffness matrix but different load vectors.
(iv)   To develop Piecewise Memory Super LU-Factorization technique.
(v)   To take advantage of the banded nature of the stiffness matrix in using SuperLU.

In order to use memory efficiently and speed up the solution process we preferred to use a sparse solver based on LU-decomposition. First, we wrote our own solver that used Harwell-Boeing Sparse Matrix storage format. Due to poor performance obtained in our original attempt, we decided to use SuperLU Solver [REF]. Harwell-Boeing Sparse Matrix format, Sparse LU-decomposition technique and SuperLU are explained in the following sections

Harwell-Boeing Sparse Matrix Format

In this project, we consider the direct solution of the linear system Ax = b where matrix A is assembled from the finite elements. It is banded and sparse.

Most entries in the sparse matrix A are zeros. This is true even within the band of A. Therefore, it is inefficient to use a 2-D array to store the sparse matrix. For example, if we have a matrix with the band width of 1,000 and 100,000 DOFs (Degrees of Freedom), we will need to store the band which has $1,000 \times 100,000 = 100M$ entries. Since 16 bytes are needed to store one complex number in double precision, totally 1.6G memory is required. This is very inefficient and unnecessary since we store many zeros. So we need special methods to store the matrix in compact format, i.e. only the non-zeros are represented.

Harwell-Boeing sparse matrix format [1] is a popular compact storage scheme. In this format, non-zero entries are stored in a 1-D array in a column-oriented way. It is called the numerical value array (VALUES). In addition, two extra arrays are needed to store row and column indices. They are called the row index array (ROWIND) and the column pointer array (COLPTR). The row index array contains the row indices of the corresponding elements stored in numerical value array. The column pointer array contains the pointers to the beginning of each column in the row index array and numerical value array. Thus, the entry of COLPTR(i) is the position in arrays ROWIND and VALUES where the i-th column starts. The length of COLPTR is $n+1$. (n is the number of columns) The entry of COLPTR($n+1$) indicates the beginning of fictitious column

number $n+1$. This entry is necessary because it is used to indicate how many non-zeros are in the last column. Following is an example as given in [1]:

$$A = \begin{bmatrix} 1. & -3. & 0 & -1. & 0 \\ 0 & 0 & -2. & 0 & 3. \\ 2. & 0 & 0 & 0 & 0 \\ 0 & 4. & 0 & -4. & 0 \\ 5. & 0 & -5. & 0 & 6. \end{bmatrix}$$

```
------------------------------------------------------
Sub      1   2   3   4   5   6   7   8   9   10  11
------------------------------------------------------
COLPTR   1   4   6   8   10  12
ROWIND   1   3   5   1   4   2   5   1   4   2   5
VALUES   1.  2.  5.  -3. 4.  -2. -5. -1. -4. 3.  6.
```

Sparse LU Factorization

To solve the linear system Ax = b, first A is factored into L and U components where L and U are lower and upper triangular matrices, respectively.

$$A = LU$$

Then the original problem becomes LUx = b. We solve for y vector using forward substitution:

$$Ly = b$$

Finally, we solve for x using backward substitution

$$Ux = y$$

If it is required to solve for many different b vectors without changing the stiffness matrix A, LU-decomposition provides an efficient algorithm. In this case, we perform LU factorization just once and then, for different b's, we simply do the forward and backward substitution.

First, we tried to develop our own sparse matrix solver. The idea is straightforward. We broke the whole matrix into blocks so that each block fits in memory.

$$A = \begin{bmatrix} A_1 & A_2 & \cdots & A_n \end{bmatrix}$$

After choosing the size of each block, we are able to extract the block A_i from the Harwell-Boeing format. The following example shows the extraction of column $3 - 4$ from the column pointer array.

```
---------------------------------------------------------------------
            ┌───────────┐
Sub     1   2 │ 3    4 │  5    6    7    8    9    10   11
            │           ├────────────────────────────────────────────
-----------─┼───────────┼─────────────────────────────────────────────
COLPTR  1   4 │ 6    8 │ 10   12
ROWIND  1   3 │ 5    1 │  4    2    5    1    4    2    5
VALUES  1.  2.│ 5.  -3.│ 4.  -2.  -5.  -1.  -4.  3.   6.
            └───────────┘
```

Then we can reconstruct block A_i in the 2-D array format (only rows having non-zeros) and apply the regular Gaussian elimination routine with *partial pivoting* to obtain the LU factorization of block A_i. Since A_i is a sparse matrix, the LU of A_i is again a sparse matrix although there might be some fill-ins (zeros becoming non-zeros). Therefore, we store the LU of block A_i again in the Harwell-Boeing format. Note that the complete LU factorization of matrix A is built block by block. We need to concatenate LU arrays of each block A_i and build global ones. A conventional storage format is used for L and U as in the following:

$$LU - I = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ m_{21} & a_{22} & a_{23} & a_{24} \\ m_{31} & m_{32} & a_{33} & a_{34} \\ m_{41} & m_{42} & m_{43} & a_{44} \end{bmatrix}$$

where

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 \\ m_{41} & m_{42} & m_{43} & 1 \end{bmatrix} \quad U = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

We implemented this algorithm in FORTRAN. Unfortunately, we found that it was very slow compared to the Frontal solver and SuperLU [2] solver. This is because we restore the matrix of block A_i in the 2-D array format, but many entries in block A_i are zeros. We spend significant time in processing zeros. To avoid this problem, the best way is to do the symbolic factorization

first before the numeric factorization. In the symbolic factorization, we first find where the fill-ins occur and perform operations only for these entries while minimizing the time spent for processing the zero entries. SuperLU which was developed at UC Berkeley uses this approach.

SuperLU Solver

SuperLU [2] is a general purpose sparse matrix LU solver. It implements the computation of the triangle factorization $P_rAP_c = LU$ and the solution of $Ax = B$ by evaluating $x = A^{-1}B = (P_c^{-1}LUP_r^{-1})^{-1}B = P_c(U^{-1}(L^{-1}(P_rB)))$. Here P_r and P_c are the permutation matrices for numerical stability and sparsity of LU. It has the following distinct steps:

1. Preorder the equations and variables to minimize the fill-ins of LU.
2. *Symbolic factorization*. In this step, it sets up data structures and allocates memory for L and U.
3. *Numeric factorization*. This step usually dominates the total time.
4. Triangular solves.

The input data format for SuperLU could be the Harwell-Boeing format, which is column compressed format.

SuperLU introduce the unsymmetric supernodes to speed up the numeric factorization. The basic idea of supernodes is to group together columns with the same non-zero structure, so they can be treated as a dense matrix for storage and computation. Supernodes were originally used for (symmetric) sparse Cholesky factorization $A = LL^T$.

Supernodes in sparse Cholesky can be determined during symbolic factorization, before numeric factorization begins. However, in unsymmetric sparse LU, partial pivoting is needed and the non-zero structure cannot be predicted before numeric factorization, so the supernodes must be identified on the fly.

Unsymmetric supernodes can be defined in different ways. The following figure shows four possible types of unsymmetric supernodes. The SuperLU solver uses the definition T2. In this definition, the supernode is a range (r:s) of columns of L with the same nonzero structure below the diagonal; that is L(r:s, r:s) is full lower triangular and every row of L(s:n, r:s) is either full or zero.
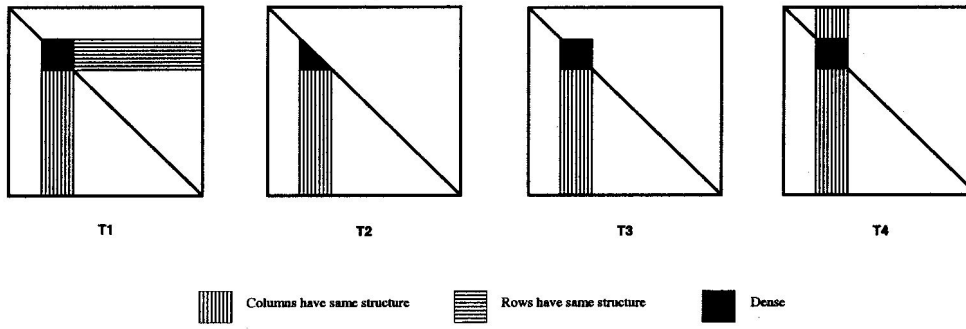
FIG. 2. *Four possible types of unsymmetric supernodes.*

The next figure shows an example from [2]. The supernodes are columns {1,2}, {3}, {4,5,6}, {7,8,9,10}.

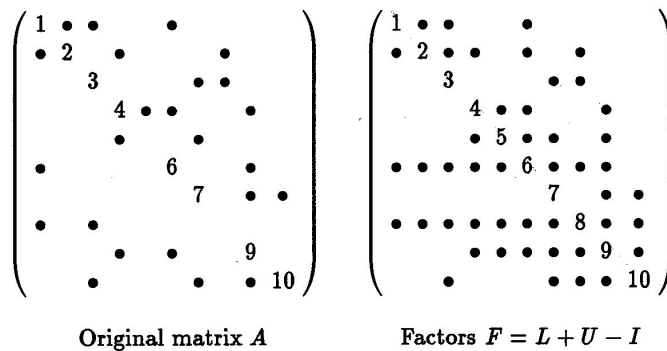Original matrix A Factors $F = L + U - I$

FIG. 3. *A sample matrix and its LU factors. Diagonal elements a_{55} and a_{88} are zero.*

The supernodes permit the use of higher level BLAS (Basic Linear Algebra Subprograms) and reduce inefficient indirect addressing. So the SuperLU solver is fast in doing the LU factorization.

Piecewise Memory Super LU-Factorization

As the DOF of the problem grows, SuperLU may also run into memory problems. This is because SuperLU needs to keep the complete A matrix and the LU factorization in primary memory. Note the structure of L and U cannot be accurately predicted prior to the factorization. SuperLU uses dynamically growing arrays to store L and U.

To improve the SuperLU solver, we need to incorporate it into our straightforward piecewise factorization algorithm. The new algorithm again breaks the entire matrix into blocks such that each block can fit in the memory. The difference is that within each block A_i we will try to use SuperLU to do the factorization. After the factorization of A_i, we need to store the A_i and its L_i

and U_i in the secondary storage (hard disk). Then we load the next block A_{i+1} into the memory and continue to perform the LU factorization. Finally, we concatenate the LU factorization of each block and get the global LU arrays which are stored in secondary storage.

The advantages of this algorithm is as follows:

(1) it uses the fast SuperLU solver;
(2) theoretically, it is able to handle the problem with any DOF. We can carefully divide the matrix A to avoid the memory problem.

The disadvantage of this algorithm is apparent. It involves some disk I/O which is time-consuming.

Banded Matrix

The SuperLU is a general-purpose sparse matrix solver. But the stiffness matrix generated by our application is a banded one. We think that we might be able to take advantage of this feature and further optimize the Piecewise Memory SuperLU for our application.

References

1. Users' Guide for the Harwell-Boeing Sparse Matrix Collection (Release I), Iain S. Duff, Roger G. Grimes, John G. Lewis, October 1992.
2. A Supernodal Approach to Sparse Partial Pivoting, James W. Demmel, Stanley C. Eisenstat, John R. Gilbert, Xiaoye S. Li, Joseph W.H. Liu, SIAM Journal on Matrix Analysis and Applications, Volume 20, Number 3, pp. 720–755.

REPORT DOCUMENTATION PAGE

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE May 2003	3. REPORT TYPE AND DATES COVERED Annual Contractor Report—Feb. 22, 2002–Sept. 30, 2002

4. TITLE AND SUBTITLE

Turbofan Noise Propagation and Radiation at High Frequencies

6. AUTHOR(S)

Walter Eversman

5. FUNDING NUMBERS

WU–708–87–23–00
NAG3–2718

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Missouri at Rolla
1870 Miner Circle
Rolla, Missouri 65409

8. PERFORMING ORGANIZATION REPORT NUMBER

E–13917

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR—2003-212323

11. SUPPLEMENTARY NOTES

Project Manager, Danielle Koch, Structures and Acoustics Division, NASA Glenn Research Center, organization code 5940, 216–433–5656.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Categories: 01 and 07   Distribution: Nonstandard

Available electronically at http://gltrs.grc.nasa.gov

This publication is available from the NASA Center for AeroSpace Information, 301–621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report summarizes progress on NASA Glenn Research Center Grant NAG3–2718 to the University of Missouri at Rolla This grant was awarded on February 22, 2002 and this report covers the performance period to September 30, 2002. There is considerable overlap in research effort with previous NASA Glenn Grant NAG3–2340, as the current effort represents a continuation and extension of this previous grant, which with a no cost supplement terminated on January 31, 2002. This report outlines progress on each task in the original proposal. In addition to progress on several of the specifically proposed tasks, considerable progress has been made in FEM algorithm development with the intent of introducing computational efficiencies required to model high frequency propagation and radiation and to open the possibility of expanding the scope of the modeling capability to three dimensional duct and nacelle geometries. Appended to this document is a paper presented at the 8th AIAA/CEAS Aeroacoustics Conference in June 2002. This paper overlaps the present grant and the previous grant identified above, and it is noted that this paper has also been appended to the final report for NAG3–2304.

14. SUBJECT TERMS

Acoustics; Fan noise; Aeroacoustics

15. NUMBER OF PAGES

21

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT