

NASA CR-159,359



NASA CR-159359

NASA-CR-159359  
19800025708

# RESEARCH REPORT

A FEASIBILITY STUDY  
OF A  
3-D FINITE ELEMENT SOLUTION SCHEME  
FOR  
AEROENGINE DUCT ACOUSTICS

By  
A. L. Abrahamson

Prepared for:



NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
LANGLEY RESEARCH CENTER  
HAMPTON, VIRGINIA 23665

CONTRACT NUMBER NAS1-15291

WYLE LABORATORIES REPORT NUMBER 51200

OCTOBER 1980

LIBRARY COPY

NOV 5 1980

LANGLEY RESEARCH CENTER  
LIBRARY, NASA  
HAMPTON, VIRGINIA



NF01171

# RESEARCH REPORT

A FEASIBILITY STUDY  
OF A  
3-D FINITE ELEMENT SOLUTION SCHEME  
FOR  
AEROENGINE DUCT ACOUSTICS

By  
A. L. Abrahamson

Prepared for:



NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
LANGLEY RESEARCH CENTER  
HAMPTON, VIRGINIA 23665

CONTRACT NUMBER NAS1-15291

WYLE LABORATORIES REPORT NUMBER 51200

OCTOBER 1980

**This Page Intentionally Left Blank**

## CONTENTS

|  | <u>Page</u> |
|--|-------------|
| INTRODUCTION . . . . .   | 1           |
| SYMBOLS . . . . .  | 2           |
| CHARACTERISTICS OF THE PROBLEM . . . . .   | 2           |
| Order . . . . .  | 2           |
| Computational Effort . . . . .   | 3           |
| Alternate Approaches for 3-D Analysis . . . . .  | 9           |
| DERIVATION . . . . .   | 11          |
| PROGRAM ORGANIZATION . . . . .   | 15          |
| SOLUTION TECHNIQUE EVALUATION . . . . .  | 16          |
| Checkout with Direct Equation Solver . . . . .   | 16          |
| Jacobi and SOR Iterative Methods . . . . .   | 16          |
| Conjugate Gradient Method . . . . .  | 17          |
| CONCLUSIONS . . . . .  | 26          |
| APPENDIX . . . . .   | 27          |
| REFERENCES . . . . .   | 33          |
| TABLES . . . . .   |             |
| I Approximate discretization order for solution of<br>sound propagation equations in an aeroengine<br>inlet . . . . .  | 3           |
| II Computing times for baseline model . . . . .  | 5           |
| III Estimated CPU times for solution of sound propa-<br>gation equations in typical aeroengine inlets . . . . .  | 6           |
| IV Estimated CPU times/iteration for iterative solution<br>of sound propagation equations in an aeroengine<br>inlet . . . . .                                  | 11          |
| FIGURES  |             |
| 1 Plot of theoretical operation counts ( $N_o$ ) versus<br>actual matrix assembly times for baseline model . . . . .   | 7           |
| 2 Plot of theoretical operation counts ( $N_o B^2$ ) versus<br>actual matrix solution times for baseline model . . . . .                                       | 8           |
| 3 3-D cubic serendipity element . . . . .  | 14          |
| 4 Effect of the number of finite elements per<br>wavelength ( $\lambda$ ) on solution accuracy . . . . .   | 21          |
| 5 Plots for Euclidean norm of residual vector<br>$\ r_i\ _2$ versus iteration number for 1 through<br>4 finite elements per wavelength ( $\lambda$ ) . . . . . | 22          |
| 6 Comparison of finite element conjugate gradient<br>solution with analytical plane wave solution<br>for poor first estimate solution vector . . . . .         | 23          |
| 7 Plot of $\ r_i\ _2$ versus iteration number for poor<br>first estimate of solution vector . . . . .  | 24          |

A FEASIBILITY STUDY  
OF A  
3-D FINITE ELEMENT SOLUTION SCHEME  
FOR  
AEROENGINE DUCT ACOUSTICS

By

A. L. Abrahamson  
Wyle Laboratories

INTRODUCTION

Development of two-dimensional (2-D) finite element solution schemes for aeroengine duct acoustics was initiated in the mid-1970s and has achieved considerable success. Analysis of acoustic propagation in axisymmetric ducts with nonuniform flow was made possible<sup>(1)</sup> and schemes for increasing computational efficiency have progressed to the stage where repeated analyses for aeroengine duct liner optimization are possible.<sup>(2)</sup>

The satisfactory results with 2-D models and the ability of segmented duct liners to give good sound attenuation characteristics indicated a potential advantage from the development of three-dimensional (3-D) finite element models.

The expected advantage was based on the rationale that since axially segmented acoustic duct liners have been demonstrated to provide improved sound attenuation characteristics over uniform liners, there is a significant likelihood that a combination of axially and circumferentially segmented liners will provide still better sound attenuation characteristics. In order to model the effect of this type of liner segmentation, it is necessary to discard the axisymmetric assumption and develop a fully 3-D model.

With currently available computers, this was anticipated to be an ambitious task. However, the use of different element formulation techniques and different linear equation solution algorithms to those used in the 2-D analyses indicated that the task might be possible.

## SYMBOLS

|                             |   |
|-----------------------------|---|
| A                           | Global matrix   |
| b                           | Right-hand side of global matrix equation                                 |
| B                           | Matrix bandwidth  |
| c                           | Speed of sound  |
| I                           | Identity of matrix  |
| L,M,N                       | Number of finite elements along coordinate directions of rectangular mesh |
| $N_0$                       | Global matrix order   |
| $N_i$                       | Finite element functional   |
| p                           | Acoustic pressure   |
| u,v,w                       | Acoustic velocities along coordinate directions                           |
| $\bar{u}, \bar{v}, \bar{w}$ | Mean aerodynamic velocities along coordinate directions                   |
| X                           | Solution vector   |
| x,y,z                       | Cartesian coordinates   |
| $\Omega$                    | Over-relaxation parameter   |
| $\omega$                    | Acoustic frequency  |
| $\phi$                      | Nodal parameter   |
| $\xi, \eta, \zeta$          | Element local coordinates   |
| $\bar{\rho}$                | Mean density  |

## CHARACTERISTICS OF THE PROBLEM

### Order

Numerical solution of the differential equations describing sound propagation in an aeroengine inlet requires discretization of the space within which the solution is required. The order of the discretization is dependent upon the spatial rate of change of the variables, which in turn is dependent upon acoustic source frequency and geometric and aerodynamic parameters. Precise calculation of the required order of discretization is not possible since the problem solution is not known a priori. Approximate calculations of required linear finite element discretization meshes for 2-D and 3-D analyses of typical aeroengine inlets, based upon a linear one-dimensional analytical model, are shown in Table I.

Table I

| Approximate Discretization Order for Solution of<br>Sound Propagation Equations in an Aeroengine Inlet |                    |             |                    |              |     |                              |                                 |              |     |     |                              |                                 |
|--|--------------------|-------------|--------------------|--------------|-----|------------------------------|---------------------------------|--------------|-----|-----|------------------------------|---------------------------------|
| Diameter<br>(Meters)   | Length<br>(Meters) | Mach<br>No. | Frequency<br>(kHz) | 2-D Solution |     |                              |                                 | 3-D Solution |     |     |                              |                                 |
|  |                    |             |                    | M            | N   | Matrix<br>Order<br>( $N_o$ ) | Matrix<br>Band-<br>Width<br>(B) | L            | M   | N   | Matrix<br>Order<br>( $N_o$ ) | Matrix<br>Band-<br>Width<br>(B) |
| 1.5  | 1.5                | 0           | 1                  | 20           | 40  | 6,400                        | 320                             | 40           | 40  | 40  | 512,000                      | 51,200                          |
|  |                    |             | 2                  | 40           | 80  | 25,600                       | 640                             | 80           | 80  | 80  | $4 \times 10^6$              | $2 \times 10^5$                 |
|  |                    |             | 3                  | 60           | 120 | 57,600                       | 960                             | 120          | 120 | 120 | $13.8 \times 10^6$           | $4.6 \times 10^5$               |
|  |                    | -.3         | 1                  | 20           | 60  | 9,600                        | 320                             | 40           | 40  | 60  | 768,000                      | 51,200                          |
|  |                    |             | 2                  | 40           | 120 | 38,400                       | 640                             | 80           | 80  | 120 | $6.1 \times 10^6$            | $2 \times 10^5$                 |
|  |                    |             | 3                  | 60           | 180 | 86,400                       | 960                             | 120          | 120 | 180 | $20.7 \times 10^6$           | $4.6 \times 10^5$               |
| 2.2  | 2.2                | 0           | 1                  | 30           | 60  | 14,400                       | 480                             | 60           | 60  | 60  | $1.7 \times 10^6$            | $1.15 \times 10^5$              |
|  |                    |             | 2                  | 60           | 120 | 57,600                       | 960                             | 120          | 120 | 120 | $13.8 \times 10^6$           | $4.6 \times 10^5$               |
|  |                    |             | 3                  | 90           | 180 | 129,600                      | 1,440                           | 180          | 180 | 180 | $46.6 \times 10^6$           | $1.04 \times 10^6$              |
|  |                    | -.3         | 1                  | 30           | 90  | 21,600                       | 480                             | 60           | 60  | 90  | $2.6 \times 10^6$            | $1.15 \times 10^5$              |
|  |                    |             | 2                  | 60           | 180 | 86,400                       | 960                             | 120          | 120 | 180 | $20.7 \times 10^6$           | $4.6 \times 10^5$               |
|  |                    |             | 3                  | 90           | 270 | 194,400                      | 1,440                           | 180          | 180 | 270 | $70 \times 10^6$             | $1.04 \times 10^6$              |

Note: Assumes linear elements, approximately 10 elements/wavelength, 8 variables/node.  
 2D Solution -  $N_o = 8MN$ ,  $B = 2(8M)$ . 3D Solution -  $N_o = 8LMN$ ,  $B = 4(8LM)$ .

The global matrix orders ( $N_o$ ) and bandwidths (B) resulting from these discretization meshes are also shown in Table I. For the 2-D model, matrix orders range from 6,400 to 194,400; while for the 3-D model, orders range from 0.5 million to 70 million. It should be emphasized that the wide range spanned by these numerical values represent relatively small changes in duct physical dimensions, mean-flow characteristics and source frequency. Matrix bandwidths span a similarly wide range, from 320 to 1,440 in the 2-D case, and 51,200 to 1.04 million in the 3-D case.

### Computational Effort

Consider matrices of the dimensions described above relative to the computational effort required to generate and solve them. Absolute evaluation of computational effort is extremely complicated and is dependent upon

computer and compiler characteristics as well as on programming techniques. The approach taken in this determination is based upon an approximate evaluation of operation counts followed by scaling of central processor unit (CPU) time taken from a baseline optimized FORTRAN program<sup>(1)</sup> on a CDC Cyber 175 computer.

The basic finite element process involves two distinct phases, i.e., assembling the global matrix, and solving the global matrix.

In the case of boundary condition optimization, this process is augmented by a third phase; specifically, use of the current solution to generate an increment in boundary conditions. The complete process is repeated until convergence to an optimum set of boundary conditions is obtained.

By defining an operation as one addition followed by one multiplication, we may estimate operation counts as follows:

- Global matrix assembly - For each element, these procedures involve numerical integration of the finite element approximation to the differential equations, over the domain. The procedures are proportional to the number of finite elements and are thus proportional to the number of degrees of freedom ( $N_0$ ). The constant of proportionality is typically of the order  $10^2$  for 2-D models and  $10^3$  for 3-D models.
- Global matrix solution - Solution of a full matrix is most commonly achieved by a direct method such as Gaussian elimination which requires approximately  $1/3 N_0^3$  operations. From Table I, however, it may be seen that the global matrix is not full, but possesses a bandwidth at least one order of magnitude (sometimes 2 orders of magnitude in the 2-D case) less than the global matrix order. In these circumstances, Gaussian elimination may be carried out in approximately  $\frac{1}{2} N_0 B^2$  operations.

- Boundary condition increment generation - Any of the standard optimization methods may be used; Davidon-Fletcher-Powell, Simplex, or Broyden-Fletcher-Shanno, for example. The computational effort in implementing any of these methods is negligible and the impact of this phase of the process may be ignored. It is important to note, however, that the efficiency of the method used has a vital bearing on overall computational time. The most efficient method requires the minimum iterations for convergence to an optimum liner configuration.

The operation counts given above may now be scaled by CPU times (Table II) from a baseline 2-D finite element model of sound propagation in an aeroengine duct. This model uses a numerically integrated bilinear isoparametric serendipity element and a block tridiagonal solution scheme specifically coded for problem sparsity. Figures 1 and 2 show that a linear relationship does hold between the theoretical operation counts given above the actual CPU times for assembly and solution phases.

Table II

| Computing Times for Baseline Model |     |                        |     |                   |                   |                               |
|------------------------------------|-----|------------------------|-----|-------------------|-------------------|-------------------------------|
| M                                  | N   | Mat Order<br>( $N_o$ ) | B   | CPU Times for     |                   | $1/4 N_o B^2$<br>(Operations) |
|                                    |     |                        |     | Assembly<br>(Sec) | Solution<br>(Sec) |                               |
| 40                                 | 110 | 36,408                 | 656 | 330               | 2,060             | $.39 \times 10^{10}$          |
| 25                                 | 59  | 12,480                 | 416 | 98                | 299               | $.55 \times 10^9$             |
| 12                                 | 30  | 3,224                  | 208 | 25                | 35                | $.35 \times 10^8$             |
| 20                                 | 200 | 33,768                 | 336 | 304               | 603               | $.95 \times 10^9$             |

Estimated computer times for the aeroengine inlet examples given in Table I, derived from the relations:

$$\text{CPU}_{\text{assembly}} = (.94 \times 10^{-2}) N_0 \text{ (secs)}$$

$$\text{CPU}_{\text{solution}} = (1.33 \times 10^{-7}) N_0 B^2 \text{ (secs)}$$

are given in Table III.

Table III

| Estimated CPU Times for Solution of Sound Propagation Equations in Typical Aeroengine Inlets |                 |          |                 |                |                |                |                   |  |
|--|-----------------|----------|-----------------|----------------|----------------|----------------|-------------------|--|
|  |                 |          |                 | CPU Times      |                |                |                   |  |
| Diameter (Meters)  | Length (Meters) | Mach No. | Frequency (kHz) | 2-D            |                | 3-D            |                   |  |
|  |                 |          |                 | Assembly (Sec) | Solution (Min) | Assembly (Min) | Solution (Hr)     |  |
| 1.5  | 1.5             | 0        | 1               | 6              | 1.5            | 8              | $4.9 \times 10^4$ |  |
|  |                 |          | 2               | 24             | 23             | 62             | $5.9 \times 10^6$ |  |
|  |                 |          | 3               | 54             | 118            | 216            |                   |  |
|  |                 | -.3      | 1               | 9              | 2.2            | 12             |                   |  |
|  |                 |          | 2               | 36             | 35             | 95             |                   |  |
|  |                 |          | 3               | 81             | 176            | 324            |                   |  |
| 2.2  | 2.2             | 0        | 1               | 14             | 7.4            | 27             |                   |  |
|  |                 |          | 2               | 54             | 118            | 216            |                   |  |
|  |                 |          | 3               | 122            | 596            | 730            |                   |  |
|  |                 | -.3      | 1               | 20             | 11             | 41             |                   |  |
|  |                 |          | 2               | 81             | 177            | 324            |                   |  |
|  |                 |          | 3               | 183            | 894            | 1,097          |                   |  |

NOTE: Assumes linear finite elements, approximately 10 elements/wavelength, 8 variables/node.

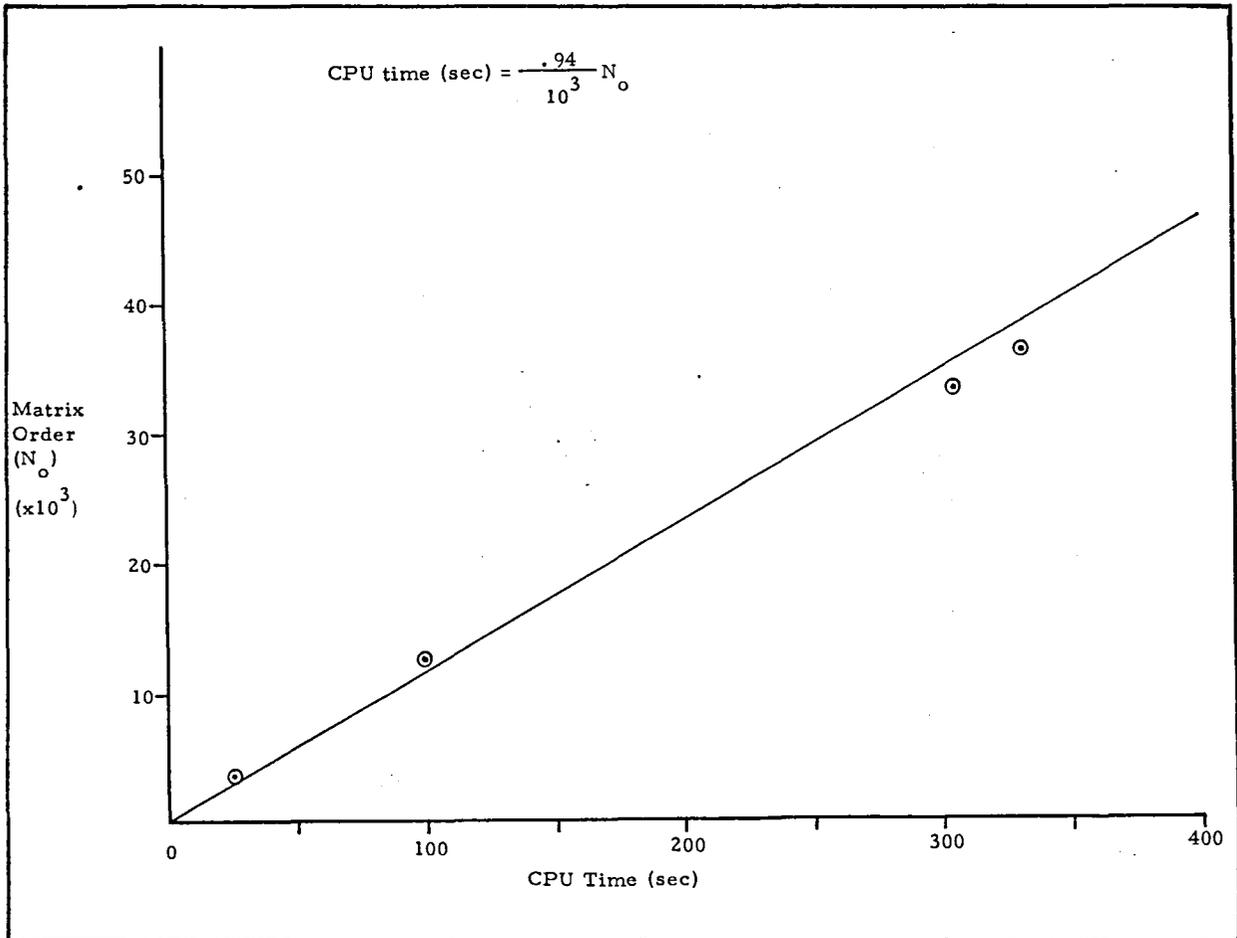


Figure 1. - Plot of theoretical operation counts ( $N_o$ ) versus actual matrix assembly times for baseline model.

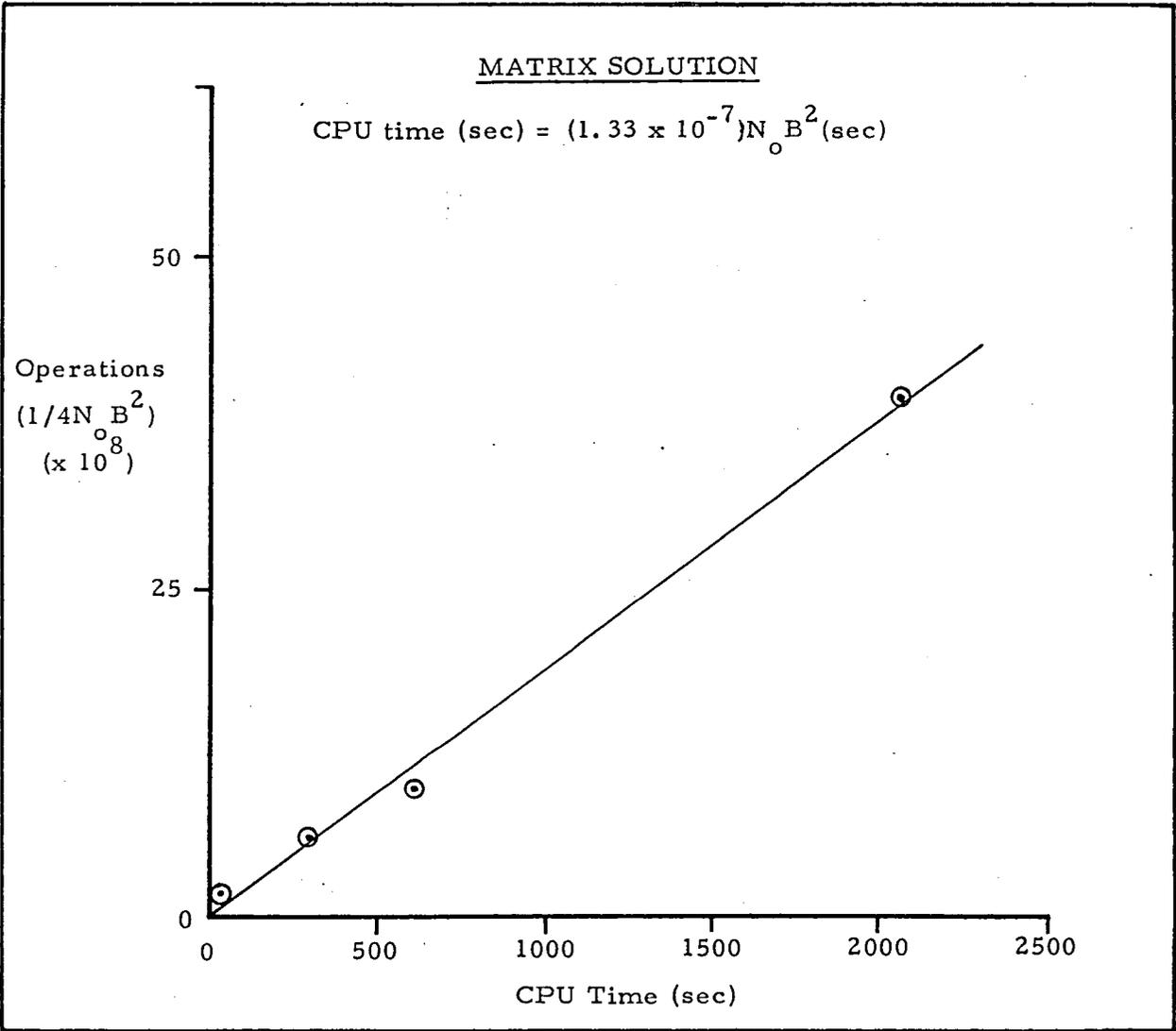


Figure 2. - Plot of theoretical operation counts ( $N_o B^2$ ) versus actual matrix solution times for baseline model.

CPU time estimates from Table III show that global matrix assembly times are substantially less than solution times. For a 2-D analysis, assembly times are measured in seconds while solution times range from a few minutes to several hours. From a cursory examination of solution times, it is evidently pointless to ever contemplate a 3-D analysis using the techniques of the baseline model since CPU times begin at 50,000 hours.

A 2-D analysis of a single case spans the range from 1.5 minutes, for a 1-kHz source and a zero Mach number in a duct 1.5 m in diameter and 1.5 m long, to 15 hours for a 3-kHz source with a flow Mach number of -0.3 in a duct 2.2 m in diameter and 2.2 m long.

Suppose we take one hour as a reasonable upper limit of CPU time per case, then a 2-D analysis of the 1.5-m duct, up to a source frequency between 2 kHz and 3 kHz is feasible. Alternately, for the 2.2-m duct, the feasible upper frequency range lies somewhere between 1 kHz and 2 kHz.

Consider the process of duct liner optimization. Depending upon the number of degrees of freedom in the duct liner specification, numerical optimization requires repetition of the analysis typically between 20 and 200 times. As shown in reference 2, it is not necessary to multiply CPU times given in Table III by this factor, since alternate direct decomposition techniques such as the boundary exclusive decomposition method are available.

#### Alternate Approaches for 3-D Analysis

Since the methods used in 2-D models are clearly inadequate for a 3-D analysis, new approaches are necessary. It has been demonstrated that the vast increases in computer time from a 2-D to a 3-D analysis come from attempting a direct solution to a matrix equation whose bandwidth has increased by two or three orders of magnitude.

The alternative to a direct method is an iterative method. Iterative solution algorithms such as Jacobi, Gauss-Seidell and Successive Over-

Relaxation (SOR) differ from Gaussian elimination in one important respect. Whereas, Gaussian elimination generates "fill," i.e., large quantities of intermediate numbers which all require processing, these algorithms always work only on the original matrix.

To provide an indication of the potential savings which may be gained from implementation of these algorithms, consider the original number of nonzeros per row in the global matrix for the examples given in Table I. For all 2-D discretizations, these number just 72; while for all 3-D discretizations, 216 nonzeros exist per row. Comparing these numbers with the matrix bandwidths shown in Table I, the original degree of sparsity is evident. In the process of Gaussian elimination, the entire bandwidth is filled in the solution process.

Two problems exist with iterative solutions, however. They are as follows:

- A positive definite global matrix is required for convergence to be guaranteed.
- The number of iterations required for adequate convergence may be very small or very large depending upon the characteristics of the matrix.

The Galerkin method of formulating finite element equations does not give a positive definite global matrix. The least squares method, however, does give a symmetric positive definite global matrix but frequently the least squares process yields equations that are poorly conditioned.

Estimated CPU times for one iteration of each of the typical inlet examples are given in Table IV.

Table IV

| Estimated CPU Times/Iteration for Iterative Solution<br>of Sound Propagation equations in an Aeroengine Inlet |                    |             |                    |                       |              |
|---|--------------------|-------------|--------------------|-----------------------|--------------|
| Diameter<br>(Meters)  | Length<br>(Meters) | Mach<br>No. | Frequency<br>(kHz) | 2-D<br>(Min.)         | 3-D<br>(Min) |
| 1.5   | 1.5                | 0           | 1                  | $1.02 \times 10^{-3}$ | .25          |
|   |                    |             | 2                  | $3.99 \times 10^{-3}$ | 1.9          |
|   |                    |             | 3                  | $9.09 \times 10^{-3}$ | 6.6          |
|   |                    | -.3         | 1                  | $1.53 \times 10^{-3}$ | .37          |
|   |                    |             | 2                  | $6.12 \times 10^{-3}$ | 2.9          |
|   |                    |             | 3                  | $1.37 \times 10^{-2}$ | 9.9          |
| 2.2   | 2.2                | 0           | 1                  | $2.30 \times 10^{-3}$ | .81          |
|   |                    |             | 2                  | $9.19 \times 10^{-3}$ | 6.6          |
|   |                    |             | 3                  | $2.07 \times 10^{-2}$ | 22           |
|   |                    | -.3         | 1                  | $3.44 \times 10^{-3}$ | 1.24         |
|   |                    |             | 2                  | $1.38 \times 10^{-2}$ | 9.9          |
|   |                    |             | 3                  | $3.10 \times 10^{-2}$ | 33.5         |

The critical question regarding the feasibility of a 3-D analysis hinges on the convergence rate. That is, how many iterations are required for convergence. Since convergence rate is problem dependent, the only way to determine it is by first assembling the global matrix equation. Thus, for practical purposes, the only way to determine the feasibility of the approach is by trying it.

#### DERIVATION

Derivation of the equations governing linearized acoustic motion for the 2-D case, is shown in reference 1. Extension of this derivation to 3-D is readily performed and will not be given here. The four equations are:

$$\begin{aligned} \omega u + u \frac{\partial \bar{u}}{\partial x} + \bar{u} \frac{\partial u}{\partial x} + \bar{v} \frac{\partial u}{\partial y} + \bar{w} \frac{\partial u}{\partial z} + v \frac{\partial \bar{u}}{\partial y} + w \frac{\partial \bar{u}}{\partial z} \\ + \frac{p}{\rho c^2} (\bar{u} \frac{\partial \bar{u}}{\partial x} + \bar{v} \frac{\partial \bar{u}}{\partial y} + \bar{w} \frac{\partial \bar{u}}{\partial z}) + \frac{1}{\rho} \frac{\partial p}{\partial x} = 0 \end{aligned} \quad (1)$$

$$\begin{aligned} u \frac{\partial \bar{v}}{\partial x} + \omega v + v \frac{\partial \bar{v}}{\partial y} + \bar{u} \frac{\partial v}{\partial x} + \bar{v} \frac{\partial v}{\partial y} + \bar{w} \frac{\partial v}{\partial z} + w \frac{\partial \bar{v}}{\partial z} \\ + \frac{p}{\rho c^2} (\bar{u} \frac{\partial \bar{v}}{\partial x} + \bar{v} \frac{\partial \bar{v}}{\partial y} + \bar{w} \frac{\partial \bar{v}}{\partial z}) + \frac{1}{\rho} \frac{\partial p}{\partial y} = 0 \end{aligned} \quad (2)$$

$$\begin{aligned} u \frac{\partial \bar{w}}{\partial x} + v \frac{\partial \bar{w}}{\partial y} + \omega w + w \frac{\partial \bar{w}}{\partial z} + \bar{u} \frac{\partial w}{\partial x} + \bar{v} \frac{\partial w}{\partial y} + \bar{w} \frac{\partial w}{\partial z} \\ + \frac{p}{\rho c^2} (\bar{u} \frac{\partial \bar{w}}{\partial x} + \bar{v} \frac{\partial \bar{w}}{\partial y} + \bar{w} \frac{\partial \bar{w}}{\partial z}) + \frac{1}{\rho} \frac{\partial p}{\partial z} = 0 \end{aligned} \quad (3)$$

$$\begin{aligned} \left( \frac{u}{\rho} \frac{\partial \bar{p}}{\partial x} + \frac{\partial u}{\partial x} \right) + \left( \frac{v}{\rho} \frac{\partial \bar{p}}{\partial y} + \frac{\partial v}{\partial y} \right) + \left( \frac{w}{\rho} \frac{\partial \bar{p}}{\partial z} + \frac{\partial w}{\partial z} \right) \\ + \frac{1}{\rho c^2} [\omega p + \bar{u} \frac{\partial p}{\partial x} + \bar{v} \frac{\partial p}{\partial y} + \bar{w} \frac{\partial p}{\partial z} + p \left( \frac{\partial \bar{u}}{\partial x} \right. \\ \left. + \frac{\partial \bar{v}}{\partial y} + \frac{\partial \bar{w}}{\partial z} \right)] = 0 \end{aligned} \quad (4)$$

where  $(u, v, w)$  and  $(\bar{u}, \bar{v}, \bar{w})$  are the  $(x, y, z)$  components of acoustic and mean-flow velocities, respectively,  $p$  is the acoustic pressure, and  $\bar{\rho}$  the mean density. Acoustic source frequency is  $\omega$  and  $c$  is the local speed of sound.

A 3-D cubic serendipity element <sup>(3)</sup> was selected for the analysis. This element shown in Figure 3 has a total of 32 nodes and when applied to solution of the above equations with four variables per node (u,v,w,p) yields a local element matrix of order 256.

The principal reason for adoption of this element over a linear element was to take full advantage of the computational speed increases offered by a vector computer. With iterative solution algorithms on a vector computer (such as the CDC STAR 100), vector lengths for global matrix solution using this element lie between 256 and 1080. These vectors are sufficiently long to minimize the effect of "pipeline" processor startup time on this machine.

Consider the volume of space enclosed by the cube shown in Figure 3. Within this cube, a parameter  $\phi(z,y,z)$  is completely defined by its values  $\{\phi\}^e$  at the nodes through the relation:

$$\begin{aligned} \phi &= (N) \{\phi\}^e \\ &= (N_1, N_2, \dots, N_{32}) \left\{ \begin{array}{c} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{32} \end{array} \right\} \end{aligned} \quad (5)$$

Functionals  ${}^1N_i$  where  $i = 1, 4, 9, 12, 21, 24, 29, 32$  take the form:

$${}^1N_i = \frac{1}{64} (1 + \xi_0) (1 + \eta_0) (1 + \zeta_0) \{9(\xi^2 + \zeta^2) - 19\}$$

Functionals  ${}^2N_i$  where  $i = 2, 3, 10, 11, 22, 23, 30, 31$  take the form:

$${}^2N_i = \frac{9}{64} (1 - \zeta^2) (1 + 9\xi_0) (1 + \eta_0) (1 + \zeta_0)$$

Functionals  ${}^3N_i$  where  $i = 5, 6, 7, 8, 25, 26, 27, 28$  take the form:

$${}^3N_i = \frac{9}{64} (1 - \eta^2) (1 + 9\eta_0) (1 + \xi_0) (1 + \zeta_0)$$

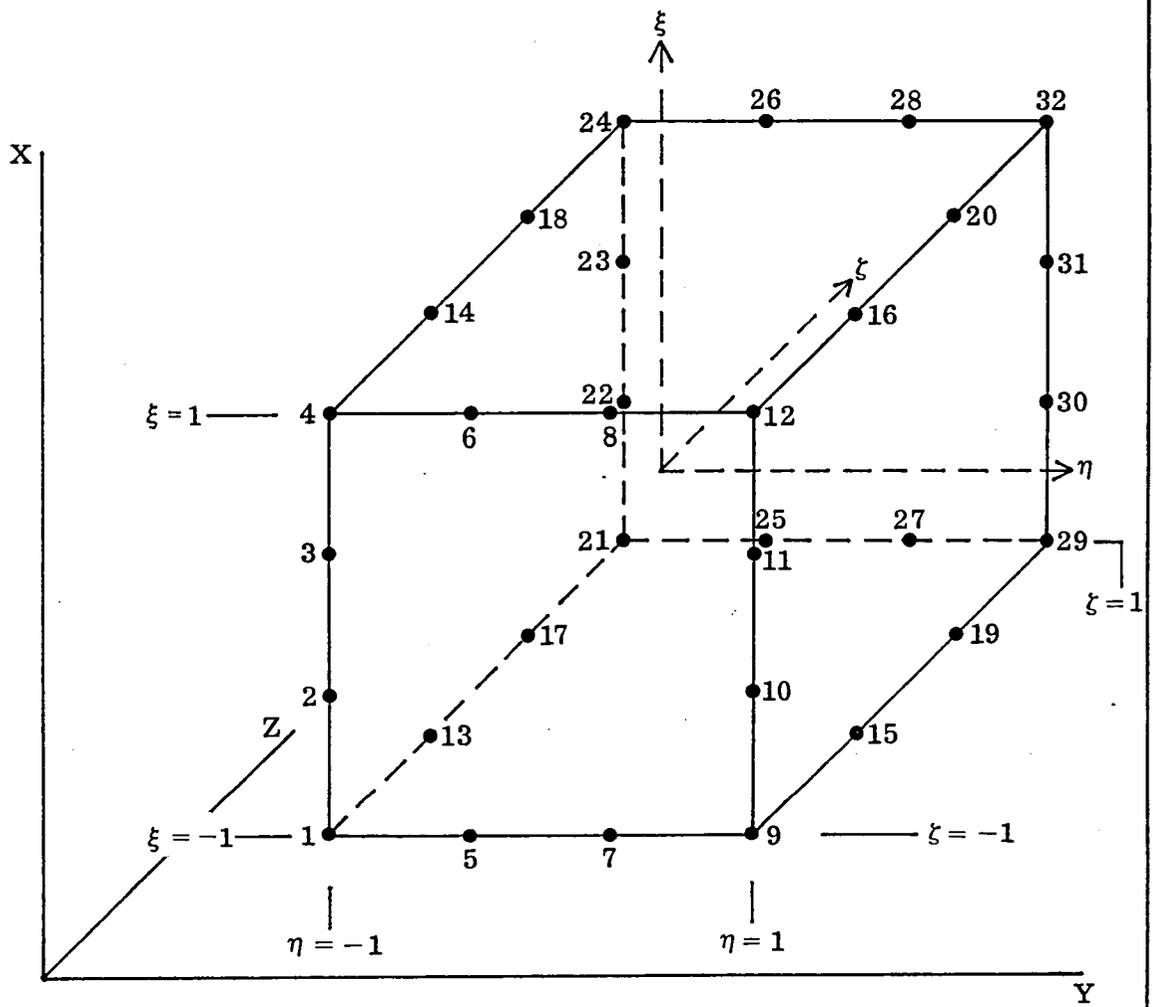


Figure 3. - 3-D cubic serendipity element.

Functionals  ${}^4N_i$  where  $i = 13, 14, 15, 16, 17, 18, 19, 20$  take the form:

$${}^4N_i = \frac{9}{64} (1 - \zeta^2) (1 + 9\zeta_0) (1 + \xi_0) (1 + \eta_0)$$

In the above expressions,

$$\xi_0 = \xi \xi_i, \quad \eta_0 = \eta \eta_i, \quad \zeta = \zeta \zeta_i$$

where  $(\xi_i, \eta_i, \zeta_i)$  represent the nondimensional coordinates of the  $i$ th node and may assume values  $\pm 1$  or  $\pm 1/3$ .

Suppose  $(u, v, w, p)$  and  $(\bar{u}, \bar{v}, \bar{w}, \bar{p})$  are given by relations of the form in equation (5), then these functionals can be substituted into equations (1) through (4).

After carrying out an isoparametric mapping of the element equations (1) through (4), they are squared and summed. The variation with respect to each nodal variable is taken to yield a set of 256 equations. These are integrated numerically over the volume to give the local element matrix. A boundary condition matrix is compiled by squaring and summing the boundary condition equations integrated over element boundary surfaces and taking variations with respect to nodal variables. The boundary condition matrix is multiplied by a constant, determined by best numerical conditioning, and summed with the element matrix. (The Least Squares Method is described in detail in references 4 and 5.)

## PROGRAM ORGANIZATION

The computer program to perform a 3-D duct acoustic analysis was designed in modular form. The principal modules and their functions are given in the Appendix.

## SOLUTION TECHNIQUE EVALUATION

### Checkout with Direct Equation Solver

To verify the element formulation, boundary condition insertion, and program coding, test cases were compiled. The first test case consisted simply of a single element, giving a system of equations of order 256, with straight sides, i.e., a cube. The x-y plane at  $z = 0$  was specified as a source plane, and at a  $z = 1$  as a radiation plane with an impedance of  $\bar{\rho}c$ . All other walls were hard (impedance =  $\infty$ ). The source frequency was chosen to be sufficiently low so that only the plane-wave mode was cut on, mean flow was zero.

A direct-solution library subroutine was used to solve the resulting system of equations. This solution coincided closely with the analytically predicted, plane-wave propagation. Conditioning of the set of equations was evaluated by investigating the range of eigenvalues. This range (a factor of  $10^9$ ) indicated poor conditioning. Modification of equation weighting factors in the element least squares formulation and the weighting factors in the boundary condition integrals gave only marginal improvement in conditioning.

### Jacobi and SOR Iterative Methods

Writing the matrix equation as  $AX = b$ , we may split the global matrix (A) into a lower triangle (L), a diagonal (D) and an upper triangle (U),

$$A = D (L + I + U)$$

Jacobi's iterative method is then written as,

$$X^{(k+1)} = (L + U) X^{(k)} + D^{-1}b$$

where  $X^{(k)}$  is the previous estimate for the solution vector and  $X^{(k+1)}$  is the new estimate.

Similarly, the SOR method is

$$X^{(k+1)} = (I + \Omega L)^{-1} \{[(1 - \Omega)U] X^{(k)} + D^{-1}b\}$$

where  $\Omega$ , the over-relaxation parameter, lies between 1 and 2.

For these methods to converge, it is necessary that all eigenvalues of their iteration matrices, viz,

$$-(L + U) \text{ and } (I + \Omega L)^{-1} \{(1 - \Omega)I - \Omega U\}$$

respectively, should be less than unity<sup>(6)</sup>.

Programs were written to construct these iteration matrices, obtain their eigenvalues, and implement the two solution schemes for the test problem already analyzed using the direct solver.

For the Jacobi method, the largest eigenvalues were greater than unity for the test problem and as expected, solutions diverged.

For the SOR method, the largest eigenvalues were less than, but extremely close to one ( $\sim 0.99998$ ) indicating slow convergence. In practice, convergence for this method was found to be so slow that its use was impractical. No noticeable improvement was found by varying  $\Omega$ .

### Conjugate Gradient Method

The conjugate gradient method is unique among iterative methods since, in the limit, it becomes a direct method. That is, for an  $n$ th order system if rounding errors are ignored, the  $n$ th iteration will give the exact solution. Often, however, a good approximation to the exact solution is obtained for an iteration well below  $n$ .

The method was first published in 1952<sup>(7)</sup> but did not receive much attention until recently when it was shown that various preconditionings can

significantly improve its rate of convergence. In most cases, it is now significantly faster than SOR.

The theory of the method is given in references 7 through 9, and only the mechanics of the method will be described here.

Given a system  $AX = B$  with a positive definite  $n \times n$  matrix  $A$  and a starting vector  $X_0$  compute

$$r_0 = B - AX_0 \quad \text{and set } p_0 = r_0.$$

For  $i = 0, 1, 2, \dots, n-1$  compute

$$a_i = \frac{\|r_i\|_2^2}{p_i^T A p_i}$$

$$X_{i+1} = X_i + a_i p_i$$

$$r_{i+1} = r_i - a_i A p_i$$

$$b_i = \frac{\|r_{i+1}\|_2^2}{\|r_i\|_2^2}$$

$$p_{i+1} = r_{i+1} + b_i p_i$$

$X_n$  will be the solution of the linear system if rounding errors are neglected. Often a good approximation to the solution is obtained by some  $X_i$  for  $i \ll n$ .

Results from application of the conjugate gradient method to the test problem showed high initial promise. Without preconditioning, solution of the test problem (order 256) was obtained in 140 iterations. With symmetric scaling, the number of iterations for solution was decreased to 50, using the same arbitrary starting vector.

General convergence trends were then investigated with the result that the total number of iterations for convergence was found to be dependent upon several factors:

- The variance between the first estimate and the actual solution
- The number of finite elements per wavelength
- The conditioning of the global matrix
- The order of the global matrix

In these investigations, the Euclidean norm of the residual vector ( $\|r_i\|_2$ ) was used to assess convergence. For a typical first estimate,  $\|r_0\|_2$  was of the order of  $10^3$ ; while for a good first estimate,  $\|r_0\|_2$  was about unity. Convergence appeared to reach a plateau at values of  $\|r_i\|_2$  between  $10^{-4}$  and  $10^{-5}$  with no further improvement obtainable regardless of the number of subsequent iterations.

Dependence of the number of iterations for convergence on the accuracy of the first estimate was expected and is a normal characteristic of iterative methods. Dependence on the number of finite elements per wavelength was surprising. After further investigation, it was found that varying the number of finite elements per wavelength affects the conditioning of the global matrix, and that the fewer finite elements allowed per wavelength, the worse the conditioning became.

It had been hoped that a firm trend could be established giving the number of iterations required for convergence versus the order of the global matrix, but it was discovered that under some circumstances the Euclidean norm of the residual vector ( $\|r_i\|_2$ ) would decrease in the usual inverse exponential manner until a low value of the error was reached (about  $10^{-6}$ ). However, the solution differed from a direct solution by unacceptably

large margins. Apparently the equations may become sufficiently ill-conditioned that solution accuracy is degraded by a significant degree.

This ill-conditioning may be seen from the plots in Figure 4. The analytical plane-wave solution is compared with finite element solutions for one through four finite elements per acoustic wavelength. In any one configuration, the global finite element matrix equation is solved in two different ways: first, using a direct method and second, using the preconditioned conjugate gradient method. In the case of one finite element per wavelength, the two solutions not only differ substantially from the analytical plane-wave solution, but also differ substantially from each other. In the cases of two and three elements per wavelength, the finite element solutions fall much closer to the analytical plane-wave solution; while for the case of four elements per wavelength, the three solutions are coincident. These results indicate that the finite element global matrix equation becomes increasingly well conditioned, the more elements allowed per wavelength.

The behavior of the norm of the residual vector  $\|r_i\|_2$ , versus iteration number for the cases in Figure 4, is illustrated in Figure 5. Here, the error norm decreases in a similar manner to similar final values in each case. This may be erroneously interpreted as indicating that the conjugate gradient solution algorithm has solved all four problems to the same degree of accuracy.

The first estimate solution vector in each case is the analytical plane-wave solution. Note that the error norm  $\|r_i\|_2$  decreases very rapidly in all cases and that the total number of iterations is substantially less than the number of degrees of freedom.

An additional problem was the unexpectedly large number of iterations required to reach an acceptably low value of the error norm when the first estimate solution vector differed significantly from the true solution. For example, the case presented in Figures 6 and 7 uses a large number of elements per wavelength (18) which should give a well conditioned matrix.

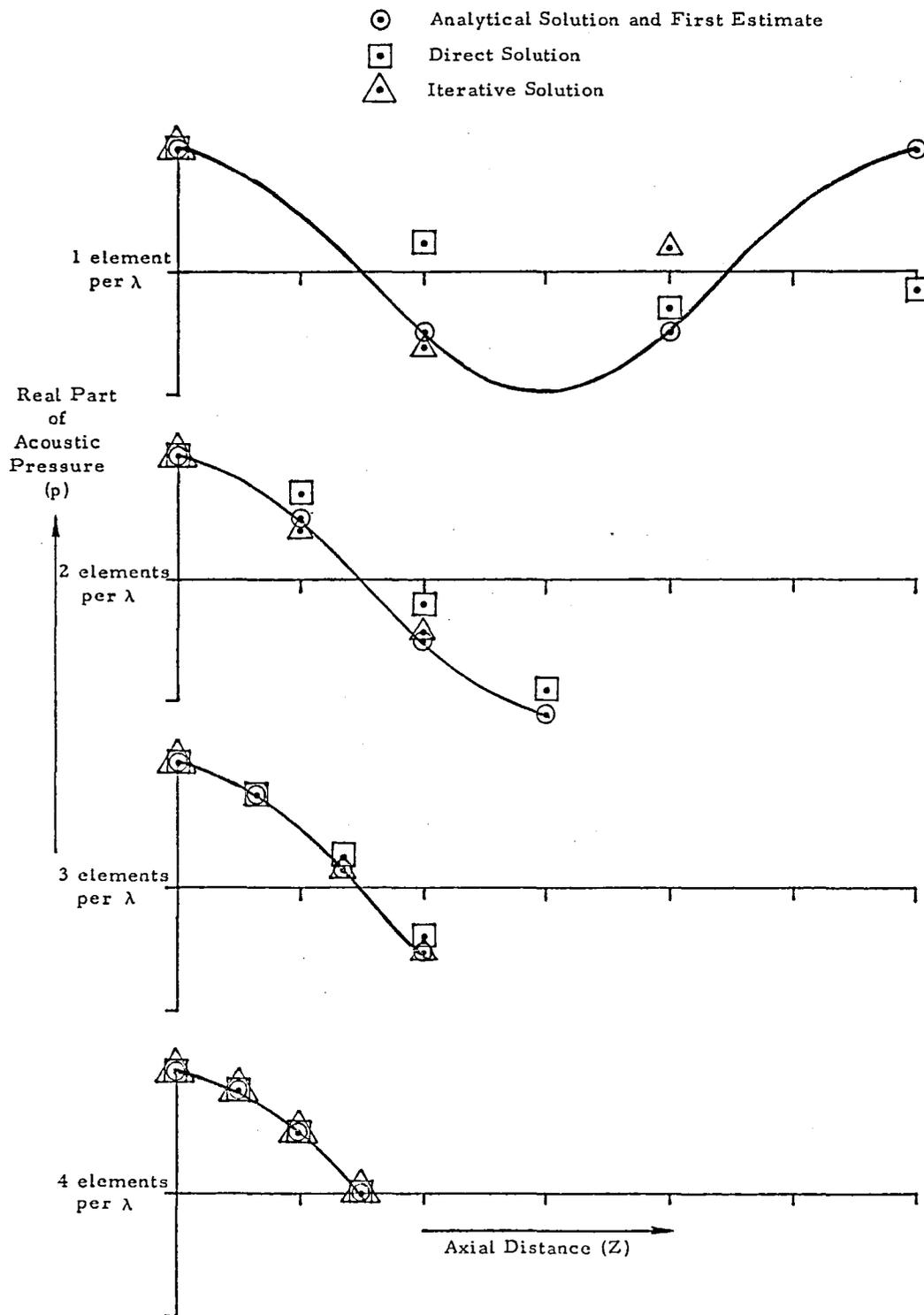


Figure 4. - Effect of the number of finite elements per wavelength ( $\lambda$ ) on solution accuracy.

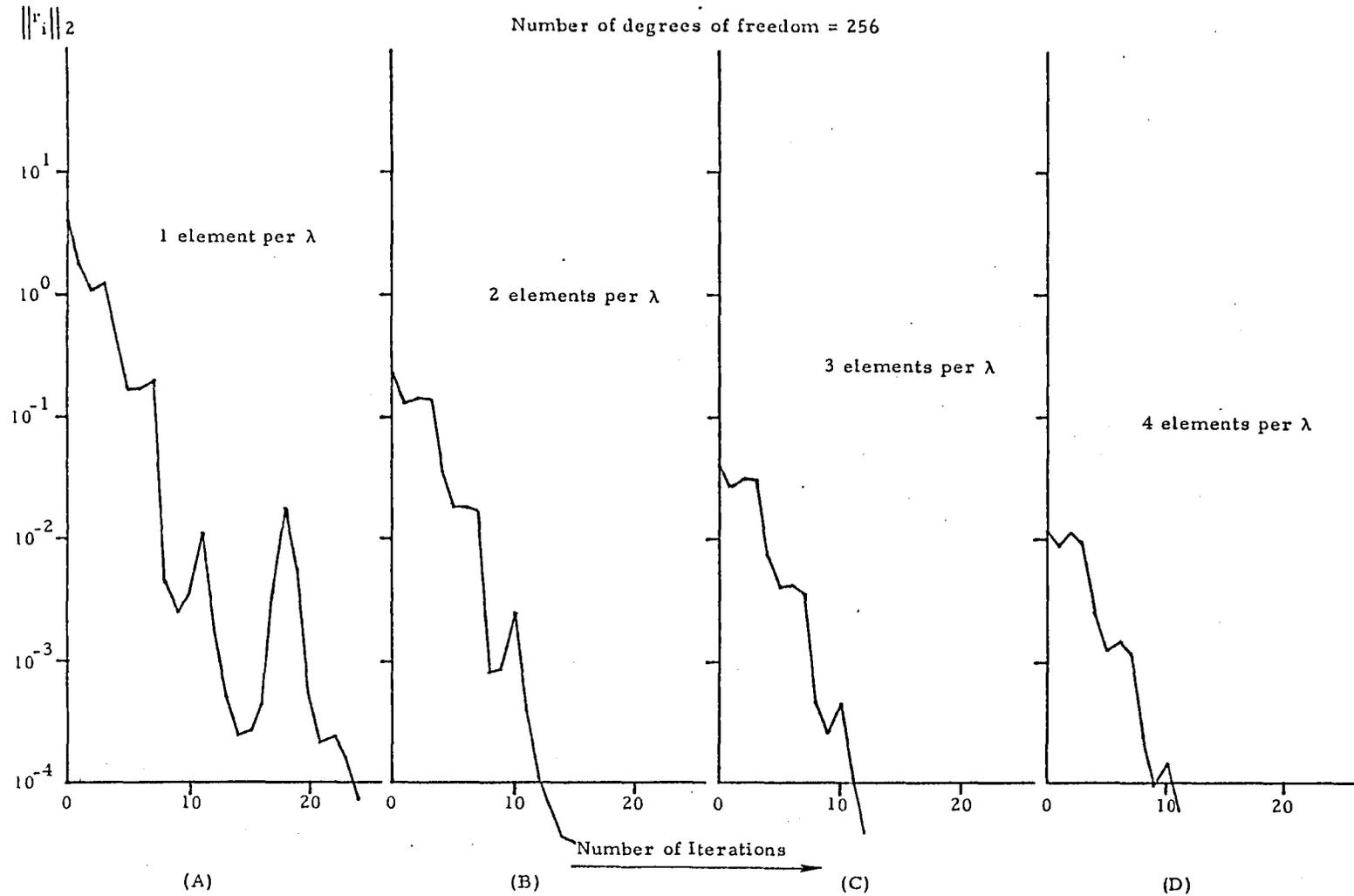


Figure 5. - Plots of Euclidean norm of residual vector  $\|r_i\|_2$  versus iteration number for 1 through 4 finite elements per wavelength ( $\lambda$ ).

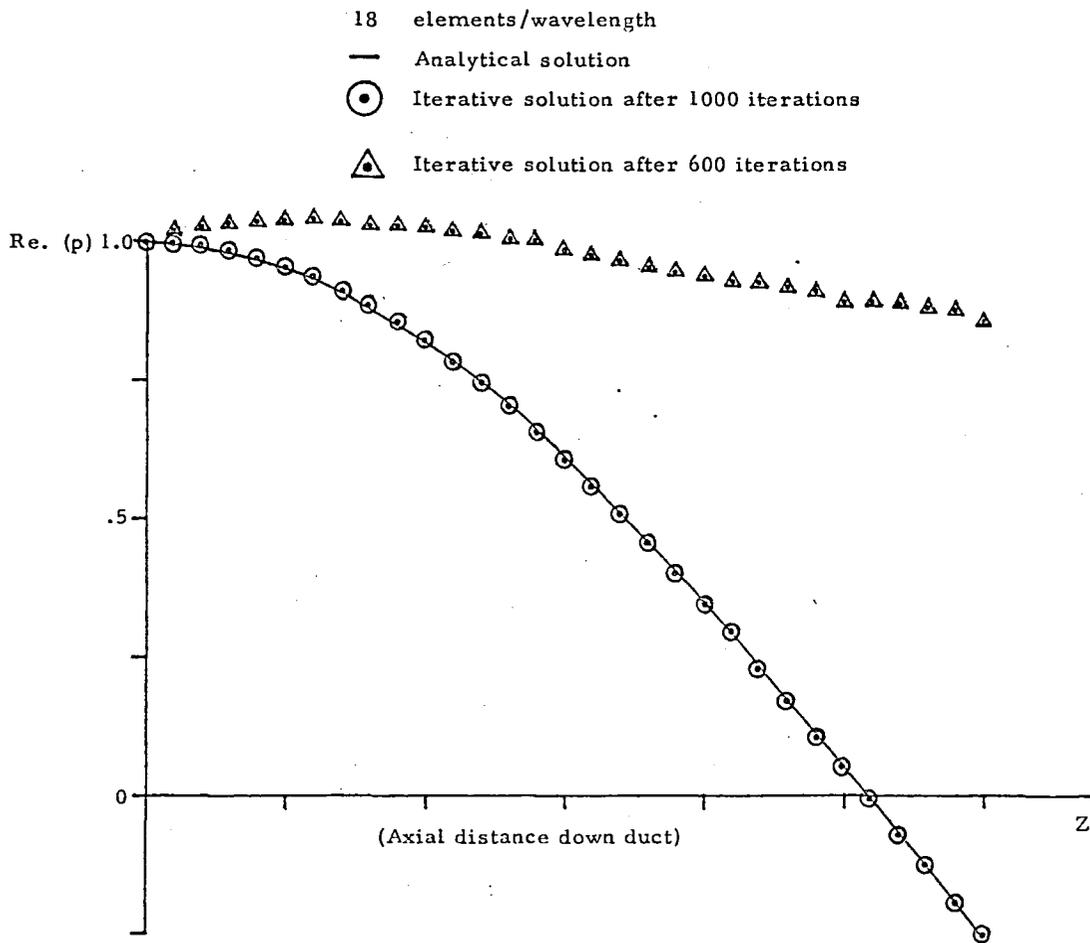


Figure 6. - Comparison of finite element conjugate gradient/ solution with analytical plane wave solution for poor first estimate solution vector.

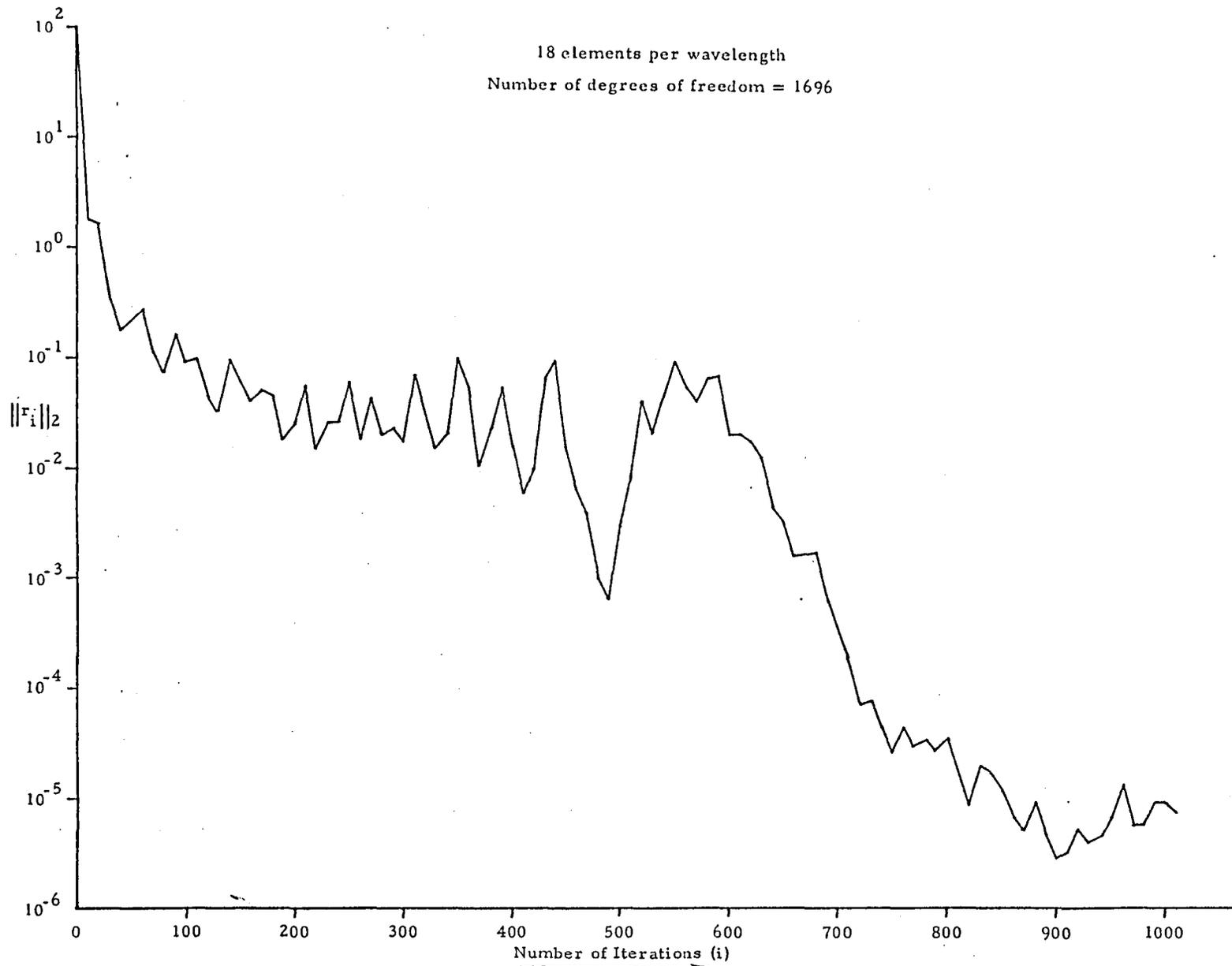


Figure 7. - Plot of  $\|r_i\|_2$  versus iteration number for poor first estimate of solution vector.

The first estimate of the solution vector was chosen to be zero except for the real pressure components which were all assumed to be unity.

The total system of equations numbered 1696. After 600 iterations the norm  $\|r_{600}\|_2$  was at a value of about  $10^{-2}$  (Figure 7) and the solution differed substantially from the correct solution. Only after 800 - 1000 iterations, the error norm had decreased to sufficiently low values of  $10^{-4}$  to  $10^{-5}$ . At these values, the solution became an accurate copy of the analytical plane-wave solution (Figures 6 and 7).

As a final exercise, some larger finite element meshes were run in order to assess typical computer resources required,

e.g.,

Mesh - 4 x 4 x 10 elements

Degrees of Freedom - 13,240

Computer times on the C.D.C. Star 100 were as follows:

Global Matrix Assembly - 312 wall clock mins.

- 76 CPU clock mins

Conjugate Gradient Solution

- 35 wall clock secs/iteration

- 5 CPU clock secs/iteration

Using four elements per wavelength with a good first estimate solution vector,  $\|r_i\|_2$  decreased from  $4.5 \times 10^{-2}$  to  $2.5 \times 10^{-4}$  in 50 iterations.

Although the performance of the model in this example was adequate, it was based on the premise of an extremely good first estimate solution vector. With a poor first estimate solution vector, several thousand iterations would be required to achieve a satisfactory solution.

The magnitude of computer time to perform several thousand iterations on this problem is several hours of CPU time and several days of wall clock time. When consideration is also made of the fact that 4 x 4 x 10 elements is

not a sufficiently large mesh to analyze even fairly trivial duct acoustic problems, continuation of the effort seems unrewarding.

It seems probable from the problems described earlier regarding poor conditioning of the global matrix when too few elements were allowed per wavelength, that severe changes in boundary conditions, element geometry or aerodynamic mean variables over a short distance, would contribute further ill-conditioning.

### CONCLUSIONS

This report describes an attempt to extend finite element modeling of duct acoustics to fully three-dimensional problems. The difficulties which arise when the successful 2-D techniques of Galerkin weighted residual formulation with a direct matrix solver are extended to 3-D, are described. The reasons for believing that an iterative matrix solver might be substantially quicker than a direct solver are indicated. Since all known iterative solvers require positive definite matrices, a least squares formulation was used instead of a Galerkin formulation.

The root cause of the difficulties encountered in application of iterative matrix solvers lies in the tendency of the least squares formulation to yield poorly conditioned matrices. Due to this fact, the Jacobi method diverged, Gauss-Seidell and SOR failed to converge at a measurable rate, while the conjugate gradient and even direct methods were uneven in their performance.

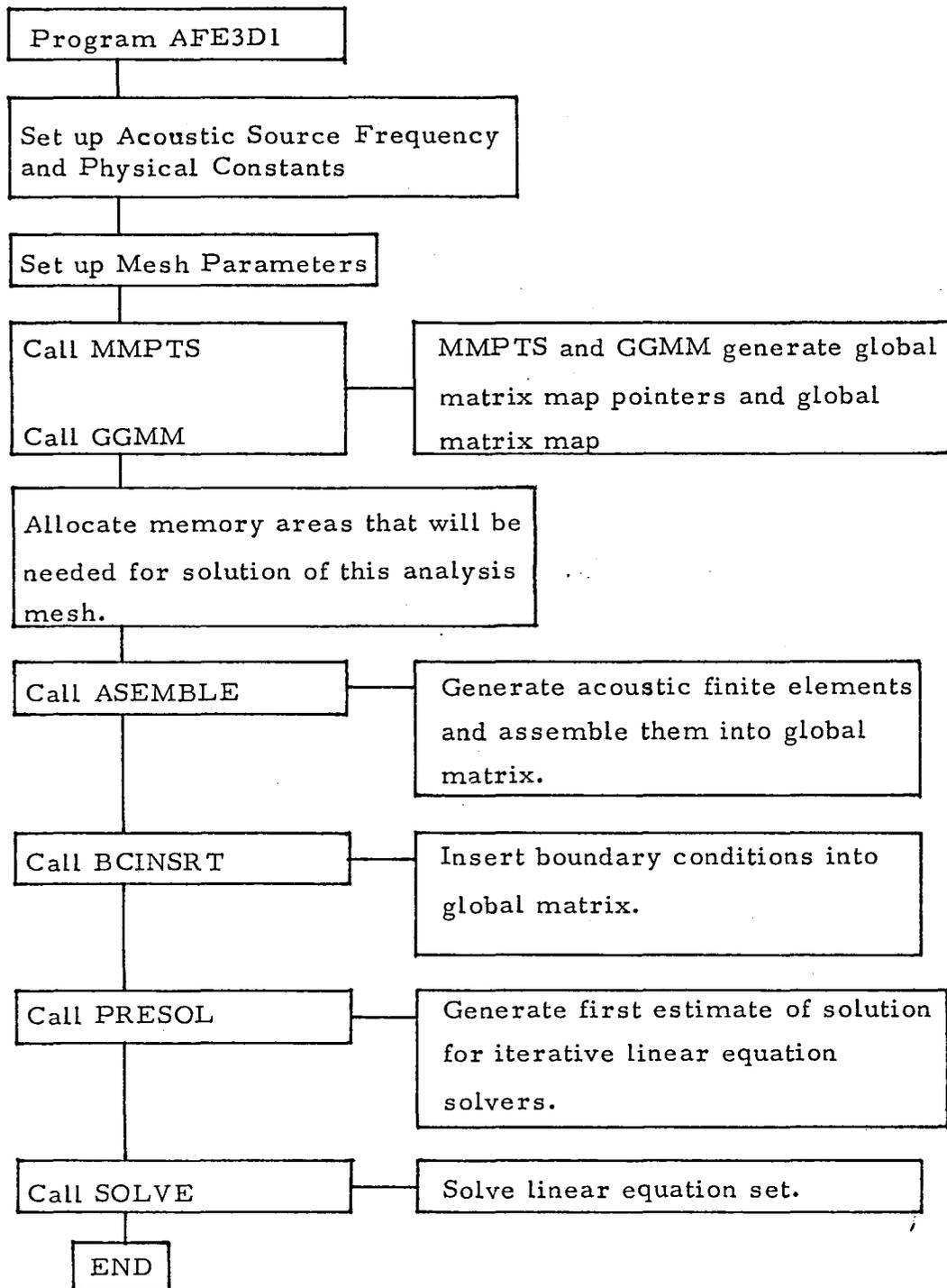
Of the iterative methods tried, there is no doubt that the preconditioned conjugate gradient method was superior. If a formulation to yield well conditioned positive-definite matrices could be found, it seems that this solver would give solutions within an acceptable amount of computing effort.

Although the goal of this work was not achieved, the steps taken and methods used represent a logical approach to the problem which should benefit future workers in this area.

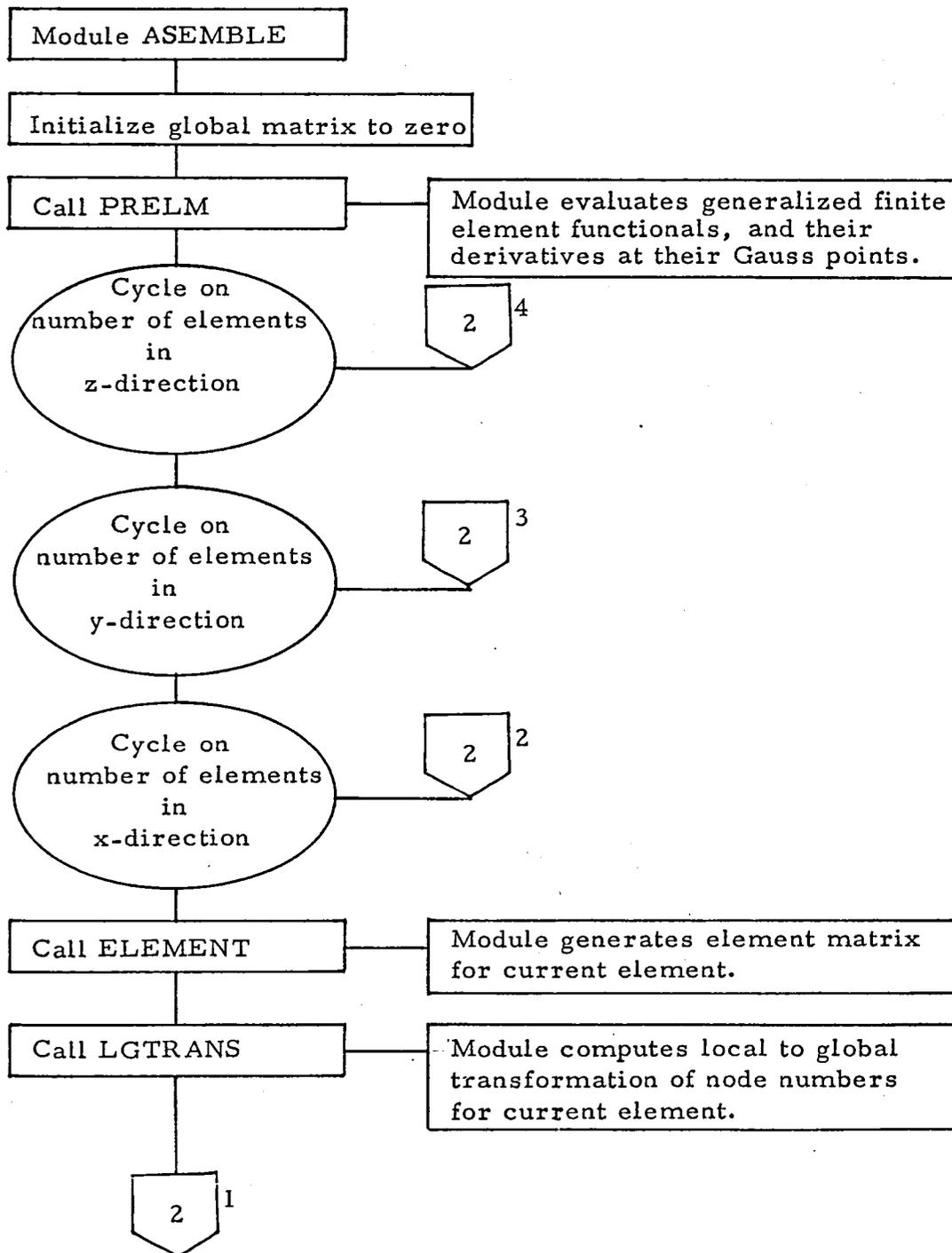
APPENDIX

Principal Modules and Functions  
of  
Computer Program  
AFE3D1

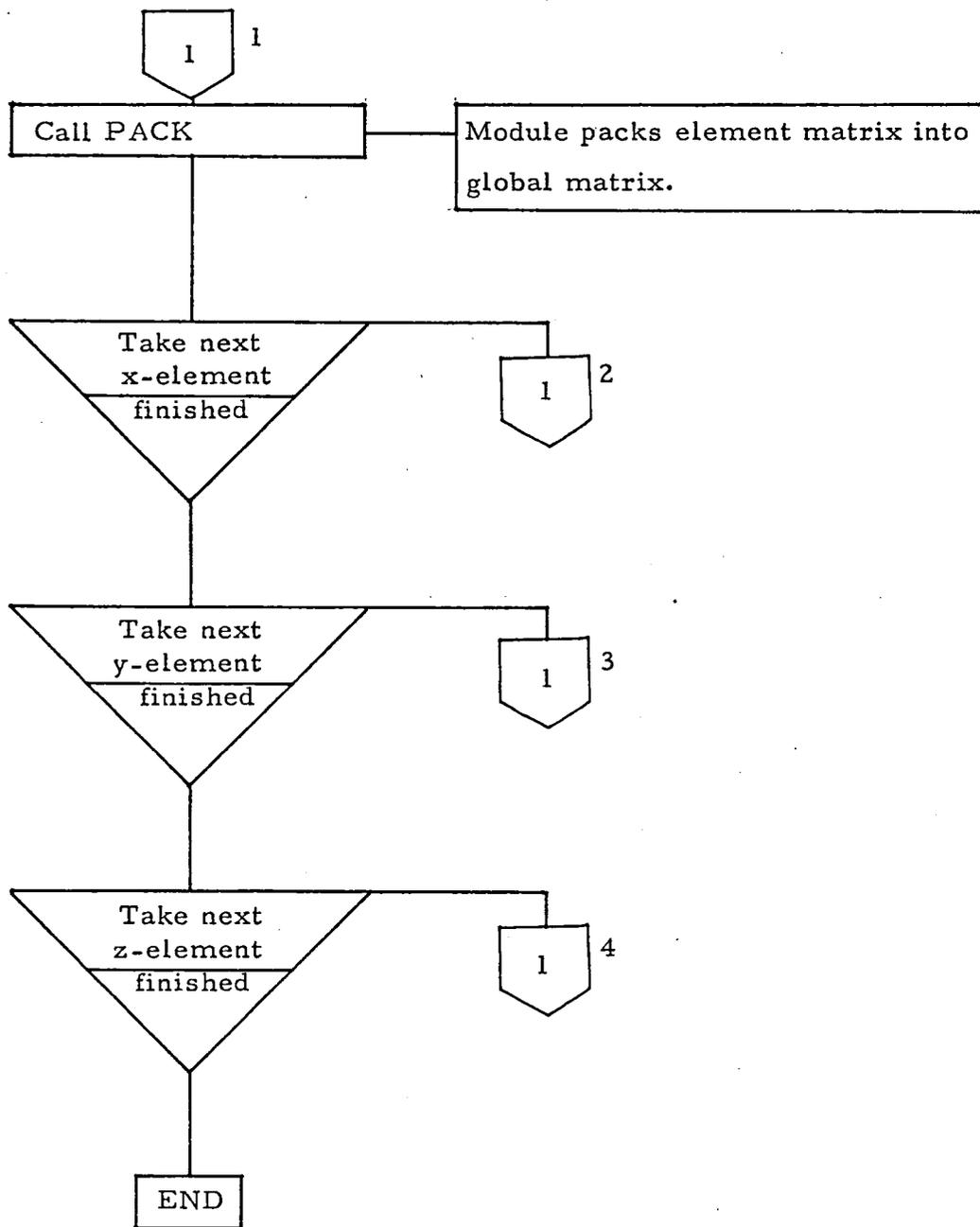
**This Page Intentionally Left Blank**



Within this structure, module ASEMBLE is the most significant high-level module. It has the following structure.



(Sheet 1 of 2)



**This Page Intentionally Left Blank**

## REFERENCES

1. Abrahamson, A. L., "A Finite Element Algorithm for Sound Propagation in Axisymmetric Ducts Containing Compressible Mean Flow," AIAA Paper 77-13-1, presented at AIAA 4th Aeroacoustics Conference, Atlanta, Georgia, October 1977.
2. Abrahamson, A. L., "Acoustic Duct Liner Optimization Using Finite Element," AIAA paper 79-0662, presented at AIAA 5th Aeroacoustics Conference, Seattle, Washington, March 1979.
3. Zienkiewicz, O. C., "The Finite Element Method in Engineering Science," McGraw-Hill, 1971.
4. Lynn, P. D., and Arya, S. K., "Use of the Least Squares Criterion in the Finite Element Formulation," Int. Journal for Num. Methods in Eng. Vol 6, 75-88, 1973.
5. Zienkiewicz, O. C., and Owen, D. R. J., "Least Square - Finite Element for Elasto-Static Problems. Use of "Reduced" Integration, Int Journal for Num. Methods in Eng. Vol 8, 341-358, 1974.
6. Dahlquist, G., and Bjorck, A., "Numerical Methods" Translated by Ned Anderson, Prentice-Hall, 1974.
7. Hestenes, M. R., and Stiefel, E., "Method of Conjugate Gradients for Solving Linear Systems," Journal of Research of Nat. Bureau of Standards, Vol 49, No. 6, December 1952.
8. Widlund, O., "A Lanczos Method for a Class of Non-Symmetric Systems of Linear Equations," Siam J. Numer. Analysis, Vol 15, No. 4, August 1978.
9. Schreiber, R. S., "Implementation of the Conjugate Gradient Method on the C.D.C. Star - 100." Applied Math Dept., Caltech, unnumbered and undated.

|   |  |  |  |   |            |
|---|--|--|--|---|------------|
| 1. Report No.<br>NASA CR-159359   |  | 2. Government Accession No.                          |  | 3. Recipient's Catalog No.                                      |            |
| 4. Title and Subtitle<br>A Feasibility Study of a 3-D Finite Element Solution Scheme for Aeroengine Duct Acoustics  |  |  |  | 5. Report Date<br>September 15, 1980                            |            |
|   |  |  |  | 6. Performing Organization Code                                 |            |
| 7. Author(s)<br>A. Louis Abrahamson   |  |  |  | 8. Performing Organization Report No.<br>51200                  |            |
| 9. Performing Organization Name and Address<br>WYLE LABORATORIES<br>3200 Magruder Boulevard<br>Hampton, Virginia 23666  |  |  |  | 10. Work Unit No.   |            |
|   |  |  |  | 11. Contract or Grant No.<br>NAS1-15291                         |            |
| 12. Sponsoring Agency Name and Address<br>National Aeronautics and Space Administration<br>Washington, D.C. 20546   |  |  |  | 13. Type of Report and Period Covered<br>Contractor; 3/78-11/79 |            |
|   |  |  |  | 14. Sponsoring Agency Code                                      |            |
| 15. Supplementary Notes<br>FINAL REPORT<br>Langley technical monitor: Dr. Harold C. Lester  |  |  |  |   |            |
| 16. Abstract<br>Development of 2-D finite element solution schemes for aeroengine duct acoustics was initiated in the mid-1970s and considerable success was achieved. Analysis of acoustic propagation in axisymmetric ducts with nonuniform flow are now possible and schemes for increasing computational efficiency are now to the stage where repeated analyses for duct liner optimization are feasible. The advantage from development of a 3-D model is the ability to analyze axial and circumferential liner segmentation simultaneously.<br><br>This paper examines the feasibility of a 3-D duct acoustics model using Galerkin or least squares element formulations combined with Gaussian elimination, successive over-relaxation, or conjugate gradient solution algorithms on conventional scalar computers and on a vector machine. A least squares element formulation combined with a conjugate gradient solver on a CDC Star vector computer initially appeared to have great promise, but severe difficulties were encountered with matrix ill-conditioning. These difficulties in conditioning rendered this technique impractical for realistic problems. |  |  |  |   |            |
| 17. Key Words (Suggested by Author(s))<br>acoustic finite elements<br>aeroengine duct acoustics   |  |  | 18. Distribution Statement<br>Unclassified - Unlimited;<br>Subject Category 71 |   |            |
| 19. Security Classif. (of this report)<br>UNCLASSIFIED  |  | 20. Security Classif. (of this page)<br>UNCLASSIFIED |  | 21. No. of Pages<br>34  | 22. Price* |

**End of Document**