A Simplified Computer Program for the Prediction of the Linear Stability Behavior of Liquid Propellant Combustors

C. E. Mitchell and K. Eckert
Colorado State University
Fort Collins, Colorado

Prepared for
Lewis Research Center
under Grant NGR-06-002-095

NASA
National Aeronautics
and Space Administration
Scientific and Technical
Information Branch
1979
# TABLE OF CONTENTS

**NOMENCLATURE** ................................. v

**INTRODUCTION** ................................ 1

**THEORY** ....................................... 3
  - Combustor Model .................................. 3
  - Method of Solution ................................ 10

**COMPUTATIONAL METHODS** ..................... 15
  - Matrix Sizing and Program Convergence .......... 15
  - Program Options .................................. 18
  - Choice of Fundamental Acoustic Mode .......... 21

**REFERENCES** .................................... 23

**Program MODULE** ............................... 25
  - General Description of Program .................. 25
  - Program Input .................................... 32
  - Program Output ................................... 34
  - Sample Run ...................................... 34
  - Program MODULE Listing ............................ 40
NOMENCLATURE

Letters

a - speed of sound
AMF - aperture mean flow
BR - backing distance for slot absorber
F - quantity for integral governing equation
f - quantity for integral governing equation
G_N - modified Green's function, defined after Equation 15
i - \( \sqrt{-1} \)
J_m - Bessel Function of first kind of order \( m \)
K - acoustic impedance of slot absorber
L - nondimensional chamber length
\( \ell \) - radial acoustic mode assumed
\( L_a \) - actual length of aperture for slot absorber
\( L_{eff} \) - effective length of aperture for slot absorber
M - mean flow Mach number
m - transverse acoustic mode assumed
\( \dot{m} \) - mass generation rate
n - longitudinal acoustic mode assumed
\( \eta \) - interaction index
\( \hat{n} \) - outward directed normal unit vector
P - pressure
r - radial dimension
\( r_c \) - radius of chamber
\( R_o \) - resistance of slot absorber
S - surface of combustion chamber over which integration is to be carried out
T - temperature

t - time

\( u_\theta \) - normal component of velocity oscillation in tangential direction

\( u_r \) - normal component of velocity oscillation in radial direction

\( u_t \) - transverse velocity

V - velocity or volume of combustion chamber

Wa - aperture width for slot absorber or length of acoustic liner

\( x_a \) - distance from injector face to beginning of slot absorber or acoustic liner

\( x_b \) - distance from injector face to end of slot absorber or acoustic liner

z - longitudinal dimension

**Greek Letters**

\( \beta \) - acoustic admittance of a surface

\( \gamma \) - ratio of specific heats

\( \epsilon \) - nondimensional wave amplitude

\( \eta \) - acoustic eigenvalue

\( \theta \) - angle in radians

\( \Lambda \) - normalization factor defined after Equation 15

\( \lambda_{lm} \) - root of Bessel Function of first kind, such that

\( J'_m(\lambda_{lm}) = 0 \)

\( \mu \) - coefficient matrix

\( \rho \) - density

\( \tau \) - sensitive time lag

\( \phi \) - velocity potential
\( \psi \) - normalization factor defined after Equation 15

\( \Omega_{lm1} \) - acoustic eigenfunction

\( \omega \) - complex frequency

**Superscripts**

+ - vector quantities

* - dimensional quantity

' - derivative with respect to argument, or perturbation quantity

- - mean or steady state quantity

**Subscripts**

I - injector

L - liner

N - nozzle
INTRODUCTION

The purpose of this report is to present an analytical technique and a computer program which can be used for the prediction of the linear stability behavior of liquid propellant combustors. The technique involved has been developed over the last few years at Colorado State University in the examination of several aspects of the instability problem. Basically, the approach employs a Green's function integral method in the iterative determination of combustor frequency, decay rate and spatial waveform.

This general approach has been applied to several different combustor models in the examination of different aspects of the linear instability problem. (Ref. 1-10.) This work was performed by several different people (mainly graduate students), and a wide variety of nomenclature and programming techniques has resulted. The details of the analytical approach have also varied from author to author though the general method remained the same. This more or less comprehensive compendium of programs and analyses as it exists in its several forms is cumbersome, somewhat redundant, and certainly hard to use as a designer's tool.

With this in mind it was decided to develop a simplified stability analysis and computer program which contained the most important features of the earlier work in a format that would be relatively easy to use. Consequently, the main goal of this effort has been the development of a computer program simple enough to be used effectively by a person without an exhaustive background in either advanced mathematics or stability theory.

In order to do this some compromises have had to be made as far as comprehensiveness and accuracy are concerned, and some aspects of the stability
problem treated previously have not been included. For example, the effect of distributing combustion sources along the combustor axis (as opposed to having a concentrated combustion zone near the injector) on overall stability has been studied and analyzed using two different approaches (Ref. 8, 9). This effect is not included in the simplified model presented here, however. The justification for this is based on the fact that much greater complexity is introduced into both the analysis and the computer program when distributed sources of combustion are considered, while the qualitative stability behavior is very similar to that predicted for concentrated combustion. Moreover, the quantitative effect of distributing the combustion is stabilizing relative to the predictions for a concentrated combustion zone. Thus, the simplified model presented here will tend to give conservative estimates of combustor stability when the combustor being examined has its combustion zone well distributed (axially).

Other effects such as irrotationality, entropy variations, and droplet drag effects have also been ignored since their influence has been found to be small, stabilizing or both.

The body of the report is divided into three main sections. The first (called "Theory") presents the model and method of analysis. The second section (called "Computational Methods") presents the basics of the computational method and the user options available. The final section (called "Program MODULE") gives a user's manual, sample input and output and a flow chart. It is not necessary for a person wishing to use the computer program (MODULE), to follow the analytical details of the first section. It will be necessary, however, for him to understand the basics of the model and general method of approach as presented in that section so that appropriate input to the program may be made and correct interpretation of the output can result.
THEORY

Combustor Model

The motor configuration considered here is characterized by circular cylindrical geometry, a concentrated combustion zone located at the injector end of the combustor, a nozzle at the opposite end, and either an acoustic liner or a slot absorber located in the cylindrical walls. A sketch of the combustor model is given in Figure 1. In the development of a linear stability model for a combustor of this type it is first necessary to represent the four main features of the configuration using appropriate mathematical models. The four aspects of the problem requiring such modeling are

1) The gasdynamic flow field
2) The combustion zone
3) The nozzle
4) The acoustic liner or absorber.

Each of these will be discussed separately before going on to a presentation of the global stability model and analytical technique.

1) The gasdynamic flow field

The flow field downstream of the concentrated combustion zone is taken to consist of a single component, single phase product gas which is non-conducting, inviscid and calorically perfect. The flow is assumed to be homentropic and irrotational. As long as the combustion zone is concentrated and pressure waves are of small amplitude, it has been shown that these approximations are not severely limiting and self-consistent (Ref. 11, 12). Before presenting the equations describing this flow field the relevant state and flow variables are non-dimensionalized as follows.
Figure 1. Combustion Chamber
\[ \rho = \rho^*/\bar{\rho}^* , \quad T = T^*/\bar{T}^* \]
\[ \dot{V} = \dot{V}^*/\bar{a}^* , \quad \dot{P} = \dot{P}^*/\bar{P}^* \]

The independent variables are nondimensionalized as follows.

\[ t = t^*(r_c^*/\bar{a}^*) , \quad r = r^*/r_c^* \]
\[ z = z^*/r_c^* \]

where * denotes dimensional quantities and - denotes mean chamber values.

Using this nondimensional scheme the conservation equations become

\[ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \dot{V}) = 0 \quad \text{CONTINUITY} \quad (1) \]
\[ \frac{\partial \dot{V}}{\partial t} + \frac{1}{Y} \nabla \dot{P} = 0 \quad \text{MOMENTUM} \quad (2) \]
\[ P = \rho^Y \quad \text{HOMENTROPIC} \quad (3) \]
\[ P = \rho T \quad \text{STATE} \quad (4) \]
\[ \nabla = \nabla \phi \quad \text{IRROTATIONALITY} \quad (5) \]

Under the assumption of small amplitude oscillations the state and flow variables are represented as the sum of a mean (steady state) component and an oscillatory component, products of which are ignored as being higher order terms. Thus

\[ P = 1 + p' \quad \phi = Mz + \phi \]
\[ \rho = 1 + \rho \]
\[ T = 1 + T' \quad \dot{V} = M\dot{z} + \nabla \phi \]
where \( \hat{e}_z \) is the unit vector in the axial direction and \( M \) is the mean flow Mach number.

After some manipulation the conservation equations can be reduced to a simple scalar partial differential equation

\[
\nabla^2 \phi - \frac{\partial^2 \phi}{\partial t^2} = 2M \frac{\partial^2 \phi}{\partial t \partial z} + M^2 \frac{\partial^2 \phi}{\partial z^2}
\]  

(6)

The equation relating the state variables to \( \phi \) is

\[
p' = -\gamma \left( \frac{\partial \phi}{\partial t} + M \frac{\partial \phi}{\partial z} \right)
\]  

(7)

Periodic oscillations in time are assumed so that

\[
p' = p(r, \theta z) e^{i\omega t}
\]

\[
\phi = \phi(r, \theta, z) e^{i\omega t}
\]

where \( \omega = \omega_R + i\lambda \) is the complex frequency, \( \omega_R \) the frequency, \( \lambda \) the decay rate. (If \( \lambda > 0 \) decay occurs.)

2) Combustion zone response model

It is assumed that all combustion occurs in a length small compared with the combustor's axial dimension. In the steady state mass is produced at the rate \( \dot{m} = M \) in the nondimensional system used here. No attempt to describe the details of the combustion process is made. Instead it is simply assumed that the combustion zone is sensitive to pressure oscillations and responds to these oscillations through a combustion zone admittance function \( B_1 \).

Thus

\[
\nabla \phi \cdot \hat{n} B_1 p' (z = 0)
\]  

(8)
$\beta_I$ is taken to be a constant for the entire combustion zone, though $p'$ is, of course, a function of $r$ and $z$ as well as time. In terms of the mass perturbation rate, $\dot{m}'$, the response condition is

$$\dot{m}' = \left(\frac{M}{\gamma} - \beta_I\right)p'$$

(9)

$\beta_I$ is, in general, complex so that all phasings between $\dot{m}'$ (or $u'$) and $p'$ are possible. Note that if the real part of $\beta_I$ is greater than $\frac{M}{\gamma}$, the combustion zone provides a damping rather than driving effect.

It is also possible to relate $\beta_I$ to the interaction index $n$, and time lag $\tau$ of the Crocco sensitive time lag model. The appropriate relationship is

$$\beta_I = M\left(\frac{1}{\gamma} - n \left(1 - e^{-i\omega\tau}\right)\right)$$

(10)

Values of $\beta_I$ (or $n$, and $\tau$) must be supplied by the program user or calculated as output, given all other parameters. These options will be discussed later.

3) Nozzle model

Here again no attempt is made to investigate the details of the nozzle flow and, instead, a nozzle admittance function $\beta_N$ is used.

$$\vec{\phi} \cdot \vec{n} = \beta_N p'\ (z = L)$$

(11)

Values of $\beta_N$ are to be supplied by the user. Tables of admittance functions are given in Ref. (13), for example, for conical nozzles. In the absence of any knowledge of the nozzle response value it is suggested that the simple "short" nozzle value

$$\beta_N = M \left(\frac{\gamma - 1}{2\gamma}\right)$$

be used.
4) Acoustic liner or slot absorber model.

Two possibilities are considered. The first is an acoustic liner of uniform average admittance, \( \beta_L \), which is uniform in the azimuthal (\( \theta \)) direction and extends along the cylindrical wall from \( z = z_A \) to \( z = z_B \). For this liner the appropriate boundary condition is

\[
\vec{\nabla} \Phi \cdot \hat{n} = \beta_L p'
\]  

(12)

No attempt is made to calculate \( \beta_L \) in either the analysis or computer program and therefore \( \beta_L \) must be supplied by the user.

The second absorber configuration considered is a circumferential slot machined into the cylindrical wall of the chamber and acting as a Helmholtz resonator. The geometry assumed is shown in Figure 2. All dimensions are nondimensional through division with the chamber radius.

The appropriate boundary condition at \( r = 1 \) (chamber wall) over the aperture width \( W_A \) (\( = [x_B - x_A] \))

\[
\vec{\nabla} \Phi \cdot \hat{n} = \beta_L p
\]

or

\[
\nabla \Phi \cdot \hat{n} = \frac{1}{\gamma K} p
\]

where \( K = \frac{1}{\gamma \beta_L} \) is the impedance at the aperture entrance. \( K \) is used in this case to be consistent with existing treatments of Helmholtz resonators of this general type. \( \beta_L \) and \( K \) are, in general, complex with \( K = R_0 + ik \), where \( R_0 \) is the resistance, \( k \) the reactance. Standard relationships for \( R_0 \), resistance and \( L_{\text{eff}} \) taken from Reference (15) and (16) respectively, are given below.
Figure 2. Slot Absorber Geometry
\[ R_0 = \left[ 0.8 \left( 1.5(AMF) R_0 + \frac{e|p|}{\gamma \left( 1 + \frac{k^2}{R_0^2} \right)^{\frac{1}{2}}} \right) \right]^\frac{1}{2} \]  

where \( e \) is wave amplitude and \( p \) is the modulus of the pressure at \( r = 1 \).

\[ L_{\text{eff}} = L_A + (0.375)(0.85)W_A \left[ 1 - 0.7 \left( \frac{W_A}{W_C} \right)^{\frac{1}{2}} \right] \]  

An expression for the reactance \( k \), comes directly from linear Helmholtz resonator theory and is given by

\[ k = \omega_R \bar{\rho}_{ap} L_{\text{eff}} - \frac{W_A \bar{a}_{a} \bar{\rho}_{a}}{\omega_R \bar{W}_{C} \bar{BR}} \]

where \( \bar{\rho}_{ap} = \left( \frac{\bar{p}_{ap}}{\bar{p}^*} \right) \), \( \bar{a}_{a} = \left( \frac{\bar{a}_{a}}{\bar{a}^*} \right) \), \( \bar{\rho}_{a} = \left( \frac{\bar{\rho}_{a}}{\bar{\rho}^*} \right) \) are, respectively,

the nondimensional aperture density, absorber cavity mean sound speed, and absorber cavity density, and \( \omega_R \) is the real part of the oscillation frequency.

All these quantities, as well as the geometrical quantities in Fig. 2, AMF, and the assumed wave amplitude, \( e \), must be supplied by the user. The expression for \( R_0 \), Equation (4), is then solved iteratively for \( R_0 \) as a function of frequency. Thus, an expression for \( K(\omega) \) (or \( \beta_L(\omega) \)) is found numerically.

**Method of Solution**

The governing partial differential equation, Equation (6) along with the necessary boundary conditions (Equations (8) (10) (12) or (13)) are transformed to integral form using a Green's function, and the resulting integral equations are solved iteratively. Details of the transformation and solution method are presented in References (1, 2, 7). Only those relationships, definitions, and
equations necessary for understanding and using the computer program which determines combustor stability will be presented here.

The transformed integral equations for \( \psi(r, \theta, z) \) and \( \omega \) are

\[
\begin{align*}
\phi &= \sum_{\ell,m,n} \Omega_{\ell,m,n} + \frac{1}{V_0} \int G_N\left(\frac{r}{r_0}\right) F_1(\phi) dV_0 \\
&+ \frac{1}{S_0} \int G_N\left(\frac{r}{r_0}\right) f_1(\phi) dS_0 \\
&+ \frac{1}{S_0} \int G_N\left(\frac{r}{r_0}\right) f_1(\phi) dS_0 \\
\omega^2 - n_{l,m,n}^2 &= \int \int \sum_{\ell,m,n} F_1(\psi) d\Omega + \int \Omega_{l,m,n} F_1(\phi) dS
\end{align*}
\]

where \( F_1(\phi) = 2i\omega M \frac{\partial \phi}{\partial z} + M^2 \frac{\partial^2 \phi}{\partial z^2} \)

\( f_1 = -\beta p \)

\( \beta = \beta_H \) at nozzle \((z = L)\)

\( \beta = \beta_I \) at combustion zone \((z = 0)\)

\( \beta = \beta_L \) (or \( \frac{1}{\gamma K} \)) at liner (or absorber) \((r = 1)\)

\( \beta = 0 \) on all other surfaces

\( G_N\left(\frac{r}{r_0}\right) = \sum_{\ell,m,n} \sum_{\ell,m,n} \frac{\Omega_{\ell,m,n}(\hat{r}) \Omega_{\ell,m,n}(\hat{r}_0)}{(\omega^2 - n_{l,m,n}^2)} \)

\( \ell \neq \ell, m \neq m, n \neq n, \text{ simultaneously} \)
\[ \Omega_{\lambda mn} = \frac{J_m(\lambda \lambda m r) \cos \frac{n \pi z}{L} \cos m \theta}{\lambda_{\lambda mn}^2} \]

\[ \lambda_{\lambda mn} = \int \int \int_{V} \left[ J_m(\lambda \lambda m r) \cos \frac{n \pi z}{L} \cos m \theta \right]^2 dV \]

\( \lambda_{\lambda m} \) are the roots of \( J'_{m}(\lambda_{\lambda m}) = 0 \)

\[ \eta_{\lambda mn}^2 = \lambda_{\lambda m}^2 + \left( \frac{m \pi}{L} \right)^2 \]

\( \Omega_{\lambda mn} \) are the normalized eigenfunctions for a cylindrical chamber with no mean flow and non reactive walls. \( \lambda, m, n \) are the set of integers giving the radial, azimuthal, and axial character of the particular eigenfunction (or acoustic mode) in question. Thus, \( \Omega_{110} \) represents a first transverse mode, \( \Omega_{120} \) a second transverse mode, \( \Omega_{200} \) a first radial mode, \( \Omega_{001} \) a first axial mode, \( \Omega_{111} \) a combined first transverse first axial mode, etc. The associated eigenvalues (acoustic frequencies) are

\[ \eta_{\lambda mn}^2 = \lambda_{\lambda m}^2 + \left( \frac{m \pi}{L} \right)^2 . \]

The solution technique revolves around the assumption that the actual solution including mean flow and reactive walls has a character that is reasonably close to one of these acoustic modes. The particular acoustic mode most characteristic of the overall oscillation is called \( \Omega_{\hat{\lambda} \hat{m} \hat{n}} \), where \( \hat{\lambda}, \hat{m}, \hat{n} \) are the associated indices giving the radial, azimuthal, and axial character. The related eigenvalue (acoustic frequency) is \( \eta_{\hat{\lambda} \hat{m} \hat{n}} \). A discussion of the selection of \( \Omega_{\hat{\lambda} \hat{m} \hat{n}} \) in applications will be given later.
The equation for $\phi$, Equation (14), implies that $\phi$ takes the following form

$$\phi = \sum_{\lambda \lambda m} \mu_{\lambda m} J_m(\lambda r) \cos \frac{nmz}{L} \cos m \theta$$

where the coefficient matrix $\mu_{\lambda mn}$ is determined by evaluation of the integrals on the right hand side of Equation (14).

Because of the symmetry in the $\theta$ direction which results from the assumptions concerning the boundary conditions, the series in $m$ actually contains only one term, $\hat{m}$.

Thus, for the model used here $\phi$ may be written

$$\phi = \sum_{\lambda \lambda n} \mu_{\lambda n} J_m(\lambda r) \cos \frac{nmz}{L} \cos \hat{m} \theta$$ (16)

where $\hat{e}_\theta^2 = \frac{2\pi}{\int_0^2 (\cos \hat{m} \theta)^2 d\theta}$. Exactly the same coefficient matrix would result if traveling waveforms were assumed. In this case

$$\phi = \phi(r,z) e^{i(\omega t + \hat{m} \theta)}$$

$$\phi = \sum_{\lambda \lambda n} \mu_{\lambda n} J_m(\lambda r) \cos \frac{nmz}{L}$$ (17)

and $\mu_{\lambda n}$ would be identical to the standing wave matrix.

The matrix $\mu_{\lambda n}$ and the complex frequency $\omega$ are determined by an iterative process. The lowest order guess for $\phi$ (or $\mu_{\lambda n}$) is used in the integral expressions of Equations (14) and (15) to compute improved values for the $\mu_{\lambda n}$ and $\omega$. The process continues until successive iterations are invariant to some degree of accuracy. A natural choice for the lowest order estimate for $\phi$ would be $\Omega_{\lambda n}$; the lowest order frequency would then be $\eta_{\lambda n}$. Though these initial guesses will work in general,
experience has indicated that convergence can be slow and matrix sizes large, particularly when the mean flow Mach number is greater than about 0.3. Better convergence and a smaller matrix size are possible if the separation of variables solution for a combustor with mean flow but without an absorber is used. This solution was originally developed by Priem and Rice (Ref. (14)); in the modified form appropriate here it is discussed in References (1) and (2). The computer program presented later uses this form as the lowest order $\phi$.

In addition to assuming a lowest order form for $\phi$ and $\omega$ and iterating, it is also possible to fix $\omega$ at some prescribed value (supplied by the user) and iterate to find the appropriate $\mu_{an}$ and $\beta_1$ (or $n$, and $\tau$) from the same equations. The latter approach is used to solve for the combustion response necessary to sustain an oscillation of a given frequency and decay (growth) rate and known absorber and nozzle admittances. It would also be possible to set up the technique to solve iteratively for another parameter, such as nozzle admittance, for given combustion admittance; however, this has not been done in the program presented here.
COMPUTATIONAL METHODS

As discussed in the "Theory" section, Equations (14) and (15) are set up for iterative solution. Computer program MODULE is an algorithm for performing the necessary iterative computations on a digital computer. Several different choices are possible as far as input, output, and accuracy are concerned. These choices will be discussed in this section.

Matrix Sizing and Program Convergence

In the solution of Equations (14) and (15) two variables are always iterated. One of these is the perturbation velocity potential \( \phi(r, \theta, z) \). The other is either the complex frequency \( \omega (\omega_R + i\omega_I) \) or the complex combustion admittance, \( \beta_I (\text{real}(\beta_I) + i \text{imag}(\beta_I)) \).

The perturbation velocity potential is represented by a series expansion (Equation (16)).

\[
\phi(r, \theta, z) = \sum_{\ell, n} \mu_{\ell n} \frac{J_{\ell n} (\lambda_{\ell n} r)}{\ell n} \cos \frac{\ell n z}{L} \cos \hat{n} \theta
\]

Thus, solution for the coefficient matrix \( \mu_{\ell n} \) yields \( \phi(r, \theta, z) \) and, in fact, it is this matrix which is the actual iterated variable in the solution algorithm. Formally, \( \mu_{\ell n} \) is doubly infinite in \( \ell \) and \( n \). That is, \( 1 \leq \ell < \infty \), \( 0 \leq n < \infty \). As a practical matter, however, limits on the largest values \( \ell \) and \( n \) may take (in other words the dimensions of matrix \( \mu_{\ell n} \)) must be determined. It should be recalled here that the integers \( n \) are associated with axial dependence (through \( \cos \frac{\ell n z}{L} \)) while the integers \( \ell \) are associated with radial dependence (through \( J_{\ell n} (\lambda_{\ell n} r) \)).

Any choice for the maximum number of "\( \ell \) terms" and "\( n \) terms" will limit accuracy. A compromise between program run time, storage requirements,
and accuracy is desirable. Naturally, no one choice will be optimal for all combustor configurations. However, hundreds of runs with "typical" designs have indicated some rules of thumb to be used.

First of all, in none of the combustors investigated was any significant increase in accuracy obtained by keeping more than 50 terms in the axial direction or ten terms in the radial direction. That is, keeping 100 terms in the axial direction or 20 terms in the radial direction affected the values of the iterated variables only very slightly (<0.25%). Consequently, the program as written accepts a 10 x 50 matrix size for $\mu_{\alpha n}$ as the maximum allowable. In the program variables this means $LTS \leq 10$, $NTS \leq 50$, where LTS and NTS are, respectively, the number of terms in the "$\alpha$" direction and the number of terms in the "n" direction.

The question as to the "best" values of LTS and NTS to use in a given combustor configuration is difficult to answer. Eckert (Ref. (14)) has studied optimal values for LTS and NTS for a "typical" configuration and suggests values of 3 for LTS and 16 for NTS. However, for a combustor with no absorber, a single term ($\hat{\alpha}$) is necessary for description of the radial field and $LTS = 1$ in this case. On the other hand, if the Mach number is small, mean flow effects are less important and fewer terms in the axial direction (smaller NTS) would be needed. However, for configurations with large absorber effects or high Mach numbers (> 0.4) it is likely that "best" values for LTS and NTS could be greater than 3 and 16, respectively.

With this in mind it is suggested that the values LTS = 3 and NTS = 16 be used as a general rule. If strong absorber or high Mach number effects are present and may compromise accuracy, it is suggested that results with LTS = 9 and NTS = 50 be computed and compared with the smaller matrix results to estimate accuracy. Values of LTS and NTS larger than 3 and 16
could then be inserted until the desired accuracy relative to the 9 x 50 size was obtained. It should be noted that improvement in accuracy is monotonic with increasing NTS. The same is not true for LTS because of the alternating nature of the series involved and best results occur if LTS is an odd number (3, 5, 7, 9).

Once the dimensions of the \( P_n \) matrix are determined it is next necessary to decide upon an acceptable convergence condition for the iteration process. The second iterated variable (either \( \omega \) or \( \beta_1 \) depending upon the application) is used to do this. Successive values of the iterated variable are compared. When the difference between the two values is less than some value, adequate convergence is assumed. Since both \( \omega \) and \( \beta_1 \) are complex numbers, it is necessary that both the real and imaginary parts converge in the sense just mentioned. In this program, however, it is convenient instead to deal with \( \omega \) (or \( \beta_1 \)) in complex polar notation, and require that successive values of the modulus and phase angle converge. This is because the phase angle is frequently near zero and can cause problems in the definition of convergence for the imaginary part of the iterated variable. In program MODULE convergence is assumed when the percent change in the modulus of the iterated variable is less than the value \( \text{ERROR} \) and, at the same time, the absolute value of the change in the phase angle is also less than \( \text{ERROR} \). \( \text{ERROR} \) can take values between \( 10^{-5} \) and 1.0.

In most cases convergence to within 0.1% or less is rapid, usually occurring in ten iterations or less. However, for some choices of parameters and for some program options it can be much slower or not occur at all. For this reason a maximum desired number of iterations must be specified. This is done through program variable IDMAX which can take any integer value. If convergence does not occur in the number of iterations
specified by IDMAX, the iterative loop terminates, and program values at the last iteration are output.

Program Options

In addition to choosing either \( \omega \) or \( \beta_I \) as the iterated variable, choices are possible as far as the form of the combustion response model and the acoustic absorber. Taken together this results in six distinct ways of running the program. These possibilities are labelled options and are described sequentially below. For all of the options it is necessary that certain design or program variables be specified by the user. These parameters are \( \gamma \) (ratio of specific heats), \( M \) (mean flow Mach number), \( L \) (chamber length to radius ratio), \( \beta_N \) (complex nozzle admittance), ERROR (maximum error allowable in determining convergence), LTS (number of terms in radial direction kept), and NTS (number of terms kept in the axial direction).

Option 1

This option is designed to compute frequency and decay rate (complex frequency) for known combustion zone admittance and known acoustic absorber (or liner) length and admittance. The iterated variable is the complex frequency. Required to be input to the program are \( \beta_I \), \( \beta_L \), \( X_A \) and \( X_B \). \( X_A \) and \( X_B \) are the nondimensional distances to the start and end of the acoustic absorber, respectively. Output are \( \omega_R \) and \( \lambda \), \( \nu_{2n} \), the input parameters, and \( n \) and \( \tau \), the interaction index and time lag corresponding to the given \( \beta_I \) and the converged value for \( \omega \).
Option 2

Option 2 is similar to Option 1 except that the combustion response is described by $n$ and $\tau$ instead of $\beta_I$. In this case $n$ and $\tau$ are input and $\beta_I$ is output. Other input and output parameters are the same as for Option 1.

Option 3

In this option the acoustic absorber is of the slot design type described earlier. The combustion response is described by $\beta_I$, and $\omega$ is the iterated variable. Required input variables are $\beta_I$, $\beta_R$ (absorber backing distance), $W_c$ (absorber cavity width), $L_a$ (absorber aperture length), $X_A$, $X_B$, $a_a$, (ratio of sound speed in the cavity to sound speed in the main chamber), $\bar{\rho}_a$ (nondimensional aperture gas density), AMF (aperture mean flow), and $\varepsilon$ (wave amplitude of the oscillation). Output variables are $\omega$, $n$ and $\tau$, $\mu_{\text{in}}$ and $\beta_L$, the equivalent absorber admittance for the given geometry.

Option 4

This option is the same as Option 3 except that $n$ and $\tau$ are input and $\beta_I$ is output.

Option 5

The last two options use $\beta_I$ as the iterated variable. They are most useful in generating stability maps in terms of $n$ and $\tau$ (Option 5) or real ($\beta_I$) and imag ($\beta_I$), (Option 6). Examples of such stability maps are presented in References 1, 2, 3 and 14. Frequency is used as parameter along these curves.
Option 5 is designed to compute $\beta_1$ for a given complex frequency and a slot absorber. All the slot absorber parameters necessary for Option 3 must be supplied here as well, in addition to the complex frequency. Output includes $\mu_{2n}$, $\beta_1$, $n$ and $\tau$, and $\beta_L$, the equivalent liner admittance.

Option 6

This option also uses $\beta_1$ as the iterated variable. In this option, however, the absorber is characterized by an admittance, $\beta_L$, and a length $W_a = X_B - X_A$. For this option $\omega$, $\beta_L$, $X_A$ and $X_B$ must be supplied and output will give $\beta_1$, $\mu_{2n}$, and $n$ and $\tau$.

A summary of the principal input and output variables for the six options is given in Table 1 below.

<table>
<thead>
<tr>
<th>OPTION</th>
<th>ONE</th>
<th>TWO</th>
<th>THREE</th>
<th>FOUR</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>real($\beta_1$)</td>
<td>imag($\beta_1$)</td>
<td>real($\beta_L$)</td>
<td>imag($\beta_L$)</td>
<td>$\omega$, $n$ &amp; $\tau$</td>
</tr>
<tr>
<td>2</td>
<td>$n$</td>
<td>$\tau$</td>
<td>real($\beta_L$)</td>
<td>imag($\beta_L$)</td>
<td>$\omega$, $\beta_1$</td>
</tr>
<tr>
<td>3</td>
<td>real($\beta_1$)</td>
<td>imag($\beta_1$)</td>
<td>$BR$</td>
<td>$AMF$</td>
<td>$\omega$, $n$ &amp; $\tau$, $\beta_L$</td>
</tr>
<tr>
<td>4</td>
<td>$n$</td>
<td>$\tau$</td>
<td>$BR$</td>
<td>$AMF$</td>
<td>$\omega$, $\beta_1$, $\beta_L$</td>
</tr>
<tr>
<td>5</td>
<td>real($\omega$)</td>
<td>imag($\omega$)</td>
<td>$BR$</td>
<td>$AMF$</td>
<td>$\beta_1$, $n$ &amp; $\tau$, $\beta_L$</td>
</tr>
<tr>
<td>6</td>
<td>real($\omega$)</td>
<td>imag($\omega$)</td>
<td>real($\beta_L$)</td>
<td>imag($\beta_L$)</td>
<td>$\beta_1$, $n$ &amp; $\tau$</td>
</tr>
</tbody>
</table>

For convenience four main input variables are called ONE, TWO, THREE and FOUR both in the program and in the table. These variables represent different quantities in the different options. For example, in Option 1 variable TWO represents the imaginary part of $\beta_1$, whereas in Option 2 it represents $\tau$, the time lag.
Choice of Fundamental Acoustic Mode

As was mentioned in the "Theory" section, the success of the iteration process revolves around the assumption that the oscillation with active walls and mean flow is similar to one of the normal acoustic modes (no flow, hard walls) of the combustion chamber. The most useful variable for determining the suitability of an acoustic mode choice is the real part of the complex frequency. As a general rule, when the frequency of oscillation in the combustor is within 10% of a particular acoustic mode frequency, convergence will usually occur if that acoustic mode is used for \( \Omega_{\lambda m n} \) in the iterative process. Since the imaginary part of the frequency can be as large as the deviation of the frequency from its acoustic value, nondimensional decay (or growth) rates as large as 0.20 (of the order of 1000 sec\(^{-1}\) for typical \( a^* \) and \( R^* \)) can occur for these conditions.

For lower acoustic modes there is considerable separation in frequencies. At higher frequencies a given frequency may be close to two (or more) acoustic modes. In this latter case convergence problems can occur and it may be necessary to test all of the possible acoustic modes sequentially. Experience with the program must be the guide in these cases.

For many (if not most) applications the acoustic mode choice is clear. For example, suppose that in a given combustor of diameter 2 ft, length 2 ft and average sound speed, \( a^* \), of 3000 ft/sec, an oscillation of frequency 5700 sec\(^{-1}\) were observed. The real part of the nondimensional frequency would be \( \Omega_R = (5700)/3000 = 1.90. \) This value is within 10% of 1.841, the acoustic frequency of the first transverse mode. \( (J_l (\lambda r), l = 1, m = 1, n = 0, \lambda_{\lambda m n} = \lambda_{11}) \) Hence, when investigating this oscillation using the iterative model, \( \Omega_{\lambda m n} = \Omega_{110} \) (i.e., \( \lambda = 1, \hat{m} = 1, \hat{n} = 0 \)). Indeed, the
choice of mode would be the same for frequencies between about 4970 sec\(^{-1}\) and 6075 sec\(^{-1}\). On the other hand, if the observed frequency were 6900 sec\(^{-1}\) the oscillation would be closer in frequency to the combined first transverse, first longitudinal mode frequency of 7260 sec\(^{-1}\) \((J_1(\tilde{\lambda}_{11})r)\) 

\[
\cos \frac{\pi \hat{\xi}}{L}, \hat{\xi} = 1, m = 1, n = 1, \eta_{2mn}^2 = \left(\frac{\lambda_{11}^2 + \frac{\pi^2}{L^2}}{L^2}\right) \quad \text{and} \quad \Omega_{111}
\]

\((\hat{\xi} = 1, \hat{m} = 1, \hat{n} = 1)\) should be used. When the frequency of oscillation is "in between" two acoustic frequencies and is within 10% of neither, it is usually best to pick the higher mode. For the example given, if \(\omega_R = 6300\) sec\(^{-1}\) it would be within 10% of neither the pure first transverse frequency nor the combined first transverse, first longitudinal frequency. The best choice in this case would be \(\Omega_{111}\) rather than \(\Omega_{110}\).

The choice of acoustic mode is input to the program through the choice of the three integers, \(\hat{\xi}\) (radial), \(\hat{m}\) (azimuthal), \(\hat{n}\) (axial). In the program these are called lHAT, mHAT, and nHAT, respectively.
REFERENCES


Program MODULE

In this section a general description of the program will be given first. Next, discussions of input and output formats will be given. Finally, a sample run will be presented and discussed, and a complete program listing will be given.

I. General Description of Program

The structure of the programs is as follows. (See Figure 3 for a flow diagram and Table 2 for a listing of program nomenclature.) After the non-default type variables have been declared, and the matrices and arrays have been dimensioned, the values for constants (such as \( \pi \)) are stored. Next, values for the two program variables \( K \) and \( IDCR \) are stated. The values for the iterated variable and the percent error in the modulus, and the absolute change in angle, will be printed out the first, last and every \( K^{\text{th}} \) iteration. \( IDCR \) is an arbitrary number, after which a percent error in the modulus of over 50\% will indicate that the problem is not converging. The values for the two constants \( K \) and \( IDCR \) (ID critical), may need to be changed, but it was felt they would not be changed often enough to warrant including them with the other input data.
Figure 3. Flow Diagram
### TABLE 2

**Computer Program Nomenclature**

**Input Variables**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMF</td>
<td>aperture mean flow</td>
</tr>
<tr>
<td>ACAV</td>
<td>$A_c$, nondimensional sound speed in slot absorber cavity</td>
</tr>
<tr>
<td>BETAI</td>
<td>$\beta_I$, acoustic admittance of injector</td>
</tr>
<tr>
<td>BETAL</td>
<td>$\beta_L$, acoustic admittance of liner</td>
</tr>
<tr>
<td>BETAN</td>
<td>$\beta_N$, acoustic admittance of nozzle</td>
</tr>
<tr>
<td>BR</td>
<td>backing distance</td>
</tr>
<tr>
<td>EPSIL</td>
<td>$\varepsilon$, amplitude of wave oscillation (only used in calculation of absorber resistance)</td>
</tr>
<tr>
<td>ERROR</td>
<td>acceptable % error in magnitude and absolute change in radians between two successive iterations</td>
</tr>
</tbody>
</table>
| FOUR   | OPTION = 1, 2 or 6, $\text{imag}(\beta^\wedge L)$ 
         | OPTION = 3, 4 or 5, $\text{AMF}$ |
| GAMMA  | $\gamma$, ratio of specific heats |
| IDMAX  | maximum number of iterations |
| IN     | $n$, interaction index |
| K      | iterated variable is output every $K^{th}$ iteration |
| LENGTH | $L$, nondimensional chamber length |
| LHAT   | $\hat{\zeta}$, radial acoustic mode, integer |
| LTS    | number of terms in radial direction |
| MACH   | $M$, Mach number |
| MHAT   | $\hat{m}$, transverse acoustic mode, integer |
| NHAT   | $\hat{n}$, longitudinal acoustic mode, integer |
| NTS    | number of terms in radial direction |
ONE - OPTION = 2 or 3, real(BETAI)
OPTION = 2 or 4, IN
OPTION = 5 or 6, real(OMEGA)

OPTION - integer value between 1 and 6, sets which way the problem will be calculated

ROAP - \( \overline{\rho}_a \), density in slot absorber aperture
ROCAV - \( \overline{\rho}_c \), density in slot absorber cavity
TAU - \( \tau \), sensitive time lag
THREE - OPTION = 1, 2 or 6, real(BETA1)
OPTION = 3, 4 or 5, BR

TWO - OPTION = 1 or 3, imag(BETAI)
OPTION = 2 or 4, TAU
OPTION = 5 or 6, imag(OMEGA)

WCAV - nondimensional slot absorber cavity width
XA - distance from injector to beginning of acoustic liner
XB - distance from injector to end of acoustic liner.

Program Variables
A1 - \( \phi \) evaluated at injector
A2 - \( \phi \) evaluated at nozzle

\[
BES = \frac{J_m^2(\lambda_{km})(\lambda_{km}^2 - m^2)}{2\lambda_{km}^2} = \int_0^1 J_m^2(\lambda_{km}r)rdr
\]

BETAIN - new BETAI
CIOM - \( i\omega \)
DZPIL - \( \frac{\partial \phi}{\partial z} \) evaluated at liner midpoint
ETA - \( \eta_{kl}^2 \)
ETAI - \( \eta_{kl}^2 \)
ID - iteration counter
LAMDA2 - \( \lambda_{km}^2 \) where, \( J_m^r(\lambda_{km}) = 0 \)
LEFF - $\varphi_{\text{eff}}$
MU - $\mu$ matrix
MUX - old AMU
NORM - $\Lambda_{\text{LIN}}^{1/2}$
NORM1 - $\Lambda_{\text{XL}}^{1/2}$
NPIL2 - $\left(\frac{\hat{n}_{\pi}}{L}\right)^2$
OMEGAN - new OMEGA
PIL - $\frac{n}{L}$
PIXL - $\frac{nXL}{L}$
PSI - $\psi$
PIL - $\phi$ evaluated at liner midpoint
SINJ - $\int \int dS_{\text{INJ}}$
SLIN - $\int \int dS_{\text{LIN}}$
SNOZ - $\int \int dS_{\text{NOZ}}$
VOL - $\int \int \int \int dV$
WA - $W_a$, width of liner aperture
WWO - $\frac{\text{OMEGA}}{\text{ETAT}}$
XL - distance from injector to midpoint of liner.

Output Variables (not defined above)
KI - Imaginary part of absorber impedance (reactance)
RO - $R_0$ real part of absorber impedance (resistance)
A point right after this (570 CONTINUE) is where the program returns to begin execution of a given set of data. The central processor (CP) time is stored at the beginning of execution of a set of data. This time is used to calculate the execution time for the set of data. Next, the data is read in and the counter ID is initialized. The constants for the set of data, such as \( \frac{m}{n} \), are then calculated. The first guess for the iterated variable and the \( \mu \) matrix (MU) are calculated, using the separation of variables solution. The input data and first guess are printed out. The setup is now complete, and each iteration returns to a point just below this (60 CONTINUE).

For each iteration the iterated variable is first updated, then variables that are functions of the iterated variable are updated. The \( \mu \) matrix is stored in an extra matrix (MUX). The portion of the program from here to 500 CONTINUE, is designed to evaluate Equations (14) and (15), which give a new \( \mu \) matrix and iterated variable, respectively. The next section, down to 550 CONTINUE, does the following: Calculates the percent error in the modulus (ERR1), and the absolute error in the angle (ERR2). Then checks to see if the problem appears to be converging (ID greater than IDCR and ERR1 greater than 50%), or has gone the maximum number of iterations. If neither of the above has happened, the iterated variable, ERR1 and ERR2 are printed out, if first or \( k \)th iteration. Then checks to see if the problem has converged (ERR1 and ERR2 are both less than ERROR). If the problem has not converged, the program returns to 60 CONTINUE. If the problem is not converging, has gone the maximum number of iterations, or has converged, the final values for the iterated variable, ERR1, ERR2, and the final \( \mu \) matrix, are printed out. If the problem does not converge, a message is printed out.
The next segment of the program, down to 710 CONTINUE, calculates and prints the other information that is to be output. If a slot absorber is used, an equivalent BETAL is calculated. If IN (n) and TAU (τ) are used, the equivalent BETAI is calculated; otherwise, the corresponding IN and TAU are calculated for the final OMEGA and BETAI. The CP time at the end of the program is stored. Running time is calculated and printed out. The calculations for this set of data are then complete and the program returns to 570 CONTINUE to begin calculations for the next set of data. If no more data is found, the program jumps to 6000 CONTINUE and stops without an error message.

The program is capable of handling up to 10 terms in the radial direction (LTS), 50 terms in the longitudinal direction (NTS), and a transverse mode as high as 4 (MHAT). This should be satisfactory for most cases. However, if the number of terms in the radial or transverse directions must be increased, the Bessel values and Bessel roots for higher modes must be added to subroutines BESVL and BESRT, respectively. Also, all relevant dimension statements must be increased. To increase the number of terms in the longitudinal direction, only the dimensions of MU and MUX must be increased. If MHAT and NHAT are zero (0), LHAT cannot be one (1). This is a trivial case.

The best compromise between good accuracy and fast running time (as discussed previously) occurs with 15-20 terms in the longitudinal direction (NTS), and 3 or 5 terms in the radial direction (LTS). LTS should always be odd, because the series has alternating signs in the radial direction. If the Mach number is greater than .40, more terms should be kept in the longitudinal direction.
Commonly, for liners covering less than one-third of the chamber walls, evaluating the integral over the surface of the liner, by evaluating at the midpoint and multiplying by the width, gives a good approximation to the integral with a big saving in running time. The program is set up to run this way. However, if it is desired to carry the integration out, replace the SLIN card with the CALL LINER card and include SUBROUTINE LINER. (See program listing after 40 CONTINUE.)

The final $\mu$ matrix, which is printed out, should always be checked to be sure that the term corresponding to the acoustic mode assumed is the largest term in the matrix. If this is not the case, the wrong primary acoustic mode has been assumed, and the answer is not a characteristic of the primary mode assumed.

**Program Input**

The input necessary and the formats for typing the data cards are listed in Table 3. Before the first data card can be typed it must be decided from what is known about the engine (or desired from the calculation) what value OPTION must take. It will be helpful to refer to the "Computational Methods" section in making this determination. Once the value of OPTION is fixed, Table 1 is used to determine which values the variables ONE, TWO, THREE and FOUR must take. The first data card can then be typed.

The second data card contains the model information which must be supplied regardless of option. All inputs are real numbers. The third card contains program variable information, including convergence and matrix size limitation information. All variables on card three are of the integer type. The fourth data card needs to be included only when
### TABLE 3

The first card:

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>VARIABLE</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-20</td>
<td>ONE</td>
<td>Real number</td>
</tr>
<tr>
<td>21-40</td>
<td>TWO</td>
<td>Real number</td>
</tr>
<tr>
<td>41-60</td>
<td>THREE</td>
<td>Real number</td>
</tr>
<tr>
<td>61-80</td>
<td>FOUR</td>
<td>Real number</td>
</tr>
</tbody>
</table>

The second card:

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>VARIABLE</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>real(BETAN)</td>
<td>Real number</td>
</tr>
<tr>
<td>11-20</td>
<td>imag(BETAN)</td>
<td>Real number</td>
</tr>
<tr>
<td>21-30</td>
<td>GAMMA</td>
<td>Real number</td>
</tr>
<tr>
<td>31-40</td>
<td>MACH</td>
<td>Real number</td>
</tr>
<tr>
<td>41-50</td>
<td>LENGTH</td>
<td>Real number</td>
</tr>
<tr>
<td>51-60</td>
<td>XA</td>
<td>Real number</td>
</tr>
<tr>
<td>61-70</td>
<td>XB</td>
<td>Real number</td>
</tr>
<tr>
<td>71-80</td>
<td>ERROR</td>
<td>Real number</td>
</tr>
</tbody>
</table>

The third card:

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>VARIABLE</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>LHAT</td>
<td>Integer</td>
</tr>
<tr>
<td>11-20</td>
<td>MHAT</td>
<td>Integer</td>
</tr>
<tr>
<td>21-30</td>
<td>NHAT</td>
<td>Integer</td>
</tr>
<tr>
<td>31-40</td>
<td>LTS</td>
<td>Integer</td>
</tr>
<tr>
<td>41-50</td>
<td>NTS</td>
<td>Integer</td>
</tr>
<tr>
<td>51-60</td>
<td>IDMAX</td>
<td>Integer</td>
</tr>
<tr>
<td>61-70</td>
<td>OPTION</td>
<td>Integer</td>
</tr>
</tbody>
</table>

The fourth card:

<table>
<thead>
<tr>
<th>COLUMNS</th>
<th>VARIABLE</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-10</td>
<td>EPSIL</td>
<td>Real</td>
</tr>
<tr>
<td>11-20</td>
<td>ROAP</td>
<td>Real</td>
</tr>
<tr>
<td>21-30</td>
<td>ACAV</td>
<td>Real</td>
</tr>
<tr>
<td>31-40</td>
<td>ROCAV</td>
<td>Real</td>
</tr>
<tr>
<td>41-50</td>
<td>WCAV</td>
<td>Real</td>
</tr>
<tr>
<td>51-60</td>
<td>LA</td>
<td>Real</td>
</tr>
</tbody>
</table>
a slot type absorber is present in the combustor. All variables appearing on this card are real.

When setting up the program the correspondence between text and program variables given in Table 2 will be useful. Also, it should be remembered that all variables in the program and text are nondimensional.

Program Output

The primary outputs of MODULE are the matrix $\mu_{\lambda n}$ and the iterated variable, either $\omega$ or $\beta_I$. Values for these quantities are computed at every step and $\omega$ (or $\beta_I$) is written out for every $K^{th}$ iteration. $K$ can be changed by the user by replacing a single card. In the program as presented here $K = 5$. The first and last iterations of $\mu_{\lambda n}$ are also printed out. The first iteration is a solution with no liner effect, consequently all terms in $\mu_{\lambda n}$ except $\mu_{\lambda n}$ are null entries.

In addition, all model design variables are printed and labelled according to the program names of Table 2. The example to be discussed next demonstrates the typical form the output takes.

Sample Run

The combustor used for this sample run has a ratio of length to radius (LENGTH) of 2.0, a mean flow Mach number (MACH) of .3, and a ratio of specific heats (GAMMA) of 1.2. From the Mach number and ratio of specific heats, a nozzle response (BETAN) was calculated using the equation for a short nozzle given in the theory section of this report. The value for BETAN is $0.025 + 0.0i$. There is an acoustic liner, of known admittance (BETAL), of $0.075 + 0.0i$ in place covering 10% of the cylindrical surface of the chamber, beginning one-tenth of the chamber length downstream of
the injector face. Since space dimensions are nondimensionalized by dividing by the chamber radius, and the nondimensional chamber length is 2.0, this gives an Xa of .2 and an Xb of .4. The interaction index IN (n) and sensitive time lag TAU (τ) are known to be equal to .30097 and 2.219497, respectively, for the injector response. With the information known about the injector and liner responses and looking at Table 1, it is determined that OPTION must equal 2, and ONE is IN, TWO is TAU, THREE is real (BETAL), and FOUR is imag (BETAL). Table 1 shows that (ω; OMEGA, the nondimensional frequency and decay rate, and the effective BETAI for the injector will be calculated. The first transverse acoustic mode is chosen as the primary mode. This corresponds to LHAT = 1, MHAT = 1, and NHAT = 0. A good compromise between running time and accuracy was desired, so by referring to the discussion in this report, the number of terms chosen in the radial direction (LTS) is equal to 3, and the number of terms chosen in the longitudinal direction (NTS) is equal to 16. A high precision is desired, so ERROR is chosen as .01. The maximum number of iterations allowed for this set of data (IDMAX) will be 50. Since an acoustic liner is used, no fourth data card is needed. The input values are then typed up on three data cards, as described in Table 3. The cards as punched and submitted appear in Figure 4.

The output from the sample run is shown in Figure 5. The program first prints out the value of every variable that is input on the data cards. This is to allow for double checking, to be sure all the input data is correct, and also so there is a complete description of the rocket engine that was simulated. Next, the fundamental frequency for the primary mode assumed, and the first guess of the iterated variable, and the
matrix are printed. (These are the separation of variables values.) If OPTION is 5 or 6, the first guess of BETAI is printed under the injector response description. Next, the value of the iterated variable is printed for the first, every $K^{th}$, and last iteration. In the sample run $K = 5$, so the first, fifth and sixth (last) iteration values are printed, along with the errors in the modulus and phase angle of the iterated variable ($\omega$ in this case). The last $\nu_{\lambda n}$ matrix (iteration 6) is then printed out. It can be seen that the term corresponding to $\lambda = \hat{\lambda} = 1$ and $n = \hat{n} = 0$ is, indeed, the largest. In fact, in this case it is an order of magnitude larger than any other matrix element. The other output information is then printed. In this case, the BETAI calculated from the input IN and TAU and the final OMEGA. The last thing printed out for each set of data is the beginning time TBG, ending time TEND, and execution time TEX, for this set of data.

Program Listing

A complete program listing is presented at the end of this report. Comment cards are used liberally and much of the program is self-explanatory. The computer program MOCDULE conforms to Fortran IV ANSI standards.
THE FOLLOWING CALCULATIONS ARE MADE FOR A COMBUSTION WITH THE FOLLOWING CONFIGURATION:

LENGTH 10 RADIUS = 2,000000
LENGTH 1 WHAT = 1
LENGTH 2 WHAT = 1
LENGTH 3 WHAT = 0
GAMMA = 1,000000

THE BURNER HAS THE FOLLOWING GEOMETRY

LENGTH 1 WHAT = 2,000000 AND END = 2,000000
LENGTH 2 WHAT = 1,000000 AND END = 1,000000

THE INJECTION RESPONSE IS GENERAT

FOR THE CO2 SENSITIVE TIME LAG THEORY

INITIAL T = 0.0

THE NOISE RESPONSE IS GENERAT

METAN = 0.0

MISCELLANEOUS INFORMATION FOR THIS RUN

NUMBER OF TERMS INITIAL 10 LONGITUDINAL 10
END = 0.0

THE FUNDAMENTAL FREQUENCY FOR THIS MODEL IS 1.00144 Hz

THE FIRST FREQUENCY OF VIBRATION IS 1.00144 Hz

THE FIRST FREQUENCY FOR THE VIBRATION IS AS FOLLOWS

<table>
<thead>
<tr>
<th>N</th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>3</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>4</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>5</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>6</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>7</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>8</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>9</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>10</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>11</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>12</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>13</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>14</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>15</td>
<td>-1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Figure 5. Output for Sample Run
<table>
<thead>
<tr>
<th>ITEM</th>
<th>REAL OMEGA</th>
<th>IMAG OMEGA</th>
<th>MODULUS</th>
<th>ANGLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.70046</td>
<td>-1.123756</td>
<td>1.007744</td>
<td>0.10132</td>
</tr>
<tr>
<td>6</td>
<td>1.74437</td>
<td>-1.167765</td>
<td>0.043414</td>
<td>0.000841</td>
</tr>
</tbody>
</table>

The final VU matrix is as follows:

<table>
<thead>
<tr>
<th>N</th>
<th>L=1</th>
<th>L=2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.04682 0.9794582</td>
<td>1.7943E-04 -6.33E-04</td>
</tr>
<tr>
<td>7</td>
<td>0.93044E-01 -4.9341E-01</td>
<td>2.731E-01 -3.93E-01</td>
</tr>
<tr>
<td>3</td>
<td>1.1792E-01 -2.3071E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>4</td>
<td>1.7467E-01 -1.8574E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>5</td>
<td>4.6971E-01 -3.1452E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>6</td>
<td>3.5532E-01 -5.4011E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>7</td>
<td>2.2209E-01 -2.4732E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>8</td>
<td>2.1374E-01 -1.4127E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>9</td>
<td>1.3427E+00 -1.9247E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>10</td>
<td>1.2045E-01 -6.6470E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>11</td>
<td>2.9420E-01 -3.1202E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>12</td>
<td>3.8492E-01 -1.5325E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>13</td>
<td>5.6467E-01 -1.7894E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>14</td>
<td>5.0359E-01 -1.4447E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
<tr>
<td>15</td>
<td>4.0946E-01 -1.2734E-01</td>
<td>1.351E-01 -3.87E-01</td>
</tr>
</tbody>
</table>

The final Reta1 is 0.00418 | 0.049174

TBG = 14.419 TFG = 15.541 TFY = 1.122

![Figure 5. (Continued)](image-url)
PROGRAM MODULE (INPUT, OUTPUT, TAPF5 = INPUT, TAPF6 = OUTPUT)

COMPUTER PROGRAM MODULF

WRITTEN AT COLORADO STATE UNIVERSITY

DEPARTMENT OF MECHANICAL ENGINEERING

FORT COLLINS, COLORADO 80523

SPONSORED BY NASA LEWIS RESEARCH CENTER

GRANT NGR 06 - 002 - 095

DIRECTED BY RICHARD J. PRIEM

THIS PROGRAM IS DOCUMENTED IN THE MASTERS THESIS OF KURTIS W. ECKERT

1976, COLORADO STATE UNIVERSITY, AND CONFORMS TO ALL FORTRAN IV ANSI

STANDARDS.

IT IS WRITTEN TO GIVE A LINEAR ANALYSIS OF HIGH FREQUENCY

COMBUSTION STABILITY IN LIQUID PROPELLANT ROCKET ENGINES. THE

PHYSICAL MODEL USED IS A RIGHT CIRCULAR CYLINDER. THE COMBUSTION

IS MODELED AS EITHER AN ARBITRARY ACoustIC ADMITTANCE (R(OMEGA)), OR

BY THE CROCCO SENSITIVE TIME LAG THEORY (IN AND TAU). THE NOZZLE

IS MODELED AS AN ARBITRARY ACoustIC ADMITTANCE (R(OMEGA)), THE LINEAR

IS MODELED AS EITHER AN ARBITRARY ACoustIC ADMITTANCE (R(OMEGA)) OVER

SOME PORTION OF THE CYLINDRICAL WALL OF THE CHAMBER, OR AS A SLOT

ABSORBER.

THIS PROGRAM HAS BEEN WRITTEN TO SOLVE FOR THE NONDIMENSIONAL

COMPLEX FREQUENCY IF ALL THE BOUNDARY RESPONSES ARE GIVEN, OR SOLVE

FOR THE INJECTOR RESPONSE IF THE COMPLEX FREQUENCY AND LINER AND

NOZZLE RESPONSES ARE GIVEN. THIS LEADS TO A TOTAL OF 6 OPTIONS

FOR RUNNING THE COMPUTER PROGRAM.

FOLLOWING IS A TABLE TO USE IN DETERMINING WHICH VALUE OF OP-

TION TO USE. THE TABLE SHOWS WHAT INFORMATION MUST BE GIVEN AND

WHAT WILL BE CALCULATED FOR EACH OPTION. R() MEANS THE REAL PART OF

THE COMPLEX VALUE INSIDE THE PARENTHESES, AND I() MEANS THE IMAGIN-

ARY PART OF THE COMPLEX VALUE INSIDE THE PARENTHESES.

TABLE OF OPTIONS

<table>
<thead>
<tr>
<th>OPTION</th>
<th>INPUT VARIABLES</th>
<th>OUTPUT VARIABLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONE</td>
<td>R(BETAI)</td>
<td>R(BETAI)</td>
</tr>
<tr>
<td>TWO</td>
<td>KERNEL</td>
<td>KERNEL</td>
</tr>
<tr>
<td>THREE</td>
<td>BETAI</td>
<td>R(BETAI)</td>
</tr>
<tr>
<td>FOUR</td>
<td>R(OMEGA)</td>
<td>R(BETAI)</td>
</tr>
</tbody>
</table>

FOLLOWING IS A TABLE LISTING THE INPUT VARIABLES THAT ARE PUT

ON EACH DATA CARD. AFTER THE VARIABLE NAME THE TYPE OF VARIABLE IS

SHOWN. THEN A BRIEF DESCRIPTION OF THE VARIABLE IS GIVEN. THEN AT

THIS END OF THE LINE ARE THE COLUMNS OF THE DATA CARD WHICH THE VALUE

FOR THIS VARIABLE MUST BE TYPED IN. ALWAYS BE SURE TO RIGHT JUSTIFY

INTEGER VALUES. THE FOURTH DATA CARD IS USED ONLY WHEN OPTION EQUALS

3, 4 OR 5.

LIST OF INPUT VARIABLES

FIRST CARD

ONE REAL SEE TABLE ABOVE 1-20
TWO REAL SEE TABLE ABOVE 21-40
THREE REAL SEE TABLE ABOVE 41-60
FOUR REAL SEE TABLE ABOVE 61-80

SECOND CARD

BETAN COMPLEX ACOUSTIC ADMITTANCE OF NOZZLE 1-10 & 11-20
MODULUS LESS THAN 5 IF NO KNOWN VALUE USE SHORT NOZZLE

GAMMA REAL RATIO OF SPECIFIC HEATS 21-30
MACH REAL  MEAN FLOW MACH NUMBER  0 < MACH < .5  31-40
LENGTH REAL  LENGTH OF CHAMBER/RADIUS OF CHAMBER  41-50
XAS REAL  DISTANCE FROM INJECTOR FACE TO START  OF LINER/RADIUS OF CHAMBER  51-60
XH REAL  DISTANCE FROM INJECTOR FACE TO END  61-70
XT REAL  TWICE LENGTH OF CHAMBER/RADIUS OF CHAMBER  71-80
ERROR REAL  MAXIMUM ALLOWABLE % ERROR IN MODULUS OF ABSOLUTE DIFFERENCE IN RADIIANS OF ANGLE TO DETERMINE CONVERGENCE OF ITERATED VARIABLE  10F-5 < ERROR < 1

THIRD CARD
LTHAT INTEGER  ASSUMED MODE IN RADIAL DIRECTION  1-10
MTHAT INTEGER  ASSUMED MODE IN TRANSVERSE DIRECTION  11-30
NTHAT INTEGER  ASSUMED MODE IN LONGITUDINAL DIRECTION  21-30
LTS INTEGER  NUMBER OF TERMS KEPT IN RADIAL DIRECTION  31-40
NTS INTEGER  NUMBER OF TERMS KEPT IN LONG. DIRECTION  41-50
MAX INTEGER  MAXIMUM NUMBER OF ITERATIONS ALLOWED  51-60
OPT INTEGER  OPTION TABLE ABOVE  61-76

FOURTH CARD
EPSIL REAL  WAVE AMPLITUDE  1-10
WAVE REAL  APERTURE DENSITY RATIO  11-20
CAYE REAL  CAVITY SOUND SPEED  21-30
PDCAV REAL  CAVITY DENSITY RATIO  31-40
WAV REAL  CAVITY WIDTH  41-50
XAS REAL  APERTURE LENGTH  51-60

REAL IN1, INO, INO, NPI2, LENGTH, LFF, LA, LAMDA, LAMDA2, NOHM, MACH, MACHC
IN1, NOHM)
INTEGER OPTION, CHECK

COMPLEX XETAN, XTA, XETA, MU, MU, OMEGA, UMEG, D, OMEGACM, CERHER, CI
XETA[O, T], XETA[1], XETA[2], XETA[3], SUM1, SUM2,
ZAV, BASF, BRETAIN, WMO
COMPLEX CTERM, CTERM, H15, H25, QM, A, EXP1, EXP2, TFRM1, TFRM2, VOL, S1NJ,
15NOZ, SINO, S1NJ, SINPST

DIMENSION MU(50*10), MUX(50*10), A(1110), A(10)

INITIALIZE CONSTANTS
P1 = 3.14159265359
C[0] = COMPLEX(0.0,0.0)
C[1] = COMPLEX(0.0,0.0)
IDCM = 20
K[5]

READ IN DATA

CALL SECOND THON
READ (5,100) ONE, TWO, THREE, FOUR, XETAN, GfftA, XETAX, LAMDA, LAMDA2, NOHM, MACH, MACHC
IF ERROR, LTHAT, MTHAT, NTHAT, LTS, NTS, MAX, OPTION
IF (EOF(5)) 40005806000

CONTINUE
IF (OPTION, GE, 3, AND, OPTION, LE, 5) READ (5, 101) EPSIL, ROA, 1 ACAYE, MUCAV, WCAV, LAMDA

INITIALIZE VARIABLES
ANGI = 0.0
ID = 0
IF (OPTION.EQ.1 .OR. OPTION.EQ.3) BETAIN = CMPLX(ONE, TWO)
IN = ONE
TAU = TWO
REAL = CMPLX(THREE, FOUR)
BW = THREE
AMF = FOUR
CALCULATE CONSTANTS

PIL = PI/LENGTH
NPIL2 = (FLOAT(NHAT)*PIL)**2
WA = XB -XA
XL = XA + WA/2.0
PIXL = PIL/LENGTH
LAMDA2 = BESHT(MHAT, NHAT)**2
IF (OPTION.LE.2 .OR. OPTION.EQ.6) GO TO 30
LEFF = LA + 0.375*0.85*WA*(1.0 - 0.7*SQRT(WA/WCAV))
CONTINUE
ETAI = SQRT(LAMDA**2 + NPIL2)

CALCULATE FIRST GUESS FOR ITERATED VARIABLES

OMEGAN = .9*REAL(CMPLX(ETA1, 0.0))
IF (OPTION.GE.5) OMEGAN = CMPLX(ONE, TWO)
OMEGAN = OMEGAN/NHAT
IF (OPTION.EQ.2 .OR. OPTION.EQ.4) 8ETAIN = MACH (|GAMMA - TN(I. -C EXP(-C_MEGAN_TAU)))

CHECK VALUES OF INPUT VARIABLES

TERM = WA_REAL(METAL)
CHECK = 0
IF (MACH.LE.0.0 .OR. MACH.GE.1.0) CHECK = 1
IF (MACH.GT.0.5) WRITE (6*,900) MACH
IF (GAMMA.LT.1.0) WRITE (6*,920) GAMMA
IF (LENGTH.LE.0.0) CHECK = 1
IF (LENGTH.GE.3.0) WRITE (6*,902) LENGTH
IF (TERM.LT.0.0) CHECK = 1
IF (TERM.GT.0.3) WRITE (6*,904) TERM
CONTINUE
IF (OPTION.NE.2 .AND. OPTION.NE.4) GO TO 44
IF (IN.LT.0.0 .OR. TAU.LT.0.0) CHECK = 1
IF (IN.GT.3.0) WRITE (6*,912) IN
IF (TAU.GT.4.0) WRITE (6*,914) TAU
CONTINUE
IF (OPTION.NE.1 .AND. OPTION.NE.3) GO TO 47
IF (CABS(MHAT).GT.2.0) WRITE (6*,916) MHAT
CONTINUE
IF (XAX.LT.0.0 .OR. XA.GT.XR.GT.LENGTH) CHECK = 1
IF (WA.GT.0.5) WRITE (6*,930) WA
IF (LHAT.LE.7.0) WRITE (6*,931) LHAT
IF (NHAT.LE.10) WRITE (6*,931) NHAT
IF (MTS.LE.10) WRITE (6*,940) MTS
CONTINUE
IF (EPSIL.LT.0.0) WRITE (6*,942) EPSIL
CONTINUE
IF (AMF.LE.1.0) WRITE (6*,950) AMF
IF (EPSIL.LE.0.0) WRITE (6*,942) EPSIL
CONTINUE
IF (TERM.EQ.1) WRITE (6*,940) MACH, GAMMA, LENGTH, METAL, ETAI, XA, XH, IN, TAU
IF (CHECK.EQ.1) GO TO 710
INITIALIZE FIRST GUESS OF MU MATRIX

DO 50 N1=1,NTS
DO 50 L=1,NTS
MU(N1,L)= CZERO
50 CONTINUE

THE TILDA SOLUTION

RV= BESVL(MHAT,LHAT)
FP= 1.
IF (MHAT.EQ.0) FP= 2.
RES1= RV*BV*(LAMDA2 = FLOAT(MHAT)**2)/2.*LAMDA2
NORM1= SORT(RES1*P*LENGTH/2.)*
MACH2= MACH*MACH
DO 75 N= 1,100
CTERM= MACH2*OMEGAN*OMEGAN*(MACH2-1.)**0.5*(LAMDA2-OMEGAN*OMEGAN)
CSTEM= CSQRT(CTERM)
B1= (MACH*OMEGAN+CTERM)/(1.-MACH2)
B2= (MACH*OMEGAN-CTERM)/(1.-MACH2)
R1SQ= B1*B1
R2SQ= B2*B2
ETA= (EXP(CI*LENGTH*(R1-H2))*(R1-EXPANOMEGAN*OMEGAN*MACH))/
(1.-(R2+BetANOMEGAN*MACH))
EXP1= EXP(CI*LENGTH*R1)
EXP2= EXP(CI*LENGTH*R2)
TERM1= B1*(EXP1*(-1.)**(NHAT-1.)/(M10*NPI12)
TERM2= A*B2*EXP2*(-1.)**(NHAT-1.)/(H2SQ*NPI2)
OMEGA= MACH*(1.**(TERM1+TERM2)-2.*OMEGAN*(R1*TFWM1*)
1.2*TERM2)
SINJ= GAMMA*(OMEGAN(1.**(A))**MACH**(M1**(A)))
SINJ= RESEXN(SINJ)
SNOZ= BETAONOMEGAN*(-1.)**(NHAT-1.**(OMEGAN*(EXP1+A**EXP2)**MACH)**R1)
IF (OPTION.EQ.0) GO TO 70
OMEGA= -(VOL + SINJH + SNOZ*/TERM1*TFWM2) + ETA**2
OMEGA= CSQRT(OMEGA)
IF (OPTION.EQ.4) RETAIN= MACH**(1.**(GAMMA-1.**(1.-C)
EXP(-CI*OMEGAN**TAN))
CERR= OMEGA - OMEGAN
OMEGAN= OMEGAN
IF (ABS(REAL(CERROR)**GT.0.0001) GO TO 75
IF (ABS(MAG(CERROR)**GT.0.0001) GO TO 75
GO TO 71
70 CONTINUE
RETAIN= ((ETA**2- OMEGAN**2)*TERM1*TERM2 - VOL + SNOZ)/SINJ
GO TO 71
75 CONTINUE
71 CONTINUE
PS1= (-2.*CI/LENGTH)**(TERM1+TFWM2)
IF (MHAT.EQ.0) PS1= PSI1/2.
MU(NHAT+1,LHAT)= CMPLX(1./NORM1,0,0)
DO 55 N1=1,NTS
N= N1-1
RN= FLOAT(N)
IF (N.EQ.NHAT) GO TO 55
TERM= FLOAT((-1)**N)
NPL2= (RN*NPL2)**2
FP= 1.
IF (N.EQ.0) FP= 2.
NORM= RES1*P*LENGTH/2.
ETA= LAMDA2 + NPI12
TERM1= (EXP1**TERM - 1.)/(B1SQ - NPL2)
TERM2= (EXP2**TERM - 1.)/(B2SQ - NPL2)
VOL= MACH2*(R1SQ*B1**TERM1 + A*B2SQ*B2**TERM2) + 2.*MACH*OMEGAN*(R1**TERM1) + A**TERM2
SNOZ= BETAONOMEGAN**TERM1*OMEGAN*(EXP1*A**EXP2) + MACH**(R1**EXP1+
A**EXP2))
SINJ= BETANONOMEGAN**(OMEGAN**(1.)**(A))**MACH**(B1**(A)**B2)
MU(N1,LHAT)= BES1*CI*(VOL + SNOZ + SINJH)/(OMEGAN**2 - ETA)/NORM1
1./NORM/P51
CONTINUE
PRINT OUT PROBLEM DESCRIPTION AND INITIAL VALUES
WRITE (6,1) LENGTH,MACH,LHAT,NHAT,OPTION,GAMMA
WRITE (6,2) XR,WA
IF (OPTION.GE.3 .AND. OPTION.LE.5) GO TO 20
WRITE (6,235) BETAL
GO TO 22
CONTINUE
WRITE (6,240) AMF,LA,LEFF,EPSIL+WCAV+ROCAV+ROAP+CAV
22 CONTINUE
WRITE (6,235) BETAL
GO TO 22
CONTINUE
WRITE (6,240) AMF,LA,LEFF,EPSIL,WCAV+ROCAV+ROAP+CAV
22 CONTINUE
WRITE (6,250) IF (OPTION.GE.5) GO TO 10
IF (OPTION.EQ.2 .OR. OPTION.EQ.4) GO TO 12
WRITE (6,260) RETAIN
GO TO 14
12 CONTINUE
WRITE (6,270) IN,TAU,RETAIN
GO TO 14
10 CONTINUE
WRITE (6,280) OMEGAN,WWO
WRITE (6,285) RETAIN
14 CONTINUE
WRITE (6,290) IN,TAU,RETAIN
WRITE (6,310) BETAN
WRITE (6,320) LTS,NTS,INMAX,EFFOR
WRITE (6,330) ETA1
WRITE (6,350) OMEGAN
DO 35 N=1,NTS
NM1= N-1
WRITE (6,104) NM1,MUX(NI,L),L=1,LTS
35 CONTINUE
WRITE (6,2)
IF (OPTION.LE.4) WRITE (6,350)
IF (OPTION.GE.5) WRITE (6,360)
BEGAN ITERATION
CONTINUE
UPDATE ITERATED VARIABLES
IF (OPTION.EQ.2 .OR. OPTION.EQ.4) RETAIN=MACH*(1.+GAMMA- IN*(1.+C
1EXP(-C1*OMEGA*TAU)))
BETAN= RETAIN
OMEGA=OMEGAN
C1OM=C1*OMEGA
WR= REAL(OMEGA)
STORE NEW MU MATRIX IN EXTRA MATRIX
DO 800 N1=1,NTS
DO 800 L=1,LTS
MUX(N1,L)=MU(N1,L)
800 CONTINUE
CALCULATE PHI AT INJECTOR, NOZZLE, MIDPOINT OF LINER
P1= CZERO
DZPI1= CZERO
DO 130 L=1,LTS
SUM1= CZERO
SUM2= CZERO
BV= BESVL(MHAT,L)
DO 120 NL=1,NTS
N= NL
RN= FLOAT(N)
SUM1= SUM1*AV
SUM2= SUM2 + AV*FLOAT((-1)**N)
P1L=P1L+AV*AV*COS(RN*PIXL)
D2P1L= D2P1L + AV*AV*P1L*SIN(RN*PIXL)
120 CONTINUE

A1(L)=SUM1
A2(L)=SUM2

130 CONTINUE

C CALCULATE BLINER FOR MIDPOINT OF LINE

PRES= GAMMA*(C10M*P1L - MACH*D2P1L)
BLINER = WA*PRES*BETAL

C CALCULATE BLINER FROM LINE GEOMETRY

IF (OPTION.LE.2.OR.OPTION.EQ.6) GO TO 3000
AK1 = LEFF*ROAP/WA*ACAV/WCAV/HR*ACAV
PRE1 = CABS(PRES)
RO1 = 1.
R01 = 0.

DO 133 I=1,100
F = SQRT(AK1 - (RO1/R01)**2 )
BASE= EPSIL*PRE1/F/GAMMA
RO2 = SQRT(RO1**2 - 1.5*AMF*RO1 + BASE )
IF(RO2.LT.1.0E-04 ) GO TO 134
R01 = RO1
133 CONTINUE

C CONTINUE

3000 CONTINUE

BLINER = WA*PRES/GAMMA/(RO1*CI*AK1)

IF (OPTION.LE.4) GO TO 3001

C CALCULATE NEW HETAT

VOL = CZERO
DO 45 NI=2,NTS
NM1= NI-1
IF (NM1.EQ.NMAT) GO TO 45
K1= NM1 - NMAT
K2= NM1 - NMAT
L1=(-1)**K1-1
L2=(-1)**K2-1
LSUM=1.*L2
IF (LSUM.EQ.0.) GO TO 45
C1= FLOAT(L1)/FLOAT(K1)
C2= FLOAT(L2)/FLOAT(K2)
SC= (C1 + C2)*FLOAT(NMAT)
VOL = VOL + MUX(NI,NMAT)*SC
45 CONTINUE

VOL = VOL*MACH*C10M*HFS)
VOL = VOL = MUX(NMAT+1,NMAT)*HFS*(MACH*FLOAT(NMAT)*PIL)**2*LENGTH
1/2.
VOL = VOL/NORM1
SN0Z= RETAN*GAMMA*C10M*HFS*2(LMAT)**2*FLOAT((-1)**NMAT)/NORM1
SLIN= BLINER*RESV/(NMAT+1)*COS(FLOAT(NMAT)*PIXL)/NORM1
SIN= GAMMA*C10M*HFS*111(NMAT/NORM1
BETA1= (OMEGA**2 - FTAN**2 - VOL -SN0Z -SLIN)/SINJ

3001 CONTINUE

C START THE LOOP FOR L SUMMATION

DO 500 L=1,LTS
LAMDA = BESRT(MHAT*L)**2
BVZ = BESVL(MHAT*L)**2

START DO LOOP FOR N SUMMATION

DO 500 NX = 1, NTS

INITIALIZE AND CALCULATE CONSTANTS FOR THIS SUMMATION

N = NX - 1
VOL = CZERO
ETA = LAMDA + (FLOAT(N)*PIL)**2
RES = BVZ*(LAMDA - FLOAT(MHAT)**2)*.5/LAMDA
EP = 0.5
IF (N.EQ.0) EP = 1
NORM = BES*EP*LENGTH
BASEW = OMEGA*OMEGETA

CALCULATE PROPAGATION TERMS

DO 40 NI = 2, NTS
NM1 = NI - 1
IF (NM1.EQ.N) GO TO 40
K1 = NM1 + N
K2 = NM1 + N
L1 = (-1)**K1
L2 = (-1)**K2
LSUM = L1 + L2
IF (LSUM.EQ.0) GO TO 40
C1 = FLOAT(L1)/FLOAT(K1)
C2 = FLOAT(L2)/FLOAT(K2)
SC = (C1 - C2)*FLOAT(NM1)
VOL = VOL - MUX(NI*L)*SC

40 CONTINUE
VOL = VOL*MACH*CIOM*BES
VOL = VOL - MUX(NX*L)*BES*(MACH*FLOAT(N)*PIL)**2*LENGTH/2.
VOL = VOL/NORM

CALCULATE NOZZLE INTEGRAL

SNOZ = BETAN*GAMMA*CIOM*BES*A2(L)*FLOAT((-1)**N)/NORM

CALCULATE LINER INTEGRAL

TO CARRY OUT THE INTEGRATION OVER THE LINER USE THE FOLLOWING TWO
CARDS, AND PUT A "C" IN THE FIRST COLUMN OF THE CARD FOR APPXIMATING THE INTEGRAL, HENCE MAKING IT A COMMENT CARD.

CALL LINER(NX, XB, PIL, L, NHAT, MUX, CIOM, BLINER, LTS, NTS, CZERO)
SLIN = GAMMA*BESVL(MHAT*L)*BETAL*BLINER/NORM

TO APPROXIMATE THE LINER INTEGRAL BY EVALUATING AT THE MIDPOINT
AND MULTIPLYING BY THE WIDTH, USE THE FOLLOWING CARDS, AND PUT A
"C" IN THE FIRST COLUMNS OF THE TWO PRECEDING CARDS FOR CARRYING
OUT THE INTEGRAL, HENCE MAKING THEM COMMENT CARDS. "SUBROUTINE
LINER" NEED NOT BE COMPILED IF THE INTEGRAL IS TO BE APPROXIMATED.

SLIN = BLINER*BESVL(MHAT*L)*COS(FLOAT(N)*PIL)/NORM

CALCULATE INJECTOR INTEGRAL

SINJ = GAMMA*CIOM*BES*A1(L)/NORM
SINJB = SINJ*BETAIN

CALCULATE NEW MU TERM

F1 = VOL + SINJB + SLIN + SNOZ
IF (N.EQ.NHAT) GO TO 430
MNU(NX*L) = F1/NORM/BASEW
GO TO 500

430 CONTINUE
FOR PRIMARY MODE CALCULATE MU TERM AND NEW OMEGA

\[ \text{MU(NX*L)} = \text{CMPLX}(1./\text{NORM},0.) \]

IF (OPTION.LE.4) OMEGAN = CSQRT(F1 * ETA)

500 CONTINUE

ID = ID + 1
IK = IFIX(REAL(ID)/REAL(K))
RK = REAL(ID)/REAL(K)

IF (OPTION.LE.4) GO TO 4000

CHECK FOR CONVERGENCE ON BETAI

CE = BETAIN - RETAI
ERR1 = ABS(CE)/ABS(BETAIN)*100.
ANG2 = ATAN(IMAG(BETAIN)/REAL(BETAIN))
ERR2 = ABS(ANG2 - ANG1)

IF (ID.GE.IDMAX) GO TO 500

500 CONTINUE

WRITE (6,365)

545 CONTINUE

WRITE (6,666) ID,BETAIN,ERR1,ERR2
GO TO 550

CHECK FOR CONVERGENCE ON OMEGA

4000 CONTINUE

CE = OMEGAN - OMEGA
ERR1 = 0.0

IF (ABS(CE).EQ.0.0) GO TO 43
ERR1 = ABS(CE)/ABS(OMEGAN)*100.
ANG2 = ATAN(IMAG(OMEGAN)/REAL(OMEGAN))

43 CONTINUE

ERR2 = ABS(ANG2 - ANG1)

IF (ID.GE.IDMAX) GO TO 500

WRITE (6,365)

550 CONTINUE

WRITE (6,666) ID,OMEGAN,ERR1,ERR2

IF (ID.GE.IDMAX) WRITE (6,390)

PRINT OUT FINAL MU MATRIX

WRITE (6,2)
WRITE (6,110)
DO 106 N1=1,NTS
N = N1 - 1
WRITE (6,304) N,(MU(N1:L),L=1:LTS)
106 CONTINUE

CALCULATE AND PRINT EQUIVALENT BETAI IF CALCULATED FROM GEOMETRY

IF (OPTION.LE.3 AND OPTION.LE.5) WRITE (6,501) RO,AK
IF (OPTION.LE.3 AND OPTION.LE.5) BETAI = BLINER/WA/PRES
IF (OPTION.LE.3 AND OPTION.LE.5) WRITE (6,370) BETAI
CALCULATE AND PRINT EITHER N*, TAU OR BETAI WHICHEVER IS APPROPRIATE.

IF (OPTION.NF.1.AND.OPTION.NK.3) WRITE (6,380) RETAIN
IF (OPTION.LQ.2.OR.OPTION.FQ.4) GO TO 710
WR = REAL (OMEGAN)
RI = AIMAG (OMEGAN)
BI = REAL (BETAI)
IN = -((BIR - MACH/GAMMA)**2 + HII**2) / (2*MACH*(BIH - MACH/GAMMA))
DO 11 N = 1,3
RN = FLOAT (N) - 1.
TAU = ABS (ASIN (-BI/MACH/INO) + RN*PI)/WR
BETAI = MACH*(1./GAMMA - INO - (1. - CEXP (-CI*OMEGAN*TAU)))
ERR1 = ABS (BIR - REAL (BETAI))/BI
ERR2 = ABS (BI - AIMAG (BETAI))/BI
IF (ERR1.LE.0.30.AND.ERR2.LE.0.30) GO TO 19
RN = RN
TAU = ABS (ASIN (-BI/MACH/INO) + RN*PI)/WR
BETAI = MACH*(1./GAMMA - INO - (1. - CEXP (-CI*OMEGAN*TAU)))
ERR1 = ABS (BIR - REAL (BETAI))/BI
ERR2 = ABS (BI - AIMAG (BETAI))/BI
IF (ERR1.LE.0.30.AND.ERR2.LE.0.30) GO TO 19
11 CONTINUE
WRITE (6,3)
GO TO 710
19 CONTINUE
WRITE (6,386) RN, IN, TAU, BETAI
IF (WI.EQ.0.0) GO TO 710
RN = -3.0
18 CONTINUE
RN = RN + 1.0
IF (RN.GE.3.5) GO TO 17
IN = IN
TAU = TAU
DO 16 I = 1,260
TERM = HI - MACH/GAMMA + MACH*IN
TAU = ABS (ATAN (-HII/TERM) + RN*PI)/WR
IN = TERM/MACH*EXP(WI*TAU)*COS(WRI*TAU)
IF (ABS (T(I,TAU) - LTE.E-5.AND.ABS (IN - INI)*LTE.E-5.AND.I3.F.5))
GO TO 16
IF (I.GE.5.AND.IN.LT.0.0) GO TO 16
IF (I.GE.5.AND.TAU.LT.0.0) GO TO 16
IN = IN
TAU = TAU
15 CONTINUE
14 CONTINUE
IF (IN.LT.0.0.WR.TAU.LT.0.0) GO TO 18
BETAIN = MACH*(1./GAMMA - IN - (1. - CEXP (-CI*OMEGAN*TAU)))
ERR1 = ABS (BIR - REAL (BETAIN))/BI
ERR2 = ABS (BI - AIMAG (BETAIN))/BI
IF (ERR1.GE.0.15.OR.ERR2.GE.0.15) GO TO 18
WRITE (6,385) RN, IN, TAU, BETAIN
GO TO 710
17 CONTINUE
SWI = SIN(WRI*TAU0)
CWI = COS(WRI*TAU0)
EWT = EXP(WRI*TAU0)
ECT = EWT*CWI + 1.
WT = WR*TAN(WRI*TAU0)*WI
TAU1 = (BIR/MACH/ECT + BI/MACH/EWT/SWIT - 1./GAMMA/EC1)/(INO
1 IN) = BIR/MACH/EWT/SWT - INO - INO*TAU1*WT
TAU = TAU0 + TAU1
IN = IN0 + IN
BETAIN = MACH*(1./GAMMA - IN - (1. - CEXP (-CI*OMEGAN*TAU)))
WRITE (6,5) IN, TAU, BETAIN
CALCULATE RUNNING TIME FOR THIS SET OF DATA
710 CONTINUE
CALL SECOND (TEND)
THIS PROBLEM HAS NOT CONVERGED

THE SLOT IMPEDANCES ARE:

\[ 10 \times 3 \times 0.5 \times 613.6 \times 613.6 \times 15 \times 10.6 \times 5 \times 4 \]

\[ 501 \text{ FORMAT} \ (20 \text{H THE SLOT IMPEDANCES ARE}, \ 10 \times 3 \times 0.5 \times 613.6 \times 613.6 \times 15 \times 10.6 \]

\[ 666 \text{ FORMAT} \ (2X,13 \text{H THE SLOT IMPEDANCES ARE}, \ 10 \times 3 \times 0.5 \times 613.6 \times 613.6 \times 15 \times 10.6 \]

\[ 900 \text{ FORMAT} \ (5X,3H THE SLOT IMPEDANCES ARE}, \ 10 \times 3 \times 0.5 \times 613.6 \times 613.6 \times 15 \times 10.6 \]

\[ 902 \text{ FORMAT} \ (5X,39H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 904 \text{ FORMAT} \ (5X,42H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 906 \text{ FORMAT} \ (5X,38H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 908 \text{ FORMAT} \ (5X,35H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 910 \text{ FORMAT} \ (5X,36H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 912 \text{ FORMAT} \ (5X,35H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 914 \text{ FORMAT} \ (5X,36H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 916 \text{ FORMAT} \ (5X,55H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 920 \text{ FORMAT} \ (5X,37H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 922 \text{ FORMAT} \ (5X,38H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 924 \text{ FORMAT} \ (5X,34H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 926 \text{ FORMAT} \ (5X,35H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 930 \text{ FORMAT} \ (5X,109H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 940 \text{ FORMAT} \ (5X,89H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 942 \text{ FORMAT} \ (5X,84H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 950 \text{ FORMAT} \ (5X,61H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 960 \text{ FORMAT} \ (5X,90H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 966 \text{ FORMAT} \ (5X,91H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 970 \text{ FORMAT} \ (5X,70H THIS PROBLEM HAS NOT CONVERGED} \]

\[ 980 \text{ FORMAT} \ (5X,79H THIS PROBLEM HAS NOT CONVERGED} \]
FUNCTION BESRT(M,L)

THIS FUNCTION SUBROUTINE IS A TABLE OF THE ROOTS OF THE BESSEL
FUNCTION OF THE FIRST KIND.

DIMENSION A(4,10)
DIMENSION B(1,10)
IF (M.EQ.0) GO TO 10
A(1,1) = 1.84118378
A(1,2) = 5.33144277
A(1,3) = 8.5631637
A(1,4) = 11.7060049
A(1,5) = 14.86358863
A(1,6) = 18.01552786
A(1,7) = 21.16436986
A(1,8) = 24.31132666
A(1,9) = 27.45705057
A(1,10) = 30.60192297
A(2,1) = 3.05423693
A(2,2) = 6.70613119
A(2,3) = 9.96946782
A(2,4) = 13.17037086
A(2,5) = 16.34752232
A(2,6) = 19.51291278
A(2,7) = 22.67158177
A(2,8) = 25.82603714
A(2,9) = 28.97767277
A(2,10) = 32.12732702
A(3,1) = 4.20110894
A(3,2) = 8.01523660
A(3,3) = 11.34592431
A(3,4) = 14.58584879
A(3,5) = 17.78877477
A(3,6) = 20.9724764
A(3,7) = 24.14899743
A(3,8) = 27.3105793
A(3,9) = 30.47026881
A(3,10) = 33.62694918
A(4,1) = 5.31753313
A(4,2) = 9.28239629
A(4,3) = 12.68190844
A(4,4) = 15.96410704
A(4,5) = 19.19602880
A(4,6) = 22.40103227
A(4,7) = 25.58975968
A(4,8) = 28.76783622
A(4,9) = 31.93853934
A(4,10) = 35.10391668
BESRT = A(M,L)
RETURN
10 CONTINUE
A(1,1) = 0.00000000
A(1,2) = 3.83170597
A(1,3) = 7.01558667
A(1,4) = 10.17346814
A(1,5) = 13.32369194
A(1,6) = 16.47063005
A(1,7) = 19.61586855
A(1,8) = 22.76008438
A(1,9) = 25.90367209
A(1,10) = 29.04682853
BESRT = B(1,L)
RETURN
END
FUNCTION BESVL(M+L)

THIS FUNCTION SUBROUTINE IS A TABLE OF THE VALUES OF THE BESSEL
FUNCTION OF THE FIRST KIND.

DIMENSION A(4*10)
DIMENSION C(1*10)
IF (M.EQ.0) GO TO 10
A(1,1) = 0.58166522
A(1,2) = -0.3612620
A(1,3) = 0.27320994
A(1,4) = -0.23330444
A(1,5) = 0.20701256
A(1,6) = -0.1801749
A(1,7) = 0.17365957
A(1,8) = -0.16183471
A(1,9) = 0.15298079
A(1,10) = -0.14424290
A(2,1) = 0.48698684
A(2,2) = -0.31353046
A(2,3) = 0.25474416
A(2,4) = -0.22085158
A(2,5) = 0.19793743
A(2,6) = -0.18101004
A(2,7) = 0.16783553
A(2,8) = -0.15195178
A(2,9) = 0.14036767
A(2,10) = -0.14087833
A(3,1) = 0.434392763
A(3,2) = -0.291154413
A(3,3) = 0.240731758
A(3,4) = -0.21066204
A(3,5) = 0.190429022
A(3,6) = -0.175058405
A(3,7) = 0.16295406
A(3,8) = -0.153102909
A(3,9) = 0.144866574
A(3,10) = -0.137844513
A(4,1) = 0.3996514545
A(4,2) = 0.2743809499
A(4,3) = -0.229590468
A(4,4) = 0.202763849
A(4,5) = 0.184029896
A(4,6) = -0.169685816
A(4,7) = 0.158653572
A(4,8) = -0.149451156
A(4,9) = 0.141714307
A(4,10) = -0.13508328
RESVL = A(M+L)
RETURN
10 CONTINUE
C(1,1) = 1.0000000
C(1,2) = 0.4027588095
C(1,3) = 0.301128303
C(1,4) = 0.249704877
C(1,5) = 0.21835947
C(1,6) = -0.19645371
C(1,7) = 0.18063375
C(1,8) = 0.16718460
C(1,9) = 0.15678485
C(1,10) = -0.14801108
RESVL = C(1,L)
RETURN
END

FND
SUBROUTINE LINER(XA, XB, PIL, L, N, MHT, MUX, CIOM, MACH, BLINER, LTS, INTS, CZERO)
* THIS SUBROUTINE IS USED TO CARRY OUT THE INTEGRATION OVER THE
* LINER
REAL NMNP, NPNP, MACH
COMPLEX MUX, CIOM, BLINER, SUM1, SUM2, CZERO
DIMENSION MUX(50, 10)
BLINER = CZERO
DO 47 LP = 1, LTS
SUM1 = CZERO
SUM2 = CZERO
DO 45 NI = 1, NTS
RN = FLOAT(NI)
RNPIL = RN * PIL
IF (NI.EQ.0) GO TO 42
NMNP = FLOAT(NP) - NI * PIL
NPNP = FLOAT(NP + NI * PIL)
SUM1 = SUM1 + MUX(NI, LP) * ((SIN(NMNP * XB) / 2. / NMNP + SIN(NPNP * XB) / 2.)
   / NMNP - SIN(NMNP * XB) / 2. / NMNP - SIN(NPNP * XB) / 2. / NMNP)
SUM2 = SUM2 + MUX(NI, LP) * RNPIL * ((COS(NMNP * XB) / 2. / NMNP + COS(NPNP
   * XB) / 2. / NMNP - COS(NMNP * XB) / 2. / NMNP - COS(NPNP * XB) / 2. / NMNP)
GO TO 45
42 CONTINUE
IF (NP.EQ.0) GO TO 44
SUM1 = SUM1 + MUX(NI, LP) * ((XB - XA) / 2. * (SIN(2. * RNPIL * XR) -
   SIN(2. * RNPIL * XR)) / (4. * RNPIL))
SUM2 = SUM2 + MUX(NI, LP) * RNPIL * ((SIN(RNPIL * XB) ** 2 - SIN(RNPIL * XB) ** 2) / (2. * RNPIL))
GO TO 45
44 CONTINUE
SUM1 = MUX(N1, LP) * (XB - XA)
45 CONTINUE
BLINER = BLINER - BESVL(MHT, LP) * (CIOM * SUM1 + MACH * SUM2)
47 CONTINUE
RETURN
END